

ASPECTOS DE LA CALIDAD Y DIFICULTADES DURANTE LA GESTION DE PROYECTOS “Estudio exploratorio”

Jorge Mazón Naranjo¹, José Alvear Cervantes², Gipsy Bracco Vera³.

¹Egresado de Ingeniería en Computación, ESPOL-FIEC, mail: jamazon@espol.edu.ec

²Egresado de Ingeniería en Computación, ESPOL-FIEC, mail: jalvear@espol.edu.ec

³Egresada de Ingeniería en Estadística Informática, ESPOL-ICM, mail: gbracco@espol.edu.ec

RESUMEN

Con el objetivo de conocer de primera mano las principales características que presenta el proceso de desarrollo de software en las empresas del Ecuador, se evaluó un cuestionario de 57 preguntas a 62 empresas ubicadas en las ciudades de Quito y Guayaquil. Dicho instrumento se construyó con las observaciones y comentarios de docentes y profesionales del área de computación, procurando tocar los temas de mayor interés. Considerando las distintas escalas de las preguntas se utilizó el coeficiente de confiabilidad Alfa de Cronbach estandarizado determinándose un 84% de confiabilidad en el instrumento.

Los resultados muestran que el 35% de las empresas exportan actualmente productos de software, esperándose en el corto plazo un incremento del 39% de nuevas empresas exportadoras. La programación extrema es la tercera metodología mayormente usada por las empresas a pesar de tener 8 años de existencia. Entre los objetivos más importantes al momento de desarrollar software está la maximización de la satisfacción del cliente, mejorar la calidad del software y mejorar la productividad. Los mayores costos, uso de recursos y tiempo recaen sobre la codificación y pruebas de unidades. Y los principales mantenimientos realizados por las empresas son para agregar funcionalidad y corregir errores en el software.

Palabras claves: Ecuador, empresas, software, calidad, ESPOL.

INTRODUCCION

La falta de información sobre los factores que influyen la gestión de proyectos de software en las empresas del Ecuador, no permite identificar sus fortalezas y debilidades en la construcción y mantenimiento de software. Por consiguiente, y siendo nuestro propósito el disponer de datos de referencia que guíen investigaciones posteriores sobre el desarrollo de software, el proyecto ESPOL-VLIR a través del SUB-componte 8 de ingeniería de software impulsó la realización de un estudio exploratorio¹ dirigido a las empresas desarrolladoras con sede en las ciudades de Quito y Guayaquil durante los meses de Abril y Mayo de 2005.

CONTENIDO

Se formó un cuestionario con preguntas generales, las mismas que fueron formadas una vez revisada bibliografía sobre el desarrollo y mantenimiento de software disponible en libros y publicaciones en Internet de autores extranjeros. Posteriormente se realizaron las siguientes actividades:

1. Se desarrolló un piloto de este cuestionario en empresas de Guayaquil en el mes de Diciembre del 2004, con el propósito de mejorar su contenido. Aunque al realizar esta actividad no obtuvimos los suficientes comentarios o correcciones que buscábamos.
2. Terminado el piloto, se formó una segunda versión del cuestionario. En esta ocasión se contó con las opiniones de otros docentes, la tutora belga del proyecto VLIR y diversos profesionales de las empresas de Quito, gracias a lo cual se mejoró considerablemente su contenido. Cabe mencionar un elevado interés en el desarrollo y resultados de la investigación por parte de los profesionales de la Capital.
3. Se solicitó telefónicamente a las empresas de Quito y Guayaquil una entrevista con un promedio de duración de 40 minutos en la cual se realizaron las siguientes actividades en su orden:
 - 1) Introducción a la investigación.
 - 2) Evaluación del cuestionario.
 - 3) Solicitud del apoyo necesario para posteriores actividades.
4. Se elabora el presente reporte tomando como muestra a 63 empresas de desarrollo entre Quito y Guayaquil para el análisis.

¹ Los estudios exploratorios se efectúan normalmente, cuando el objetivo es examinar un tema o problema de investigación poco estudiado o que no ha sido abordado antes. Es decir, cuando la revisión de la literatura reveló que únicamente hay guías no investigadas e ideas vagamente relacionadas con el problema de estudio. [Hernández 95]

Las variables fueron analizadas de manera independiente, determinándose estadísticas básicas acerca del entrevistado, la empresa, el desarrollo y mantenimiento de software. De estas se obtienen los siguientes resultados.

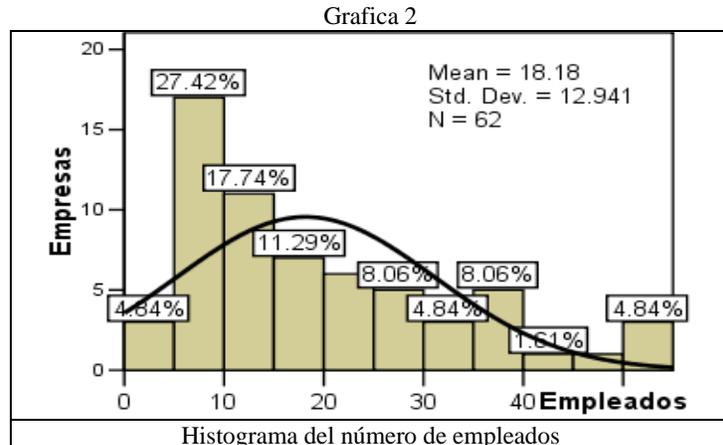
Información de la empresa y del entrevistado

Número de empleados

Se conoce que en el año 2003 en Ecuador existían 134 empresas, distribuidas de la siguiente forma: 98 en Quito y 36 en Guayaquil, [1] para este estudio se contacto a 63 empresas distribuidas como sigue: 50 de Quito y 13 de Guayaquil.

