

Son muchas las definiciones que se pueden encontrar sobre qué es una metodología de desarrollo de software, tan es así que hay quienes hablan de ella como organización sistemática para el ciclo de vida del sistema y sus partes, pero también hay quienes la definen como conjunto de herramientas, procedimientos y técnicas para lograr el desarrollo de un nuevo software.

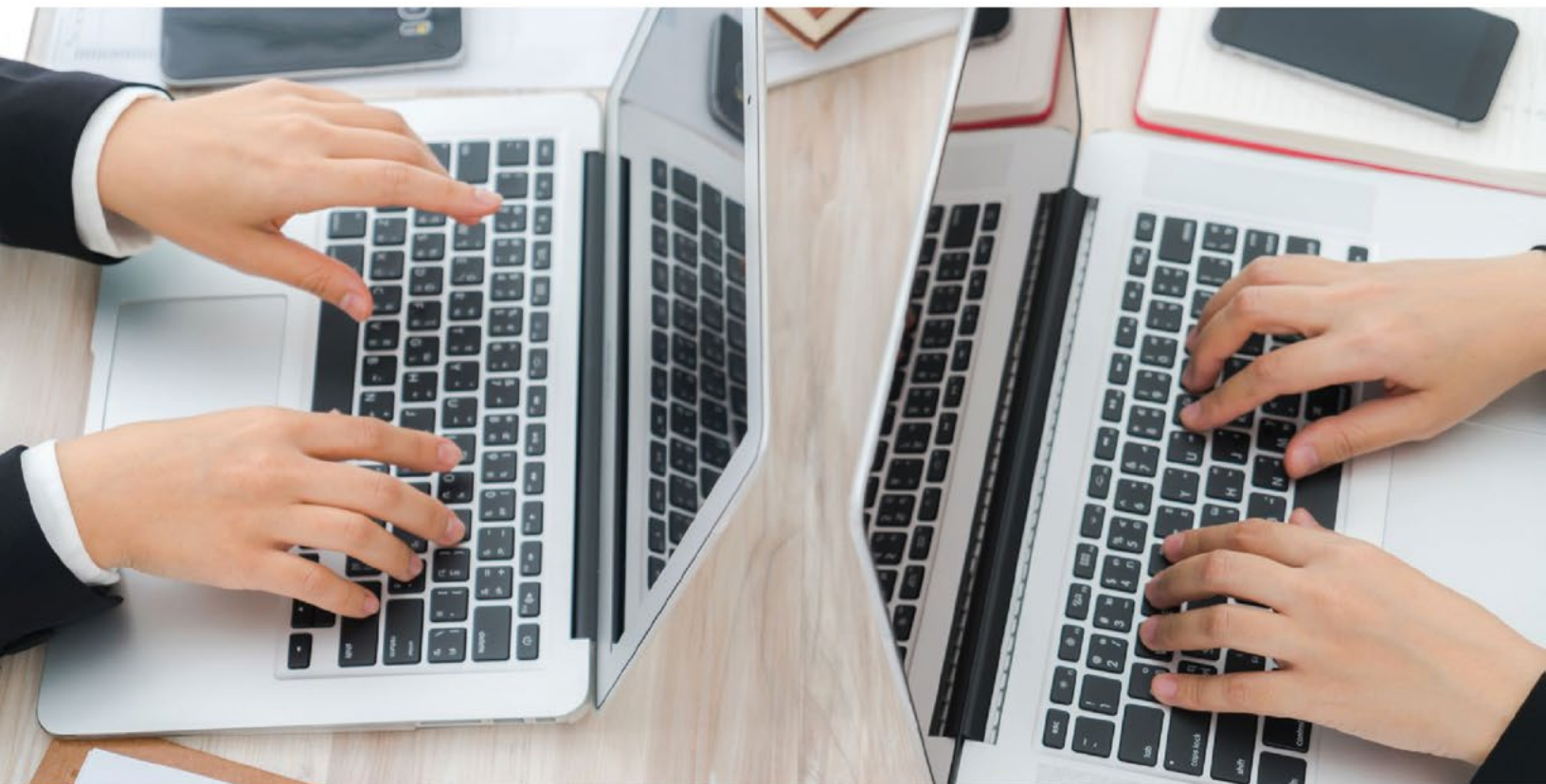
Sin embargo, para hacerlo sencillo, se puede definir la metodología de software como un enfoque, una manera de interpretar la realidad o la disciplina en cuestión, que en este caso particular correspondería a la Ingeniería de Software. De hecho, la metodología destinada al desarrollo de software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializada.

