

Ingeniería de Software en Paraguay

Ellen Lujan Méndez Xavier

Facultad Politécnica-Universidad Nacional de Asunción,
San Lorenzo, Paraguay
emendez84@gmail.com

Resumen

A varias décadas, la Ingeniería del Software ha ido consolidándose como una rama importante en la informática en busca de métodos de desarrollo y técnicas que permitan producir software de calidad. Según la definición del IEEE, la Ingeniería del Software es "un enfoque sistemático y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software: es decir, la aplicación de la ingeniería del software"

En este documento se presenta un análisis del sector de desarrollo de software en 20 empresas de Paraguay, así también como los conceptos básicos de los temas que fueron analizados en cada empresa. Para la recolección de informaciones fueron distribuidos cuestionarios que luego fueron analizados según los resultados obtenidos. Se puede observar que la industria de desarrollo cada día va implementado los conceptos de Ingeniería de Software para mejorar su proceso de desarrollo.

Palabras claves: Ingeniería de Software, Paraguay.

1 Introducción

La **Ingeniería de Software** (SE - *Software Engineering*) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software". Se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software permitiendo elaborar consistentemente productos correctos, utilizables y costo-efectivos"[3]

El estudio de mercado The Chaos Report realizado por Standish Group Internacional, Inc [14] en 1996, reportó que sólo un 16% de los proyectos de software son exitosos (terminan dentro de plazos y costos y cumplen los requerimientos acordados), 53 % sobrepasa costos y plazos y cumple parcialmente los requerimientos; y el porcentaje restante ni siquiera llega a término.

La investigación y el desarrollo de técnicas y métodos de ingeniería del software son constantes en la actualidad. Sin embargo, es común que en la práctica diaria profesional no se incluya prácticamente ninguna de las recomendaciones más elementales de la ingeniería del software.

A partir de esto, fueron creados modelos de evaluación (internacionales o regionales) del proceso de producción del software los cuales confirman el estado caótico del proceso de desarrollo de software [10].

A continuación se presenta la organización del documento:

- En la sección 2 se presenta algunos conceptos básicos que se tuvieron en cuenta para el desarrollo de los cuestionarios de evaluación.
- En la sección 3 se puede observar los resultados obtenidos con los cuestionarios recibidos.
- En la sección 4 se presentan las conclusiones obtenidas a partir del análisis de los cuestionarios respondidos y la visión de nuestra realidad.

2 Conceptos importantes

En esta sección se presenta algunos conceptos que se tuvieron en cuenta para la recopilación de datos en los cuestionarios. Estos conceptos abarcan uso de modelo de ciclo de vida, planificaciones, documentación de procesos, gestión de riesgos, entre otros.

2.1 Modelo de ciclo de vida

La finalidad de todo proyecto ingeniería de software es la obtención de un producto de calidad. Para ello cada proyecto pasa por una serie de actividades. Generalmente estas actividades se agrupan en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de fases requeridas para el desarrollo de software se denomina **ciclo de vida**.

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas.

Un modelo de ciclo de vida del software:

- Describe las fases principales de desarrollo de software.
- Define las fases primarias esperadas de ser ejecutadas durante esas fases.
- Ayuda a administrar el progreso del desarrollo, y
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Actualmente existen variadas propuestas de modelos de ciclo de vida que pueden adaptarse a diversos proyectos según sus características principales, como por ejemplo tamaño del proyecto, alcance, etc.

Algunos modelos de ciclo de vida más comunes son:

- Modelo de cascada puro y modificado
- Modelo en espiral
- Entrega por etapas

2.2 Estimación de proyectos

En [7] se define estimación como la predicción más optimista con una probabilidad distinta de cero de ser cierta.

Se puede estimar el tamaño de un proyecto de varias formas [7]:

- Utiliza un enfoque algorítmico, como los puntos de función, para estimar el tamaño del programa a partir de las prestaciones.
- Utilizar un software de estimación para estimar el tamaño del programa, a partir de la descripción de las prestaciones del programa (pantallas, diálogos, archivos, tablas de la base de datos, etc)
- Si ya se ha trabajado en proyectos similares, y se conoce el tamaño, estimar cada una de las partes principales del nuevo sistema como un porcentaje del tamaño de una parte similar del sistema antiguo. Estimar el tamaño total del sistema nuevo sumando los tamaños estimados de cada una de las partes.