Estadísticos	
Empresas	*62.00
Media	18.18
Mediana	14.50
Moda	8.00
Desv. Estándar	12.94
Varianza	167.46
Kurtosis	0.61
Rango	53.00
Mínimo	2.00
Máximo	55.00
Cuartil 25	8.00
Cuartil 50	14.50
Cuartil 75	25.00

Tabla I



* Se excluye el número de empleados de una empresa que por su gran diferencia sesga los resultados

Antigüedad de las empresas

Estadísticos	
Empresas	*62.00
Media	8.56
Mediana	8.00
Moda	5.00
Desv. Estándar	4.90
Varianza	24.00
Kurtosis	1.63
Rango	22.00
Mínimo	2.00
Máximo	24.00
Cuartil 25	5.00
Cuartil 50	8.00
Cuartil 75	10.25

Tabla II



* Se excluye el número de empleados de una empresa que por su gran diferencia sesga los resultados

Exportación de software y certificación ISO 9001:2000

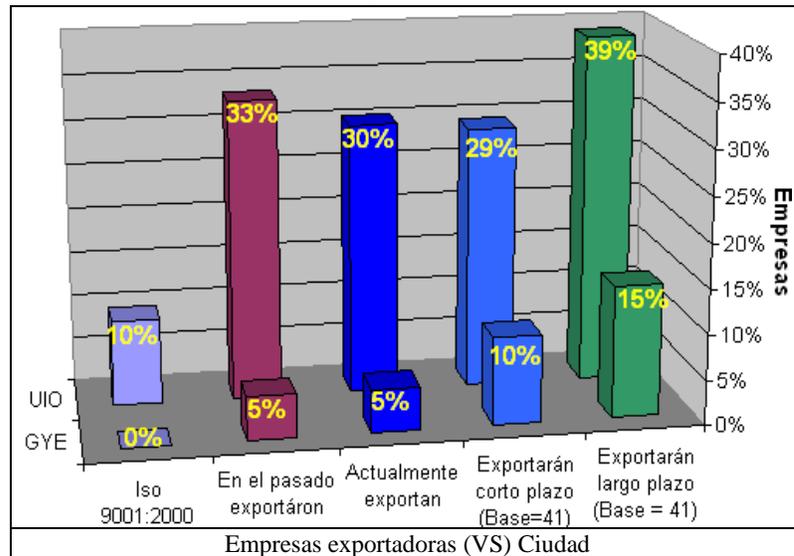
En Octubre de 2003, el 63,6% de las empresas comerciaban solo en el mercado nacional y el 36,4% lo hacia tanto en el mercado nacional como en el internacional con un promedio de exportaciones del 45,2%. [1]

Actualmente el 65% de las empresas no exporta, el restante 35% si lo hace. Para estas empresas este rubro representa en promedio el 48% del software desarrollado y el 37% del valor facturado.

De las 41 empresas que actualmente no exportan, el 39% de estas (29% Quito, 10% Guayaquil) esperan exportar en el corto plazo (menos de 1 año), como lo muestra la grafica 4.

En cuanto a la certificación ISO 9001:2000, todas las empresas que tienen esta certificación están en Quito, todas ellas son exportadoras y en promedio exportan el 51% de sus productos de software, tienen entre 35 y 50 empleados y entre 8 y 11 años de funcionamiento.