¿Cuáles son los objetivos de una Metodología de Desarrollo de Software?

- ❏ Establecer acertadamente cada uno de los requisitos de un sistema software.
- ❏ Suministrar un método sistemático de desarrollo de manera tal que se pueda controlar su proceso.
- ❏ Hacer la construcción de un sistema de software dentro de un tiempo apropiado y costos aceptables.
- ❏ Hacer la construcción de un sistema que además de estar bien documentado, sea fácil de mantener.
- ❏ Ayudar a identificar, lo antes posible, cualquier cambio que sea necesario realizar dentro del proceso de desarrollo.
- ❏ Proveer un sistema que satisfaga a las personas afectadas por el mismo

¿Qué aspectos debería cubrir una Metodología de Desarrollo de Software?

- Un proceso de ciclo de vida completo, es decir, uno que comprenda aspectos tanto del negocio como aquellos que son de origen técnico.
- Un conjunto completo de conceptos y modelos que sean internamente consistente.
- Una descripción completa de artefactos a desarrollar
- Un conjunto de técnicas probadas.
Identificación de los roles organizacionales
- Guías para la gestión de proyectos y aseguramiento de la calidad
Asesoramiento para la gestión de bibliotecas y reutilización.



2. MODELO DE CASCADA

Cada etapa de este enfoque representa una unidad de desarrollo con un pequeño descanso en el medio. Por tal motivo, cada etapa que sigue comienza tan pronto como la anterior está terminada.

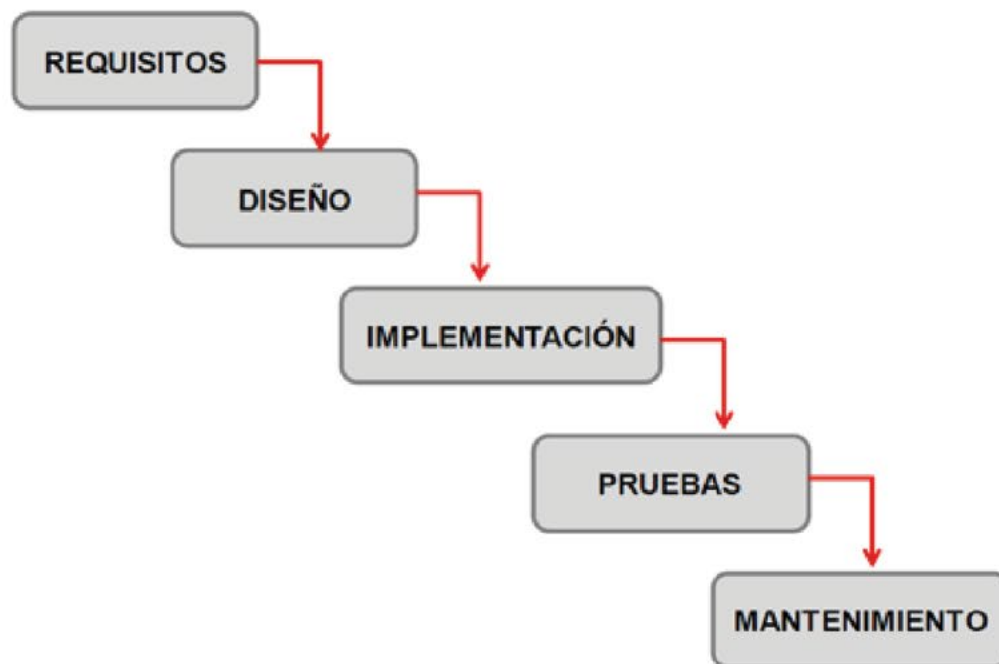
Adicionalmente, este es considerado como el método tradicional de explicar el proceso de desarrollo de software en ingeniería de software, por lo que actualmente es visto como anticuado. Sin embargo, aún sigue siendo aplicado a proyectos con metas claras y requisitos que demandan hasta 100 horas de desarrollo, sobre todo considerando que este enfoque permite a los negocios deshacerse del papeleo innecesario, reuniones regulares que consumen mucho tiempo y retrasos en sus procesos de negocio.

