

Ingeniería del Software

4. Implementación y pruebas

David Guijo Rubio

Asignatura “Ingeniería del Software”
2º Curso Grado en Ingeniería Informática
Escuela Politécnica Superior
(Universidad de Córdoba)
dguijo@uco.es

27 de noviembre de 2018



- 4.1. Scrum.
- 4.2. Implementación.
- 4.3. Técnicas de validación.
 - 4.3.1. Matriz RF/CU.
 - 4.3.2. Matriz CU/Clases.
- 4.4. Entregables.

SCRUM

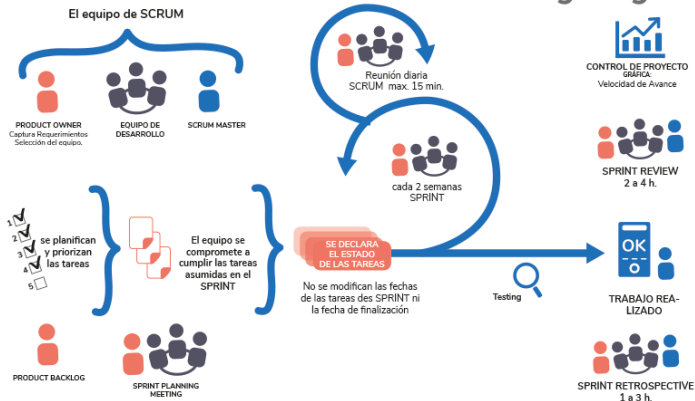
- Es una metodología ágil que se basa en entregas parciales y frecuentes de un producto para obtener resultados con rapidez.
- Es una metodología incremental, cuyos objetivos varían debido a que están poco definidos.



SCRUM

scrum

Proceso de la metodología Ágil



SCRUM

SCRUM es propuesto para:

- **No desperdiciar** el tiempo.
- La **calidad** del producto comienza en el principio del desarrollo.
- **Crear conocimiento.**
- Tomar las decisiones en el **momento idóneo.**
- Entregas **más rápidas.**
- **Motivar** a todo el equipo.
- **Optimizar** todos los procesos al máximo.



SCRUM

Tiene los siguientes roles:

- **Product Owner:** marca las pautas de actuación del proyecto general.
- **Scrum Master:** guía las reuniones y coordina el equipo.
- **Team:** implementan las funcionalidades.
- **Users:** beneficiarios finales.



SCRUM

Tiene las siguientes herramientas:

- **Product Backlog:** contiene todas las funcionalidades ordenadas por prioridades. Es la agrupación de todas las historias de usuario del producto.
- **Sprint Backlog:** funcionalidad a desarrollar en un *sprint* determinado.
- **Burndown Chart:** gráfico que muestra cantidad de trabajo hecho.



Restricciones

- El proyecto se realizará haciendo uso del sistema de control de versiones Git y la plataforma GitHub para el almacenamiento del repositorio de forma remota.
- El historial de cambios queda guardado en las cuentas de Git por lo que la evaluación será incremental y no servirá subir los ficheros de un día para otro en Git.
- Todas las semanas se realizará una evaluación, en la que habrá que detectar los problemas encontrados y las soluciones propuestas.
- La entrega final hay que mandar el software implementado y documentación técnica, profesional y formal, con todas las prácticas incluidas en el documento, siguiendo las directrices indicadas por el profesorado.



Restricciones

- Para seguir la metodología SCRUM, el *scrum master* debe dividir el trabajo entre los miembros del equipo, otorgando una cantidad de tiempo a cada tarea.
- Cada miembro del equipo, debe hacer la funcionalidad otorgada e implementarla. Puede haber un programador, un tester, etc.
- El *scrum master* se encargará de realizar el *product backlog*, distribuir en *sprint backlogs* y realizar el *burndown chart*.



Implementación

- Tal y como se comentó en la primera práctica, haremos uso del lenguaje C++.
- En caso de querer utilizar un IDE, dispondremos de Eclipse.
- El código debe ser legible y estar comentado correctamente, para una adecuada interpretación.



Pruebas

- Siguiendo la metodología SCRUM, debemos detectar posibles fallos y proponer soluciones.
- Este flujo de acciones debemos realizarlas de forma incremental (semanalmente).
- Los fallos deben estar bien documentados, así como las posibles soluciones.



Matriz de trazabilidad

- Matriz requisitos funcionales (RF) - casos de uso (CU): cada RF debe quedar cubierto por al menos un CU. Con esto nos aseguramos que todas las funcionalidades requeridas son tenidas en cuenta.
- Matriz casos de uso (CU) - clases: cada caso de uso debe tener asignado una clase al menos, en caso contrario, faltaría mejorar la definición de la clase, o la creación de otra.



Entregables SCRUM

Todos los entregables serán subidos a Git semanalmente, con lenguaje Markdown. Se entregarán tres documentos

- 1 **Product Backlog:** un fichero Markdown con todas las historias de usuario detectadas y con una prioridad otorgada, así como un tiempo estimado de implementación.
- 2 **Sprint Backlog:** un fichero Markdown por cada sprint desarrollado (uno por semana).
- 3 **Burndown Chart:** después de cada reunión realizada por el equipo, hay que actualizar el *burndown chart*, subiéndolo a Git después de cada reunión.

Fecha límite de entrega I: 09 de diciembre a las 23.55h.

Fecha límite de entrega II: 16 de diciembre a las 23.55h.






Entregable final

- 1 Software implementado correctamente documentado.
- 2 Documentación técnica, profesional y formal, con todas las prácticas incluidas en él (incluir las técnicas de validación).

Fecha límite de entrega: 23 de diciembre a las 23.55h.



Referencias

-  Arlow, J. y Neustadt, I. (2016). Programación UML 2. Programación (Anaya Multimedia) Anaya Multimedia.
-  Debrauwer, L., y Van der Heyde, F. (2016). UML 2.5: iniciación, ejemplos y ejercicios corregidos. Ediciones ENI.
-  Miles, R., y Hamilton, K. (2006). Learning UML 2.0. O'Reilly Media, Inc. 2



Ingeniería del Software

4. Implementación y pruebas

David Guijo Rubio

Asignatura “Ingeniería del Software”
2º Curso Grado en Ingeniería Informática
Escuela Politécnica Superior
(Universidad de Córdoba)
dguijo@uco.es

27 de noviembre de 2018