2.3 Planificación de proyectos

El objetivo de la Planificación del proyecto de Software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

El Objetivo de la planificación se logra mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables.

2.4 Gestión de Riesgos

La función de la gestión de riesgos del software es identificar, estudiar y eliminar las fuentes de riesgos antes de que empiecen a amenazar la finalización satisfactoria de un proyecto software.

Generalmente, la gestión de riesgos se dividen en valoración de riesgos y control de riesgos.

La valoración o estimación de riesgos se compone de [7]:

- **Identificación de riesgos:** genera una lista de riesgos capaces de romper la planificación del proyecto
- **Análisis de riesgos:** mide la probabilidad y el impacto de cada riesgo, y los niveles de riesgo de los métodos alternativos.
- **Asignación de prioridades de riesgos:** genera una lista de riesgos ordenados por su impacto

El control de riesgos se compone de [7]:

- **Planificación de la gestión de riesgos:** genera un plan de tratar cada riesgo significativo. También asegura que los planes para la gestión de riesgos de cada uno de los riesgos individuales son consistentes entre sí y con el plan de proyecto
- **Resolución de riesgos:** es la ejecución del plan para resolver cada uno de los riesgos significativos
- **Monitorización de riesgos:** es la actividad del progreso de la monitorización dirigido a la resolución de cada elemento del riesgo

2.5 Ingeniería de Requerimientos

De las muchas definiciones que existen para requerimiento, ha continuación se presenta la definición que aparece en el glosario de la IEEE [5].

(1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. (3) Una representación documentada de una condición o capacidad como en (1) o (2).

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema es llamado Ingeniería de Requerimientos. La meta de la ingeniería de requerimientos (IR) es entregar una especificación de requisitos de software correcta y completa [11].

Existen varias técnicas para la IR, donde cada técnica puede aplicarse en una o más actividades de la IR; en la práctica, la técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose[4].

Algunas técnicas de IR son:

- **Entrevistas y cuestionarios:** se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el encuestador conversa con el encuestado; el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema.
- **Prototipos:** Los prototipos permiten al desarrollador crear un modelo del software que debe ser construido. Al igual que todos los enfoques al proceso de desarrollo del software, el prototipado comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un “diseño rápido”
- **Casos de uso:** Los casos de uso son una técnica para especificar el comportamiento de un sistema: “Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios.” Esta técnica se basa en un concepto muy interesante: al definir requerimientos, es importante describir al sistema desde el punto de vista de aquél que lo va a usar, y no desde el punto de vista del que lo va a construir. De esta forma, es más fácil validar que los requerimientos documentados son los verdaderos requerimientos de los usuarios, ya que éstos comprenderán fácilmente la forma en la que están expresados.

2.6 Metodologías Ágiles

En 2001, tras una reunión celebrada en Utah-EEUU, se reunieron para debatir los puntos que tenía en común y el resultado fue el Manifiesto Ágil [2]. Este manifiesto establece las ideas principales de las metodologías ágiles:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Hay diversos métodos ágiles que recogen estas ideas como [2]:

- **eXtreme Programming (XP):** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y

coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

- **SCRUM:** es una metodología para el desarrollo ágil de productos, expuesta por Hirotaka Takeuchi e Ikujiro Nonaka, en el artículo *The New New Product Development Game* en el que ponen de manifiesto que:
 - El mercado competitivo de los productos tecnológicos, además de los conceptos básicos de calidad, coste y diferenciación, exige también rapidez y flexibilidad.
 - Los nuevos productos representan cada vez un porcentaje más importante en el volumen de negocio de las empresas.
 - El mercado exige ciclos de desarrollo más cortos.

Estas metodologías ponen de relevancia que la capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan. Nos lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significa reducir su coste [2].

2.7 Modelos de Procesos

Actualmente existen modelos de procesos así también como modelos que permiten evaluar los procesos de una organización. Estos modelos pueden ser a nivel internacional o también a nivel regional, ya que diversos países han realizado su propio modelo según la realidad del mercado local.

