

Universidad Pedagógica de el Salvador.

Cátedra:

**Ingeniería en sistema de
software.**

Alumno:

Alex Balmore Gómez Cornejo.

Tema:

**Mitos y desarrollo del
Software.**

Universidad Pedagógica de el Salvador.

Cátedra:

Ingeniería en sistema de software.

Alumno:

Alex Balmore Gómez Cornejo.

Tema:

**Mitos y desarrollo del
Software.**

Índice.

Introducción.	3
Objetivos.	4
Mitos del software.	5-6
Mitos de gestión.	7
Mitos del cliente.	8
Desarrollo ágil.	9
Procesos xp.	10
DAS.	11
Scrum.	12
Procesos escrum.	13
Conclusión.	14

Fin.

ALEX BALMORE GOMEZ CORNEJO.

Índice.

Introducción. 3 Objetivos. 4 Mitos del software. 5-6 Mitos de gestión. 7 Mitos del cliente. 8 Desarrollo ágil. 9 Procesos xp. 10 DAS. 11 Scrum. 12 Procesos escrum. 13 Conclusión. 14 Fin.

Introducción.

Actualmente en nuestro País resaltan personajes con buena calidad de software

La humanidad sin creerlo pero nosotros tenemos la capacidad de ser grandes desarrolladores de ello surge la necesidad de comprender como evaluar u software dejando de fuera mitos surgidos en muchos casos por la manera de cómo fue desarrollado el sistema

ALEX BALMORE GOMEZ CORNEJO.

Introducción.

Actualmente en nuestro País resaltan personajes con buena calidad de software

La humanidad sin creerlo pero nosotros tenemos la capacidad de ser grandes desarrolladores de ello surge la necesidad de comprender como evaluar u software dejando de fuera mitos surgidos en muchos casos por la manera de cómo fue desarrollado el sistema

Objetivos.

General.

Desarrollo del software.

Específico.

- **Comprendes los mitos de software.**
- **Tipos de mitos que surgen y como invadirlos.**
- **Indagar sobre otros mitos de programación y procesos.**

ALEX BALMORE GOMEZ CORNEJO.

Objetivos.

General.

Desarrollo del software.

Específico.

- **Comprendes los mitos de software.**
- **Tipos de mitos que surgen y como invadirlos.**
- **Indagar sobre otros mitos de programación y procesos.**

Mitos Del Software (admón.).

Los mitos del software-creencias del desarrollo acerca del software y de los procesos empleados para construirlo- se pueden rastrear hasta los primeros días de la computación del desarrollo. Los mitos tienen ciertos atributos que los convierten en reales no cambiantes.

Muchas de las causas de la crisis del software pueden ser encontradas en una mitología que surge durante los primeros años del desarrollo del software. Los mitos del software propagaron información errónea y con fusión.

Mito:

Si fallamos en la planificación podemos añadir más programadores y recuperar el tiempo perdido.

Realidad:

Ley de Brooks: "Agregar gente a un proyecto atrasado, lo atrasa aún más".

Razón: Crear software no es una tarea particional, como dice el Principio de Brooks: "Gestar a un bebé tarda 9 meses, no importa cuántas mujeres sean asignadas a la tarea.

ALEX BALMORE GOMEZ CORNEJO.

Mitos Del Software

(admón.).

Los mitos del software-creencias del desarrollo acerca del software y de los procesos empleados para construirlo- se pueden rastrear hasta los primeros días de la computación del desarrollo. Los mitos tienen ciertos atributos que los convierten en reales no cambiantes. Muchas de las causas de la crisis del software pueden ser encontradas en una mitología que surge durante los primeros años del desarrollo del software Los mitos del software propagaron información errónea y con fusión

Mito:

Si fallamos en la planificación podemos añadir más programadores y recuperar el tiempo perdido. Realidad:

Ley de Brooks: "Agregar gente a un proyecto atrasado, lo atrasa aún más". Razón: Crear software no es una tarea particional, como dice el Principio de Brooks: "Gestar a un bebé tarda 9 meses, no importa cuántas mujeres sean asignadas a la tarea.

Mito:

Una declaración general de los objetivos es suficiente para comenzar a escribir los programas; podemos dar los detalles más adelante.

