

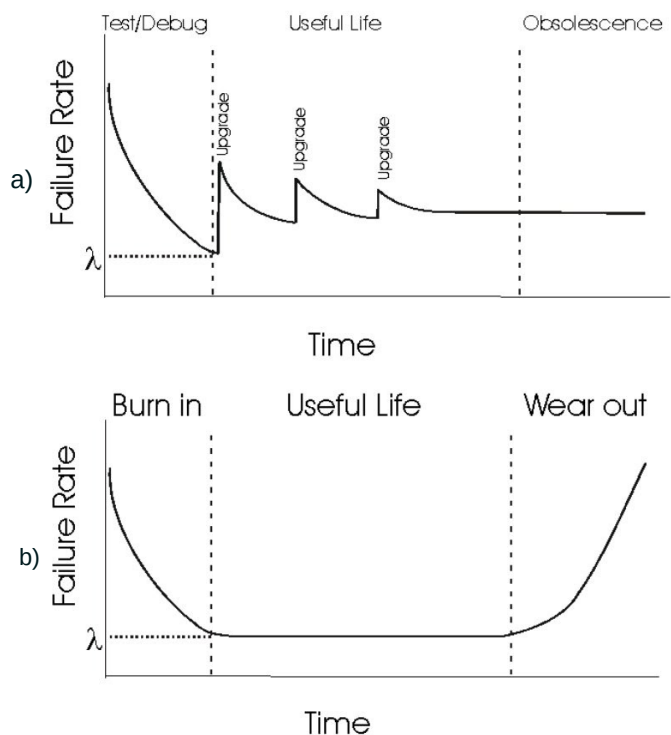


[Área personal](#) / [Mis cursos](#) / [Ingeniería de Software II](#) / [Primer parcial](#) / [Primer parcial 13/10/2021 - 14:30hs](#)

Comenzado el	miércoles, 13 de octubre de 2021, 14:33
Estado	Finalizado
Finalizado en	miércoles, 13 de octubre de 2021, 16:16
Tiempo empleado	1 hora 42 minutos
Calificación	Sin calificar aún

1.5

1) Dadas las siguientes curvas de proporsión de fallas en el tiempo, identifique y explique cada una de ellas usando palabras propias.



Expanda su respuesta explicando brevemente el envejecimiento del software.

a) En este grafico se tiene en cuenta las fallas que puede tener un desarrollo de software en relacion al tiempo. Durante el inicio del desarrollo de un SW, empieza teniendo muchas fallas. Una vez que se corrigen todas las fallas, al finalizar las pruebas y debuggearlo, recién se puede liberar el software para su entrega. Cada vez que se introduce una mejora o una modificacion y se comienza a usar el programa, se pueden introducir o descubrir nuevas fallas que hay que corregir. Al final de su vida util, cuando el programa se vuelve obsoleto, ya no se introducen nuevas mejoras al sistema y por lo tanto no hay necesidad de corregir errores porque ya no se usa con mucha frecuencia el programa. ✓

b) En este grafico se tiene en cuenta las fallas que puede tener un desarrollo de hardware en relacion al tiempo. Al principio del grafico se puede observar que el hardware desarrollado tiene muchas fallas, pero a medida que se van fabricando mas unidades estas fallas van bajando. Durante la vida util del hardware ya no se experimentan fallas debido a que ya fueron corregidas y ya al final de la vida util, vuelven a aparecer las fallas debido al desgaste que va sufriendo el hardware. ✓

El envejecimiento del software se puede producir porque se introducen mejoras que no fueron solicitadas por el dueño del producto o que los cambios nuevos introducidos introducen nuevas fallas. Cuando se intenta mejorar el código del programa para corregir fallas o introducir mejoras, se esta haciendo envejecer al software, ya que con todas estas cosas introducidas, el software no vuelve a ser el mismo de antes. ✓