2.7.1 CMMI - Capability Maturity Model Integration.

CMMI es un enfoque de mejora de procesos que proporciona organizaciones con los elementos esenciales de procesos efectivos. Se puede utilizar para indicar mejoras de procesos a través de un proyecto, una división, o una organización entera. CMMI ayuda a integrar las funciones de la organización tradicionalmente separadas, poner las metas de mejora del proceso y prioridades, proporcionar la guía para procesos de calidad, y proporcionar un punto de referencia para valorar los procesos actuales [12].

2.7.2 ISO/IEC 15504 – SPICE

ISO/IEC 15504 es el estándar internacional que establece un marco para la evaluación y mejora de los procesos de desarrollo y mantenimiento de software.

En enero de 1993 la comisión ISO/IEC JTC1 aprobó un programa de trabajo para el desarrollo de un modelo que fuera la base de un futuro estándar internacional para la evaluación de los procesos del ciclo de vida del software. Este trabajo recibió el nombre de proyecto SPICE (Software Process Improvement and Capability dEtermination), y en junio de 1995, con la publicación de su primer borrador, desde ISO fueron invitadas diferentes organizaciones para aplicarlo y valorar sus resultados [6].

La instrucción técnica consta de 9 apartados, recogidos en volúmenes independientes, que se han ido publicando como redacción definitiva del estándar internacional ISO/IEC 15504 durante el periodo 2003-2005.

2.7.3 MPS/BR - Modelo de Proceso del Software /Brasil

El MPS.BR está en desarrollo desde diciembre de 2003 y es coordinado por la Asociación para Promoción de la Excelencia del Software Brasileiro (SOFTEX), con el apoyo del Ministerio de Ciencias y Tecnología (MCT), de la financiadora de Estudios y Proyectos (FINEP) y del Banco Interamericano de Desarrollo (BID) [13].

La base técnica para la construcción de este modelo de mejora y evaluación del proceso de software es compuesta por las normas NBR ISO/IEC 12207 (Proceso de Ciclo de Vida del Software), por las enmiendas 1 y 2 de la norma internacional ISO/IEC 12207 y por la ISO/IEC 15504 - Evaluación del Proceso (también conocida por SPICE: *Software Process Improvement and Capability dEtermination*), por lo tanto, el modelo está en conformidad con esas normas. Este modelo también cubre el contenido del CMMI-SE/SWSM, a través de la incorporación de procesos y resultados [13].

2.7.4 MoProSoft – Modelo de Procesos de Software y EvalProSoft – Evaluación de Procesos de Software

MoProSoft es un modelo de procesos para la industria de software mexicana, que fomenta la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permite elevar la capacidad de las organizaciones que desarrollan o mantienen software para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad. Es también aplicable en áreas internas de desarrollo de software de las empresas de diversos giros [8].

La evaluación de este modelo se realiza a través del Método de Evaluación, EvalProSoft, aplicada a las organizaciones dedicadas al desarrollo y/o mantenimiento de software [9].

La capacidad de proceso se evalúa en una escala de 0 a 5. El valor cero se asocia al nivel de capacidad más bajo, y significa que no se alcanza el propósito del proceso. El valor 5 se asocia al nivel de capacidad más alto y significa que se logran las metas de negocio actuales y proyectadas a través de la optimización y mejora continua del proceso [9].

2.7.5 Modelo Cóndor

El modelo Cóndor está formado por una arquitectura bidimensional donde una dimensión corresponde al mejoramiento de procesos y la otra a las evaluaciones de procesos. El mejoramiento de procesos corresponde a 5 fases y 18 áreas clave de procesos para determinar la madurez, que se denominan KPA (*Key Process Areas*) [1].

3 Análisis de mercado y resultados

El análisis de mercado fue realizado a través de la distribución de un total de 35 cuestionarios con preguntas varias que abarcan temas relacionados con los conceptos desarrollados anteriormente entre empresas que se dedican al desarrollo así también como a empresas que poseen departamentos para el desarrollo de software.

En la figura 1 se presenta los porcentajes correspondientes a los cuestionarios recibidos y no respondidos. El 57% de los cuestionarios remitidos fueron recibidos mientras que el 43% de los cuestionarios remitidos no fueron respondidos.