Por lo tanto, esta es una gran opción para los pequeños proyectos donde todos los aspectos del desarrollo son conocidos de antemano y una mala solución para los proyectos complicados, ya que es bastante inflexible, característica que ha sido fuente de crítica por parte de los defensores de modelos más flexibles.



Algunas características del Modelo de Cascada

- Sólo cuando se finaliza una fase, comienza la otra.
- En ocasiones se realiza una revisión antes de iniciar la siguiente fase, lo que permite la posibilidad de cambios (lo que puede incluir un proceso de control formal de cambio).
- Las revisiones también se utilizan para asegurar que la fase anterior ha sido totalmente finalizada.
- Los criterios para completar una fase se conocen frecuentemente con el término inglés "gate" (puerta).
- Este modelo desaconseja visitar y revisar fases que ya se han completado.

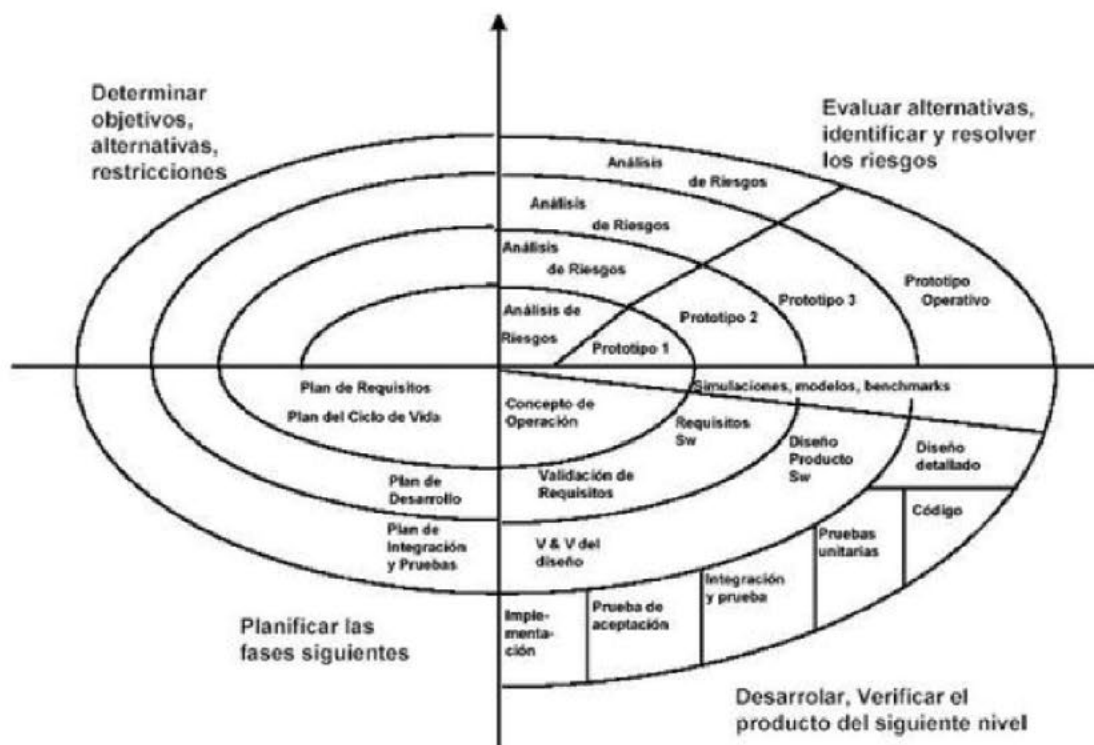


3. MODELO DE ESPIRAL

Mientras que el modelo de cascada ofrece una estructura ordenada para el desarrollo de software, las demandas de tiempo reducido al mercado hacen que sus pasos en serie sean inapropiados.