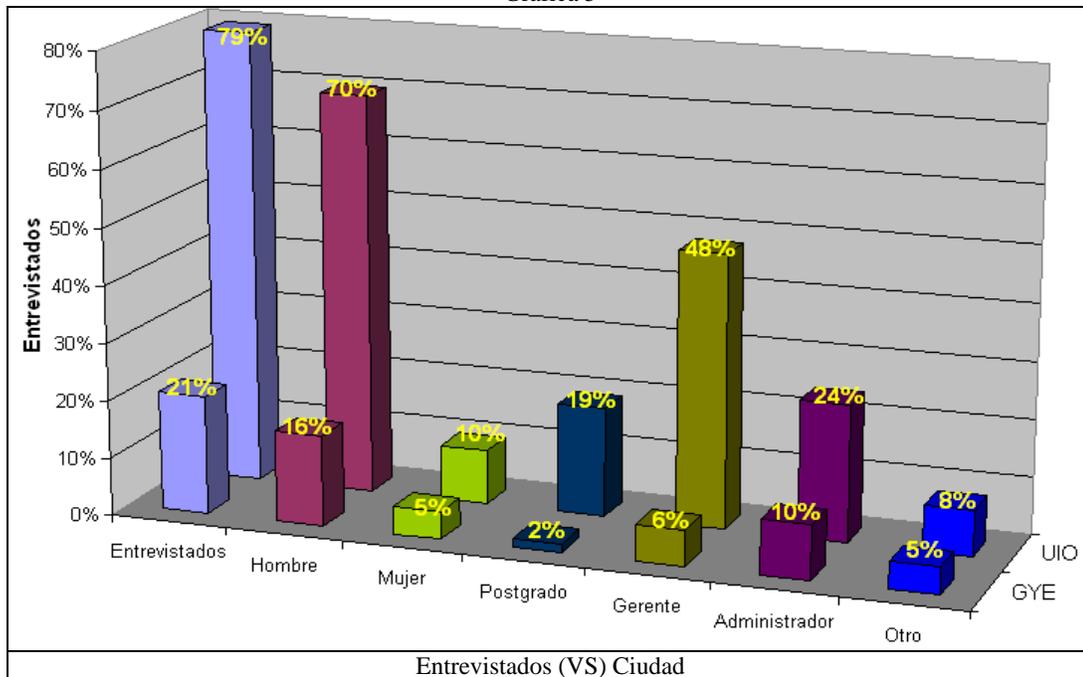
Grafica 4



Entrevistados

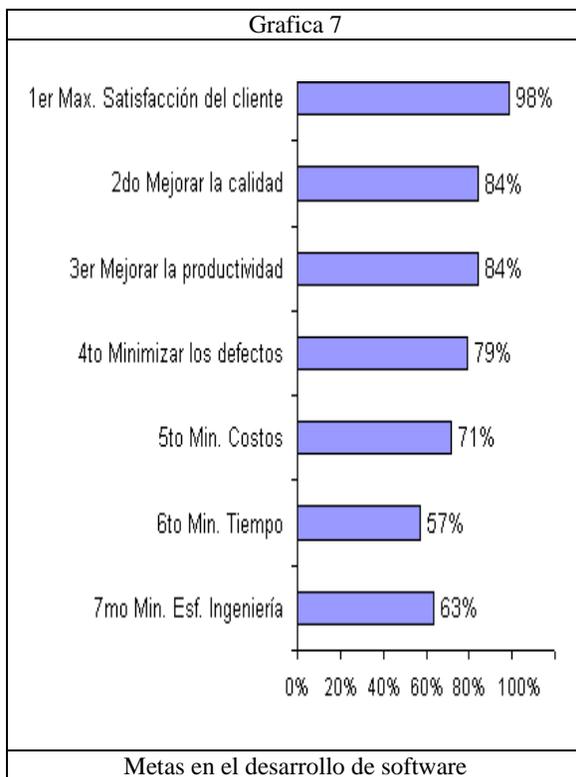
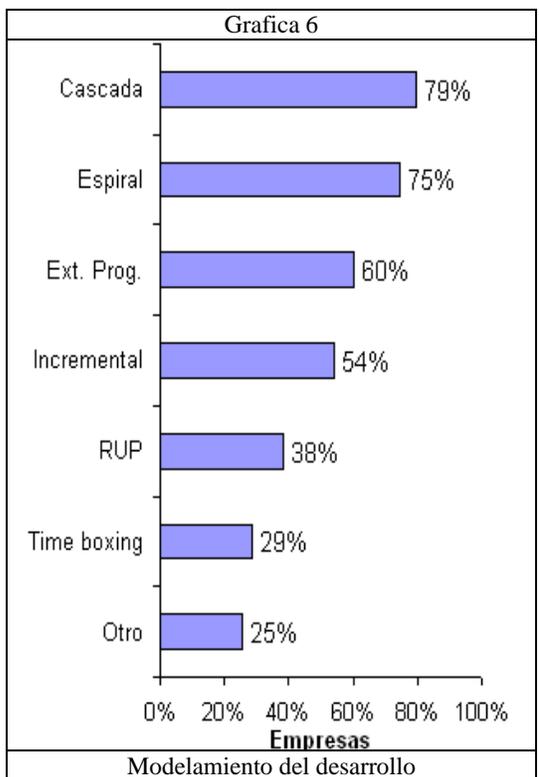
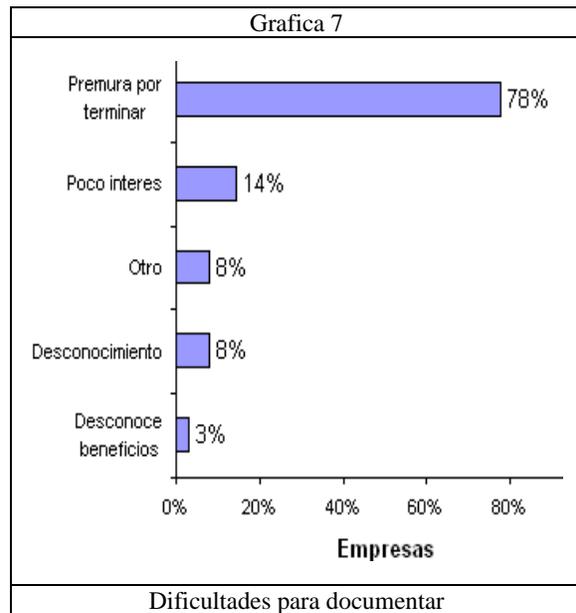
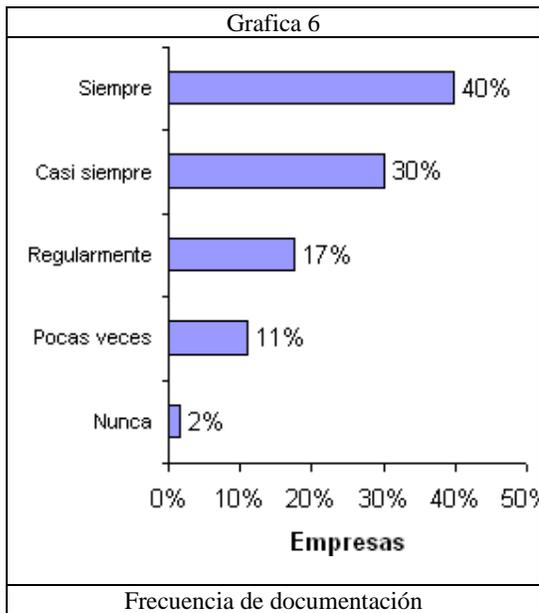
Al momento de solicitar la entrevista, buscamos reunirnos con la(s) persona(s) encargadas de la gestión del desarrollo del software, siendo los gerentes (54%) y los administradores (34%) de las empresas las personas más entrevistadas. De estos el 21% tiene postgrados. En la grafica 5 se puede observar estos datos por ciudad.

Grafica 5



Generalidades

Se analizaron aspectos que cubren todo el proceso de desarrollo como el tiempo, modelo del proceso de software, entre otros que no se limitan a una sola fase del desarrollo de proyectos de software².

