udp Escuela de Informática y Telecomunicaciones
FACULTAD DE INGENIERÍA

Facultad de Ingeniería y Ciencias
Escuela de Informática y Telecomunicaciones

Metodologia RUP

Christian López
Thomas Muñoz
Flavio Pallini
christian.lopeza@mail.udp.cl
thomas.munoz@mail.udp.cl
flavio.pallini@mail.udp.cl

29 de agosto de 2017

Índice general

Índice de figuras

1. Introducción

Según la IEEE, "la ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software."Bajo este concepto, si bien es posible construir software sin seguir un protocolo para la planificación de su desarrollo, resulta mucho más conveniente y seguro utilizar alguna metodología aplicada de la ingeniería como pauta a seguir, a fin de modelar la solución a un problema en una empresa que pueda ser resuelto con la implementación o mejora de un nuevo software. De esta forma se gestionan múltiples variables y procesos esperados de forma más eficiente y con un riesgo de error menor.

Por lo general, las metodologías de trabajo se basan en un conocimiento de las necesidades a suplir por el proyecto, la determinación de su alcance, su construcción, y su evaluación como respuesta al problema original. Dependiendo de la metodología usada, estas etapas pueden sucederse de forma lineal, cíclica, en ese mismo o distinto orden, dependiendo de lo que el proyecto amerite. Así, en este informe, se detallará la metodología RUP, o *Rational Unified Process*.

2. Explicación de Metodología

La metodología RUP es un proceso de desarrollo de software creado por *Rational Software*, y actualmente propiedad de IBM. Es una de las metodologías de desarrollo de software más utilizadas para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por no ser estática, sino compuesta por un conjunto de fases adaptables según la necesidad del cliente o avance del equipo de desarrollo.

RUP es una forma de UP (textitUnified Process), una metodología caracterizada por la planificación basada en arquitectura y casos de uso. Esta metodología trabaja con iteraciones, haciendo a RUP una forma de desarrollo incremental. Cada iteración incluye los roles que se van a ver involucrados, las labores que se van a realizar, el componente o producto que van a producir, y un ligero análisis del riesgo involucrado. Sin embargo, estas iteraciones se diseñan tal que puedan dividir el proceso de desarrollo en cuatro grandes fases, donde cada una incluye en mayor o menor medida el modelamiento del negocio, análisis, diseño, construcción, pruebas e implantación. Lo que esto permite es un desarrollo maleable donde nuevos desafíos o requerimientos pueden ser adheridos al proyecto, y donde el producto final necesariamente debe haber sido compuesto por, y probado como, componentes interdependientes, lo cual permite probar más de cerca la calidad del código.

2.1. ¿Por qué utilizar RUP?

RUP proporciona información sobre lo que puede esperarse de la tarea de desarrollo. Ofrece un glosario de terminología y una enciclopedia de conocimiento que le ayuda a comunicar sus necesidades de forma eficaz al equipo de desarrollo de software.

Para un gestor o jefe de equipo, RUP proporciona un proceso el cual le permite comunicarse de forma eficaz con el personal, gestionar la planificación y el control de su trabajo. Para un Diseñador, RUP proporciona una buena base de arquitectura y una gran cantidad de materiales con las que construir una definición de un proceso, lo que le permite configurar y ampliar dicha base como desee.

2.2. ¿Cuándo debo utilizar RUP?

Se utiliza RUP desde el inicio de un proyecto de software, y puede seguir utilizándolo en los ciclos de desarrollo subsiguientes tiempo después de que le proyecto inicial haya terminado.

La forma de utilizar RUP varía para ajustarse a sus necesidades. Existen unas pocas consideraciones que determinarán cuando y como utilizar partes diferentes de RUP.

- Ciclo vital del proyecto(numero de iteraciones, longitud de cada fase)
- Propósitos empresariales, visión, ámbito y riesgo del proyecto
- Tamaño del esfuerzo de desarrollo de software

En la siguiente figura 2.1 muestra a través de un gráfico de complejidad de técnica y gestión:

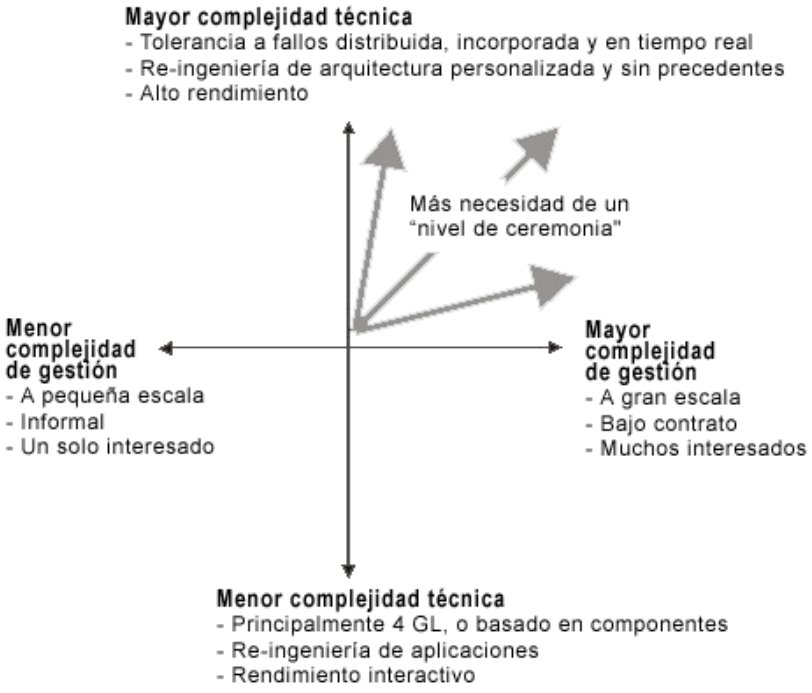

Figura 2.1: Gráfico complejidad de técnica y gestión.

2.3. Fases de la metodología RUP

Las fases indican el énfasis que se da a atributos particulares en el proyecto en un instante dado. Para capturar la dimensión temporal de un proyecto, RUP divide el proyecto en cuatro fases diferentes(ver figura 2.2):

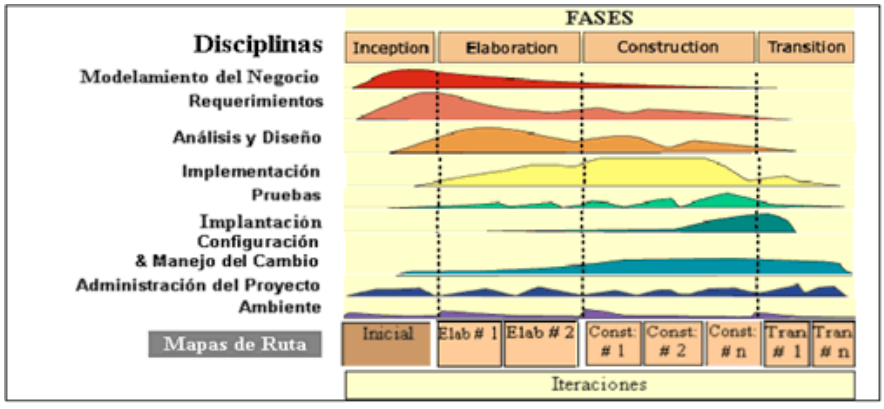

Figura 2.2: Fases de RUP.

- Iniciación o Diseño: énfasis en el alcance del sistema
- Preparación: énfasis en la arquitectura
- Construcción: énfasis en el desarrollo
- Transición: énfasis en la aplicación

2.3.1. Fase de Inicio

El objetivo preferente de esta fase es alcanzar un acuerdo entre todos los interesados respecto a los objetivos del ciclo vital del proyecto. Es muy significativa la fase de inicio, pues son más arriesgados para los requisitos y para la actividad comercial y deben abordarse antes de que el proyecto pueda continuar.

La duracion de esto es generalmente corto y se utiliza para definir si es factible seguir y definir los riesgos. Un prototipo se puede hacer para que el cliente pruebe el software.

2.3.2. Fase de elaboración

El propósito de esta fase es el establecimiento de una arquitectura de sistema para proporcionar una base estable para el diseño y la implementación en la fase de construcción. La arquitetura va evolucionando a partir de una consideración de requisitos mas significativos y una valoración a los riesgos.

Al final de la fase de elaboración se encuentra el segundo objetivo mas importante del proyecto que es el ciclo vital de la arquitectura. En esta se examina los objetivos y el ámbito del sistema detallado, la elección de arquitectura y la resolución a los principales riesgos.

2.3.3. Fase de Construcción

El objetivo de la fase de construcción es aclarar lo requisitos restantes y completar el desarrollo del sistema basándose en la base de la arquitectura. En esta fase se pone el énfasis en la gestión de los recursos y el control de la operaciones para poder optimizar los costes, la planificación y la calidad.

2.3.4. Fase de Transición

En esta fase se garantiza que el software este disponible para los usuarios finales, además se puede acarrear varias iteraciones e incluye las pruebas del producto en preparación para el release, asi como los ajustes menores basados en la información obtenida de los usuarios que han probado el software.

En este momento del ciclo vital, la información recibida de los usuarios debe centrarse especialmente en el ajuste del producto, las cuestiones de configuración, instalación y utilización.

2.4. Desventajas de RUP

Siendo una metodología iterativa donde cada iteración tiene un grado significativo de independencia, las principales desventajas de RUP tienen que ver con la integración de todas las iteraciones en un proyecto coherente. En particular, se debe poner atención especial en compatibilizar los procesos y tecnologías de cada componente entre sí. Asimismo, si bien es conveniente evaluar la calidad de los componentes por separado, no se puede obviar el control de calidad del producto completo, gracias al problema de compatibilidad ya mencionado.