El siguiente paso evolutivo desde la cascada es donde se realizan los diversos pasos para múltiples entregas o traspasos. La última evolución de la caída del agua es la espiral, aprovechando el hecho de que los proyectos de desarrollo funcionan mejor cuando son incrementales e iterativos.

La metodología espiral refleja la relación de tareas con prototipos rápidos, mayor paralelismo y concurrencia en las actividades de diseño y construcción. El método en espiral debe todavía ser planificado metódicamente, con las tareas y entregables identificados para cada paso en la espiral.



Es un procedimiento de desarrollo especializado que permite a los desarrolladores la posibilidad de poder solo hacer la muestra de la resolución para poder validar su esencia funcional ante los clientes, y hacer los cambios que sean fundamentales antes de crear la solución final auténtica. De hecho, la mejor parte de esta metodología es que tiende a resolver un conjunto de problemas de diversificación que ocurren con el método de la cascada.

Además de esto, la gran ventaja de optar por este enfoque es que da una idea clara sobre el proceso funcional del software, reduce el riesgo de falla en una funcionalidad de software y asiste bien en la recolección de requisitos y en el análisis general.

Para ilustrar mejor lo hasta aquí dicho, se muestra a continuación algunos aspectos básicos de este modelo:

```
63 while (i < 10) {
64     var elementId = 'foto' + i;
65     var elementIdBig = 'bigImage' + i;
66     if (page * 9 + i - 1 < photos.length) {
67         document.getElementById( elementId ).src = 'images/min/' + photos[i];
68         document.getElementById( elementIdBig ).src = 'images/max/' + photos[i];
69     } else {
70         document.getElementById( elementId ).src = '';
```

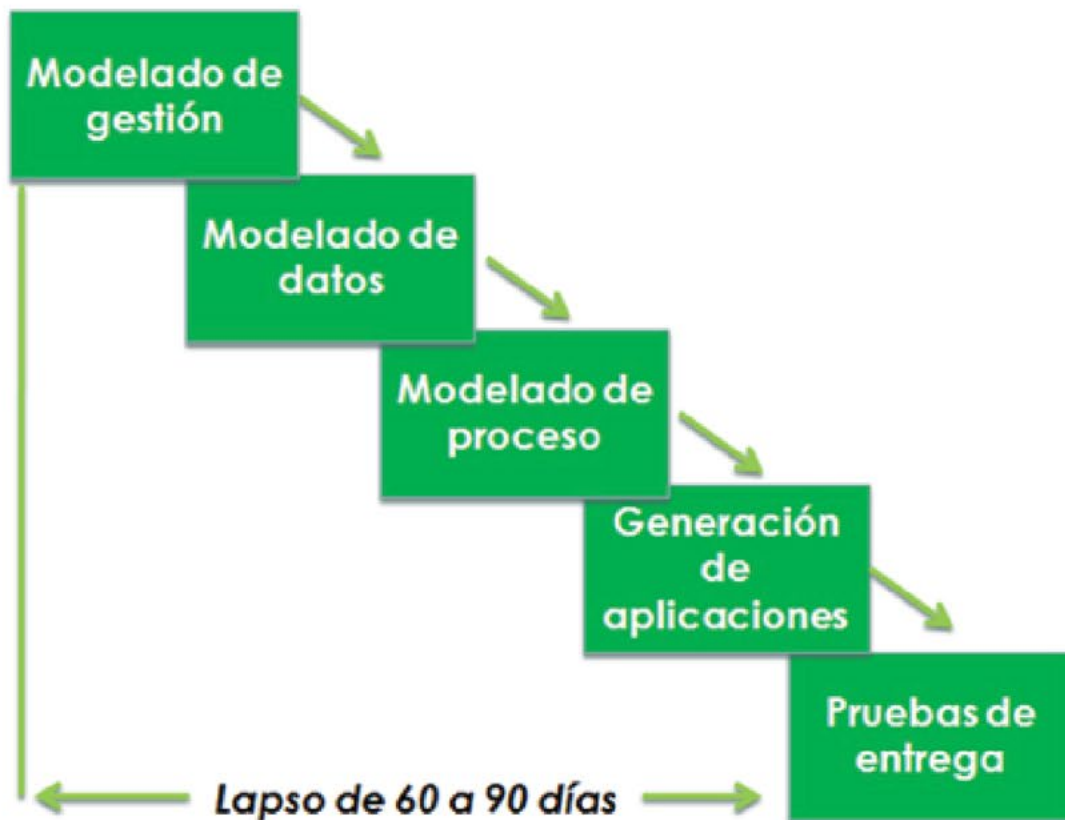
- Se trata de un modelo que ayuda a reducir los riesgos inherentes del proyecto estableciendo el desarrollo en fragmentos más pequeños y logrando, en un entorno propenso a cambios, que estos tengan menor impacto.
- El usuario involucrado durante el desarrollo (probando prototipos) incrementa la aceptación de la implementación del producto final.
- Pequeños prototipos con modificaciones son mostrados al cliente y sirve para confirmar que se han comprendido sus requisitos.
- Muchos de los prototipos se generan con la expectativa de ser descartados. Sin embargo, en algunos casos el prototipo puede evolucionar y convertirse en el producto final.
- Es necesario un entendimiento fundamental de los problemas del negocio para evitar resolver los problemas incorrectos, despilfarrando esfuerzo al desarrollar prototipos que son prescindibles.