Figura 1. Cuestionarios recibidos y no respondidos

A continuación se presentan los resultados correspondientes a cada pregunta que se presenta en el cuestionario

3.1 Pregunta 1: *Que modelo de ciclo de vida es utilizada en el proceso de desarrollo de software?*

En la figura 2 se observan de los porcentajes de utilización de modelos de ciclo de vida. El modelo de entrega por etapas obtiene el mayor porcentaje de utilización debido a la facilidad que este presenta para implementarse en el mercado nacional. Además de los modelos presentados son utilizados también la metodología *Microsoft Solution Framework - MSF* y el modelo presentado para el *Proceso Unificado- UP*. Algunas compañías utilizan más de un modelo de ciclo de vida dependiendo con la complejidad del proyecto.

Se puede observar que el porcentaje de empresas que se dedican a la codificación directa (considerado como ninguno) es igual al porcentaje de empresas que utilizan el Modelo Cascada. Cabe resaltar también que en algunas bibliografías este proceso de codificación directa es considerado también como un ciclo de vida denominado "Codificar y corregir" [7]

Este tipo de desarrollo es recomendable para proyectos muy pequeños donde la complejidad es mínima, pero actualmente es aplicado en proyectos medianos o grandes los cuales puede acarrear a una mala calidad del software desarrollado.

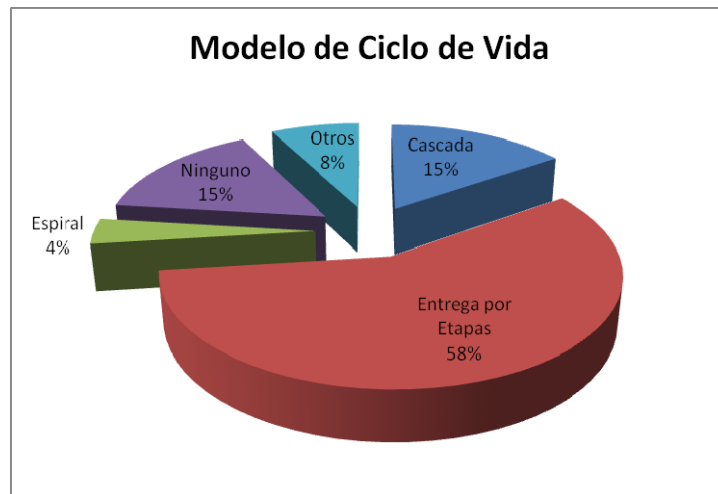


Figura 2. Utilización de modelos de ciclo de vida

3.2 Pregunta 2: De que forma se realiza la estimación del tamaño de un proyecto?

– Pregunta 3: De utilizar un enfoque algorítmico o software de estimación ¿cuál utiliza?

En la figura 3 se puede observar que la estimación del tamaño de software se realiza de acuerdo con la experiencia de las personas con respecto a proyectos similares, así como también se puede observar un gran porcentaje que no utiliza ningún enfoque para la estimación de tamaño. Esta falta de estimación de tamaño puede ser la causa de los atrasos en los proyectos. Otros enfoques utilizados es la estimación a través de la definición de requerimientos de usuario.

Con respecto al enfoque algorítmico utilizado todas las empresas encuestadas señalaron al COCOMO II como principal modelo.

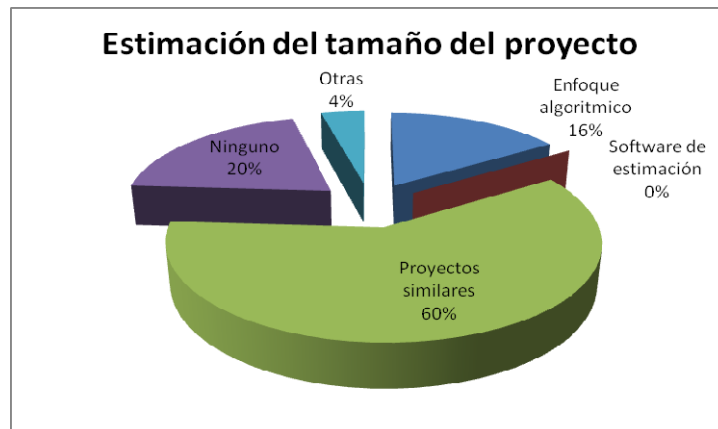


Figura 3. Estimación del tamaño del proyecto

3.3 Pregunta 4: Se realiza algún tipo de planificación o plan de proyecto al inicio del mismo?

En la figura 4 se observan los resultados correspondientes a la formulación de plan de proyecto al inicio del mismo. Un punto importante en el gráfico es el porcentaje de las empresas encuestadas que no realizan una planificación de proyecto debido a que la planificación al inicio, así como la estimación del tamaño son puntos muy importantes en la parte inicial de los proyectos.

