



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DESARROLLO DE UN CÓDIGO DE MÉTRICAS PARA EMPRESAS
DESARROLLADORAS DE SOFTWARE PEQUEÑAS”**

TESIS DE GRADO

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

Especialización: Sistemas de Información

Autores:

RAÚL GONZÁLEZ CARRIÓN

HENRY HERNÁNDEZ RENDÓN

Directora:

Ingeniera MÓNICA VILLAVICENCIO CABEZAS

GUAYAQUIL – ECUADOR

Año: 2006

DEDICATORIA

A mis padres Raúl González Jiménez y Rosa Carrión Armijos que han sido pilar fundamental en mi vida y me han enseñado valores que los tengo siempre presentes y los práctico; a mis hermanos Carlos Aurelio, Raúl Agustín y mi hermanita Celinita que están siempre a mi lado y sin ellos mi vida sería vacía; por todo el amor, toda la lealtad y la solidaridad que me brinda mi familia es que yo puedo superarme y seguir siempre adelante y lograr las metas que me propongo, gracias...

Raúl Vicente González Carrión

A Dios por haberme dado salud y la fuerza necesaria para seguir adelante y a mis padres Víctor Hernández Tello y Blanca Rendón Briones por haberme apoyado de manera incondicional en todos estos años de formación profesional.

Henry Xavier Hernández Rendón

AGRADECIMIENTO

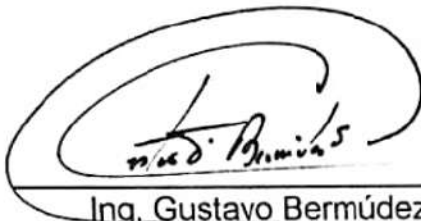
A la Escuela Superior Politécnica del Litoral y a la Facultad de Ingeniería en Electricidad y Computación, que durante estos cuatro años de estudio nos han enseñado y formado para el duro momento que se vive en el País y estamos seguros que gracias a la formación recibida saldremos adelante y nos superaremos en esta nueva etapa que nos toca vivir.

A la ingeniera Mónica Villavicencio Cabezas, por su gentileza al haber aceptado dirigir la presente tesis y que ciertamente con mucha solvencia profesional ha sabido otorgar su invaluable aporte para el éxito del presente trabajo.

A los docentes, ingeniera Verónica Macías e ingeniera Verónica Uquillas, quienes revisaron esta tesis y aportando con sus conocimientos de ingeniería de software, supieron contribuir invaluablemente con sus recomendaciones.

Al ingeniero Raúl González Jiménez, padre de uno de los autores de esta tesis, que pese a no ser especialista en esta rama y no disponer de mucho tiempo debido a su apretada agenda supo aportar con toda su experiencia profesional en cuanto a la revisión de la redacción de la presente tesis.

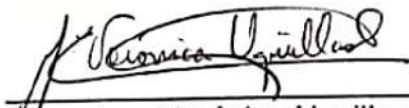
TRIBUNAL DE GRADUACIÓN



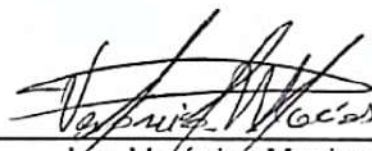
Ing. Gustavo Bermúdez
DECANO DE LA FIEC
PRESIDENTE



Ing. Mónica Villavicencio C.
DIRECTORA DE TESIS



Ing. Verónica Uquillas
MIEMBRO DEL TRIBUNAL




Ing. Verónica Macías.
MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)



Raúl V. González Carrión



Henry X. Hernández Rendón

RESUMEN EJECUTIVO

El presente estudio fue orientado a validar un instrumento de medición, el mismo que fue utilizado para la obtención de métricas, esto para empresas pequeñas ecuatorianas desarrolladoras de software. El estudio se orientó también a determinar los factores que inciden en la administración de proyectos de software; al efecto se han desarrollado cinco capítulos en donde se explica el proceso para cumplir lo propuesto.

Capítulo I

El capítulo uno referente a antecedentes y justificación de la tesis, tiene como propósito describir, la situación de la industria del software ecuatoriana con sus estudios realizados y las perspectivas futuras de la misma; además de la situación de la industria del software en otros países; explicar el plan piloto realizado previo al presente estudio; y, finalmente explicar la importancia del estudio realizado.

Capítulo II

El capítulo dos denominado marco teórico, hace una recopilación general de la literatura revisada para la realización del presente estudio.

Capítulo III

El capítulo tres denominado desarrollo contiene las hipótesis que se pretenden demostrar; los objetivos planteados; la metodología empleada para la recopilación de la información; explicar la población objetivo; con esto se procede a explicar en detalle la muestra seleccionada; al mismo tiempo se explica el plan de capacitación realizado; y, finalmente se explica el instrumento de medición conteniendo los 17 indicadores generados por el presente estudio.

Capítulo IV

El capítulo cuatro denominado proceso de aplicación del instrumento como su nombre lo indica se encarga de explicar en detalle lo realizado por los autores del presente estudio en las ciudades de Quito y Guayaquil, esto con el propósito de obtener la información necesaria para generar los indicadores desarrollados en el capítulo anterior.

Capítulo V

El capítulo cinco denominado análisis y presentación de resultados como su nombre lo indica se encarga de mostrar los resultados de los indicadores que se hayan podido formar al tiempo de realizar su respectivo análisis;

posteriormente se verificará las hipótesis planteadas en capítulo dos; y, finalmente se realiza la comparación con estudios similares encontrados que hayan sido realizados en otros países.

INDICE GENERAL

RESUMEN EJECUTIVO.....	I
INDICE GENERAL	IV
ABREVIATURAS	VIII
INDICE DE FIGURAS Y GRÁFICOS	X
INDICE DE TABLAS	XIII
INTRODUCCIÓN.....	XIV
CAPÍTULO I.....	16
1. ANTECEDENTES Y JUSTIFICACIÓN DE LA TESIS.....	16
1.1. Situación de la industria del software en el Ecuador.....	16
1.1.1. La industria de software en el Ecuador	16
1.1.2. Estudios realizados	18
1.1.3. Perspectivas futuras para el desarrollo de esta industria..	26
1.2. Situación de la industria del software en otros países.....	28
1.2.1. Estudio México	28
1.2.2. Estudio España.....	31
1.3. Plan piloto	35
1.3.1. Descripción.....	35
1.3.2. Resultados	36
1.4. Justificación	37
1.4.1. Importancia del proyecto	37
CAPÍTULO II.....	38
2. MARCO TEÓRICO	38
2.1. Métricas	38
2.1.1. Conceptos básicos de métricas.....	38
2.1.2. Métricas de software	39
2.1.3. Clasificación de métricas.....	42
2.1.4. Diferentes enfoques de métricas	43
2.1.5. Las métricas en el proceso y dominio del proyecto	46

2.2.	Estándares de calidad	47
2.2.1.	ISO 9001	47
2.2.2.	CMM	50
2.3.	Proyecto y aplicación de software	52
2.4.	Requerimientos del software	53
2.4.1.	Requerimientos funcionales	54
2.4.2.	Requerimientos no funcionales	54
2.5.	Pruebas del software	57
2.6.	GQIM	61
2.6.1.	¿Qué es el GQIM?	61
2.6.2.	Propósito del GQIM	61
2.6.3.	Pasos para desarrollar el GQIM	62
2.7.	Herramientas de interpretación de datos	66
2.7.1.	Diagramas de dispersión	66
2.7.2.	Diagramas causa-efecto.....	67
2.7.3.	Histogramas	68
2.7.4.	Gráficas de barra.....	69
2.7.5.	Cartas de control.....	69
2.8.	Análisis Univariado.....	70
2.9.	Análisis Multivariado.....	71
CAPÍTULO III		76
3.	DESARROLLO.....	76
3.1.	Hipótesis.....	76
3.2.	Objetivos	78
3.3.	Metodología de recopilación de la información	79
3.4.	Población objetivo	80
3.5.	Muestra para aplicación de Instrumento	82
3.6.	Plan de capacitación	83
3.7.	Instrumento de medición	84
3.7.1.	Definición	84
3.7.2.	Antecedentes	84
3.7.3.	Plantilla de levantamiento de información.....	85

3.7.4. Indicadores de medición.....	87
CAPÍTULO IV.....	130
4. PROCESO DE APLICACIÓN DEL INSTRUMENTO	130
4.1. Desarrollo de un instrumento en Access para la recopilación de datos	131
4.2. Plan de capacitación para empresas desarrolladoras de software	146
4.3. Diagrama GQIM explicativo	147
4.4. Transferencia de conocimientos y visitas a empresas.....	148
4.4.1. Sesión 1: “Presentación y explicación del proyecto de investigación”	148
4.4.2. Sesión 2: “Capacitación sobre la herramienta de recolección de datos” . “Aplicación de una encuesta”	150
4.4.3. Sesión 3: “Verificación de datos recolectados y recepción de sugerencias de la herramienta de recolección”	151
4.4.4. Sesión 4: “Monitoreo de la actividad de recolección y verificación de los datos”	151
CAPÍTULO V	152
5. ANÁLISIS Y PRESENTACIÓN DE RESULTADOS	152
5.1. Análisis de indicadores obtenidos.....	152
5.2. Prueba para verificar Hipótesis.....	197
5.3. Comparación con estudios de otros países	208
CONCLUSIONES Y RECOMENDACIONES	215
BIBLIOGRAFÍA.....	227
ANEXO 1: Plan de Capacitación (Primera versión)	234
ANEXO 2: Plan de métricas y Plantillas de levantamiento de información (Plan Piloto)	235
ANEXO 3: Plan de métricas y Plantillas de levantamiento de información (Desarrollado).....	236
ANEXO 4: Plan de Capacitación (Última versión)	237
ANEXO 5: Diagrama GQIM explicativo	238
ANEXO 6: Metodología proceso GQIM.....	239

ANEXO 7.1: Relación entre variables esfuerzo en el desarrollo versus complejidad técnica y versus complejidad del negocio.	250
ANEXO 7.2: Relación entre variables esfuerzo del cliente y esfuerzo del usuario versus satisfacción del cliente y satisfacción del usuario.	251
ANEXO 7.3: Relación entre variables esfuerzo en el desarrollo versus número de personas involucradas.	252
ANEXO 7.4: Relación entre variables defectos y fallas versus complejidad técnica y versus complejidad del negocio.	253
ANEXO 7.5: Relación entre variables esfuerzo en documentación versus fallas.	254
ANEXO 8: Complejidad ciclomática.....	255
ANEXO 9: Puntos de Función.....	257

ABREVIATURAS

Las abreviaturas presentadas en la siguiente tesis son las siguientes:

- AESOFT:** Asociación Ecuatoriana de Software.
- CAF:** Corporación Andina de Fomento.
- CASE:** Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador)
- CMM:** Capability Maturity Model (Modelo de Capacidad y Madurez).
- CMMI:** Capability Maturity Model Integration (Modelo de Capacidad y Madurez Integrado).
- CORPEI:** Corporación de Promoción de Exportación e Inversiones del Ecuador.
- GQIM:** Goals Questions Indicators and Measures (Proceso de medición dirigido a objetivos)
- IEEE:** The Institute of Electrical and Electronics Engineers, (Instituto de Ingenieros Eléctricos y Electrónicos).
- ISO:** International Organization for Standardization (Organización Internacional para la Estandarización).
- LDC:** Líneas de código.
- MPYMES:** Micro, pequeñas y medianas empresas.
- MSF:** Microsoft Solution Framework.
- PDCA:** Plan, Do, Check, Act (Planificar, Hacer, Verificar, Actuar)

- PF:** Punto de Función.
- PIB:** Producto Interno Bruto.
- PROSOFT:** Programa para el Desarrollo de la Industria del Software.
- PSP:** Personal Software Process (Proceso personal del software)
- SEI:** Software Engineering Institute.
- TI:** Tecnologías de la Información.
- VLIR:** Vlaamse Interuniversitaire Raad

INDICE DE FIGURAS Y GRÁFICOS

Figura 1.1. Mapa del sector de los desarrolladores de software [1].....	18
Gráfico 1.1. Número de empresas por ciudad [2]	19
Gráfico 1.2. Proporción de empresas según ciudad y tamaño [2].....	20
Gráfico 1.3. Duración de los Proyectos [4].....	24
Gráfico 1.4. Proyectos que se interrumpieron [4].....	24
Gráfico 1.6. Personal estimado (Vs.) Real [4]	25
Gráfico 1.7. Varianza de costo y tiempo estimados (Vs.) los reales [4].....	25
Gráfico 1.8. Exportaciones de TI por segmento (<i>millones de dólares</i>) [7].....	32
Gráfico 1.9. Participación de mercado español de TI, 2000 [7]	33
Gráfico 1.10. Participación de mercado de software, 2000 [7].....	34
Gráfico 1.11. Producto Interno Bruto – Software [7]	35
Figura 2.1. Servicios Relacionados al Software [10].....	40
Figura 2.2. Tipos de requerimientos no funcionales [16]	56
Figura 2.3. Proceso GQIM [20]	65
Figura 2.4. Ejemplo de un diagrama de dispersión [21]	66
Figura 2.5. Ejemplo de un diagrama causa-efecto	67
Figura 2.6. Ejemplo de un histograma [21]	68
Figura 2.7. Ejemplo de una gráfica de barra [21]	69
Figura 2.8. Ejemplo de una carta de control [21].....	70
Figura 2.9. Diagramas de dispersión con correlaciones negativa y positiva perfecta [22].....	72
Figura 2.10. Diagramas de dispersión que muestran correlación cero, débil y fuerte [22].....	73
Figura 2.11. Esquema de intensidad y dirección de r [22]	73
Figura 2.12. Distribución t Student para prueba de coeficiente correlación [22]	75
Figura 4.1. Ingreso de fase	132
Figura 4.2. Ingreso de roles.....	133

Figura 4.3. Ingreso de personas	134
Figura 4.4. Datos generales del proyecto	136
Figura 4.5. Definición de tareas/entregables.....	139
Figura 4.6. Filtro de tareas/entregables	139
Figura 4.7. Reporte de problemas	142
Figura 4.8. Tareas por asignar detalles	143
Figura 4.9. Líneas de código producidas	144
Figura 4.10. Complejidad del software	145
Gráfico 5.1.1. Satisfacción del Usuario por Proyectos	152
Gráfico 5.1.2. Satisfacción del Cliente por Proyectos	154
Gráfico 5.1.3. Satisfacción del Usuario versus Esfuerzo del Usuario	157
Gráfico 5.1.4. Satisfacción del Cliente versus Esfuerzo del Cliente.....	158
Gráfico 5.2.1. Porcentaje de error en la estimación del esfuerzo.....	160
Gráfico 5.3.1. Porcentaje del esfuerzo por persona en las diferentes fases	162
Gráfico 5.4.1. Porcentaje de esfuerzo empleado en documentación	164
Gráfico 5.5.1. Complejidad del negocio del cliente.....	166
Gráfico 5.5.2. Complejidad técnica del proyecto.....	168
Gráfico 5.5.3. Complejidad de los proyectos.....	171
Gráfico 5.6.1. Porcentaje de error en la estimación del costo.....	172
Gráfico 5.7.1. Costo promedio hora real.....	174
Gráfico 5.8.1. Porcentaje consolidado de las inspecciones en las fases	177
Gráfico 5.8.2 Distribución de las inspecciones por fases	178
Gráfico 5.8.3. Porcentaje de inspecciones realizadas por fase en los proyectos.....	179
Gráfico 5.9.1. Porcentaje de variación de los requerimientos	181
Gráfico 5.10.1. Complejidad del código fuente	183
Gráfico 5.11.1. Indicador ponderado de la complejidad y demora en el aprendizaje	184
Gráfico 5.13.2. Porcentaje consolidado de fallas respecto al número total de tareas realizadas.....	186
Gráfico 5.14.1. Porcentaje consolidado de defectos respecto a las tareas, identificados por su severidad	188
Gráfico 5.14.3. Porcentaje de defectos más ocurrentes	189

Gráfico 5.14.4. Porcentaje de fallas más ocurrentes	189
Gráfico 5.15.1. Porcentaje de tiempo utilizado en corrección de defectos	191
Gráfico 5.15.2. Porcentaje de tiempo utilizado en corrección de fallas.....	191
Gráfico 5.16.1. Eficiencia en atención de defectos	193
Gráfico 5.16.2. Eficiencia en atención de fallas	193
Gráfico 5.17.1. Promedio de defectos por hora de programación	195
Gráfico 5.18. Porcentaje de error en estimación del esfuerzo versus porcentaje de esfuerzo empleado en planificar	198
Gráfico 5.19. Esquema de distribución t Student para hipótesis 1	202
Gráfico 5.20. Porcentaje de esfuerzo en inspecciones en fases iniciales versus porcentaje de esfuerzo en solucionar fallas.....	204
Gráfico 5.21. Esquema de distribución t Student para hipótesis 2	207
Gráfico 5.3.1. [4] obtenida por plan piloto	212
Gráfico 5.3.2. obtenido por este estudio	212
Gráfico 5.3.3. [4] obtenida por plan piloto	213
Gráfico 5.3.4. obtenido por este estudio	213
Gráfico 5.3.5 [4] obtenida por plan piloto	213
Gráfico 5.3.6. obtenido por este estudio	213

INDICE DE TABLAS

Tabla 1.1. Familiaridad y estado de uso de estándares de calidad según el tamaño de las empresas desarrolladoras [2]	21
Tabla 1.2. Clasificación total por tamaño de empresa de la industria mexicana de software [6]	28
Tabla 1.3. Número de empleados total [6].....	29
Tabla 1.4. Ventas de software totales [6].....	30
Tabla 1.5. Exportaciones de software [6]	30
Tabla 1.6. Exportaciones de TI por segmento, 1999 – 2000 [7].....	32
Tabla 1.7. Participación de mercado español de TI, 1999 – 2000 [7]	33
Tabla 5.1. Esfuerzo empleado en planificación versus error en la estimación del esfuerzo	199
Tabla 5.2. Porcentaje de esfuerzo empleado en solucionar fallas versus porcentaje de esfuerzo inspecciones en las 4 fases iniciales	205

INTRODUCCIÓN

Estudios previos realizados en nuestro País respecto al área de ingeniería de software han demostrado que la gran mayoría de empresas desarrolladoras de software pequeñas no documentan el proceso de desarrollo, aspecto preocupante, ya que una adecuada documentación del mismo permitiría realizar mejores estimaciones para futuros proyectos, reduciendo las pérdidas económicas que causa una mala estimación y produciendo un software de calidad.

Por lo anterior es que hemos enfocado el objetivo del presente estudio a la validación y aplicación de un instrumento de medición que pueda ser utilizado por empresas pequeñas ecuatorianas desarrolladoras de software, el mismo que está basado en un estudio previo realizado para empresas desarrolladoras de software de tamaño mediano y grande; en este sentido, se ha estructurado el presente estudio en cinco capítulos, los mismos que en síntesis abarcan los aspectos posteriormente descritos.

El capítulo uno, relacionado con los antecedentes y la justificación de la tesis trata los estudios realizados en el Ecuador, además de dar a conocer las perspectivas futuras del mismo; así también se explica la situación de la industria extranjera que desarrolladora de software, específicamente México y España; además se explica el estudio que sirvió de base para el presente, este es el “Plan Piloto”; y, finalmente la justificación del presente estudio.

El capítulo dos por su parte, enfoca la literatura revisada y necesaria para la comprensión del presente estudio.

Por su parte, el capítulo tres hace referencia al desarrollo, esto es, hipótesis, objetivos, metodología de recopilación de la información, población objetivo, selección de la muestra para la aplicación del instrumento de recolección de los datos, plan de capacitación y finalmente el instrumento de medición el cual lo conforman las plantillas y los indicadores generados por el presente estudio en base a lo sugerido por el “Plan Piloto”.

El capítulo cuatro, se encarga describir el proceso utilizado para la aplicación del instrumento desarrollado y descrito en el capítulo anterior. En el mismo se describe la herramienta con la cual se recopila los datos, la validación del plan de capacitación realizado, el desarrollo de un diagrama explicativo y finalmente las diferentes sesiones realizadas en las empresas.

Finalmente, el capítulo cinco presenta la información recolectada con su respectivo análisis, además de la demostración de hipótesis planteadas en el capítulo tres, para finalizar con la comparación de los resultados obtenidos respecto a estudios similares que hayan sido realizados en otros países.

CAPÍTULO I

1. ANTECEDENTES Y JUSTIFICACIÓN DE LA TESIS

1.1. Situación de la industria del software en el Ecuador

En este apartado pretendemos dar a conocer la situación en la que se desarrollan las empresas que actualmente están proveyendo software en nuestro país. Más adelante, en este documento, realizaremos una comparación con empresas extranjeras dedicadas al desarrollo de software.

En este capítulo nos enfocaremos en conocer la industria ecuatoriana desarrolladora de software basada en estudios previos realizados en el Ecuador y al final daremos las perspectivas futuras de la misma.

1.1.1. La industria de software en el Ecuador

El software ha demostrado en el mundo ser un motor del crecimiento económico y social de los países, un inyector para el aumento de productividad y un generador de plazas de trabajo. Así mismo es la industria de todas las industrias, su desarrollo impacta en la competitividad de la Banca, Municipios, Instituciones Educativas; es decir impacta en todas las actividades tanto privadas como

gubernamentales. Finalmente el software del Ecuador cuenta con un potencial exportador real demostrado en múltiples casos de éxito de la empresa privada que han sido reconocidos en los mercados internacionales [1].

El Primer Estudio de la Industria de Software del Ecuador desarrollado por la Asociación Ecuatoriana de Software (AESOFT) en Junio del 2005 logró obtener un perfil de la industria del software, el mismo que logró identificar 223 empresas con un total en ventas de 62 millones de dólares generando 2600 empleos directos fijos, 633 empleos directos a destajo y 3988 empleos indirectos aportando fiscalmente con 21.6 millones de dólares y con exportaciones de 10.7 millones [1].

La industria del software genera ventas que equivalen al 0.35% del Producto Interno Bruto (PIB), las mismas que corresponden al 2.1% de los ingresos no petroleros. Sus ventas al exterior representan el 1.1% de las exportaciones no petroleras. Finalmente la industria del software contrata al 0.12% de la fuerza laboral de nuestro país [1].

En el mismo estudio se generó un mapa que muestra las 3 ciudades principales del Ecuador en las cuales se concentra la mayor cantidad de empresas relacionadas con la actividad del software.

Figura 1.1. Mapa del sector de los desarrolladores de software [1]



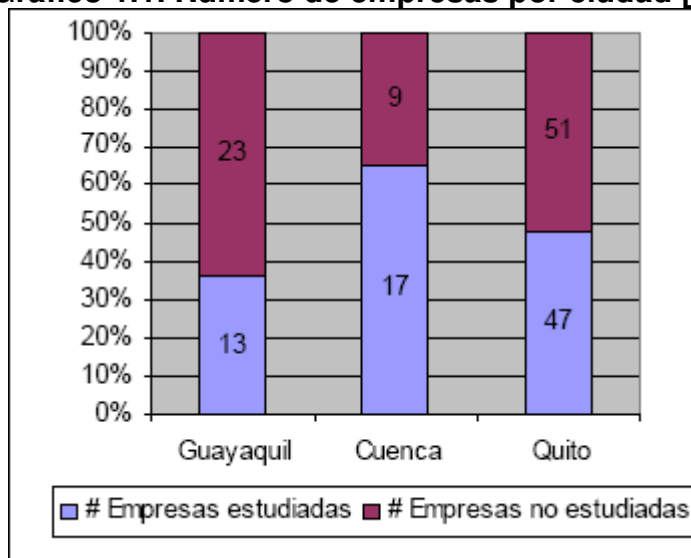
1.1.2. Estudios realizados

En otros países como Estados Unidos, México, Argentina y España, el campo de la Ingeniería de Software está bastante avanzado, no así en nuestro país, en donde la Ingeniería de Software recién está dando sus primeros pasos, prueba de esto es la carencia de estudios en este campo. Tres estudios realizados por el proyecto VLIR-ESPOL – Componente 8 (Ingeniería de Software), muestran aspectos que justifican plenamente y nos sirven de base para proponer la elaboración de nuestra investigación.

Es importante recalcar que estos estudios fueron llevados a cabo en orden y su objetivo era profundizar el conocimiento sobre el proceso de desarrollo utilizado por las empresas ecuatorianas desarrolladoras de software.

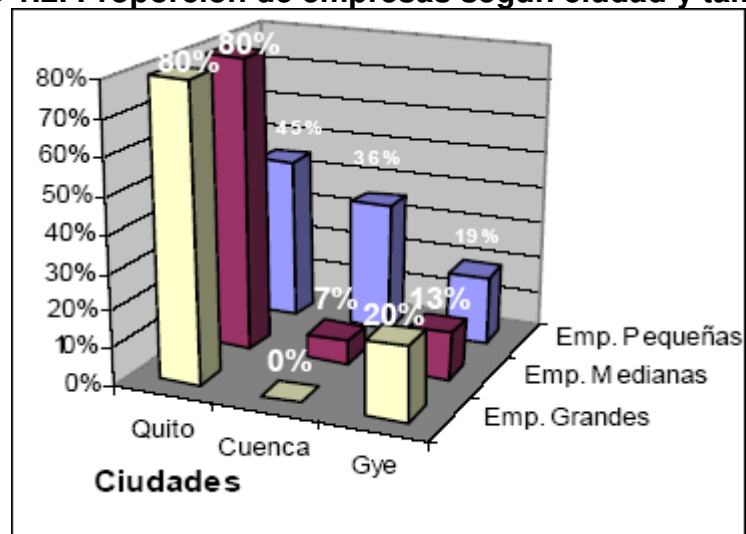
El Primer estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en Guayaquil, Quito y Cuenca se llevó a cabo en Octubre del 2003 con el objetivo de conocer las características más relevantes de las mismas. Luego del proceso de depuración de la base de datos de las empresas ecuatorianas desarrolladoras de software proporcionada por la CORPEI, se determinó que en Ecuador existen aproximadamente 160 empresas, distribuidas de la siguiente manera: 36 en Guayaquil, 98 en Quito, y 26 en Cuenca [2].

Gráfico 1.1. Número de empresas por ciudad [2]



Se observó que en Guayaquil la mayor parte de las empresas son pequeñas. En Cuenca, el 88.2% de las empresas son pequeñas mientras que en Quito la mayoría de las empresas son medianas (51%). De toda la población, la mayoría de las empresas grandes están asentadas en Quito como se muestra en el Gráfico 1.2.

Gráfico 1.2. Proporción de empresas según ciudad y tamaño [2]



Con respecto al conocimiento y uso de estándares de calidad en el desarrollo de software, el 94.8% de las empresas dijeron conocer acerca de ISO 9001, el 48% acerca de MSF y el 29.8% acerca de CMM. Los indicadores más importantes en este estudio estuvieron relacionados a la utilización de normas de calidad en las empresas [2].

Tabla 1.1. Familiaridad y estado de uso de estándares de calidad según el tamaño de las empresas desarrolladoras [2]

	Total		Emp. Pequeñas		Emp. Medianas		Emp. Grandes	
	No. Emp	%	No. Emp	%	No. Emp	%	No. Emp	%
Conocimiento								
ISO 9001	73	94,8	39	92,8	30	100	4	80
CMM	23	29,8	7	16,6	13	43,3	3	60
MSF	37	48	18	42,8	16	53,3	3	60
Estado de Uso								
Son utilizados	28	36,3	12	28,5	13	43,3	3	60
En el pasado se utilizaron	2	2,5	0	0	2	6,6	0	0
Actualmente en estudio	29	37,6	16	38	11	36,6	2	40
En el pasado estuvo en estudio	3	3,8	2	4,7	1	3,3	0	0
Nunca han utilizado	18	23,3	12	28,5	6	20	0	0

En el 2003 el 36.3% de las empresas encuestadas utilizaban estándares de calidad en el desarrollo de software, de los cuales, sólo el 24.6% correspondía a estándares internacionalmente reconocidos. Este indicador es un tanto deficiente con lo obtenido en otros países. Esto último podría ser el motivo de la poca penetración en el mercado internacional. Un resultado alentador es que el 37.6% de las empresas actualmente estarían analizando la posibilidad de implantar alguno. El 48% de las empresas conocen los múltiples beneficios proporcionados por la norma MSF. La norma ISO 9001 es la más conocida por las empresas ya que dicho estándar asegura que actividades como el compromiso de la dirección, evaluación del estado actual, planeación

cuidadosa, diseño, desarrollo, operación, monitoreo del progreso y buena administración del proyecto sean realizados en el proceso integral del desarrollo de software [2].

Este primer estudio exploratorio nos dio una primera vista de la realidad de las empresas de desarrollo de software. En un segundo estudio exploratorio que tuvo por nombre *“Aspectos de la calidad y dificultades en la gestión de proyectos de software”*, se profundizó un poco más en los diferentes factores que afectan al proceso de desarrollo.

En este segundo estudio exploratorio se encontró que la premura por terminar los proyectos es lo que impide a las empresas de software ecuatorianas documentar el proceso de desarrollo del software. Solo el 40% de las empresas siempre documentan el proceso durante el ciclo del desarrollo de software. Según el 78% de los casos, el principal motivo para no hacerlo es la premura por terminar el proyecto, seguido por un 14% que mostró poco interés en realizar esta actividad. La falta de documentación puede afectar al proyecto en curso así como a los proyectos futuros, ya que al no tener una referencia de cómo fueron construidos previamente los productos de software, seguramente los involucrados en el desarrollo de un

proyecto invertirán más tiempo y esfuerzo en reconocer sus diferentes componentes y la evolución de los mismos [3].

Alrededor de la mitad de las empresas no conocen el concepto de métricas de software y por lo tanto sus beneficios. El 68% de las empresas realizan diferentes mediciones del proceso de desarrollo de software pero sólo el 24% de éstas siempre toman decisiones usando estos datos [3].

El principal problema al que se enfrentan las empresas de software cuando se recopilan los requerimientos es el desconocimiento del cliente sobre el alcance del proyecto. A pesar de esto, la mayoría de las empresas, durante la etapa de requerimientos, hacen la estimación de factores como el tiempo, costos, recursos y tamaño del sistema. En la fase de diseño hacen nuevas estimaciones de los factores antes mencionados [3].

Finalmente un tercer estudio [4] fue realizado con el objetivo de ahondar más en el proceso de desarrollo de las empresas y obtener información más específica de los principales factores que afectan al mismo al igual que determinar el origen de dicha información. Este estudio fue realizado durante los meses de Abril y Mayo del 2006 en

40 empresas dedicadas al desarrollo de proyectos de software en las ciudades de Quito y Guayaquil, con el propósito de proporcionar información actualizada acerca del proceso de desarrollo de software. Cada empresa aportó datos relacionados a un solo proyecto de software que había sido terminado y entregado a su cliente.

La principal información obtenida fue que el 43% de los proyectos de software tienen una duración de 3 a 9 meses. El 54% de los proyectos sufrieron interrupciones. [4]

Gráfico 1.3. Duración de los Proyectos [4]

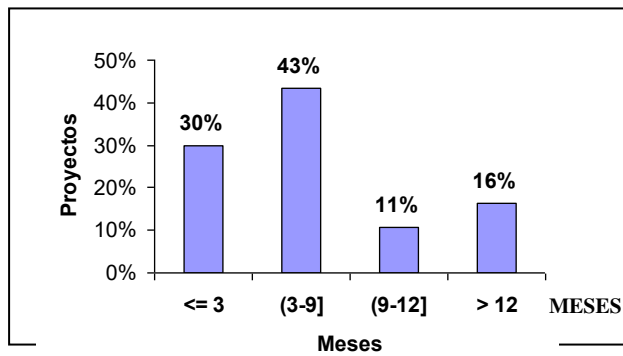
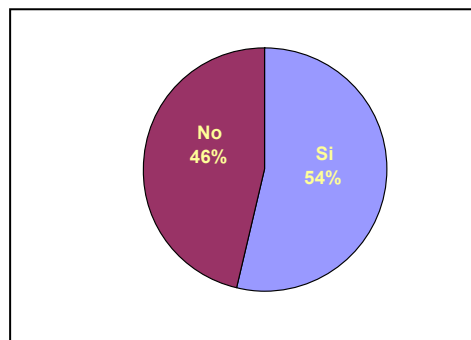


Gráfico 1.4. Proyectos que se interrumpieron [4]



Además se encontró que las empresas estiman menos tareas de las que realmente realizan, al igual que estiman menos personal del que realmente van a utilizar. [4]

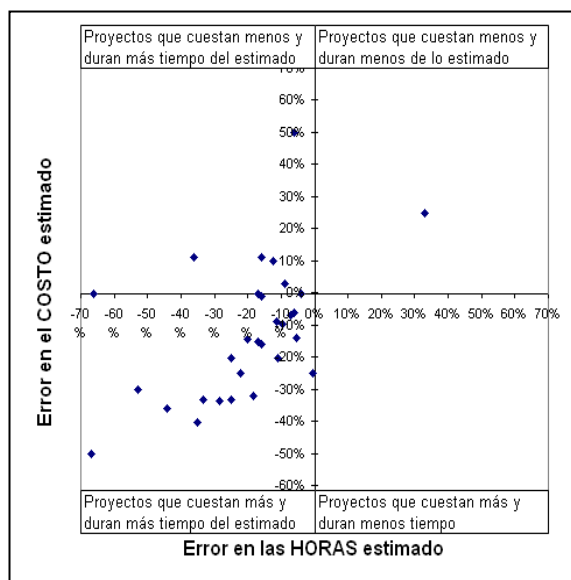
Gráfico 1.6. Personal estimado (Vs.) Real [4]



Este error en la estimación de las tareas y el personal conlleva a que los proyectos terminen costando más y durando más de lo planificado.

Lo que se muestra en el Gráfico 1.7. [4]

Gráfico 1.7. Varianza de costo y tiempo estimados (Vs.) los reales [4]



En la aplicación de este estudio se colocaron preguntas para validar de dónde obtenían las empresas los datos aportados al mismo, es decir si llevaban un registro ya sea en programas o en documentos de la información de su proceso de desarrollo o si los datos obtenidos provenían de estimaciones hechas por ellas. Lo que se obtuvo es que la mayoría de los datos provenían de estimaciones.

1.1.3. Perspectivas futuras para el desarrollo de esta industria

La industria de desarrollo de software en el Ecuador ha tenido un gran crecimiento en los últimos años, prueba de esto es que en el año 2000 el país se encontraba ubicado en el tercer puesto en América latina de países que exportan software. Sin embargo esta ubicación descendió para el año 2005 y el país se ubicó en el octavo lugar. Esto fue debido en gran parte a que empresas desarrolladoras que exportan software de otros países empezaron a obtener certificaciones de calidad y madurez de procesos reconocidas internacionalmente como lo son CMMI e ISO9001. [5]

Muchas empresas extranjeras que contratan software de otros países prefieren empresas que posean alguna certificación reconocida. Algunas empresas del Ecuador nos han sabido comunicar que han

perdido negocios porque las empresas contratistas prefirieron una empresa con certificación.

Dada esta realidad muchas empresas nacionales han puesto la mira en la obtención de una certificación pero lastimosamente los altos costos les impiden realizarlo. Sin embargo, hace poco la AESOFT (Asociación Ecuatoriana de Software), gracias a un convenio firmado con la CAF (Corporación Andina de Fomento), inició un proceso que permitirá a las empresas desarrolladoras de software ir a la implementación de CMMI. Para ello AESOFT, desarrolló un proyecto de participación de las empresas de desarrollo en un proceso de capacitación para la implementación de CMMI.

Al momento de escribir esta tesis 9 empresas del gremio están participando de dicho proceso: Kruger, Carrasco & Asociados, Yagé, Eikon, Bayteq, Tropitdatos, Grupo Mas y Macosa. Bajo la dirección del experto en CMMI, Marcelo Jenkins las empresas empezaron un período de capacitación sobre cómo implementar y cumplir las metas del modelo.

Gracias a que las universidades de nuestro país han incorporado las materias de Ingeniería de Software y Emprendimiento en sus

pénsums académicos, se espera un mayor auge de empresas de desarrollo de software pequeñas que satisfagan la demanda local.

1.2. Situación de la industria del software en otros países

El objetivo de esta sección es mostrarnos cómo se encuentra la industria de software en otros países que por su desarrollo económico y tecnológico sirven de referencia.

1.2.1. Estudio México

La Secretaría de Economía de México a través del Programa para el Desarrollo de la Industria de Software (PROSOFT) con el propósito de evaluar el estado actual del nivel de madurez y capacidad de procesos de las empresas de software y servicios relacionados en el Área Metropolitana de Monterrey, Nuevo León (AMM) y la Zona Metropolitana del distrito federal (ZMDF) procedió a realizar un diagnóstico durante el último trimestre del 2003 y el primer trimestre de 2004.

Tabla 1.2. Clasificación total por tamaño de empresa de la industria mexicana de software [6]

Tamaño	Número de empleados	Resultados	Porcentaje
Micro	1 – 10	48	39.0
Pequeña	11 – 50	53	43.0
Mediana	51 – 100	12	9.8
Grande	101 en adelante	10	8.2
Total		123	100

En lo relativo al tamaño de las empresas encuestadas de acuerdo con la clasificación por número de empleados de la Ley para la competitividad de la Micro, Pequeña y Mediana Empresas, el 92% son micro, pequeña y mediana empresas (MPyMEs). Esta clasificación por tamaño de empresas es similar a la composición que presenta la estructura industrial de México. [6]

Con base en el promedio de empleados por tamaño de empresas se calculó un total de 3686 empleos directos del total de empresas encuestadas. Dicha estimación se realizó mediante un promedio simple de los rangos considerados en el tamaño de empresas por número de empleados. La estimación de empleos generados permite inferir que las pequeñas empresas aportan más del 36% de los empleos, y en conjunto la micro, pequeña y mediana empresa aportan un poco menos del 70% de los empleos generados y sólo el 9% de las empresas grandes aportan el 30% de los empleos. [6]

Tabla 1.3. Número de empleados total [6]

Tamaño	Empresas	Promedio de empleados	Total de empleados	Porcentaje
Micro	50	5	250	6.8
Pequeña	54	25	1350	36.6
Mediana	13	75	975	26.4
Grande	11	101	1111	30.2
Total	128		3686	100%

Los resultados correspondientes a las ventas exclusivamente de software muestran que el 76% de las empresas tienen ingresos anuales promedio por ese concepto de menos de 2.5 millones de pesos y sólo el 4% de las empresas encuestadas facturan anualmente más de \$30 millones de pesos. (Tasa de conversión: 1 peso = 0.09 dólares).

Tabla 1.4. Ventas de software totales [6]

Ventas	Empresas	Porcentaje	Acumulado (menos a más)	Acumulado (más a menos)
Menos de 2.5	85	76	76	100
2.5 – 7	14	13	88	23
7 – 15	5	4	93	11
15 – 30	3	3	96	6
30 – 50	3	3	98	4
50 – 150	1	1	100	1
Total	112	100		

Los resultados correspondientes a las exportaciones exclusivamente de software muestran que 15 empresas realizan dicha actividad, esta cifra representa sólo el 12% del total de empresas encuestadas. Además es importante señalar que 8 empresas se ubican en la ZMDF (Zona Metropolitana del distrito federal) y 7 en el AMM (Nuevo León) . (Tasa de conversión: 1 peso = 0.09 dólares).

Tabla 1.5. Exportaciones de software [6]

Tamaño	Exportación de ventas totales (millones de pesos)						
	Menos 2.5	2.5 - 7	7 - 15	15 - 30	30 - 50	50 – 150	Total
Micro	2	2	0	0	0	0	4
Pequeña	2	3	2	0	0	0	7
Mediana	0	0	0	0	1	0	2
Grande	0	1	0	0	1	1	2
Total	4	6	2	0	2	1	15

1.2.2. Estudio España

Un estudio de mercado de las tecnologías de la información de España realizado por MINCOMEX y PROEXPORT de Colombia en el 2002, en el cual se encuentra incluida la industria del software española, presenta información interesante sobre la misma.

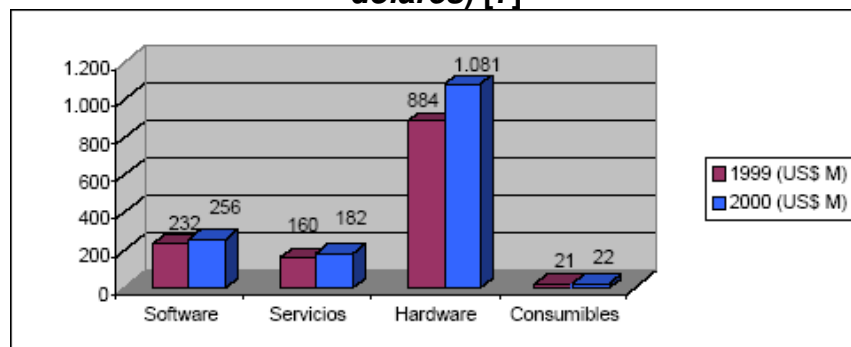
La información contenida en el estudio se obtuvo a través de la investigación y análisis de diversos documentos, artículos y estudios sobre los sectores español y europeo de las Tecnologías de la Información. Así mismo, se realizaron entrevistas con las principales asociaciones del sector y con un grupo representativo de las principales empresas españolas, al igual que con algunos expertos en el tema de las TI en España. [7]

Dentro del total de exportaciones del sector de TI en el año 2000, los segmentos que mayor participación presentaron fueron hardware y software, con un 70% y un 17% respectivamente. Estos datos muestran la importancia de estos dos segmentos dentro del comercio exterior del sector español de TI.

Tabla 1.6. Exportaciones de TI por segmento, 1999 – 2000 [7]

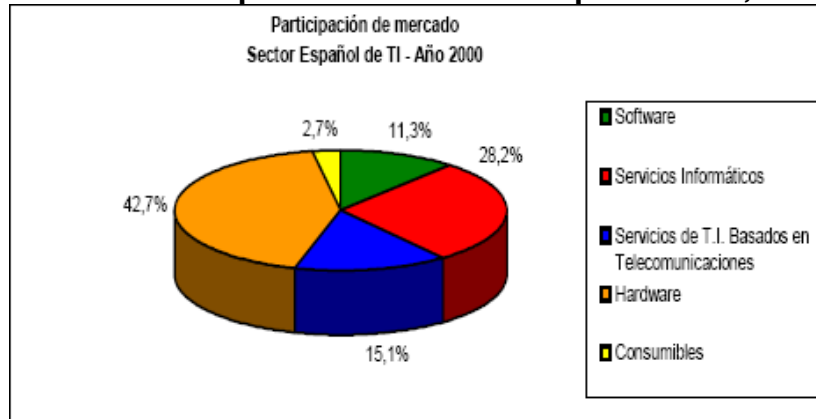
Segmento	1999	2000	Incremento %
Software	232	256	10,0%
Servicios	160	182	13,1%
Hardware	884	1.081	22,2%
Consumibles	21	22	1,4%
Total Exportaciones	1.298	1.539	18,6%

Gráfico 1.8. Exportaciones de TI por segmento (millones de dólares) [7]



La gráfica siguiente muestra la participación de los diferentes segmentos del sector en el año 2000. Se puede apreciar la importancia de los servicios, tanto informáticos como de telecomunicaciones, dentro del total.

Gráfico 1.9. Participación de mercado español de TI, 2000 [7]



Para tener una aproximación a la demanda real del sector, es decir la generada por los usuarios finales, en el cuadro siguiente se desglosan las cifras de las que se consideran como operaciones intermedias: la venta indirecta y la subcontratación de servicios.

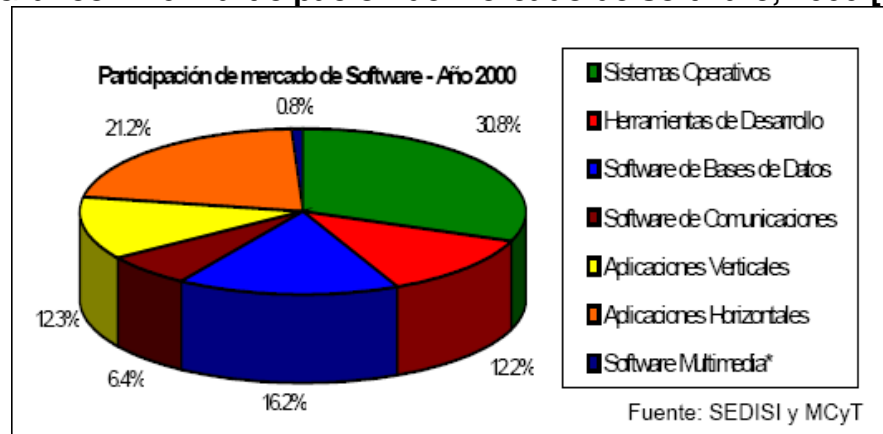
Tabla 1.7. Participación de mercado español de TI, 1999 – 2000 [7]

Segmento	1999 (US\$Mill)	2000 (US\$Mill)	Crecim.	Partic. en el PIB
Software	384	477	24,4%	35,6%
Servicios Informáticos	262	365	39,3%	10,9%
Servicios de TI Basados en Telecomunicaciones	-	-	-	-
Hardware	1.616	1.966	21,7%	38,7%
Consumibles	89	108	21,2%	33,5%
Total Operaciones intermedias	2.351	2.917	24,1%	24,5%

Como se observó en la gráfica de participación del mercado de TI de 2000, dentro del total el segmento de software tuvo una participación del 11,3%. Este porcentaje se encuentra por debajo de la media europea en donde el software representa el 21,0% evidenciando un

mercado aún en desarrollo, en donde la demanda debe consolidarse todavía. El comportamiento de las diferentes categorías que conforman el segmento de software ha sido:

Gráfico 1.10. Participación de mercado de software, 2000 [7]



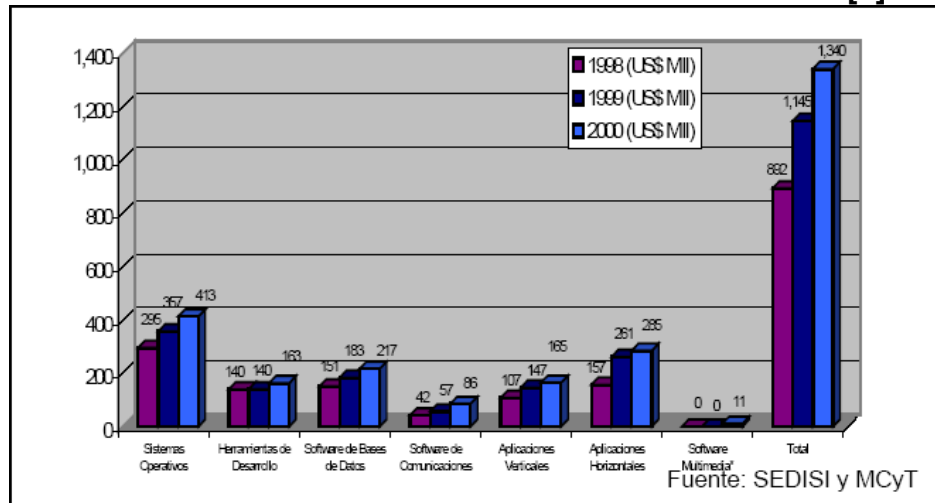
La categoría de mayor participación en el año 2000 correspondió a los sistemas operativos, sin embargo no es la que mayor dinamismo ha mostrado, por el contrario junto con las aplicaciones verticales y horizontales son las que mayor desaceleración han sufrido en el año de análisis. En contraste, las categorías de software de comunicaciones, software de base de datos y herramientas de desarrollo, han mostrado los mayores incrementos, con el 49,9%, 18,4% y 16,4% respectivamente.

El notable crecimiento del software y en general de todos los productos y servicios relacionados con las comunicaciones, puede atribuirse al

importante desarrollo que está experimentando este sector en España, en parte por el impacto de la liberalización de las telecomunicaciones.

En la siguiente gráfica pueden verse con mayor detalle las cifras del Producto Interno Bruto para cada una de las categorías del segmento de software:

Gráfico 1.11. Producto Interno Bruto – Software [7]



1.3. Plan piloto

1.3.1. Descripción

Previo al inicio del estudio descrito en esta tesis, se emprendió un plan piloto [8] que se basó en la aplicación de un instrumento de medición, elaborado por un grupo de tesistas de la Facultad de Ingeniería en Electricidad y Computación de la Escuela Superior Politécnica del Litoral dirigidos por una investigadora ecuatoriana y una belga, en 3

empresas desarrolladoras de software grandes situadas en las ciudades de Guayaquil y Quito.

1.3.2. Resultados

Varios inconvenientes en la aplicación del plan piloto no han permitido su oportuna finalización, en razón que dicho plan aún se encuentra en ejecución; sin embargo, los indicadores y lineamientos generales sugeridos en el mencionado plan piloto, nos han permitido redefinir y adicionar parámetros y como consecuencia de ellos formular los indicadores definitivos para nuestra investigación.

La situación anterior implicó concretar la debida denominación de los indicadores; exponer y ratificar sus objetivos; y, la interpretación de cada uno, aspectos que se tratan con amplitud y detalle en el Capítulo 3 “Desarrollo”.

Lo anterior exige la estricta observancia de los objetivos de la tesis y del plan piloto, y sobre todo procurando que apunten a configurar la información y resultados de manera que se demuestre los dos aspectos fundamentales planteados en las hipótesis.

1.4. Justificación

1.4.1. Importancia del proyecto

La realidad de las empresas de desarrollo de software en nuestro país deja en evidencia que la mayoría de empresas son pequeñas y no poseen un proceso de desarrollo debidamente documentado, no recopilan mediciones para generar indicadores y mucho menos toman decisiones en base a estos.

Basándonos en esta realidad y tomando en cuenta que existen otras empresas de Latinoamérica que ya poseen certificaciones en CMMI e ISO9001, hemos creído importante aplicar y validar un instrumento de medición que pueda ser aplicado a empresas desarrolladoras de software pequeñas en nuestro país que permita mejorar sus procesos de desarrollo y que posteriormente facilite la obtención de este tipo de certificaciones.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Métricas

2.1.1. Conceptos básicos de métricas

Empezaremos por definir los posibles términos que se encuentran encerrados en la palabra *métrica*, porque es muy común asociarla con las palabras medición y medida, aunque estas tres son distintas. La *medición* “es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas”. Una *medida* “proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto”. El IEEE “*Standard Glossary of Software Engineering Terms*”[9] define *métrica* como “*una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado*”.