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	

Paso	Hora	Acción	Estado	Puntos
2	13/10/2021 14:50	Guardada: a) En este grafico se tiene en cuenta las fallas que puede tener un desarrollo de software en relacion al tiempo. Durante el inicio del desarrollo de un SW, empieza teniendo muchas fallas. Una vez que se corrigen todas las fallas, al finalizar las pruebas y debuggearlo, recien se puede liberar el software para su entrega. Cada vez que se introduce una mejora o una modificacion y se comienza a usar el programa, se pueden introducir o descubrir nuevas fallas que hay que corregir. Al final de su vida util, cuando el programa se vuelve obsoleto, ya no se introducen nuevas mejoras al sistema y por lo tanto no hay necesidad de corregir errores porque ya no se usa con mucha frecuencia el programa. b) En este grafico se tiene en cuenta las fallas que puede tener un desarrollo de hardware en relacion al tiempo. Al principio del grafico se puede observar que el hardware desarrollado tiene muchas fallas, pero a medida que se van fabricando mas unidades estas fallas van bajando. Durante la vida util del hardware ya no se experimentan fallas debido a que ya fueron corregidas y ya al final de la vida util, vuelven a aparecer las fallas debido al desgaste que va sufriendo el hardware. El envejecimiento del software se puede producir porque se introducen mejoras que no fueron solicitadas por el dueño del producto o que los cambios nuevos introducidos introducen nuevas fallas. Cuando se intenta mejorar el código del programa para corregir fallas o introducir mejoras, se esta haciendo envejecer al software, ya que con todas estas cosas introducidas, el software no vuelve a ser el mismo de antes.	Respuesta guardada	
3	13/10/2021 16:16	Intento finalizado	Finalizado	

1.0

¿Porqué se introduce en 1960 el concepto de 'ingeniería' en el desarrollo de Software? Conteste nombrando los conceptos claves (que normalmente aparecen en una definición de ingeniería de software) para hacer del desarrollo de software una ingeniería.

¿Cree que actualmente se solucionaron las deficiencias en el desarrollo de software? Para responder esta pregunta apóyese en los conceptos de complejidad esencial y accidental.

El concepto de ingeniería de software se introduce para poder crear software mas rentable y confiable, ya que antes el costo del desarrollo solia ser mas alto de lo normal, y la confiabilidad del programa era dudosa. Además, con este concepto se introdujo una forma de desarrollar el software de forma sistemática, disciplinada, mejorar el mantenimiento. También cuando se desarrollan sistemas grandes o complejos, se lo debe hacer en equipos de personas capaces de desarrollarlo bien.

Yo creo que actualmente se solucionaron las mayorías de deficiencias durante el desarrollo de software, pero a veces suele aparecer algunas. Estas deficiencias pueden aparecer en la especificación, el diseño del software o cuando no se lo prueba correctamente, la comunicación del equipo de desarrollo no suele ser buena, al escalar el sistema se aumenta la complejidad de este, etc. O también suelen aparecer cuando hay dificultades durante la implementación del diseño o al usar técnicas o metodologías de desarrollos que no son aptas para el tipo de sistema que se desarrolla.

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 15:08	Guardada: El concepto de ingeniería de software se introduce para poder crear software mas rentable y confiable, ya que antes el costo del desarrollo solia ser mas alto de lo normal, y la confiabilidad del programa era dudosa. Además, con este concepto se introdujo una forma de desarrollar el software de forma sistemática, disciplinada, mejorar el mantenimiento. También cuando se desarrollan sistemas grandes o complejos, se lo debe hacer en equipos de personas capaces de desarrollarlo bien. Yo creo que actualmente se solucionaron las mayorías de deficiencias durante el desarrollo de software, pero a veces suele aparecer algunas. Estas deficiencias pueden aparecer en la especificación, el diseño del software o cuando no se lo prueba correctamente, la comunicación del equipo de desarrollo no suele ser buena, al escalar el sistema se aumenta la complejidad de este, etc. O también suelen aparecer cuando hay dificultades durante la implementación del diseño o al usar técnicas o metodologías de desarrollos que no son aptas para el tipo de sistema que se desarrolla.	Respuesta guardada	
3	13/10/2021 16:16	Intento finalizado	Finalizado	

1,5

¿Por qué un proceso iterativo hace más fácil administrar el cambio? ¿El framework SCRUM propone un proceso iterativo, incremental, o ambos? Explique estos conceptos usando los elementos (seremonias y artefactos) que SCRUM propone.

Un proceso iterativo hace mas facil administrar el cambio porque al realizar un desarrollo de forma iterativa, se pueden introducir nuevos cambios en cada iteracion y obtener una demo de un programa despues de cada iteracion donde se puede probarlo y corregir los errores necesarios.