Con el objetivo de otorgar resultados rápidos, se trata de un enfoque que está destinado a proporcionar un excelente proceso de desarrollo con la ayuda de otros enfoques, pero además, está diseñado para aumentar la viabilidad de todo el procedimiento de desarrollo de software para resaltar la participación de un usuario activo.

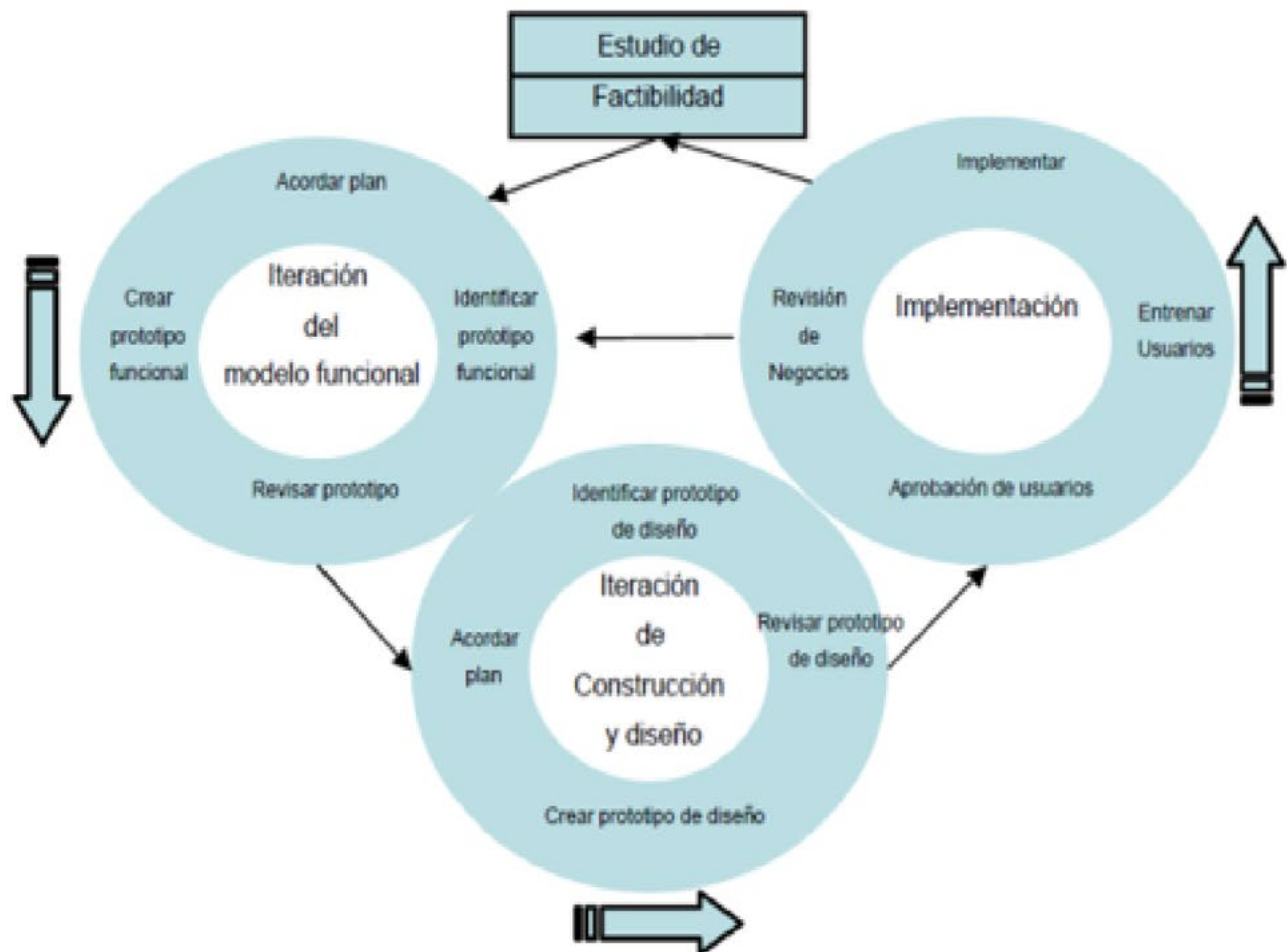
Dicho esto, algunas de las ventajas a destacar de este tipo de desarrollo son las siguientes:

- Hace todo el proceso de desarrollo sin esfuerzo.
- Asiste al cliente en la realización de revisiones rápidas.
- Alienta la retroalimentación de los clientes para su mejora.



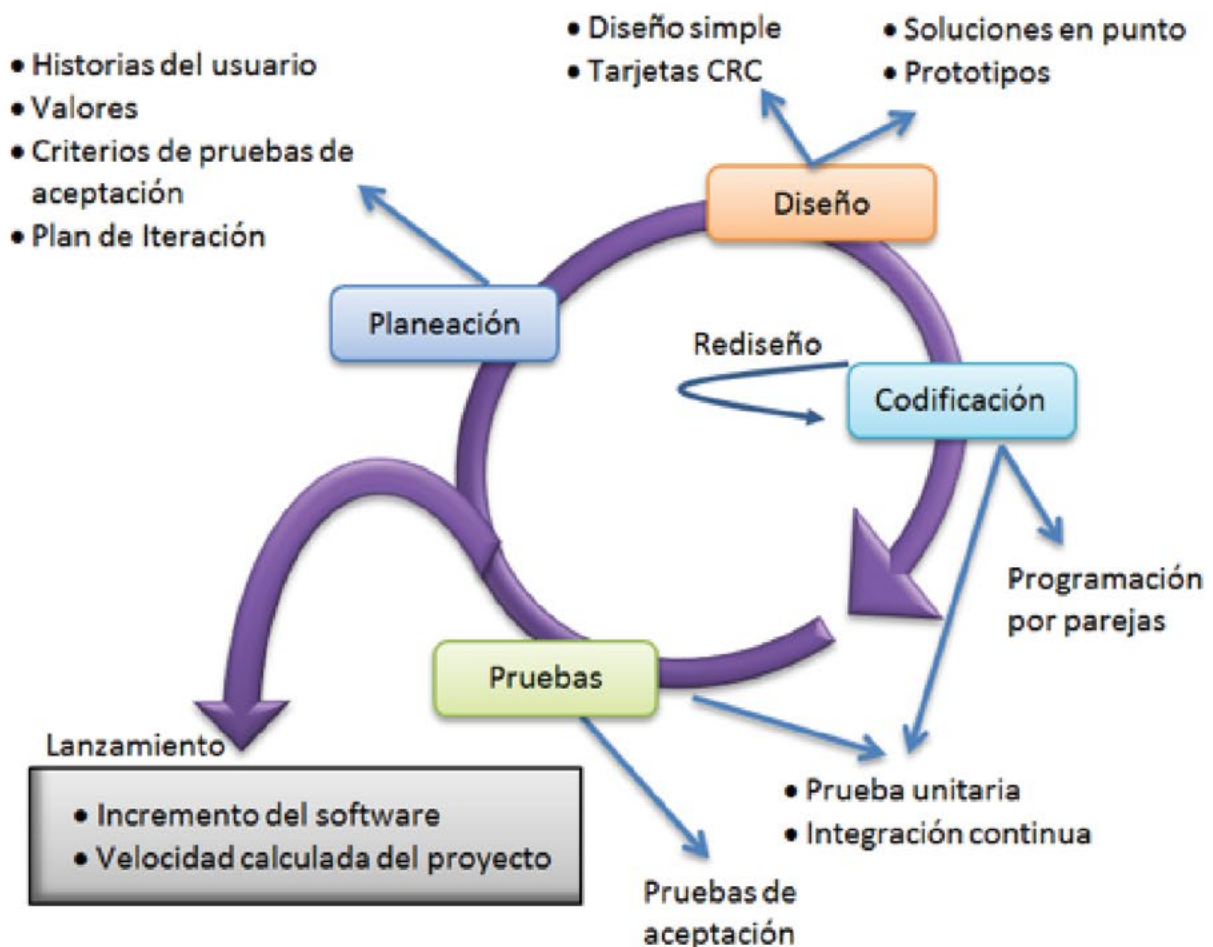
6. MÉTODO DE DESARROLLO DE SISTEMAS DINÁMICOS

Auténticamente formulado y derivado de la Metodología de Desarrollo Rápido de Aplicaciones. Este se trata de un enfoque iterativo e incremental que se centra en la participación del usuario, y cuya tarea es proporcionar sistemas de desarrollo de software dentro del plazo especificado y el presupuesto asignado. Siendo esta la principal razón por la que es una metodología con un alta demanda dentro del mundo del desarrollo de software.



7.METODOLOGÍA DE PROGRAMACIÓN EXTREMA (XP)

Como metodología ágil de ingeniería de software, la metodología de programación extrema se conoce actualmente como metodología de XP (eXtreme Programming). Esta metodología, se utiliza principalmente para evitar el desarrollo de funciones que actualmente no se necesitan, pero sobre todo para atender proyectos complicados. Sin embargo, sus métodos peculiares pueden tomar más tiempo, así como recursos humanos en comparación con otros enfoques.