Muchos investigadores han intentado desarrollar una sola métrica que proporcione una medida completa de la complejidad del software. Aunque se han propuesto docenas de métricas o medidas, cada una de éstas tiene un punto de vista diferente; y por otro lado, aunque bien se sabe que existe la necesidad de medir y controlar la complejidad

del software, es difícil de obtener un solo valor de estas métricas de calidad. Aún así debería ser posible desarrollar medidas de diferentes atributos internos del programa.

Aunque todos estos obstáculos son motivo de preocupación, no son motivo de desprecio hacia las métricas. En este sentido se dice que la medición es esencial, si se desea realmente conseguir la calidad de software.

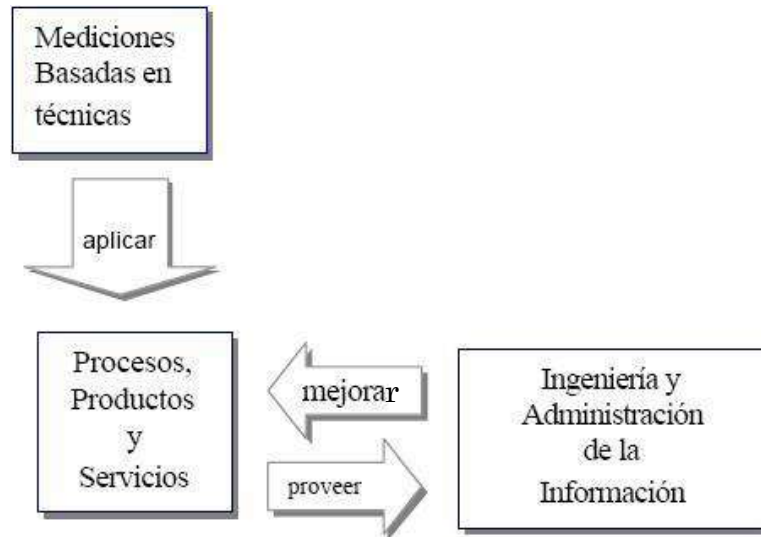
Es por eso que existen distintos tipos de métricas para poder evaluar, mejorar y clasificar al software final, en donde serán manejadas dependiendo del entorno de desarrollo del software al cual pretendan orientarse. **[10]**

2.1.2. Métricas de software

Las métricas de software son la aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo de software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos. Estas métricas también proveen la información necesaria para la toma de decisiones técnicas. En la figura 2.1 se ilustra una extensión de esta definición para incluir los

servicios relacionados al software como respuesta a los resultados del cliente. [10]

Figura 2.1. Servicios Relacionados al Software [10]



Las métricas son el inicio de una disciplina que van a ayudar a la evaluación de los modelos de análisis y de diseño, en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente, y ayudarán al diseño de pruebas más efectivas; razón por la que se propone un proceso de medición, el cual se puede caracterizar por cinco actividades:

- **Formulación:** Diseño, creación, obtención e ingenio de medidas y métricas del software apropiadas para la representación del software en cuestión.

- **Colección:** El mecanismo o método empleado para acumular datos necesarios tendentes a la obtención de las métricas ingenieras y formuladas.
- **Análisis:** Proceso de revisión y configuración de cálculos de las métricas, el uso y la aplicación de herramientas matemáticas.
- **Interpretación:** La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
- **Realimentación:** Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software. **[10]**

Es por eso que existen distintos tipos de métricas para poder evaluar, mejorar y clasificar al software final, en donde serán manejadas dependiendo del entorno de desarrollo del software al cual pretendan orientarse.

Las métricas de software incluyen otras actividades, tales como:

- Estimación de costo y el esfuerzo.
- Medición de la productividad.
- Acumulación de datos.
- Realización de modelos y mediciones de la calidad.
- Elaboración de modelos de seguridad.
- Evaluación y modelos de desempeño.

- Valoración de las capacidades y de la madurez.
- Administración por métricas.
- Evaluación del método y herramientas. **[10]**

2.1.3. Clasificación de métricas

La clasificación de una métrica de software refleja o describe la conducta del software. A continuación se muestra una breve clasificación de métricas de software:

- **Métricas de complejidad:** Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; tales como volumen, tamaño, anidaciones, costo (estimación), agregación, configuración y flujo. Estas son los puntos críticos de la concepción, viabilidad, análisis, y diseño de software.
- **Métricas de calidad:** Son todas las métricas de software que definen de una u otra forma la calidad del software; tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.
- **Métricas de competencia:** Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes con respecto a su certeza,

rapidez, eficiencia y competencia. No se ha alcanzado mucho en esta área, a pesar de la intensa investigación académica.

- **Métricas de desempeño:** Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales, etc.
- **Métricas estilizadas:** Son las métricas de experimentación y de preferencia; por ejemplo: estilo de código, indentación, las convenciones de datos, las limitaciones, etc. Pero estas no se deben confundir con las métricas de calidad o complejidad.
- **Variedad de métricas:** Tales como portabilidad, facilidad de localización, consistencia. Existen pocas investigaciones dentro del área.

Estas clasificaciones de métricas fortalecen la idea de que más de una métrica puede ser deseable para valorar la complejidad y la calidad del software. [10]

2.1.4. Diferentes enfoques de métricas

Se han propuesto cientos de métricas para el software, pero no todas proporcionan suficiente soporte práctico para su desarrollo. Algunas

demandan mediciones que son demasiado complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas, y otras violan las nociones básicas intuitivas de lo que realmente es el software de alta calidad. Es por eso que se han definido una serie de atributos que deben acompañar a las métricas efectivas de software, por lo tanto la métrica obtenida y las medidas que conducen a ello deben cumplir con las siguientes características fundamentales además de la objetividad, factibilidad y credibilidad:

- **Simple y fácil de calcular:** Debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no obligará a un esfuerzo o a una cantidad de tiempo inusuales.
- **Empírica e intuitivamente persuasiva:** La métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión (por ejemplo: una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).
- **Consistente en el empleo de unidades y tamaños:** El cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por las variables del lenguaje de programación en el programa resulta

una sospechosa mezcla de unidades que no son intuitivamente concluyentes.

- **Independiente del lenguaje de programación:** Las métricas deberían apoyarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de los caprichos de sintaxis o semántica del lenguaje de programación.
- **Un mecanismo eficaz para la retroalimentación de calidad:** La métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.

[10]

No obstante que la mayoría de las métricas de software compensan las características anteriores, algunas de las métricas usualmente empleadas no cumplen una o dos características. Un ejemplo es el punto de función (PF) en donde se consigue argumentar que el atributo es consistente y objetivo pero falla porque un equipo ajeno independiente puede no ser capaz de conseguir el mismo valor de PF que otro equipo que emplee la misma información del software. En tal caso la siguiente pregunta se manifiesta, ¿debería entonces rechazar la medida de PF? La respuesta es por supuesto que no. El PF aporta una visión interna útil y por lo tanto provee un valor claro, incluso si no

satisface un atributo perfectamente. Este es un ejemplo en donde el uso de las métricas depende de los factores individuales del desarrollador o desarrolladores de software. **[10]**

2.1.5. Las métricas en el proceso y dominio del proyecto

Las métricas de proceso de software se emplean para fines estratégicos, y las métricas del proyecto de software son tácticas, éstas últimas van a permitir proporcionar al desarrollador de proyectos de software una evaluación al proyecto que sigue en continuo desarrollo, equivalentemente podrá ver los defectos que logren provocar riesgos a largo plazo (áreas problema); y observar si el área de trabajo (equipo) y las distintas tareas se ajustarán. **[10]**

Existe una preocupación en la unificación de las métricas dentro del proceso del software, dado que los desarrolladores no tienen la cultura de la medición. Es por eso que el uso de métricas demanda un cambio cultural, tanto en el recopilado de datos (investigación histórica), como en el cálculo de métricas (LDC, PF, métricas de calidad y orientadas a objetos) y asimismo en la evaluación (resultados obtenidos), para poder comenzar un programa de bases de métricas y justamente obtener una guía, asimismo una base de datos tanto de los procesos

como de los productos y de este modo adquirir una visión más profunda de lo que se está elaborando. **[10]**

Las métricas de software nos aportan una manera de estimar la calidad de los atributos internos del producto, permitiendo así al ingeniero de software valorar la calidad antes de construir el producto, así el tiempo invertido será identificando, examinando y administrando el riesgo, este esfuerzo merece la pena por muchas razones ya que habrá disminución de disturbios durante el proyecto, asimismo se podrá desarrollar una habilidad de seguir y controlar el proyecto y se alcanzará la seguridad que da planificar los problemas antes de que ocurran, además conseguiremos absorber una cantidad significativa del esfuerzo en la planificación del proyecto. **[10]**

2.2. Estándares de calidad

2.2.1. ISO 9001

La Norma ISO 9001 especifica los requisitos para un sistema de gestión de la calidad que pueden utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales. Se centra en la eficacia de la gestión de la calidad para dar cumplimiento a los requisitos del cliente. Esta norma internacional no incluye requisitos específicos de otros sistemas de gestión, tales como

aquellos particulares para la gestión ambiental, gestión de la seguridad y salud ocupacional, gestión financiera o gestión de riesgos. Sin embargo esta Norma internacional permite a una organización integrar o alinear su propio sistema de gestión de la calidad con requisitos de sistemas de gestión relacionados. **[11]**

La nueva ISO 9001:2000, que es la única certificable de la familia, tiene una nueva estructura basada en procesos, y consta de los siguientes puntos principales:

- Responsabilidad de la Dirección
- Gestión de recursos
- Realización del Producto
- Medición, análisis y mejora
- Anexo A (Tabla A1 Correspondencia entre ISO 9001:94 e ISO 9001:2000)

Como se puede comprobar, esta nueva revisión ISO:

a) se basa en el famoso “Círculo de Deming”: PDCA - acrónimo de Plan, Do, Check, Act (Planificar, Hacer, Verificar, Actuar)-

b) está estructurada en cuatro grandes bloques, completamente lógicos, y esto significa que con el modelo de sistema de gestión de calidad basado en ISO se puede desarrollar en su seno cualquier

actividad. La ISO 9000:2000 se va a presentar con una estructura válida para diseñar e implantar cualquier sistema de gestión, no solo el de calidad, e incluso, para integrar diferentes sistemas. Otra novedad que presenta es el concepto de mejora continua. Se insiste en que el sistema de gestión de la calidad tiene que ser algo dinámico que se va enriqueciendo continuamente alimentado por la satisfacción/insatisfacción de los clientes y por sus diferentes demandas a lo largo del tiempo. Ya no habrá sitio para sistemas de gestión estáticos que, aun hoy en día, abundan. **[12]**

En definitiva esta norma internacional deriva los requisitos de un sistema de gestión de calidad para que una organización demuestre su capacidad de proporcionar de forma coherente productos que satisfagan los requisitos del cliente y las recomendaciones aplicables, al tiempo que incremente la satisfacción del cliente a través de la aplicación eficaz del sistema lo que ciertamente incluye procesos de mejora continúa el aseguramiento de la conformidad con lo requisitos del cliente y las exigencias regulatorias aplicables.

Todo los requisitos de esta norma internacional son genéricos, ya que pueden ser aplicables a cualquier organización sin importar el tamaño tipo o producto que proveen.

2.2.2. CMM

El Modelo de Capacidad y Madurez o CMM (Capability Maturity Model), es un método de definir y gestionar los procesos a realizar por una organización. Fue desarrollado inicialmente para los procesos relativos al software por la Universidad Carnegie-Mellon para el SEI (Software Engineering Institute). Actualmente este modelo ha evolucionado al CMMI. **[13]**

El Capability Maturity Model Integration (CMMI) es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por el Instituto de Ingeniería del Software de la Universidad Carnegie Mellon (SEI), y publicado en su primera versión en enero de 2002. **[14]**

Durante los años 90 el SEI desarrolló modelos para la mejora y medición de la madurez específicos para varias áreas:

- CMM-SW: CMM for software
- SA-CMM: Software Acquisition CMM.
- SSE-CMM: Security Systems Engineering CMM.
- SE-CMM: Systems Engineering CMM.
- IPD-CMM: Integrated Product Development CMM.

A finales de la década era habitual que una organización implantara de forma simultánea el modelo CMM-SW (CMM for software) y SE-CMM (Systems Engineering Capability Maturity Model). CMMI se desarrolló para facilitar y simplificar la adopción de varios modelos de forma simultánea, y su contenido integra y da relevo a la evolución de sus predecesores:

- CMM-SW (CMM for Software)
- SE-CMM (Systems Engineering Capability Maturity Model)
- IPD-CMM (Integrated Product Development)

Las áreas de proceso que ayuda a mejorar o evaluar CMMI son 22 en la versión que integra desarrollo de software e ingeniería de sistemas (CMMI-SE/SW) y 25 en la que cubre también integración de producto (CMMI-SE/SW/IPPD). Un área de proceso es un conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir un conjunto de objetivos. **[14]**

Vistas desde la representación continua del modelo, se agrupan en 4 categorías según su finalidad: Gestión de proyectos, Ingeniería, Gestión de procesos y Soporte a las otras categorías. Vistas desde la representación escalonada, se clasifican en los 5 niveles de madurez. Al nivel de madurez 2 pertenecen las áreas de proceso cuyos

objetivos debe lograr la organización para alcanzarlo, ídem con el 3, 4 y 5. **[14]**

Los 6 niveles definidos en CMMI para medir la capacidad de los procesos son:

0.- Incompleto: El proceso no se realiza, o no se consiguen sus objetivos.

1.- Ejecutado: El proceso se ejecuta y se logra su objetivo.

2.- Gestionado: Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos.

3.- Definido: Además de ser un proceso "gestionado" se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.

4.- Cuantitativamente gestionado: Además de ser un proceso definido se controla utilizando técnicas cuantitativas.

5.- Optimizado: Además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica o cambia para adaptarlo a los objetivos del negocio. **[14]**

2.3. Proyecto y aplicación de software

- **Proyecto de Software:** Un conjunto administrado de recursos interrelacionados que entrega uno o más productos a un consumidor o

usuario. Este conjunto de recursos tiene inicio y fin definidos y generalmente operan de acuerdo a un plan. Tal plan es frecuentemente documentado y especifica el producto a ser entregado o implementado, los recursos y fondos usados, el trabajo a ser realizado y el calendario para realizar dicho trabajo. Un proyecto de software puede ser clasificado en varios tipos: Nuevo, Ampliación o Mejora o Adaptación a una nueva tecnología. [15]

- **Aplicación de Software:** Software diseñado para satisfacer las necesidades de un usuario. [15]

2.4. Requerimientos del software

Los problemas que a menudo tienen que resolver los ingenieros de software son tremendamente complejos. Comprender la naturaleza de los problemas puede ser muy difícil, especialmente si el sistema es nuevo. En consecuencia, es difícil establecer exactamente lo que el sistema debe hacer. Las descripciones de los servicios y restricciones son los requerimientos para el sistema, y el proceso de describir, analizar, documentar y verificar estos servicios y restricciones se llama ingeniería de requerimientos.

El término requerimientos no se utiliza de forma consistente en la industria de software. En algunos casos, un requerimiento se visualiza

como una declaración abstracta de alto nivel de un servicio que debe proveer el sistema o como una restricción de éste. A menudo los requerimientos de sistemas de software se clasifican en funcionales y no funcionales. [16]

2.4.1. Requerimientos funcionales

Los requerimientos funcionales de un sistema describen la funcionalidad o los servicios que se espera que éste provea. Estos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios del software. Cuando se expresan como requerimientos del usuario, habitualmente se describen de forma general mientras que los requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas y salidas, excepciones, etc. [16]

2.4.2. Requerimientos no funcionales

Estos, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la

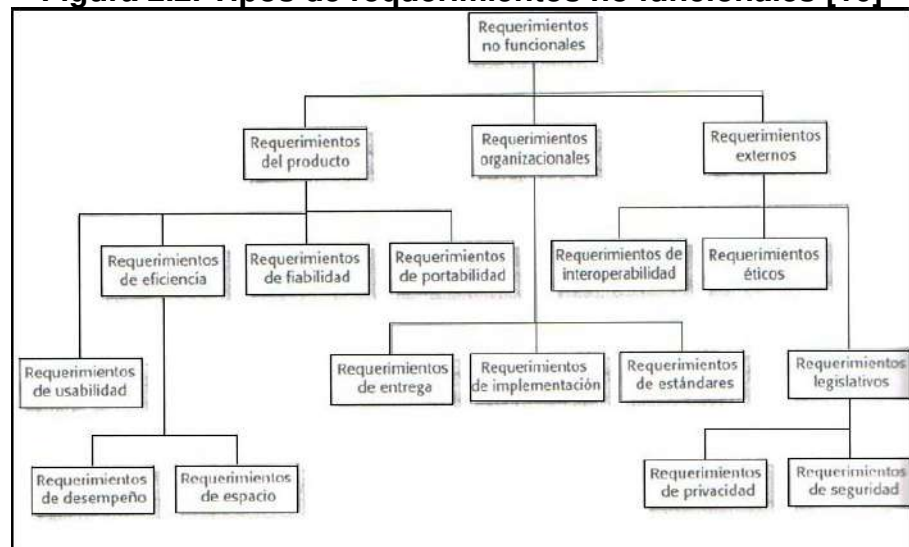
capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en las interfaces del sistema.

Muchos requerimientos no funcionales se refieren al sistema como un todo más que a rasgos particulares del mismo. Esto significa que a menudo son más críticos que los requerimientos funcionales particulares. Mientras que el cumplimiento de este último degradará al sistema, una falla en un requerimiento no funcional del sistema lo inutiliza. Por ejemplo, si un sistema de vuelo no cumple los requerimientos de fiabilidad, no se certificará como seguro para su operación; si un sistema de control de tiempo real no cumple los requerimientos de desempeño, las funciones de control no operarán correctamente.

Sin embargo, los requerimientos no funcionales no siempre se refieren al sistema de software a desarrollar. Algunos de estos requerimientos restringen el proceso a utilizar en el desarrollo del sistema. Algunos ejemplos de requerimientos de procesos son la especificación de los estándares de calidad que se deben utilizar en el proceso, una especificación que el diseño debe producir con una herramienta CASE particular y una descripción del proceso a seguir.

Los requerimientos no funcionales surgen de las necesidades del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, etc. La figura 2.2 es una clasificación de los diferentes tipos de requerimientos no funcionales que pueden surgir. [16]

Figura 2.2. Tipos de requerimientos no funcionales [16]



2.5. Pruebas del software

Una de las últimas fases del ciclo de vida antes de entregar un programa para su explotación, es la fase de pruebas. La fase de pruebas añade valor al producto que se maneja: todos los programas tienen errores y la fase de pruebas los descubre; ese es el valor que añade. El objetivo específico de la fase de pruebas es encontrar la mayor cantidad de errores posibles.

Entre las principales pruebas de software tenemos:

- **Pruebas de caja negra:** Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

El problema con las pruebas de caja negra no suele estar en el número de funciones proporcionadas por el módulo (que siempre es un número muy limitado en diseños razonables); sino en los datos que se le pasan a estas funciones. El conjunto de datos posibles suele ser muy amplio (por ejemplo, un entero, un string, un boolean, etc.).

Un programa puede pasar con holgura millones de pruebas y sin embargo tener defectos internos que surgen en el momento más inoportuno. Las pruebas de caja negra nos convencen de que un programa hace lo que queremos; pero no de que haga (además) otras cosas menos aceptables. [17]

- **Pruebas de caja blanca:** Las pruebas de caja blanca son mucho mas amplias, normalmente se denominan pruebas de cobertura o pruebas de caja transparente, al total de pruebas se caja blanca se le llama cobertura, la cobertura es un número porcentual que indica cuanto código del programa se ha probado.

Básicamente la idea de pruebas de cobertura consiste en diseñar un plan de pruebas en las que se vaya ejecutando sistemáticamente el código hasta que haya corrido todo o la gran mayoría de el, esto que parece complicado es mas aún cuando el programa contiene código de difícil alcance, como por ejemplo manejadores de errores o "código muerto". Entiéndase por código muerto a aquellas funciones y/o procedimientos que hemos incluido por encontrarse en recopilaciones pero que estas nunca son ejecutadas por el programa, estas funciones no necesariamente deberán ser removidas pero si probadas por si algún día en revisiones futuras son incluidas.

Para los módulos que no poseen condiciones basta con ejecutar una vez el programa para asegurar una cobertura total. [18]

- **Pruebas por pares (peer review):** En los medios académicos, la revisión por pares (peer review en inglés) o arbitraje es un método usado para validar trabajos escritos y solicitudes de financiamiento con el fin de medir su calidad, factibilidad, rigurosidad científica, etc. Este método deja abierto el trabajo al escrutinio y frecuentemente, la anotación o edición, por un número de autores iguales en rango al autor.

La razón principal de este mecanismo de evaluación es que a veces los mismos autores o equipos de investigación no pueden encontrar todos los errores o fallos que puede haber en un trabajo complejo. Al ser revisado por otros grupos igualmente capaces, éstos pueden encontrar aspectos, características o debilidades no observadas por los autores, lo cual redundará en un mejor trabajo escrito al corregir errores menores y, en algunos casos, la reescritura de éste al cambiar una idea o un trabajo con inconvenientes insolubles. Generalmente, los evaluadores son anónimos e independientes, lo que fomenta las críticas completas y sin matices.

Diferentes metodologías de desarrollo de software incluyen pasos en donde se realiza revisión por pares, incluyendo etapas de definición de requerimientos, diseño detallado y desarrollo de código. Algunos de los enfoques más rigurosos, llamado inspección de software. En el movimiento de software libre, se utiliza algo parecido a la revisión por pares dado que el software es revisado, criticado y mejorado por todos aquellos que lo desean. En este contexto, el papel de la revisión por pares tiene su correspondiente en la ley de Linux, que generalmente se expresa como “dados suficientes ojos, todo error es superficial”, lo que se interpreta como Con suficientes revisores, cualquier problema puede resolverse fácilmente. [19]

- **Pruebas de Estrés:** Consiste en someter al software a condiciones extremas para verificar su correcto funcionamiento. Ej: Someterlo a grandes volúmenes de datos en horas pico del día.
- **Inspección:** Es una actividad similar a medir, examinar o probar una o más características de una entidad y comparar los resultados con requerimientos especificados para establecer si la conformidad está siendo alcanzada por cada característica.
- **Simulación:** Consiste en representar escenarios reales para probar el software y detectar problemas.

2.6. GQIM

2.6.1. ¿Qué es el GQIM?

El GQIM (Goals Questions Indicators and Measures) es un Proceso de Medición dirigido a objetivos, el mismo que fue desarrollado por el SEI (Software Engineering Institute) en la universidad de Carnegie Mellon en Pittsburg y está enmarcado dentro del área de medición y análisis en el nivel 2 del CMMI (Capability Maturity Model Integration).

2.6.2. Propósito del GQIM

El propósito del GQIM es plasmar en mediciones específicas los objetivos de negocio de las empresas desarrolladoras de software, para asegurar de esta manera que cada dato recopilado ayudará al cumplimiento de los mismos.

El proceso empieza con identificar los objetivos del negocio y dividirlos en sub-objetivos más manejables, finalizando con un plan para implementar mediciones bien definidas e indicadores que ayuden a cumplir los objetivos del negocio. En todo el proceso se mantiene el seguimiento de los objetivos iniciales de esta manera al recolectar los datos no se los pierde de vista.

2.6.3. Pasos para desarrollar el GQIM

El GQIM consta de 10 pasos:

1. **Identificar los objetivos de negocio:** Es el primer paso para identificar y definir mediciones específicas del software y consiste en identificar los objetivos de negocio en los cuales están basados los esfuerzos de la organización. Este paso debe dar inicio en cualquier nivel organizacional donde los objetivos puedan ser razonablemente identificados;
2. **Identificar qué se desea saber o aprender:** Consiste en identificar que es necesario conocer para entender, medir, predecir o mejorar las actividades necesarias para lograr los objetivos. La herramienta recomendada para identificar estas preguntas es la lista de entidades y preguntas;
3. **Identificar los sub-objetivos de negocio:** Consiste en transformar los objetivos de negocio de alto nivel en Sub-Objetivos que se relacionen específicamente con actividades que se manejan o desarrollan. Se agrupan las preguntas que se refieran a problemas similares. Estos problemas son potenciales Sub-Objetivos;
4. **Identificar entidades y atributos:** Utilizando la lista de preguntas que se refieren a un mismo problema se identifican la entidad implícita en cada pregunta y los atributos

relacionados con cada entidad. Los atributos son aquellos que si son cuantificados ayudarán a responder la pregunta o establecer un contexto para interpretar las respuestas;

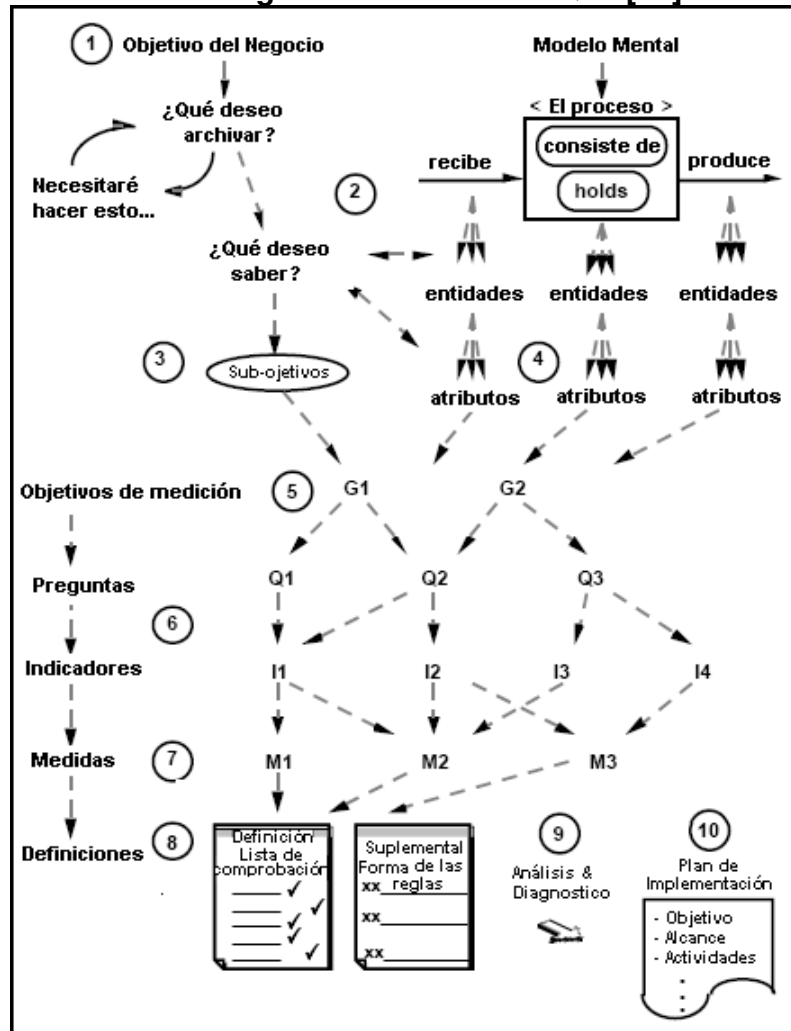
5. **Formalizar los objetivos de medición:** Utilizando los Sub-Objetivos, Entidades y Atributos se generan los Objetivos de Medición utilizando la siguiente estructura: Un objeto de interés, un propósito, una perspectiva y una descripción del entorno y restricciones;
6. **Identificar preguntas e indicadores cuantificables:** En este paso se generan preguntas cuantificables relacionadas a los objetivos de medición que se desean ser contestadas. Se definen los indicadores que contestarán dichas preguntas. Cuando se identifican preguntas y definen indicadores es importante tener en cuenta el objetivo al cual se dirigen. Ej: En entornos cambiantes es menos útil precisar números y hacer análisis estadísticos;
7. **Identificar los elementos de datos:** Se listan los datos que son necesarios obtener para formar los indicadores. Algunos datos obtenidos pueden ayudar a formar más de un indicador;
8. **Definir las mediciones:** Los nombres de las mediciones no son suficientes. Se tiene que poder decirle a otras personas

exactamente como cada medición debe ser obtenida, y de esta manera poder ser interpretadas correctamente;

9. **Identificar las acciones necesarias para implementar las mediciones:** En este paso se añade información acerca del estado actual y el uso de las mediciones, de esta manera se puede preparar un plan efectivo para implementar las mismas; y,
10. **Preparar un plan:** Se prepara un plan para implementar las acciones que han sido identificadas. **[20]**

En la figura 2.3 se muestra un gráfico que resume el proceso GQIM:

Figura 2.3. Proceso GQIM [20]



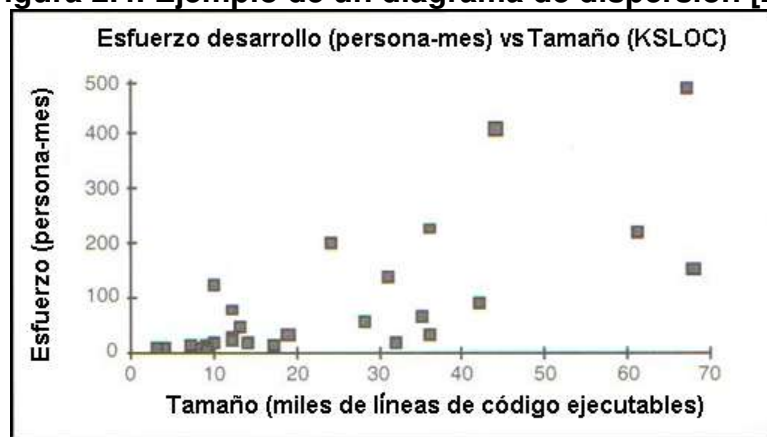
2.7. Herramientas de interpretación de datos

Los datos que son recolectados para visualizar el proceso de desarrollo o investigar causas asignables y potenciales mejoras, a menudo necesitan ser ordenados para poder entender la información obtenida. Esto significa organizar y resumir los datos y buscar patrones, tendencias y relaciones. Herramientas tales como diagramas de dispersión, diagramas causa efecto, histogramas, gráficas de barra y cartas de control son de gran ayuda. [21]

2.7.1. Diagramas de dispersión

Un diagrama de dispersión, como el que se muestra en la figura 2.4, es un gráfico de valores observados que muestra como una variable se ha comportado con respecto a otra. Estos diagramas son usados como un primer paso en la exploración de los datos, especialmente como parte de una búsqueda de relaciones causa-efecto. [21]

Figura 2.4. Ejemplo de un diagrama de dispersión [21]



Los diagramas de dispersión a menudo son usados para responder preguntas tales como: ¿El producto de la compañía A trabaja mejor que el de la compañía B? Cuando estos diagramas sugieren que una relación podría existir entre dos variables, su uso es a menudo seguido por métodos estadísticos más formales tales como el análisis exploratorio de datos o análisis de regresión. [21]

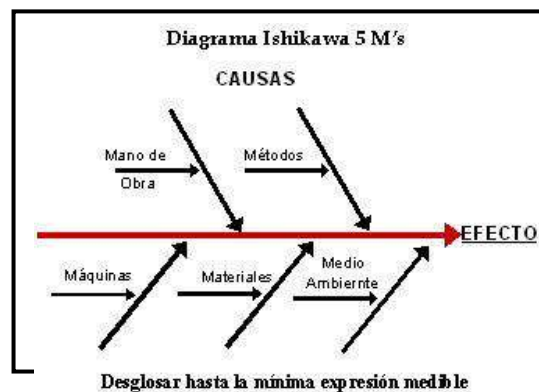
2.7.2. Diagramas causa-efecto

Un diagrama causa-efecto es una muestra gráfica que es usada para probar y mostrar relaciones entre un problema (efecto) y sus posibles causas. Son llamados gráficos *fishbone* porque tienen similitud al esqueleto de un pez.

Cuando los diagramas causa-efecto son utilizados para explorar el entorno de un proceso, es más conveniente que los diagramas sean ensamblados por gente que trabaja actualmente en el proceso. [21]

También se los conoce como diagramas Ishikawa.

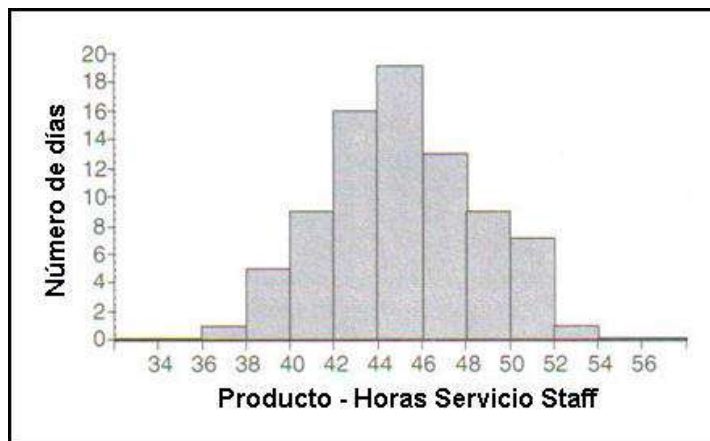
Figura 2.5. Ejemplo de un diagrama causa-efecto



2.7.3. Histogramas

Un histograma toma las mediciones de datos y muestra la distribución de los valores observados. Son creados para agrupar los resultados de las mediciones en “celdas” y contar el número en cada celda. Las alturas de las barras en los histogramas son proporcionales al número de ocurrencias dentro de cada celda, como se muestra en la figura 2.6. [21]

Figura 2.6. Ejemplo de un histograma [21]

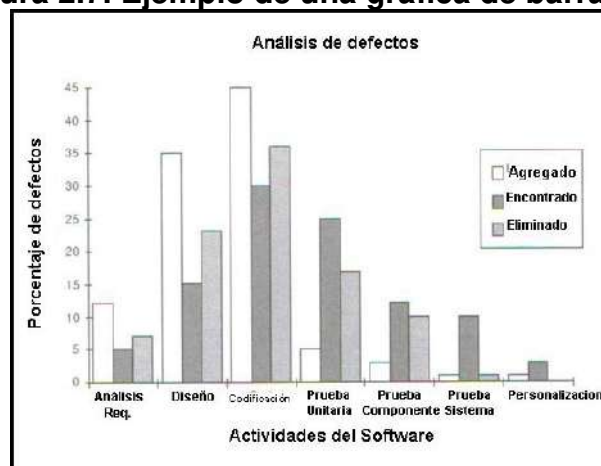


Los histogramas son útiles para investigar y resumir el desempeño de un proceso con respecto a límites especificados que el proceso o sus productos deben satisfacer.

2.7.4. Gráficas de barra

Una gráfica de barra, al igual que un histograma, es usada para investigar la forma de un conjunto de datos. La figura 2.7 es un ejemplo. Estas gráficas son similares a los histogramas en muchos aspectos pero son definidas sobre un conjunto de valores discretos. Así, ellas pueden mostrar cualquier valor numérico, no solamente conteos o frecuencias relativas. De esta manera, las gráficas de barra pueden ser usadas para mostrar datos tales como el total del tamaño, costo o tiempo trascendido asociado con entidades individuales o con un conjunto de productos o pasos de proceso. [21]

Figura 2.7. Ejemplo de una gráfica de barra [21]

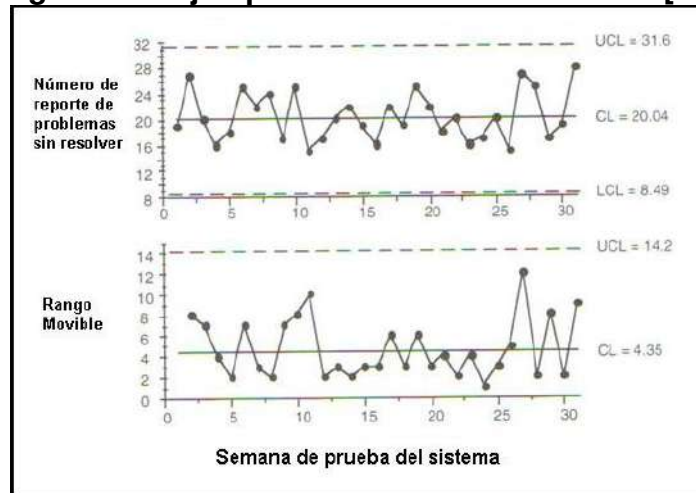


2.7.5. Cartas de control

Con una carta de control como la que se muestra en la figura 2.8 se tiene una imagen que indica el comportamiento de un proceso con

respecto a un producto en particular o un proceso en un periodo de tiempo.

Figura 2.8. Ejemplo de una carta de control [21]



La carta de control clásica posee una línea central y límites de control en ambos lados de la línea central. Tanto la línea central como los límites de control representan estimaciones que son calculadas de un conjunto de observaciones recolectadas mientras el proceso se encuentra en ejecución. Mediante una carta de control se pueden detectar inestabilidades y situaciones fuera de control en un proceso.

[21]

2.8. Análisis Univariado

El análisis univariado en un sentido amplio, se refiere a todos los métodos estadísticos que analizan la distribución de una sola variable. Con el

objeto de realizar este tipo de análisis se procederá a utilizar gráficas de barra para mostrar cual es el comportamiento promedio de ciertas variables de importancia. Para esto se obtendrá la media o promedio estándar de dichas variables. Otro método de análisis univariado es observar cuál es la composición porcentual de los elementos de dicha variable, para esto se utilizarán gráficas de barra que permitirán observar dicha composición.

2.9. Análisis Multivariado

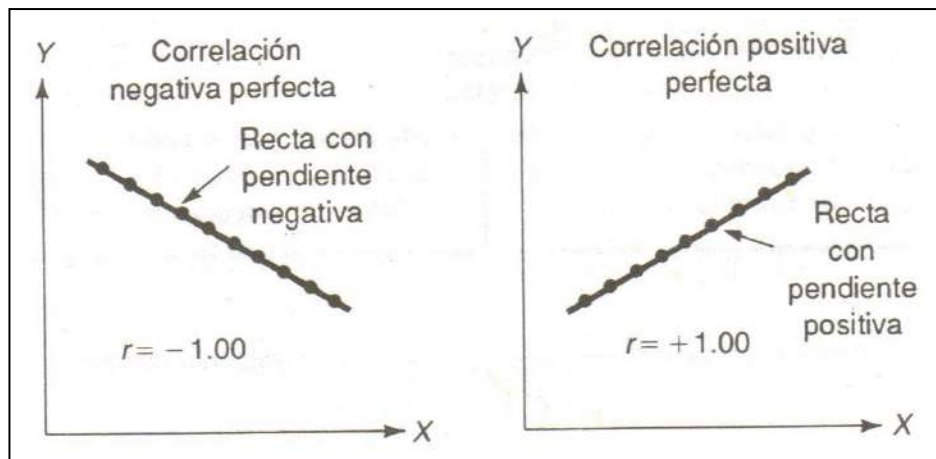
Para este análisis se aplicará el análisis de correlación, el mismo que es un grupo de técnicas estadísticas empleado para medir la intensidad de la relación (correlación) entre dos variables. [22]

El principal objetivo del análisis de correlación consiste en determinar qué tan intensa es la relación entre dos variables. Una medida de esta relación es el coeficiente de correlación, el mismo que fue originado por el investigador Kart Pearson y describe la intensidad de la relación entre dos conjuntos de variables de nivel de intervalo o de nivel de razón. Ya que se le denota con r , con frecuencia se menciona también como r de Pearson o coeficiente de correlación producto-momento de Pearson. Puede tomar cualquier valor de -1 a $+1$, inclusive. Un coeficiente de correlación de -1 o de $+1$ indica correlación perfecta. Un valor calculado de -1 revela que la

variable independiente X y la variable dependiente Y están perfectamente relacionadas en forma lineal negativa. [22]

La forma como quedarían los diagramas de dispersión si la relación entre los dos conjuntos de datos fueran lineales y perfectos se muestra en la figura 2.9.

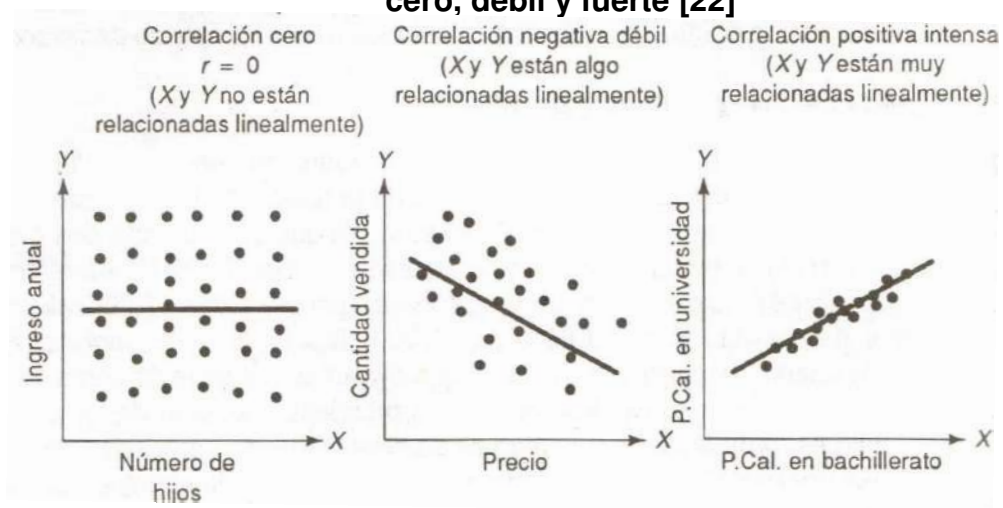
Figura 2.9. Diagramas de dispersión con correlaciones negativa y positiva perfecta [22]



Si no existe en absoluto relación entre los dos conjuntos de variables, la r de Pearson será cero. Un coeficiente de correlación r cercano a 0 (por ejemplo, 0,08) indica que la relación es poco intensa o débil. Se llega a la misma conclusión si $r = -0,08$. Coeficientes de $-0,91$ y $+0,91$ tienen igual fuerza; ambos indican una correlación muy intensa entre los dos conjuntos de variables. De esta forma, la fuerza de la correlación no depende de la dirección (ya sea $-$ o $+$). [22]

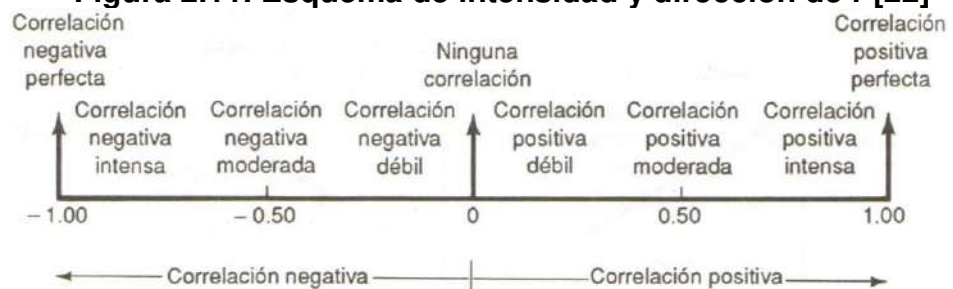
En la figura 2.10 se muestran diagramas de dispersión para $r = 0$, una r débil (por ejemplo, $-0,23$) y una r fuerte (por ejemplo, $+0,87$). Obsérvese que si la correlación es débil, existe una dispersión considerable con respecto a una línea recta trazada a través del espacio central de los datos. [22]

Figura 2.10. Diagramas de dispersión que muestran correlación cero, débil y fuerte [22]



La figura 2.11 representa adecuadamente la intensidad y la dirección del coeficiente de correlación.

Figura 2.11. Esquema de intensidad y dirección de r [22]



Para calcular r se utiliza la siguiente fórmula:

$$r = \frac{S_{xy}}{S_x S_y}$$

De donde:

r = Coeficiente de correlación

S_{xy} = Covarianza de x y y

S_x = Desviación estándar de x

S_y = Desviación estándar de y

Al ser una muestra pequeña ($n < 50$) es necesario probar que el coeficiente de correlación encontrado es representativo de la industria, para esto se procede a realizar la prueba de significación utilizando la distribución t de Student para muestras pequeñas.

$H_0 : \rho = 0$ (La correlación en la población es cero).

$H_1 : \rho \neq 0$ (La correlación en la población es distinta de cero).

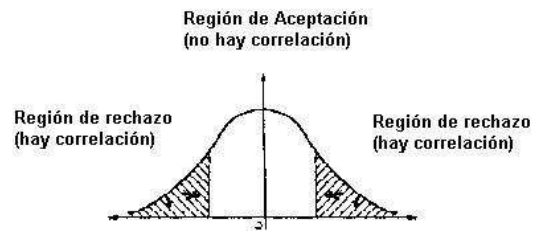
$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \text{ con } n - 2 \text{ grados de libertad}$$

Utilizando el nivel de significación 0.01, la regla de decisión indica que si la t calculada se encuentra en el área no sombreada, se aceptará la hipótesis nula. Para localizar el valor crítico del área sombreada se

consultó la tabla de distribución t Student para $n - 2$ grados de libertad.

[22]

Figura 2.12. Distribución t Student para prueba de coeficiente correlación [22]



CAPÍTULO III

3. DESARROLLO

En el presente capítulo se procede a describir las hipótesis que se verificarán en el Capítulo 5 “Análisis y presentación de resultados” Sección 5.2 “Pruebas para verificar hipótesis”; se enumeran los objetivos de la tesis; se describe la metodología de recolección de información; se detallan las características, elementos y demás aspectos relacionados con la población objetivo; se explica el tamaño e incidencia de la muestra para aplicación del instrumento; así como también la metodología seguida para ejecutar el plan de capacitación; finalmente se muestran las plantillas de medición y los indicadores generados que configuran el instrumento de medición.

3.1. Hipótesis

Analistas e investigadores sobre temas relacionados con la Ingeniería de Software indican que, prestar atención a la calidad de las especificaciones en las primeras etapas del proceso de desarrollo del software, asegura que pocos serán los defectos detectados y corregidos en fases posteriores. Así también concluyen que los defectos afectan los costos de desarrollo, las fechas de entrega y la satisfacción del cliente; en consecuencia es importante tener presente que la existencia de defectos derivan en la baja calidad del producto. Un estudio llevado a cabo en el

uso de PSP (Personal Software Process) en una compañía ecuatoriana demuestra que aproximadamente el 50% de los defectos son causados en la etapa de diseño. Los defectos pueden también ser detectados durante inspecciones, y se ha demostrado que, en promedio, hasta el 57% son detectados en el código y en los documentos de diseño. Una regla es que entre el 50% y el 75% de todos los defectos de diseño pueden ser encontrados cuando se hacen las inspecciones. [5]

Las hipótesis son suposiciones sujetas a prueba que toman como base un razonamiento derivado de una recopilación de ideas relacionadas con nuestra realidad, y de los procesos investigativos realizados; en tal sentido, atendiendo las orientaciones fundamentales y las descripciones hipotéticas contenidas en el documento de aplicación para la aprobación del financiamiento del presente estudio las hipótesis a probar son:

- Cuanto mejor es la estimación del tamaño de un proyecto de software, mejor es la estimación del tiempo de realización del mismo. No obstante, debido a la complejidad que en la práctica derivaba la determinación del tamaño del proyecto, la presente hipótesis se sustituye por la que se describe a continuación:

“Cuanto mejor es el tiempo empleado en la planificación, menor es el error en la estimación del tiempo”; planteamiento hipotético

que en todo caso confirma la afirmación de que a “mejor planificación menos errores”.

- **“Cuanto mayor es el número de inspecciones en las etapas de diseño y de implementación, más pequeño es el número de los errores en el producto final entregado a los clientes”.**

3.2. Objetivos

Los objetivos del presente estudio son los siguientes:

1. Validar un instrumento de medición para obtener métricas de empresas desarrolladoras de software pequeñas.
2. Aplicar y analizar este conjunto de métricas en pequeñas empresas de software.
3. Determinar los factores que inciden en la administración de proyectos de software en las pequeñas empresas desarrolladoras de software ecuatorianas.
4. Determinar la relación entre las siguientes variables: esfuerzo en el desarrollo versus complejidad técnica y versus complejidad del negocio; esfuerzo del cliente y esfuerzo del usuario versus satisfacción del cliente y satisfacción del usuario; esfuerzo en el desarrollo versus número de personas involucradas; defectos y fallas versus complejidad técnica y versus complejidad del negocio; y, esfuerzo en documentación versus fallas.

3.3. Metodología de recopilación de la información

El estudio inicialmente contemplaba la construcción de un Sitio Web a través del cual las empresas debían ingresar la información de sus proyectos conforme vayan avanzando en el desarrollo de los mismos, sin embargo, basándonos en proyectos de investigaciones anteriores determinamos que en la práctica la obtención de datos tenía alta efectividad si se buscaban directamente de las empresas, por lo que se planteó la duda con respecto a la aceptación del Sitio Web como herramienta para la recopilación de la información, debido a que las entidades participantes preferirían enviar los datos vía mail, ya que en nuestro medio lo encontraban un mecanismo de mayor aplicación práctica. Tomando en cuenta esto se procedió a la elaboración de una pequeña encuesta, la cual buscaba obtener la herramienta preferida por las empresas para la recopilación de los datos.

En el sentido anterior la encuesta concluyó que la mayoría de las empresas optaron por una herramienta que funcione localmente, dejando de esta manera descartado el mecanismo inicialmente planificado (un Sitio Web). Entre las principales razones por las cuales las entidades escogieron el mencionado mecanismo, se encuentran las siguientes:

- Que existen problemas con la conexión a Internet, como el hecho de eventuales desconexiones imprevistas, lo que dejaría inconclusa la

actividad de llenado lo cual incurre en más tiempo para volver a realizar esa actividad.

- Que la información a proporcionar en muchos casos es de carácter confidencial, como el dato costo hora hombre, el costo del proyecto, etc., ante lo cual un Sitio Web es percibido de alto riesgo, en razón que puede ser objeto de interferencia y ataque de cualquier parte y tipo.
- Que el tener una aplicación que recopile datos de su proceso de desarrollo de manera local les asegura que los datos proporcionados se van a quedar con ellos y van a formar parte de su registro histórico.

