|  |  |  |
| --- | --- | --- |
|  | **Evaluación de Aprendizaje III**    Pág. 1 de 2 |  |

2637 - Introducción a la Administración de Proyectos

EVALUACIÓN DE APRENDIZAJE III

# GESTIÓN DE LA CONFIGURACIÓN Y CALIDAD

**ALUMNO:**

**DNI:**

**FECHA DE ENTREGA: 30/11/20**

|  |  |  |
| --- | --- | --- |
|  | **Evaluación de Aprendizaje III**    Pág. 2 de 2 |  |

Consigna

1. ¿Para que es importante tener un proceso de Gestión de la Configuración?
2. Defina y describa como seria una estrategia de **branch** and **merge** para un proyecto en el cual trabajan 3 equipos con diferentes objetivos en el mismo proyecto que utiliza el framework Scrum Ejemplifique gráficamente al menos 2 sprints.
3. De un ejemplo de un **Control de Cambios**.
4. Haga un diagrama entidad-relación de cómo sería un sistema de gestión de versiones.
5. ¿Que es la calidad del Software? ¿Cómo se puede medir?
6. ¿Qué es y que diferencia hay entre **Garantía de calidad** y el **Control de calidad**?

*(continuar la resolución en una hoja nueva)*

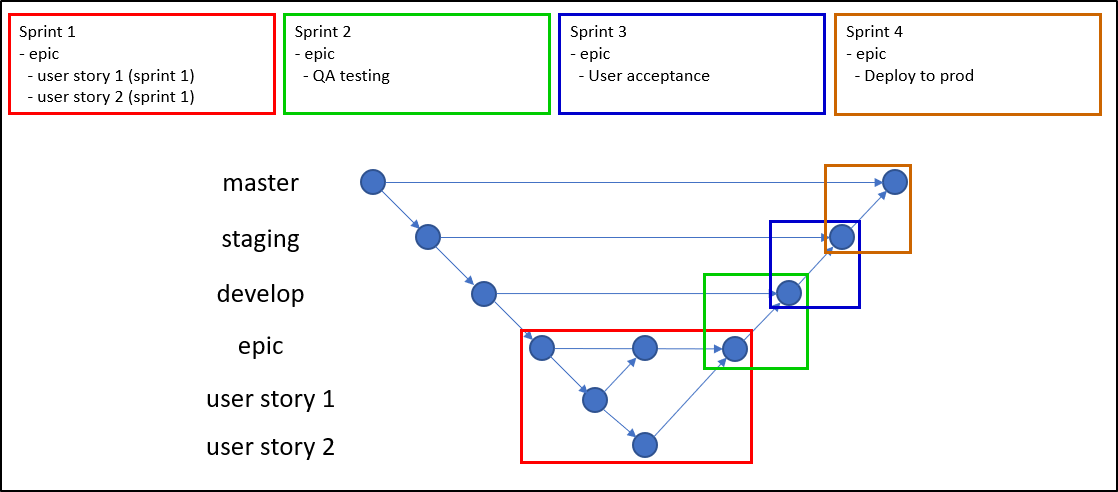
1. ¿Para que es importante tener un proceso de Gestión de la Configuración?

Para poder entender para qué es importante tener un proceso de gestión de la configuración, primero necesitamos saber la definición. La Gestión de la configuración se puede definir como una disciplina cuya misión es controlar la evolución de un sistema de software. Es el arte de identificar, organizar y controlar las modificaciones que sufre un software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores.

1. Defina y describa como seria una estrategia de **branch** and **merge** para un proyecto en el cual trabajan 3 equipos con diferentes objetivos en el mismo proyecto que utiliza el framework Scrum Ejemplifique gráficamente al menos 2 sprints.

Una estrategia de Branch y Merge para un proyecto de varios equipos podría ser:

* Tener una rama denominada **master** la cual contendría la versión de producción
* Tener una rama denominada **staging** que contendría features ya finalizadas y que deben ser sometidas a pruebas de aceptación de usuario. Hay que tener en cuenta que esta rama es la de pre-producción
* Tener una rama denominada **develop** en la cuál el equipo de QA realice pruebas
* Y finalmente, tener distitnas ramas por cada épica del backlog y por cada historia de usuario. De esta manera, cada rama perteneciente a una historia de usuario se mergea a la rama de la épica a la que pertenece. Una vez finalizada la épica, está lista para ser mergeada a **develop**. Y así de esa manera, se va mergeando esos cambios a las ramas padres (y se van probando los distintos cambios/features agregadas) hasta llegar a **master**



Para entender un poco la imagen:

En el sprint 1 se desarrollaron las historias de usuario, mergeando cada una de ellas a la rama de la épica.  
En el sprint 2, se mergea la rama de la épica a develop, para que se realicen pruebas de QA.  
En el sprint 3, se mergea la rama de develop a staging, para que se realicen pruebas de aceptación de usuario.  
En el sprint 4, se mergea la rama de staging a master y se realiza el deploy al entorno productivo.