El framework SCRUM propone un proceso iterativo e incremental. El desarrollo de un programa se realiza de forma iterativa (sprints) que dan un resultado. Antes de realizar una iteracion se debe realizar un sprint backlog, que consiste en un plan para los desarrolladores hecho por ellos mismos donde se refleja el progreso en tiempo real del desarrollo. Durante un sprint o iteracion se realizan reuniones cortas que se denominan "daily scrum" donde se informa la situacion actual del equipo y el desarrollo para detectar si hay algun problema o reportar progresos. Luego al resultado de un sprint se le hace una revision donde se le presenta una demo al cliente para informarle los avances o cambios que se introdujeron. De esta revision se obtiene una validacion de parte del cliente y este puede agregar nuevas tareas, proponer cambios o modificaciones. Y al final, cuando termina la revision, se realiza un incremento del software, donde se le entrega al cliente la ultima version funcional para su uso.

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 15:21	Guardada: Un proceso iterativo hace mas facil administrar el cambio porque al realizar un desarrollo de forma iterativa, se pueden introducir nuevos cambios en cada iteracion y obtener una demo de un programa despues de cada iteracion donde se puede probarlo y corregir los errores necesarios. El framework SCRUM propone un proceso iterativo e incremental. El desarrollo de un programa se realiza de forma iterativa (sprints) que dan un resultado. Durante un sprint o iteracion se realizan reuniones cortas que se denominan "daily scrum" donde se informa la situacion actual del equipo y el desarrollo para detectar si hay algun problema o reportar progresos. Luego al resultado de un sprint se le hace una revision donde se le presenta una demo al cliente para informarle los avances o cambios que se introdujeron. De esta revision se obtiene una validacion de parte del cliente y este puede agregar nuevas tareas, proponer cambios o modificaciones. Y al final, cuando termina la revision, se realiza un incremento del software, donde se le entrega al cliente la ultima version funcional para su uso.	Respuesta guardada	

Paso	Hora	Acción	Estado	Puntos
3	13/10/2021 15:51	Guardada: Un proceso iterativo hace mas facil administrar el cambio porque al realizar un desarrollo de forma iterativa, se pueden introducir nuevos cambios en cada iteracion y obtener una demo de un programa despues de cada iteracion donde se puede probarlo y corregir los errores necesarios. El framework SCRUM propone un proceso iterativo e incremental. El desarrollo de un programa se realiza de forma iterativa (sprints) que dan un resultado. Antes de realizar una iteracion se debe realizar un sprint backlog, que consiste en un plan para los desarrolladores hecho por ellos mismos donde se refleja el progreso en tiempo real del desarrollo. Durante un sprint o iteracion se realizan reuniones cortas que se denominan "daily scrum" donde se informa la situacion actual del equipo y el desarrollo para detectar si hay algun problema o reportar progresos. Luego al resultado de un sprint se le hace una revision donde se le presenta una demo al cliente para informarle los avances o cambios que se introdujeron. De esta revision se obtiene una validacion de parte del cliente y este puede agregar nuevas tareas, proponer cambios o modificaciones. Y al final, cuando termina la revision, se realiza un incremento del software, donde se le entrega al cliente la ultima version funcional para su uso.	Respuesta guardada	
4	13/10/2021 16:16	Intento finalizado	Finalizado	



Las pruebas de caja negra nos permiten:
(Puede haber más de una respuesta correcta y las incorrectas tienen puntaje negativo)

- ☒ a. asegurar que la interfaz implementada es la especificada
- ☐ b. asegurar que las condiciones lógicas son adecuadas
- ☐ c. asegurar que los lazos de los algoritmos son adecuados
- ☒ d. comprobar que las funciones del sistema son operativas
- ☐ e. probar caminos independientes de control
- ☒ f. saber que las entradas son adecuadas
- ☐ g. comprobar que las estructuras de datos internas son válidas
- ☒ h. verificar que se produce una salida correcta

Respuesta correcta

Las respuestas correctas son: asegurar que la interfaz implementada es la especificada, comprobar que las funciones del sistema son operativas, saber que las entradas son adecuadas, verificar que se produce una salida correcta

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 14:53	Guardada: asegurar que la interfaz implementada es la especificada ; comprobar que las funciones del sistema son operativas ; saber que las entradas son adecuadas ; verificar que se produce una salida correcta	Respuesta guardada	
3	13/10/2021 16:16	Intento finalizado	Correcta	1,00

Pregunta **5**

Incorrecta

Puntúa 0,00 sobre 1,00



La responsabilidad de un Product Owner se enfoca en lo siguiente:

(Puede haber más de una respuesta correcta y las incorrectas tienen puntaje negativo)

- ☒ a. Escribir historias de usuarios claras y participar en los daily meetings ✗
- ☐ b. Trabajar conjuntamente con los clientes y stakeholders para identificar los requerimientos más importantes
- ☒ c. Estar con el scrum team todo el tiempo por si necesitan clarificar un requerimiento ✗
- ☐ d. Comunicar claramente las historias de usuarios, el estado de un proyecto o release y estrategias a los clientes y stakeholders

Respuesta incorrecta.

