Ideas para cada PPT:

1. Presentación del grupo y del tema que se presentará. Esto incluye número de grupo, cátedra, año de cursado, facultad y todo eso que la cñora meles quiere.

Pasando en claro: Los principios del testing ágil son 7 y en algunas bibliografías se plantea un octavo, nos pareció importante agregarlo. La idea es “materializar” los valores del manifiesto del testing ágil que fueron desarrollados por el grupo anterior. Como el testing en ágil se debe hacer en cada iteración, es importante que se describan ciertos principios que ayuden al equipo a tener ideas más claras a la hora de testear el producto y no se desvíen de lo que le da valor al negocio.

1. Primer Principio: Pruebas Continuas

Los testers deben realizar pruebas continuas, que permite realizar un control de la calidad del producto a medida que el proceso de desarollo avanza. Es decir que, a medida que se va desarrollando se va testeando, pudiendo encontrar defectos lo antes posible, tal que su impacto y costo sea el menor. Provocando además una mejora continua, ya que se detectan oportunidades de mejora en etapas tempranas evitando que se repliquen en el proceso.

Al probar y corregir, permite tener software funcionando en cada operación logrando obtener una versión potencialmente entregable.

Esto debería dar 20 seg justo, sin querer lo hice pensando que era mi filmina, he.

NO explayar lo de hacer al final las pruebas, ya que es el séptimo principio.

1. Primer Principio: ¡Automatizar pruebas!

El testing ágil plantea la automatización de pruebas de regresión, la automatización de pruebas unitarias entre sus prácticas y el desarrollo guiado por pruebas (TDD) para facilitar el proceso de prueba y aplicarlas mientras se desarrolla.

1. Segundo Principio: Proveeer Retroalimentación Constante: A medida que el testing se va llevando a cabo, es esencial asegurar que se tenga una retroalimentación continua de las pruebas al desarrollador, para que él mismo pueda hacer cambios en etapas tempranas y para que el diseño de producto alcance los objetivos de negocio.
2. Segundo Principio: Es esencial porque ayuda al equipo a mantenerse en el camino correcto en términos de la lógica de negocio y comportamientos del software. Es lo que efectivamente permite que el producto cumpla con las necesidades y expectativas del negocio.
3. Tercer Principio: El equipo completo realiza pruebas: maru

Cuando se trata de todo el equipo serán todos sus integrantes, se habla tanto de testers, desarrolladores y analistas del negocio. De tal manera que forman un solo equipo para realizar todas las pruebas necesarias.

Mientras los desarrolladores prueban que el producto se ejecute correctamente, los analistas validan que hagan lo correcto y se tenga una mejor visión del trabajo que se está realizando.

1. Tercer Principio:

A diferencia del tradicional, sólo ciertas personas se dedican y concentran en el testing, de manera que el desarrollador se desentendía del testing y lo dejaba al otro equipos las pruebas. Que, resultaba ser al final de la implementación, provocando que la corrección sea más cara y el impacto mayor de los defectos.

En cambio ágil, todos son responsables y trabajan en conjunto para ejecutar las pruebas.

1. Cuarto Principio: Retroalimentación rápida.

El equipo involucra Product Owner, desarrolladores/testers en cada iteración. El realizar pruebas constantes con todo el equipo probando nos brinda una retroalimentación rápida, acortando los tiempos de respuesta y mejorando la toma de decisiones, de forma tal que eso se vea efectivamente reflejado en una mejora en el producto.

Una manera eficaz de responder rápidamente es a través de la automatización de pruebas, esto hace que quienes vayan a probar el producto lo hagan de una forma rápida para que puedan dar una devolución a los desarrolladores y que puedan asegurarse de que lo implementado funciona correctamente.

(Bonus track por si no llegas a los 40segs mi broder): Una retroalimentación rápida permite identificar errores en el mismo momento en que son creados, evitando que se transformen en defectos costosos que se detectan en etapas muy avanzadas del producto.

1. Cuarto Principio:
2. Quinto Principio: Crear un código simplificado y limpio.

Todos los fallos o problemas hallados por el grupo de trabajo durante el proceso de desarrollo del producto, se arreglan dentro de la misma iteración, lo que ayuda a crear un código limpio y simplificado, es decir, un código que nos permite tener un incremento totalmente funcional del producto y libre de errores asegurando un alto nivel de calidad del software. El testing es un factor fundamental para crear un código simple, limpio, no ambiguo y libre de errores.

La ventaja de tener un código limpio, es que ante un cambio en las necesidades del usuario, entrar a tocar el código ya escrito va a ser más fácil, porque va a ser más legible y a la hora de hacer testing, encontrar los errores será más sencillo.

1. Quinto Principio:
2. Sexto Principio: Producir menos documentación: El desarrollo ágil se enfoca en cumplir las necesidades del cliente, más que tener que desarrollar documentos con instrucciones y requerimientos de manera comprensiva.