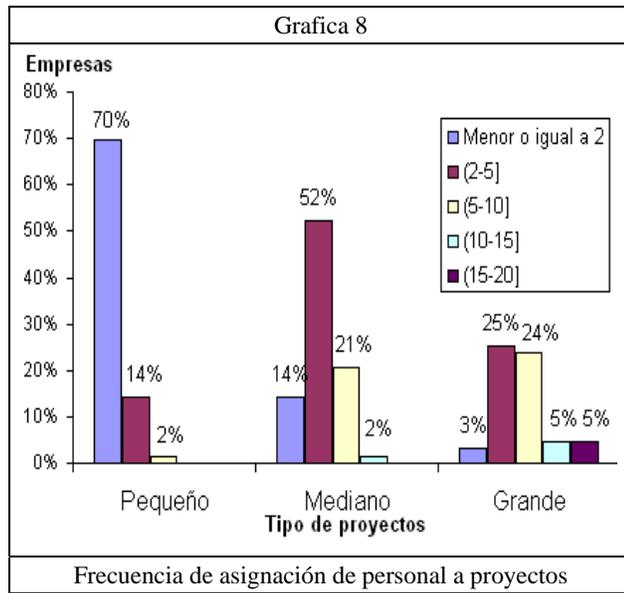


² Para efectos de nuestro estudio, se entiende como proyectos pequeños aquellos que duran entre 1 y 6 meses, proyectos mediano los que duran entre 7 y 15 meses y grandes los que consumen mas de 15 meses.[1]

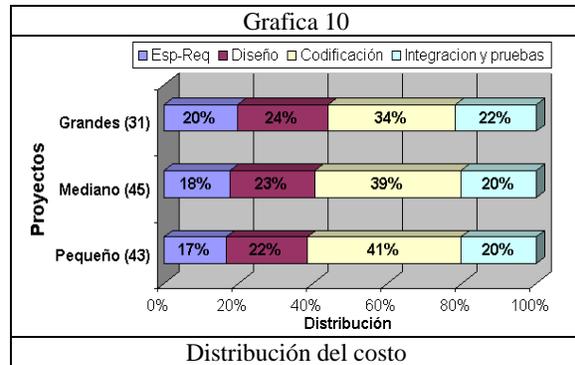
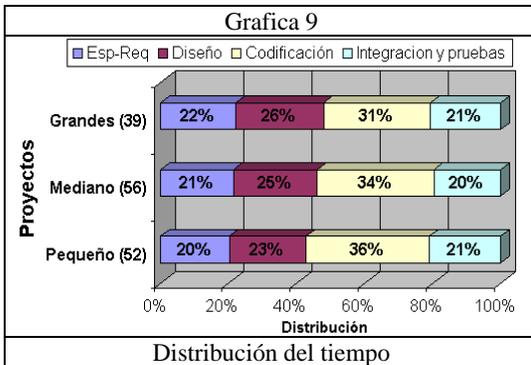
Asignación del personal a los proyectos

Estadísticos			
	PEQ.	MED.	GRD.
Total	54	56	39
Media	2.04	4.39	7.49
Mediana	2.00	4.00	7.49
Moda	2.00	4.00	4.00
Desv. Estándar	1.01	2.42	4.51
Varianza	1.02	5.84	20.31
Kurtosis	4.50	6.07	1.48
Rango	5.00	14.00	19.00
Mínimo	1.00	1.00	1.00
Máximo	6.00	15.00	20.00
Cuartil 25	1.00	3.00	4.00
Cuartil 50	2.00	4.00	6.00
Cuartil 75	2.00	5.75	10.00

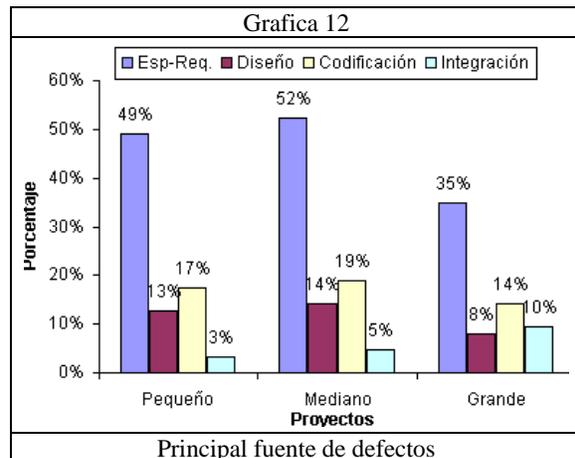
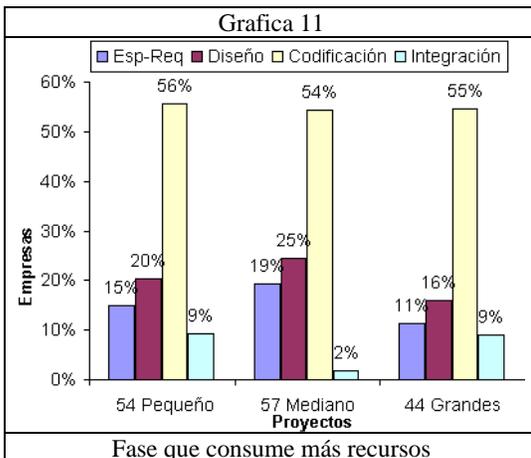
Tabla III