Una vez conocida la preferencia de las empresas por una herramienta de recopilación de información que funcione de manera local se procedió a preguntar cuál sería el software de mayor agrado para la construcción de dicha herramienta. Casi todas prefirieron Microsoft Access debido a que poseen licencias Microsoft para su paquete utilitario Office.

3.4. Población objetivo

Con relación a la elección de la población objetivo se consideró como elemento fundamental la premisa técnica de que para realizar una correcta comparación de los datos era necesario que las empresas tuvieran similares características; en tal sentido, se han considerado los siguientes principios:

Seleccionar únicamente empresas que dentro de sus líneas de negocios tengan el desarrollo de software y/o la provisión de servicios vinculados al desarrollo del mismo y su operación, por dicha razón se excluyeron las empresas que tienen como actividad exclusiva la comercialización.

Que la magnitud de las empresas corresponda a las de tamaño pequeño basándonos en el número de empleados. Como guía recurrimos a estudios anteriores del Proyecto VLIR-ESPOL Componente 8 Área de Ingeniería de Software realizados en los años 2003 [2] y 2005 [3], donde se aplicaban los siguientes rangos:

- **Pequeñas:** 10 empleados o menos.
- **Medianas:** entre 10 y 50 empleados.
- **Grandes:** 50 empleados o más.

Utilizando estas definiciones se procedió a seleccionar las empresas pequeñas.

Que la magnitud de las empresas también corresponda a las de tamaño pequeño con base al criterio de la duración de sus proyectos; al efecto, se utilizó como término de referencia el mismo estudio realizado por el proyecto VLIR-ESPOL en el año 2003 [2] y 2005 [3], en el que se cataloga proyectos pequeños aquellos cuya duración esté entre 1 y 6 meses; proyectos medianos los que estén entre 7 y 15 meses; y, grandes los que requieren más de 15 meses. [2]

Finalmente se eligieron empresas cuyo proceso siga la metodología denominada “cascada”, es decir, el inicio de cada fase debe esperar a la finalización de la inmediatamente anterior.

3.5. Muestra para aplicación de Instrumento

Referente a la muestra para la aplicación del instrumento, inicialmente se contaba con una base de datos que fue facilitada por el VLIR – ESPOL, componente 8, la misma que involucraba un total de 200 empresas, ubicadas en las ciudades de Quito, Guayaquil y Cuenca, conteniendo información actualizada de todas las empresas que desarrollan software en el Ecuador.

De la base de datos en mención, se tomó una muestra de 20 empresas desarrolladoras de software, en el mes de mayo del 2006; decisión que responde al principio estadístico de que cuando se realizan estas pruebas, es suficiente tomar el 10% del universo de estudio [23].

A efecto de cumplir con el tamaño de muestra calculado en 20 empresas, fue necesario visitar y/o llamar a un total de 73 empresas con el perfil de la población objetivo del estudio. Al término de este ejercicio los resultados de la indagación fueron que:

Número de empresas	Resultado de la indagación
1	No desarrolla.
3	Manifestaron interés, pero al final nunca participaron
4	Quedaron pendientes de confirmar (nunca confirmaron)
7	No tienen proyecto para aplicar
7	Contactadas por el plan piloto
13	No aceptaron participar (rechazo explícito)
18	Nunca se pudo localizar a la persona encargada
20	Aceptaron participar

A las 20 empresas que aceptaron la participación, se les proporcionó el programa, instrucciones y más aspectos, sin embargo, 5 de tales empresas no cumplieron el compromiso adquirido, por lo que se optó por solicitar a las 15 restantes, aporten con proyectos adicionales de manera que podamos completar al menos 20 proyectos.

3.6. Plan de capacitación

Como parte del proceso de capacitación, para efectuar las visitas a las empresas fue necesario desarrollar una herramienta en donde se expliquen las bases teóricas de las plantillas que recolectarían los datos.

El citado mecanismo de inducción tuvo un total de 40 diapositivas; las primeras diapositivas explican el objetivo que tiene el Proceso de Medición Dirigido a Objetivos (GQIM). En las diapositivas siguientes se explican los 10 pasos ha seguir para lograr el GQIM, mismos que son explicados en detalle y de manera sencilla para mayor entendimiento. Cabe precisar que al final se colocaron las respectivas referencias.

ANEXO 1

3.7. Instrumento de medición

3.7.1. Definición

Se define como instrumento de medición a la herramienta de la que nos servimos para lograr el objetivo fundamental de la presente tesis que es obtener métricas de empresas desarrolladoras de software, cuyo mecanismo de funcionamiento se apoya en el levantamiento de información, cálculo, y evaluación de los resultados obtenidos a fin de inferir acciones y recomendaciones tendentes a optimizar los resultados en esta parte de la industria informática.

3.7.2. Antecedentes

El Instrumento de Medición con el cual se inició el presente estudio, fue propuesto por una tesis anterior [8], cuyo objetivo es crear un plan de métricas que sea aplicable a empresas desarrolladoras de software

grandes, el cual está en proceso de aplicación en 3 empresas de tales características.

Al mismo tiempo que se llevaba a cabo el plan piloto, se dio inicio al presente estudio para lo cual se tuvo que realizar un corte a la información generada por el plan piloto, para el efecto nos fue proporcionado el “Plan de Métricas” con las respectivas plantillas para el levantamiento de información. **ANEXO 2**

3.7.3. Plantilla de levantamiento de información

Las plantillas de levantamiento de información derivadas del Plan de Métricas se refieren a los siguientes aspectos:

- Datos generales del proyecto;
- Definición de tareas / entregables;
- Reportes de problemas o Cambios en los requerimientos;
- Líneas de código producidas; y,
- Complejidad del software.

Dichas plantillas contienen los datos requeridos para formar los indicadores de medición, las mismas que sin embargo previo a ser entregadas a las empresas objetivo fue necesario incorporarles los siguientes datos adicionales:

En la plantilla Datos generales del proyecto:

- **Costo hora hombre:** tiene como finalidad ayudar a mejorar el cálculo del costo del proyecto, basados en las horas que dedica una persona al desarrollo del mismo.

En la plantilla Definición de tareas/entregables:

- **Problemas – fecha abierta:** tiene como finalidad incorporar el dato cuando a la empresa se le presente un problema (consideramos problema a cambios de requerimiento, reporte de defectos o fallas), se debe registrar en el momento que empezó a tratar dicho problema para conocer cuan eficiente sería en tratar los problemas.
- **Tipo de Tarea:** la finalidad es separar las tareas en: inspecciones, tareas normales, cambios de requerimientos, reporte de defectos y reporte de fallas.

En la plantilla Complejidad del software:

- **Horas de desarrollo del componente:** dato necesario para conocer la cantidad total de esfuerzo (horas) que se utilizó para desarrollar un componente específico.
- **Número de defectos:** con el propósito de registrar la cantidad total de defectos en los módulos del software que hubieren sido detectados durante la fase de pruebas.

3.7.4. Indicadores de medición

Los indicadores de medición constituyen la expresión numérica y gráfica de los resultados obtenidos en el proceso de desarrollo y personalización de los sistemas, a través de los datos e información provistos por los participantes en el estudio y que relacionan de manera general el esquema de planificación y estimación con los resultados reales luego del proceso, cuya finalidad es proporcionar a las distintas empresas una visión general de la situación resultante en un proceso de desarrollo de software, anotando que los indicadores y más variables de medición utilizados se sustentan básicamente en:

Indicador 1	Satisfacción del Cliente y Satisfacción del Usuario V.S. Esfuerzo del cliente y esfuerzo del usuario
Indicador 2	Porcentaje de error en la estimación del esfuerzo
Indicador 3	Porcentaje del esfuerzo por persona en las diferentes fases
Indicador 4	Porcentaje de esfuerzo empleado en documentación
Indicador 5	Complejidad del Negocio y Complejidad Técnica
Indicador 6	Porcentaje de error en la estimación del costo

Indicador 7	Costo promedio hora real
Indicador 8	Porcentaje de inspecciones realizadas por fases y Porcentaje de inspecciones realizadas en el proyecto
Indicador 9	Porcentaje de variación de los requerimientos
Indicador 10	Complejidad del Código Fuente
Indicador 11	Indicador ponderado de la complejidad y demora del aprendizaje
Indicador 12	Tamaño de los componentes de software
Indicador 13	Ocurrencia de defectos y fallas por fases
Indicador 14	Tipo de defectos y fallas más ocurrentes
Indicador 15	Porcentaje del tiempo utilizado en corrección de defectos y fallas
Indicador 16	Eficiencia en atención de defectos y fallas
Indicador 17	Promedio de defectos por hora de programación

El uso de indicadores es de vital importancia en el análisis, evaluación y calificación de proyectos de software, ya que orienta a la formulación de correctivos y la adopción de mecanismos de ajuste y calibración capaces de mejorar o lograr la mayor calidad posible en este sentido. Además dichos indicadores podrían constituir también aporte

fundamental en las estimaciones de proyectos de software, toda vez que al tiempo de planificarlos permiten formular términos de referencia sobre la base de los cuales se acoplará el desarrollo del proyecto.

Los principales indicadores de medición que se han considerado y que cumplen los atributos de alta objetividad, credibilidad y factibilidad, se describen en los apartados siguientes, mismos que para fines didácticos y mayor comprensión se exponen con los siguientes elementos:

Denominación

Corresponde al nombre asignado al indicador.

Objetivo

Propósito que se pretende demostrar a través de los resultados obtenidos con el indicador.

Dinámica

Mecánica operativa para determinar el indicador, esto es, la forma de cálculo y la precisión de los elementos que intervienen en el mismo.

Interpretación

Exposición detallada de la forma como leer los resultados obtenidos, con énfasis en las aclaraciones que fueren necesarias en el caso de atipicidades o sesgos.

INDICADOR 1

➤ **Denominación:**

Satisfacción del Cliente y Satisfacción del Usuario Vs. Esfuerzo del cliente y esfuerzo del usuario

➤ **Objetivo:**

Determinar si el nivel de satisfacción del cliente está relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto. [8]

➤ **Dinámica:**

Para obtener el presente indicador determinamos cual ha sido el esfuerzo dedicado al proyecto tanto por el Cliente como por el Usuario. Denominando esfuerzo a la cantidad de horas que estos han participado en el mismo. También tenemos que usar el dato esfuerzo total, este dato nos indica cual ha sido el esfuerzo de la empresa en desarrollar el proyecto. Al efecto proceden las siguientes operaciones:

$$\text{Esfuerzo del cliente} = \frac{\text{Horas participación cliente}}{\text{Esfuerzo total proyecto}} * 100$$

$$\text{Esfuerzo del usuario} = \frac{\text{Horas participación usuario}}{\text{Esfuerzo total proyecto}} * 100$$

Los indicadores se expresan en términos porcentuales, toda vez que en valores absolutos no posibilitan la comparación.

Otro dato que necesitamos para formar este indicador es el nivel de satisfacción que el Cliente y Usuario han tenido en el proyecto, el

mismo que se lo solicitó mediante el llenado de plantillas previamente validadas en un estudio previo, plan piloto. [8]

Una vez calculados estos datos, esfuerzo y nivel de satisfacción, se los relacionan entre si.

PARA SATISFACCIÓN CONSOLIDADA DEL USUARIO Y CLIENTE:

El nivel de satisfacción consolidada del usuario y el nivel de satisfacción consolidada del cliente: trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera que el numerador del indicador representa la suma de los niveles de satisfacción de todos los proyectos; y, el denominador refiere al total de proyectos. Al efecto procede la siguiente fórmula:

$$\text{Satisfacción consolidada del cliente} = \frac{\sum \text{satisfacciones de clientes obtenidas}}{\text{número de proyectos}}$$

$$\text{Satisfacción consolidada del usuario} = \frac{\sum \text{satisfacciones de usuarios obtenidas}}{\text{número de proyectos}}$$

El indicador aquí obtenido constituye la media o promedio que a su vez sirve como término de referencia para comparar el nivel de satisfacción de cada proyecto.

➤ **Interpretación:**

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

En la representación gráfica, el presente indicador deriva 4 ilustraciones, 2 para el cliente y 2 para el usuario.

En las dos primeras, esto es, “satisfacción cliente y usuario” la media, o promedio se va a comparar de manera inequívoca con los demás integrantes del estudio. Todas las empresas que estén por arriba de ese indicador, denotarían un buen nivel de satisfacción. No así las que estén por debajo a ese promedio, mismas que no habrían alcanzado el nivel promedio de satisfacción.

En las 2 gráficas siguientes, esto es, “participación versus el nivel de satisfacción del cliente y usuario, respectivamente”, un sesgo alto en el dato de satisfacción y una baja participación, denotaría que el cliente o usuario no necesitó mucho tiempo de participación para estar satisfecho con lo que se estaba realizando, una causa puede ser que sea un cliente o usuario con experiencia en este tipo de proyectos o que conozca bien lo que se iba a realizar. Ahora bien, si tenemos un sesgo con alta participación y a la vez alto nivel de satisfacción, implicaría que el cliente o usuario necesitó más tiempo para poder estar satisfecho y poder comprender lo que se estaba realizando.

INDICADOR 2

➤ **Denominación:**

Porcentaje de error en la estimación del esfuerzo

➤ **Objetivo:**

Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros. [8]

➤ **Dinámica:**

Este indicador determina el grado de acercamiento de la empresa a sus estimaciones presupuestales del proyecto. La obtención se la hace, comparando el esfuerzo que se empleó en el proyecto versus el esfuerzo que se presupuestó para el mismo, con la aclaración válida que el esfuerzo significa las horas estimadas y dedicadas según corresponda, esto es, ver en que proporción estaría lo real respecto a lo estimado. Al efecto procede la siguiente fórmula:

$$\% \text{ error en estimación de esfuerzo} = \frac{\text{esfuerzo estimado} - \text{esfuerzo real}}{\text{esfuerzo estimado}} * 100$$

PORCENTAJE CONSOLIDADO DE ERROR EN LA ESTIMACIÓN DEL ESFUERZO

El porcentaje consolidado de error en estimación del esfuerzo se trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera que en el numerador del indicador se representa la diferencia entre los esfuerzos estimados y reales consolidados; y, el

denominador refiere el esfuerzo estimado consolidado; cuyo cociente es multiplicado por cien para obtener el indicador en término porcentual, es decir, la proporción consolidada del error en estimación del esfuerzo. Al efecto procede la siguiente fórmula:

$$\% \text{ Consolidado de error en estimación del esfuerzo} = \frac{\text{esfuerzo estimado consolidado} - \text{esfuerzo real consolidado}}{\text{esfuerzo estimado consolidado}} * 100$$

El indicador aquí obtenido deriva el nivel de posicionamiento de cada proyecto en particular, gráfica se presenta en el Capítulo 5 “Análisis y presentación de resultados”, cuya interpretación se describe a continuación.

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

Un sesgo muy bajo (negativo) indicaría que posiblemente la persona encargada de estimar no lo habría realizado correctamente, lo que podría derivarse de las siguientes causales: subestimó el proyecto, falta de previsión de contratiempos, contingencias, falta de planificación para la realización de las actividades, falta de experiencia o el no conocimiento de toda la amplitud del proyecto.

De su lado, un sesgo muy alto indicaría una sobreestimación del proyecto, cuyos causales podrían obedecer al poco conocimiento de

parte de la empresa hacia lo que se estaba realizando o al hecho de haber tomado como término de referencia experiencias de proyectos aparentemente similares pero que en su complejidad resultaron disímiles.

Los proyectos que tengan como resultado un valor 0 o cercano, indicarían que habrían realizado una apropiada planificación, un prudente y puntual conocimiento de la cobertura, magnitud, naturaleza, y por menores de todas y cada una de las tareas involucradas en el proyecto.

INDICADOR 3

➤ ***Denominación:***

Porcentaje del esfuerzo por persona en las diferentes fases

➤ ***Objetivo:***

Establecer la distribución del esfuerzo adecuado que se requerirá del personal asignado a cada fase del proyecto, para mantener un nivel sostenido de rendimiento entre sus miembros.

➤ ***Dinámica:***

El numerador del indicador representa el esfuerzo por persona en cada fase; y, el denominador se refiere al esfuerzo total en cada fase; cuyo cociente es multiplicado por cien para obtener el indicador en

término porcentual, es decir, la proporción que cada persona habría aportado en cada fase. Al efecto procede la siguiente fórmula:

$$\% \text{ de participación de persona en la fase} = \frac{\text{Esfuerzo de persona en la fase}}{\text{Esfuerzo total de fase}} * 100$$

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

La representación gráfica del indicador se la realiza por proyectos. Se colocan las personas participantes, las fases y el respectivo porcentaje de participación de las personas en las fases. Precisa indicar que en cada fase no necesariamente sumará el 100%, habida cuenta que no se considera el esfuerzo dedicado por el usuario y/o el cliente, sino únicamente el que corresponde al personal de la empresa.

Esta gráfica nos indicaría con cuanto esfuerzo (horas) aportó cada persona para cada una de las fases. Indicador que serviría para que en futuros proyectos de similares características la distribución de las personas se la realice examinando el esfuerzo obtenido con el indicador del proyecto anterior, lo que evitaría inequidades laborales que podrían constituirse en perturbadoras para el rendimiento del personal.

INDICADOR 4

➤ **Denominación:**

Porcentaje de esfuerzo empleado en documentación.

➤ **Objetivo:**

Determinar el esfuerzo empleado en realizar la documentación del proyecto durante su desarrollo.

➤ **Dinámica:**

PARA EL ESFUERZO EN DOCUMENTACIÓN POR PROYECTO

El numerador representa el esfuerzo en documentar el proyecto; y, el denominador se refiere al esfuerzo total empleado en el proyecto; cuyo cociente es multiplicado por cien para obtener el indicador en término porcentual, es decir, la proporción del esfuerzo total dedicado a documentar cada proyecto. Al efecto procede la siguiente fórmula:

$$\% \text{ del esfuerzo en documentación} = \frac{\text{Esfuerzo en documentar}}{\text{Esfuerzo total del proyecto}} * 100$$

PARA EL ESFUERZO CONSOLIDADO EN DOCUMENTACIÓN

El esfuerzo consolidado en documentación se trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera que en el numerador del indicador se representa el esfuerzo conjunto que han empleado todos los proyectos en la tarea de documentación; y, el denominador refiere el esfuerzo total empleado en todos los proyectos; cuyo cociente es multiplicado por cien para obtener el

indicador en término porcentual, es decir, la proporción del esfuerzo total que han dedicado todos los proyectos para la documentación. Al efecto procede la siguiente fórmula:

$$\% \text{ del esfuerzo consolidado en documentación} = \frac{\text{Esfuerzo en documentar consolidado}}{\text{Esfuerzo total del proyecto consolidado}} * 100$$

El indicador aquí obtenido constituye la media, promedio o estándar que a su vez sirve como término de referencia para comparar el desempeño en cuanto a documentación de cada proyecto respecto a su segmento o grupo objetivo.

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

En el proceso de comparación, un sesgo alto infiere que el proyecto habría generado mayor documentación de sus tareas realizadas; y, un sesgo bajo en cambio infiere que la empresa no habría generado documentación apropiada; sin dejar de anotar que cualquier sesgo atípico por arriba o por debajo de la media constituiría un dato aberrante que además de ser investigado debería excluirse en el análisis de comparabilidad, un indicador ideal sería aquel que guarde razonable aproximación a la media.

En cuestión de proyectos informáticos es imperativo que se dedique la mayor cantidad de tiempo a la documentación. Si la empresa pretende obtener algún tipo de certificación, sea esta: CMMI ó ISO9001:2000, como requisito fundamental es que la empresa lleve sus procesos debidamente documentados. [34] [35]

INDICADOR 5

➤ ***Denominación:***

Complejidad del Negocio y Complejidad Técnica

➤ ***Objetivo:***

Mostrar la complejidad del negocio y complejidad técnica que se presentan en cada proyecto de desarrollo de software y compararlas con el promedio general consolidado de todos los proyectos en su conjunto; distinguir los de alta y baja complejidad para efectos de análisis y examen de sus características; y, analizar la relación entre la complejidad del negocio y complejidad técnica.

➤ ***Dinámica:***

Para obtener el presente indicador se ha procedido a determinar el nivel de complejidad del negocio y técnica a través de información suministrada por las empresas a base de los cálculos promedios de la valoración asignada a distintos factores que derivan tales complejidades, con niveles de medición que van de 0 a 4 según la

evaluación dada por los participantes, se utilizaron las mismas plantillas aplicadas y validadas por el plan piloto [8].

El esquema de comparabilidad del indicador antes descrito consiste en cotejarlo con la media, promedio o estándar que se obtiene para todo el proyecto en su conjunto, de manera que el nivel de desempeño de cada proyecto en particular estará dado por el grado de aproximación por arriba o por abajo que estos observen respecto al promedio general.

➤ ***Interpretación:***

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

En la representación gráfica el presente indicador deriva 3 ilustraciones, una para la complejidad técnica, complejidad del negocio y otra para complejidad técnica versus complejidad del negocio.

La primera gráfica: “complejidad técnica” indicaría cuan complejo han sido técnicamente los proyectos en relación a la organización, suficiencia y coordinación de recursos. En tal sentido un sesgo alto o por arriba del promedio infiere que el proyecto habría implicado una complejidad técnica superior, mientras al contrario un sesgo bajo e inferior a la media derivaría un entorno de poca complejidad; cuya lectura podría permitir inclusive segmentar grupos de proyectos por su

nivel de complejidad, orientado a la fijación de precios de los proyectos y otras estrategias de competitividad.

La segunda gráfica: “complejidad del negocio” indicaría cuan complejo han sido los proyectos en relación al conocimiento del producto y las características, perfiles y necesidades del cliente y la forma de armonizar tales elementos. En tal sentido un sesgo alto o por arriba del promedio infiere que el proyecto habría implicado una complejidad superior, mientras al contrario un sesgo bajo e inferior a la media derivaría un entorno de poca complejidad o buen nivel de conocimiento y manejo del cliente; cuya lectura podría permitir también segmentar grupos de proyectos por su nivel de complejidad de clientes, orientado a la fijación de precios de los proyectos y otras estrategias de competitividad.

La tercera gráfica: “complejidad técnica versus complejidad del negocio” indicaría la relación entre ambas, obteniendo como resultado una separación entre los proyectos participantes, esta separación se daría por proyectos técnicamente complejos; proyectos realizados para negocios complejos; proyectos con alto grado de complejidad; y, proyectos que no implican mayor grado de complejidad.

INDICADOR 6

➤ Denominación:

Porcentaje de error en la estimación del costo

➤ Objetivo:

Determinar el grado de eficiencia en el cumplimiento de los costos estimados para los proyectos, esto es, si su materialización se ha concretado sin excederse de forma importante respecto al presupuesto establecido.

➤ Dinámica:

PARA EL PORCENTAJE DE ERROR POR PROYECTO

El numerador del indicador representa la diferencia entre el costo estimado y real del proyecto; y, el denominador se refiere al costo estimado en el proyecto; cuyo cociente es multiplicado por cien para obtener el indicador en término porcentual, es decir, la proporción del error en estimación del costo en cada proyecto. Al efecto procede la siguiente fórmula:

$$\% \text{ error en estimación del costo} = \frac{\text{costo estimado} - \text{costo real}}{\text{costo estimado}} * 100$$

PARA EL PORCENTAJE CONSOLIDADO DE ERROR EN ESTIMACIÓN

El porcentaje consolidado de error en estimación se trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera

que en el numerador del indicador se representa la diferencia entre los costos estimados y reales consolidados; y, el denominador refiere el costo estimado consolidado; cuyo cociente es multiplicado por cien para obtener el indicador en término porcentual, es decir, la proporción consolidada del error en estimación del costo. Al efecto procede la siguiente fórmula:

$$\% \text{ consolidado de error en estimación del costo} = \frac{\text{costo estimado consolidado} - \text{costo real consolidado}}{\text{costo estimado consolidado}} * 100$$

El indicador aquí obtenido constituye la media o promedio que a su vez sirve como término de referencia para comparar los demás integrantes del estudio.

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 "Análisis y presentación de resultados".

Dicha gráfica daría los porcentajes de error en estimación del costo, donde cualquier diferencia positiva implicaría un margen, una eficiencia en manejar el costo, esto quiere decir, que se habría empleado menos de los recursos estimados a usar en ese proyecto. También determinaríamos en que medida la empresa o el líder de proyecto fue eficiente en manejar los insumos, decimos por insumos todos los elementos que integran un proyecto. Básicamente veríamos

el grado en que la empresa puede utilizar los menores insumos posibles para alcanzar un objetivo. Esto es, la medida que la empresa puede economizar al máximo los insumos. Dichos insumos cuantificados son el costo estimado y costo real.

El sesgo más alto (positivo) sería el proyecto más eficiente en manejar los insumos. Ahora bien, el sesgo más bajo (negativo) sería el proyecto más ineficiente en el manejo de los mismos.

INDICADOR 7

➤ ***Denominación:***

Costo promedio hora real.

➤ ***Objetivo:***

Determinar la inversión por costo hora real empleado en cada proyecto y su comparación con la media general de todos los proyectos objetos del estudio.

➤ ***Dinámica:***

PARA EL COSTO PROMEDIO HORA REAL POR PROYECTO

El numerador del indicador representa el costo real total del proyecto; y, el denominador se refiere al esfuerzo real total empleado en el proyecto. Al efecto procede la siguiente fórmula:

$$\text{Costo promedio hora real} = \frac{\text{Costo real total}}{\text{Esfuerzo real total}}$$

PARA EL COSTO PROMEDIO HORA REAL CONSOLIDADO

El costo promedio hora real consolidado obtiene una media para todos los proyectos participantes, de manera que en el numerador del indicador se representa el costo real total consolidado que tienen todos los proyectos; y, el denominador refiere el esfuerzo real total empleado en todos los proyectos. Al efecto procede la siguiente fórmula:

$$\text{Costo promedio hora real consolidado} = \frac{\text{costo real total consolidado}}{\text{esfuerzo real total consolidado}}$$

El indicador aquí obtenido constituye la media o promedio que a su vez sirve como término de referencia para comparar los proyectos en particular en relación a los insumos utilizados.

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

La representación gráfica obtenida es “el costo promedio real”, en donde, todas las empresas que estén por arriba del indicador promedio, serían más costosas. No así las que estén por debajo al promedio, que implicarían un menor costo promedio real.

Con este indicador se puede determinar si en próximos proyectos convendría o no emplear las mismas personas que colaboraron en el

proyecto del cual se tiene el indicador Costo Promedio Hora, analizando que tan costosos han salido los recursos con respecto al promedio general y adoptar las estrategias más adecuadas en el empleo de tales recursos.

INDICADOR 8

➤ **Denominación:**

Porcentaje de inspecciones realizadas por fases y Porcentaje de inspecciones realizadas en el proyecto

➤ **Objetivo:**

Determinar el porcentaje de inspecciones que se realizan en cada una de las fases; y, determinar el porcentaje de inspecciones totales realizadas.

➤ **Dinámica:**

PARA EL PORCENTAJE DE INSPECCIONES REALIZADAS POR FASE EN EL PROYECTO

Para la obtención del indicador es necesario cuantificar el número de inspecciones que se hayan realizado en cada fase y relacionarlo en términos porcentuales con el número de las tareas totales realizadas en el proyecto; al efecto procede la siguiente fórmula:

$$\% \text{ inspecciones realizadas por fase en el proyecto} = \frac{\text{Total inspecciones por fase}}{\text{Total tareas en el proyecto}} * 100$$

PARA PORCENTAJE DE INSPECCIONES REALIZADAS EN EL PROYECTO

Para la obtención del indicador es necesario cuantificar las inspecciones realizadas en cada proyecto y relacionarlo en términos porcentuales con el número total de tareas realizadas en los mismos, al efecto procede la siguiente fórmula:

$$\% \text{ inspecciones realizadas en el proyecto} = \frac{\text{Total inspecciones}}{\text{Total tareas}} * 100$$

PARA EL PORCENTAJE CONSOLIDADO DE INSPECCIONES DE LOS PROYECTOS

El porcentaje consolidado de inspecciones de los proyectos es un indicador que relaciona las inspecciones totales que han realizado todos los proyectos participantes en el estudio con las tareas totales ejecutadas en tales proyectos, lo que sirve como término de referencia o media general para comparar la situación individual de cada proyecto participante; en consecuencia, el numerador contiene el total de inspecciones que tienen todos los proyectos; y, el denominador refiere el total de tareas empleadas en todos los proyectos. Al efecto procede la siguiente fórmula:

$$\% \text{ consolidado de inspecciones de los proyectos} = \frac{\text{Total inspecciones consolidado}}{\text{Total tareas consolidado}} * 100$$

PORCENTAJE CONSOLIDADO DE INSPECCIONES POR FASES DE LOS PROYECTOS

El porcentaje consolidado de inspecciones por fases de los proyectos es un indicador que relaciona el total de inspecciones por fases que han realizado todos los proyectos participantes en el estudio con las tareas totales ejecutadas en tales proyectos, lo que sirve como término de referencia o media general para comparar la situación individual de cada proyecto participante; en consecuencia, el numerador contiene el total de inspecciones por fases que tienen todos los proyectos; y, el denominador refiere el total de tareas empleadas en todos los proyectos. Al efecto procede la siguiente fórmula:

$$\% \text{ consolidadas de inspecciones por fases de los proyectos} = \frac{\text{Total inspecciones por fase consolidadas}}{\text{Total tareas consolidadas}} * 100$$

➤ **Interpretación:**

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

En la representación gráfica el presente indicador deriva 3 ilustraciones; el porcentaje de las inspecciones realizadas por fase en el proyecto; el porcentaje de inspecciones realizadas en el proyecto; y, el porcentaje consolidado de las inspecciones en las fases.

En la primera, esto es, “porcentaje de las inspecciones realizadas por fase en el proyecto” mostraría el porcentaje que el líder del proyecto habría asignado en cada fase a inspeccionar. Contrastando el mismo con la media, promedio. Un sesgo alto es indicativo que la empresa habría realizado mayor número de inspecciones; un sesgo bajo indicaría que se habría realizado menos inspecciones en la fase indicada.

En la segunda gráfica “el porcentaje de inspecciones realizadas en el proyecto” mostraría el porcentaje que la empresa ha tenido de inspecciones, para posteriormente comparar con el promedio obtenido del estudio. Un sesgo alto indicaría mayor número inspecciones realizadas, no así un sesgo bajo que indicaría lo contrario.

En la tercera gráfica “el porcentaje consolidado de las inspecciones en las fases” mostraría en detalle el promedio consolidado de inspecciones realizadas en las fases.

INDICADOR 9

➤ **Denominación:**

Porcentaje de variación de los requerimientos.

➤ **Objetivo:**

Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto. [8]

➤ **Dinámica:**

El indicador se obtiene sumando los requerimientos iniciales y agregados y restando los eliminados, para luego obtener el porcentaje de requerimientos modificados, agregados, eliminados e iniciales; al efecto proceden las siguientes fórmulas:

total requerimientos = requerimientos iniciales + requerimientos agregados – requerimientos eliminados

$$\% \text{ requerimientos modificados} = \frac{\text{requerimientos modificados}}{\text{total requerimientos}} * 100$$

$$\% \text{ requerimientos agregados} = \frac{\text{requerimientos agregados}}{\text{total requerimientos}} * 100$$

$$\% \text{ requerimientos eliminados} = \frac{\text{requerimientos eliminados}}{\text{total requerimientos}} * 100$$

$$\% \text{ requerimientos iniciales} = 1 - \% \text{ requerimientos (agregados + modificados + eliminados)}$$

Este indicador es obtenido de manera porcentual para facilitar el análisis no solamente interno sino también con los demás proyectos.

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

Dicha gráfica se la realiza mostrando los porcentajes de los 4 tipos de requerimientos (agregados, modificados, eliminados e iniciales). Lo ideal sería que el 100% de los requerimientos iniciales se observen en la gráfica, esto es, que no hayan existido variaciones para no incurrir en costos adicionales.

Sin embargo, en caso de existir requerimientos adicionales (agregados, modificados o eliminados), estos podrían responder a las siguientes razones:

Requerimientos agregados: sus causales pueden ser: deficiencia en la planificación y especificación; debilidades en el levantamiento de información; y, falta de experiencia o de conocimiento del entorno del cliente/usuario.

Requerimientos modificados y eliminados: Su causal principal podría responder a una débil coordinación con los clientes y/o usuarios.

Esta información es de vital importancia para que en futuros proyectos se analicen adecuadamente los encargados de ejecutar el levantamiento de información y de interactuar con los clientes y/o usuarios.

INDICADOR 10

➤ **Denominación:**

Complejidad del Código Fuente.

➤ **Objetivo:**

Controlar la complejidad de los componentes del SW para disminuir el esfuerzo utilizado por terceros en comprenderlo y modificarlo. **[8]**

➤ **Dinámica:**

Para cuantificar la complejidad de cada módulo del proyecto de software se planteó una escala de evaluación arbitraria, pero que consideramos razonablemente aplicable para el efecto, esto es, asignando una calificación de 1 a 5, de manera que los proyectos menos complejos derivarían la menor cuantificación y obviamente los altamente complejos la más alta calificación; evaluación que será asignada por el líder del proyecto en conjunto con el equipo de desarrollo. En este aspecto cabe aclarar que la escala utilizada únicamente responde al criterio de calificación valorado por los encargados del desarrollo del proyecto, ya que no fue posible configurar una encuesta que involucre diversos factores, elementos y particularidades que definen el alcance de la complejidad ciclomática, de manera que respalde el cuantificador obtenido (1 a 5); En tal sentido en la sección de recomendaciones se presenta la sugerencia con respecto a este punto.

El indicador de la referencia relaciona el total de la complejidad ciclométrica obtenida de la forma antes indicada con el total de componente, métodos o procedimientos, obteniendo la siguiente fórmula:

$$\text{Complejidad del código fuente} = \frac{\text{Total de complejidad ciclométrica}}{\text{Total de componentes, métodos o procedimientos}}$$

Para determinar el parámetro de comparación entre las empresas participantes del estudio, se calcula la complejidad promedio de la industria usando la siguiente fórmula:

$$\text{Complejidad promedio} = \frac{\text{Complejidad del código fuente consolidado}}{\text{total de proyectos}}$$

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 "Análisis y presentación de resultados".

Con el gráfico obtenido tendremos que todas las empresas que estén por arriba del indicador promedio, derivarían mayor complejidad en su código fuente, no así las que estén por debajo de este promedio, que podrían catalogarse menos complejas que las demás.

INDICADOR 11

➤ **Denominación:**

Indicador ponderado de la complejidad y demora del aprendizaje.

➤ **Objetivo:**

Determinar la magnitud de cada uno de los proyectos medidos por su complejidad ciclomática y demora en el aprendizaje de los componentes, y su razonabilidad comparativa con el indicador promedio del estudio en su conjunto.

➤ **Dinámica:**

El indicador se formula con dos datos relacionados entre si, estos son, demora en el aprendizaje y la complejidad ciclomática. Al efecto procede la siguiente fórmula:

Indicador ponderado de la complejidad = demora en el aprendizaje * complejidad ciclomática

Para comparar de manera eficiente los proyectos es necesario contrastarlos con el promedio, al efecto se procede: colocar en el numerador el indicador ponderado de la complejidad consolidado, es decir la suma de las ponderaciones de complejidad de todos los participantes del estudio; y, en el denominador el número de proyectos. Al efecto procede la siguiente fórmula:

Indicador ponderado de complejidad promedio = $\frac{\text{Indicador ponderado de la complejidad consolidado}}{\text{Número de proyectos}}$

➤ ***Interpretación:***

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

Para la gráfica obtenida, cualquier proyecto que supere el indicador promedio infiere que su magnitud en cuanto a complejidad y aprendizaje de los componentes supera la media general del segmento en análisis; y, los proyectos que se encuentran por debajo de esta media indicarían que siendo de menor tamaño sin embargo aportan al universo en análisis. En este aspecto es importante destacar que el alejamiento de la media, por arriba o por debajo de la misma no sea de considerable importancia, toda vez que de ser este el caso, se haría evidente la presencia de proyectos que no se encajan al perfil elegido, objetivo fundamental de este indicador que fue ideado por los autores de esta tesis.

INDICADOR 12

➤ **Denominación:**

Tamaño de los componentes de software

➤ **Objetivo:**

Conocer el tamaño (expresado en líneas de código) de los componentes del software para de esta manera conocer el grado de complejidad del proyecto y justificar la ocurrencia de defectos en el desarrollo del mismo.

➤ **Dinámica:**

Con el indicador tamaño de los componentes se pretende periódicamente monitorear las líneas de código que realice cada integrante, a fin de establecer el tamaño de los componentes que han tenido los proyectos. Al efecto se realiza la siguiente operación:

Tamaño de los componentes de software = líneas de código producidas de todo el proyecto

Para comparar los proyectos es necesario contrastarlos con la media. Para su cálculo en el numerador se incluye el total de los componentes de software de los participantes del estudio; y, en el denominador el número de proyectos. Al efecto procede la siguiente fórmula:

$$\text{Tamaño de los componentes de software promedio} = \frac{\text{Tamaño de los componentes de software consolidado}}{\text{Número de proyectos}}$$

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

Un sesgo muy bajo indicaría que el proyecto no habría sido muy extenso. De su lado, un sesgo muy alto indicaría un tamaño mayor del proyecto.

INDICADOR 13

➤ **Denominación:**

Ocurrencia de defectos y fallas por fases

➤ **Objetivo:**

Determinar las fases en las cuales se presentan la mayor cantidad de defectos y fallas.

➤ **Dinámica:**

PARA EL PORCENTAJE DE DEFECTOS Y FALLAS POR FASE POR PROYECTO

Con este indicador se podría observar qué fases del proceso de desarrollo de software son más propensas a la aparición de defectos y fallas. La elaboración del indicador se la realiza colocando en el numerador la suma de todos los defectos; y, en el denominador el total de defectos que se hubieren presentado. Para el cálculo de fallas se

realiza el mismo procedimiento. Al efecto proceden las siguientes fórmulas:

$$\% \text{ defectos en planificación} = \frac{\text{defectos en planificación}}{\text{total defectos del proyecto}}$$

$$\% \text{ fallas en planificación} = \frac{\text{fallas en planificación}}{\text{total fallas del proyecto}}$$

Esta fórmula se repite por cada una de las 7 fases que se ha considerado para el presente estudio: planificación, especificación, diseño, construcción, pruebas, instalación y otros.

Para la comparabilidad es necesario obtener la media general. Para su cálculo en el numerador se incluye el total de defectos presentados en la fase, tomando a todos los proyectos participantes en el estudio; y, en el denominador el total de defectos presentados en los proyectos. Al efecto proceden las siguientes fórmulas:

$$\% \text{ defectos en planificación consolidado} = \frac{\text{defectos en planificación consolidado}}{\text{total defectos consolidado}}$$

$$\% \text{ fallas en planificación consolidado} = \frac{\text{fallas en planificación consolidado}}{\text{total fallas consolidado}}$$

La misma fórmula se la realiza en cada una de las fases.

PARA EL PORCENTAJE CONSOLIDADO DE DEFECTOS Y FALLAS
POR PROYECTO CONSOLIDADO RESPECTO AL NÚMERO TOTAL
DE TAREAS REALIZADAS

Se contrasta los defectos y fallas de todos los proyectos participantes en el estudio contra las tareas realizadas en los mismos. Al efecto proceden las siguientes fórmulas:

$$\% \text{ fallas consolidado} = \frac{\text{total de fallas en los proyectos}}{\text{total de tareas en todos los proyectos}}$$

$$\% \text{ defectos consolidado} = \frac{\text{total de defectos en los proyectos}}{\text{total de tareas en todos los proyectos}}$$

$$\% \text{ defectos en planificación consolidado} = \frac{\text{defectos en planificación consolidado}}{\text{total tareas consolidado}}$$

La misma fórmula se la realiza en cada una de las fases.

➤ ***Interpretación:***

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

La mayor cantidad de defectos y fallas en las fases finales del proceso de desarrollo, implicaría que no se han realizado las inspecciones necesarias en fases iniciales, lo que podría generar demoras y como consecuencias mayor tiempo en la conclusión del proyecto debido a las instancias de corrección requeridas. Este indicador debe ser analizado conjuntamente con el indicador porcentaje del tiempo

utilizado en corrección de defectos y fallas, para determinar el espacio de la demora en la corrección de los mismos.

La otra gráfica ilustra los defectos y fallas consolidadas con respecto al número de tareas consolidadas de los integrantes del estudio.

INDICADOR 14

➤ ***Denominación:***

Tipo de defectos y fallas más ocurrentes, identificados por su severidad

➤ ***Objetivo:***

Conocer qué clase de defectos y fallas son los de mayor presencia en el proyecto de desarrollo de software.

➤ ***Dinámica:***

Los términos usados en el presente indicador tienen los siguientes significados:

Defectos y fallas leves:

Son aquellos de menor importancia y que no impiden el normal funcionamiento del software o aplicación.

Defectos y fallas graves:

Son aquellos que inhabilitan parte del software o aplicación.

Defectos y fallas críticos:

Son aquellos que no permiten la operabilidad del software o aplicación.

PARA EL PORCENTAJE DE DEFECTOS Y FALLAS MÁS OCURRENTES

El indicador se obtiene sumando todos los defectos y fallas del proyecto, por separado, a fin de obtener la proporción que representa cada tipo de defectos y fallas respecto al total de dichas deficiencias.

Al efecto proceden las siguientes fórmulas:

$$\% \text{ de ocurrencia de defectos leves} = \frac{\text{total defectos leves}}{\text{total defectos}}$$

$$\% \text{ de ocurrencia de defectos graves} = \frac{\text{total defectos graves}}{\text{total defectos}}$$

$$\% \text{ de ocurrencia de defectos críticos} = \frac{\text{total defectos críticos}}{\text{total defectos}}$$

Las mismas operaciones se realizan para obtener el porcentaje de ocurrencia de las fallas.

PARA EL PORCENTAJE CONSOLIDADO DE DEFECTOS Y FALLAS MÁS OCURRENTES.

Este indicador considera defectos y fallas de forma consolidada y lo relaciona con el número de tareas consolidadas. Al efecto proceden las siguientes fórmulas:

$$\% \text{ de ocurrencia de defectos consolidado} = \frac{\text{total defectos en los proyectos}}{\text{total de tareas en los proyectos}}$$

$$\% \text{ de ocurrencia de defectos leves consolidado} = \frac{\text{total defectos leves en los proyectos}}{\text{total de tareas en los proyectos}}$$

$$\% \text{ de ocurrencia de defectos graves consolidado} = \frac{\text{total defectos graves en los proyectos}}{\text{total de tareas en los proyectos}}$$

$$\% \text{ de ocurrencia de defectos críticos consolidado} = \frac{\text{total defectos críticos en los proyectos}}{\text{total de tareas en los proyectos}}$$

El mismo proceso aplica para el indicador porcentaje ocurrencia en fallas.

➤ **Interpretación:**

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

El primer gráfico explica los defectos y fallas leves, graves y críticos más ocurrentes por proyecto participante y el promedio de los integrantes.

Los defectos y fallas leves generalmente requieren soluciones de menor tiempo y esfuerzo; los defectos y fallas críticas demandaran mayor tiempo y esfuerzo en su solución, lo que implica mayor requerimiento de recursos.

El segundo gráfico explica el porcentaje de defectos y fallas a nivel consolidado con relación al total de tareas consolidadas.

La ocurrencia de defectos y fallas por lo general responde a la ausencia o a las precarias inspecciones ejecutadas en las etapas iniciales del desarrollo del proyecto.

INDICADOR 15

➤ **Denominación:**

Porcentaje del tiempo utilizado en corrección de defectos y fallas

➤ **Objetivo:**

Determinar qué porcentaje del tiempo en el desarrollo del proyecto se ha empleado en la corrección de defectos y fallas.

➤ **Dinámica:**

PORCENTAJE DEL TIEMPO UTILIZADO EN CORRECCIÓN DE DEFECTOS Y FALLAS

El numerador del indicador representa el esfuerzo utilizado para la corrección de fallas; y, el denominador se refiere al esfuerzo total del proyecto; cuyo cociente es multiplicado por cien para obtener el indicador en término porcentual, es decir, la proporción de tiempo utilizado en corrección de fallas. Al efecto procede la siguiente fórmula:

$$\% \text{ de tiempo utilizado en corrección defectos} = \frac{\text{Esfuerzo utilizado para corrección de defectos}}{\text{Esfuerzo de desarrollo del proyecto}} * 100$$

El mismo proceso aplica para el indicador porcentaje utilizado en corrección de fallas.

PORCENTAJE CONSOLIDADO DEL TIEMPO UTILIZADO EN CORRECCIÓN DE DEFECTOS Y FALLAS

El porcentaje consolidado del tiempo utilizado en corrección del defectos y fallas se trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera que en el numerador del indicador se representa el esfuerzo conjunto que han empleado todos los proyectos en la corrección de defectos; y, el denominador refiere el esfuerzo total empleado en todos los proyectos; cuyo cociente es multiplicado por cien para obtener el indicador en término porcentual, es decir, la proporción del esfuerzo total que han dedicado todos los proyectos para la corrección de defectos. Al efecto procede la siguiente fórmula:

$$\% \text{ consolidado de tiempo utilizado en corrección de defectos} = \frac{\sum \text{Esfuerzo utilizado para corrección de defectos proyectos}}{\text{Esfuerzo utilizado para el desarrollo de todos los proyectos}} * 100$$

El mismo proceso aplica para el indicador porcentaje consolidado utilizado en corrección de fallas.

➤ **Interpretación:**

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

Un sesgo por encima del promedio indicaría que en dicho proyecto habría existido un mayor esfuerzo en solucionar los problemas originados por mala planificación, falta de inspecciones, o mal levantamiento de requerimientos. Aquellos proyectos que se ubiquen por debajo del promedio se podría decir que tuvieron menos esfuerzo en corrección de defectos y fallas.

INDICADOR 16

➤ **Denominación:**

Eficiencia en atención de defectos y fallas.

➤ **Objetivo:**

Con este indicador se pretende conocer el grado de efectividad de la empresa para atender los defectos y fallas que se presenten.

➤ **Dinámica:**

Para el cálculo de este indicador se requiere previamente determinar el número de días en atender los defectos y fallas; en tal caso es necesario deducir de la fecha inicio real, la fecha de registro. Al efecto procede la siguiente operación:

Número días en atender el defecto = fecha inicio real tarea defecto – fecha registro tarea defecto

El mismo proceso se aplica para fallas.

EFICIENCIA EN ATENCIÓN DE DEFECTOS Y FALLAS POR PROYECTOS

El numerador del indicador representa la suma del número de días en atender los defectos en el proyecto; y, el denominador se refiere al total de defectos en el proyecto. Al efecto procede la siguiente fórmula:

$$\text{Promedio de días en atender defectos} = \frac{\sum \text{Número de días en atender todos los defectos}}{\text{número tareas tipo defectos}}$$

El mismo proceso aplica para el indicador eficiencia en atención de fallas.

EFICIENCIA CONSOLIDADA EN ATENCIÓN DE DEFECTOS Y FALLAS

La eficiencia consolidada en atención de defectos y fallas se trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera que en el numerador del indicador se representa la suma del promedio de días en atender defectos en todos los proyectos; y, el denominador refiere al total de proyectos participantes del estudio. Al efecto procede la siguiente fórmula:

$$\text{Promedio de días en atender defectos de los proyectos} = \frac{\sum \text{Promedio de días en atender defectos en los proyectos}}{\text{número de proyectos}}$$

El mismo proceso aplica para el indicador eficiencia consolidada en atención de fallas.

➤ ***Interpretación:***

Las gráficas con los resultados obtenidos del presente indicador se las muestran y analizan en el Capítulo 5 “Análisis y presentación de resultados”.

Los proyectos que se encuentren por arriba del promedio han tomado más tiempo en dar atención a los problemas presentados, lo que eventualmente podría tener incidencia negativa en la calidad del servicio; y, los proyectos que se ubiquen por debajo del promedio denotarían mayor eficiencia en este aspecto.

INDICADOR 17

➤ **Denominación:**

Promedio de defectos por hora de programación.

➤ **Objetivo:**

Determinar cuántos defectos se presentan por cada hora de programación.

➤ **Dinámica:**

PROMEDIO DE DEFECTOS POR HORA DE PROGRAMACIÓN

El numerador del indicador representa el número de defectos en el proyecto; y, el denominador se refiere al esfuerzo en el desarrollo del mismo. Al efecto procede la siguiente fórmula:

$$\text{Promedio de defectos por hora de programación} = \frac{\text{número de defectos}}{\text{esfuerzo en el desarrollo}}$$

PROMEDIO CONSOLIDADO DE DEFECTOS POR HORA DE PROGRAMACIÓN

El promedio consolidado de defectos por hora de programación se trata del indicador que se obtiene para todos los proyectos en su conjunto, de manera que en el numerador del indicador se representa la suma del promedio de defectos por hora de programación; y, el denominador refiere al total de proyectos participantes del estudio. Al efecto procede la siguiente fórmula:

$$\text{Promedio defectos por hora de programación} = \frac{\sum \text{Promedio defectos por hora de programación en los proyectos}}{\text{total proyectos}}$$

➤ **Interpretación:**

La gráfica con los resultados obtenidos del presente indicador se muestra y analiza en el Capítulo 5 “Análisis y presentación de resultados”.

Los proyectos que se encuentren por arriba del promedio indicarían que han existido mayor cantidad de defectos por hora de programación, lo que eventualmente podría tener incidencia en el costo del proyecto debido al mayor uso de recursos; y, los proyectos que se ubiquen por debajo del promedio denotarían mayor eficiencia en este aspecto.

Es incorrecto decir que una empresa se encuentre en peores condiciones que otra evaluando únicamente su número elevado de defectos. Este indicador se genera para conocer de alguna manera si se justifica o no el número de defectos presentados en un desarrollo.

Con los nuevos indicadores generados se ha creado un nuevo Plan de Métricas detallado en el **ANEXO 3** que se validaría con los proyectos participantes en el estudio, para en el capítulo 5 analizar si han sido factibles los cambios realizados.

