

Solución Primer examen parcial – marzo 2016

Serie de teoría

Temario a

1. b,d
2. c
3. a
4. b,d
5. a,c
6. b,d
7. a.b.d

Temario b

1. b,d
2. a
3. a,c
4. c
5. b,d
6. a,b,d
7. b,d

Tema 1

En el artículo “The practice and future of release engineering” se mencionan varias métricas para medir la calidad de un release, explique (no solo enumere) al menos 2 de las métricas sugeridas en dicho artículo.

- Our main measures are threefold: the number of open bugs ranked by priority, the number and percentage of successful releases, and the number and percentage of releases that are abandoned late in the game.
- For Web deployment, we use the metrics of the code going into the master branch, the test results, and performance lab results. The next level includes metrics for products being released, such as core tests, unit tests, and performance experiments like time to interaction (TTI), fatal-error rates, the number of errors per page, and any new errors that we hadn't seen in the production logs.

Explique las responsabilidades de un arquitecto de software ágil, con base en lo indicado en el artículo “Towards an agile software architecture”

- Balances between the big picture and now – an agile software architect has to think about what is happening during the development together with aligning it with the big picture of the entire system.
- Hands-on experience – an agile software architect is also a developer and works on the implementation of the system. This gives first-hand feedback on the taken architectural decisions.
- Produces prototypes to make informed decisions – when an important technical decision has to be made, a quick prototype could reveal whether this decision is feasible and how it would affect the existing system. Furthermore, communication with the entire development team is essential as it is a collaborative effort, rather than a single-man activity.
- Focus on sustainability – it is extremely important that architectural decisions lead to a sustainable software architecture – one that will support the project in the long term

Tema 2, implementación de integración continua

- Establezca los objetivos del proyecto. Debe tomar en cuenta que los objetivos deben ser medibles y alcanzables. Indique como el proyecto que plantea solución o apoya a solucionar la problemática planteada.

Los objetivos planteados tendrían que ir en línea con la problemática que se busca resolver. Los objetivos no son implementar una o varias herramientas, sino, a través de ellas resolver algún problema. Un punto de partida para plantear los objetivos podrían ser los objetivos de un proceso de liberación: alta calidad y velocidad.

Los objetivos podrían plantearse así:

- Reducir los defectos para publicaciones de software en un período de 3 meses
 - Reducir el tiempo de ciclo para la entrega de productos de desarrollo a los clientes solicitantes en un período de 3 meses
- Defina métricas y metas para los objetivos planteados, para un período de tiempo que considere conveniente

Partiendo de los objetivos planteados, las métricas y metas podrían ser.

- Reducir los defectos para publicaciones de software en un período de 3 meses
 - Métrica: cantidad semanal de defectos en producción
 - Meta. Que la cantidad semanal de defectos en producción baje de 7.5 a 5 en un período de 3 meses
- Reducir el tiempo de ciclo para la entrega de productos de desarrollo a los clientes solicitantes en un período de 3 meses
 - Métrica: tiempo promedio de publicación de una solicitud, desde que la solicita el cliente hasta que es puesta en producción
 - Meta: que tiempo promedio de publicación de una solicitud, disminuya de 6 días a 4 días en un período de 3 meses

- Defina las políticas que se utilizarían para manejo de control de versiones, el alcance del script de build y todo el software requerido para el proyecto.

Las políticas deben incluir:

- Definición de ramas a utilizar
- Definición de procedimientos para pasar de una rama a otra y quién debe hacerlo
- Definición de estándar de etiquetado a utilizar
- Frecuencia de commits
- Restricciones para hacer commit

El alcance del script de build puede incluir:

- Compilar software (si aplica)
- Correr pruebas unitarias
- Correr pruebas funcionales automatizadas
- Correr validaciones de estándares de código

El software requerido dependerá del alcance definido en el script de build, indicando las herramientas necesarias para control de versiones, integración continua y diferentes etapas del script de build que se definieron.

- Haga una versión inicial de un plan de actividades a desarrollar para la implementación del proyecto.

El resumen de las principales actividades a desarrollar es:

- Definir alcance del proceso de integración continua (sistemas, elementos a controlar, etc.)
- Definir proceso a seguir
- Definir herramientas a utilizar
- Instalar y configurar herramientas
- Capacitar a personal en uso de procesos y herramientas
- Piloto de implementación de proceso y herramientas
- Soporte a implementación del proceso y herramientas
- Ajuste de procesos
- Capacitar a personal en uso de procesos y herramientas
- Implementación en la organización
- Soporte a implementación del proceso y herramientas

- ¿Qué cambiaría en su plan y métricas si la implementación fuese de entrega continua?

Ya que la entrega continua implica la publicación automatizada en diferentes ambientes, se podría agregar como objetivo: Aumentar la cantidad de publicaciones automatizadas en producción. Esto implicaría la creación de una métrica para llevar un conteo de las publicaciones manuales y las automatizadas, para medir si se alcanza alguna meta definida.

Con relación al plan de actividades, el alcance de la primera actividad sería mayor al ser necesario tomar en cuenta los ambientes y sus configuraciones como parte del control de versiones. Esto podría afectar todas las demás actividades ya que ahora habría elementos adicionales a controlar y pasos de proceso (publicación en un ambiente) que no estaban antes.