Realidad:

Una mala definición inicial es la principal causa del trabajo en vano. Es esencial una descripción formal y detallada del ámbito de la información, funciones, rendimiento, interfaces y criterios de validación. Esto solo puede determinarse después de una exhaustiva comunicación entre el cliente y el analista.

Mito:

Una vez que hicimos el programa y funciona, nuestro trabajo ha terminado.

Realidad:

Los datos industriales indican que entre el 50% y el 70% de todo el esfuerzo dedicado a un programa se realizará después de que se le haya entregado al cliente por primera vez



Mito: Una declaración general de los objetivos es suficiente para comenzar a escribir los programas; podemos dar los detalles más adelante.

Realidad:

Una mala definición inicial es la principal causa del trabajo en vano. Es esencial una descripción formal y detallada del ámbito de la información, funciones, rendimiento, interfaces y criterios de validación. Esto solo puede determinarse después de una exhaustiva comunicación entre el cliente y el analista.

Mito:

Una vez que hicimos el programa y funciona, nuestro trabajo ha terminado. Realidad:

Los datos industriales indican que entre el 50% y el 70% de todo el esfuerzo dedicado a un programa se realizará después de que se le haya entregado al cliente por primera vez

ALEX BALMORE GOMEZ CORNEJO.

Mitos _de Gestión
(Profesional).

Los gestores están normalmente bajo la presión de cumplir presupuestos, hacer que no se retrase el proyecto y mejorar la calidad. El gestor se agarra a un mito del software aun que tal creencia sólo disminuya la presión temporalmente.

Mito:

¿Por qué debemos cambiar nuestra forma de desarrollar el Software? Estamos haciendo el mismo tipo de programación a hora que hace diez años.

Realidad:

Aunque el dominio de la aplicación puede ser el mismo, la demanda de una mayor productividad y calidad, y el papel crítico del software en objetivos comerciales estratégicos, ha aumentado sustancialmente.

ALEX BALMORE GOMEZ CORNEJO.

Mitos _de Gestión

(Profesional).

Los gestores están normalmente bajo la presión de cumplir presupuestos, hacer que no se retrase el proyecto y mejorar la calidad. El gestor se agarra a un mito del software aun que tal creencia sólo disminuya la presión temporalmente.

Mito:

¿Por qué debemos cambiar nuestra forma de desarrollar el Software? Estamos haciendo el mismo tipo de programación a hora que hace diez años.

Realidad:

Aunque el dominio de la aplicación puede ser el mismo, la demanda de una mayor productividad y calidad, y el papel crítico del software en objetivos comerciales estratégicos, ha aumentado sustancialmente.

Mitos_del_cliente.

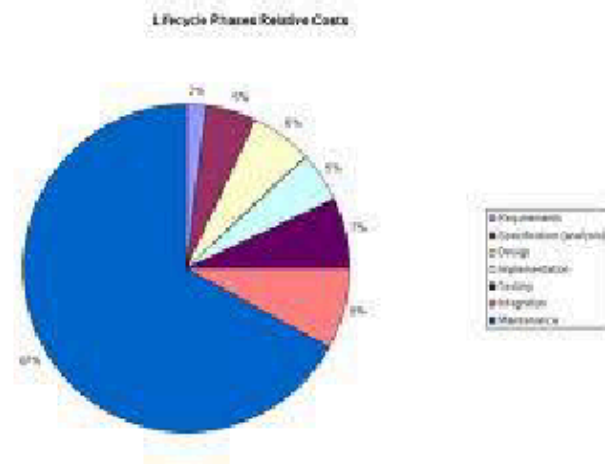
Un cliente que solicita una aplicación software puede ser interno a la compañía o una compañía exterior El cliente cree en los mitos que existen sobre el software debido a que los gestores y trabajadores responsables hacen muy poco para corregir la mala Información Los mitos conducen a que el cliente se cree una falsa expectativa y finalmente, quede insatisfecho con el desarrollo del software

Mito:

Una declaración general de los objetivos es suficiente para comenzar a escribir los programas, podemos dar los detalles más adelante

Realidad:

Una mala definición inicial es la principal causa del trabajo baldío en software. Una descripción formal y detallada del dominio de la información, funciones, rendimiento, interfaces, ligaduras de diseño y criterios de Validación es esencial. Estas características pueden determinarse sólo después de una exhaustiva comunicación entre el cliente y el analista.