CAPÍTULO IV

4. PROCESO DE APLICACIÓN DEL INSTRUMENTO

El presente capítulo explica el proceso utilizado para la aplicación del instrumento de medición (plantillas), el mismo que tuvo lugar en las ciudades de Quito y Guayaquil durante los meses de Junio a Septiembre; al efecto fue necesario observar los siguientes factores claves de organización:

- Reuniones adecuadamente planificadas y coordinadas que optimicen el tiempo y garanticen el cumplimiento de los objetivos;
- Herramientas de levantamiento de información (plantillas) que puedan ser manejadas de forma sencilla pero que provean la información necesaria para la realización del proyecto; y,
- Documentación suficiente, competente y pertinente que permita además de contribuir de manera eficaz para la realización del trabajo, sustentar y respaldar su contenido.

Este proceso demandó: el desarrollo de un instrumento en Microsoft Access que optimice la recopilación de datos; la validación del plan de capacitación; el diagrama resumen del proceso GQIM aplicado para obtener las plantillas; y, un resumen ejecutivo de las sesiones que se mantuvieron con las empresas.

4.1. Desarrollo de un instrumento en Access para la recopilación de datos

En el Capítulo 3, Sección 3.3, se dio a conocer las razones por lo que se seleccionó Microsoft Access como herramienta para el desarrollo de las plantillas de recolección de datos.

El instrumento en Access desarrollado consta de 8 pantallas (plantillas) cuyo contenido se explicará brevemente en los párrafos siguientes, junto con las sugerencias y recomendaciones recibidas de las empresas participantes, en la medida de su factibilidad.

El proceso de ingreso de datos en las diferentes pantallas, debió responder a un orden de llenado; en mismo que podía iniciarse en las pantallas 1, 2 ó 3, de manera indiferente, para luego poder llenar la pantalla 4, indispensable a su vez para cumplimentar las pantallas 5, 7 u 8 de igual forma utilizadas de manera indistinta; sin embargo, para el llenado de la pantalla 6 cuando sea del caso necesariamente se requiere haber cumplimentado la pantalla 5. Con tal ordenamiento el esquema de pantallas es el siguiente:


Pantalla 1: Ingreso de fase

La pantalla 1 tiene relación con el nombre de la fase y la descripción de la misma, tal como se demuestra en la figura 4.1.

Las fases preestablecidas en la pantalla son: planificación, especificación, diseño, construcción, pruebas, instalación y otros, con 3 opciones, agregar, modificar y eliminar, a fin de que las empresas ajusten sus propias fases, caso en el que se debió realizar un proceso de equivalencia.

De esta forma y con el esquema de levantamiento e ingreso de información aquí referido, las empresas participantes cumplimentaron los datos que se exige en esta pantalla, lo que permitió continuar con las pantallas siguientes.

Figura 4.1. Ingreso de fase



The screenshot shows a software window titled "Ingreso de Fases". The window has a blue header bar with the title and a close button. Below the header, the word "Fase" is displayed in a large blue font. To the right of "Fase" is a small image of interlocking gears. Below this, there are two input fields: "Nombre:" with the value "Planificación" and "Descripción:" with the text "Corresponde a la realización de tareas encamadas a detallar de mejor manera cada una de las siguientes fases del proyecto y los recursos utilizados para la realización de las mismas." At the bottom of the window, there is a toolbar with icons for search, save, delete, and navigation. The navigation part shows a sequence of arrows and the number "1" followed by "de 7", indicating the current page in a list of 7 items.

Pantalla 2: Ingreso de rol

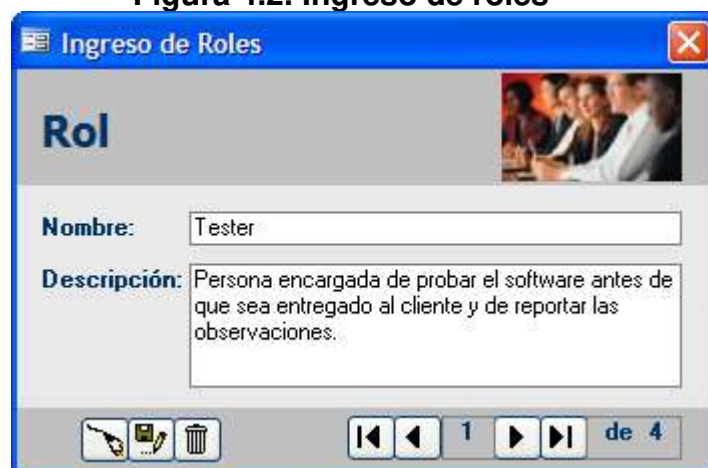
La pantalla 2 tiene relación con el nombre del rol y la descripción del mismo, tal como se demuestra en la figura 4.2.

Se ha denominado rol al cargo o función que ostenta o desarrolla una persona en un proyecto.

Como valor agregado en esta pantalla, cuando una empresa ha logrado ingresar todos los roles existentes, a futuro para otros proyectos únicamente le corresponderá seleccionarlos evitando confusiones.

De la forma descrita y con el esquema de levantamiento e ingreso de información aquí referido, las empresas participantes cumplimentaron los datos que se exige en esta pantalla, lo que permitió continuar con las pantallas siguientes.

Figura 4.2. Ingreso de roles



The screenshot shows a software window titled "Ingreso de Roles". The window has a blue header bar with the title and a close button. Below the header, the word "Rol" is displayed in a large, bold, blue font. To the right of "Rol" is a small image of three people in a meeting. Below this, there are two input fields: "Nombre:" with the text "Tester" and "Descripción:" with the text "Persona encargada de probar el software antes de que sea entregado al cliente y de reportar las observaciones." At the bottom of the window, there is a toolbar with icons for a mouse, a notepad, a trash can, and navigation buttons (back, forward, and a page indicator showing "1 de 4").

Pantalla 3: Ingreso de personas

La pantalla 3 contiene la información relacionada con los nombres y apellidos de las personas; y, la experiencia de las mismas en años de desarrollo de proyectos informáticos, tal como se demuestra en la figura 4.3.

Como valor agregado en esta pantalla, cuando una empresa ha logrado ingresar todas las personas, a futuro únicamente corresponderá la asignación de proyectos.

De la forma anotada y con el esquema de levantamiento e ingreso de información aquí referido, las empresas participantes cumplimentaron los datos que se exige en esta pantalla, lo que permitió continuar con la pantalla 4.

Figura 4.3. Ingreso de personas



The screenshot shows a web application window titled "Ingreso de Personas". The main heading is "Personas" with a small image of three people. Below the heading are three input fields: "Nombres:" with the value "Andrés Vicente", "Apellidos:" with the value "Paz Salazar", and "Experiencia en el Desarrollo:" with a numeric input containing "4" and a label "[Años]". At the bottom, there is a navigation bar with icons for search, edit, and delete, and a pagination control showing "1 de 6".

Pantalla 4: Datos generales del proyecto.

Como consta en la figura 4.4, esta pantalla contiene los principales datos con información general del proyecto tales como:

- Nombre del proyecto;
- Número de requerimientos;
- Tipo de proyecto, de aplicación;
- Fecha de inicio y fin del proyecto;
- Personal estimado, personal real;
- Esfuerzo estimado y real;
- Complejidad técnica y de negocio;
- Satisfacción de usuario y cliente;
- Personal involucrado en el proyecto;
- Interrupciones del proyecto; y,
- Costos directos e indirectos y total del proyecto.

Además, entre las sugerencias y recomendaciones recibidas de las empresas para la presente pantalla y que fueron admitidas por considerarlas de utilidad práctica están las siguientes:

- Agregar un control para facilitar el ingreso de las fechas;
- Desglosar el costo total en costo directo e indirecto; y,
- Agregar la opción comentarios.

De esta forma y con el esquema de levantamiento de información aquí referido, las empresas participantes cumplimentaron los datos que se

exige en esta pantalla, lo que permitió continuar con las pantallas siguientes.

Figura 4.4. Datos generales del proyecto

Proyectos

Nombre: Caso de Ejemplo Nro. Requerimientos: 8

Tipo de Aplicación de Software: Financiera [Ej: Financiera, Administración, Contabilidad, etc.]

Tipo de Proyecto de Software: Nuevo Proyecto [Ej: Nuevo Proyecto, Ampliación o Mejora, etc.]

Fecha	Personal	Esfuerzo Total	Satisfacción	Complejidad
Inicio: 03-Jul-06	Estimado: 5	Estimado: 357	Usuario: 5 [Calcular]	Negocio: 4 [Calcular]
Fin: 18-Ago-06	Real: 5	Real: 323	Cliente: 5 [Calcular]	Técnica: 4 [Calcular]

Rotación:

Persona	Rol	Rotación	Exp. Desarrollo	Exp. Negocio	Costo Hora
Paz Andres	Analista	Permaneció el tiempo requerido	1	1	\$ 1,00
Paz Andres	Programador	Permaneció el tiempo requerido	1	1	\$ 1,00
Calderón Luis	Gerente Comercial	Permaneció el tiempo requerido	1	1	\$ 1,00
González Orlando	Tester	Permaneció el tiempo requerido	1	1	\$ 1,00
Monteverde Danny	Programador	Permaneció el tiempo requerido	1	1	\$ 1,00
Ruiz Saul	Tester	Permaneció el tiempo requerido	1	1	\$ 1,00
Villa Juan	Tester	Permaneció el tiempo requerido	1	1	\$ 1,00

Interrupciones:

Tiempo (Dias)	Causa
0	

COSTO TOTAL DIRECTO ESTIMADO: TOTAL DE TAREAS (Esf. Estimado de Recurso x Tarea * Costo Hora Recurso)	\$ 357,00	COSTO TOTAL DIRECTO REAL: TOTAL DE TAREAS (Esf. Estimado de Recurso x Tarea * Costo Hora Recurso)	\$ 323,00
COSTO TOTAL INDIRECTO ESTIMADO: (Otros Costos del Proyecto)	\$ 0,00	COSTO TOTAL INDIRECTO REAL: (Otros Costos del Proyecto)	\$ 0,00
COSTO TOTAL ESTIMADO DEL PROYECTO:	\$ 357,00	COSTO TOTAL REAL DEL PROYECTO:	\$ 323,00

Comentarios

1 de 1

Pantalla 5: Definición de Tareas / Entregables

Previo a la explicación de la presente pantalla es necesario referir que se ha denominado tarea a todo lo que implique el uso de personal y que demande un esfuerzo en horas, en tal sentido la configuración de la misma implica la siguiente estructura y aplicación:

Como consta en la figura 4.5, la pantalla contiene los datos que se prescriben a continuación:

- Proyecto a ingresar una tarea;
- Fase en la que se presentó la tarea;
- Nombre de la tarea;
- Tipo de tarea realizada;
- Documento y número de páginas de la tarea (si se generó)
- Fechas inicio y fin, estimados y reales;
- Duración de la tarea;
- Costo real y estimado; y,
- Personal involucrado en la tarea con sus horas de aportación;

Las sugerencias y recomendaciones que consideramos aplicables en esta pantalla fueron las siguientes:

- Crear una pantalla previa (Figura 4.6.) que visualice las tareas ingresadas y que permita filtrarlas por fase o proyecto según el caso; al efecto a dar doble clic sobre el registro se accede a la información de la tarea, así como también que para ingresar una nueva se oprime el botón inferior derecho “Agregar nueva tarea/entregable”.
- En la pantalla 4.5, para el “tipo de tarea” se colocaron las siguientes opciones
 - a. **Inspección:** Para monitorear el cumplimiento de los objetivos en las tareas;

- b. **Cambio en los requerimientos:** Para registrar requerimientos no definidos previamente por el cliente;
- c. **Reporte de Defecto:** Para identificar una tarea como defecto;
- d. **Reporte de Fallas:** Para identificar una tarea como falla; y,
- e. **Tarea Normal:** Cualquier tarea que no se enmarquen en los tipos antes mencionados.

De esta forma y con el esquema de levantamiento de información aquí referido, las empresas participantes cumplimentaron los datos que se exige en esta pantalla, lo que permitió continuar con la pantalla 6 si es del caso, o las subsiguientes.

Figura 4.5. Definición de tareas/entregables

Definición de Tareas / Entregables

Definición de Tareas / Entregables

Proyecto: Caso de Ejemplo **Fase:** Planificación

Tipo de Tarea:

Tarea Normal
 Inspección
 Cambio de Requerimiento
 Reporte de (DEFECTO)
 Reporte de (FALLAS)

Nombre de la Tarea: Selección de base de datos

Responsable: Paz Andres

Páginas: 0 **Documentación:**

Descripción: Determinación de parte del departamento de sistemas para escoger una base de datos optima para almacenar y trabajar con los datos del anexo.

Criterio de Terminación: Selección óptima de la base de datos

Fecha Estimada

Inicio: 14-Jul-06
Fin: 14-Jul-06

Fecha Real:

Inicio: 15-Jul-06
Fin: 15-Jul-06

Duración:

Estimada: 1 [Días]
Real: 1 [Días]

Costo Total Tarea:

Estimada: \$ 8,00
Real: \$ 4,00

Recursos y Esfuerzo:

	Rol	Persona	Esfuerzo Estimado (horas)	Esfuerzo Real (horas)
▶	Programador	Paz Andres	2	1
	Tester	González Orlando	2	1
	Tester	Ruiz Saul	2	1
	Tester	Villa Juan	2	1
*				

Total Esfuerzo Estimado: 8
Total Esfuerzo Real: 4

Comentarios Consultar Tarea / Entregables

Figura 4.6. Filtro de tareas/entregables

Definición de Tareas / Entregables

Proyecto: Caso de Ejemplo **Fase:**

Proyecto	Fase	Tarea Nombre	Tarea Responsable	Esf. Estimado	Esf. Real
▶ Caso de Ejemplo	Planificación	Levantamiento de Requerimientos	Calderón Luis	15	15
Case de Ejemplo	Planificación	Capacitación externa de personal	Calderón Luis	6	4
Case de Ejemplo	Planificación	Capacitación interna de personal	Villa Juan	40	20
Case de Ejemplo	Planificación	Análisis de precio	Calderón Luis	4	1
Case de Ejemplo	Planificación	Selección de base de datos	Paz Andres	8	4
Case de Ejemplo	Diseño	Diseño de base de datos	Paz Andres	40	24
Case de Ejemplo	Diseño	Entorno gráfico del software atcompras	Paz Andres	25	13
Case de Ejemplo	Construcción	Creación de maestro at compras	Paz Andres	12	6
Case de Ejemplo	Construcción	Creación de método de grabación atcompras	Paz Andres	8	8
Case de Ejemplo	Construcción	Creación de base de datos para atcompras	Paz Andres	1	1

Total Esfuerzo Estimado: 357 [horas]
Total Esfuerzo Real: 323 [horas]

Pantalla 6: Reporte de problemas o cambios en requerimientos

Previo a la explicación de esta pantalla cabe indicar que para realizar un reporte de problemas o cambios en requerimientos es condición indispensable ingresar una tarea de tipo defecto, falla o cambio en el requerimiento de acuerdo a lo señalado en la pantalla 5 (figura 4.5.).

Como consta en la figura 4.7, la pantalla contiene los datos que se prescriben a continuación:

- Proyecto en donde se presento la tarea;
- Fase en la que se presentó la tarea;
- Tipo de tarea realizada;
- Estado y severidad del defecto o falla;
- Tipo y clase de requerimiento presentado;
- Fecha en la que se registro el problema o el cambio en el requerimiento;
- Fecha en la que se trato el problema o cambio en el requerimiento;
- Método que detecto el problema o el cambio en el requerimiento;
- Descripción del problema o cambio en el requerimiento;
- El componente que afecto el problema o el cambio en el requerimiento; y,
- Causa y efecto del problema o cambio en el requerimiento.

En adición, entre las sugerencias y recomendaciones recibidas de las empresas para la presente pantalla y que fueron admitidas por considerarlas de utilidad práctica están las siguientes:

- Crear una pantalla previa (Figura 4.8) que visualice las tareas ingresadas del tipo defecto, falla o cambio en el requerimiento y que permita filtrarlas por fase, proyecto o el tipo de tarea según el caso; al efecto al dar doble clic sobre el registro se accede a la información de la tarea.
- Que en la figura 4.7, en campo descripción, se incluya guías - leyenda para una mejor descripción del problema, con los términos “¿Qué?”, “¿Cuándo?”, “¿Dónde?” y “¿Cómo?”, a fin de orientar al usuario para una mejor utilización.
- Que en la figura 4.7, campo causas, se coloquen guías con las leyendas: “Infraestructura”; “Herramientas SW”; “Políticas/ Procedimientos”; y, Personal.
- Que en la figura 4.7, campo efectos, se coloquen guías para conocer los efectos más frecuentes de los problemas: “stress del cliente”, “retrasos”, “retrabajo” y “aumento de costos”.
- Que en la sección “Método de detección del problema” de la figura 4.7, se agregue la opción cliente.
- Que en la sección “Componente afectado” de la figura 4.7, se incorpore las opciones: *proyecto*, *sistema*, *módulo* y *clase/método*.

De esta forma y con el esquema de levantamiento de información aquí referido, las empresas participantes cumplimentaron los datos que se exige en esta pantalla, lo que permitió continuar con las siguientes pantallas, si es del caso.

Figura 4.7. Reporte de problemas

Reporte de Problemas o Cambios en Requerimientos

Proyecto: Caso de Ejemplo **Tarea:** Defecto en módulo contable 504 **No. Reporte:** 456

Fase: Pruebas **Tipo:** Defecto **Fecha:** 08-May-06

Estado: Abierto **Severidad:** Grave **Requerimiento:** Tipo: [] Funcional (No): [] **Esfuerzo:** Estimado: 6 [Horas] Real: 12 [Horas]

Fecha Abierto: 10-May-06

Método que detectó el Problema:

Inspección Caja Negra Cliente Peer review Caja Blanca P. Stress Simulación Otro: [] Otro: [] **Observaciones:** []

Descripción del Problema:

¿Qué? ¿Cuándo? ¿Dónde? ¿Cómo? Problema con la modificación de las fechas en el módulo contable al momento de especificar los nuevos sectores asegurados

Componente Afectado:

Proyecto: Caso de Ejemplo Sistema: [] Módulo: Contable Clase/Método: fecha

¿Por qué se produjo? (Causas)

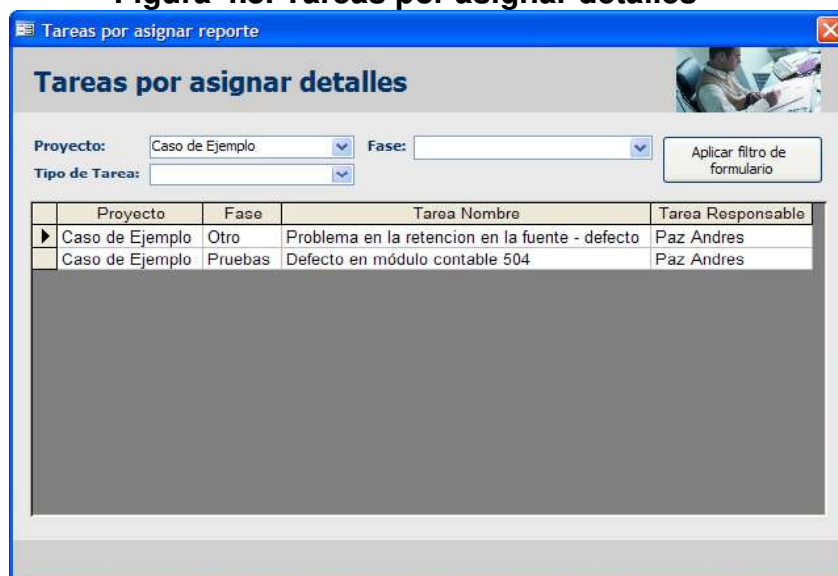
Infraestructura Herramientas SW Políticas/Proced. Personal Falta del personal al dejar pasar por alto las consideraciones hechas por el solicitante

Impacto en el Proyecto (Efectos)

Stress cliente Retrasos Aumento de costos Retrabajo Retraso en la publicación del módulo final

Comentarios [] [] []

Figura 4.8. Tareas por asignar detalles



Proyecto	Fase	Tarea Nombre	Tarea Responsable
▶ Caso de Ejemplo	Otro	Problema en la retencion en la fuente - defecto	Paz Andres
Caso de Ejemplo	Pruebas	Defecto en módulo contable 504	Paz Andres

Pantalla 7: Líneas de código

Como consta en la figura 4.9, la pantalla contiene los datos que se prescriben a continuación:

- Proyecto;
- Mes;
- Semana;
- Recursos que han desarrollado líneas de código; y,
- El total de líneas de código realizadas en el mes y semana ingresados previamente.

Las empresas participantes explicaron lo complicado de llenar el campo líneas de código realizadas, aspecto que se revisará en el Capítulo 5

“Análisis y presentación de resultado” Sección 5.1 “Análisis de indicadores obtenidos”.

Figura 4.9. Líneas de código producidas

	Total Lineas de Código:	Recurso
	450	Paz Andres
▶	1350	Monteverde Danny
	240	González Orlando
*		

Registro: 1 de 1

Pantalla 8: Complejidad del software

Como consta en la figura 4.10, la pantalla contiene los datos que se prescriben a continuación:

- Proyecto;
- Módulo del proyecto;
- Horas para desarrollar el modulo;
- Número de defectos presentados en el modulo;
- Complejidad; y,
- Horas para comprender el modulo se iba a realizar.

Las sugerencias y recomendaciones que consideramos aplicables en esta pantalla fueron las siguientes:

- Solicitar que la pantalla sea cumplimentada por el encargado del proyecto en conjunto con el equipo de desarrollo.

De esta forma y con el esquema de levantamiento de información aquí referido, las empresas participantes cumplimentaron los datos que se exige en esta pantalla, lo que permitió continuar con el estudio.

Figura 4.10. Complejidad del software



The screenshot shows a web form titled "Complejidad del Software". The form includes a header with the title and a decorative image. Below the header, there is a instruction box: "Debe ser llenada por el Encargado del Proyecto en conjunto con las personas que participaron en el desarrollo del mismo." The form fields are: "Proyecto:" with a dropdown menu showing "Caso de Ejemplo"; "Módulo:" with a text input field containing "Compras"; "Horas para desarrollarlo:" with a text input field containing "450"; "Número de Defectos:" with a text input field containing "45"; "Complejidad:" with a text input field containing "3" and a label "[Escala del 1 al 5]"; and "Horas para comprenderlo:" with a text input field containing "20". At the bottom of the form, there are navigation icons (back, forward, home, delete) and a page indicator showing "1 de 3".

4.2. Plan de capacitación para empresas desarrolladoras de software

Como paso inicial para la aplicación del plan de capacitación explicado en el Capítulo 3, Sección 3.6, se estimó conveniente validar su contenido en 3 empresas, elegidas por evidenciar mayores exigencias organizativas, de cuyas sugerencias y recomendaciones se consideró los siguientes cambios:

- Se diseñó diapositivas alusivas al marco teórico, agregadas con los números 2 y 3;
- Se diseñó diapositivas alusivas a la descripción del proyecto, agregadas con los números 4 y 5;
- En las diapositivas 23 a la 26 “Objetos de interés”, se debió corregir un error que presentaban las imágenes; y,
- Se agregaron las diapositivas 47, 48, 49 y 50 relacionadas con las plantillas de recolección de datos.
- Se colocó en todas las diapositivas la alusión de la reserva de derechos “copyright”.

Con las precitadas adecuaciones se obtuvo la nueva versión del plan de capacitación, que fue difundido a todas las empresas participantes.

ANEXO 4

4.3. Diagrama GQIM explicativo

Para continuar con el proceso de aplicación del instrumento fue necesario crear un diagrama que permita la explicación rápida del proceso GQIM, del cual se obtuvieron las plantillas de recolección de datos explicadas en el Capítulo 3, Sección 3.7 del presente estudio.

En este aspecto precisa indicar que el diagrama GQIM explicativo consta de 7 niveles mismos que en resumen se refieren a continuación:

1. Indica los 3 objetivos principales que caracterizan el negocio de las empresas y que tienen relación con maximizar la satisfacción del cliente, mejorar la calidad del producto de software y mejorar la productividad.;
2. Identifica los sub-objetivos del negocio, que se derivan de la lista entidad pregunta (proceso GQIM) ;
3. Muestra los objetivos de medición que están relacionados con el nivel superior (sub-objetivos) a través de líneas;
4. Muestra las preguntas que deben ser contestadas para asegurar el cumplimiento del objetivo de medición referido en el tercer nivel;
5. Prescribe los indicadores que ayudarán a responder las preguntas detalladas en el nivel anterior;
6. Detalla los datos necesarios para formar los indicadores; y finalmente,
7. Concreta las plantillas de medición que contienen los datos.

El diagrama del GQIM explicativo desarrollado para este estudio se incluye en el **ANEXO 5**.

Con el propósito de explicar en detalle la metodología seguida para la obtención del diagrama GQIM, derivado del Proceso GQIM facilitado por el Plan Piloto [8] excepto la sección de indicadores y datos, que han sido desarrollados por el presente estudio. En **ANEXO 6** se detalla la metodología.

4.4. Transferencia de conocimientos y visitas a empresas.

Con el instrumento de medición desarrollado en Access, la validación del plan de capacitación y el diagrama explicativo del GQIM (impreso en Hoja A3) se procedió a contactar a las empresas que se tenía en la base de datos, indicadas en el Capítulo 3, Sección 3.5., a fin de concretar las sesiones de trabajo correspondientes.

4.4.1. Sesión 1: “Presentación y explicación del proyecto de investigación”

La sesión 1 en promedio tuvo una duración de 35 minutos, para lo cual fue necesario trabajar con la siguiente información:

- **Resultados del proyecto anterior:** Como elemento introductorio se presento y explicó los resultados del proyecto anterior a través de gráficas impresas que básicamente tratan lo siguiente: [4]
 - Tipos de proyectos que se tenían en la industria;

- Duración de los proyectos;
 - Metodologías y modelos utilizados;
 - Herramientas para la gestión de los proyectos;
 - Ciclo de desarrollo de los proyectos;
 - Interrupciones;
 - Fases de participación del cliente/ usuario;
 - Complejidad de los proyectos;
 - Esfuerzo empleado del personal;
 - Estándares que usan las empresa;
 - Porcentaje de inspecciones realizadas, entre otros
- **Diagrama explicativo GQIM:** en hoja A3 se llevó impreso el diagrama explicado GQIM, enviando a cada empresa el digital del mismo.
 - **Plan de Capacitación:** Se llevó impreso la última versión del Plan de Capacitación tratado en el punto 4.2.

La explicación del nuevo estudio se realizó con el Diagrama GQIM, el plan de capacitación se dejó a cada empresa para que lo revisen y que conozcan más sobre el tema.

4.4.2. Sesión 2: “Capacitación sobre la herramienta de recolección de datos” . “Aplicación de una encuesta”

La sesión 2 en promedio tuvo una duración de 40 minutos, para su realización fue necesaria la siguiente información impresa:

- **Plantillas:** Las plantillas de recolección para hacer la explicación de cada uno de los datos que la forman;
- **Diagrama explicativo QGIM:** El diagrama para mostrar los indicadores y hacer referencia de la obtención de las plantillas; y,
- **Encuesta:** Se realizó una encuesta para conocer la forma de recolección de datos que preferirían aplicar las empresas.

Se receptaron comentarios y sugerencias respecto a las plantillas, analizando con el personal de la empresa la factibilidad del llenado de las mismas, esto es, ver que dato sería más complicado de llenar.

El resultado de la encuesta en mención fue expuesto en el Capítulo 3 Sección 3.3.

Una vez desarrollado el programa que recolectaría los datos (Capítulo 4, Sección 4.1.), se procedió a enviar vía correo electrónico a las empresas y esperar un tiempo prudencial para realizar la siguiente sesión de verificación de los datos ingresados.

4.4.3. Sesión 3: “Verificación de datos recolectados y recepción de sugerencias de la herramienta de recolección”

La sesión 3 tuvo una duración promedio de 20 minutos, donde se procedió a revisar plantilla por plantilla, verificando que los datos estén siendo ingresados correctamente, para así, a futuro no tener problemas e inconsistencias.

Conforme se iba avanzando en el revisado de los datos se iban aceptando las sugerencias y se iban resolviendo las dudas.

4.4.4. Sesión 4: “Monitoreo de la actividad de recolección y verificación de los datos”

La sesión 4 fue de monitoreo, con una duración promedio de 10 minutos. En esta sesión se veía y analizaba el avance de la empresa en el llenado de las plantillas al tiempo de verificar si estaban siendo llenadas correctamente. Al igual que en las sesiones anteriores se resolvían las dudas y aceptaron sugerencias.

CAPÍTULO V

5. ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

El objetivo de este capítulo es presentar y analizar los indicadores obtenidos en el presente estudio; demostrar las hipótesis formuladas; y, realizar la evaluación comparativa de los resultados obtenidos con las experiencias de otros países.

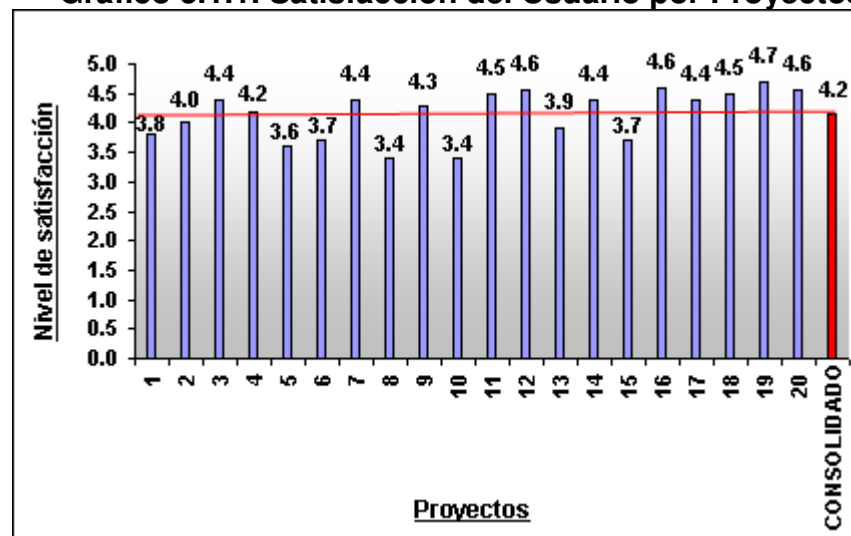
5.1. Análisis de indicadores obtenidos

INDICADOR 1

Satisfacción del Cliente y Satisfacción del Usuario versus Esfuerzo del cliente y esfuerzo del usuario.

Este indicador se compone de 4 gráficas que procedemos a explicar a continuación:

Gráfico 5.1.1. Satisfacción del Usuario por Proyectos



Como se ha explicado en el capítulo 3, este indicador mide la satisfacción del usuario respecto del proyecto elaborado por el proveedor, a través de la metodología que usa para el efecto, indicadores de evaluación cuantificados en puntuaciones de 1 a 5, de cuyo resultado la gráfica permite inferir los aspectos que se analizan a continuación:

El promedio general consolidado de todos los proyectos participantes alcanza una puntuación de 4.2 sobre 5, esto es, que el nivel de satisfacción del usuario promedio es del 83% respecto a los parámetros evaluados.

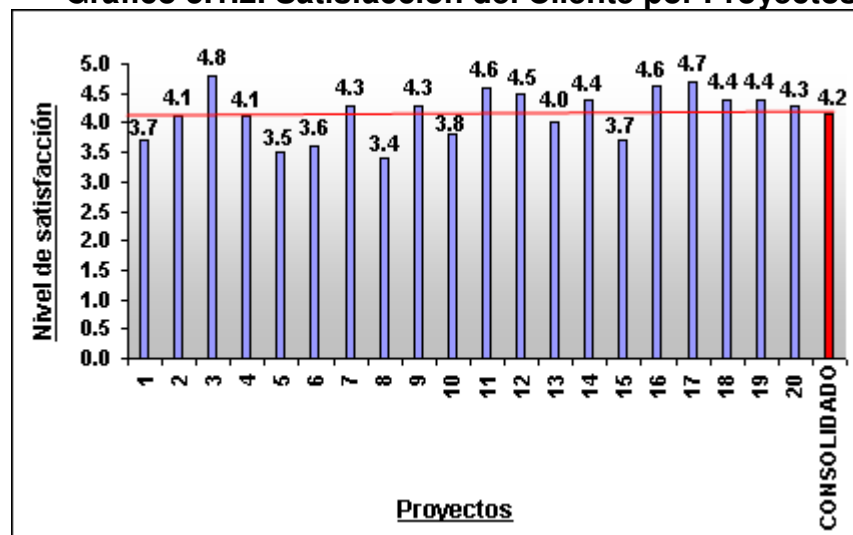
De su lado, al comparar el promedio antes indicado con cada uno de los proyectos en particular se determina que 12 de ellos se encuentran por arriba del mismo, es decir, que estos encuentran un mayor nivel de satisfacción que el promedio general, entre los cuales se encuentra el caso del proyecto 19 que manifiesta un nivel de satisfacción de 4.7 puntos sobre 5 equivalente al 94% de satisfacción, razón que a su vez explica que el usuario en análisis estaría prácticamente satisfecho con la totalidad de los parámetros medidos.

Por su parte existen 8 proyectos que manifiestan un nivel de satisfacción del usuario inferior al promedio general de 4.2, entre los que se

encuentran los proyectos 8 y 10 con una puntuación de 3.4 sobre 5 equivalente al 68% de satisfacción, lo que podría explicarse por eventuales inconvenientes que habrían tenido dichos usuarios al tiempo de comprender el proyecto o al interpretar apropiadamente el alcance de la calificación.

De manera general y en virtud que en el presente estudio la proporción de los proyectos que manifiestan satisfacción inferior al promedio general es menor a los que manifiestan alto grado de satisfacción, es decir, los que superan la media; podría concluirse que el esquema de medición es apropiado y proporciona elementos suficientes para ratificar su validez.

Gráfico 5.1.2. Satisfacción del Cliente por Proyectos



Este indicador mide la satisfacción del cliente respecto del proyecto desarrollado por el proveedor, a través de la metodología que usa para el

efecto, indicadores de evaluación cuantificados en puntuaciones de 1 a 5, de cuyo resultado la gráfica permite inferir los aspectos que se analizan a continuación:

El promedio general consolidado de todos los proyectos participantes alcanza una puntuación de 4.2 sobre 5, esto es, que el nivel de satisfacción del cliente promedio es del 83% respecto a los parámetros evaluados, calificación similar a la que otorgó el usuario promedio.

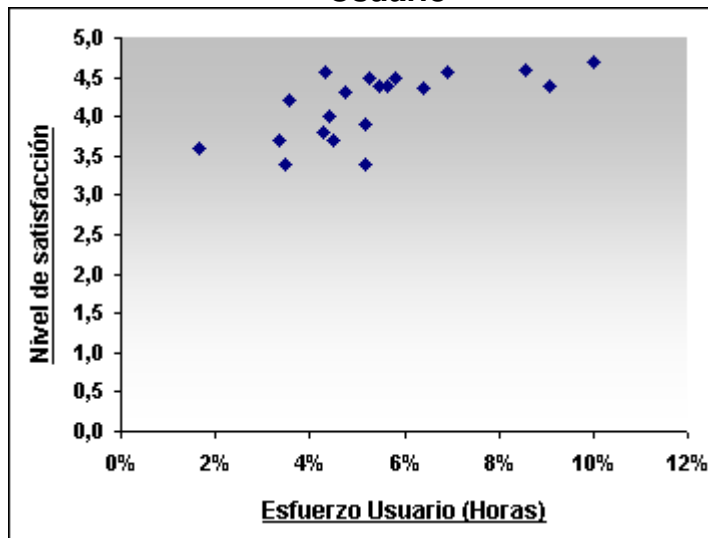
De su lado, al comparar el promedio antes indicado con cada uno de los proyectos en particular, se determina que 11 de ellos se encuentran por arriba del mismo, es decir, que estos encuentran un mayor nivel de satisfacción que el promedio general, entre los cuales se encuentra el caso del proyecto 3 que manifiesta un nivel de satisfacción de 4.8 puntos sobre 5 equivalente al 96% de satisfacción, ligeramente mayor a la mejor calificación otorgada por el usuario, que fue del 94%, razón que al igual que en el caso del usuario explica que el cliente en análisis estaría prácticamente satisfecho con la totalidad de los parámetros medidos.

Por su parte, existen 9 proyectos que manifiestan un nivel de satisfacción del cliente inferior al promedio general de 4.2, entre los que se encuentra el proyecto 8 con una puntuación de 3.4 sobre 5 equivalente al 68% de satisfacción, similar al sesgo presentado en la evaluación de satisfacción

del usuario, lo que igual podría explicarse por eventuales inconvenientes que podría haber tenido dicho cliente al tiempo de comprender el proyecto o al interpretar apropiadamente el alcance de la calificación.

De manera general y en virtud que en el presente estudio la proporción de los proyectos que manifiestan satisfacción inferior al promedio general es menor a los que manifiestan alto grado de satisfacción, es decir, los que superan la media y que existe razonable similitud entre los resultados de evaluación del usuario y del cliente; podría concluirse que el esquema de medición es apropiado y proporciona elementos suficientes para ratificar su validez.

Gráfico 5.1.3. Satisfacción del Usuario versus Esfuerzo del Usuario



Este indicador relaciona el esfuerzo en horas dedicadas por el usuario y la calificación por satisfacción que el otorgó al proyecto.

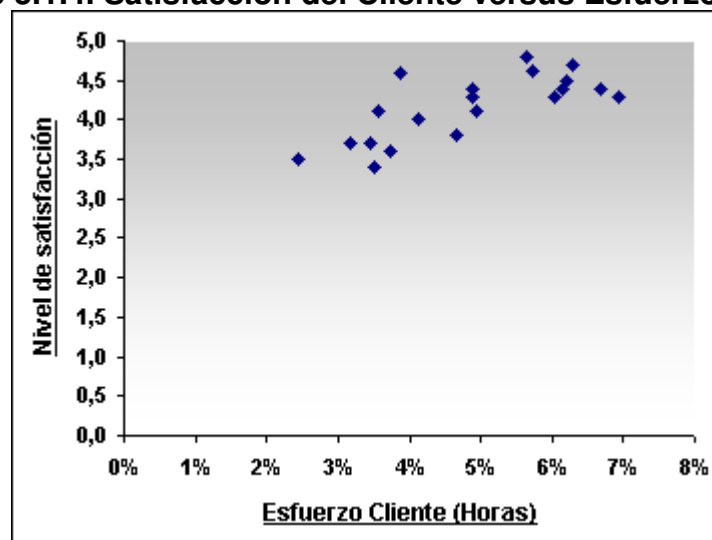
En este sentido, de la gráfica anterior se concluye que los usuarios que han participado con mayor esfuerzo en horas, los mismos que se ubican en el rango del 8% a más del 10% de las horas totales del proyecto, manifiestan la más alta calificación en cuanto a satisfacción, lo que podría explicar que estos tuvieron más elementos de análisis y conocimiento al tiempo de concretar la evaluación.

Por su lado, los usuarios que se ubican en el rango de 4% a 6% de horas de participación en el proyecto, de manera general han aportado calificaciones que están por debajo de la media general, y que

básicamente estarían entre los 8 proyectos que se mencionaron en el análisis del gráfico 5.1.1 satisfacción de usuario por proyecto.

De manera general se ratifica el criterio de que a menor participación en el proyecto, el usuario podría encontrar limitantes para comprender y como consecuencia evaluar inadecuadamente el grado de satisfacción, lo que es concordante con la conclusión contenida en la gráfica de satisfacción del usuario.

Gráfico 5.1.4. Satisfacción del Cliente versus Esfuerzo del Cliente



Este indicador relaciona el esfuerzo en horas dedicado por el cliente y la calificación por satisfacción que ha otorgado al proyecto.

La gráfica anterior deduce que las mayores calificaciones en cuanto a satisfacción del cliente están en el rango de 5% a 6% de las horas totales

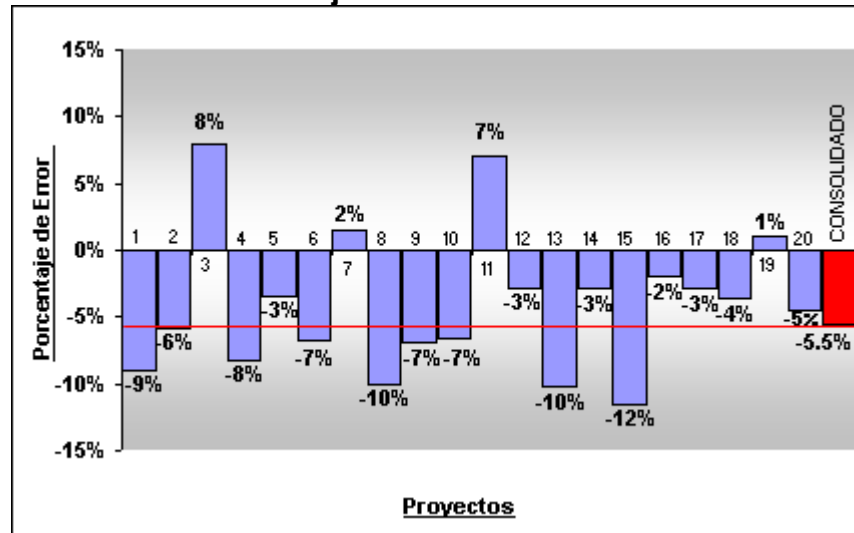
de participación en el proyecto, manifestando además que a mayor participación, más del 6.5%, la calificación dada decrece, lo que podría explicarse por el perfil del evaluador, que al tener criterios y funciones de tipo comercial, prefiere asignar el menor tiempo posible a aspectos de orden operativo.

Por otro lado, los clientes ubicados entre el rango del 2% al 5% del total de horas de participación asignan niveles de calificación menores a la media.

INDICADOR 2

Porcentaje de error en la estimación del esfuerzo

Gráfico 5.2.1. Porcentaje de error en la estimación del esfuerzo



Este indicador determina el grado de acercamiento de los proyectos a sus estimaciones de planificación de tiempo. De cuyo resultado la gráfica permite inferir los aspectos que se analizan a continuación:

El promedio general de todos los proyectos estimó el 5.5% menos del tiempo necesario para desarrollarlos, lo que explica la posición negativa que se evidencia en las barras.

Por su parte, 16 de los proyectos motivo del estudio estimaron menos tiempo del requerido, destacándose un sesgo importante en el proyecto 15 que estimó un 12% menos del tiempo total demandado, esta situación ratifica la tendencia en el sentido de que la gran mayoría de los proyectos

informáticos carecen de una adecuada estimación de tiempo, lo que respondería al escaso esfuerzo en la planificación de los mismos.

Por otro lado, la gráfica evidencia dos proyectos, esto es, el 7 y 19 que acusan apropiado nivel de estimación de tiempo, toda vez que, a pesar de haber estimado más del tiempo requerido dicho exceso únicamente se ubica en el 2% y 1% respectivamente, discrepancias que se consideran razonables en estos casos.

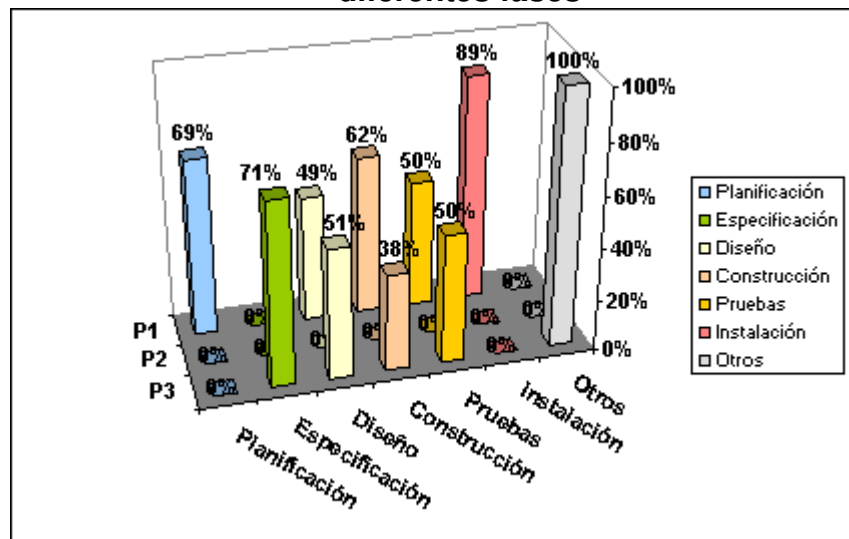
Así también, de la gráfica se concluye que los proyectos 3 y 11 sobrestimaron la cobertura de los mismos, ya que la planificación del tiempo excede en el 8% y 7% respectivamente, aspecto que también respondería a una deficiente planificación.

Como conclusión en este indicador se podría mencionar que todo sesgo importante o toda sobre o subestimación se derivaría de una deficiente planificación, aspecto que se lo analizará mas adelante.

INDICADOR 3

Porcentaje del esfuerzo por persona en las diferentes fases

Gráfico 5.3.1. Porcentaje del esfuerzo por persona en las diferentes fases



Este indicador refleja el esfuerzo realizado por cada persona en las distintas fases del proyecto, el cual es evaluado para cada proyecto en particular; en el presente se ha evaluado el proyecto 5, de cuyo resultado se infiere lo siguiente:

El gráfico demuestra que en este proyecto participan 3 personas, evidenciando que en la fase de planificación la persona 1 aportó con el 69% de esfuerzo en la misma, en consecuencia la diferencia de participación estaría dada por aporte del cliente y/o usuario; en la fase de especificación se observa que la persona 3 aportó el 71% del esfuerzo; en la fase de diseño las personas 1 y 3 aportaron con el 49% y 51%

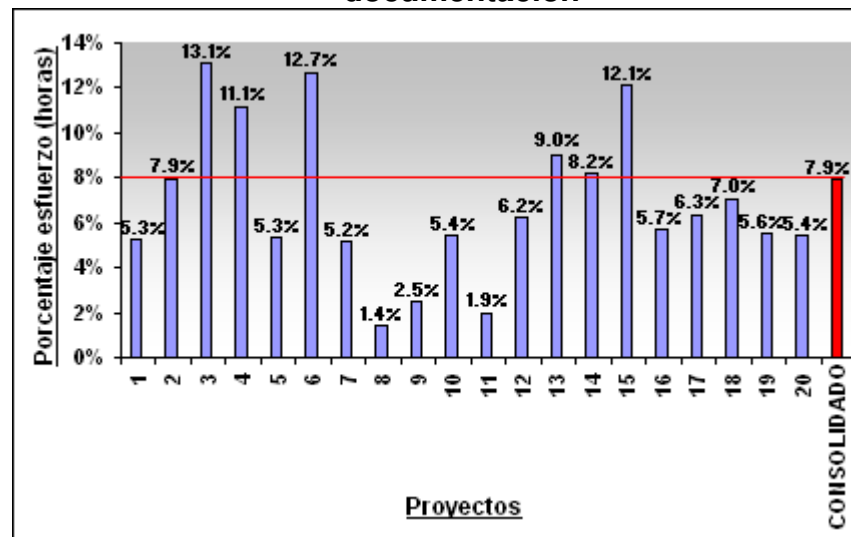
respectivamente, lo que indicaría que las dos personas cubren la totalidad de la fase; en la fase de construcción las personas 1 y 3 aportan con el 62% y 38% respectivamente; en la fase de pruebas se tiene que las personas 1 y 3 aportan cada una con el 50%; en la fase de instalación la persona 1 aporta el 89%; y, en la fase “Otros”, la persona 3 participa en la totalidad de las actividades de la misma.

El promedio simple de los aportes por personas en cada una de las fases determinó que las personas 1 y 3 habrían aportado para la realización del proyecto en su conjunto con el 45% y 44% respectivamente, lo que nos indicaría que ambos han realizado similar esfuerzo para el proyecto, con la particularidad que la persona 2 no aporta con esfuerzo alguno; en consecuencia el 11% restante le correspondería al aporte del cliente y/o usuario.

INDICADOR 4

Porcentaje de esfuerzo empleado en documentación

Gráfico 5.4.1. Porcentaje de esfuerzo empleado en documentación



Este indicador refleja el porcentaje de las horas totales que cada proyecto ha dedicado a la documentación del mismo.

El consolidado de todos los proyectos participantes alcanza el 7.9%, el mismo que en la práctica se considera relativamente bajo, ya que la experiencia de personas conocedoras de estos temas sugieren mayor dedicación en horas al proceso de documentar los proyectos, debido al alto riesgo que su ausencia implica. **[31]**

En el caso de este estudio existen sesgos significativos que superan de manera importante al promedio como los casos de los proyectos 3 con 13.1%, 6 con 12.7%, 15 con 12.1% y 4 con 11.1% los mismos que por

exceder de manera evidente al promedio se consideran datos cuya atipicidad estaría dada porque se tratan de nuevos proyectos o porque el encargado de esta función no posee adecuada experiencia en el manejo de técnicas de documentación, más no porque posean cultura de documentación.

Por otro lado, se considera situaciones atípicas aquellos casos que se ubican muy por debajo de la media, tratándose de la gráfica, los proyectos 9 con 2.5%, 11 con 1.9% y 8 con 1.4% de horas dedicadas al proceso de documentación respecto al total de horas de desarrollo del proyecto.

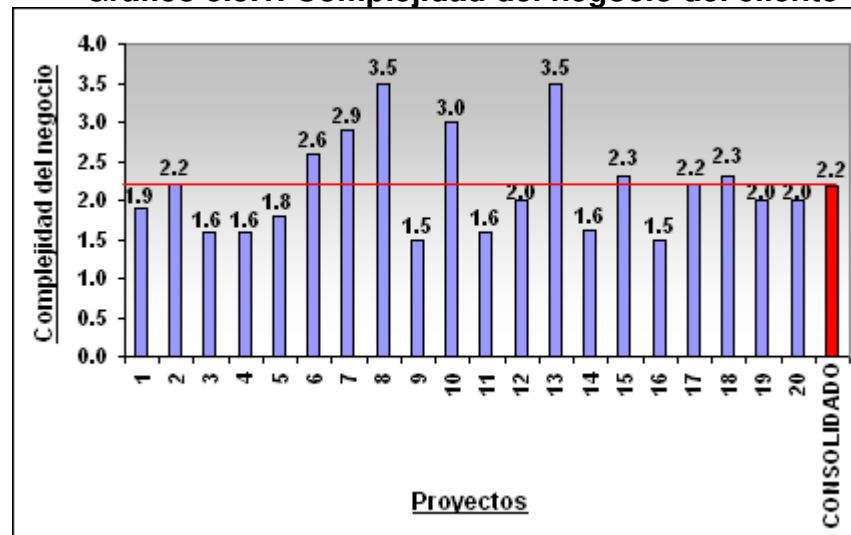
De lo anterior, la atipicidad podría obedecer a que los proyectos participantes se refieran a una mejora de uno existente, lo que ciertamente no requiere mayor documentación dado que ésta ya existe o también porque la cultura de no documentación en dichas empresas es mucho más arraigada.

