

1. ¿Qué es SCRUM?

SCRUM es un framework ágil para el desarrollo de proyectos (no solo tecnología), este a su vez usa otros marcos de trabajo como AGIL y KANBAN.

De manera resumida permite cambios a lo largo del proyecto y va orientado a entregas rápidas, lo que permite que el desarrollo del proyecto se ajuste y se actualice a los cambios y necesidades que surgen en la línea del tiempo.

Para esto su visión de proyecto es descomponer el monolito en partes pequeñas para trabajar y a su vez entregas más frecuentes.

Los actores dentro de un proyecto pueden cambiar, la flexibilidad del marco de trabajo permite que sea más dinámico la asignación de roles con el fin de explotar más las capacidades del equipo.

2. ¿Especifique que es, como se divide y cómo se construye una historia de usuario?

- ☐ Las historias de usuario son pequeñas descripciones de los requerimientos esta incluye a su vez los criterios de aceptación, es como una carta de lo que el usuario necesita pero debe definirse el como, lo que se quiere y el para que (todo se hace usando un lenguaje común no técnico) esta debe de ser precisa y corta.

- ☐ Como parte inicial lo podemos ver como una carta o tarjeta donde en la parte frontal definimos la historia sin detallarla
 - Nombre: (comúnmente se define con un verbo) identifica de forma directa la historia
 - Descripción: especificamos la funcionalidad, el cómo (rol de la persona), el puedo (la acción) y el para (el valor de la acción o justificación)
 - Comprobación: en donde definimos los criterios de aceptación y este va orientado al estado previo “dado que”, “al cuándo” y el “entonces”

y en la parte trasera detallamos la historia resultante de la interacción o conversación entre el team y el owner

- Reglas de negocio
- Requerimientos no funcionales
- Otros detalles importantes

- ☐ Para construir una historia tenemos que tomar en consideración lo siguiente
 - Empezar por el final
 - Trabajar hacia atrás
 - Hacer pasos pequeños
 - Una vez definido el rol, la funcionalidad y el resultado que se espera se crea la carta o tarjeta de la historia

3. ¿Qué es un criterio de aceptación?

Son los detalles de una historia, esta mejora el entendimiento, clarifica lo que ocurre y facilita la medición para saber cuándo es que realmente terminamos la historia.

Está de finido como el DADO - CUANDO - ENTONCES.

4. ¿Qué es y para que se utilizan la “Review Meeting” dentro del ciclo de Scrum?

Es la ventana de tiempo que se convoca al final de cada sprint y es el momento en el que el equipo scrum presenta los entregables del sprint al product owner y stakeholder relevantes, estos son recibidos sólo si cumplen con los criterios de aceptación definidos.

El tiempo recomendado es una hora por semana de nuestro sprint, si este es de una semana se recomienda que sea de una hora, es un momento muy importante pues se tiene el feedback y es un momento en el que el equipo de desarrollo puede interactuar con los stakeholders para aclarar dudas y tener una mejor perspectiva del producto, a la vez que los stakeholder ven el incremento del producto, el equipo puede saber cómo vamos y si es necesario hacer ajustes.

Está propenso a que surjan nuevos requerimientos, épicas, historias, dependiendo del caso se puede priorizar o solamente agregar al backlog.

5. ¿Qué es y para que se utilizan la “Retrospective Meeting” dentro del ciclo de Scrum?

Es el momento en el que el equipo analiza y reflexiona lo que ha ido mal o bien, lo que se necesita mejorar, reforzar o cambiar para ser más efectivos ajustando y perfeccionando el comportamiento como consecuencia.

Se realiza al terminar el desarrollo de un sprint o versión del producto e incluso al terminar un proyecto.

Se recomienda

- Contar con un moderador
- Los asistentes deben conocer el objetivo y la mecánica de la reunion
- participar en la reunión tomando en cuenta
 - Indagar antes que apoyar
 - dialogar antes que debatir
 - conversar antes que argumentar
 - entender antes que defender

La reunión termina con la lista de las acciones elegidas para el próximo sprint.

6. Documente 2 historias de usuario, los criterios de aceptación y sus casos de prueba funcionales para la creación de un Login.

1	Ingresar al sistema
COMO usuario final QUIERO autenticar mis credenciales PARA ingresar al sistema	
Estimacion: 3	
Prioridad: 1	
Dependencia: 0	
<ul style="list-style-type: none">● El usuario debe ser nombre.apellido● La contraseña debe ser mayor a 8 caracteres● Escribir en el log el evento del usuario● Mostrar un mensaje al usuario del resultado de login	

2	Mostrar menu
COMO usuario final QUIERO cargar mi menú PARA utilizar las funcionalidades del sistema	
Estimacion: 3	
Prioridad: 3	
Dependencia: 1	
<ul style="list-style-type: none">● El usuario tiene asignado un rol específico● El rol tiene definido las opciones del sistema● Al ingresar a una opción se guarda en el log el usuario, la opción y la fecha y hora en que se utiliza la opción	

7. Describa un patrón de arquitectura que usted utilice más frecuentemente y la razón por la que usted prefiere este patrón.