Evaluación

Mitos_del_cliente.

Un cliente que solicita una aplicación software puede ser interno a la compañía o una compañía exterior. El cliente cree en los mitos que existen sobre el software debido a que los gestores y trabajadores responsables hacen muy poco para corregir la mala información. Los mitos conducen a que el cliente se cree una falsa expectativa y finalmente, quede insatisfecho con el desarrollo del software.

Mito:

Una declaración general de los objetivos es suficiente para comenzar a escribir los programas, podemos dar los detalles más adelante.

Realidad:

Una mala definición inicial es la principal causa del trabajo baldío en software. Una descripción formal y detallada del dominio de la información, funciones, rendimiento, interfaces, ligaduras de diseño y criterios de Validación es esencial. Estas características pueden determinarse sólo después de una exhaustiva comunicación entre el cliente y el analista.

Evaluación

ALEX BALMORE GOMEZ CORNEJO.

Desarrollo ágil.

¿Qué es?

Podría decirse que es el trabajo desarrollado para la realización de un proyecto. Hay innumerables métodos de desarrollo ágil pero todos tienen un fin que es crear un software confiable. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas.

Esto consta de:

Planificación.
Análisis de requerimientos.
Diseño.
Codificación.
Revisión y documentación.

Lo que se pretende con esto es crear un demo el cual se haya desarrollado sin ningún error al momento de la ejecución. Estos métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación.

Programación extrema xp.

Este método de programación está basado en metodologías tradicionales adaptables a los cambios, en cualquier etapa del desarrollo. Es considerado como la mejor metodología de desarrollo todo por lo que esta de acorde y se puede modificar y incluso darle cambios drásticos sin que el obtenga daños en ninguna de sus etapas.

ALEX BALMORE GOMEZ CORNEJO.

Desarrollo ágil.

¿Qué es?

Podría decirse que es el trabajo desarrollado para la realización de un proyecto. Hay innumerables métodos de desarrollo ágil pero todos tienen un fin que es crear un software confiable. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas.

Esto consta de:

Planificación.

Análisis de requerimientos.

Diseño.

Codificación.

Revisión y documentación.

Lo que se pretende con esto es crear un demo el cual se haya desarrollado sin ningún error al momento de la ejecución. Estos métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación.

Programación extrema xp.

Este método de programación está basado en metodologías tradicionales adaptables a los cambios, en cualquier etapa del desarrollo. Es considerado como la mejor metodología de desarrollo todo por lo que esta de acorde y se puede modificar y incluso darle cambios drásticos sin que el obtenga daños en ninguna de sus etapas.

Procesos de xp.

Codificar

Es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.

Hacer pruebas

Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.

Escuchar

Si vamos a hacer pruebas tenemos que preguntar si lo obtenido es lo deseado, y tenemos que preguntar a quien necesita la información. Tenemos que escuchar a nuestros clientes cuáles son los problemas de su negocio, debemos de tener una escucha activa explicando lo que es fácil y difícil de obtener, y la realimentación entre ambos nos ayudan a todos a entender los problemas.

Diseñar

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

Resumiendo las actividades de Xp: Tenemos que codificar porque sin código no hay programas, tenemos que hacer pruebas por que sin pruebas no sabemos si hemos acabado de codificar, tenemos que escuchar, porque si no escuchamos no sabemos qué codificar ni probar, y tenemos que diseñar para poder codificar, probar y escuchar indefinidamente.

ALEX BALMORE GOMEZ CORNEJO.

Procesos de xp.

Codificar

Es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.

Hacer pruebas

Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.

Escuchar

Si vamos a hacer pruebas tenemos que preguntar si lo obtenido es lo deseado, y tenemos que preguntar a quien necesita la información. Tenemos que escuchar a nuestros clientes cuáles son los problemas de su negocio, debemos de tener una escucha activa explicando lo que es fácil y difícil de obtener, y la realimentación entre ambos nos ayudan a todos a entender los problemas.

Diseñar

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

Resumiendo las actividades de Xp: Tenemos que codificar porque sin código no hay programas, tenemos que hacer pruebas por que sin pruebas no sabemos si hemos acabado de codificar, tenemos que escuchar, porque si no escuchamos no sabemos qué codificar ni probar, y tenemos que diseñar para poder codificar, probar y escuchar indefinidamente.