Como conclusión del presente indicador se establece que similar a lo que se observa en las empresas latinoamericanas no existe cultura de documentación de los proyectos informáticos [32], lo que como se ha mencionado en líneas anteriores, constituye un alto nivel de riesgo operativo.

INDICADOR 5

Complejidad del Negocio y Complejidad Técnica

Gráfico 5.5.1. Complejidad del negocio del cliente



Este indicador mide la complejidad del negocio del cliente evaluada por las empresas, a través de la metodología que usa para el efecto cuantificadores con puntuaciones asignadas de 0 a 4, a los diferentes parámetros que configurarían el grado de complejidad inherente al negocio del cliente, de cuyo resultado la gráfica permite inferir los aspectos que se analizan a continuación:

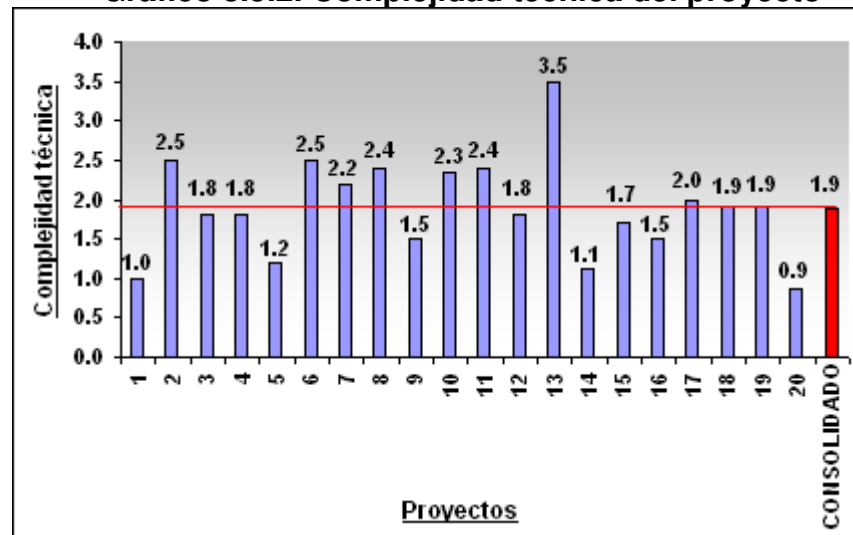
El promedio general consolidado de todos los proyectos participantes alcanza una puntuación de 2.2 sobre 4, esto es, que la complejidad promedio del negocio del cliente en todos los proyectos en su conjunto es del 55% respecto a los parámetros evaluados.

Al comparar la media antes indicada con cada uno de los proyectos en particular se determina que 9 de ellos se encuentran por arriba del mismo, es decir, que éstos infieren un mayor nivel de complejidad del negocio respecto al promedio general, entre los que se encuentran los casos de los proyectos 8 y 13, que evidencian niveles de complejidad de 3.5 puntos sobre 4 equivalente al 88% de complejidad, razón que a su vez explica que los citados clientes acusan alto nivel de complejidad, toda vez que, los parámetros de evaluación en análisis habrían contenido para el proveedor del software aspectos de importante magnitud en cuanto al entendimiento y comprensión de la naturaleza, características y peculiaridades del negocio.

Por su parte, existen 11 proyectos que manifiestan un nivel de complejidad del negocio del cliente, inferior al promedio general de 2.2 puntos, entre los que se encuentran los proyectos 9 y 16 con una puntuación de 1.5 sobre 4 equivalente al 38% de complejidad, lo que podría explicarse por el hecho que tales clientes tendrían reglas definidas, estarían en el mercado objetivo, son de magnitud relativamente pequeña, cuentan con políticas y tienen clientes relativamente informados, de acuerdo a lo aseverado por las empresas.

De manera general y en virtud que en el presente estudio la proporción de los proyectos manifiestan complejidad del negocio del cliente inferior al promedio general, ya que 11 de los 20 proyectos analizados se encuentran en dicha situación se podría concluir que más de la mitad de la muestra infiere baja complejidad y que por lo tanto tendrían un buen nivel de conocimiento y manejo de sus negocios.

Gráfico 5.5.2. Complejidad técnica del proyecto



Este indicador mide la complejidad de técnica del proyecto evaluada por las empresas, a través de la metodología que usa para el efecto cuantificadores con puntuaciones asignadas de 0 a 4, a los diferentes parámetros que configurarían el grado de complejidad inherente a la seguridad, transacciones, condiciones de operación y experticia del personal, esto es, el entorno técnico del proyecto en si, de cuyo resultado la gráfica permite inferir los aspectos que se analizan a continuación:

El promedio general consolidado de todos los proyectos participantes alcanza una puntuación de 1.90 sobre 4, esto es, que la complejidad técnica promedio de todos los proyectos en su conjunto es del 48% respecto a los parámetros evaluados.

Al comparar la media antes indicada con cada uno de los proyectos en particular se determina que 8 de ellos se encuentran por arriba del mismo, es decir, que estos deducen un mayor nivel de complejidad técnica en los términos antes anotados respecto al promedio general, entre los que se encuentra el caso del proyecto 13 que evidencia nivel de complejidad de 3.5 puntos sobre 4 equivalente al 88% de complejidad, sesgo que explica el alto nivel de complejidad del mismo, ya que este habría implicado exigentes medidas de seguridad, alto volumen de transacciones, elevada cobertura de operación, sumado a la necesidad urgente del cliente por el software, de acuerdo a lo aseverado por las empresas.

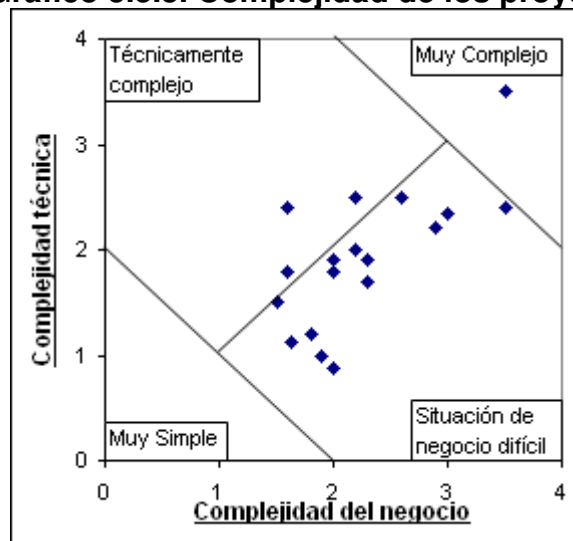
Existen 10 proyectos que manifiestan un nivel de complejidad técnica, inferior al promedio general de 1.90 puntos, entre los que se encuentra el proyecto 20 con una puntuación de 0.90 sobre 4 equivalente al 22% de complejidad, lo que podría explicarse por el hecho de que contaba con personal experto, no demandaba urgencia en su desarrollo, baja

seguridad, entre otras características que definen la baja complejidad del mismo.

Precisa indicar que, los proyectos 18 y 19 acusan niveles de complejidad iguales al promedio general, esto es, de 1.9 puntos sobre 4 equivalentes al 48%.

De manera general en el presente estudio la proporción de los proyectos manifiestan complejidad técnica inferior al promedio general, ya que 10 de los 20 proyectos analizados se encuentran en dicha situación y 2 son iguales a la media, por lo que podría concluirse que más de la mitad de la muestra deduce baja complejidad técnica y que por lo tanto se caracterizaron por una aceptable articulación entre la organización y utilización de recursos tecnológicos.

Gráfico 5.5.3. Complejidad de los proyectos



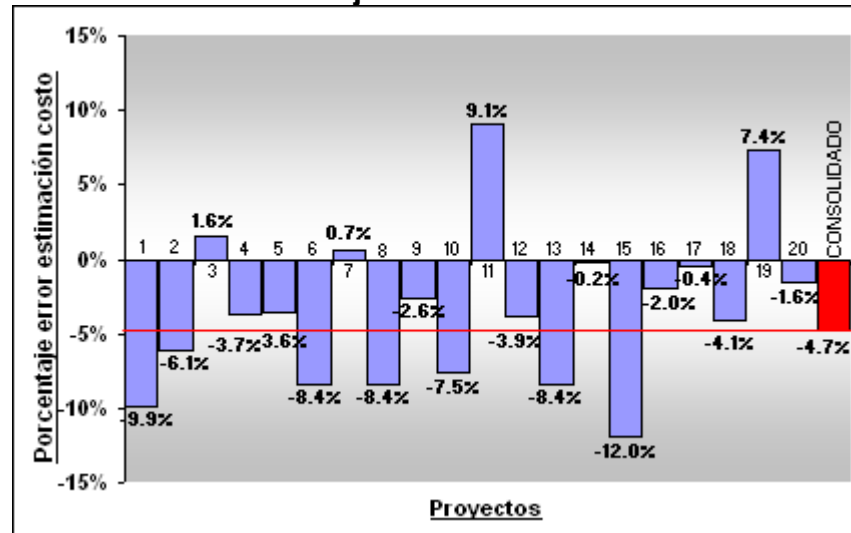
Este indicador relaciona la complejidad del negocio con la complejidad técnica, apuntando a una separación de los proyectos participantes en 4 cuadrantes, de tal manera que se evidencia la virtual segregación existente entre proyectos simples, muy complejos, técnicamente complejos y situaciones de negocio difíciles.

De la gráfica se puede inferir que la mayor parte de proyectos han sido realizados para negocios complejos, lo que ratifica los resultados evidenciados de la gráfica anterior; al tiempo que se aprecia pocos proyectos que han derivado alta complejidad y más aún encontramos que en rango de proyectos muy complejos existiría un solo caso; a su vez, la gráfica demuestra de manera clara que no han existido proyectos que puedan catalogarse en la calidad de muy simples, aspecto que efectivamente se ratifica en las gráficas anteriores.

INDICADOR 6

Porcentaje de error en la estimación del costo

Gráfico 5.6.1. Porcentaje de error en la estimación del costo



Este indicador determina el grado de acercamiento de los proyectos a sus estimaciones de costo presupuestadas.

El promedio general de todos los proyectos estimó el 4.7% menos del costo necesario para desarrollarlos, lo que explica la posición negativa que se evidencia en las barras. Dieciséis de los proyectos motivo del estudio estimaron menos del costo total requerido, destacándose un sesgo importante en el proyecto 15 que estimó un 12% menos del costo total demandado, esta situación ratifica la tendencia en el sentido que la gran mayoría de los proyectos informáticos, objeto del estudio, carecen de una adecuada estimación en el costo lo que respondería al escaso esfuerzo en la planificación los mismos.

Por otro lado, la gráfica evidencia, que 3 proyectos, esto es, el 7, 14 y 17 presentan un apropiado nivel de estimación de costo, ya que a pesar de haber estimado más o menos el costo requerido, dicho exceso o faltante según el caso, únicamente se discrepa en el 0.7%, -0.2% y -0.4% respectivamente, lo que se considera razonable en estos casos.

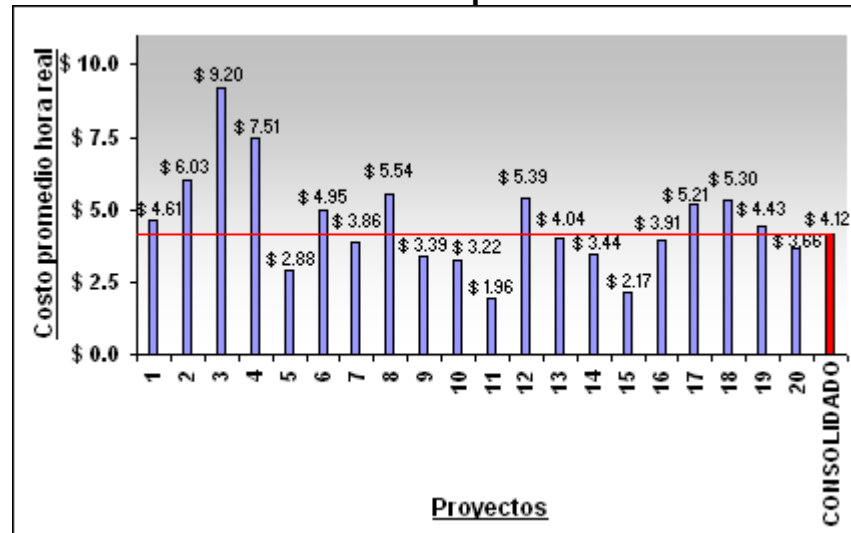
Así también, de la gráfica se deduce que los proyectos 11 y 19 sobrestimaron la cobertura del mismo, ya que la presupuestación del costo excede en el 9.1% y 7.4% respectivamente, aspecto que también respondería a una deficiente planificación.

Como conclusión en este indicador se podría mencionar que todo sesgo importante o toda sobre o subestimación se deriva de una deficiente planificación.

INDICADOR 7

Costo promedio hora real.

Gráfico 5.7.1. Costo promedio hora real



Este indicador refleja el costo promedio hora real de los proyectos participantes.

El consolidado de todos los proyectos alcanza un costo promedio de \$4.12, que si se lo examina con los demás participantes, observamos que 10 proyectos están por arriba y 10 por debajo de dicho promedio, lo que nos llevaría a concluir la razonabilidad de la tendencia.

En el caso de este estudio existe un sesgo significativo que supera de manera importante al promedio, como el caso del proyecto 3 con \$9.20, promedio hora real, el mismo que por exceder de manera evidente a la media se considera atípico, en razón de que en el mismo habrían

participado en mayor proporción recursos de más alto costo en la empresa, como el caso de gerentes o líderes de proyectos o también podría haber la presencia de costos indirectos adicionales que inciden en el promedio hora real, como el caso de personal indirecto que habría participado en los proyectos, tales como secretarias, personal de apoyo y hasta eventuales honorarios a consultores.

Por otro lado, también se consideran situaciones atípicas aquellos casos que se ubican muy por debajo de la media. El proyecto 11 con un costo hora promedio es \$1.96, cuya atipicidad podría obedecer a que en el proyecto han participado recursos de bajo costo, los que no necesariamente podrían catalogarse como malos recursos, pero que la realidad del mercado salarial ha evidenciado la existencia de buenos desarrolladores a un costo hora bajo particularmente en empresas pequeñas.

Un artículo publicado por la CORPEI indica la encuesta salarial realizada por Price Waterhouse SIREM "Servicio Integrado de Remuneraciones" a Empresas Nacionales y Multinacionales del Ecuador [33]. En dicho artículo se tiene lo siguiente:

Costo hora promedio:

Ingeniero / Analista de Sistemas Senior = \$6.18 (calculado de 160 horas).

Ingeniero / Analista de Sistemas Junior = \$3.80 (calculado de 160 horas).

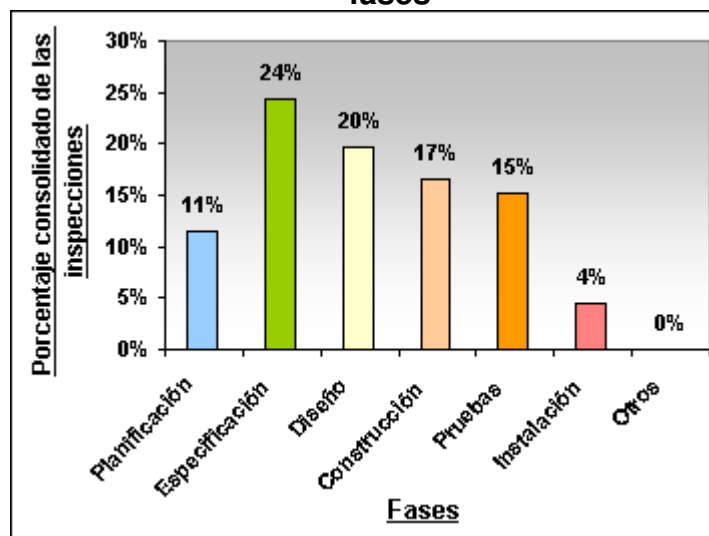
Como conclusión del presente indicador se establece que el costo promedio hora varía según la categoría de recursos participantes en los proyectos, de manera que en proyectos en los que intervienen categorías jerárquicas altas derivarán mayor costo hora promedio y contrariamente proyectos en los que intervienen categorías jerárquicas de menor nivel tendrán bajo costo promedio hora. En el presente estudio se evidencia un relativo equilibrio en este sentido.

Respecto al costo hora promedio obtenido por el presente estudio se observa que guarda relación con el publicado por la CORPEI, aspecto que confirmaría la validez del mismo.

INDICADOR 8

Porcentaje de inspecciones realizadas por fases y Porcentaje de inspecciones realizadas en el proyecto

Gráfico 5.8.1. Porcentaje consolidado de las inspecciones en las fases



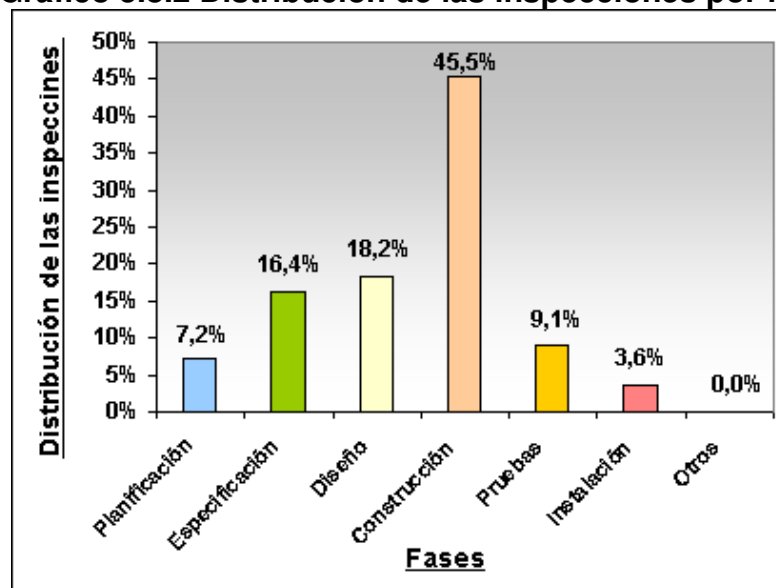
Este indicador determina a nivel consolidado la proporción que representan las inspecciones realizadas en cada fase, respecto al total de tareas realizadas en las mismas, de cuyo resultado la gráfica permite inferir los aspectos que se analizan a continuación:

La mayor proporción de inspecciones a nivel consolidado se concentran en la fase de especificación, ya que del total de las tareas ejecutadas en esta, el 24% correspondieron a inspecciones; seguido la fase de diseño con el 20%; luego le sigue la fase de construcción con el 17%; el 15% en la fase de pruebas; en la fase de planificación el 11%; y finalmente, en la

fase de instalación el 4% de las tareas orientadas a inspecciones; además se establece que la fase “Otros” no existieron inspecciones.

La situación anterior permite concluir que la mayor cantidad de inspecciones se realizan a partir de la segunda fase, esto es, en la especificación, donde se observa que las fases siguientes tuvieron niveles de inspección porcentualmente decrecientes.

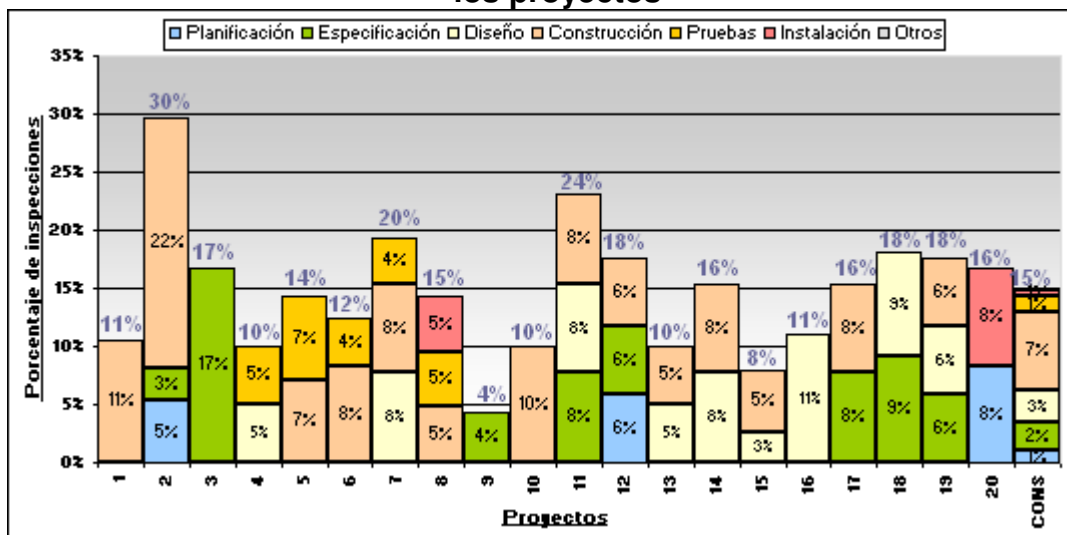
Gráfico 5.8.2 Distribución de las inspecciones por fases



Este indicador obtenido de estudios anteriores [4], muestra la forma como están distribuidas proporcionalmente las actividades de inspecciones, de manera que, la sumatoria de todos los porcentajes indicados suman el 100% de las inspecciones ejecutadas. A partir de la gráfica podemos concluir lo siguiente:

La mayor proporción de las inspecciones a nivel consolidado se concentran en la fase de construcción, ya que del 100% el 45.5% están en dicha fase, aspecto que es absolutamente explicable en razón que la mayor cantidad de tareas en un proyecto se ejecutan en esta fase. El resto de fases tienen porcentajes menores distribuidos de la siguiente manera: diseño 18.2%, especificación 16.4%, pruebas 9.1%, planificación 7.2% e instalación 3.6%.

Gráfico 5.8.3. Porcentaje de inspecciones realizadas por fase en los proyectos



Este indicador demuestra a nivel de cada proyecto y consolidado la proporción de inspecciones en cada fase.

El consolidado de todos los proyectos participantes demuestra que el 15% de las tareas representaron inspecciones, el mismo que a su vez se descompone en 7% de inspecciones en la construcción, 3% en el diseño,

2% en la especificación, 1% en la planificación, 1% en las pruebas y 1% en la instalación.

El consolidado comparativo determina que 10 proyectos están por arriba, 9 por debajo y 1 igual al promedio.

A su vez se evidencia un sesgo significativo que supera de manera importante al promedio, como el caso del proyecto 2 que demuestra que el 30% de las tareas fueron inspecciones, de las cuales el 22% estuvieron en la fase de construcción, el 5% en planificación y el 3% en especificación, seguido del proyecto 11 en el que el 24% de las tareas están representadas por inspecciones, las mismas que se distribuyen en 8% en la especificación , 8% en diseño y 8% en construcción.

Por otro lado, también se consideran situaciones atípicas aquellos casos que se ubican muy por debajo de la media, por ejemplo el proyecto 9 demuestra que sólo el 4% de las tareas correspondieron a inspecciones, todas en la fase de especificación.

Como conclusión general podemos inferir que el 50% de los proyectos superaron el promedio consolidado de inspecciones. Sólo 3 proyectos, el 2, 12 y 20 iniciaron sus inspecciones en la fase de planificación; 6

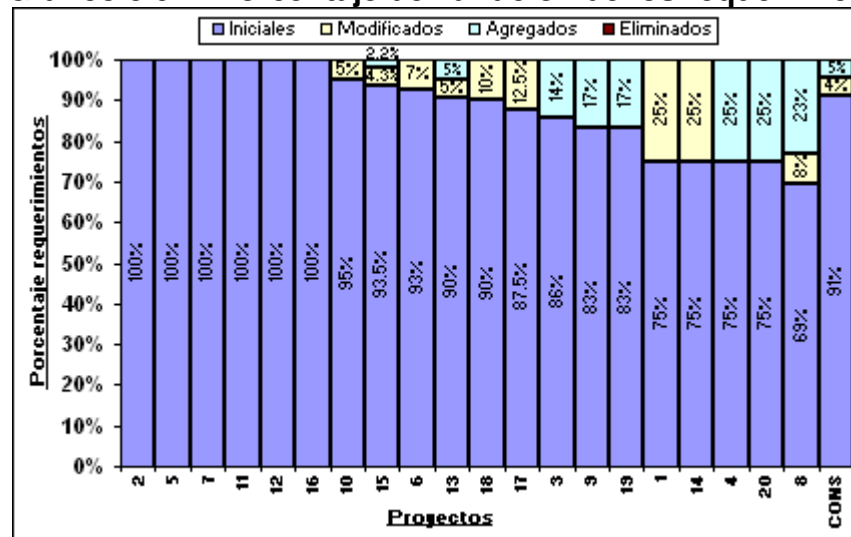
proyectos iniciaron las inspecciones en la especificación; 6 proyectos, en el diseño y finalmente 5 proyectos en la fase de construcción.

INDICADOR 9

Porcentaje de variación de los requerimientos.

Previo a la explicación de los resultados obtenidos del presente indicador, es necesario explicar lo que las empresas consideran como requerimiento, el cual es definido como una funcionalidad que actualmente no posee la aplicación y que se la puede agregar de forma o de fondo.

Gráfico 5.9.1. Porcentaje de variación de los requerimientos



Este indicador mide la variación en porcentaje de los requerimientos originalmente propuestos respecto a los ejecutados en el proyecto. En la gráfica podemos notar que el 91% de los requerimientos iniciales se

mantuvieron, el 4% fueron modificados y el 5% fueron nuevos requerimientos.

Al comparar el promedio antes indicado con cada uno de los proyectos, determinamos que 9 están por arriba del promedio, debiéndose probablemente a un buen levantamiento de requerimientos.

La mayor variación de cambio de requerimientos originales (nuevos o modificados) se puede deber a la falta de experiencia o conocimiento del entorno del cliente y/o usuario, o a la deficiente recolección de dichos requerimientos.

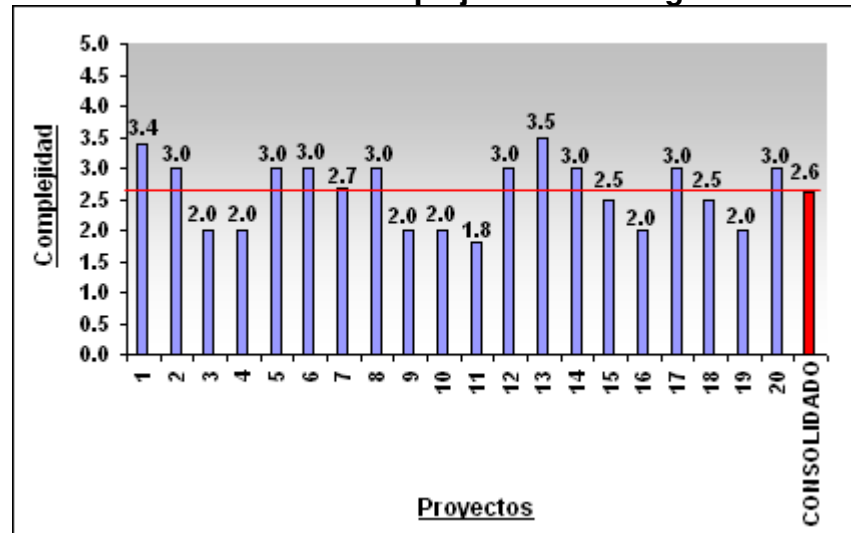
En los proyectos analizados no se presentaron casos de eliminación de requerimientos.

En conclusión, la información proporcionada en el presente indicador se considera de alta utilidad práctica para futuros proyectos a fin de que los encargados de ejecutarlos analicen y planifiquen apropiadamente el levantamiento de información y la coordinación con los clientes y/o usuarios.

INDICADOR 10

Complejidad del código fuente.

Gráfico 5.10.1. Complejidad del código fuente



Este indicador mide la complejidad promedio del código fuente en una escala de 0 a 5. Esa asignación de complejidad fue realizada por el líder del proyecto en conjunto con su equipo de desarrolladores.

El promedio general consolidado de todos los proyectos participantes alcanza una complejidad de 2.6 sobre 5.

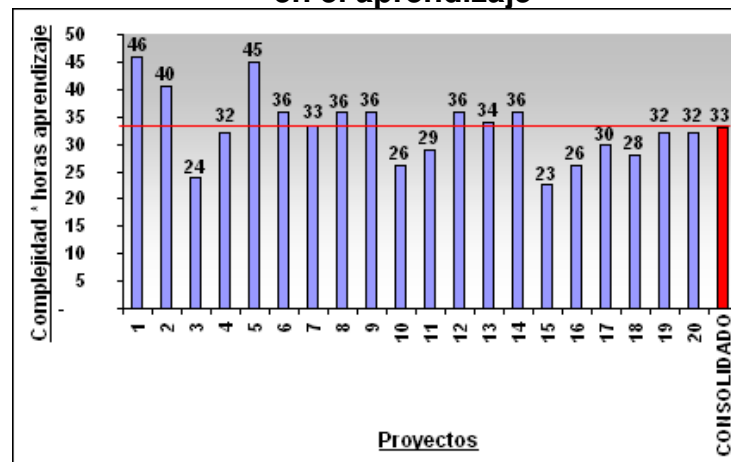
Once proyectos se encuentran por arriba del promedio. Los proyectos 1 y 13 evidencian niveles de complejidad del código fuente de 3.4 y 3.5 puntos sobre 5 equivalentes al 68% y 70% de complejidad respectivamente. Un alto nivel de complejidad, de acuerdo a sus evaluadores, demanda un mayor esfuerzo para comprender y desarrollar la aplicación.

De manera general, en el presente estudio la mayor proporción de los proyectos manifiestan complejidad del código fuente, superior al promedio general, ya que 11 de los 20 proyectos analizados se encuentran en dicha situación y 2 son casi iguales a la media, por lo que podría concluirse que más de la mitad de la muestra infiere una alta complejidad del código fuente y que por lo tanto pudieron haber tenido mayor esfuerzo en el desarrollo, así como un mayor esfuerzo para aprender lo que iban a realizar.

INDICADOR 11

Indicador ponderado de la complejidad y demora en el aprendizaje.

Gráfico 5.11.1. Indicador ponderado de la complejidad y demora en el aprendizaje



El indicador promedio de complejidad consolidado, alcanza un valor ponderado de 33, el mismo que al ser comparado con cada uno de los

proyectos en particular determina que 9 de ellos se encuentran por arriba del promedio, es decir que estos tienen un mayor nivel de complejidad del código fuente y como consecuencia más demora en el aprendizaje. Los proyectos 1 y 5 presentaron mayor complejidad, de 46 y 45 respectivamente, puesto que fueron de mayor magnitud que el resto de los casos de estudio, a pesar de estar dentro del perfil elegido.

Existen 10 proyectos que manifiestan un indicador ponderado de complejidad y demora en el aprendizaje inferior al promedio general consolidado de 33, entre los que se encuentran los proyectos 15, 3 y 10 con una valoración de 23, 24 y 26 respectivamente. Tales proyectos presentaron menor nivel de complejidad y rápido aprendizaje. De manera general en el presente estudio se evidencia una equilibrada proporción entre los proyectos que manifiestan mayor complejidad ponderada y demora en el aprendizaje y aquellos que revelan menor nivel de complejidad respecto al promedio general consolidado, con la virtuosidad que el alejamiento de cada uno de los proyectos respecto a la media no es importante, lo que ciertamente ratifica el hecho que la elección de la muestra se caracteriza por ser proyectos de perfiles similares, aspecto que da mayor credibilidad a los resultados obtenidos y tendrían alta utilidad práctica para empresas de similares tamaños.

INDICADOR 13

Ocurrencia de defectos y fallas por fases.

Con el concepto de defectos y fallas ya explicado en el capítulo 3, sección 3.7, procedemos a mostrar los resultados obtenidos:

Gráfico 5.13.1. Porcentaje consolidado de defectos respecto al número total de tareas realizadas

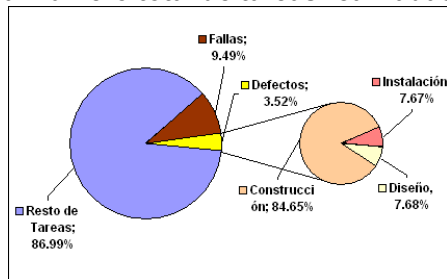
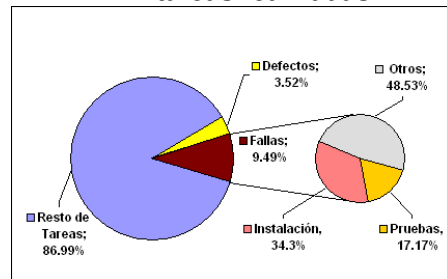


Gráfico 5.13.2. Porcentaje consolidado de fallas respecto al número total de tareas realizadas



Este indicador mide la proporción que representan la ocurrencia de defectos y/o las fallas con respecto a la totalidad de las tareas a nivel consolidado.

El primer gráfico muestra que el 3.52% de las tareas estuvieron orientadas a atender los defectos presentados, de las cuales el 84.65% se concentran en la fase de construcción, el 7.68% en la fase de instalación y el 7.67% en diseño.

Con relación a la ocurrencia de fallas, el segundo gráfico determina que el 9.49% de las tareas se orientaron a atender las fallas presentadas. El 48.53% se concentran en la fase "Otros", el 34.3% en la fase de instalación y el 17.17% en la fase de pruebas.

Gráfico 5.13.3 Porcentaje de defectos por fase y por proyecto

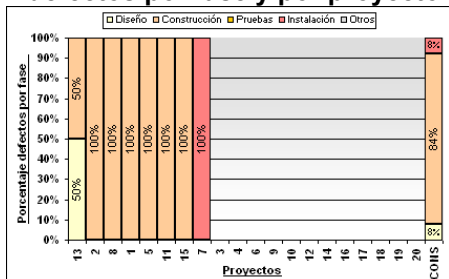
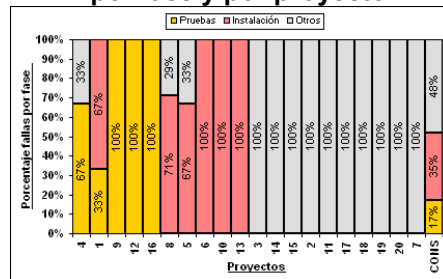


Gráfico 5.13.4 Porcentaje de fallas por fase y por proyecto



Este indicador mide el porcentaje de defectos y fallas por fases y por proyecto.

A nivel consolidado la mayor proporción de los defectos se presentan en la fase de construcción (84%), seguido por la fase de diseño e instalación. En doce proyectos no presentaron defectos documentados, mientras que 6 proyectos reportaron que el 100% de sus defectos se presentaron en la fase de construcción. EL proyecto 7 presentó la totalidad de los defectos en la fase de instalación.

La segunda gráfica muestra que el 48% de las fallas se presentan en la fase “Otros”, seguidos por las fases de instalación con un 34%, y pruebas de aceptación con un 17%.

INDICADOR 14

Tipo de defectos y fallas más ocurrentes, identificados por severidad.

Gráfico 5.14.1. Porcentaje consolidado de defectos respecto a las tareas, identificados por su severidad

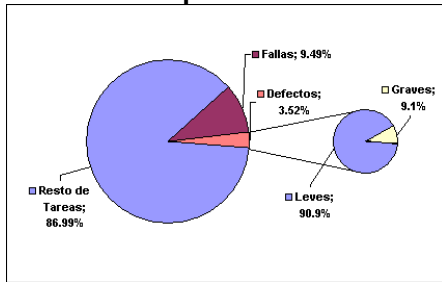
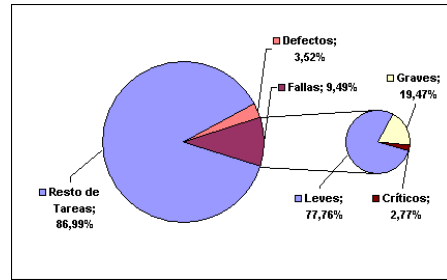


Gráfico 5.14.2. Porcentaje consolidado de fallas respecto a las tareas, identificadas por su severidad



Este indicador mide la proporción que representan los tipos de defectos y/o fallas con respecto a la totalidad de las tareas, identificados por su severidad.

En el primer gráfico presenta la proporción de los defectos, que con respecto al total de tareas alcanza el 3.52%, se establece que por su severidad, el 90.9% fueron de carácter leve, el 9.1% de carácter grave y no se evidencian defectos críticos.

En el segundo gráfico, que como también se analizó en el indicador anterior corresponde a la proporción de las fallas, que con respecto al total de tareas alcanzan el 9.49%, se establece que por su severidad, el 77.76% fueron de carácter leve, el 19.47% de carácter grave y el 2.77% de carácter crítico.

Como conclusión del presente indicador se puede concluir que la mayor proporción de los defectos y fallas presentados en el estudio, catalogados por su severidad se ubican en la categoría de leves, aspecto que no evidencia mayor riesgo, ya que la demanda de recursos para su regularización es relativamente baja.

Gráfico 5.14.3. Porcentaje de defectos más ocurrentes

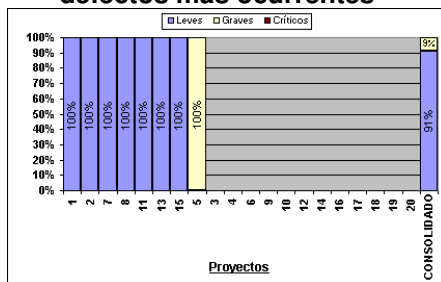
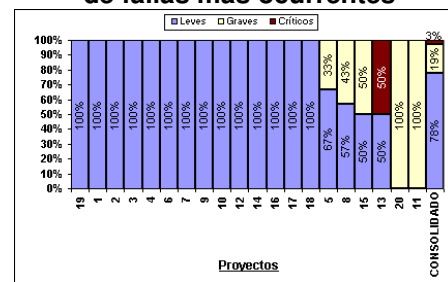


Gráfico 5.14.4. Porcentaje de fallas más ocurrentes



Este indicador mide la proporción que representan los tipos de defectos y/o fallas con respecto a la totalidad de las tareas por proyecto y el análisis comparativo con la situación consolidada, identificados por su severidad; de cuyos gráficos independientes para cada aspecto se infiere lo siguiente:

Con relación al porcentaje de defectos, identificados por su severidad, por proyecto y su comparativo con la posición consolidada, el primer gráfico determina que la mayor proporción de defectos, son de carácter leve equivalentes al 91% respecto al total de defectos, seguido de defectos

graves con el 9%, sin evidenciarse defectos críticos. Se anota además que 12 proyectos no evidencian defectos; que 7 proyectos presentan el 100% de defectos leves; y, que 1 proyecto, esto es, el proyecto 5, presenta el 100% de defectos graves, sin embargo, dicho sesgo se explica ya que se trata de un proyecto pequeño, que tuvo la presencia de un solo defecto grave, pero que en ausencia de otro tipo de defectos, derivó dicho resultado.

Con relación al porcentaje de fallas, identificadas por su severidad, por proyecto y su comparativo con la posición consolidada, el segundo gráfico determina que las fallas de carácter leve equivalen al 78% respecto al total de fallas, seguido de fallas graves con el 19% y fallas críticas con el 3%. Se anota además que 14 proyectos evidencian la totalidad de fallas leves; que 2 proyectos presentaron el 100% de fallas graves; que el proyecto 5 presentó fallas leves por un equivalente al 67% y graves por un 33%; que el proyecto 8 presentó fallas leves del 57% y graves del 43%; que el proyecto 15 presentó fallas leves del 50% y graves del 50%; y, que el proyecto 13 presentó fallas leves del 50% y críticas del 50%.

INDICADOR 15

Porcentaje de tiempo utilizado en corrección de defectos y fallas.

Gráfico 5.15.1. Porcentaje de tiempo utilizado en corrección de defectos

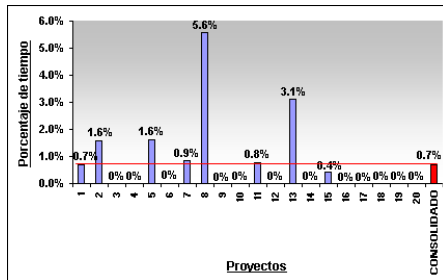
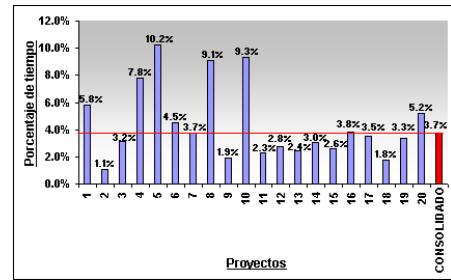


Gráfico 5.15.2. Porcentaje de tiempo utilizado en corrección de fallas



Este indicador revela el tiempo dedicado en cada proyecto en la corrección de defectos o fallas, expresado en porcentaje respecto a la totalidad del tiempo empleado en el desarrollo de tal proyecto.

El primer gráfico muestra que el tiempo promedio utilizado para la corrección de defectos es del 0.7%, el mismo que en la práctica se considera mínimo, ya que la experiencia de personas conocedoras de estos temas opinan que la dedicación de tiempo a la corrección de defectos sería mucho mayor [36]. Los resultados que presentan en este estudio estarían dados por la mala práctica de los miembros de los equipos de desarrollo de no registrar los defectos. Los defectos aquí considerados fueron los reportados, en su mayoría, por el líder de proyectos o el cliente/usuario.

No obstante existe un sesgo importante que supera significativamente al promedio indicado, esto es el caso del proyecto 8 que evidencia el 5.6% del tiempo total utilizado para la corrección de defectos.

Por otro lado, se consideran situaciones atípicas aquellos casos que se ubican muy por debajo de la media, como son los proyectos 3, 4, 6, 9, 10, 12, 16, 17, 18, 19 y 20 que no registraron defecto alguno, seguido del proyecto 15 que únicamente revela el 0.4% del tiempo utilizado para la corrección de defectos, comentario que en todo caso es concordante con lo que se ha explicado anteriormente.

El segundo gráfico presenta el porcentaje de tiempo utilizado en la corrección de fallas, que para este conjunto de proyectos analizados alcanza el 3.7%.

No obstante existen 4 sesgos importantes que superan significativamente al promedio indicado, como son los proyectos 5, 10, 8 y 4 que evidencian el 10.2%, 9.3%, 9.1% y el 7.8% del tiempo total utilizado para la corrección de fallas, aspecto que se considera económica y técnicamente inconveniente, ya que deja en evidencia el alto esfuerzo que estaría orientando la empresa a la corrección de fallas.

Por otro lado, se considera situaciones atípicas aquellos casos que se ubican muy por debajo de la media, como son los proyectos 2, 18 y 9 que registraron 1.1%, 1.8% y 1.9% respectivamente del tiempo total utilizado.

INDICADOR 16

Eficiencia en atención de defectos y fallas.

Este indicador revela la eficiencia de la empresa proveedora de software en atender los defectos y/o fallas. La eficiencia se asocia al número de días que se cuentan desde la fecha de notificación del defecto y/o falla hasta la fecha de atención al mismo. A este indicador se lo suele denominar tiempo de respuesta o tiempo promedio de ciclo [37].

Gráfico 5.16.1. Eficiencia en atención de defectos

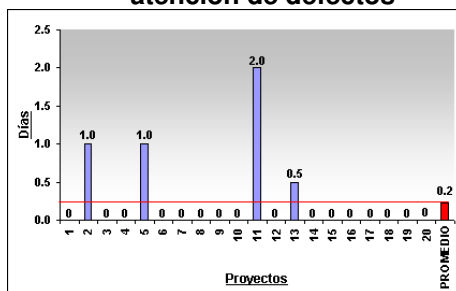
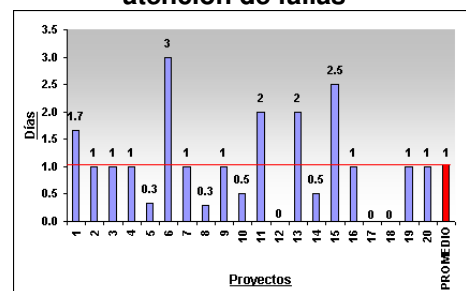


Gráfico 5.16.2. Eficiencia en atención de fallas



En la primera gráfica, el consolidado de todos los proyectos evidencia un tiempo de respuesta promedio general equivalente a 0.2 días, esto es, aproximadamente 5 horas que habría demandado la respuesta para la atención de los defectos denunciados, él mismo que comparado con los

demás se observa que 4 proyectos están por arriba y 16 por debajo de dicho promedio.

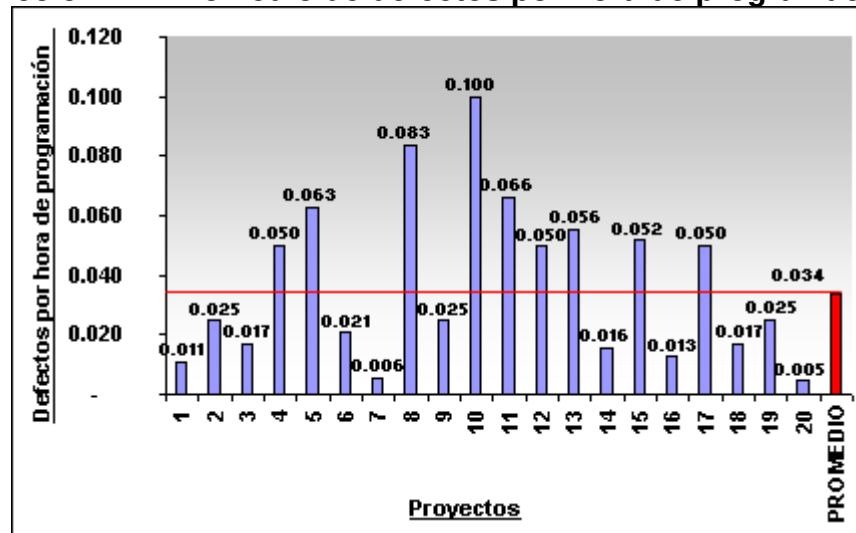
Se consideran situaciones atípicas aquellos casos que se ubican muy por debajo de la media. Un total de 16 proyectos no presentan indicador de tiempo de respuesta en la atención de los defectos, ya que estos como se ha mencionado en el análisis de otros indicadores, tienen como práctica no calificar los defectos sino cuando son reportados por terceros.

En la segunda gráfica, el consolidado de todos los proyectos evidencia un tiempo de respuesta promedio general equivalente a 1 día. Cinco proyectos están por arriba y 7 por debajo de dicho promedio, recalcando que 8 proyectos son iguales al promedio, lo que excepto por el sesgo evidenciado en el proyecto 6, nos lleva a concluir la razonabilidad de la tendencia.

INDICADOR 17

Promedio de defectos por horas de programación.

Gráfico 5.17.1. Promedio de defectos por hora de programación



Este indicador mide el promedio de defectos por horas de programación del proyecto. El promedio general consolidado de todos los proyectos alcanza una puntuación de 0.034, esto es, que por cada hora de programación se encuentran 0.034 defectos.

Nueve proyectos se encuentran por arriba del promedio, es decir que éstos encontraron más defectos por hora de programación respecto al promedio general. Existen 11 proyectos que manifiestan un promedio de defectos por hora de programación inferior al promedio general de 0.034.

Los datos obtenidos para este indicador son considerablemente bajos, debido fundamentalmente a la ausencia de documentación de los defectos, cumpliéndose la aseveración de algunas empresas que dicen que “para los desarrolladores (programadores) su software es casi perfecto, que nunca tiene defectos, sino hasta que el líder del proyecto o el cliente/usuario los encuentra en la fase de pruebas”, opinión que no es compartida por los autores de este estudio, en virtud de que los defectos cometidos en las fases anteriores no fueron documentados.

5.2. Prueba para verificar Hipótesis

En el presente apartado procedemos a verificar las hipótesis, las mismas que fueron detalladas en el Capítulo 3 Sección 3.1, para el efecto se ha utilizado el fundamento teórico detallado en el Capítulo 2 Sección 2.9.

Prueba de la hipótesis 1

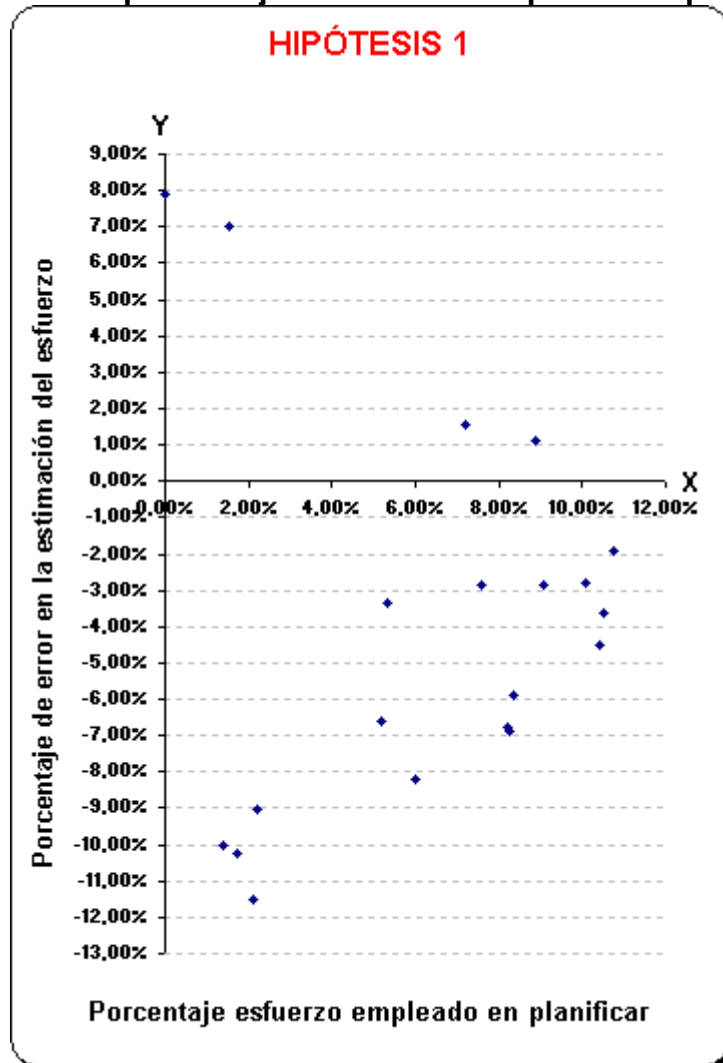
La primera hipótesis a demostrar en el presente estudio responde al planteamiento hipotético referido en el Capítulo 3, que textualmente dice:

“Cuanto mejor es el tiempo empleado en la planificación, menor es el error en la estimación del tiempo”

Para verificar la validez de la presente hipótesis debemos determinar la existencia de la relación entre el porcentaje de esfuerzo empleado en planificar y el error de la estimación del esfuerzo, como elementos fundamentales para concebir la cobertura de la planificación y su efecto en la administración del tiempo. Para tal efecto, realizamos un análisis de correlación, que consiste en demostrar la posible existencia de una recta que debería pasar por la mayoría de los puntos que constituyen la convergencia entre el esfuerzo empleado en planificar y el error en la estimación del esfuerzo, cuyo procedimiento estadístico se refiere en párrafos posteriores, con base a la gráfica 5.18

En la gráfica 5.18 se observan dos puntos apartados del resto, los mismos que fueron considerados como datos atípicos, procediendo a separarlos de la muestra a fin de no alterar el análisis de correlación, al ser considerados como aberrantes.

