



Buenas prácticas de software



UNIDAD 3 Mejores prácticas para la ingeniería de diseño.



Buenas prácticas de software



3.2 Mejores prácticas para la planeación del desarrollo del producto.



Buenas prácticas de software



3.2 Mejores prácticas para la planeación del desarrollo del producto.

Vamos a describir 4 pasos para describir la planeación de proyectos:

- Las actividades a realizar.
- Hacer roles al interior del equipo para asignar tareas.
- Dar seguimiento a las tareas.
- Fijar compromisos al interior del equipo dando seguimiento a las tareas.

Para cumplir con los puntos anteriores se necesitan herramientas para la planeación.



Buenas prácticas de software



3.2.1 Herramientas abiertas para la planeación del proyecto.

Características de las herramientas para la planeación del proyecto.

Accesible.

El tener acceso a la planeación del proyecto desde todo tipo de dispositivo permite tener el seguimiento y cumplimiento de los entregables

Colaborativa

La retroalimentación es importante con el cliente, es por ello que las herramientas permiten tener colaboración y retroalimentación con el usuario final.

Usable

Hay que facilitar la descarga de documentos en diferentes formatos para asegurar su lectura en el mayor número de dispositivos posible.

Organizada

La asignación de responsables y de roles al interior del equipo permite tener asignadas tareas donde cada responsable debe cumplir en tiempo y forma con sus funciones.



Buenas prácticas de software



3.2.1 Herramientas abiertas para la planeación del proyecto.

Visual

La planeación del proyecto permite tener el seguimiento puntual de las actividades.

En tiempo real

La rapidez en la transmisión de información es directamente proporcional a la capacidad de resolver incidencias.

Abierta

La flexibilidad de conectar con otras aplicaciones este proyecto ayuda a mejorar su usabilidad.

Segura

La seguridad de la información así como de los datos es una función que no debe olvidarse en el desarrollo de los sistemas por lo que el desarrollador debe hacer uso de técnicas y métodos que permita la privacidad.



Buenas prácticas de software



Ejemplos de herramientas de acceso abierto.

Herramienta	Accesible	Colaborativa	Usable	Organizada	Visual	Tiempo real	Abierta	Segura
Bubble Plan	x	x	x	x	x		x	x
Bitrix24	x	x	x	x	x		x	x
OpenProject	x	x	x	x	x	x		x



Buenas prácticas de software



3.2.2 Herramientas de pago para la planeación del proyecto.

Ejemplo de herramientas, puedes consultala en la siguiente liga.

https://www.capterra.mx/sem/compare/directory/30002/project-management/software?account_campaign_id=13246187966&account_adgroup_id=131319888188&target=herramienta%20gestio%20proyectos&ad_id=523807643665&matchtype=p&gclid=CjwKCAiAp8iMBhAqEiwAJb94z3jNNdNubWUnwq28DsFZ3ZlmZgcTQWefd6ci7raThjw3WjnS2S7okxoCW8kQAvD_BwE













Introduce el nombre o categoría de software

Categorías de software

Software de gestión de proyectos

Compara características y valoraciones de los productos para encontrar tu software adecuado. ⓘ

Producto	Gestión de presupuestos	Herramientas de colaboración	Vista de Gantt/cronología	Gestión de carteras	Gestión de recursos	Seguimiento de hitos	Gestión de ideas	Metodología ágiles
 monday.com ★★★★★ (2471)	✓	✓	✓	✓	✓	✓	✓	✓
 Asana ★★★★★ (10084)	✓	✓	✓	✓	✓	✓	✓	✓
 Nifty ★★★★★ (118)	✓	✓	✓	✓	✓	✓	✓	✓
 Wrike ★★★★★ (1687)	✓	✓	✓	✓	✓	✓	✓	✓
 GoodDay ★★★★★ (100)	✓	✓	✓	✓	✓	✓	✓	✓
 Shortcut ★★★★★ (303)	✓	✓	✓	✓	✓	✓	✓	✓
 airfocus ★★★★★ (95)	✓	✓	✓	✓	✓	✓	✓	✓
 SpiraPlan ★★★★★ (18)	✓	✓	✓	✓	✓	✓	✓	✓
 Zoho Projects ★★★★★ (237)	✓	✓	✓	✓	✓	✓	✓	✓
 Celoxis ★★★★★ (246)	✓	✓	✓	✓	✓	✓	✓	✓



Buenas prácticas de software



3.2.4 Dividir los desarrollos en fases o entregables.

¿Qué son las fases de un proyecto?