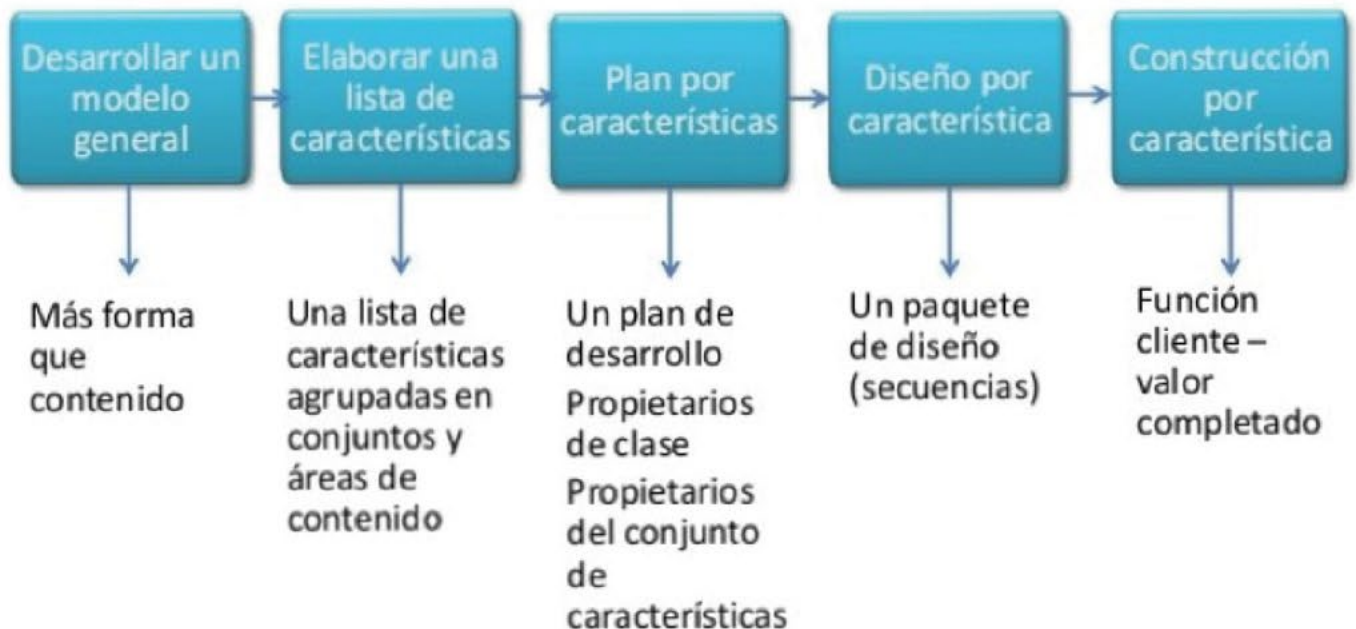


8.DESARROLLO CONDUCIDO POR CARACTERÍSTICAS

Se trata de una metodología iterativa para el desarrollo de software, la cual está dirigida a servir a un gran número de equipos que trabajan en un proyecto basado en la tecnología orientada a objetos.

Este tipo de modelo es bueno para las empresas que están pasando de un método basado en la fase a un enfoque iterativo.

Actualmente se conoce como una metodología FDD (por su nombre en inglés, Feature Driven Development) y es altamente funcional y creativo, por lo menos lo suficiente para hacer frente a diversas complejidades.



Scrum es un enfoque bastante flexible que ayuda al equipo a reaccionar rápidamente en los diferentes cambios en los requisitos, pero que además provee la gran ventaja de que puede ser aplicado consecutivamente a todos los proyectos. Es por esto que viene siendo adecuado para aquellos proyectos de desarrollo que están constantemente cambiando o desarrollando requisitos.

El modelo de desarrollo Scrum se inicia con una planificación efímera, una conferencia y se completa con una revisión final. Esta metodología de crecimiento se utiliza para el desarrollo rápido de software que pasa a incluir una serie de iteraciones para generar el software necesario, por lo que este toma su lugar entre las metodologías flexibles que son apropiadas para el desarrollo a largo plazo con frecuentes cambios en los requerimientos.



10. MODELO DE MADUREZ DE CAPACIDADES DE INTEGRACIÓN (CMMI)