Gráfico 5.18. Porcentaje de error en estimación del esfuerzo versus porcentaje de esfuerzo empleado en planificar



La variable “x” corresponde al “porcentaje de esfuerzo empleado en planificar” y la variable “y” al “porcentaje de error en la estimación del esfuerzo”, la tabla 5.1 contiene la información obtenida del estudio.

Tabla 5.1. Esfuerzo empleado en planificación versus error en la estimación del esfuerzo

PROYECTO	% TOTAL ESFUERZO EMPLEADO PLANIFICAR (x)	% ERROR EN LA ESTIMACIÓN DEL ESFUERZO (y)
1	2,21%	-9,05%
2	8,37%	-5,88%
3	6,01%	7,92%
4	5,31%	-3,38%
5	8,21%	-6,77%
6	7,18%	1,56%
7	1,40%	-10,00%
8	8,25%	-6,86%
9	5,18%	-6,63%
10	7,59%	-2,84%
11	1,72%	-10,27%
12	10,06%	-2,82%
13	2,13%	-11,53%
14	10,75%	-1,90%
15	9,09%	-2,88%
16	10,53%	-3,64%
17	8,89%	1,10%
18	10,41%	-4,54%

Para demostrar la existencia de alguna recta obtendremos el coeficiente de correlación lineal “r”, cuyo cálculo es el siguiente:

Fórmula:

$$r = \frac{S_{xy}}{S_x S_y}$$

Donde:

r = Coeficiente de correlación

S_{xy} = Covarianza de x y

S_x = Desviación estándar de x

S_y = Desviación estándar de y

Proceso de cálculo:

$S_{xy} = 0,000817235$

$S_x = 0,031928307$

$S_y = 0,037518993$

$$r = \frac{0,000817235}{(0,031928307)(0,037518993)}$$

$r = 0.68$ **“COEFICIENTE DE CORRELACIÓN”**

Sobre este aspecto la teoría dice que: “el valor de coeficiente de relación (r) puede ir de -1.00 a +1.00”, por lo tanto el valor obtenido en el presente estudio de **r= 0.68** demostraría que las variables porcentaje de esfuerzo empleado en planificar y el porcentaje de error en la estimación del esfuerzo se encontrarían coherentemente correlacionadas.

En nuestro caso, por ser una muestra pequeña ($n < 50$), $n=18$ ($20-2$), es necesario probar que el coeficiente de correlación (r) encontrado es representativo de la industria, para lo cual se procede a realizar la prueba de significación utilizando la distribución “**t de Student**” para muestras pequeñas.

En este procedimiento se proponen dos hipótesis alternativas:

$H_0 : \rho = 0$ (La correlación en la población es cero).

$H_1 : \rho \neq 0$ (La correlación en la población es distinta de cero).

Debido a la forma en que está enunciada H_1 la prueba es de dos colas, es decir, se establecen valores críticos tanto positivos como negativos entre los cuales la t puede estar ubicada en cualquiera de los dos cuadrantes (ver gráfico 5.18).

El procedimiento de cálculo para encontrar t es el siguiente:

Fórmula:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \quad \text{con } n - 2 \text{ grados de libertad}$$

Donde:

r = coeficiente de correlación

n = total de la muestra

t = distribución de student

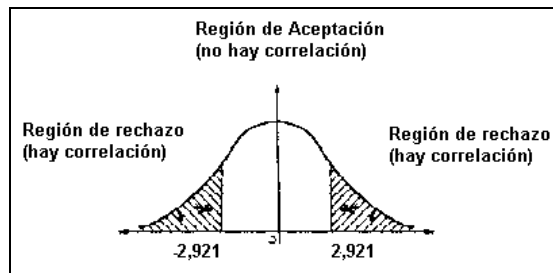
Proceso de cálculo:

$$t = \frac{0.68\sqrt{18-2}}{\sqrt{1-0,68^2}} = 3.70976541$$

Utilizando el nivel de significación 0.01, la regla de decisión indica que si la t calculada se encuentra en el área entre + 2,921 y - 2,921, se aceptará la hipótesis nula. Para localizar el valor crítico 2,921 se consultó

la tabla de distribución “t Student” para $n - 2$ grados de libertad, o sea $18-2= 16$. Mostrado en un esquema (Gráfico 5.19):

Gráfico 5.19. Esquema de distribución t Student para hipótesis 1



Conforme a la gráfica y los resultados de la t en la fórmula anterior se comprueba que la distribución de student está fuera de la región de aceptación o área de rechazo, de manera que H_1 es aceptada al nivel 0,01, lo cual significa que la correlación en la población es distinta de cero.

Como conclusión, se acepta que:

Cuanto mejor es el tiempo empleado en la planificación, menor es el error en la estimación del tiempo, con un coeficiente de correlación igual a 0,68 y un nivel de $p < 0,01$.

Prueba de la hipótesis 2

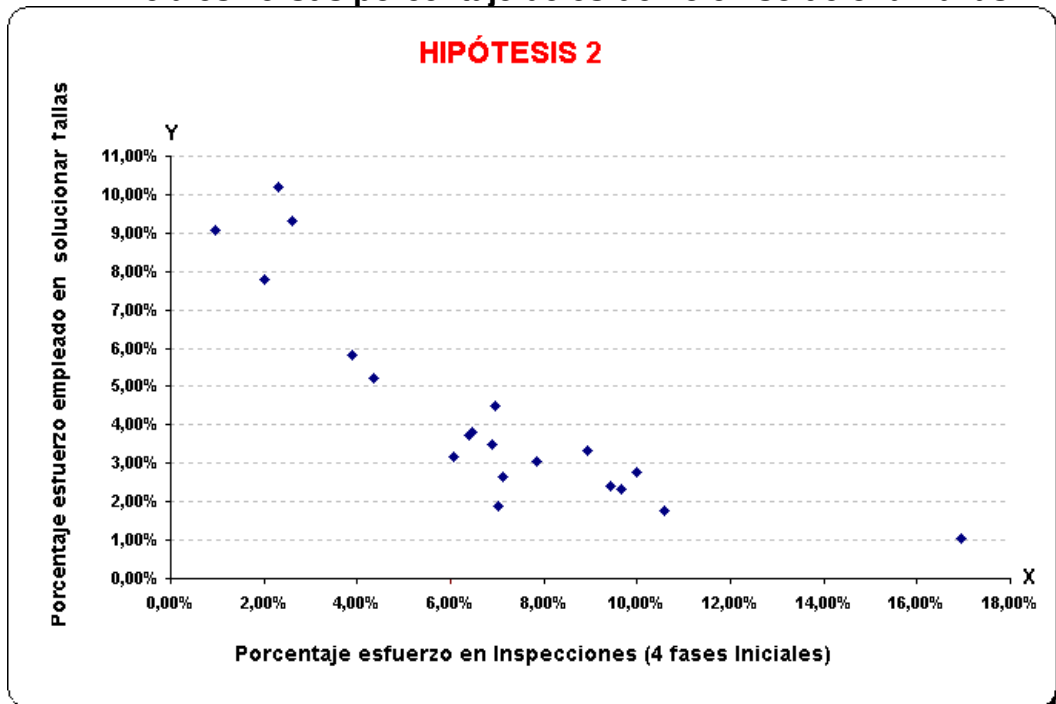
La segunda hipótesis a demostrar en el presente estudio responde al planteamiento hipotético referido en el Capítulo 3, que textualmente dice:

“Cuanto mayor es el número de inspecciones en las etapas de diseño y de implementación, más pequeño es el número de los errores en el producto final entregado a los clientes”

Para verificar la validez de la presente hipótesis debemos determinar la existencia de la relación entre el porcentaje de esfuerzo en inspecciones en fases iniciales (planificación, especificación, diseño y construcción) y el porcentaje de esfuerzo en solucionar fallas, como elementos fundamentales para concebir la cobertura de las inspecciones y su efecto en solucionar las fallas.

En la gráfica 5.20 se observa un punto apartado del resto, el mismo que se ha considerado como un dato aberrante.

Gráfico 5.20. Porcentaje de esfuerzo en inspecciones en fases iniciales versus porcentaje de esfuerzo en solucionar fallas



La variable “x” corresponde al dato “porcentaje de esfuerzo en inspecciones en fases iniciales” y la variable “y” para el dato “porcentaje de esfuerzo en solucionar fallas”. La tabla 5.2 muestra la información obtenida del estudio.

Tabla 5.2. Porcentaje de esfuerzo empleado en solucionar fallas versus porcentaje de esfuerzo inspecciones en las 4 fases iniciales

PROYECTO	% ESFUERZO EN INSPECCIONES EN FASES INICIALES (x)	% ESFUERZO EN SOLUCIONAR FALLAS (y)
1	3,89%	5,81%
2	6,06%	3,17%
3	2,00%	7,80%
4	2,30%	10,20%
5	6,96%	4,48%
6	6,40%	3,74%
7	0,97%	9,09%
8	7,01%	1,90%
9	2,60%	9,33%
10	9,66%	2,33%
11	10,00%	2,76%
12	9,44%	2,41%
13	7,86%	3,05%
14	7,11%	2,64%
15	6,45%	3,81%
16	6,90%	3,50%
17	10,58%	1,75%
18	8,93%	3,33%
19	4,36%	5,21%

Para demostrar la existencia de alguna recta obtendremos el coeficiente de correlación lineal r , cuyo cálculo es similar a lo realizado en la hipótesis 1:

$$S_{xy} = -0,0006579$$

$$S_x = 0,02892737$$

$$S_y = 0,02657549$$

$$r = \frac{-0,0006579}{(0,02892737)(0,02892737)}$$

$$r = -0,85 \quad \text{“COEFICIENTE DE CORRELACIÓN”}$$

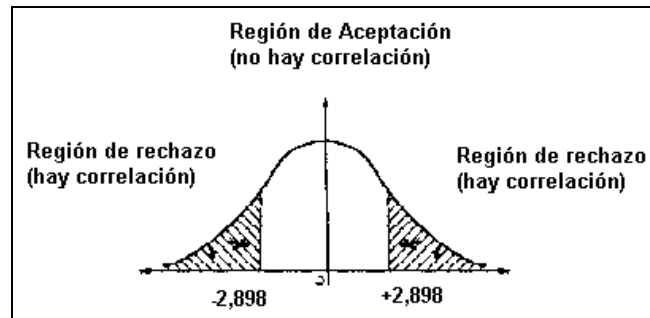
Sobre este aspecto la teoría dice que: “el valor de coeficiente de relación (r) puede ir de -1.00 a +1.00”, por lo tanto el valor obtenido en el presente estudio de **r= -0.85** demostraría que las variables porcentaje de esfuerzo en inspecciones en fases iniciales y el porcentaje de esfuerzo en solucionar fallas bien correlacionadas se encontrarían coherentemente correlacionadas.

En el presente caso, al ser una muestra pequeña ($n < 50$), $n=19$ ($20-1$), es necesario probar que el coeficiente de correlación (r) encontrado es representativo de la industria, para lo cual se procede a realizar la prueba de significación utilizando la distribución **t de Student** para muestras pequeñas, igual a lo realizado en la hipótesis 1, obteniendo lo siguiente:

$$t = \frac{0.85\sqrt{19-2}}{\sqrt{1-0,85^2}} = -6.82102134$$

Utilizando el nivel de significación 0.01, la regla de decisión indica que si la **t** calculada se encuentra en el área entre + 2,898 y - 2,898, se aceptará la hipótesis nula. Para localizar el valor crítico 2,898 se consultó la tabla de distribución t Student para $n - 2$ grados de libertad, o sea $19-2=17$. Mostrado en un esquema (Gráfico 5.21):

Gráfico 5.21. Esquema de distribución t Student para hipótesis 2



Conforme a la gráfica y los resultados de la t en la fórmula anterior se comprueba que la distribución de student está fuera de la región de aceptación o área de rechazo, de manera que H_1 es aceptada al nivel 0,01, lo cual significa que la correlación en la población es distinta de cero.

Como conclusión, se acepta que:

Cuanto mayor es el porcentaje de esfuerzo en inspecciones en fases iniciales, más pequeño es el porcentaje de esfuerzo empleado en solucionar las fallas, con un coeficiente de correlación igual a -0,85 y un nivel de $p < 0,01$.

5.3. Comparación con estudios de otros países

A través de la investigación realizada encontramos un estudio publicado por la Escuela de Negocios de la Universidad de Carleton Ontario Canadá [25], el cual tenía como objetivo determinar factores que podrían incidir en la reducción del tiempo de desarrollo de software. Una hipótesis planteada fue: “un mayor tiempo invertido en la planificación está relacionado con un menor tiempo de desarrollo del software”, similar a una de nuestras hipótesis. El estudio canadiense fue realizado con una muestra de 44 empresas estadounidenses y canadienses, obteniendo un coeficiente de correlación r de -0.12 entre las variables esfuerzo en planificación y la duración del proyecto. Concluyendo que, la planificación como proporción del total del tiempo de desarrollo no está correlacionado con la duración del mismo, es decir, que los proyectos que requieren una mayor planificación toman más tiempo realizarlo.

Dicho resultado complementa nuestro estudio (hipótesis 1). Analizando el contexto sobre el cual se llevó a cabo el estudio canadiense, ciertamente conformado por empresas grandes, podríamos deducir que una buena planificación utilizando un mayor número de horas está relacionada con el tamaño de la empresa y del proyecto. Es decir, que si una empresa grande con proyectos grandes planifica arduamente, su proyecto podría

extenderse por la grandeza del mismo, mientras que una empresa pequeña (menor a 10 empleados) con proyectos pequeños (menor a 6 meses) podría mejorar su estimación del tiempo si le da mayor importancia a la planificación.

A continuación describimos brevemente el contenido de los estudios de otros países:

- **El sector de software y servicios informáticos (SSI) en Argentina [26]**

Este estudio se enfoca en el análisis de la situación y perspectivas del sector proveedor de software y servicios informáticos, tratando de encontrar las oportunidades, restricciones y desafíos para su futuro desarrollo. Inicialmente se presentan los principales datos del mercado mundial de SSI y se analiza las experiencias de otros países que han tenido un tardío ingreso a este sector. Luego se discute la oferta que localmente ofrece el sector de software y servicios informáticos, además de los niveles de facturación y empleo de las empresas del sector, presencia de empresas extranjeras, áreas de negocio, perfil de usuarios, empleo de recursos humanos y herramientas de programación. Posteriormente se discute el acceso al financiamiento, disponibilidad y costo de los recursos humanos (enfocándose en la

oferta laboral), infraestructura tecnológica y de comunicaciones e incidencia de las políticas públicas. Finalmente se presentan las conclusiones de lo anteriormente presentado y se discuten las perspectivas que enfrenta el sector a futuro.

- **Challenges of Software Product Companies: Results of a National Survey in Finland [27]**

Este estudio presenta los resultados obtenidos de una encuesta realizada en Finlandia con el fin de analizar como han madurado y evolucionado a través de los años las compañías de software. Inicialmente se presentan datos correspondientes a las características de la industria tales como, las ganancias reportadas por las compañías de software en el 2002, total de beneficios por empleado en base a la madurez (años) de las empresas en donde trabajan, entre otras. Luego muestra las funciones que desempeñan los empleados de la industria del software al igual que las áreas que mayor desarrollo han tenido en el periodo 2003 – 2005 (Ej: Desarrollo de software, redes, planificación estratégica, entre otras). Finalmente se presentan las principales funciones que las empresas estarían dispuestas a subcontratar (Ej: Planificación, Programación, Administración de proyectos, entre otras) y los factores que evitan que

estas empresas subcontraten dichas funciones (Ej: Dificultad de modularidad, falta de calidad en el trabajo, entre otras).

Los estudios realizados por otros países no se enfocan directamente al proceso de desarrollo de software sino en el contexto administrativo, económico y humano en el cual la actividad de desarrollo de software se realiza.

Otra clase de estudios relacionados con el desarrollo de software presentan métodos para mejorar el mismo, más no resultados obtenidos del proceso de desarrollo. A continuación nombramos algunos de ellos:

- **On the problem of the software cost function [28]**

Presenta algunas variantes para una función que describa el costo de un proyecto utilizando el esfuerzo empleado en el desarrollo del mismo. Utilizando Programación Genética (GP) se exploran las posibles funciones de costo.

- **Measuring the usability of software components [29]**

Este estudio presenta un conjunto de mediciones para determinar la usabilidad de los componentes de software y describe el método utilizado para obtenerla y validarla.

Pese a la investigación realizada para obtener estudios enfocados al proceso de desarrollo de software en otros países con indicadores similares a los realizados y analizados por esta tesis, no se logró obtener un volumen considerable de información para realizar la comparación respectiva, sin embargo, se comparará con un último estudio realizado en el Ecuador por el Vllir-Espol a 40 empresas en las ciudades de Quito y Guayaquil, en los meses de Abril y Mayo de 2006. [4], mismo que fue descrito en el Capítulo 1.

Al efecto se realizará la comparación, respecto a:

Complejidad

Gráfico 5.3.1. [4] obtenida por plan piloto

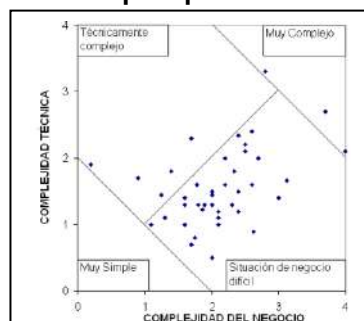
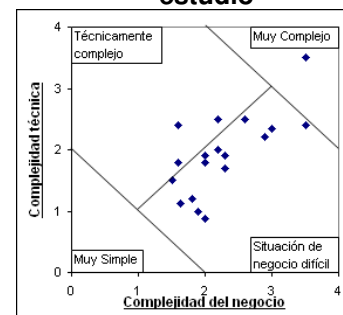


Gráfico 5.3.2. obtenido por este estudio



En ambos gráficos se observa que los proyectos han sido realizados para negocios complejos. Al igual, pocos proyectos han sido altamente complejos en la parte técnica. No han existido proyectos catalogados como muy simples, por lo que estaríamos hablando de empresas de similares características.

Distribución de inspecciones

Gráfico 5.3.3. [4] obtenida por plan piloto

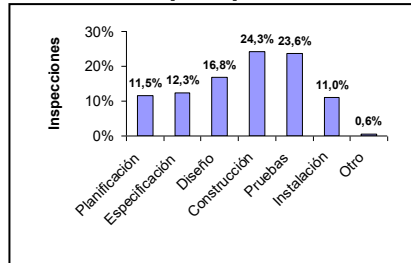
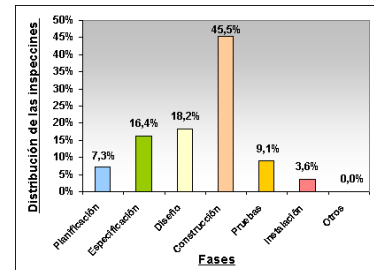


Gráfico 5.3.4. obtenido por este estudio



Como se evidencia en las gráficas, el actual estudio ha tenido más porcentaje de inspecciones realizadas en la fase de construcción.

Ocurrencia de defectos

Gráfico 5.3.5 [4] obtenida por plan piloto

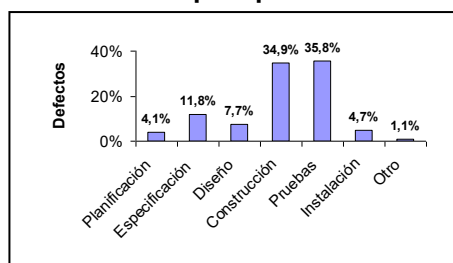
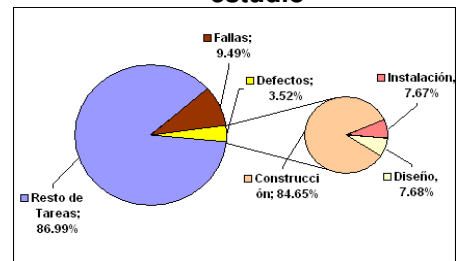


Gráfico 5.3.6. obtenido por este estudio



Respecto a la ocurrencia de defectos en las fases se tiene cierta concordancia con el estudio realizado por el Vllr-Espol a mediados del 2006, ya que la mayor cantidad de defectos en ambos estudios se demuestran en las fases de construcción e instalación.

El estudio realizado por Vllir-Espol responde a datos estimados y que sirvieron de punto de partida para el Plan Piloto [8] y a su vez para el presente estudio.

CONCLUSIONES Y RECOMENDACIONES

En los apartados siguientes se procede a mencionar las conclusiones y recomendaciones que el presente estudio ha llegado

CONCLUSIONES

Sobre las hipótesis:

- Que la variable porcentaje de esfuerzo empleado en planificar, esta relacionada con el porcentaje de error en la estimación del esfuerzo; lo que afirma el planteamiento hipotético en el sentido que “a mayor esfuerzo (horas) por parte de las empresas desarrolladoras de software en la planificación, disminuye el porcentaje de error en la estimación del proyecto”.
- Que la variable esfuerzo empleado en las 4 fases iniciales esta relacionada con el esfuerzo empleado en solucionar las fallas; lo que afirma el planteamiento hipotético en el sentido que “a mayor esfuerzo (horas) de las empresas en sus fases iniciales (planificación, especificación, diseño, construcción), disminuirían el esfuerzo (horas) que se emplea en solucionar las fallas que se les presenta al usuario”.

Sobre los indicadores en general:

- Que a menor participación en el proyecto del usuario, este podría encontrar limitantes para comprenderlo y como consecuencia no hacer una buena evaluación de la satisfacción.
- Que a mayor participación del cliente, la calificación del nivel de satisfacción decrece, lo que podría explicarse por el perfil del evaluador, que al tener criterios y funciones de tipo comercial, prefiere asignar el menor tiempo posible a aspectos de orden operativo.
- Que la gran mayoría de los proyectos producto de este estudio carecen de adecuada estimación, lo que respondería al escaso esfuerzo en la planificación de los mismos.
- Que la sobre o subestimación del esfuerzo en los proyectos se deriva de una deficiente planificación.
- Que en lo concerniente a documentación se concluye que en el Ecuador al igual que en las empresas latinoamericanas no existe cultura de documentación de los proyectos informáticos [32], lo que constituye un alto nivel de riesgo operativo.
- Que en la complejidad técnica versus la complejidad del negocio una considerable parte de proyectos han sido realizados para negocios complejos; así mismo se apreció pocos proyectos que han derivado alta complejidad y más aún encontramos que en el rango de proyectos muy complejos existió un solo caso, y además que no han existido proyectos

que puedan catalogarse en la calidad de muy simples, lo que ratifica el perfil de las empresas elegidas para el estudio.

- Que la gran mayoría de los proyectos informáticos carecen de una adecuada estimación en el costo, lo que respondería al escaso esfuerzo en la planificación de los mismos.
- Que toda sobre o subestimación en el costo del proyecto se derivaría de deficiente planificación.
- Que la realidad del mercado salarial ha evidenciado la existencia de buenos desarrolladores a un costo hora relativamente bajo, particularmente en empresas pequeñas.
- Que el costo promedio hora varía según la categoría de recursos participantes en los proyectos, de manera que proyectos en los que intervienen categorías jerárquicas altas derivaran mayor costo hora promedio y contrariamente proyectos en los que intervienen categorías jerárquicas de menor nivel tienen bajo costo promedio hora; en el presente estudio se evidenció un relativo equilibrio en este sentido.
- Que el costo hora promedio obtenido por el presente estudio de \$4.12 guarda relación con publicado por la CORPEI, esto es, Ingeniero / Analista de Sistemas Senior = \$6.18 (calculado de 160 horas); y, Ingeniero / Analista de Sistemas Junior = \$3.80 (calculado de 160 horas). **[33]**, aspecto que confirmaría la validez del mismo.

- Que la mayor cantidad de inspecciones en el presente estudio se realizaron a partir de la segunda fase, esto es, en la especificación.
- Que indicadores como la variación porcentual en los requerimientos proporcionan información que se considera de alta utilidad práctica para futuros proyectos a fin de que los encargados de ejecutarlos analicen y planifiquen apropiadamente el levantamiento de información y la coordinación con los clientes y/o usuarios.
- Que la mayor proporción de los defectos y fallas presentados en el estudio, catalogados por su severidad, se ubican en la categoría de leves, aspecto que no evidencia mayor riesgo, ya que la demanda de recursos para su regularización es relativamente baja.
- Que respecto al porcentaje de tiempo utilizado para la corrección de defectos en el presente estudio fue bajo, ya que la experiencia de personas conocedoras de estos temas opinan que la dedicación de tiempo a la corrección de defectos sería mucho mayor **[36]**, por lo que los resultados que presenta el estudio estaría dados por la práctica de no calificar a los defectos sino únicamente cuando son reportados por el líder de proyectos o el cliente/usuario.
- Que respecto al tiempo utilizado en la corrección de fallas en la práctica se considera razonable **[36]**.
- Que el tiempo de respuesta promedio general para la atención de fallas se lo consideraría razonable en este estudio. **[36]**

Sobre la relación entre variables

En el capítulo 3 “Desarrollo” sección 3.2 “Objetivos” se plantearon relaciones entre algunas variables, en este apartado se procede a explicar las conclusiones a las que se ha llegado.

Para encontrar la existencia de las relaciones entre las variables se ha empleado la metodología seguida en el capítulo 5 “Análisis y presentación de resultados” sección 5.2 “Pruebas para verificar Hipótesis”, misma que sirvió para verificar las hipótesis.

Con la aclaración anterior, se procede a detallar las conclusiones a las cuales el presente estudio ha llegado:

- Que no se ha encontrado evidencia que exista relación entre las variables esfuerzo en desarrollar el proyecto y la complejidad técnica, ya que el coeficiente de correlación “r” es de -0.058. **ANEXO 7.1**
- Que no se ha encontrado evidencia que exista relación entre las variables esfuerzo en desarrollar el proyecto y la complejidad del negocio, ya que el coeficiente de correlación “r” es de -0.07. **ANEXO 7.1**
- Que las variables esfuerzo del cliente y la satisfacción del mismo se encuentran relacionadas con un coeficiente de correlación “r” de 0.71, lo que indicaría que a mayor cantidad de horas de participación el cliente tendría un mayor nivel de satisfacción. Con la aclaración de que a mucha

participación el nivel de satisfacción baja, esto podría ser causado por la naturaleza del cliente, ya que el se encarga más del aspecto comercial que del operativo, como se ha mencionado en conclusiones anteriores.

ANEXO 7.2

- Que las variables esfuerzo del usuario y la satisfacción del mismo se encuentran relacionadas con un coeficiente de correlación “r” de 0.63, lo que indicaría que a mayor cantidad de horas de participación el usuario tendría un mayor nivel de satisfacción. **ANEXO 7.2**
- Que no se ha encontrado evidencia que exista relación entre las variables esfuerzo en desarrollar el proyecto y el número de personas involucradas en el mismo, ya que el coeficiente de correlación “r” es de 0.13.

ANEXO 7.3

- Que las variables defectos y complejidad técnica se encuentran débilmente relacionadas con un coeficiente de correlación “r” de 0.35, es decir que no se puede asegurar que la cantidad de defectos dependa de la complejidad técnica. **ANEXO 7.4.**
- Que las variables defectos y complejidad del negocio se encuentran débilmente relacionadas con un coeficiente de correlación “r” de 0.44, es decir que no se puede asegurar que la cantidad de defectos dependa de la complejidad del negocio. **ANEXO 7.4.**

- Que las variables fallas y complejidad técnica se encuentran débilmente relacionadas con un coeficiente de correlación “r” de 0.32, es decir que no se puede asegurar que la cantidad de fallas dependa de la complejidad técnica del proyecto. **ANEXO 7.4.**
- Que no se encontró evidencia que las variables fallas y complejidad del negocio se encuentran relacionadas con un coeficiente de correlación “r” de 0.01. **ANEXO 7.4.**
- Respecto a los defectos y fallas presentadas no se encontró evidencia que la presencia de los mismos tenga relación con las complejidades del negocio y la complejidad técnica del proyecto, ya que pueden existir otros factores como se lo ha explicado en conclusiones anteriores.
- Que no se ha encontrado evidencia que exista relación entre las variables esfuerzo en documentación empleado en el proyecto y el número de fallas encontradas en el mismo, ya que el coeficiente de correlación “r” es de -0.16. **ANEXO 7.5.**

Sobre aspectos generales

- Que los resultados respecto a los defectos presentados son considerablemente bajos, lo que indicaría ausencia de documentación en todos los casos, que a su vez confirmaría la aseveración de algunas empresas, en el sentido que, “para los desarrolladores (programadores)

su software es casi perfecto, que nunca tiene defectos, sino hasta que el líder del proyecto o el cliente/usuario los encuentra, todo esto siempre en la fase de pruebas”.

- Que para realizar este tipo de estudios es necesario además de un medido análisis, una prudente investigación, unido a una actitud perseverante y persistente, ya que las empresas ecuatorianas si bien están interesadas en este tipo de proyectos, desafortunadamente no disponen de tiempo ni recursos para coadyuvar a dichos estudios.

RECOMENDACIONES

Complejidad Ciclomática

- Es recomendable en proyectos futuros utilizar otro método para obtención de la complejidad ciclomática, el mismo que podría involucrar los siguientes aspectos técnicos y doctrinarios:

La complejidad ciclomática es una medida cuantitativa de la complejidad lógica de un programa. Define el número de caminos independientes del conjunto básico de un módulo y por tanto nos proporciona una cota superior del número máximo de pruebas que se han de realizar para garantizar que cada sentencia se ejecuta al menos una vez. **[24]**

Un camino independiente es cualquier camino del programa que incorpora frente a otros caminos un nuevo conjunto de sentencias o una

nueva condición, es decir, ha de recorrer al menos una arista no recorrida hasta entonces. [24]

Un conjunto de caminos componen un conjunto básico de caminos si al recorrerlos se recorre al menos una vez cada sentencia del programa y cada condición por tanto se habrá ejecutado en su parte verdadera y falsa. Un conjunto básico de caminos no es único. [24]

El número de caminos que componen un conjunto básico de caminos viene determinado por la complejidad ciclomática, que se calcula como:

[24]

1. El número de regiones del grafo de flujo.
2. $C(G) = A - N + 2$ $C(G)$: complejidad ciclomática,
G: grafo de flujo.
A: número de aristas.
N: número de nodos.
3. $C(G) = P + 1$ P: número de nodos predicado.

Adicionalmente en el **ANEXO 8** se encontrará un ejemplo sencillo para el cálculo de complejidad ciclomática.

Aplicar este cálculo al código fuente de las empresas desarrolladoras de software sería bastante complejo y demandaría una cantidad considerable de tiempo. Es por eso, que para futuros estudios se recomienda desarrollar instrumentos de medición sencillos enfocados a la

medición de la complejidad ciclomática, tomando en consideración que dichos instrumentos deben facilitar el llenado y el cálculo de la misma.

Líneas de código

El tamaño del software ha sido medido por el conteo de líneas de código, la cantidad de programas fuentes o técnicas similares. El plan piloto utilizó el conteo por líneas de código, el mismo que fue adoptado por el presente estudio para su validación. Lastimosamente el conteo semanal de líneas de código desarrolladas, no se lo pudo obtener debido a que las empresas no llenaron el dato solicitado (indicador 12), entre las causas citamos las siguientes:

- El resultado del conteo depende de la complejidad técnica y el lenguaje de programación (Ej: Java, PHP, C++, ASP, etc).
- No es un indicador muy bueno debido a que puede variar con la habilidad de cada programador o con el uso de metodologías adoptadas por cada empresa.

Por lo tanto se recomienda pasar a la medición del tamaño funcional del software, a través de la utilización del método de puntos de función, el mismo que pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada. **[30]**

El IFPUG (Internacional Function Point Users Group), llamado Function Point Analysis (FPA), es definido como método estándar para medir el desarrollo de software desde el punto de vista del usuario.

Dicho método se basa en la identificación de componentes del sistema en términos de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio. A cada uno de estos componentes, les asigna un número de puntos por función basándose en el tipo de componente y su complejidad; y la sumatoria de esto nos da los puntos de función sin ajustar. El ajuste es un paso final basándose en las características generales de todo el sistema informático que se está contando.

En el **ANEXO 9**, se presenta el procedimiento de dicho método, mismo que ha sido tomado del “Manual para la Medición de Puntos de Función” publicado por la IFPUG **[30]**

Sobre aspectos generales

- Es importante diferenciar al cliente y al usuario, dado que el perfil del cliente tiene criterios y funciones comerciales, prefiriendo asignar menor tiempo a aspectos operativos, no así el usuario que necesita estar al tanto de lo que se está realizando.

- Es necesario realizar una buena planificación de los proyectos, puesto que como se ha analizado una buena fase de planificación disminuye el riesgo de defectos, fallas, errores en la estimación del costo y del esfuerzo.
- Es necesario realizar una apropiada planificación con un adecuado levantamiento de información sumado a una correcta coordinación con los clientes y/o usuarios. Esto con el propósito de no tener problemas conforme avanza el proyecto.
- Es importante dedicar una considerable cantidad de tiempo a la documentación de los procesos, puesto que el requisito fundamental para obtener certificaciones ISO9001:2000 ò CMMI es que la empresa tenga y lleve los procesos debida y correctamente documentados **[34]**
[35].

BIBLIOGRAFÍA

[1] AESOFT, “Publicación de la AESOFT sobre la industria del Software en Ecuador”, disponible en <http://www.aesoft.com.ec>, última visita: Septiembre 2006.

[2] D. Salazar, M. Villavicencio, V. Macías, M. Snoeck, “Estudio Estadístico Exploratorio de las Empresas Desarrolladoras de Software Asentadas en Guayaquil, Quito y Cuenca”, Anales 2004 33 JAIIO (Jornadas Argentinas de Informática e Investigación Operativa), Córdoba, Argentina.

[3] J. Mazón, M. Villavicencio, J. Alvear, G. Bracco, “Aspectos de la Calidad y Dificultades en la Gestión de Proyectos de Software: Estudio exploratorio, Proceedings de las Segundas Jornadas de Ingeniería de Software ESPOL 2005, Guayaquil-Ecuador.

[4] M. Villavicencio, J. Mazón, J. Alvear, “Elaboración y análisis de métricas para el proceso de desarrollo de software, “aplicado a pymes ecuatorianas desarrolladoras de software””: Tercer Estudio, Proceedings de las Terceras Jornadas de Ingeniería de Software ESPOL 2006, Guayaquil-Ecuador.

[5] M. Villavicencio, E. Izquierdo, M. Snoeck, "Aspects Regarding Management of Software Projects that Influence the Quality of Software Products Released: The Case of Ecuador", 2005.

[6] PROSOFT, "Estudio del nivel de madurez y capacidad de procesos de la industria de tecnologías de información", México 2004.

[7] MINCOMEX y PROEXPORT COLOMBIA, "Estudio de mercado - España", disponible en <http://www.proexport.com.co/VBeContent/library/documents/DocNewsNo1535DocumentNo924.PDF>, última visita: Septiembre 2006.

[8] J. Mazón, J. Alvear, "Elaboración y análisis de métricas para el proceso de desarrollo de software para empresas desarrolladoras de software del ecuador": Plan piloto 2006 - TESIS, ESPOL, Guayaquil-Ecuador.

[9] IEEE, "Standard glossary of software engineering terminology", disponible en (última visita 25 de octubre de 2006) www.swen.uwaterloo.ca/~bpekilis/public/SoftwareEngGlossary.pdf

[10] González D. H. (2001) Las métricas de software y su uso en la región, Tesis de Licenciatura, Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas-Puebla.

[11] Wikipedia, “ISO 9001”, disponible en http://es.wikipedia.org/wiki/ISO_9001, última visita: Septiembre 2006.

[12] Wikipedia, “ISO 9001:2000”, disponible en http://es.wikipedia.org/wiki/ISO_9001:2000, última visita: Septiembre 2006.

[13] Wikipedia, “Modelo de Capacidad y Madurez”, disponible en http://es.wikipedia.org/wiki/Modelo_de_Capacidad_y_Madurez, última visita: Septiembre 2006.

[14] Wikipedia, “CMMI”, disponible en <http://es.wikipedia.org/wiki/CMMI>, última visita: Septiembre 2006.

[15] SYNSPACE, “Software Engineering Glossary”, disponible en <http://www.synspace.com>, última visita: Septiembre 2006.

[16] Ian Sommerville. (2002). Ingeniería de Software, 6ta edición. Addison Wesley.

[17] José A. Mañas. “Pruebas de Programas”, disponible en <http://www.upv.es/protel/usr/jotrofer/pascal/testing.htm>, última visita: Septiembre 2006.

[18] Diego Lucio D'Onofrio. “Probando Software y números de versión”, disponible en <http://www.elguille.info/Clipper/probando.htm>, última visita: Septiembre 2006.

[19] Wikipedia, “Revisión por pares”, disponible en http://es.wikipedia.org/wiki/Peer_review, última visita: Septiembre 2006.

[20] Robert E. Parl, Wolfhart B. Goethert, William A. Florac , 1996, “Goal-Driven Software Measurement – A Guidebook”, Software Engineering Institute.

[21] William Florac, Anita Carleton, “Measuring the Software process”, 1999, Software Engineering Institute, Addison Wesley.

[22] Mason y Lind, “Estadística para Administración y Economía”, 1992, Alfaomega.

[23] Martínez E. Medición. Requisitos y procedimientos para construir un instrumento de medición. Universidad Metropolitana, Venezuela. 2004, <http://medusa.unimet.edu.ve/didactica/fpdd49/Lecturas/Archivo%20Word/MEDICIONREQUISITOS.doc>. (última visita: Julio 2004).

[24] Escuela Superior de Ingeniería Informática, “Complejidad ciclomática”, disponible en <http://trevinca.ei.uvigo.es/~rlaza/teoria.htm> (Técnicas de verificación y pruebas), última visita: 9 de Octubre de 2006.

[25] John Callahan , Brian Moretton, “Reducing software product development time”, 2000, School of Business, Carleton University

[26] Daniel Chudnovsky, Andrés López, Silvana Melitsko, “El sector de software y servicios informáticos (SSI) en la Argentina: situación actual y perspectivas futuras”, disponible en: www.fundcenit.org.ar/Descargas/dt27.pdf (Ultima Visita) Octubre 2006

[27] Juhana Hietala, Jyrki Kontio, Jani-Pekka Jokinen, Jarkko Pyysiäinen, “Challenges of Software Product Companies: Results of a National Survey

in Finland”, 2004, Helsinki University of Technology (Software Business and Engineering Institute)

[28] J.J. Dolado, “On the problem of the software cost function”, 2000, Facultad de Informática, Universidad del País Vasco-Euskal Herriko Unibertsitatea, 20 009 Donostia, España.

[29] Manuel F. Bertoa , José M. Troya, Antonio Vallecillo, “Measuring the usability of software components”, 2005, Dpto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga.

[30] International Function Points Users Group (IFPUG), “Manual para la Medición de Puntos Función”.

[31] Microsoft, “¿Por qué su empresa no se puede permitir una mala documentación?”, disponible en http://www.microsoft.com/spain/empresas/rpp/mala_documentacion.msp x, última visita: Octubre 2006

[32] Deloitte & Touch, “Estudios de Gestión de Riesgos en Instituciones del Sistema Financiero Latinoamericano”, 2004

[33] Corporación de Promoción de Exportaciones e Inversiones - CORPEI, "Condiciones del mercado laboral", disponible en, www.ecuadorinvest.org/ecuadorinvest/docs/10_8Condiciones_del_Mercado_Laboral.pdf, última visita: Octubre 2006.

[34] Instituto Argentino de normalización y certificación “ Orientación acerca del enfoque basado en procesos para los sistemas de gestión de calidad”, Argentina, Mayo 2001 , pp. 1 – 12

[35] Joaquín García, “CMMI – CMMI nivel 2”, España ,Noviembre 2005, pp. 1-5

[36] Empresas de la muestra en estudio Quito y Guayaquil, Información verbal de representantes de las compañías, Octubre 2006.

[37] Horngren-Sundem-Stratton, “Contabilidad Administrativa”, México, Prentice Hall, 2001

ANEXO 1: Plan de Capacitación (Primera versión)

Proceso de medición dirigido a objetivos



Proceso de Medición dirigido a objetivos

- El objetivo de este proceso es obtener información que ayude a lograr los objetivos de negocio de las empresas desarrolladoras de software:
 - Maximizar la **Satisfacción** del Cliente.
 - Mejorar la **Calidad** del Producto de Software.
 - Mejorar la **Productividad**.

- Y además mantener un seguimiento desde las mediciones específicas hacia los objetivos de negocio, para que los esfuerzos de medición no sean en vano.

Proceso de Medición dirigido a objetivos

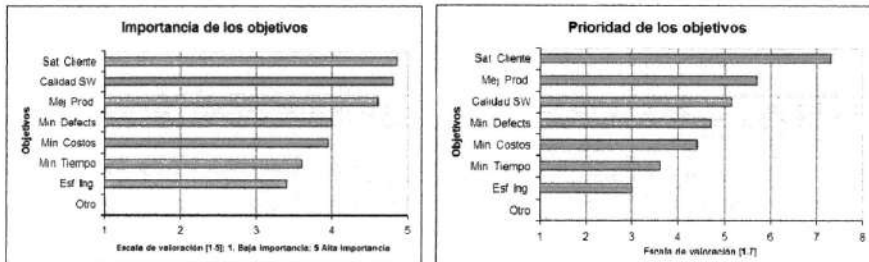
■ Consta de 10 pasos:

- 1.-) Identificar los Objetivos del Negocio.
- 2.-) Identificar qué se desea saber o aprender.
- 3.-) Identificar los Sub-Objetivos.
- 4.-) Identificar las entidades y atributos relacionadas a los Sub-Objetivos.
- 5.-) Formalizar los Objetivos de Medición.
- 6.-) Identificar preguntas cuantificables y sus indicadores relacionados que se usarán para ayudar a conseguir los objetivos de medición.
- 7.-) Identificar los elementos de datos que se recolectarán para construir los indicadores que ayudarán a responder las preguntas.
- 8.-) Definir las mediciones a ser usadas, y convertir esas definiciones de manera operacional.
- 9.-) Identificar las acciones que se tomarán para implementar las mediciones.
- 10.-) Preparar un plan para implementar las mediciones.

Paso 1: Identificar los objetivos de negocio

- Es el primer paso para identificar y definir mediciones específicas del software y consiste en identificar los objetivos de negocio en los cuales están basados los esfuerzos de la organización.
- Este paso debe dar inicio en cualquier nivel organizacional donde los objetivos puedan ser razonablemente identificados.
- Inicialo en los niveles altos tiene la ventaja de asegurar el seguimiento de las mediciones resultantes hacia los objetivos de negocio primarios, lo que no se tiene en niveles bajos.

Paso 1: Identificar los objetivos de negocio



- En una encuesta aplicada a Gerentes y Administradores de Empresas Desarrolladoras de Software se obtuvieron los siguientes 3 Objetivos de Negocio:
 - Maximizar la **Satisfacción** del Cliente.
 - Mejorar la **Calidad** del Producto de Software.
 - Mejorar la **Productividad**.

Paso 2: Identificar qué se desea saber o aprender

- Consiste en identificar que es necesario conocer para entender, medir, predecir o mejorar las actividades necesarias para lograr los objetivos.
- La herramienta recomendada para identificar estas preguntas es la lista de entidades y preguntas, la cual tiene los siguientes pasos:
 - 1.-) Seleccionar un objetivo de negocio.
 - 2.-) Identificar las personas o grupos involucrados con este objetivo.
 - 3.-) Por cada rol de las personas involucradas identificar los procesos que manejan o afectan.

Paso 2: Identificar qué se desea saber o aprender

- 4.-) Listar las cosas importantes (entidades) que son manejadas dentro de cada proceso por cada rol. Las entidades a ser obtenidas se encuentran dentro de los siguientes procesos:
 - *Entradas y recursos.
 - *Productos y derivados.
 - *Artefactos internos tales como inventario y trabajo en proceso.
 - *Actividades y Flujo.

- 5.-) Por cada entidad, listar las preguntas que si fueran contestadas ayudarían a dicha persona en su rol a conseguir el objetivo de negocio seleccionado. Ej:
 - *¿Qué tan grande es?
 - *¿Cuán rápido se lo realiza?
 - *¿Cuántos componentes tiene?

- 6.-) Revisar por cada uno de los procesos si no falta nada.

- 7.-) Repetir las tareas 1 al 6 para cada uno de los objetivos de negocio.

Paso 2: Identificar qué se desea saber o aprender

Entradas y Recursos	Preguntas relacionadas a los objetivos de negocio
Cliente	Están nuestros clientes satisfechos con el SW que producimos ¿Cuáles son los factores del SW que afectan la satisfacción del cliente En que fases del desarrollo del SW está involucrado el cliente
Personal	La experiencia del personal asignado al proyecto favorece el desarrollo del SW En que medida, la rotación del personal afecta el desarrollo del SW Se asigna al proyecto de SW el personal necesario para el desarrollo del mismo
Consumibles	Se han cambiado el tiempo y costo establecidos durante la realización del proyecto Las estimaciones de tiempo y costo son apropiadas para desarrollar el proyecto
Herramientas	Que herramientas se prefiere utilizar durante el desarrollo del proyecto Los diferentes entregables principalmente los diseños se desarrollaron utilizando herramientas CASE Las herramientas utilizadas permiten mejorar el desempeño del personal

Paso 2: Identificar qué se desea saber o aprender

Actividades y flujos	
Planeación del proyecto	El cliente participó en el desarrollo de los proyectos El proyecto se planifica usando datos de referencia de proyectos anteriormente realizados El proyecto se desarrolla de acuerdo al cronograma establecido
Especificación de requerimientos	Los requerimientos están adecuadamente definidos al inicio del proyecto Se utiliza el tiempo necesario para el buen establecimiento de los requerimientos Se cuenta con el apoyo de personas con conocimiento del dominio del negocio
Diseñar	Los diseños son revisados y aprobados por el cliente Que proporción del tiempo del proyecto se dedica al diseño Se toman en cuenta patrones de diseño en el desarrollo del proyecto
Construir	El SW se construye utilizando algún estándar de codificación En que proporción se han reutilizado componentes de software en nuevos proyectos Cuanto tiempo toma construir el SW
Probar	Cuanto tiempo se emplea en probar el software desarrollado Se ejecutaron las pruebas suficientes en el SW Qué efecto tienen en la duración del proyecto los defectos detectados en las pruebas
Instalar	Se presentan defectos durante la instalación y posterior uso del SW La decisión de liberar el producto está debidamente planificada La instalación del SW se hace en el tiempo planificado
Inspecciones y Reuniones	Con que frecuencia al cliente participa en reuniones Con que frecuencia se realizan inspecciones Cuanto tiempo emplea el personal para realizar reuniones

Paso 2: Identificar qué se desea saber o aprender

Artefactos Internos	
Presupuesto	Se cuenta con el presupuesto adecuado para todas las actividades planificadas Se dan cambios al presupuesto durante el desarrollo del proyecto
Cronograma	El tiempo establecido es el adecuado para el desarrollo de SW de calidad Se conoce la diferencia entre el tiempo planificado y el consumido durante el proyecto

Productos y derivados	
Requerimientos	Cuantos requerimientos están desarrollados y cuantos faltan desarrollar Están los requerimientos claros y completamente definidos De qué forma cambian los requerimientos durante el desarrollo del proyecto
Diseño del SW	El diseño del SW está aprobado por el cliente y/o usuario final Los diseños están libres de defectos Se realizan inspecciones sobre los diseños (Quién, Frecuencia)
Código fuente	Se utiliza algún estándar para la codificación El código fuente está libre de defectos cuando es liberado Se han re-utilizado componentes de SW
Pruebas	El cliente ayuda a establecer los escenarios que servirán para probar el SW Cuanto tiempo se dedica para las pruebas del SW Cuál es la cantidad de defectos descubiertos y cuanto toma corregirlos
Documentación	Se dispone de una adecuada documentación sobre el desarrollo del proyecto Se utilizarán plantillas para la documentación del proyecto Cuál es el esfuerzo necesario para llevar la documentación del desarrollo del SW
Conocimientos, Experiencias	El personal cuenta con experiencia en el dominio de negocio del cliente Es necesario capacitar al personal durante el desarrollo del proyecto Los involucrados en el desarrollo del proyecto cuentan con el apoyo de expertos

Paso 3: Identificar los Sub-Objetivos

- Consiste en transformar los objetivos de negocio de alto nivel en Sub-Objetivos que se relacionen específicamente con actividades que se manejan o desarrollan.
- Se agrupan las preguntas que se refieran a problemas similares. Estos problemas son potenciales Sub-Objetivos.