Distribución del tiempo y costo



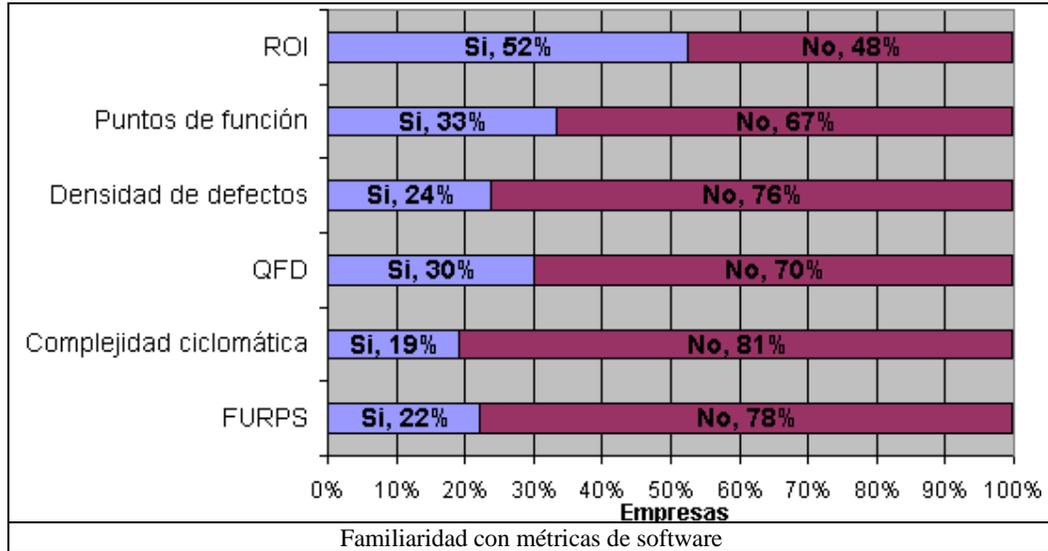
Fase de mayor consumo de recursos y principal origen de defectos



Medición del desarrollo de software

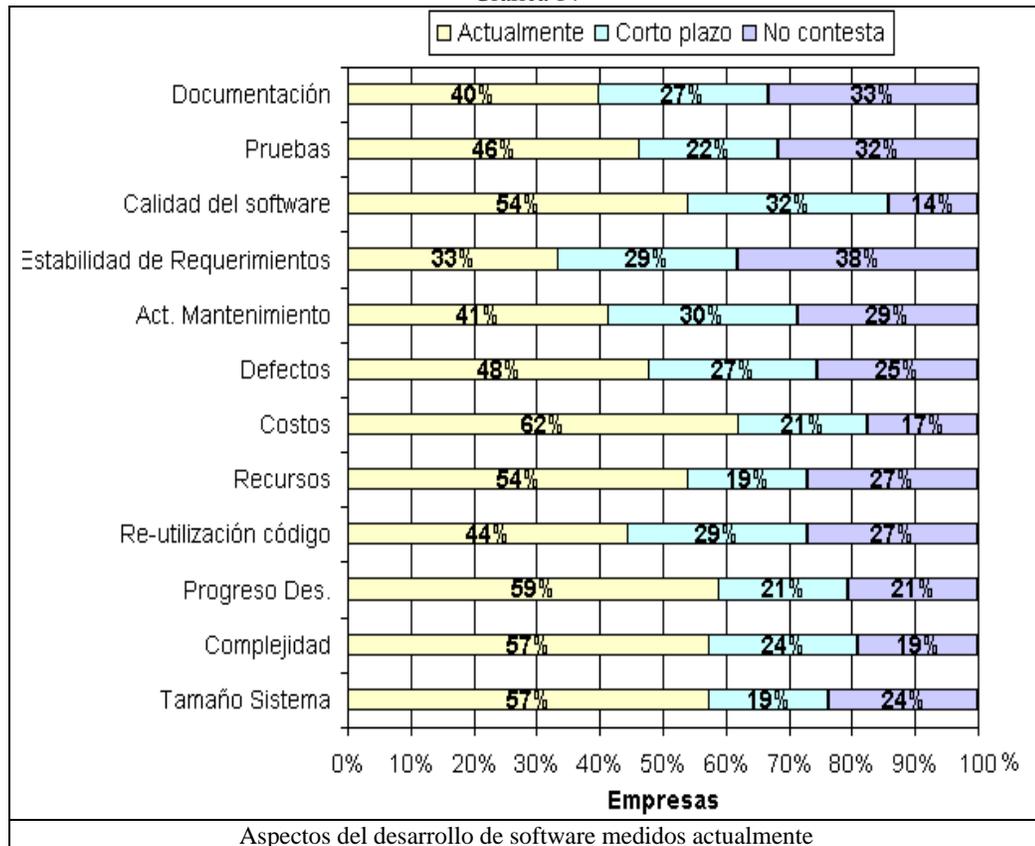
El 51% de los entrevistados desconoce el concepto de métrica de software³.

Grafica 13



Mediciones actuales y por considerarse

Grafica 14



³ Se define como cualquier tipo de medida relacionada con un sistema proceso o documentación de software. [6]

Especificación y requerimientos

Estimación

En su orden, los factores estimados por las empresas en la fase de especificación y requerimientos son: El tiempo (El 89% de las empresas estiman este factor), los recursos necesarios (86%), el tamaño del sistema y el costo (78%), la complejidad (67%) y finalmente la documentación (33%).

Problemas más frecuentes

Los dos problemas más comunes al momento de recibir las especificaciones y requerimientos del usuario son: El desconocimiento del alcance o límites del proyecto por parte del cliente, según el 92% de los entrevistados, y en segundo lugar los requerimientos incompletos según el 94% de los entrevistados. La tabla IX muestra la prioridad de estos y otros problemas.

Problemas frecuentes con los requerimientos de usuario

Prioridad de los problemas	Empresas
1 El cliente desconoce el alcance del proyecto	92%
2 Requerimientos incompletos	94%
3 Requerimientos confusos e inconsistentes	79%
4 No se mencionan restricciones	83%
5 Requerimientos agrupados	71%
6 Otro	10%