Una de las técnicas es utilizar listas reutilizables o checklist para evitar invertir tiempo en documentar y poder dedicar mayor tiempo en hacer tareas que al cliente le beneficien. Debe enfocarse más en realizar las pruebas del producto en lugar de los detalles incidentales. Esto va de la mano con uno de los valores del manifiesto ágil que es software funcionando sobre documentación extensiva. A veces nos preocupamos más por la documentación que terminamos dejando de lado el producto funcional que al final de cuentas es lo que al cliente realmente le importa. Entonces este principio nos ayuda a que sólo documentemos lo indispensable.

1. Sexto Principio:
2. Séptimo Principio: Conducido por pruebas: maru

Muy relacionado con el primer principio, pero destaca la buena práctica de realizar pruebas en todo el proceso de desarrollo de software. Como se ve en la imagen, ante una solicitud se define la prueba (Que falla pues aún está en desarrollo), para implementar en base al camino deseado y los nos deseados de la prueba confeccionada.

Cómo rayos es que llego a 20 segundos justo con esta miércole de párrafo?

1. Séptimo Principio:

En términos de implementación, la definición de pruebas para codificar ayuda un diseño simple y código testable, con lo que el mismo debe estar desacoplado. Lo que se realiza es ir construyendo las pruebas mientras el desarrollador testea, para ir logrando guiar al programador.

1. Octavo Principio: Satisfacción del Cliente (Customer Satisfaction):

(Algunas biblios lo plantean, y bla bla). Ágil es una metodología que acepta que los requerimientos cambien y el cliente puede ir adaptando y actualizando los requerimientos a medida que avanza el desarrollo del producto. Lo que plantea este principio es que el testing ágil tiene como desafío aceptar la modificación de las pruebas en base a los cambios en los requerimientos. Los casos de prueba deben irse modificando o se deben crear nuevos casos de prueba a lo largo del desarrollo del producto según el cliente agregue o cambie funcionalidades del mismo, de manera tal de generar una mayor satisfacción de él con lo que se está desarrollando.

Además, al hacer cambios en los requerimientos y cuando éstos son llevados al desarrollo, se pueden modificar partes del código que ya fueron testeadas, lo que puede producir la aparición de nuevos errores, por lo tanto, se deben modificar los casos de prueba y ejecutarlos nuevamente.

1. Octavo Principio:
2. Conclusión:

(falta terminar)

(Mastra) Los principios de testing ágil favorecen al desarrollo ágil, ya que, el crear código que se testea en el mismo momento que se desarrolla, permite favorecer la idea de ir construyendo calidad junto al desarrollo del producto. Esto permite al final de cada iteración generar un entregable o porción del producto que se pueda poner en producción

(Chona) Como ya habrán notado, los principios de testing ágil coinciden de gran manera con los principios ágiles, respecto al feedback, menos documentación, involucrar diferentes personas dentro del mismo equipo de desarrollo, etcétera.

1. Bibliografía

Distribución Tentativa 1: (Seleccionada)

Persona 1: Introducción y Primer Principio. (2, 3 y 4) – Alex

Persona 2: Presentación (1) y Segundo Principio (5 y 6) – Fabri

Persona 3: Tercer Principio (7 y 8) y Séptimo Principio (15 y 16) Maru

Persona 4: Cuarto Principio (9 y 10) y Octavo Principio (17 y 18) - Sabri

Persona 5: Quinto Principio (11 y 12) Conclusión y bibliografía (19 y 20) Chona

Persona 6: Sexto Principio (13 y 14) Colusión y bibliografía (19, 20)

Bibliografía

De todos los principios:

<https://enlabsoftware.com/agile-management/agile-testing-principles-for-tester-and-agile-software-development-team.html>

<https://www.crehana.com/blog/diseno-productos/agile-testing/>

<https://www.coderhouse.com/blog/agile-testing-la-mejor-estrategia-para-disen-ar-productos-de-calidad>

<https://www.testscenario.com/agile-testing-principles-and-challenges/>

<https://www.javatpoint.com/agile-testing>

<https://www.sealights.io/agile-testing/agile-testing-8-principles-7-challenges-and-how-to-master-them/>

Pruebas Continuas:

<https://es.parasoft.com/blog/what-is-continuous-testing-and-how-to-bring-it-into-your-organization/>

Clases Covaro donde hace una introducción del pecha kucha:

<https://www.youtube.com/watch?v=hKoJBlhxuN8>

<https://www.youtube.com/watch?v=3EdfYvpko8A&list=PLsI9mUiT17yxu5x12_pGNW_Afuf_lY4x2&index=19>

XP:

<https://dosideas.com/noticias/metodologias/371-programacion-en-parejas>

Testing Continuo:

<https://www.linkedin.com/pulse/la-carrera-del-testing-continuo-rodrigo-andr%C3%A9s-montenegro-tagle/?trk=pulse-article&originalSubdomain=es>