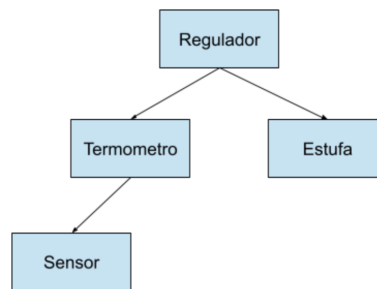
Las respuestas correctas son: Trabajar conjuntamente con los clientes y stakeholders para identificar los requerimientos más importantes, Comunicar claramente las historias de usuarios, el estado de un proyecto o release y estrategias a los clientes y stakeholders

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 14:54	Guardada: Estar con el scrum team todo el tiempo por si necesitan clarificar un requerimiento	Respuesta guardada	
3	13/10/2021 15:09	Guardada: Escribir historias de usuarios claras y participar en los daily meetings ; Estar con el scrum team todo el tiempo por si necesitan clarificar un requerimiento	Respuesta guardada	
4	13/10/2021 16:16	Intento finalizado	Incorrecta	0,00

4.5

Considere el escenario de desarrollo de una aplicación que regula la temperatura de una habitación. Esta incluye 4 módulos:



Se necesita ir testeando incrementalmente los módulos inmediatamente según se van desarrollando.

- 1) Suponga que sólo se terminó de desarrollar el módulo **Regulador**. Se espera que al detectar una temperatura más baja de la indicada, el regulador enciende la estufa. ¿Cómo realizaría la prueba del módulo **Regulador** sin los módulos en los que depende? Necesita escribir algún tipo de código extra al de las pruebas propiamente dicho? Explique con sus propias palabras esta situación y qué tipo de prueba es.
- 2) Como el equipo trabaja de manera concurrente, se terminó el módulo **Sensor** que capta los datos del ambiente. ¿Cómo realizaría la prueba de este módulo? Necesita escribir algún tipo de código extra al de las pruebas propiamente dicho? Explique con sus propias palabras esta situación y qué tipo de prueba es.
- 3) Dada la sucesión de módulos desarrollados hasta el momento. Cuando se termine de desarrollar todo el sistema, ¿puede indicar qué tipo de estrategia de prueba se aplicó?

- 1) Para probar el modulo "regulador", se deben introducir un modulo de resguardo que reemplace al modulo "termometro", el cual se encarga de brindar los datos necesarios al modulo "regulador" para probar su funcionalidad y asi poder activar la "estufa". Este tipo de prueba es una **prueba de integracion descendente**. *✓* *my Estufa*
- 2) Para probar el modulo "sensor", se puede hacer una **prueba de integracion ascendente** del modulo. Se podria incluir una unidad controladora o *driver* reemplazando el "termometro" para comprobar que los datos obtenidos por el sensor sean los correctos. *✓*
- 3) Una vez desarrollado todo el sistema, se realizo una prueba **de integracion "sandwich"** ya que se uso un enfoque mixto de la prueba de integracion (ascendente y descendente). *✓*

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 15:28	Guardada: 1) Para probar el modulo "regulador", se deben introducir un modulo de resguardo que reemplace al modulo "termometro", el cual se encarga de brindar los datos necesarios al modulo "regulador" para probar su funcionalidad y asi poder activar la "estufa".	Respuesta guardada	