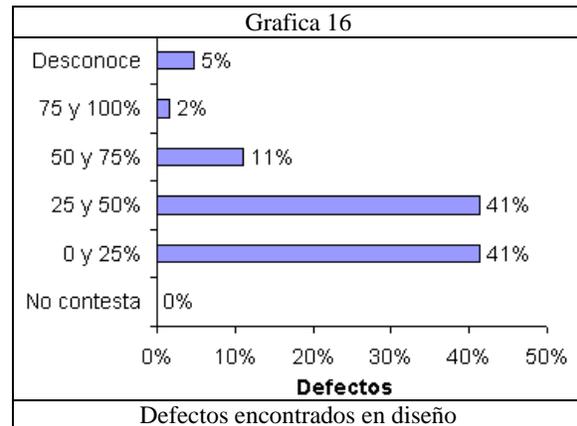
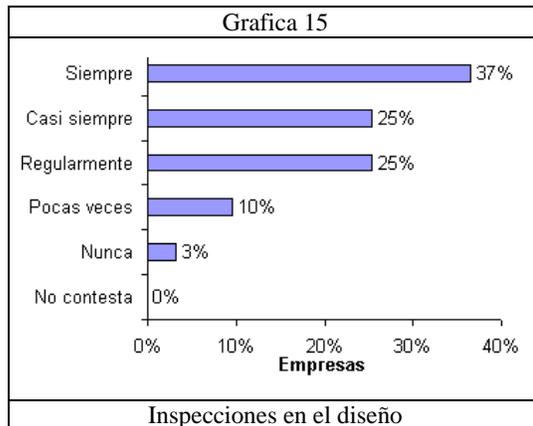
Tabla IV

Solo el 21% de las empresas siempre utilizan datos históricos para estimar y un 29% casi siempre utiliza esta información.

Diseño

Inspecciones en la fase de diseño

Solo el 37% de los entrevistados dice hacer siempre inspecciones en la fase de diseño y 25% casi siempre lo hace, este dato no es favorable, pues con las aplicaciones complejas de hoy, solo el 44% de los diseños de las aplicaciones cumplen con el 20% de las expectativas de los usuarios. [13]



Defectos detectados en diseño

Las inspecciones de diseño, son particularmente importantes ya que los defectos de diseño ocurren con frecuencia y el costo que implican se hacen altos si no se los detecta a tiempo, una regla practica enuncia que un 50 y 75% de todos los defectos de diseño pueden ser encontrados con inspecciones [9]. En nuestro estudio, el 41% de las empresas detectan una cantidad de defectos entre 0 y 25% en el diseño, el 41% de las empresas detectan entre el 25 y 50% de defectos y un 11% de las empresas detectan entre el 50 y 75% defectos durante el diseño.

Tipo de diseño

El 86% de las empresas utiliza el diseño orientado a objetos, el 41% de todas las empresas usa el diseño estructurado y un 17% de las empresas utilizan patrones de diseño.

Estimación al terminar el diseño

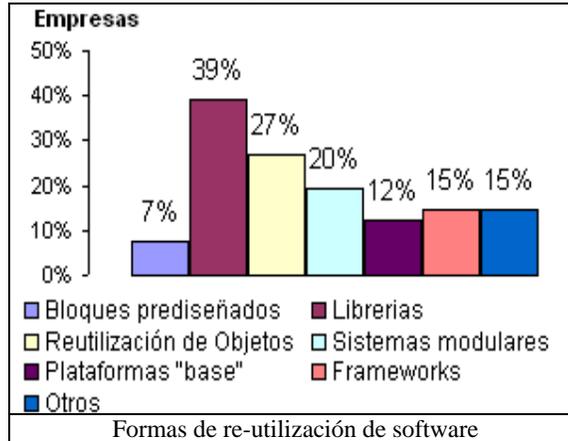
Los entrevistados indican que al finalizar el diseño, realizan una estimación más precisa especialmente del tiempo (71%), los recursos necesarios (63%), los costos (60%) y la complejidad del sistema (51%).

Codificación, integración y pruebas

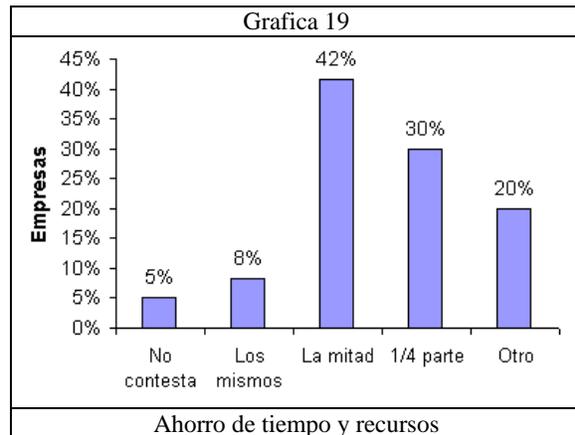
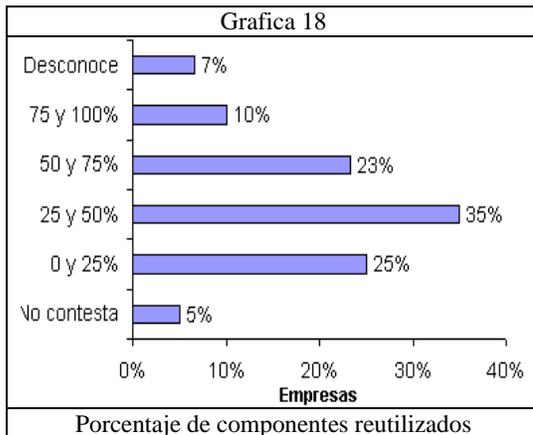
Reutilización de componentes de software

El 95% de las empresas reutilizan los componentes de software desarrollados, esto es bueno, debido a que se conoce que los proyectos creados primariamente de software re-usado experimentan solo 1/3 de la densidad de defectos de aquellos que son nuevos [9]. La grafica XIV muestra un resumen de los componentes que se vuelven a utilizar.

Grafica 17



Tiempo y recursos reutilizando componentes de software



Control de defectos y pruebas

Para procurar controlar la cantidad y gravedad de los defectos, las inspecciones son realizadas por el 33% de las empresas, el análisis de defectos lo hace el 32% de las empresas, el 67% de estas ejecuta revisiones periódicas y en general el 60% de las empresas los controlan cuando estos se producen.