Paso 3: Identificar los Sub-Objetivos

Grupo No 1 Cliente	Están nuestros clientes satisfechos con el SW que producimos
	El cliente participó en el desarrollo de los proyectos
	Los diseños son revisados y aprobados por el cliente
	Con que frecuencia el cliente participa en reuniones
	El diseño del SW está aprobado por el cliente y/o usuario final
	El cliente ayuda a establecer los escenarios que servirán para probar el SW
	Cuales son los factores del SW que afectan la satisfacción del cliente
1	En que fases del desarrollo del SW está involucrado el cliente
Conocer el nivel de satisfacción del cliente y su participación en el proyecto	
Grupo No 2 Personal conocimientos experiencias	La experiencia del personal asignado al proyecto favorece el desarrollo del SW
	El personal cuenta con experiencia en el dominio de negocio del cliente
	En que medida la rotación del personal afecta el desarrollo del SW
	Es necesario capacitar al personal durante el desarrollo del proyecto
	Se asigna al proyecto de SW el personal necesario para el desarrollo del mismo
	Se cuenta con el apoyo de personas con conocimiento del dominio del negocio
	Los involucrados en el desarrollo del proyecto cuentan con el apoyo de expertos
2	Mejorar la estimación del personal requiendo y la planificación de las capacitaciones
Grupo No 3 Herramientas	Que herramientas se prefiere utilizar durante el desarrollo del proyecto
	Los diferentes entregables principalmente los diseños se desarrollaron utilizando herramientas CASE
	Las herramientas utilizadas permiten mejorar el desempeño del personal
	3
Grupo No 4 Documentación	Se dispone de una adecuada documentación sobre el desarrollo del proyecto
	Se utilizaron plantillas para la documentación del proyecto
	Cuál es el esfuerzo necesario para llevar la documentación del desarrollo del SW
	4

Paso 3: Identificar los Sub-Objetivos

Grupo No. 5 Proceso de desarrollo del SW (Consumibles planeación del proyecto, cronograma presupuesto)	Se han cambiado el tiempo y costo establecidos durante la realización del proyecto
	El SW se construye utilizando algún estándar de codificación
	Cuanto tiempo se emplea en probar el software desarrollado
	El proyecto se planifica usando datos de referencia de proyectos anteriormente realizados
	Se utiliza el tiempo necesario para el buen establecimiento de los requerimientos
	Que proporción del tiempo del proyecto se dedica al diseño
	Se ejecutaron las pruebas suficientes en el SW
	La decisión de liberar el producto está debidamente justificada
	Con que frecuencia se realizan inspecciones
	Se cuenta con el presupuesto adecuado para todas las actividades planificadas
	El tiempo establecido es el adecuado para el desarrollo de SW de calidad
	Cuanto tiempo se dedica para las pruebas del SW
	Las estimaciones de tiempo y costo son apropiadas para desarrollar el proyecto
	El proyecto se desarrolla de acuerdo al cronograma establecido
	Cuanto tiempo toma construir el SW
La instalación del SW se hace en el tiempo planificado	
Cuanto tiempo emplea el personal para realizar reuniones	
Se dan cambios al presupuesto durante el desarrollo del proyecto	
Se conoce la diferencia entre el tiempo planificado y el consumido durante el proyecto	
5	Planificar con mayor precisión el proyecto
Grupo No. 6 Requerimientos	Los requerimientos están adecuadamente definidos al inicio del proyecto
	Cuántos requerimientos están desarrollados y cuántos faltan desarrollar
	Están los requerimientos claros y completamente definidos
	De que forma cambian los requerimientos durante el desarrollo del proyecto
5	Planificar con mayor precisión el proyecto
6	Monitorear los cambios en los requerimientos durante el desarrollo del proyecto

Paso 3: Identificar los Sub-Objetivos

Grupo No. 7 Diseño	Se toman en cuenta patrones de diseño en el desarrollo del proyecto
	Se realizan inspecciones sobre los diseños (Cúen, Frecuencia)
	5
7	Generar diseños confiables
Grupo No. 8 Código fuente	Se utiliza algún estándar para la codificación
	En que proporción se han reutilizado componentes de software en nuevos proyectos
	Se han re-utilizado componentes de SW
5	Planificar con mayor precisión el proyecto
8	Disminuir el tiempo de codificación del SW
Grupo No. 9 Defectos	Se presentan defectos durante la instalación y posterior uso del SW
	Los diseños están libres de defectos
	El código fuente está libre de defectos cuando es liberado
	Qué efectos tienen en la duración del proyecto los defectos detectados en las pruebas
	Cuál es la cantidad de defectos descubiertos y cuanto toma corregirlos
5	Planificar con mayor precisión el proyecto
9	Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final

Paso 3: Identificar los Sub-Objetivos

Sub-Objetivos de Negocio identificados:

- 1.-) Conocer el nivel de satisfacción del cliente y su participación en el proyecto.
- 2.-) Mejorar la estimación del personal requerido y la planificación de las capacitaciones.
- 3.-) Conocer la utilidad de las herramientas para el desarrollo de SW.
- 4.-) Evaluar el impacto de documentar o no documentar los proyectos de software.
- 5.-) Planificar con mejor precisión el proyecto.
- 6.-) Monitorear los cambios en los requerimientos durante el desarrollo del proyecto.
- 7.-) Generar diseños confiables.
- 8.-) Disminuir el tiempo de codificación del SW.
- 9.-) Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final.

Paso 4: Identificar Entidades y Atributos

- Utilizando la lista de preguntas que se refieren a un mismo problema se identifican la entidad implícita en cada pregunta y los atributos relacionados con cada entidad.
- Los atributos son aquellos que si son cuantificados ayudarán a responder la pregunta o establecer un contexto para interpretar las respuestas.

Paso 4: Identificar Entidades y Atributos

Sub-Objetivo No Conocer el nivel de satisfacción del cliente y su participación en el proyecto

Entidad **CLIENTE / USUARIO FINAL**

Atributo 1	Tipo	(Actividad comercial, financiera, industrial, etc.)
Atributo 2	Participación en el proyecto	(Participación del cliente durante el desarrollo del SW)
Atributo 3	Nivel de satisfacción	(Con el SW según los atributos de FURPS y con la gestión del proyecto de SW)

Sub-Objetivo No Mejorar la estimación del personal requerido y la planificación de las capacitaciones

Entidad **PERSONAL**

Atributo 1	Disponibilidad	(Una misma persona pueda trabajar en varios proyectos)
Atributo 2	Composición	(Máx. Nivel de educación y rol del personal asignado al proyecto)
Atributo 3	Conocimiento	(De las herramientas necesarias para el desarrollo del SW)
Atributo 4	Experiencia	(En el dominio de negocio del cliente y/o en desarrollo de proyectos)
Atributo 5	Capacitación	(Durante el desarrollo del proyecto)
Atributo 6	Tamaño	(No. de personas involucradas en el proyecto)
Atributo 7	Esfuerzo	(Horas x Personas estimadas y/o efectivas para la culminación del proyecto)
Atributo 8	Rotación	(Por enfermedad, renuncia o despido)

Sub-Objetivo No Conocer la utilidad de las herramientas para el desarrollo de SW

Entidad **HERRAMIENTAS**

Atributo 1	Tipo	(Sistema Operativo, Lenguaje de programación, Base de datos, etc.)
Atributo 2	Cantidad	(No. de computadoras, impresoras, equipo de comunicación, etc.)
Atributo 3	Utilidad	(Conformidad del personal con las herramientas utilizadas)

Paso 4: Identificar Entidades y Atributos

Sub-Objetivo No Evaluar el impacto de documentar o no documentar los proyectos de software

Entidad **DOCUMENTOS**

Atributo 1	Tipo	(Todos los diferentes documentos producidos durante el proyecto)
Atributo 2	Plantillas	(Desarrolladas por la empresa, de estándares o de otras fuentes (S/N))
Atributo 3	Esfuerzo	(Horas x Persona empleadas en elaborar los diferentes tipos de documentos)
Atributo 4	Volumen	(No. de páginas por tipo de documento)

Sub-Objetivo No Planificar con mejor precisión el proyecto

Entidad **PROCESO DE DESARROLLO**

Atributo 1	Tipo de aplicación de software	(Financiera, Contable, etc.)
Atributo 2	Tipo de proyecto de software	(Nuevo desarrollo, mejora, adaptación, etc.)
Atributo 3	Complejidad	(Estimada técnica y del negocio)
Atributo 4	Metodología o Modelo de desarrollo de SW	(Desarrollada en la empresa o adquirida a terceros)
Atributo 5	Inspecciones	(Reportes de problemas)
Atributo 6	Progreso	(Continuidad, cumplimiento de las actividades y tareas planificadas)
Atributo 7	Tiempo	(Días, semanas o meses calendario)
Atributo 8	Costo	(Estimado vs Real)

Sub-Objetivo No Planificar con mejor precisión el proyecto

Sub-Objetivo No Monitorear los cambios en los requerimientos durante el desarrollo del proyecto

Entidad **REQUERIMIENTOS**

Atributo 1	Cantidad de requerimientos a desarrollar	(Definidos en las fases iniciales del proyecto)
Atributo 2	Cambios durante el desarrollo del proyecto	(Nuevos, borrados, modificados)

Paso 4: Identificar Entidades y Atributos

Sub-Objetivo No Planificar con mejor precisión el proyecto
 Sub-Objetivo No Generar diseños confiables

Entidad DISEÑO

Atributo 1	Tipo	(Orientado a objetos, estructurado, etc.)
Atributo 2	Tamaño	(Número de Clases, Número de Casos de uso)
Atributo 3	Número de inspecciones	(Se acepta, se rechaza o se piden modificaciones)

Sub-Objetivo No Planificar con mejor precisión el proyecto
 Sub-Objetivo No Disminuir el tiempo de codificación del SW

Entidad CÓDIGO FUENTE

Atributo 1	Estándar de codificación	(Definido y utilizado o no)
Atributo 2	Complejidad	(De los métodos, funciones y procedimientos)
Atributo 3	Re-utilización por componentes	(re-utilizados, nuevos, borrados, modificados)

Sub-Objetivo No Planificar con mejor precisión el proyecto
 Sub-Objetivo No Monitorear los cambios en los requerimientos durante el desarrollo del proyecto
 Sub-Objetivo No Generar diseños confiables
 Sub-Objetivo No Disminuir el tiempo de codificación del SW
 Sub-Objetivo No Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final

Entidad DEFECTOS / FALLAS

Atributo 1	Cantidad	(Defectos descubiertos * Inspección y/o en las pruebas)
Atributo 2	Tipo	(origen) (Requerimientos, Diseño, Codificación, etc.)
Atributo 3	Severidad	(Severa, crítica)
Atributo 4	Tiempo en cola	(Tiempo transcurrido desde que se generó el reporte de defectos)
Atributo 5	Tiempo de corrección	(Tiempo que se requiere para corregir los defectos)
Atributo 6	Estatus	(Abierto, cerrado, descartado)
Atributo 7	Fallas	(Defectos que se presentan después de liberar el software)

Paso 5: Formalizar los Objetivos de Medición

- Utilizando los Sub-Objetivos, Entidades y Atributos se generan los Objetivos de Medición utilizando la siguiente estructura:
 - Un Objeto de Interés.
 - Un Propósito.
 - Una Perspectiva.
 - Una descripción del entorno y restricciones.

Paso 5: Formalizar los Objetivos de Medición

- **Objeto de Interés:** Puede ser un producto, proceso, recurso, agente, artefacto, actividad, métrica o entorno. También es una entidad. En resumen es cualquier cosa real o abstracta de la cual se desea conocer más sobre ella.
- **Propósito:** El propósito de una actividad de medición puede ser entender, predecir, planear, controlar, comparar, evaluar o mejorar algún aspecto con respecto a la calidad o productividad de un objeto.

Paso 5: Formalizar los Objetivos de Medición

- **Perspectiva:** Identifica quién está interesado en los resultados de la medición. Es el principal punto de vista de la actividad de medición, tal como: Un desarrollador, un administrador, un cliente, etc.
- **Entorno:** Provee un contexto para interpretar los resultados de medición. Incluye todo lo que afecta o es afectado por el objeto a ser medido.

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 1

Objeto de Interés: Proceso de desarrollo del SW	
Propósito:	Determinar si el nivel de satisfacción del cliente está relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.
Perspectiva:	Examinar la satisfacción del cliente
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	Para evaluar la calidad de producto, se hará referencia al modelo (FURPS) de un producto de software El desarrollo se realiza en unos casos dentro y/o fuera de la empresa de SW

Objetivo de medición No. 2

Objeto de Interés: Proceso de desarrollo del SW	
Propósito:	Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros.
Perspectiva:	Examinar el esfuerzo y tamaño del personal asignado al proyecto
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	Es común que una persona participe en varios proyectos a la vez. El personal puede laborar en el sitio de trabajo del cliente. La rotación es el principal problema con el personal. Se dedica más personal al desarrollo de proyectos que al mantenimiento. Hasta 2 personas trabajan en proy. Pequeños, 6 en medianos y más de 6 en grandes

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 3

Objeto de Interés: Proceso de desarrollo del SW	
Propósito:	Conocer la conformidad del personal con las herramientas utilizadas en el desarrollo del proyecto para evaluar opciones de adquisición y/o cambio de herramientas.
Perspectiva:	Examinar el grado de aceptación de las herramientas utilizadas durante el proyecto
Desde el punto de vista del	Personal asignado al proyecto
Entorno y restricciones	Las empresas constantemente consultan por herramientas que sirvan de apoyo o mejoren la productividad

Objetivo de medición No. 4

Objeto de Interés: Proceso de desarrollo del SW	
Propósito:	Determinar el esfuerzo empleado en realizar la documentación durante el desarrollo del proyecto para estimar el esfuerzo que sería necesario utilizar en una adecuada documentación en proyectos similares.
Perspectiva:	Examinar el esfuerzo en realizar los diferentes tipos de documentos
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	La premura por terminar el proyecto dificulta la documentación de los proyectos. Solo un 40% de las empresas de software siempre documentan el desarrollo de sus proyectos

Objetivo de medición No. 5

Objeto de Interés: Proceso de desarrollo del SW	
Propósito:	Monitorear el avance de las actividades y tareas planificadas para cumplir o reconsiderar los compromisos establecidos al inicio del proyecto.
Perspectiva:	Examinar el cumplimiento de las tareas planificadas, así como su costo y tiempo consumido.
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	El modelo de desarrollo tradicional (cascada) y espiral son los más utilizados. La empresa puede tener certificación ISO 9001:2000, y/o estar trabajando en CMMI nivel 2 o 3

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 6	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto
Perspectiva:	Examinar la el número de cambios en los requerimientos durante el transcurso del proyecto
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	En general se destina la 1/5 parte del tiempo del proyecto a la actividad de especificación de requerimientos En la mayoría de los casos el cliente desconoce el alcance del proyecto Los requerimientos del proyecto usualmente están incompletos Es usual que el cliente realice modificaciones a los requerimientos durante el desarrollo de proyectos

Objetivo de medición No. 7	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Estimar la cantidad adecuada de inspecciones sobre los diseños (actividades, tareas y/o entregables) para disminuir la propagación y severidad de los defectos
Perspectiva:	Examinar los resultados de las inspecciones realizadas sobre los diseños
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	En general se destina un 25% del tiempo del proyecto a la actividad de diseño El tipo de diseño más utilizado es el Orientado a objetos

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 8	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Controlar la complejidad de los componentes del SW para disminuir el esfuerzo utilizado por terceros en comprenderlo y modificarlo
Perspectiva:	Examinar la complejidad de los componentes del SW
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	La actividad de codificación es la que toma más tiempo en ser completada Los objetos son los componentes de SW más reutilizados

Objetivo de medición No. 9	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Conocer la ocurrencia y causa de los defectos durante el desarrollo de los proyectos de SW para identificar acciones de prevención y/o contingencia
Perspectiva:	Examinar el efecto de los defectos sobre el proyecto y la cantidad de fallos que se presentan en el producto final
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	El producto final que se entrega al cliente no está libre de una pequeña cantidad de defectos La fuente principal de defectos es la fase de especificación y requerimientos

Paso 6: Identificar Preguntas e Indicadores cuantificables

- En este paso se generan preguntas cuantificables relacionadas a los objetivos de medición que se desean ser contestadas.
- Se definen los indicadores que contestarán dichas preguntas.
- Cuando se identifican preguntas y definen indicadores es importante tener en cuenta el objetivo al cual se dirigen. Ej: En entornos cambiantes es menos útil precisar números y hacer análisis estadísticos.

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 1	Determinar el nivel de satisfacción del cliente este relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.
Pregunta No. 1	Cuál es el nivel de satisfacción del cliente y/o usuario con el producto de software desarrollado?
Pregunta No. 2	Cuál es la percepción del cliente con la metodología de desarrollo de proyectos de la empresa?
Pregunta No. 3	En que actividades durante el desarrollo del proyecto participó el cliente y/o usuario final?
Nombre del indicador	Clientes Satisfechos v. Porcentaje de horas de participación en el proyecto
Método de cálculo	1) Encuesta con escala de medición Likert
Ambito	Aplicado a Clientes y usuarios al término del proyecto

Objetivo de medición No. 3	Conocer la conformidad del personal con las herramientas utilizadas en el desarrollo del proyecto para evaluar opciones de adquisición y/o cambio de herramientas.
Pregunta No. 1	Se dispone la infraestructura (hardware y software) adecuada para un desarrollo sin contratiempos?
Pregunta No. 2	Cuál es el nivel de conformidad del personal respecto a la utilidad de las herramientas utilizadas?
Nombre del indicador	Conformidad del personal con las herramientas utilizadas
Método de cálculo	1) Encuesta con escala de medición Likert
Ambito	Aplicado al personal del área de sistemas al término del proyecto

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 2	Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros.	
Pregunta No. 1	Se dispone del personal adecuado para el desarrollo del proyecto?	
Pregunta No. 2	El esfuerzo requerido al personal asignado al proyecto es adecuado con la carga de trabajo que tienen actualmente?	
Nombre del Indicador	Módulo de cálculo	Ámbito
Error en la estimación del esfuerzo	1) $\text{EsfuerzoReal} = [(1+\text{Error}) * \text{EsfuerzoEstimado}]$	Calculado al final del proyecto
Disponibilidad del personal	2) $\text{HorasTrabajadas} / \text{Horas estimadas}$	Calculado de forma semanal para todo el personal

Objetivo de medición No. 4	Determinar el esfuerzo empleado en realizar la documentación durante el desarrollo del proyecto para estimar el esfuerzo que sería necesario utilizar en una adecuada documentación en proyectos similares.	
Pregunta No. 1	Qué documentos se generaron durante el desarrollo del proyecto?	
Pregunta No. 2	Cuál es el esfuerzo utilizado en la documentación del proyecto?	
Nombre del Indicador	Fracción del tiempo del proyecto empleado en documentación	
Módulo de cálculo	1) $\text{Horas} \times \text{Persona de documentación} / \text{Total Horas} \times \text{Persona}$	
Ámbito	Calculado al término del proyecto de software	

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 5	Monitorar el avance de las actividades y tareas planificadas para cumplir o reconsiderar los compromisos establecidos al inicio del proyecto.	
Pregunta No. 1	Las actividades y tareas se planificaron teniendo en cuenta la complejidad del proyecto?	
Pregunta No. 2	Las actividades y tareas planificadas para el proyecto se cumplen según el cronograma establecido?	
Pregunta No. 3	Las actividades y tareas se completan sin excederse del presupuesto establecido al inicio del proyecto?	
Nombre del Indicador	Módulo de cálculo	Ámbito
Complejidad estimada	1) Evaluación con escala de 5 opciones y 2 referencias	Inicio del proyecto de software
Avance del proyecto	2) Variación entre fechas de inicio y fin de tareas del proyecto	Durante el proyecto
Índice de rendimiento de Tiempo SPI	3) EVMS	Durante el proyecto
Índice de rendimiento de costo CPI	4) EVMS	Durante el proyecto

Objetivo de medición No. 7	Estimar la cantidad adecuada de inspecciones sobre los diseños (actividades, tareas y/o entregables) para disminuir la propagación y severidad de los defectos.	
Pregunta No. 1	Cuántas inspecciones (revisiones) se efectúan por actividad (no solo de diseño) y/o entregables del proyecto?	
Pregunta No. 2	Qué diferencia se aprecia entre proyectos que tienen inspecciones regulares y aquellos proyectos con pocas inspecciones?	
Nombre del Indicador	Inspecciones por actividad o entregable generado	
Módulo de cálculo	1) Histograma de inspecciones por actividad o entregable del proyecto	
Ámbito	Durante el proyecto	

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 6	Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto
Pregunta No. 1	El cliente define concisa y completamente todos sus requerimientos en las fases iniciales o de especificación del proyecto?
Pregunta No. 2	Los requerimientos del proyecto se mantienen estables (sin cambios) durante el desarrollo del mismo?
Nombre del indicador	Tasa de variación en los requerimientos del software
Método de cálculo	1) $TasaVariación = \frac{CambiosEnReq.}{(Requiciales + CambiosEnReq)}$
Ámbito	Calculado a la terminación del proyecto de software

Objetivo de medición No. 8	Controlar la complejidad de los componentes del SW para disminuir el esfuerzo utilizado por terceros en comprenderlo y modificarlo				
Pregunta No. 1	Cuál es la complejidad estructural de los métodos y procedimientos del software?				
Pregunta No. 2	Cuánto tiempo necesita el personal inexperto en comprender los componentes de SW existentes?				
Pregunta No. 3	Cuál es el tamaño (expresado en líneas de código) de los componentes del software?				
Nombre del indicador	Complejidad ciclométrica o estimada	Método de cálculo	1) Estadísticos de complejidad de cada componente de software	Ámbito	Todos los componentes de software que disponga la empresa
	Tamaño de los componentes de software		2) Graf. Dispersión Complejidad v.s. Horas de aprendizaje		
			3) Líneas de código x componente de software		

Objetivo de medición No. 9	Conocer la ocurrencia y causa de los defectos durante el desarrollo de los proyectos de SW para identificar acciones de prevención y/o contingencia			
Pregunta No. 1	De qué forma se distribuyen los defectos durante el ciclo de desarrollo del proyecto?			
Pregunta No. 2	Qué clases de defectos son los más recurrentes?			
Pregunta No. 3	Qué fracción del tiempo del proyecto se utiliza para corregir defectos?			
Pregunta No. 4	Cuánto eficiencia es la empresa en corregir defectos en el software antes de entregarlo al cliente?			
Nombre del indicador	Defectos v.s. Fallos			
Método de cálculo		1) Fuga de defectos = $Fallos / (Fallos + Defectos)$		
		2) Histograma de ocurrencia de defectos		
		3) Análisis Causa-Efecto		
Ámbito	Durante el tiempo de desarrollo y posterior puesta en producción del software			

Paso 7: Identificar los Elementos de Datos

- Se listan los datos que son necesarios obtener para formar los indicadores.
- Algunos datos obtenidos pueden ayudar a formar más de un indicador.

Cod	DATOS	Definición del Código y Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	Formas de APLICAR la Utilización del Número y A.	FUENTE
57	Efectos															FORMA P02
58	Fecha Mes															FORMA P03
59	Líneas de código															FORMA P03
60	Componente, método o procedimiento				X	X	X									FORMA P04
61	Horas de desarrollo del componente													X		FORMA P04
62	Número de defectos													X		FORMA P04
63	Complejidad ciclométrica				X	X										FORMA P04
64	Densidad de aprendizaje						X									FORMA P04

Paso 8: Definir las Mediciones

- Los nombres de las mediciones no son suficientes. Se tiene que poder decirle a otras personas exactamente como cada medición debe ser obtenida, y de esta manera poder ser interpretadas correctamente.
- Para lograr esto se obtienen las definiciones operacionales de cada medición.
- Definiciones Operacionales:** Le dicen a los usuarios cómo los datos deben ser recolectados. Cuando los usuarios no conocen la manera de recolectarlos, fácilmente se puede llegar a asunciones inválidas.

Paso 8: Definir las Mediciones

Definition Checklist for Source Statement Counts

Definition name: Physical Source Lines of Code Date: 11/7/82
 (Metric definition) Originator: NI

Measurement unit:	Logical source statements	Physical source lines		
Statement type	Definition <input checked="" type="checkbox"/> Data array <input type="checkbox"/>		Includes	Excludes
Lines of the program contain more than one type classified as the type with the highest precedence Order of precedence ->				
1	Executable			
2	Non-executable			
3	Declarations			
4	Control structures			
5	Comments			
6	On-line code			
7	On-line with source code			
8	Barren and non-barren spaces			
9	Blank empty comments			
10	Blank lines			
11				
12				
New product	Definition <input checked="" type="checkbox"/> Data array <input type="checkbox"/>		Includes	Excludes
1	Programmed			
2	Generated with source code generators			
3	Compatible with previous versions			
4	Created or re-created without change			
5	Modified			
6	Removed			
7				
Origin	Definition <input checked="" type="checkbox"/> Data array <input type="checkbox"/>		Includes	Excludes
1	New work, no prior existence			
2	Work taken or adapted from			
3	Acquired without build or release			
4	Commercial off-the-shelf software (COTS); store-bought items			
5	Government furnished software (GFS); other than lease or lease			
6	Another product			
7	A vendor-supplied or leased source entry (unmodified)			
8	A vendor-supplied operating system or utility (unmodified)			
9	A lock or modified drug support library or coding system			
10	Other commercial library			
11	Archive library; software designed for reuse			
12	Other software composed of library			
13				
14				
Usage	Definition <input checked="" type="checkbox"/> Data array <input type="checkbox"/>		Includes	Excludes
1	1 - or as part of the primary product			
2	2 - as a tool or a subset of the primary product			

Figure 4-5: A Checklist-Based Definition for Source Lines of Code

Paso 9: Identificar las acciones necesarias para implementar las mediciones

- En este paso se añade información acerca del estado actual y el uso de las mediciones, de esta manera se puede preparar un plan efectivo para implementar las mismas. Existen 3 pasos:
 - Análisis
 - Diagnóstico
 - Acción.

Paso 9: Identificar las acciones necesarias para implementar las mediciones

- **Análisis:** Identificar las mediciones que la organización ya está utilizando y entender cómo están siendo recolectadas.
- **Diagnóstico:** Evaluar los elementos de datos que la organización ya está recolectando. Conocer también la disponibilidad de las mediciones propuestas.
- **Acción:** Empieza con identificar los elementos que se construirán en el plan de medición.
 - Identificar las fuentes de datos dentro del proceso de software existente.
 - Definir los métodos que serán usados para recolectar y reportar los datos.
 - Identificar las herramientas que serán requeridas para ayudar a la recolección, reporte y almacenamiento de los datos.
 - Entre otros.

Paso 10: Preparar un Plan

- Se prepara un Plan para implementar las acciones que han sido identificadas.
- La estructura del Plan es la siguiente:
 - 1.-) Objetivo
 - 2.-) Descripción
 - *Objetivos
 - *Ámbito
 - *Relaciones con otros procesos de mejora de software.
 - 3.-) Implementación
 - *Actividades, Productos y Tareas.
 - *Calendario.
 - *Recursos.
 - *Responsabilidades.
 - *Medición y Monitoreo.
 - *Supuestos.
 - 4.-) Operación Sostenida.

[Referencias



- Goal-Driven Software Measurement – A Guidebook , Robert E. Parl, Wolfhart B. Goethert, William A. Florac , 1996.
- “Elaboración y análisis de métricas para el proceso de desarrollo de software para empresas desarrolladoras de software del ecuador ” - Plan piloto - Tesis ESPOL, Jorge H. Mazón, José L. Alvear,2006.

ANEXO 2: Plan de métricas y Plantillas de levantamiento de información (Plan Piloto)

REPORTE DE PROBLEMAS O CAMBIOS EN REQUERIMIENTOS (F02)

PROYECTO
1

Nro. REPORTE
40

FASE
22

FECHA
41

ESTADO	
En cola	42
Abierto	
Cerrado	
Descartado	

TIPO	
Defecto	43
Falla	
Otro	

SEVERIDAD	
Leve	44
Grave	
Crítica	

ESFUERZO	
Estimado	45
Real	46

REQUERIMIENTO	
Agregado	47
Modificado	
Eliminado	
<i>Funcional</i>	48
<i>No Funcional</i>	

METODO QUE DETECTO EL PROBLEMA

Inspección	49	Caja Negra		Otro		Observaciones: 50
Peer review		Caja Blanca		Otro		
P. Stress		Simulación		Otro		

DESCRIPCION DEL PROBLEMA
51

COMPONENTE(s) AFECTADO
52

POR QUÉ SE PRODUJO? (CAUSAS)
53

IMPACTO EN EL PROYECTO? (EFECTOS)
54

Cod.	DATOS	INDICADORES											FUENTE		
		Satisfacción del Cliente / Usuario	Error en la estimación del esfuerzo	Disponibilidad del personal	Esfuerzo empleado en la documentación	Complejidad	Avance del proyecto	Índice de rendimiento	Índice de rendimiento de costo	Inspecciones durante el proyecto	Variación en los requerimientos	Tamaño de los componentes de SW		Diagrama Causa-efecto	
1	Proyecto														FORMA F00
2	Requerimientos									X					FORMA F00
3	Aplicación de software														FORMA F00
4	Proyecto de software			X											FORMA F00
5	Satisfacción cliente	X													FORMA F00, ENCUESTA CLIENTE
6	Satisfacción usuario	X													FORMA F00, ENCUESTA USUARIO
7	Fecha inicio	X				X									FORMA F00, PROJECT
8	Fecha fin	X				X									FORMA F00, PROJECT
9	Personal estimado			X											FORMA F00, PROJECT
10	Personal real			X											FORMA F00, PROJECT
11	Costo estimado								X						FORMA F00, EXCEL
12	Costo real								X						FORMA F00, EXCEL
13	Complejidad negocio					X									FORMA F00, ENCUESTA COMP. T
14	Complejidad técnica					X									FORMA F00, ENCUESTA COMP. N
15	Rol														FORMA F00
16	Nombre														FORMA F00
17	Experiencia desarrollo														FORMA F00
18	Experiencia negocio														FORMA F00
19	Rotacion														FORMA F00
20	Tiempo														FORMA F00
21	Causa														FORMA F00
22	Fase								X				X		FORMA F01
23	Número														FORMA F01
24	Tarea / Entregable					X			X						FORMA F01, PROJECT
25	Responsable														FORMA F01
26	Páginas			X											FORMA F01
27	Documentación			X											FORMA F01
28	Descripción														FORMA F01
29	Criterio terminación														FORMA F01
30	Fecha estimada inicio	X	X			X	X	X							FORMA F01, PROJECT
31	Fecha estimada fin	X	X			X	X	X							FORMA F01, PROJECT
32	Fecha real inicio	X				X	X	X							FORMA F01, PROJECT
33	Fecha real fin	X				X	X	X							FORMA F01, PROJECT
34	Duración estimada														FORMA F01
35	Duración real														FORMA F01
36	Costo estimado							X	X						FORMA F01, EXCEL
37	Costo real							X	X						FORMA F01, EXCEL
38	Esfuerzo estimado	X	X												FORMA F01, PROJECT
39	Esfuerzo real	X		X											FORMA F01, PROJECT
40	Nro. reporte									X				X	FORMA F02
41	Fecha									X				X	FORMA F02
42	Estado													X	FORMA F02
43	Tipo													X	FORMA F02
44	Severidad													X	FORMA F02
45	Esfuerzo estimado													X	FORMA F02
46	Esfuerzo real													X	FORMA F02
47	Requerimiento									X					FORMA F02
48	Funcional / No funcional									X					FORMA F02
49	Metodo que detecto el problema													X	FORMA F02
50	Observaciones													X	FORMA F02
51	Descripción del problema													X	FORMA F02
52	Componente afectado													X	FORMA F02
53	Causas													X	FORMA F02
54	Efectos													X	FORMA F02
55	Mes														FORMA F03
56	Líneas de código												X		FORMA F03
57	Componente, Método o Procedimiento									X	X	X			FORMA F04
58	Complejidad ciclométrica / estimada									X	X		X		FORMA F04
59	Horas necesarias para comprenderlo										X				FORMA F04

**ANEXO 3: Plan de métricas y Plantillas de
levantamiento de información (Desarrollado)**

DEFINICION DE TAREAS / ENTREGABLES (F01)

PROYECTO		FASE	
1		25	

NUMERO	TAREA / ENTREGABLE	TIPO DE TAREA	
26	27	28	

RESPONSABLE		PAGINAS	DOCUMENTACIÓN
29		30	31

DESCRIPCION		CRITERIO DE TERMINACIÓN	
32		33	

FECHA ESTIMADA		FECHA REAL		DURACIÓN		COSTO	
Inicio	Fin	Inicio	Fin	Estimada	Real	Estimado	Real
34	35	36	37	38	39	40	41

RECURSOS	
Rol	Nombre
17	18
TOTAL	

ESFUERZO	
Estimado	Real
42	43
50	51

REPORTE DE PROBLEMAS O CAMBIOS EN REQUERIMIENTOS (F02)

PROYECTO
1

Número de REPORTE
44

FASE
25

FECHA
36

ESTADO	
En cola	45
Abierto	
Fecha abierto	46
Cerrado	
Descartado	

TIPO	
Defecto	28
Falla	
Otro	

SEVERIDAD	
Leve	47
Grave	
Crítica	

ESFUERZO	
Estimado	48
Real	49

REQUERIMIENTO	
Agregado	50
Modificado	
Eliminado	
<i>Funcional</i>	51
<i>No Funcional</i>	

METODO QUE DETECTO EL PROBLEMA

Inspección	52	Caja Negra		Cliente		Observaciones: 53
Peer review		Caja Blanca		Otro		
P. Stress		Simulación		Otro		

DESCRIPCION DEL PROBLEMA
54
Qué?
Cuándo?
Dónde?
Cómo?

COMPONENTE(s) AFECTADO
55
Proyecto
Sistema
Modulo
Clase / Método

POR QUÉ SE PRODUJO? (CAUSAS)
56
Infraestructura
Herramientas SW
Políticas / Procedimientos
Personal

IMPACTO EN EL PROYECTO? (EFECTOS)
57
Strees del cliente o personal
Retrasos
Aumento costos
Retrabajo

ANEXO 4: Plan de Capacitación (Última versión)

Plan de Capacitación vs.0

Proyecto: Aplicación de Métricas para Empresas Desarrolladoras de Software.



¿Por qué medir?

- Existen 4 razones para medir los procesos de software, productos y recursos:
 - Para caracterizar.
 - Para evaluar.
 - Para predecir.
 - Para mejorar.
- Nosotros **caracterizamos** para aumentar el entendimiento de los procesos, productos, recursos y entornos, y para establecer el punto de partida para comparaciones en futuras mediciones.
- **Evaluamos** para determinar el estado respecto a planes. Las mediciones son los sensores que nos permiten conocer cuando los proyectos y procesos están fuera del camino y poder tenerlos bajo control.

[¿Por qué medir?]

- **Predecimos** para poder planear. Medir para poder predecir involucra aumentar el entendimiento de las relaciones entre los procesos y productos, y la construcción de modelos para dichas relaciones, para que los valores observados en algunos atributos puedan ser usados para predecir otros. Mediciones predecibles son la base para encontrar tendencias, estimar costos, tiempo e incrementar la calidad basados en la evidencia actual.
- Medimos para **Mejorar** la obtención de información cuantitativa que nos ayude a identificar causas de falla, ineficiencias y otras oportunidades para mejorar la calidad del producto de software y el proceso para desarrollarlo.

[Descripción del Proyecto]

- Los resultados obtenidos del Formulario aplicado a Administradores de Proyectos de Software entre los meses de Marzo y Mayo del 2006 revelaron que las empresas desarrolladoras de software estiman la mayoría de los datos de los principales factores del Proceso de Desarrollo de Software.
- La mayoría de las respuestas no provenían de datos reales y documentados.

ENCUESTA SOBRE EL PROCESO DE DESARROLLO DE SOFTWARE

1. ¿Qué tipo de empresa desarrolla software?

A. Pymes B. Mediana C. Otro

2. ¿El proyecto que maneja es?

A. Desarrollo de software B. Mantenimiento y mejora de software C. Otro

3. ¿Cuál es el tamaño del proyecto?

A. Menos de 10 personas B. Entre 10 y 20 personas C. Más de 20 personas

4. ¿Cuál es el tipo de proyecto?

A. Desarrollo de software B. Mantenimiento y mejora de software C. Otro

5. ¿Cuál es el tipo de proyecto?

A. Desarrollo de software B. Mantenimiento y mejora de software C. Otro

6. ¿Cuál es el tipo de proyecto?

A. Desarrollo de software B. Mantenimiento y mejora de software C. Otro

7. ¿Cuál es el tipo de proyecto?

A. Desarrollo de software B. Mantenimiento y mejora de software C. Otro

ORIGEN DE LA DATA: 1) La respuesta proporciona evidencia de datos reales y documentados 2) Son estimaciones de personal involucrado en el proyecto 3) La respuesta no aplica al proyecto

[Descripción del Proyecto]

- Es por eso que se ha elaborado un Instrumento de Medición (Recolección de Datos) que permita obtener datos reales, del día a día, de las empresas.
- Este Instrumento será aplicado a un proyecto que la empresa esté desarrollando o que esté próximo a realizar.
- Los datos recolectados por medio de este Instrumento fueron obtenidos a través de un Proceso de Medición dirigido a objetivos denominado GQIM, el mismo que garantiza que los datos recolectados ayudan a cumplir los Objetivos de la Empresa.
- Dicho proceso se describe a continuación.

[Proceso de medición dirigido a objetivos]



[Proceso de Medición dirigido a objetivos]

- Es un Proceso creado por el SEI (Software Engineering Institute) en Carnegie Mellon University en Pittsburgh.
- Está enmarcado dentro del área de proceso de Medición y Análisis en el Nivel 2 del CMMI (Capability Maturity Model Integration).

[Proceso de Medición dirigido a objetivos]

- El objetivo de este proceso es obtener información que ayude a lograr los objetivos de negocio de las empresas desarrolladoras de software:
 - Maximizar la **Satisfacción** del Cliente.
 - Mejorar la **Calidad** del Producto de Software.
 - Mejorar la **Productividad**.
- Y además mantener un seguimiento desde las mediciones específicas hacia los objetivos de negocio, para que los esfuerzos de medición no sean en vano.

Proceso de Medición dirigido a objetivos

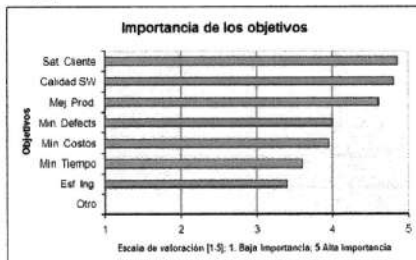
- Consta de 10 pasos:

- 1.-) Identificar los Objetivos del Negocio.
- 2.-) Identificar qué se desea saber o aprender.
- 3.-) Identificar los Sub-Objetivos.
- 4.-) Identificar las entidades y atributos relacionadas a los Sub-Objetivos.
- 5.-) Formalizar los Objetivos de Medición.
- 6.-) Identificar preguntas cuantificables y sus indicadores relacionados que se usarán para ayudar a conseguir los objetivos de medición.
- 7.-) Identificar los elementos de datos que se recolectarán para construir los indicadores que ayudarán a responder las preguntas.
- 8.-) Definir las mediciones a ser usadas, y convertir esas definiciones de manera operacional.
- 9.-) Identificar las acciones que se tomarán para implementar las mediciones.
- 10.-) Preparar un plan para implementar las mediciones.

Paso 1: Identificar los objetivos de negocio

- Es el primer paso para identificar y definir mediciones específicas del software y consiste en identificar los objetivos de negocio en los cuales están basados los esfuerzos de la organización.
- Este paso debe dar inicio en cualquier nivel organizacional donde los objetivos puedan ser razonablemente identificados.
- Iniciarlo en los niveles altos tiene la ventaja de asegurar el seguimiento de las mediciones resultantes hacia los objetivos de negocio primarios, lo que no se tiene en niveles bajos.

Paso 1: Identificar los objetivos de negocio



- En una encuesta aplicada a Gerentes y Administradores de Empresas Desarrolladoras de Software se obtuvieron los siguientes 3 Objetivos de Negocio:
 - Maximizar la **Satisfacción** del Cliente.
 - Mejorar la **Calidad** del Producto de Software.
 - Mejorar la **Productividad**.

Paso 2: Identificar qué se desea saber o aprender

- Consiste en identificar que es necesario conocer para entender, medir, predecir o mejorar las actividades necesarias para lograr los objetivos.
- La herramienta recomendada para identificar estas preguntas es la lista de entidades y preguntas, la cual tiene los siguientes pasos:
 - 1.-) Seleccionar un objetivo de negocio.
 - 2.-) Identificar las personas o grupos involucrados con este objetivo.
 - 3.-) Por cada rol de las personas involucradas identificar los procesos que manejan o afectan.

Paso 2: Identificar qué se desea saber o aprender

- 4.-) Listar las cosas importantes (entidades) que son manejadas dentro de cada proceso por cada rol. Las entidades a ser obtenidas se encuentran dentro de los siguientes procesos:
- *Entradas y recursos.
 - *Productos y derivados.
 - *Artefactos internos tales como inventario y trabajo en proceso.
 - *Actividades y Flujo.
- 5.-) Por cada entidad, listar las preguntas que si fueran contestadas ayudarían a dicha persona en su rol a conseguir el objetivo de negocio seleccionado. Ej:
- *¿Qué tan grande es?
 - *¿Cuán rápido se lo realiza?
 - *¿Cuántos componentes tiene?
- 6.-) Revisar por cada uno de los procesos si no falta nada.
- 7.-) Repetir las tareas 1 al 6 para cada uno de los objetivos de negocio.

Paso 2: Identificar qué se desea saber o aprender

Entradas y Recursos	Preguntas relacionadas a los objetivos de negocio
Ciente	Están nuestros clientes satisfechos con el SW que producimos Cuales son los factores del SW que afectan la satisfacción del cliente En que fases del desarrollo del SW está involucrado el cliente
Personal	La experiencia del personal asignado al proyecto favorece el desarrollo del SW En que medida, la rotación del personal afecta el desarrollo del SW Se asigna al proyecto de SW el personal necesario para el desarrollo del mismo
Consumibles	Se han cambiado el tiempo y costo establecidos durante la realización del proyecto Las estimaciones de tiempo y costo son apropiadas para desarrollar el proyecto
Herramientas	Que herramientas se prefiere utilizar durante el desarrollo del proyecto Los diferentes entregables principalmente los diseños se desarrollaron utilizando herramientas CASE Las herramientas utilizadas permiten mejorar el desempeño del personal

Paso 2: Identificar qué se desea saber o aprender

Actividades y flujos	
Planeación del proyecto	El cliente participó en el desarrollo de los proyectos
	El proyecto se planea usando datos de referencia de proyectos anteriormente realizados
Especificación de requerimientos	El proyecto se desarrolla de acuerdo al cronograma establecido
	Los requerimientos están adecuadamente definidos al inicio del proyecto
	Se utiliza al tiempo necesario para el buen establecimiento de los requerimientos
Diseñar	Se cuenta con el apoyo de personas con conocimiento del dominio del negocio
	Los diseños son revisados y aprobados por el cliente
	Que proporción del tiempo del proyecto se dedica al diseño
Construir	Se toman en cuenta patrones de diseño en el desarrollo del proyecto
	El SW se construye utilizando algún estándar de codificación
	En qué proporción se han reutilizado componentes de software en nuevos proyectos
Probar	Cuanto tiempo toma construir el SW
	Cuanto tiempo se emplea en probar el software desarrollado
	Se ejecutaron las pruebas suficientes en el SW
Instalar	Qué efecto tienen en la duración del proyecto los defectos detectados en las pruebas
	Se presentan defectos durante la instalación y posterior uso del SW
	La decisión de liberar el producto está debidamente planificada
Inspecciones y Reuniones	La instalación del SW se hace en el tiempo planificado
	Con qué frecuencia el cliente participa en reuniones
	Con qué frecuencia se realizan inspecciones
	Cuanto tiempo emplea el personal para realizar reuniones

Paso 2: Identificar qué se desea saber o aprender

Artefactos Internos	
Presupuesto	Se cuenta con el presupuesto adecuado para todas las actividades planificadas
	Se dan cambios al presupuesto durante el desarrollo del proyecto
Cronograma	El tiempo establecido es el adecuado para el desarrollo de SW de calidad
	Se conoce la diferencia entre el tiempo planificado y el consumido durante el proyecto
Productos y derivados	
Requerimientos	Cuántos requerimientos están desarrollados y cuántos faltan desarrollar
	Están los requerimientos claros y completamente definidos
	De qué forma cambian los requerimientos durante el desarrollo del proyecto
Diseño del SW	El diseño del SW está aprobado por el cliente y/o usuario final
	Los diseños están libres de defectos
	Se realizan inspecciones sobre los diseños (Quién, Frecuencia)
Código fuente	Se utiliza algún estándar para la codificación
	El código fuente está libre de defectos cuando es liberado
	Se han re-utilizado componentes de SW
Pruebas	El cliente ayuda a establecer los escenarios que servirán para probar el SW
	Cuanto tiempo se dedica para las pruebas del SW
	Cuál es la cantidad de defectos descubiertos y cuánto toma corregirlos
Documentación	Se dispone de una adecuada documentación sobre el desarrollo del proyecto
	Se utilizarán plantillas para la documentación del proyecto
	Cuál es el esfuerzo necesario para llevar la documentación del desarrollo del SW
Conocimientos, Experiencias	El personal cuenta con experiencia en el dominio de negocio del cliente
	Es necesario capacitar al personal durante el desarrollo del proyecto
	Los involucrados en el desarrollo del proyecto cuentan con el apoyo de expertos

Paso 3: Identificar los Sub-Objetivos

- Consiste en transformar los objetivos de negocio de alto nivel en Sub-Objetivos que se relacionen específicamente con actividades que se manejan o desarrollan.
- Se agrupan las preguntas que se refieran a problemas similares. Estos problemas son potenciales Sub-Objetivos.

Paso 3: Identificar los Sub-Objetivos

Grupo No. 1 Clientes	Estan nuestros clientes satisfechos con el SW que producimos
	El cliente participó en el desarrollo de los proyectos
	Los diseños son revisados y aprobados por el cliente
	Con qué frecuencia el cliente participa en reuniones
	El diseño del SW está aprobado por el cliente y/o usuario final
	El cliente ayuda a establecer los escenarios que se van a probar el SW
	Cuales son los factores del SW que afectan la satisfacción del cliente
	En que fases del desarrollo del SW está involucrado el cliente
1	Conocer el nivel de satisfacción del cliente y su participación en el proyecto
Grupo No. 2 Personal conocimientos experiencias	La experiencia del personal asignado al proyecto favorece el desarrollo del SW
	El personal cuenta con experiencia en el dominio de negocio del cliente
	En que medida la rotación del personal afecta el desarrollo del SW
	Es necesario capacitar al personal durante el desarrollo del proyecto
	Se asigna al proyecto de SW el personal necesario para el desarrollo del mismo
	Se cuenta con el apoyo de personas con conocimiento del dominio del negocio
Los involucrados en el desarrollo del proyecto cuentan con el apoyo de expertos	
2	Mejorar la estimación del personal requerido y la planificación de las capacitaciones
Grupo No. 3 Herramientas	Que herramientas se prefiere utilizar durante el desarrollo del proyecto
	Los diferentes entregables principalmente los diseños se desarrollaron utilizando herramientas CASE
	Las herramientas utilizadas permiten mejorar el desempeño del personal
3	Conocer la utilidad de las herramientas para el desarrollo de SW
Grupo No. 4 Documentación	Se dispone de una adecuada documentación sobre el desarrollo del proyecto
	Se utilizaron plantillas para la documentación del proyecto
	Cual es el esfuerzo necesario para llevar la documentación del desarrollo del SW
	Cual es el impacto de documentar o no documentar los proyectos de software
4	Evaluar el impacto de documentar o no documentar los proyectos de software

Paso 3: Identificar los Sub-Objetivos

Grupo No. 5 Proceso de desarrollo del SW (Consumibles, planeación del proyecto, cronograma, presupuesto)	Se han cambiado el tiempo y costo establecidos durante la realización del proyecto
	El SW se construye utilizando algún estándar de codificación
	Cuanto tiempo se emplea en probar el software desarrollado
	El proyecto se planifica usando datos de referencia de proyectos anteriormente realizados
	Se utiliza el tiempo necesario para el buen establecimiento de los requerimientos
	Que proporción del tiempo del proyecto se dedica al diseño
	Se ejecutaron las pruebas suficientes en el SW
	La decisión de liberar el producto está debidamente planificada
	Con que frecuencia se realizan inspecciones
	Se cuenta con el presupuesto adecuado para todas las actividades planificadas
	El tiempo establecido es el adecuado para el desarrollo de SW de calidad
	Cuanto tiempo se dedica para las pruebas del SW
	Las estimaciones de tiempo y costo son apropiadas para desarrollar el proyecto
	El proyecto se desarrolla de acuerdo al cronograma establecido
	Cuanto tiempo toma construir el SW
La instalación del SW se hace en el tiempo planificado	
Cuanto tiempo emplea el personal para realizar reuniones	
Se dan cambios al presupuesto durante el desarrollo del proyecto	
Se conoce la diferencia entre el tiempo planificado y el consumido durante el proyecto	
5	Planificar con mejor precisión el proyecto
Grupo No. 6 Requerimientos	Los requerimientos están adecuadamente definidos al inicio del proyecto
	Cuántos requerimientos están desarrollados y cuántos faltan desarrollar
	Están los requerimientos clara y completamente definidos
	De que forma cambian los requerimientos durante el desarrollo del proyecto
5	Planificar con mejor precisión el proyecto
6	Monitorear los cambios en los requerimientos durante el desarrollo del proyecto

Paso 3: Identificar los Sub-Objetivos

Grupo No. 7 Diseño	Se toman en cuenta patrones de diseño en el desarrollo del proyecto
	Se realizan inspecciones sobre los diseños (Quién Frecuencia)
	5
7	Generar diseños confiables
Grupo No. 8 Código fuente	Se utiliza algún estándar para la codificación
	En que proporción se han reutilizado componentes de software en nuevos proyectos
	Se han re-utilizado componentes de SW
5	Planificar con mejor precisión el proyecto
8	Disminuir el tiempo de codificación del SW
Grupo No. 9 Defectos	Se presentan defectos durante la instalación y posterior uso del SW
	Los diseños están libres de defectos
	El código fuente está libre de defectos cuando es liberado
	Qué efecto tienen en la duración del proyecto los defectos detectados en las pruebas
5	Planificar con mejor precisión el proyecto
9	Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final

Paso 3: Identificar los Sub-Objetivos

Sub-Objetivos de Negocio identificados:

- 1.-) Conocer el nivel de satisfacción del cliente y su participación en el proyecto.
- 2.-) Mejorar la estimación del personal requerido y la planificación de las capacitaciones.
- 3.-) Conocer la utilidad de las herramientas para el desarrollo de SW.
- 4.-) Evaluar el impacto de documentar o no documentar los proyectos de software.
- 5.-) Planificar con mejor precisión el proyecto.
- 6.-) Monitorear los cambios en los requerimientos durante el desarrollo del proyecto.
- 7.-) Generar diseños confiables.
- 8.-) Disminuir el tiempo de codificación del SW.
- 9.-) Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final.