Esta falta de planificación puede ser un punto muy importante para el desarrollo del proyecto así también como para determinar la calidad del software.

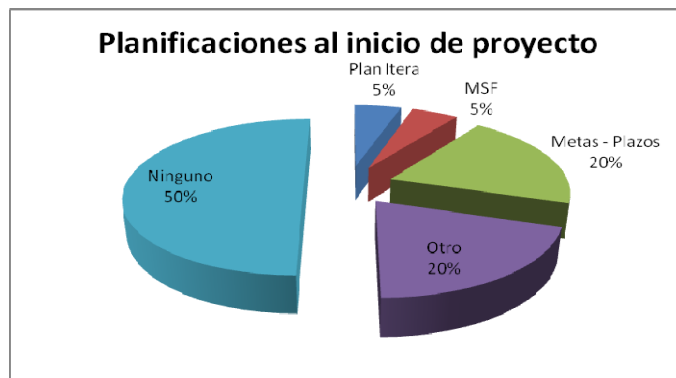


Figura 4. Realización de planificaciones al inicio del proyecto

3.4 Pregunta 5: ¿Que herramienta de Planificación de Proyecto utiliza? Porque?

Entre las herramientas de planificación más utilizadas se cita al MS-Project, elegido por simplicidad y facilidad de uso debido a las prestaciones que este presenta como facilidad en la administración y seguimiento de proyectos.

Aquí nuevamente se puede observar en la figura 5 un porcentaje del 57% de empresas que no utilizan ninguna herramienta de planificación. Este resultado va muy relacionado con el anterior debido al porcentaje de empresas que no realizan planificaciones.

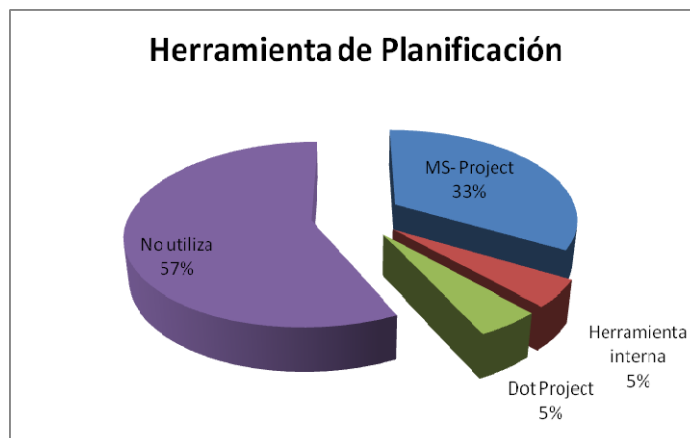


Figura 5. Herramientas de planificación de proyectos

3.5 Pregunta 6: ¿Que puntos incluye la gestión de riesgos?

Con respecto a la gestión de riesgos, la figura 6 muestra un gran porcentaje de realización tanto de estimación y control, el 50%, mientras que existen otros porcentajes menores que realizan solo uno de estos puntos.

Nuevamente podemos ver un gran porcentaje de empresas que no realizan ningún tipo de gestión sobre los riesgos, los cuales pueden afectar al desarrollo del proyecto dado que ante la aparición de un riesgo de alta prioridad puede causar retrasos en el proyecto debido a la falta de control sobre ellos.

Una buena gestión de riesgos puede prevenir desastres durante el proyecto los cuales deben ser bien identificados y analizados para definir una buena resolución de los mismos.