El tipo de pruebas más utilizadas son las de integración (79%), de defectos (70%), de caja negra (65%), unitarias (44%) y de stress (40%). Estas pruebas de software son realizadas principalmente por el cliente (67%) el mismo equipo de desarrollo (56%) y equipos dedicados a pruebas (48%).

Se pregunto si realizaban pruebas de "code coverage"⁴, El 59% de los entrevistados nunca a realizado este tipo de pruebas, de echo en la entrevista manifestaron no conocerla.

⁴ Las pruebas que no toman en cuenta el código que ejecutan, solo prueban el 55% del mismo. La información que nos proporciona el probar el código fuente contabilizando las líneas que se ejecutan, nos ayuda a identificar otras pruebas necesarias que ejecuten todo el código. [9]

3.5.5) Defectos detectados y acciones de contingencia

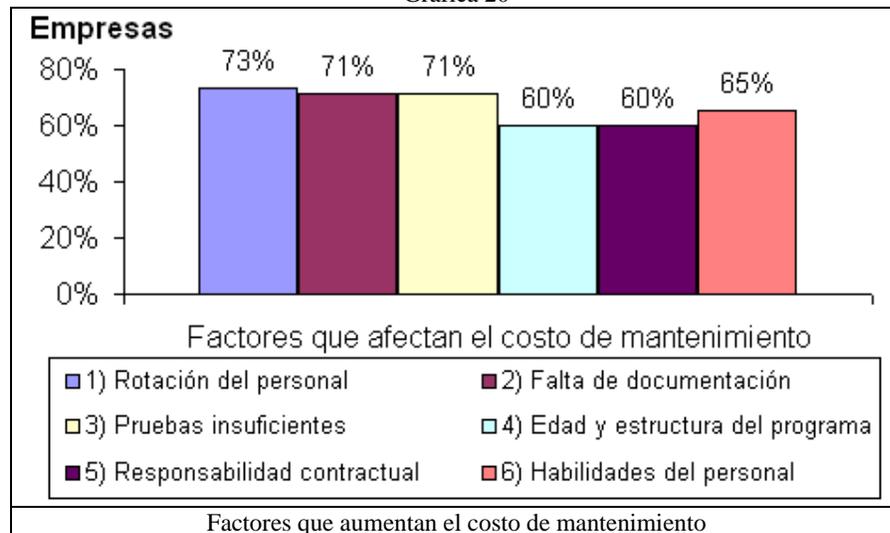
Casi la mitad de las empresas (44%) encuentra entre 0 y 25% de defectos durante las pruebas, el 27% de las empresas detecta entre 25 y 50% de defectos, el 16% de las empresas encuentra defectos entre 50 y 75% y un 8% de empresas detecta entre 75 y 100% los defectos durante las pruebas y un 5% desconoce estos valores.

La acción más común que toman las empresas cuando detectan demasiados defectos es modificar el tiempo planificado (68%), el 40% de las empresas asignan más personal, el 6% no hace modificaciones y un 10% toma otras medidas cuando esto ocurre.

Mantenimiento

Los tres factores que sobresalen como los que más afectan el costo de mantenimiento según los entrevistados son: en primer lugar la rotación del personal (73%), La falta de documentación (71%), y las pruebas insuficientes (71%).

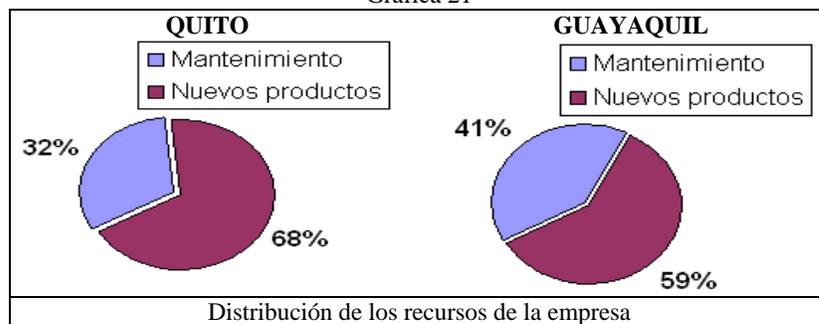
Grafica 20



Distribución de los recursos

En promedio, las empresas destinan el 66% de sus recursos al desarrollo de nuevos productos y el 34% a la actividad de mantenimiento. La grafica XVI muestra la distribución de los recursos totales de empresa por ciudad.

Grafica 21

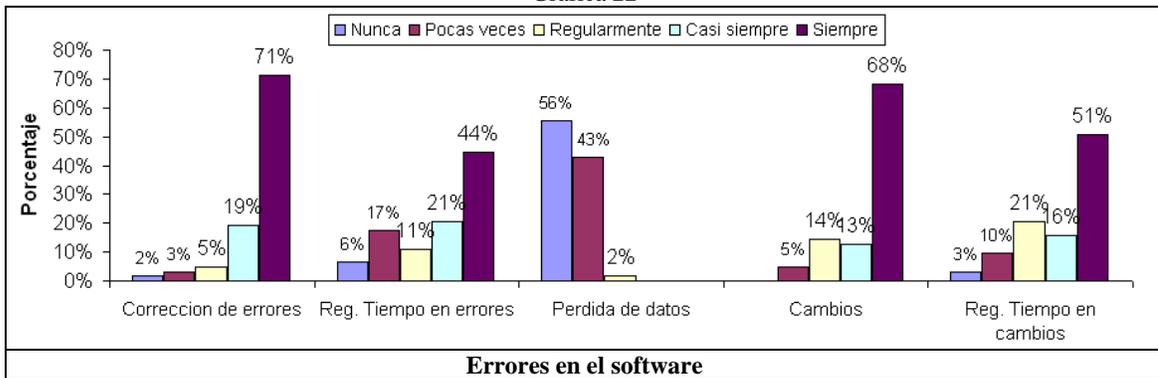


Errores en el software

Los datos del cliente pocas veces o nunca se han dañado o perdido por fallas remanentes en el software, aunque no se ha tomado en cuenta otros aspectos importantes que son afectados cuando el software no funciona bien, por ejemplo el tiempo que no se puede usar el software, o el stress que le causa al usuario.