Paso	Hora	Acción	Estado	Puntos
3	13/10/2021 15:45	Guardada: 1) Para probar el modulo "regulador", se deben introducir un modulo de resguardo que reemplace al modulo "termometro", el cual se encarga de brindar los datos necesarios al modulo "regulador" para probar su funcionalidad y asi poder activar la "estufa". Este tipo de prueba es una prueba de INTEGRACION DESCENDENTE. 2) Para probar el modulo "sensor", se puede hacer una PRUEBA DE UNIDAD del modulo, donde se prueba la logica interna del modulo para asegurarse de que funcione y tome las mediciones correctamente. Se podria incluir una unidad controladora o driver reemplazando el "termometro" para comprobar que los datos obtenidos por el sensor sean los correctos. 3) Una vez desarrollado todo el sistema, se realiza una prueba COMPLETA DEL SISTEMA para comprobar que funcione todo correctamente.	Respuesta guardada	
4	13/10/2021 15:54	Guardada: 1) Para probar el modulo "regulador", se deben introducir un modulo de resguardo que reemplace al modulo "termometro", el cual se encarga de brindar los datos necesarios al modulo "regulador" para probar su funcionalidad y asi poder activar la "estufa". Este tipo de prueba es una PRUEBA DE INTEGRACION DESCENDENTE. 2) Para probar el modulo "sensor", se puede hacer una PRUEBA DE INTEGRACION ASCENDENTE del modulo. Se podria incluir una unidad controladora o driver reemplazando el "termometro" para comprobar que los datos obtenidos por el sensor sean los correctos. 3) Una vez desarrollado todo el sistema, se realizo una prueba DE INTEGRACION "SANDWICH" ya que se uso un enfoque mixto de la prueba de integracion (ascendente y descendente).	Respuesta guardada	
5	13/10/2021 16:16	Intento finalizado	Finalizado	

0.75

Explique porque una historia de usuario en un sprint backlog debería incluir lo siguientes ítems:

- ID
- Nombre
- Como demostrar (How to demo)

¿Qué ítem o atributo falta y es necesario e indispensable para que una historia de usuario entre a un sprint backlog? ¿Por qué?

Una user story debe incluir un ID para identificar de alguna manera las tareas que se van a desarrollar, un nombre para identificar de que tarea se esta hablando y una demo donde se muestra la tarea desarrollada funcionando.

El atributo que falta en este caso es el "done", donde se define cuando se termino la tarea y se especifica que se obtiene dicho resultado de la tarea.

- la estimación y la prioridad de una historia de usuario es indispensable para una hist. de usuario entre a un sprint.

- el atributo sería el estado de la tarea
- cuando se termino se refleja en el burndown chart.

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 15:57	Guardada: Una user story	Respuesta guardada	
3	13/10/2021 16:11	Guardada: Una user story debe incluir un ID para identificar de alguna manera las tareas que se van a desarrollar, un nombre para identificar de que tarea se esta hablando y una demo donde se muestra como se desarrollo la tarea. El atributo que	Respuesta guardada	
4	13/10/2021 16:14	Guardada: Una user story debe incluir un ID para identificar de alguna manera las tareas que se van a desarrollar, un nombre para identificar de que tarea se esta hablando y una demo donde se muestra la tarea desarrollada funcionando. El atributo que falta en este caso es es el "done", donde se define cuando se termino la tarea y se especifica que se obtiene dicho resultado de la tarea.	Respuesta guardada	
5	13/10/2021 16:16	Intento finalizado	Finalizado	

Pregunta 8

Correcta

Puntúa 1,00 sobre 1,00

1,0

Las pruebas de caja blanca nos permiten:

(Puede haber más de una respuesta correcta y las incorrectas tienen puntaje negativo)

- ☐ a. verificar que se produce una salida correcta
- ☒ b. probar caminos independientes de control
- ☐ c. comprobar que las funciones del sistema son operativas
- ☒ d. comprobar que las estructuras de datos internas son válidas
- ☐ e. saber que las entradas son adecuadas
- ☒ f. asegurar que las condiciones lógicas son adecuadas
- ☒ g. asegurar que los lazos de los algoritmos son adecuados
- ☐ h. asegurar que la interfaz implementada es la especificada



Respuesta correcta

Las respuestas correctas son: asegurar que los lazos de los algoritmos son adecuados, asegurar que las condiciones lógicas son adecuadas, probar caminos independientes de control, comprobar que las estructuras de datos internas son válidas

Historial de respuestas

Paso	Hora	Acción	Estado	Puntos
1	13/10/2021 14:33	Iniciado/a	Sin responder aún	
2	13/10/2021 14:56	Guardada: probar caminos independientes de control ; comprobar que las estructuras de datos internas son válidas ; asegurar que las condiciones lógicas son adecuadas ; asegurar que los lazos de los algoritmos son adecuados	Respuesta guardada	
3	13/10/2021 16:16	Intento finalizado	Correcta	1,00

◀ <https://scrumguides.org/download.html>

Ir a...

Apuntes: Arquitectura de Software y Requerimientos (oculto) ▶