

La Metodología del Proyecto

Descripción de la Metodología del Proyecto

Lo primero que hay que decir es que el proceso de desarrollo de software del curso no es ágil. Las metodologías ágiles tienen un requisito fundamental: el desarrollador debe tener una experiencia en el área de desarrollo que le permita cumplir con sus tareas de forma dinámica y continua. Al requerir un poco menos de documentación, dichas metodologías se basan en un conocimiento claro de los procesos de análisis y diseño que, a estas alturas del curso, se asume que los estudiantes no poseen.

¿Por qué razón se asume esto? La razón es porque hay que recordar que el curso es un curso de inicio de la carrera, en la que el estudiante solamente ha llevado el curso de fundamentos de programación, y en el mejor de los casos ha llevado bases de datos. No en todos los casos el estudiante ha cursado programación orientada a objetos. El lugar donde el estudiante está en el proceso de aprendizaje general de la carrera, es importante a la hora de especificar la metodología que se va a implementar en el curso. Pero hay otro elemento también: las metodologías tradicionales son más estructuradas, exigen un mayor ritmo de documentación y un proceso de análisis y diseño documentado. Además, desde el punto de vista didáctico, UML y específicamente los casos de uso son una herramienta que ha demostrado estar entre las mejores para enseñar la interacción entre el usuario y el sistema, y la forma de identificar dichas interacciones y ponerlas en código de una forma más exacta.

Metodología incremental iterativa

Como se ve en el material de las metodologías de desarrollo, una de las metodologías de desarrollo tradicionales es la incremental iterativa. Dicha metodología divide el proceso de desarrollo en fases. Cada fase va a tener un grupo de funcionalidades que deben ser realizadas, siguiendo una serie de etapas, generalmente análisis, diseño, construcción y validación. Estas etapas se repiten dentro de cada fase (de ahí el nombre iterativo) y en cada fase se va a ir agregando una funcionalidad al producto, hasta abarcar la totalidad (de ahí el nombre incremental).

Metodología ágil

Si bien la metodología empleada no es ágil por definición, se toman herramientas que se usan en dichas metodologías para administrar las tareas, y se disminuye el tiempo total de la documentación del proyecto. En las metodología iterativa, el análisis y diseño se hace en cada iteración, mientras que en las metodología ágiles el desarrollador documenta según su criterio, o según el estándar de la organización, conforme va desarrollando las funcionalidad.

Las metodologías ágiles se basan en el concepto de sprints, que representan el rango de tiempo en el que una funcionalidad completa debe ser desarrollada. Cada desarrollador debe entregar, en ese tiempo, una funcionalidad específica, de forma completa. Una de las herramientas más usadas para la administración de los proyecto ágiles se llama Trello. En el curso se va a utilizar dicha herramienta para controlar el proyecto y el avance de las tareas de cada equipo.

Metodología híbrida

Entonces ¿cómo se trabaja en el curso?

En el curso hay dos fases:

- Una fase documental.
- Una fase de construcción.

Fase documental

La fase documental cuenta con tres etapas, cada una de ellas generará un documento entregable, que cada grupo deberá de entregar al profesor en la semana 15. Las etapas se describen a continuación

Etapas de iniciación: La fase de iniciación va desde la conformación de los equipos hasta la toma y documentación de los requerimientos. Durante esta etapa los equipos son conformados y se les asignan los roles a cada uno de los miembros. Cada equipo deberá de generar una ficha técnica y enviarla al profesor de factor humano para su aprobación. En esta etapa se tienen dos reuniones con el cliente:

- La primera es donde se les presenta y exponen los requerimientos que debe cumplir la aplicación que el equipo deberá desarrollar.
- La segunda, que se realizará una semana después, se hace para que los equipos puedan aclarar dudas luego de analizar la información brindada. Luego de estas sesiones, los equipos deben redactar los informes relevantes para ser revisados, y que conformarán luego la especificación de requerimientos. Las fechas relevantes de esta etapa

son:

- Semana 3 liberación de los equipos de trabajo.
- Semana 4 primera reunión de requerimientos.
- Semana 5 segunda reunión de requerimientos.
- Semana 6 entrega de informes de iniciación.
- Semana 15 entrega de Especificación de requerimientos.

Etapas de análisis y diseño: Esta etapa inicia luego de la entrega de los informes de iniciación, y finaliza con la entrega de los informes de análisis y diseño. Durante esta etapa los grupos deberán de realizar el análisis de los requerimientos entregados con el fin de obtener los casos de uso que cada integrante va a desarrollar.

Como se vió anteriormente, un requerimiento es algo que el producto o servicio debe satisfacer. El concepto de requerimiento aplica para todos los productos, y es contra lo que van a evaluar el producto una vez que se entrega.

Un caso de uso, por su parte, es la unidad básica de programación, y relata la interacción entre el usuario y el sistema; interacción que deberá de ser programada por cada desarrollador. En un producto de software hay muchas de estas interacciones. Es durante esta etapa que los desarrolladores deberán de identificar estas interacciones, detallarlas y dividir las.

Como cada caso de uso debe ser programado, se deberá de diseñar cómo se van a ver cada una de las páginas que van a permitir la interacción entre el usuario y el sistema. La maquetación es de vital importancia, ya que es la que le va a permitir al desarrollador visualizar la interacción que definió en el proceso de análisis.

Las fechas relevantes de esta etapa se detallan a continuación:

- **Semana 8 Entrega de informes de la etapa de análisis.**

Fase de construcción: La fase de construcción es en la que cada desarrollador programa lo que analizó y diseñó. En esta etapa es donde pone en acción los elementos que ha aprendido de la parte técnica durante las primeras ocho semanas del curso. Conforme los desarrolladores van avanzando en su programación, el producto se va unificando, hasta lograr un alcance específico. Esta fase se divide en dos iteraciones, en cada una de las cuales se le va a agregar funcionalidad al producto.

- **Primera iteración:** En esta primera iteración se entrega la funcionalidad especificada por los profesores, generalmente relacionada con la inserción, listado y búsquedas de información. Esta iteración cuenta con cinco sprints, o etapas.
 - **Sprint arquitectónico:** El sprint arquitectónico se desarrolla durante la semana 9, y es en el que el desarrollador va a implementar un caso de uso usando toda la arquitectura del curso. Esta forma de programar (que se le conoce como programación vertical, o por caso de uso) es obligatoria.
 - **Sprints incrementales:** Los sprints incrementales son tres micro ciclos que se realizan durante la semana 10 y parte de la semana 11. Estos micro ciclos tendrán una duración de tres días el primero, cuatro días el segundo y tres días el tercero, y en ellos cada programador deberá de desarrollar el resto de los casos de uso que se le asignaron.

- **Sprint pruebas:** Este último sprint está compuesto por los últimos tres días de la semana 11, justo antes de la realización de la presentación de mitad de período. Este sprint está hecho para que los grupos prueben la funcionalidad que han logrado desarrollar hasta la fecha.
- **Segunda iteración:** En la segunda iteración, los grupos deberán de corregir las observaciones que se les dieron en la primera iteración y terminar la aplicación. Generalmente en esta etapa se le añade la funcionalidad de eliminación, modificación y activación/desactivación, así como la parte de mayor lógica de proceso del proyecto. Esta iteración contará con los mismos sprints de la iteración anterior.

Licencia: licencia propietaria