El patrón de arquitectura que utilizo es PATRON EN CAPAS, este patrón me permite separar en capas mi aplicación creando una separación de actividades en las que un nivel de capa se comunica con otro como una forma en vertical, con esto ordené mejor las funciones o tareas según la capa en la que me encuentre.

me permite mantener un código más ordenado y escalable tomando en consideración que las aplico ya sea en servicios web y/o aplicaciones web que es lo que comúnmente me piden.

Con esto puedo incluso cambiar la capa UI de ser necesario y hacer de la solución un servicio web, o a la inversa tener un servicio web y poner una capa de presentación para tener como resultado una aplicación web.

8. ¿Qué es un patrón de diseño?

Un patrón de diseño en su práctica es una plantilla que da solución a problemas comunes, son reutilizables y a su vez opcionales, el desarrollador decide si lo utiliza o no.

Las categorías principales son:

- Patrones creacionales (abstract factory, prototype, singleton, etc.)
- Patrones estructurales (decorator, proxy, adapter, etc.)
- Patrones de comportamiento (iterator, observer, etc)

9. ¿Qué es TDD? y describa sus 3 principales pasos.

TDD (Test Driven Development) es una metodología de desarrollo que tiene el objetivo principal de crear primero las pruebas y luego escribir el software (Desarrollo guiado por pruebas), al finalizar es necesario refactorizar el código.

Sus tres principales partes son:

- la prueba debe fallar
- la prueba debe pasar
- se debe mejorar el código

10. Defina un ciclo CI/CD detallado y describa cada uno de sus pasos, enfoque esta descripción como si fuera una propuesta de mejora en el proceso SecDevOps de la compañía y lo debe enviar escrito como propuesta para el gerente de desarrollo.

Tenga un cordial saludo GERENTE DE DESARROLLO

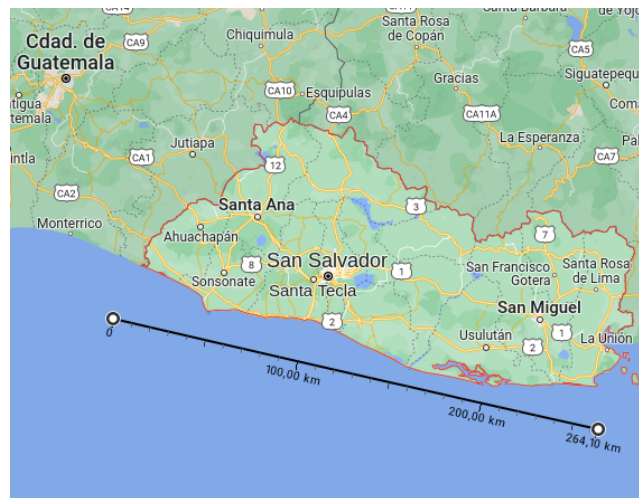
De manera general en un ciclo CI/CD se deben considerar aspectos como:

- Lenguaje de programación (lenguaje compilado/interpretado):
- Recurso disponibles (area de desarrollo, equipo QA)
- Definición de contenedores a utilizar (tradicionales/dockers)
- Definición de infraestructura

Es necesario tener en cuenta si la infraestructura, si esta será administrada por la empresa o si se contratará servicios de infraestructuras en la nube.

En la definición de la infraestructura a utilizar es importante destacar las ventajas de usar servicios cloud con el fin de no invertir en recursos propios, administración y mantenimiento del mismo.

Considerando que al menos en El Salvador tiene un área aproximada de 250km*95km no es recomendable tener un sitio de contingencia, esto a razón que el país es vulnerable a catástrofes naturales como volcanes que incluso están dentro del cinturón de fuego, expuesto a tsunamis, terremotos, etc.



En tal sentido mi recomendación es contratar un servicio de infraestructura con el cual se puede reducir los costos y responsabilidades, alcanzando con mayor eficiencia una alta disponibilidad y contingencia del negocio.

El ciclo propuesto puede necesitar ajustes según las consideraciones anteriores sin embargo los pasos generales de la propuesta son los siguientes:

- ☐ La compilación debe de ser completa y sin errores.
- ☐ Debe superar todas las pruebas unitarias.
- ☐ El código no debe tener alertas de seguridad continua realizados con herramientas como SONAR.
- ☐ Actualizar la rama de DESARROLLO en el repositorio GIT.
- ☐ Al completar el desarrollo del requerimiento se debe migrar los cambios a la rama PRUEBAS.
- ☐ Se debe desplegar la aplicación en un ambiente de PRUEBAS.
- ☐ Los usuarios designados deben verificar el funcionamiento de la aplicación en el ambiente de PRUEBAS.
 - En caso que las pruebas fallen debe de realizar el proceso correspondiente para corregir los problemas y/o aplicar las mejoras.
 - En caso el usuario avale los cambios se debe programar el paso a PRODUCCION.
- ☐ Programar el paso a PRODUCCION
 - Definir la ventana de tiempo a soportar en caso de fallo antes de realizar un rollback automático.
 - Realizar un respaldo de la aplicación en PRODUCCION en caso de ser necesario un rollback.
 - Definir el tiempo de monitoreo de la aplicación.
- ☐ Luego de realizar el despliegue y concluir el tiempo de monitoreo se da por completado la puesta en producción del requerimiento(s).

Nota: Es recomendable exponer la propuesta con el área de operaciones y desarrollo con el fin de tener en consideración otros puntos de vista que ayude a mejorar la propuesta anterior.

Att.

Marco Martinez