Dependiendo de la granularidad de las iteraciones contra el alcance del proyecto, trabajar con RUP puede ser más desorganizado que otras metodologías. Si bien son claras las fases de RUP y éstas otorgan un nivel de orden similar a la de métodos secuenciales, existen varios proyectos que no se pueden subdividir tan óptimamente, alterando la organización. Similarmente, esto puede llevar a una construcción más difícil que con otras metodologías.

Además, si bien RUP divide el proyecto final en componentes, estos componentes no son de ninguna forma un producto finalizado presentable. Esto puede causar fricción con el cliente del proyecto, de la misma forma que las metodologías como Cascada pueden causar fricción. La falta de productos tangibles que presentar, o productos tangibles que no representan la realidad final del proyecto, puede ser problemática.

Finalmente, dividir el trabajo de esta forma puede volver mas compleja la documentación, en especial existiendo tantos documentos y artefactos comunes en RUP a usarse en la planificación del proyecto.

3. Aplicación en Canchapp

Cada iteración de RUP considera metas, roles, y componentes definidos, con documentación y artefactos usualmente específicos. Para no realizar todo el trabajo exhaustivo que esto conllevaría, se presenta un bosquejo de lo que podría realizarse en éstas.

La primera iteración sería parte de la fase de iniciación. Se buscaría especificar requisitos generales y, con gran prioridad, esbozar un plan para el trabajo siguiente en general. En el caso de Canchapp, esto podría traducirse en identificar que se va a necesitar información de mapas y GPS e información de contacto a canchas que dispongan de ésta; definir qué funcionalidades son más primordiales, tales como la implementación de formas de conexión de personas por medio de (o ligada a) la aplicación, una base de datos que registre qué canchas están en uso qué días, y si existe una forma de reservarlas. Se incluiría además un análisis del riesgo y de la prioridad de estas características.

Lo documentado en esta iteración se discutiría con los textitstakeholders, y de ser necesario se incorporarían una o más iteraciones en esta fase a fin de pulir la imagen inicial que se espera para Canchapp. Para el caso de estudio, se considerará una segunda iteración.

Luego, comenzaría la fase de elaboración. Ahora ya comenzaría en más detalle la definición de casos de uso y, por medio de UML, la documentación de la arquitectura que se espera que la aplicación contenga. Es en estas iteraciones que se especificaría qué datos de las canchas y usuarios es pertinente almacenar en bases de datos en internet, la pertinencia de trabajar o no con perfiles de usuario, o cuántas facultades se les otorgaría a los usuarios para manipular estos datos -esto es, si se les permite un grado de gestión sobre las canchas (tal como sugerir dónde hay canchas no documentadas) o si su participación se limitaría a consumir. Se debe considerar además especificaciones estéticas como el diseño de la interfaz gráfica, si una implementación minimalista o detallista, y de qué forma hacer la aplicación más intuitiva. Habiendo dado como ejemplo tres funcionalidades a definir, una de las cuales puede influir a otra, se considerarán dos iteraciones para esta fase.

A continuación, vendría la fase de desarrollo. De forma general, se trataría de seguir el plan definido dos fases atrás con la prioridad mencionada. Considerando una iteración por cada una de las funcionalidades mencionadas (teniendo en mente que seguro existen varias funcionalidades necesarias no mencionadas), una más para control de calidad de detalle y una más por algún requerimiento que pudiera surgir y evaluarse en el desarrollo de la aplicación, se podrían considerar seis iteraciones para esta fase.

Finalmente, en la fase de transición se ensamblarían los componentes desarrollados en la fase anterior, y se realizarían pruebas generales de control de calidad, terminando de pulir el producto. Para esto se puede, de forma optimista, esperar una única iteración necesaria. Además, siendo este un emprendimiento más que un producto interno de una empresa, se realizarían campañas publicitarias y de inserción en tienda, lo cual requeriría la atención de un último ciclo.

Con esto se considerarían en total doce iteraciones, de las cuales la fase de desarrollo sería notablemente la más amplia. En el ejemplo, las tres otras fases tenían largos similares, y en todos los diagramas estudiados este parecía ser el caso real.

4. Conclusión

Esta metodología si bien es una potente herramienta para el desarrollo de software, ya que permite darle valor al cliente al hacerlo parte del proceso de desarrollo obteniendo tempranamente el feedback necesario para realizar modificaciones o pasar al siguiente módulo. No es recomendable para proyectos pequeños, ya que los pasos que se deben seguir podrían ser incluso mas tediosos que el proyecto en si y de esta forma provocar que se alargue mucho más de lo necesario.