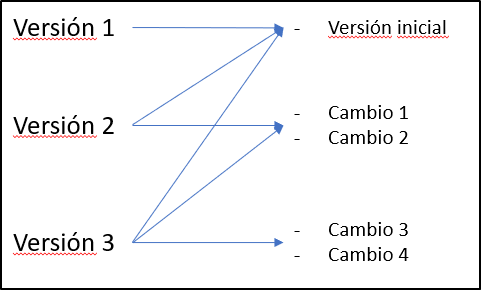
Hay que tener en cuenta que lo anterior es una versión simplificada, generalmente se trabaja con varias épicas en el mismo sprint.

1. De un ejemplo de un **Control de Cambios.**

Un ejemplo de control de cambios podría ser cuando tenemos un cliente el cual hace una petición de cambio relacionado con uno de los contratos de una API de su plataforma. Por ejemplo, el cliente ahora en su API de alta de usuarios necesita pasarle en vez de un rol para el usuario nuevo, un conjunto de roles.  
Para poder desarrollar ese cambio, es necesario primero evaluar el esfuerzo técnico que ese cambio conlleva; hay que evaluar el posible impacto sobre otras partes del software (el cambio en el contrato de una API es un cambio con mucho impacto); hay que evaluar las prioridades para saber si ese cambio se necesita lo más pronto posible o puede esperar; hay que evaluar los recursos disponibles; hay que evaluar si el cambio solicitado es viable.

El análisis y evaluación lo lleva a cabo el comité de control de cambios el cuál dependiendo del tamaño del proyecto puede ser una persona sola o un grupo de personas. El comité es el encargado de tomar la decisión para seguir adelante con el cambio o no.

1. Haga un diagrama entidad-relación de cómo sería un sistema de gestión de versiones.



El control de versiones es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o documento. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Se puede definir una versión como una instancia de un elemento de configuración, en un momento dado del proceso de desarrollo, que es almacenada en un repositorio y que puede ser recuperada en cualquier momento para su uso o modificación.

A las distintas versiones que surgen en el tiempo se las suele llamar **Revisiones**.

1. ¿Que es la calidad del Software? ¿Cómo se puede medir?

La definición de la calidad del software según la IEEE es “el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Otra definición del libro Ingeniería del Software de Pressman, dice que la calidad es la “Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario”.

En otras palabras, Calidad del Software es un proceso que se aplica al desarrollo del software para poder obtener un producto que cumpla las expectativas del cliente y que tenga valor.  
Hay distintas metodologías, prácticas, procesos que se pueden utilizar en el desarrollo del software y de esa manera obtener un producto con un nivel de calidad elevado. Por ejemplo, metodologías de análisis y diseño de software permiten desarrollar una solución robusta a las necesidades del cliente. De esta manera, se pueden satisfacer los requerimientos y/o expectativas del cliente.

La calidad del software se puede medir en base métricas preestablecidas. Para cada criterio de medición de la calidad, se define un valor que define una medida cuantitativa para cierta característica del producto. De esta manera, se puede medir la calidad que tiene el software. Ejemplo: se define un atributo de calidad con respecto a la performance en el buscador de la aplicación. La definición del atributo es “El resultado de una búsqueda no debe superar los 2 segundos”. De esta manera, al momento de evaluar la calidad con respecto a esta característica, podemos medir el grado de calidad del software.

1. ¿Qué es y que diferencia hay entre **Garantía de calidad** y el **Control de calidad**?

La **Garantía de calidad** es un conjunto de actividades para garantizar la calidad de los procesos en los que el software es desarrollado. Su objetivo es evitar defectos en el proceso utilizado para hacer el software. Se realiza estableciendo un sistema de gestión de calidad y la evaluación de la adecuación de este sistema con auditorías de conformidad de las operaciones del sistema.

Por otra parte, el **Control de calidad** es un conjunto de actividades para garantizar la calidad del software. Busca identificar y corregir defectos en el software después de finalizada una versión particular y antes de ser entregado.

En resumen, la **Garantía de calidad** se enfoca en el conjunto de actividades llevadas a cabo para desarrollar el software. Mientras que el **Control de calidad** se enfoca en el software en sí mismo para asegurar cumplir los requerimientos del software.