Son el conjunto de actividades enlistadas con la finalidad de dar seguimiento a un proyecto.

Las actividades se deben enlistar en cada etapa del desarrollo del proyecto en función a la metodología o estrategia de desarrollo.

Fase inicial.

- Fase conceptual.
- Fase intermedia.
 - Fase organizacional
 - Fase ejecutiva
- Fase final.
 - Fase de completación.



Buenas prácticas de software



3.2.4 Dividir los desarrollos en fases o entregables.

Primera Fase

- En esta fase se conceptualiza el proyecto.

Segunda Fase

- Se planifica el periodo.

Tercera Fase

Se arranca el proyecto



Buenas prácticas de software



¿Qué es el Ciclo de vida del proyecto?

Es el conjunto de etapas por las que pasara el desarrollo del proyecto.

Las etapas son:

Análisis o análisis de requerimientos

Diseño o diseño de requerimientos

Desarrollo

Pruebas

Implementación



Buenas prácticas de software



¿Qué son los entregables de un proyecto?

Entregables ("*deliverables*").

Cada fase genera productos y estos son los que se van entregando cuando así esta el acuerdo con el cliente y se definen como el resultado de la planeación del proyecto.

Algunas ocasiones estos son materiales como pueden ser equipos o inmateriales que pueden ser documentos o software.

Estos entregables permiten medir el avance del proyecto para algunas ocasiones detectar posibles errores.



Buenas prácticas de software



Tipos de modelos de ciclos de vida

Prototipos

Es una producción del software que se elabora con los requerimientos iniciales.



Buenas prácticas de software



Etapas en las que se llevan para desarrollar un prototipo





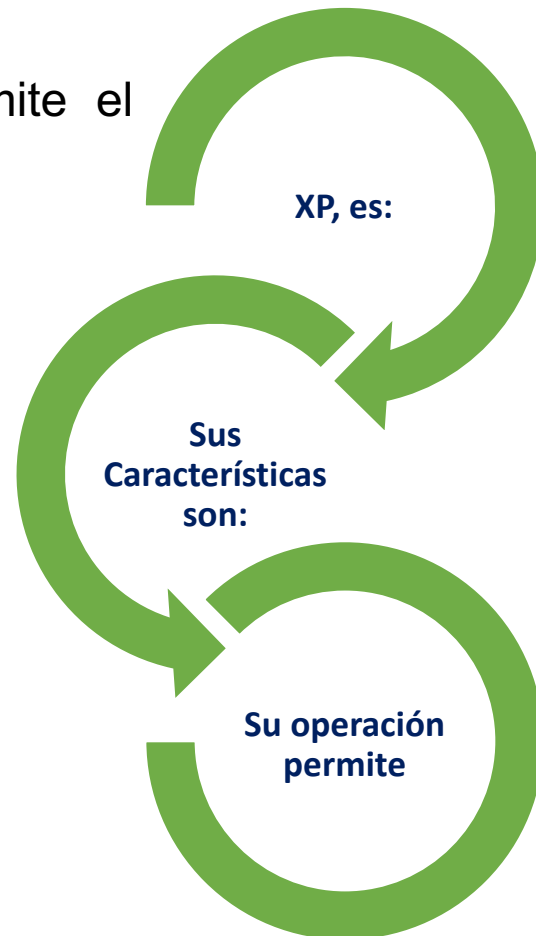
Buenas prácticas de software



XP (Xtreme Programming)

Metodología de desarrollo ágil que permite el desarrollo del software.

Desarrollo de aplicaciones sencillas.



Mejoras en el software



Buenas prácticas de software



Fases para la construcción mediante metodologías de desarrollo rápido (RAD)

Inicia con el desarrollo de la aplicación o software, utilizando Herramientas Case.

Generación de la aplicación

Prueba de la entrega de la aplicación

Permite comprobar requerimientos del modelado de gestión.

Liberación de la aplicación

Al concluir con la fase de prueba se entrega la aplicación junto con el manual de desarrollo, porque aunque se elabore con una metodología de desarrollo rápido se debe elaborar el manual al concluir el proyecto, para su mantenimiento o futura aplicación o adecuación.