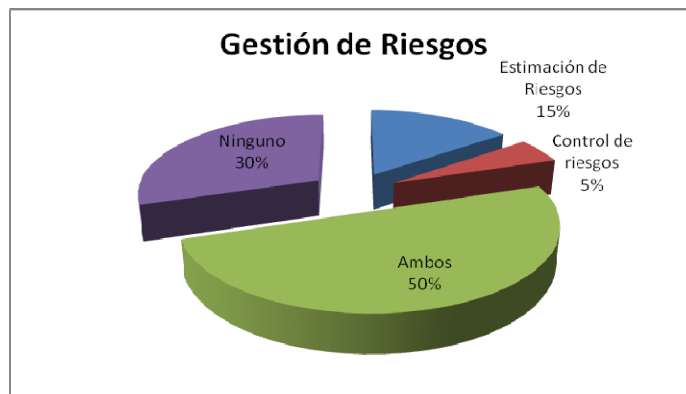


Figura 6. Gestión de riesgos

3.6 Pregunta 7: Que técnica de levantamiento de requerimiento es utilizada?

Otro punto muy importante en el desarrollo de proyectos es la forma de levantamiento de requerimientos de usuarios. En la figura 7 se observa que la técnica más popular es la Entrevista con un porcentaje del 50% de los cuestionarios. Este de tipo de técnica es muy adecuada debido al involucramiento del cliente en el proceso el cual es muy de los puntos más importante para el éxito del proyecto. Los porcentajes siguientes correspondientes a las técnicas de prototipado, cuestionarios y utilización de casos de usos tienen casi los mismos porcentajes de utilización. En este punto las empresas también utilizan varias técnicas de acuerdo al proyecto o al tipo de cliente.

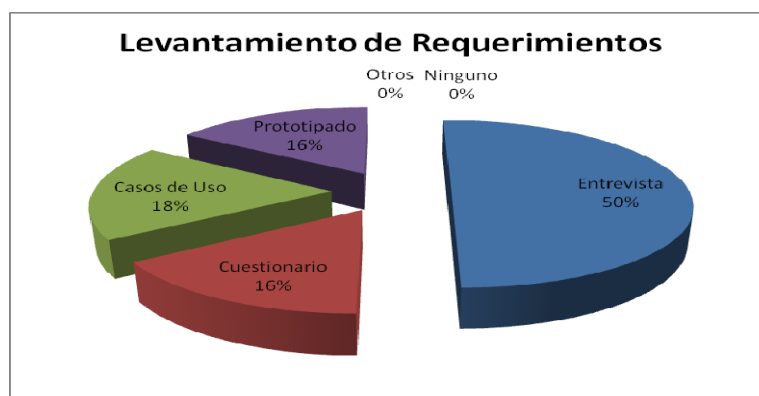


Figura 7. Tecnicas de levantamiento de requerimientos

3.7 Pregunta 8: Indique porcentajes aproximados de la duración del proyecto total para cada etapa

En la figura 8 se presentan los porcentajes obtenidos en cada fase del desarrollo de software. En la primera figura se presenta el modelo básico de 5 fases, mayormente utilizado en las instituciones de nuestro mercado: Análisis, Diseño, Implementación, Prueba y Mantenimiento. En la segunda figura se presenta las fases del modelo utilizado por Proceso Unificado: Inicio, Elaboración, Construcción y Transición. Los porcentajes obtenidos fueron promediados de acuerdo a los resultados proporcionados. Los mayores porcentajes obtenidos se encuentran en las fases de implementación o construcción.

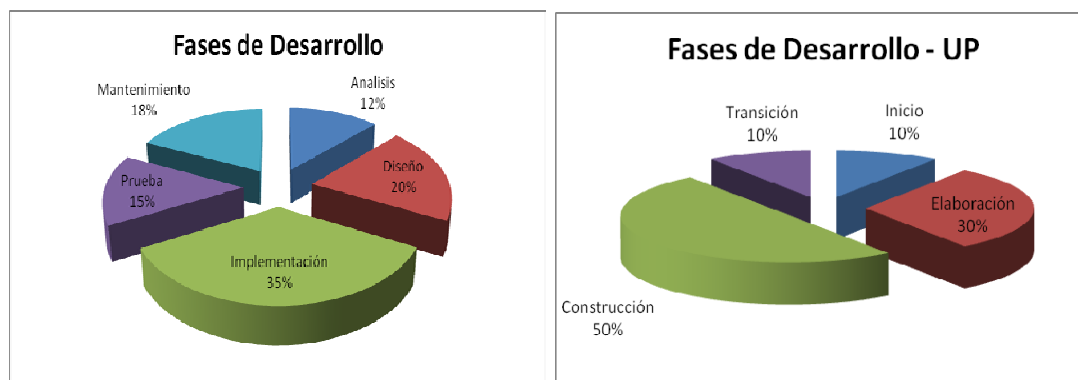


Figura 8. Porcentajes de fases de desarrollo

3.8 Pregunta 9: Utiliza metodología ágil para el desarrollo rápido de proyectos?

Con respecto a la utilización de metodologías ágiles, la figura 9 indica la popularidad del eXtreme Programming, mientras que también existe un gran porcentaje de empresas que no utilizan metodologías de desarrollo rápido.