Para arreglar algún mal funcionamiento del software, en promedio el 35% de las veces las empresas realiza reinstalaciones parciales y un 25% de las veces reinstalan completamente el software. Es decir cuando el software es modular solo instalan el componente afectado, pero si el software no es modular tienen que instalarlo completamente.

Grafica 22



Tipo de mantenimientos

Finalmente, se pregunto por los tipos de mantenimiento que se hacían en las empresas, observándose que el perfectivo (aquel que agrega o mejora la funcionalidad al sistema) es el principal tipo de mantenimiento realizado, seguido por el correctivo (aquel que repara las fallas en el software).

CONCLUSIONES

1. El crecimiento de empresas exportadoras es alentador, pues el 39% de las empresas que actualmente no exportan esperan hacerlo en el corto plazo.
2. El principal motivo que aducen las empresas para no documentar el proceso de desarrollo, es la premura por terminar los proyectos. Pero el no contar con un registro de las actividades realizadas durante el proyecto influye negativamente tanto al proyecto en curso como a futuros desarrollos, ya que al no tener una referencia de cómo fueron construidos, se tiene que invertir mas tiempo y esfuerzo en reconocer sus diferentes componentes.
3. El modelo cascada es utilizado mayormente para los proyectos pequeños y medianos, mientras que para los grandes se utiliza el modelo espiral. Destacándose el uso de la metodología programación extrema en todos los tipos de proyectos, a pesar de ser relativamente nueva.
4. La actividad que necesita más tiempo, tiene los mayores costos y ocupa la mayor cantidad de recursos es la codificación y pruebas de unidades. Además la fase de especificación y requerimientos fue identificada como la principal fuente de defectos. Esto se manifiesta sin importar que el proyecto sea pequeño, mediano o grande.
5. La mayoría de las empresas, no conocen las métricas de software y por tanto sus beneficios, además un escaso 68% de las empresas realizan mediciones del proceso de desarrollo pero solo un 24% de estas siempre toma de decisiones usando estos datos.
6. El principal problema al recibir requerimientos es el desconocimiento del cliente del alcance del proyecto, seguido de los requerimientos incompletos. A pesar de esto, la mayoría de las empresas hacen la estimación de factores como el tiempo, costos, recursos y tamaño del sistema y después de la fase de diseño hacen nuevos estimados de los factores antes mencionados
7. Prácticamente todas las empresas reutilizan el software para crear nuevos proyectos siendo las librerías y los objetos los principales componentes reutilizados.
8. Los costos del mantenimiento se incrementan principalmente por la rotación del personal, la falta de documentación y las pruebas insuficiente. Esto implica que se debe seguir trabajando el mismo proyecto después de ser liberado, cuando se debería aprovechar este tiempo en investigaciones o nuevos desarrollos.

9. Como se esperaba, el software entregado al cliente posee defectos, aunque en mínima cantidad los mismos que son corregidos al ser reportados y no han tenido un impacto negativo sobre los datos de los clientes.
10. En promedio el 66% de los recursos totales de la empresa están asignados a las actividades de mantenimiento. Siendo el preventivo y correctivo los tipos de mantenimiento mayormente realizados.

REFERENCIAS:

- [1] David Garmus. (2001). Function point analysis, Measurement practice for successful software projects. Addison Wesley.
- [2] ESPOL – VLIR, Componente 8 Ingeniería de Software. (Octubre, 2003). Estudios estadístico exploratorio de las empresas desarrolladoras de software asentadas en Guayaquil, Quito y Cuenca. <http://www.jornadasis.espol.edu.ec/IJornadas/> (Última visita: Febrero 2005)
- [3] ESPOL – VLIR, Componente 8 Ingeniería de Software. Estudio de las herramientas de desarrollo de software utilizadas por las empresas desarrolladoras de software en Guayaquil y Quito. <http://www.jornadasis.espol.edu.ec/IJornadas> (Última visita: Febrero 2005)
- [4] Cem Kaner and Walter P. Bond. (2004). Software Engineering Metrics: What Do They Measure and How Do We Know? <http://www.kaner.com/pdfs/metrics2004.pdf> (Última visita: Marzo 2005)
- [5] Hitoshi Kume. (2002). Herramientas estadísticas básicas para el mejoramiento de la calidad. Grupo Norma.
- [6] Ian Sommerville. (2002). Ingeniería de software, 6ta edición. Addison Wesley.
- [7] Juan Leyva. Artículos, tutoriales, proyectos y aplicaciones desarrolladas http://www.programacion.com/blogs/84_metrics_web/archive/524_fundamentos_de_extreme_programming_parte_i.html (Ultima visita Julio 2005).
- [8] Escuela Superior de Informática de Ciudad Real. Mantenimiento del software <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema16.pdf> (Última visita Marzo 2005)
- [9] Robert B. Grady. (1992). Practical Software Metrics for project management and process improvement, Prentice Hall.
- [10] Roberto Hernández Sampieri, Carlos Fernández Collado, Pilar Baptista Lucio. Metodología de la investigación.
- [11] Robert E. Park, Wolfhart B. Goethert, William A. Florac. Goal-Driven Software Measurement - A guide book. Software Engineering Institute. August 1996. <http://www.sei.cmu.edu> (Última visita Marzo 2005)
- [12] Center for Software Engineering. Software Metrics Guide http://sunset.usc.edu/classes/cs577b_2001/metricsguide/metrics.html (Última visita Marzo 2005)
- [13] Víctor Betancourt
Identificación de oportunidades (IO) para la IBM software development platform – v11. Febrero 2005
- [14] Extreme programming. The XP philosophy. <http://www.extremeprogramming.org/Kent.html> (Última visita Marzo 2005)