Paso 4: Identificar Entidades y Atributos

- Utilizando la lista de preguntas que se refieren a un mismo problema se identifican la entidad implícita en cada pregunta y los atributos relacionados con cada entidad.
- Los atributos son aquellos que si son cuantificados ayudarán a responder la pregunta o establecer un contexto para interpretar las respuestas.

Paso 4: Identificar Entidades y Atributos

Sub-Objetivo No Conocer el nivel de satisfacción del cliente y su participación en el proyecto

Entidad	CLIENTE / USUARIO FINAL	
Atributo 1	Tipo	(Actividad comercial, financiera, industrial, etc.)
Atributo 2	Participación en el proyecto	(Participación del cliente durante el desarrollo del SW)
Atributo 3	Nivel de satisfacción	(Con el SW según los atributos de FURPS y con la gestión del proyecto de SW)

Sub-Objetivo No Mejorar la estimación del personal requerido y la planificación de las capacitaciones

Entidad	PERSONAL	
Atributo 1	Disponibilidad	(Una misma persona puede trabajar en varios proyectos)
Atributo 2	Composición	(Max. Nivel de educación y rol del personal asignado al proyecto)
Atributo 3	Conocimiento	(De las herramientas necesarias para el desarrollo del SW)
Atributo 4	Experiencia	(En el dominio de negocio del cliente y/o en desarrollo de proyectos)
Atributo 5	Capacitación	(Durante el desarrollo del proyecto)
Atributo 6	Tamaño	(Nro. de personas involucradas en el proyecto)
Atributo 7	Esfuerzo	(Horas x Personas estimadas y/o efectivas para la culminación del proyecto)
Atributo 8	Rotación	(Por enfermedad, renuncia o despido)

Sub-Objetivo No Conocer la utilidad de las herramientas para el desarrollo de SW

Entidad	HERRAMIENTAS	
Atributo 1	Tipo	(Sistema Operativo, Lenguaje de programación, Base de datos, etc.)
Atributo 2	Cantidad	(Nro. de computadoras, impresoras, equipos de comunicación, etc.)
Atributo 3	Utilidad	(Conformidad del personal con las herramientas utilizadas)

Paso 4: Identificar Entidades y Atributos

Sub-Objetivo No Evaluar el impacto de documentar o no documentar los proyectos de software

Entidad	DOCUMENTOS	
Atributo 1	Tipo	(Todos los diferentes documentos producidos durante el proyecto)
Atributo 2	Plantillas	(Desarrolladas por la empresa, de estándares o de otras fuentes) (SI/NI)
Atributo 3	Esfuerzo	(Horas x Persona empleadas en elaborar los diferentes tipos de documentos)
Atributo 4	Volumen	(Nro. de páginas por tipo de documento)

Sub-Objetivo No Planificar con mayor precisión el proyecto

Entidad	PROCESO DE DESARROLLO	
Atributo 1	Tipo de aplicación de software	(Financiera, Contable, etc.)
Atributo 2	Tipo de proyecto de software	(Nuevo desarrollo, mejora, adaptación, etc.)
Atributo 3	Complejidad	(Estimada técnica y del negocio)
Atributo 4	Metodología o Modelo de desarrollo de SW	(Desarrollada en la empresa o adquirida a terceros)
Atributo 5	Incursiones	(Reportes de problemas)
Atributo 6	Progreso	(Continuidad, cumplimiento de las actividades y tareas planificadas)
Atributo 7	Tiempo	(Días, semanas o meses calendario)
Atributo 8	Costo	(Estimado vs Real)

Sub-Objetivo No Planificar con mayor precisión el proyecto

Sub-Objetivo No Monitorear los cambios en los requerimientos durante el desarrollo del proyecto

Entidad	REQUERIMIENTOS	
Atributo 1	Cantidad de requerimientos a desarrollar	(Definidos en las fases iniciales del proyecto)
Atributo 2	Cambios durante el desarrollo del proyecto	(Nuevos, borrados, modificados)

Paso 4: Identificar Entidades y Atributos

Sub-Objetivo No Planificar con mejor precisión el proyecto
 Sub-Objetivo No Generar diseños confiables

Entidad DISEÑO

Atributo 1	Tipo	(Orientado a objetos estructurado etc.)
Atributo 2	Tamaño	(Número de Clases / Número de Casos de uso)
Atributo 3	Tipo de inspecciones	(Se acepta, se rechaza o se piden modificaciones)

Sub-Objetivo No Planificar con mejor precisión el proyecto
 Sub-Objetivo No Disminuir el tiempo de codificación del SW

Entidad CÓDIGO FUENTE

Atributo 1	Estandar de codificación	(Definido y utilizado s/n)
Atributo 2	Complejidad	(De los metodos, funciones y procedimientos)
Atributo 3	Re-utilización por componentes	(re-utilizados, nuevos, borrados, modificados)

Sub-Objetivo No Planificar con mejor precisión el proyecto
 Sub-Objetivo No Monitorear los cambios en los requerimientos durante el desarrollo del proyecto
 Sub-Objetivo No Generar diseños confiables
 Sub-Objetivo No Disminuir el tiempo de codificación del SW
 Sub-Objetivo No Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final

Entidad DEFECTOS / FALLAS

Atributo 1	Cantidad	(Defectos descubiertos * Inspección y/o en las pruebas)
Atributo 2	Tipo	"origen" (Requerimientos, Diseño, Codificación etc.)
Atributo 3	Severidad	(leve, grave, critica)
Atributo 4	Tiempo en cola	(Tiempo transcurrido desde que se genera el reporte de defectos)
Atributo 5	Tiempo de corrección	(Tiempo que se requiere para corregir los defectos)
Atributo 6	Estatus	(Abierto, cerrado, descartado)
Atributo 7	Fallas	(Defectos que se presentan despues de liberar el software)

Paso 5: Formalizar los Objetivos de Medición

- Utilizando los Sub-Objetivos, Entidades y Atributos se generan los Objetivos de Medición utilizando la siguiente estructura:
 - Un Objeto de Interés.
 - Un Propósito.
 - Una Perspectiva.
 - Una descripción del entorno y restricciones.

Paso 5: Formalizar los Objetivos de Medición

- **Objeto de Interés:** Puede ser un producto, proceso, recurso, agente, artefacto, actividad, métrica o entorno. También es una entidad. En resumen es cualquier cosa real o abstracta de la cual se desea conocer más sobre ella.
- **Propósito:** El propósito de una actividad de medición puede ser entender, predecir, planear, controlar, comparar, evaluar o mejorar algún aspecto con respecto a la calidad o productividad de un objeto.

Paso 5: Formalizar los Objetivos de Medición

- **Perspectiva:** Identifica quién está interesado en los resultados de la medición. Es el principal punto de vista de la actividad de medición, tal como: Un desarrollador, un administrador, un cliente, etc.
- **Entorno:** Provee un contexto para interpretar los resultados de medición. Incluye todo lo que afecta o es afectado por el objeto a ser medido.

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 1

Objeto de interés: Proceso de desarrollo del SW	
Propósito:	Determinar el nivel de satisfacción del cliente esta relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.
Perspectiva:	Examinar la satisfacción del cliente
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	Para evaluar la calidad de producto, se hará referencia al modelo (FURPS) de un producto de software El desarrollo se realiza en uncs casos dentro y/o fuera de la empresa de SW

Objetivo de medición No. 2

Objeto de interés: Proceso de desarrollo del SW	
Propósito:	Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros
Perspectiva:	Examinar el esfuerzo y tamaño del personal asignado al proyecto
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	Es común que una persona participe en varios proyectos a la vez. El personal puede laborar en el sitio de trabajo del cliente. La rotación es el principal problema con el personal. Se dedica más personal al desarrollo de proyectos que al mantenimiento. Hasta 2 personas trabajan en proy. Pequeños, 6 en medianos y más de 6 en grandes

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 3

Objeto de interés: Proceso de desarrollo del SW	
Propósito:	Conocer la conformidad del personal con las herramientas utilizadas en el desarrollo del proyecto para evaluar opciones de adquisición y/o cambio de herramientas.
Perspectiva:	Examinar el grado de aceptación de las herramientas utilizadas durante el proyecto
Desde el punto de vista del	Personal asignado al proyecto
Entorno y restricciones	Las empresas constantemente consultan por herramientas que sirvan de apoyo o mejoren la productividad

Objetivo de medición No. 4

Objeto de interés: Proceso de desarrollo del SW	
Propósito:	Determinar el esfuerzo empleado en realizar la documentación durante el desarrollo del proyecto para estimar el esfuerzo que sería necesario utilizar en una adecuada documentación en proyectos similares.
Perspectiva:	Examinar el esfuerzo en realizar los diferentes tipos de documentos
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	La premura por terminar el proyecto dificulta la documentación de los proyectos Solo un 40% de las empresas de software siempre documentan el desarrollo de sus proyectos

Objetivo de medición No. 5

Objeto de interés: Proceso de desarrollo del SW	
Propósito:	Monitorear el avance de las actividades y tareas planificadas para cumplir o reconsiderar los compromisos establecidos al inicio del proyecto.
Perspectiva:	Examinar el cumplimiento de las tareas planificadas, así como su costo y tiempo consumido.
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	El modelo de desarrollo tradicional (caescada) y espiral son los más utilizados La empresa puede tener certificación ISO 9001:2000, y/o estar trabajando en CMMI nivel 2 o 3

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 6	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto
Perspectiva:	Examinar el número de cambios en los requerimientos durante el transcurso del proyecto
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	En general se destina la 1/5 parte del tiempo del proyecto a la actividad de especificación de requerimientos En la mayoría de los casos el cliente desconoce el alcance del proyecto Los requerimientos del proyecto usualmente están incompletos Es usual que el cliente realice modificaciones a los requerimientos durante el desarrollo de proyectos

Objetivo de medición No. 7	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Estimar la cantidad adecuada de inspecciones sobre los diseños (actividades, tareas y/o entregables) para disminuir la propagación y severidad de los defectos
Perspectiva:	Examinar los resultados de las inspecciones realizadas sobre los diseños
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	En general se destina un 25% del tiempo del proyecto a la actividad de diseño El tipo de diseño más utilizado es el Orientado a objetos

Paso 5: Formalizar los Objetivos de Medición

Objetivo de medición No. 8	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Controlar la complejidad de los componentes del SW para disminuir el esfuerzo utilizado por terceros en comprenderlo y modificarlo
Perspectiva:	Examinar la complejidad de los componentes del SW
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	La actividad de codificación es la que toma más tiempo en ser completada Los objetos son los componentes de SW más reutilizados

Objetivo de medición No. 9	
Objeto de Interés:	Proceso de desarrollo del SW
Propósito:	Conocer la ocurrencia y causa de los defectos durante el desarrollo de los proyectos de SW para identificar acciones de prevención y/o contingencia
Perspectiva:	Examinar el efecto de los defectos sobre el proyecto y la cantidad de fallos que se presentan en el producto final
Desde el punto de vista del	Administrador de proyectos
Entorno y restricciones	El producto final que se entrega al cliente no está libre de una pequeña cantidad de defectos La fuente principal de defectos es la fase de especificación y requerimientos

Paso 6: Identificar Preguntas e Indicadores cuantificables

- En este paso se generan preguntas cuantificables relacionadas a los objetivos de medición que se desean ser contestadas.
- Se definen los indicadores que contestarán dichas preguntas.
- Cuando se identifican preguntas y definen indicadores es importante tener en cuenta el objetivo al cual se dirigen. Ej: En entornos cambiantes es menos útil precisar números y hacer análisis estadísticos.

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 1	Determinar si el nivel de satisfacción del cliente está relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.
Pregunta No. 1	Cuál es el nivel de satisfacción del cliente y/o usuario con el producto de software desarrollado?
Pregunta No. 2	Cuál es la percepción del cliente con la metodología de desarrollo de proyectos de la empresa?
Pregunta No. 3	En que actividades durante el desarrollo del proyecto participó el cliente y/o usuario final?
Nombre del indicador	Clientes Satisfechos y e Porcentaje de horas de participación en el proyecto
Método de cálculo	1) Encuesta con escala de medición Likert
Ámbito	Aplicado a Clientes y usuarios al término del proyecto
Objetivo de medición No. 3	Conocer la conformidad del personal con las herramientas utilizadas en el desarrollo del proyecto para evaluar opciones de adquisición y/o cambio de herramientas.
Pregunta No. 1	Se dispone la infraestructura (hardware y software) adecuada para un desarrollo sin contratiempos?
Pregunta No. 2	Cuál es el nivel de conformidad del personal respecto a la utilidad de las herramientas utilizadas?
Nombre del indicador	Conformidad del personal con las herramientas utilizadas
Método de cálculo	1) Encuesta con escala de medición Likert
Ámbito	Aplicado al personal del área de sistemas al término del proyecto

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 2	Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros.	
Pregunta No. 1	Se dispone del personal adecuado para el desarrollo del proyecto?	
Pregunta No. 2	El esfuerzo requerido al personal asignado al proyecto es adecuado con la carga de trabajo que tienen actualmente?	
Nombre del Indicador	Método de cálculo	Ámbito
Error en la estimación del esfuerzo	1) $\text{EsfuerzoReal} = [(1+\text{Error})] * \text{EsfuerzoEstimado}$	Calculado al final del proyecto
Disponibilidad del personal	2) $\text{HorasTrabajadas} / \text{Horas estimadas}$	Calculado de forma semanal para todo el personal

Objetivo de medición No. 4	Determinar el esfuerzo empleado en realizar la documentación durante el desarrollo del proyecto para estimar el esfuerzo que sería necesario utilizar en una adecuada documentación en proyectos similares.	
Pregunta No. 1	Qué documentos se generaron durante el desarrollo del proyecto?	
Pregunta No. 2	Cuál es el esfuerzo utilizado en la documentación del proyecto?	
Nombre del Indicador	Fracción del tiempo del proyecto empleado en documentación	
Método de cálculo	1) $\text{Horas} \times \text{Persona de documentación} / \text{Total Horas} \times \text{Persona}$	
Ámbito	Calculado al término del proyecto de software	

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 5	Monitorear el avance de las actividades y tareas planificadas para cumplir o reconsiderar los compromisos establecidos al inicio del proyecto.	
Pregunta No. 1	Las actividades y tareas se planificaron teniendo en cuenta la complejidad del proyecto?	
Pregunta No. 2	Las actividades y tareas planificadas para el proyecto se cumplieron según el cronograma establecido?	
Pregunta No. 3	Las actividades y tareas se completan sin excederse del presupuesto establecido al inicio del proyecto?	
Nombre del Indicador	Método de cálculo	Ámbito
Complejidad estimada	1) Evaluación con escala de 5 opciones y 2 referencias	Inicio del proyecto de software
Avance del proyecto	2) Variación entre fechas de inicio y fin de tareas del proyecto	Durante el proyecto
Índice de rendimiento de Tiempo SPI	3) EVMS	Durante el proyecto
Índice de rendimiento de costo CPI	4) EVMS	Durante el proyecto

Objetivo de medición No. 7	Estimar la cantidad adecuada de inspecciones sobre los diseños (actividades, tareas y/o entregables) para disminuir la propagación y severidad de los defectos.	
Pregunta No. 1	Cuántas inspecciones (revisiones) se efectúan por actividad (no solo de diseño) y/o entregables del proyecto?	
Pregunta No. 2	Qué diferencia se aprecia entre proyectos que tienen inspecciones regulares y aquellos proyectos con pocas inspecciones?	
Nombre del Indicador	Inspecciones por actividad o entregable generado	
Método de cálculo	1) Histograma de inspecciones por actividad o entregable del proyecto	
Ámbito	Durante el proyecto	

Paso 6: Identificar Preguntas e Indicadores cuantificables

Objetivo de medición No. 6	Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto.	
Pregunta No. 1	El cliente define concisa y completamente todos sus requerimientos en las fases iniciales o de especificación del proyecto?	
Pregunta No. 2	Los requerimientos del proyecto se mantienen estables (sin cambios) durante el desarrollo del mismo?	
Nombre del Indicador	Tasa de variación en los requerimientos del software	
Método de cálculo	1) $TasaVariación = \frac{CambiosEnReq.}{(Requiciales + CambiosEnReq)}$	
Alcance	Calculado a la terminación del proyecto de software	
Objetivo de medición No. 8	Controlar la complejidad de los componentes del SW para disminuir el esfuerzo utilizado por terceros en comprender y modificarlo.	
Pregunta No. 1	Cuál es la complejidad estructural de los métodos y procedimientos del software?	
Pregunta No. 2	Cuánto tiempo necesita el personal inexperto en comprender los componentes de SW existentes?	
Pregunta No. 3	Cuál es el tamaño (expresado en líneas de código) de los componentes del software?	
Nombre del Indicador	Método de cálculo	Alcance
Complejidad ciclomática o estimada	1) Estadísticos de complejidad de cada componente de software	Todos los componentes de software que disponga la empresa
Tiempo de aprendizaje	2) Graf. Dispersión Complejidad v.s. Horas de aprendizaje	
Tamaño de los componentes de software	3) Líneas de código x componente de software	
Objetivo de medición No. 9	Conocer la ocurrencia y causa de los defectos durante el desarrollo de los proyectos de SW para identificar acciones de prevención y/o contingencia.	
Pregunta No. 1	De qué forma se distribuyen los defectos durante el ciclo de desarrollo del proyecto?	
Pregunta No. 2	Qué clase de defectos son los más recurrentes?	
Pregunta No. 3	Qué fracción del tiempo del proyecto se utiliza para corregir defectos?	
Pregunta No. 4	Cuán eficiente es la empresa en corregir defectos en el software antes de entregarlo al cliente?	
Nombre del Indicador	Defectos v.s. Fallos	
Método de cálculo	1) Fuga de defectos = Fallos / (Fallos + Defectos) 2) Histograma de ocurrencia de defectos 3) Análisis Causa-Efecto	
Alcance	Durante el tiempo de desarrollo y posterior puesta en producción del software	

Paso 7: Identificar los Elementos de Datos

- Se listan los datos que son necesarios obtener para formar los indicadores.
- Algunos datos obtenidos pueden ayudar a formar más de un indicador.
- Las Plantillas de Recolección de Datos se encuentran al final del Paso 10, cada dato se encuentra numerado para que pueda ser ubicado en la tabla de datos e indicadores.

Cod	DATOS	Definición de la Clave y Características del Dato y/o	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	Forma de recolectar el dato en los sistemas de	FUENTE
57	Efectos										FORMA F02
58	Fecha Mes										FORMA F03
59	Líneas de código										FORMA F04
60	Componente, método o procedimiento				X	X					FORMA F04
61	Horas de desarrollo del componente								X		FORMA F04
62	Número de defectos				X	X				X	FORMA F04
63	Complejidad sistemática				X	X					FORMA F04
64	Demora de aprendizaje					X					FORMA F04

Paso 8: Definir las Mediciones

- Los nombres de las mediciones no son suficientes. Se tiene que poder decirle a otras personas exactamente como cada medición debe ser obtenida, y de esta manera poder ser interpretadas correctamente.
- Para lograr esto se obtienen las definiciones operacionales de cada medición.
- Definiciones Operacionales:** Le dicen a los usuarios cómo los datos deben ser recolectados. Cuando los usuarios no conocen la manera de recolectarlos, fácilmente se puede llegar a asunciones inválidas.

Paso 8: Definir las Mediciones

Definition Checklist for Source Statement Counts

Definition name: *Physical Source Lines of Code* Date: *8/7/82*
(basic definition) Originator: *UJ*

Measurement unit:	Physical source lines	Logical source statements		
Statement type	Definition <input checked="" type="checkbox"/>	Date array <input type="checkbox"/>	Includes	Excludes
Notes: a line or statement counts only once; classify 7 as the size and the highest precedence.				
1	Executable	Order of precedence >		
2	Non-executable			
3	Declarations			
4	Control directives			
5	Comments			
6	On-line test files			
7	On-lines with source code			
8	Indexes and cross-reference			
9	Block-structure comments			
10	Blank lines			
11				
12				
How produced:	Definition <input checked="" type="checkbox"/>	Date array <input type="checkbox"/>	Includes	Excludes
1	Programmer			
2	Generated with source code generators			
3	Converted with automatic translators			
4	Copied or reused without change			
5	Modified			
6	Revised			
7	Removed			
8				
Origin:	Definition <input checked="" type="checkbox"/>	Date array <input type="checkbox"/>	Includes	Excludes
1	New work, no other existence			
2	Prior work, later or revised form			
3	A previous version, built or rebuilt			
4	Commercial off-the-shelf software (COTS), other than libraries			
5	Government furnished software (GFS), other than reuse changes			
6	Another product			
7	A vendor-supplied language support library, unmodified			
8	A vendor-supplied operating system or utility, unmodified			
9	A piece of modified language support library or operating system			
10	Other commercial library			
11	A reuse library, software designed for reuse			
12	Other software component or library			
13				
14				
Usage:	Definition <input checked="" type="checkbox"/>	Date array <input type="checkbox"/>	Includes	Excludes
1	It is all part of the primary product			
2	Essential to or in support of the primary product			
3				

Figure 4-51: A Checklist-Based Definition for Source Lines of Code

Paso 9: Identificar las acciones necesarias para implementar las mediciones

- En este paso se añade información acerca del estado actual y el uso de las mediciones, de esta manera se puede preparar un plan efectivo para implementar las mismas. Existen 3 pasos:
 - Análisis
 - Diagnóstico
 - Acción.

Paso 9: Identificar las acciones necesarias para implementar las mediciones

- **Análisis:** Identificar las mediciones que la organización ya está utilizando y entender cómo están siendo recolectadas.
- **Diagnóstico:** Evaluar los elementos de datos que la organización ya está recolectando. Conocer también la disponibilidad de las mediciones propuestas.
- **Acción:** Empieza con identificar los elementos que se construirán en el plan de medición.
 - Identificar las fuentes de datos dentro del proceso de software existente.
 - Definir los métodos que serán usados para recolectar y reportar los datos.
 - Identificar las herramientas que serán requeridas para ayudar a la recolección, reporte y almacenamiento de los datos.
 - Entre otros.

Paso 10: Preparar un Plan

- Se prepara un Plan para implementar las acciones que han sido identificadas.
- La estructura del Plan es la siguiente:
 - 1.-) Objetivo
 - 2.-) Descripción
 - *Objetivos
 - *Ámbito
 - *Relaciones con otros procesos de mejora de software.
 - 3.-) Implementación
 - *Actividades, Productos y Tareas.
 - *Calendario.
 - *Recursos.
 - *Responsabilidades.
 - *Medición y Monitoreo.
 - *Supuestos.
 - 4.-) Operación Sostenida.

[Referencias]



- Goal-Driven Software Measurement – A Guidebook , Robert E. Parl, Wolfhart B. Goethert, William A. Florac , 1996.
- “Elaboración y análisis de métricas para el proceso de desarrollo de software para empresas desarrolladoras de software del ecuador ”- Plan piloto - Tesis ESPOL, Jorge H. Mazón, José L. Alvear,2006.

ANEXO 5: Diagrama GQIM explicativo

ANEXO 6: Metodología proceso GQIM

PRIMER NIVEL:

OBJETIVOS DEL NEGOCIO [8]:

MAXIMIZAR LA SATISFACCIÓN DEL CLIENTE

La percepción del cliente / usuario de haber utilizado correctamente su tiempo, habiendo recibido respecto de sus propias expectativas y en un determinado contexto ambiental la mejor prestación posible del servicio o producto adquirido.

MEJORAR LA CALIDAD DEL PRODUCTO DE SOFTWARE

Es la conformidad con los requerimientos y la conveniencia para el uso de un producto o servicio.

MEJORAR LA PRODUCTIVIDAD

La productividad es la medida de cómo los recursos de una empresa son administrados para cumplir a tiempo los objetivos establecidos en términos de cantidad y calidad.

SEGUNDO NIVEL:

SUB-OBJETIVOS DE NEGOCIO [8]:

SUB-OBJETIVO DE NEGOCIO No. 1

Conocer el nivel de satisfacción del cliente y su participación en el proyecto.

SUB-OBJETIVO DE NEGOCIO No. 2

Mejorar la estimación del personal requerido y la planificación de las capacitaciones.

SUB-OBJETIVO DE NEGOCIO No. 3

Conocer la utilidad de las herramientas para el desarrollo de SW.

SUB-OBJETIVO DE NEGOCIO No. 4

Evaluar el impacto de documentar o no documentar los proyectos de software.

SUB-OBJETIVO DE NEGOCIO No. 5

Planificar con mejor precisión el proyecto.

SUB-OBJETIVO DE NEGOCIO No. 6

Monitorear los cambios en los requerimientos durante el desarrollo del proyecto.

SUB-OBJETIVO DE NEGOCIO No. 7

Generar diseños confiables.

SUB-OBJETIVO DE NEGOCIO No. 8

Disminuir el tiempo de codificación del SW.

SUB-OBJETIVO DE NEGOCIO No. 9

Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final.

TERCER NIVEL:

OBJETIVOS DE MEDICIÓN [8]:

OBJETIVO DE MEDICIÓN No. 1

Determinar si el Nivel de Satisfacción del Cliente está relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.

OBJETIVO DE MEDICIÓN No. 2

Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros.

OBJETIVO DE MEDICIÓN No. 3

Conocer la conformidad del personal con las herramientas utilizadas en el desarrollo del proyecto para evaluar opciones de adquisición y/o cambio de herramientas.

OBJETIVO DE MEDICIÓN No. 4

Determinar el esfuerzo empleado en realizar la documentación durante el desarrollo del proyecto para estimar el esfuerzo que sería

necesario utilizar en una adecuada documentación en proyectos similares.

OBJETIVO DE MEDICIÓN No. 5

Monitorear el avance de las actividades y tareas planificadas para cumplir o reconsiderar los compromisos establecidos al inicio del proyecto.

OBJETIVO DE MEDICIÓN No. 6

Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto.

OBJETIVO DE MEDICIÓN No. 7

Estimar la cantidad adecuada de inspecciones sobre los diseños (actividades, tareas y/o entregables) para disminuir la propagación y severidad de los defectos.

OBJETIVO DE MEDICIÓN No. 8

Controlar la complejidad de los componentes del SW para disminuir el esfuerzo utilizado por terceros en comprenderlo y modificarlo.

OBJETIVO DE MEDICIÓN No. 9

Conocer la ocurrencia y causa de los defectos durante el desarrollo de los proyectos de SW para identificar acciones de prevención y/o contingencia.

CUARTO NIVEL:

PREGUNTAS [8]:

OBJETIVO DE MEDICIÓN No. 1

PREGUNTA 1.1

¿Cuál es el nivel de satisfacción del cliente y/o usuario con el producto de software desarrollado?

PREGUNTA 1.2

¿Cuál es la percepción del cliente con la metodología de desarrollo de proyectos de la empresa?

PREGUNTA 1.3

¿En que actividades durante el desarrollo del proyecto participó el cliente y/o usuario final?

OBJETIVO DE MEDICIÓN No. 2

PREGUNTA 2.1

¿Se dispone del personal adecuado para el desarrollo del proyecto?

PREGUNTA 2.2

¿El esfuerzo requerido al personal asignado al proyecto es adecuado con la carga de trabajo que tienen actualmente?

OBJETIVO DE MEDICIÓN No. 3

PREGUNTA 3.1

¿Se dispone la infraestructura (hardware y software) adecuada para un desarrollo sin contratiempos?

PREGUNTA 3.2

¿Cuál es el nivel de conformidad del personal respecto a la utilidad de las herramientas utilizadas?

OBJETIVO DE MEDICIÓN No. 4

PREGUNTA 4.1

¿Qué documentos se generaron durante el desarrollo del proyecto?

PREGUNTA 4.2

¿Cual es el esfuerzo utilizado en la documentación del proyecto?

OBJETIVO DE MEDICIÓN No. 5

PREGUNTA 5.1

¿Las actividades y tareas se planificaron teniendo en cuenta la complejidad del proyecto?

PREGUNTA 5.2

¿Las actividades y tareas planificados para el proyecto se cumplen según el cronograma establecido?

PREGUNTA 5.3

¿Las actividades y tareas se completan sin excederse del presupuesto establecido al inicio del proyecto?

OBJETIVO DE MEDICIÓN No. 6

PREGUNTA 6.1

¿El cliente define concisa y completamente todos sus requerimientos en las fases iniciales o de especificación del proyecto?

PREGUNTA 6.2

¿Los requerimientos del proyecto se mantienen estables (sin cambios) durante el desarrollo del mismo?

OBJETIVO DE MEDICIÓN No. 7

PREGUNTA 7.1

¿Cuántas inspecciones (revisiones) se efectúan por actividad (no solo de diseño) y/o entregables del proyecto?

PREGUNTA 7.2

¿Qué diferencia se aprecia entre proyectos que tienen inspecciones regulares y aquellos proyectos con pocas inspecciones?

OBJETIVO DE MEDICIÓN No. 8

PREGUNTA 8.1

¿Cuál es la complejidad estructural de los métodos y procedimientos del software?

PREGUNTA 8.2

¿Cuánto tiempo necesita el personal inexperto en comprender los componentes de SW existentes?

PREGUNTA 8.3

¿Cuál es el tamaño (expresado en líneas de código) de los componentes del software?

OBJETIVO DE MEDICIÓN No. 9

PREGUNTA 9.1

¿De qué forma se distribuyen los defectos durante el ciclo de desarrollo del proyecto?

PREGUNTA 9.2

¿Qué clase de defectos son los más recurrentes?

PREGUNTA 9.3

¿Qué fracción del tiempo del proyecto se utiliza para corregir defectos?

PREGUNTA 9.4

¿Cuán eficiente es la empresa en corregir defectos en el software antes de entregarlo al cliente?

QUINTO NIVEL:

INDICADORES

OBJETIVO DE MEDICIÓN No. 1

INDICADOR 1

Satisfacción del Cliente y Satisfacción del Usuario V.S. Esfuerzo del cliente y esfuerzo del usuario

OBJETIVO DE MEDICIÓN No. 2

INDICADOR 2

Porcentaje de error en la estimación del esfuerzo

INDICADOR 3

Porcentaje del esfuerzo por persona en las diferentes fases

OBJETIVO DE MEDICIÓN No. 4

INDICADOR 4

Porcentaje de esfuerzo empleado en documentación

OBJETIVO DE MEDICIÓN No. 5

INDICADOR 5

Complejidad del Negocio y Complejidad Técnica

INDICADOR 6

Porcentaje de error en la estimación del costo

OBJETIVO DE MEDICIÓN No. 7

INDICADOR 8

Porcentaje de inspecciones realizadas por fases y Porcentaje de inspecciones realizadas en el proyecto

OBJETIVO DE MEDICIÓN No. 6

INDICADOR 9

Porcentaje de variación de los requerimientos

OBJETIVO DE MEDICIÓN No. 8

INDICADOR 10

Complejidad del Código Fuente

INDICADOR 12

Tamaño de los componentes de software

OBJETIVO DE MEDICIÓN No. 9

INDICADOR 13

Ocurrencia de defectos y fallas por fases

INDICADOR 14

Tipo de defectos y fallas más ocurrentes

INDICADOR 15

Porcentaje del tiempo utilizado en corrección de defectos y fallas

INDICADOR 16

Eficiencia en atención de defectos y fallas

INDICADORES ADICIONALES

INDICADOR 7

Costo promedio hora real

INDICADOR 11

Indicador ponderado de la complejidad y demora del aprendizaje

INDICADOR 17

Defectos por horas de programación

SEXTO NIVEL:

DATOS

INDICADOR 1:

Satisfacción del Cliente y Satisfacción del Usuario V.S. Esfuerzo del cliente y esfuerzo del usuario

DATOS:

- 1 - Proyecto
- 5 - Satisfacción cliente
- 6 - Satisfacción usuario
- 12 - Esfuerzo total real

INDICADOR 2:

Porcentaje de error en la estimación del esfuerzo

DATOS:

- 1 - Proyecto
- 11 - Esfuerzo total estimado
- 12 - Esfuerzo total real

INDICADOR 3

Porcentaje del esfuerzo por persona en las diferentes fases

DATOS:

- 1 - Proyecto
- 18 - Nombre
- 25 - Fase
- 43 - Esfuerzo real

INDICADOR 4

Porcentaje de esfuerzo empleado en documentación

DATOS:

- 1 - Proyecto
- 12 - Esfuerzo total real
- 30 – Páginas
- 31 – Documentación

43 - Esfuerzo real

INDICADOR 5

Complejidad del Negocio y Complejidad Técnica

DATOS:

1 - Proyecto

15 - Complejidad negocio

16 - Complejidad técnica

INDICADOR 6

Porcentaje de error en la estimación del costo

DATOS:

1 - Proyecto

13 – Costo total estimado

14 – Costo total real

INDICADOR 8

Porcentaje de inspecciones realizadas por fases y Porcentaje de inspecciones realizadas en el proyecto

DATOS:

1 - Proyecto

25 - Fase

28 – Tipo tarea

INDICADOR 9

Porcentaje de variación de los requerimientos

DATOS:

1 - Proyecto

2 – Requerimientos

28 – Tipo tarea

50 - Requerimiento

INDICADOR 10

Complejidad del Código Fuente

DATOS:

1 - Proyecto

60 - Componente, método o procedimiento

63 - Complejidad ciclomática

INDICADOR 12

Tamaño de los componentes de software

DATOS:

1 - Proyecto

59 - Líneas de código

60 - Componente, método o procedimiento

INDICADOR 13

Ocurrencia de defectos y fallas por fases

DATOS:

1 - Proyecto

25 - Fase

28 – Tipo tarea

INDICADOR 14

Tipo de defectos y fallas más ocurrentes

DATOS:

1 - Proyecto

28 – Tipo tarea

47 – Severidad

INDICADOR 15

Porcentaje del tiempo utilizado en corrección de defectos y fallas

DATOS:

1 - Proyecto

12 – Esfuerzo total real

28 – Tipo tarea

49 – Esfuerzo real

INDICADOR 16

Eficiencia en atención de defectos y fallas

DATOS:

1 - Proyecto

28 – Tipo tarea

36 – Fecha real inicio

46 – Fecha de abierto

INDICADORES ADICIONALES

INDICADOR 7

Costo promedio hora real

DATOS:

1 - Proyecto

12 – Esfuerzo total real

14 – Costo total real

INDICADOR 11

Indicador ponderado de la complejidad y demora del aprendizaje

DATOS:

1 - Proyecto

60 - Componente, método o procedimiento

63 - Complejidad ciclomática

64 - Demora de aprendizaje

INDICADOR 17

Promedio de defectos por hora de programación

DATOS:

1 - Proyecto

61 - Horas de desarrollo del componente

62 - Número de defectos

**ANEXO 7.1: Relación entre variables esfuerzo
en el desarrollo versus complejidad técnica y
versus complejidad del negocio.**

ANEXO 7.1

Relación entre:

Esfuerzo en el desarrollo versus complejidad técnica

Esfuerzo en el desarrollo versus complejidad del negocio

PROYECTO	ESFUERZO (HORAS)	COMPLEJIDAD NEGOCIO	COMPLEJIDAD TÉCNICA
1	723	1,90	1,00
2	1135	2,20	2,50
3	221	1,60	1,80
4	449	1,60	1,80
5	245	1,80	1,20
6	268	2,60	2,50
7	348	2,90	2,20
8	143	3,50	2,40
9	315,25	1,5	1,50
10	193	3,00	2,33
11	516	1,6	2,4
12	290	2	1,8
13	290	3,50	3,50
14	328	1,63	1,13
15	1973	2,3	1,7
16	214	1,5	1,5
17	143	2,2	2
18	114	2,3	1,9
19	90	2	1,9
20	461	2	0,875

Esfuerzo en el desarrollo versus complejidad técnica

Covarianza = -15,91820313 r = -0,058199036

Desviación = 436,9821768
Estándar X

Desviación = 0,625913806
Estándar Y

No existe relación entre las variables esfuerzo en desarrollar y complejidad técnica

Esfuerzo en el desarrollo versus complejidad del negocio

Covarianza = -19,26970833 r = -0,07061779

Desviación = 436,9821768
Estándar X

Desviación = 0,624449465
Estándar Y

No existe relación entre las variables esfuerzo en desarrollar y complejidad del negocio

**ANEXO 7.2: Relación entre variables esfuerzo
del cliente y esfuerzo del usuario versus
satisfacción del cliente y satisfacción del
usuario.**

ANEXO 7.2

Relación entre:

Esfuerzo del cliente versus satisfacción del cliente

Esfuerzo del usuario versus satisfacción del usuario

PROYECTO	% ESFUERZO CLIENTE	% ESFUERZO USUARIO	SATISFACCIÓN CLIENTE	SATISFACCIÓN USUARIO
1	3,5%	4,3%	3,7	3,8
2	4,9%	4,4%	4,1	4
3	5,7%	5,7%	4,8	4,4
4	3,6%	3,6%	4,1	4,2
5	2,4%	1,6%	3,5	3,6
6	3,7%	4,5%	3,6	3,7
7	4,9%	5,5%	4,3	4,4
8	3,5%	3,5%	3,4	3,4
9	6,0%	4,8%	4,3	4,3
10	4,7%	5,2%	3,8	3,4
11	3,9%	5,8%	4,6	4,5
12	6,2%	6,9%	4,5	4,56
13	4,1%	5,2%	4	3,9
14	4,9%	6,4%	4,4	4,38
15	3,2%	3,4%	3,7	3,7
16	5,7%	8,6%	4,63	4,6
17	6,3%	9,1%	4,7	4,4
18	6,1%	5,3%	4,4	4,5
19	6,7%	10,0%	4,4	4,7
20	6,9%	4,3%	4,3	4,57

Esfuerzo cliente versus satisfacción cliente

Covarianza = 0,003926279 r = 0,709298875

Desviación = 0,013142426
Estándar X

Desviación = 0,421188403
Estándar Y

Existe relación entre las variables esfuerzo cliente versus satisfacción cliente

Esfuerzo usuario versus satisfacción usuario

Covarianza = 0,005420122 r = 0,631338914

Desviación = 0,020309092
Estándar X

Desviación = 0,422723137
Estándar Y

Existe relación entre las variables esfuerzo usuario y satisfacción usuario

**ANEXO 7.3: Relación entre variables esfuerzo
en el desarrollo versus número de personas
involucradas.**

ANEXO 7.3

Relación entre:

Esfuerzo en el desarrollo versus número de personas involucradas

PROYECTO	ESFUERZO (HORAS)	NUMERO DE PERSONAS
1	723	2
2	1135	3
3	221	3
4	449	4
5	245	3
6	268	2
7	348	4
8	143	2
9	315,25	1
10	193	3
11	516	3
12	290	3
13	290	3
14	328	3
15	1973	3
16	214	2
17	143	3
18	114	3
19	90	2
20	461	3

Esfuerzo en el desarrollo versus número de personas involucradas

Covarianza = 42,165625

$r = 0,134700532$

Desviación = 436,9821768
Estándar X

Desviación = 0,716350399
Estándar Y

No existe relación entre las variables esfuerzo y número de personas involucradas

ANEXO 7.4: Relación entre variables defectos y fallas versus complejidad técnica y versus complejidad del negocio.

Fallas versus complejidad técnica

Covarianza = 0,28625

$r = 0,318393895$

Desviación = 0,62591381
Estándar X

Desviación = 1,43636969
Estándar Y

Existe cierta relación entre las variables fallas y complejidad técnica

Fallas versus complejidad del negocio

Covarianza = 0,00891667

$r = 0,009941204$

Desviación = 0,62444947
Estándar X

Desviación = 1,43636969
Estándar Y

No existe relación entre las variables fallas y complejidad del negocio

**ANEXO 7.5: Relación entre variables esfuerzo
en documentación versus fallas.**

ANEXO 7.5

Relación entre:

Esfuerzo en documentación versus fallas

PROYECTO	% ESFUERZO DOCUMENTACIÓN	FALLAS
1	5,26%	3
2	7,93%	1
3	13,12%	2
4	11,14%	3
5	5,31%	3
6	12,69%	1
7	5,17%	1
8	1,40%	7
9	2,53%	1
10	5,44%	1
11	1,94%	1
12	6,21%	1
13	8,97%	2
14	8,23%	2
15	12,11%	2
16	5,71%	1
17	6,29%	1
18	7,02%	1
19	5,56%	1
20	5,42%	1

Esfuerzo en documentación versus fallas

Covarianza = -0,007864637

$r = -0,16276483$

Desviación = 0,033639683
Estándar X

Desviación = 1,436369693
Estándar Y

No existe relación entre las variables esfuerzo en documentación y fallas

ANEXO 8: Complejidad ciclomática

El presente anexo muestra un ejemplo para calcular la complejidad ciclomática, al efecto se ha utilizado un procedimiento para unir dos arreglos:

[24]

```
int [] Intercalar( int [] a1, int [] a2)
{
  int i = 0;
  1   int j = 0;
     int k = 0;
     int [] a3 = new int [a1.length + a2.length];
  2,3 while (( i < a1.length ) && ( j < a2.length ) )
     {
  4       if (a1[i] < a2[j])
  5           a3[k++] = a1[i++];
  6       else a3[k++] = a2[j++];
     }
  7   if (j == a2.length) //¿El 2º array no tiene datos?
  8       while (i < a1.length)
  9           a3[k++] = a1[i++];
  10  else while (j < a2.length)
  11      a3[k++] = a2[j++];
  12  return a3;
}
```

camino 1 : 1-2-7-8-12 Los dos arrays no tienen elementos.
RESULTADOS ESPERADOS: *No hacer nada, a3 vació.*

camino 2 : 1-2-3-7-8-12 a1 con datos, a2 sin datos. *Camino imposible.*
{Se tendrían que verificar a la vez: por (2) $i < a1.length$ es verdad y por (8) $i < a1.length$ es falso}
Solución del camino 2 que entre en el bucle.

camino 3 : 1-2-3-7-8-9-8-12 a1 con datos y a2 sin datos.

RESULTADOS ESPERADOS: *Se introduce el elemento del array 1 en el array de Salida.*

camino 4 : 1-2-3-4-5-2-7-8-9-8-12 a1 con datos y a2 con datos. $a1[i] < a2[j]$. *Camino imposible.*

{Se tendrían que verificar a la vez: por (3) $j < a2.length$ es verdad y por (7) $j == a2.length$ es verdad

Además, también se verifica: (2) $i < a1.length$ es falso y por (8) $i < a1.length$ es verdad}

Solución del camino 4, no ir por 8 e ir por 10, selecciono el camino sencillo.

camino 5 : 1-2-3-4-5-2-7-10-12 a1 con datos y a2 con datos. $a1[i] < a2[j]$. *Camino imposible.*

{pues se tendrían que verificar simultáneamente: por (3) $j < a2.length$ es verdad y por (10) $j < a2.length$ es falso}

Solución del camino 5 que entre en el bucle.

camino 6 : 1-2-3-4-5-2-7-10-11-10-12 a1 con datos, a2 con datos. $a1[i] < a2[j]$.

RESULTADOS ESPERADOS: *Se introducen en el array de Salida a3 el elemento del primer array y después el del segundo.*

Queda por analizar la arista de 6 a 2, 1-2-3-4-6-2 ahora hay dos opciones ir a 7 o ir a 3, con el análisis de los caminos anteriores determino que debo ir a 3 (el último elemento introducido es de a2, así que en a1 todavía quedan elementos).

camino 7 : 1-2-3-4-6-2-3-7-8-9-8-12 a1 con datos, a2 con datos. $a1[i] > a2[j]$.

RESULTADOS ESPERADOS: *Se introducen en el array de Salida a3 el elemento del segundo array y después el elemento del primero.*

Complejidad ciclomática = 7

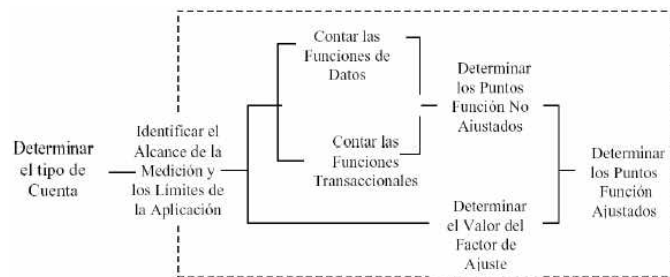
ANEXO 9: Puntos de Función

Paso 1. Determinar el tipo de conteo

Este paso consiste en definir el tipo de conteo entre desarrollo, mantenimiento o de una aplicación ya instalada. Esta es una forma de determinar el objetivo del conteo. [30]

Gráfico 6-1

Determinación tipo de cuenta Puntos de Función [30]

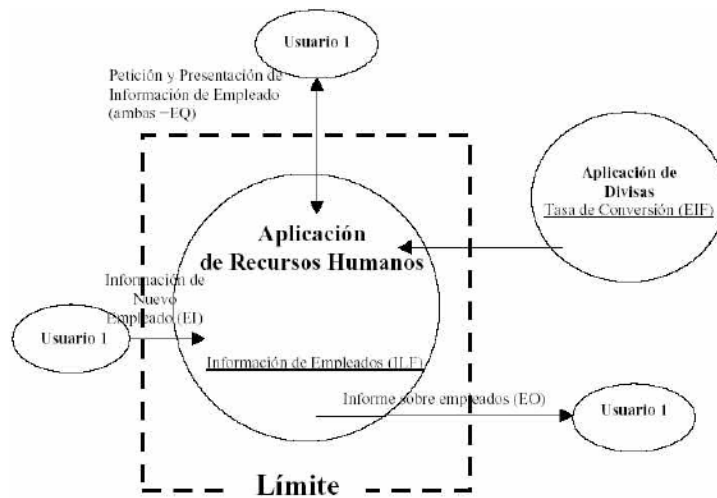


Paso 2. Identificar los alcances de la medición y los límites de la aplicación.

El propósito de una medición consiste en dar una respuesta a un problema de negocio. El alcance de la medición define la funcionalidad que va a ser incluida en una medición específica y puede abarcar más de una aplicación. [30]

Gráfico 6-2

Identificación alcance medición y límites de la aplicación [30]



Paso 3. Contar las funciones de datos

Este paso consiste en identificar y contar la capacidad de almacenamiento de los datos. Se distinguen dos tipos de funciones de datos:

- Archivo Lógico Interno: es un grupo de datos relacionados que el usuario identifica, cuyo propósito principal es almacenar datos mantenidos a través de alguna transacción que se está considerando en el conteo.
- Archivo de Interfaz Externo: es un grupo de datos relacionados y referenciados pero no mantenidos por alguna transacción dentro del conteo.

A cada componente identificado se le asigna una complejidad (bajo, medio o alto) considerando principalmente el número de datos. [30]

Paso 4. Contar las funciones transaccionales.

Este paso consiste en identificar y contar la capacidad de realizar operaciones.

Se distinguen tres tipos de funciones transaccionales:

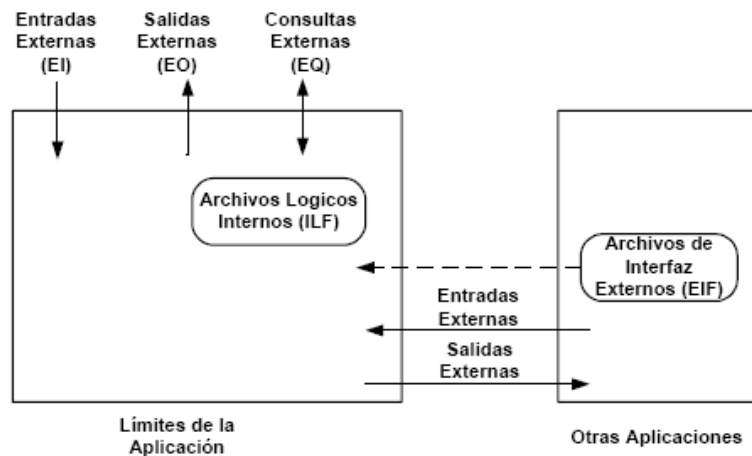
- Entrada Externa: es un proceso cuyo propósito principal es mantener uno más archivos lógicos internos.
- Salida Externa: es un proceso cuyo propósito principal es presentar información al usuario mediante un proceso lógico diferente al de sólo recuperar los datos.
- Consulta Externa: es un proceso cuyo propósito principal es presentar información al usuario leída de uno o más grupos de datos.

A cada componente identificado se le asigna una complejidad (bajo, medio o alto) considerando el número de datos utilizado en el proceso y los archivos referenciados.

Estos 5 componentes lógicos básicos son con los que se describe la funcionalidad de una aplicación y los podemos representar gráficamente de la siguiente forma: **[30]**

Gráfico 6-3

Procedimiento de conteo de funciones transaccionales [30]



Paso 5. Determinar los puntos de función no ajustados.

Este paso consiste en sumar el número de componentes de cada tipo conforme a la complejidad asignada y utilizar la siguiente tabla para obtener el total. [30]

Gráfico 6-4

Tabla determinación puntos de función no ajustados [30]

	Bajo	Medio	Alto	Total
EI	___ x 3= ___	___ x 4= ___	___ x 6= ___	___
EO	___ x 4= ___	___ x 5= ___	___ x 7= ___	___
EQ	___ x 3= ___	___ x 4= ___	___ x 6= ___	___
ILF	___ x 7= ___	___ x 10= ___	___ x 15= ___	___
EIF	___ x 5= ___	___ x 7= ___	___ x 10= ___	___

Paso 6. Determinar el valor del factor de ajuste.

El factor de ajuste se obtiene sumando 0.65 a la sumatoria de los grados de influencia de las 14 características generales del sistema, multiplicado por 0.01. Dentro de las características hay criterios como: complejidad del proceso, facilidad de instalación, entrada de datos en línea, etc. [30]

Paso 7. Determinar los puntos función ajustados.

Para determinar los puntos función ajustados se consideran los puntos función no ajustados por el factor de ajuste. [30]