Este tipo de metodologías se caracterizan por intentar evitar los burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Una buena elección de este tipo de metodologías podría ser aplicada en los proyectos locales, pero actualmente un 38% de las empresas encuestadas utiliza este tipo de metodología.

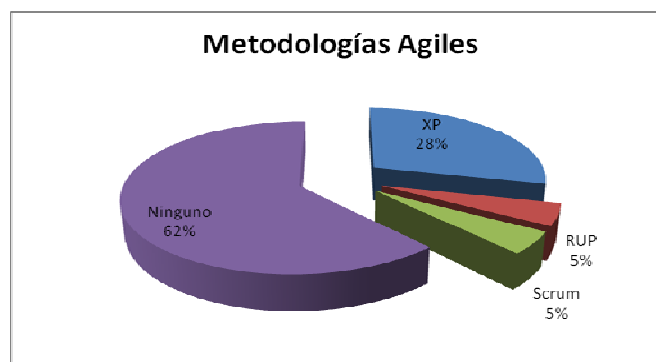


Figura 9. Utilización de Metodologías Ágiles

3.9 Pregunta 10: Se aplican conceptos de procesos en el desarrollo de proyectos?

- Pregunta 11: Los procesos aplicados están debidamente documentados?

Los conceptos de procesos son aplicados en un 55% de las instituciones, y en la mayoría de estas, la documentación correspondiente a estos procesos no se encuentra diseñada.

La documentación de los procesos es un punto muy importante a la hora de solicitar cualquier tipo de certificación tanto internacional como para modelos que actualmente rigen a nivel regional.

Los resultados obtenidos demuestran que la mayoría de las instituciones nacionales no posee documentaciones de los procesos realizados, los cuales podría mejorar la calidad y eficiencia de los procesos realizados.

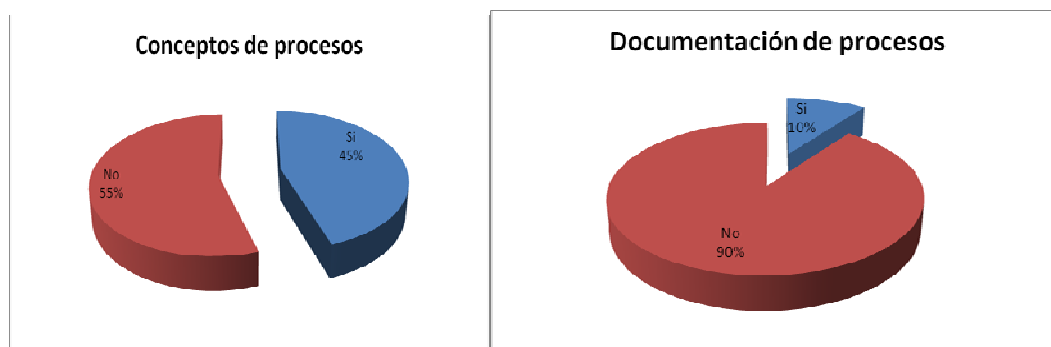


Figura 10. Conceptos de procesos y documentación de estos

3.9 Pregunta 12: Posee conocimientos básicos sobre algunos de estos modelos?

- Pregunta 13: Existe algún plan de calidad en el desarrollo de software a llevar a cabo a mediano o corto plazo?

Los conocimientos básicos de algunos de los modelos de procesos citados en la sección anterior se encuentran presente en los directivos de estas y existen instituciones con planes de calidad relacionados con la obtención de certificaciones internacionales a mediano y a corto plazo.

Mirando la realidad de nuestras empresas se puede observar que actualmente conseguir una certificación se encuentra en los planes de algunos directivos, pero en su mayoría la puesta en practica es muy difícil debido a la organización y falta de documentación que poseen un porcentaje muy elevado.

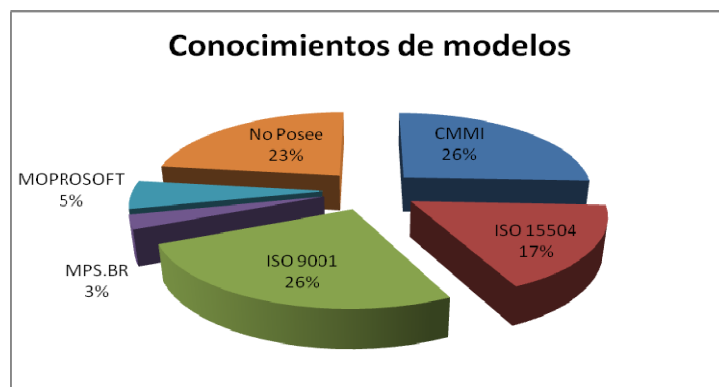


Figura 11. Conocimientos de modelos de procesos.

4 Conclusiones

Las posibilidades reales de aplicación de las técnicas de ingeniería del software quedan limitadas por los profesionales que dirigen los proyectos en general.

Es necesario que el ingeniero de software sea, ante todo, ingeniero y que sea capaz de trasladar con sentido práctico los conocimientos científicos de la informática al desarrollo y mantenimiento de software.

En nuestro mercado local se puede observar que con el pasar del tiempo, las empresas buscan la calidad en los productos para la diferenciación requerida ante la competencia que existe.

Un punto importante de destacar es el interés de algunas instituciones hacia modelos orientados a procesos o certificaciones de calidad, así también como técnicas de levantamiento de requerimientos para una mejor comunicación con los usuarios.

En contra partida, debido al poco interés que se observa en las realizaciones de documentaciones así como en las planificaciones, estas certificaciones de calidad son más difíciles de obtenerse en el país.

Se puede observar también un porcentaje muy bajo en la utilización de metodologías ágiles o variantes de metodologías exitosas en el mercado mundial, las cuales podría ser una buena opción para la optimización del desarrollo de software de algunas empresas.

Con respecto a la planificación, se observa la falta de planificación debido a un gran porcentaje que no realiza este proceso que puede ser muy importante en proyectos mediados o grandes.

Actualmente la ingeniería de software nos provee de herramientas que pueden ayudarnos en la organización de proyectos para la obtención de productos de calidad. Queda a responsabilidad de los jefes de proyecto la buena utilización y elección de técnicas que puedan ser de utilidad a nuestras necesidades.

5. Referencias

- [1] Bedini, Alejandro. *Condor: Una propuesta de modelo de Proceso/Madurez/Mejoramiento de Software de Aplicarse en Unidades Informáticas Inmaduras*
- [2] Canós, Jose H; Letelier, Patricio, Penades, M. Carmen - “*Metodologías Ágiles en el Desarrollo del Software*”- Universidad Politécnica de Valencia
- [3] Cota A. 1994 "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13.
- [4] Hofmann, Hubert. “Requirements Engineering”. Institute for Informatics – University of Zurich. 1993.
- [5] IEEE Std. 610.12-1990
- [6] ISO/IEC 15504 - *Software Process Improvement and Capability dEtermination* – Año 1998
- [7] McConnel, Steve – “Rapid development” – McGraw Hill – pp. 93, 188 – 217
- [8] Oktaba, Hanna. *Modelo de Procesos para la Industria del Software –MoProSoft*. Versión 1.3 Agosto/2005
- [9] Oktaba, Hanna. *Método de Evaluación de Procesos para la industria del software – EvalProSoft*. Versión 1.1 Marzo/2004
- [10] Paulk, M.C., García, G.M., Chrissis, M.B. y Bush, M., Capability maturity model for software, versión 1.1 CMU/SEI-93-TR-24, Software Engineering Institute y Universidad Carnegie Mellon, febrero, 1993.
- [11] Saiedian, H.; Dale, R. “Requirements Engineering: Making the connection between the software developer and customer”. Department of Computer Science – University of Nebraska. 1999.
- [12] SEI. Software Engineering Institute, Carnegie Mellon - <http://www.sei.cmu.edu/cmmi/>
- [13] SOFTEX. *MPS.BR- Melhoria de Processo do Software Brasileiro*. Guía General V1.1-Maio/2006.
- [14] Standish Group International - <http://www.standishgroup.com/>