No se trata de una metodología, pero resulta importante hacer mención de este modelo de procesos porque este viene siendo fundamental para que el desarrollo de software sea realmente óptimo, sobre todo teniendo en cuenta que precisamente este el que contiene las mejores prácticas de la industria del desarrollo de software, tanto para el desarrollo del mismo, como para su mantenimiento, adquisición y operación de productos y servicios.

CMMI (por Capability Maturity Model Integration) es un modelo que además de contener las mejores prácticas para el desarrollo de software, provee a las organizaciones de aquellos elementos que son esenciales para que sus procesos de negocio sean realmente efectivos.

Este modelo fue inicialmente desarrollado para los procesos relativos al desarrollo e implementación de Software por la Carnegie-Mellon University. Este vio la luz por primera vez en el año 1987 como Capability Maturity Model CMM. Dicho nombre, tanto como los cinco niveles de la representación por etapas, están inspirados en el modelo de madurez Manufacturing Maturity Model de Crosby.

En principio el modelo CMM era aplicado en programas de defensa, pero lo cierto es que este ha logrado gran aceptación, tan es así que ha sido sometido a varias revisiones e iteraciones. Debido a su éxito se llevó a cabo el desarrollo de modelos CMM para para diversos ámbitos más allá del software.

El problema con esto, es que debido a la gran proliferación de modelos de desarrollo de software comenzaron a surgir confusiones, motivo por el que el gobierno terminó financiando un proyecto de dos años en el que participaron más de 200 expertos del mundo industrial y académico, con el fin de crear un solo marco extensible para la ingeniería de sistemas, la ingeniería de software y el desarrollo de productos ¿el resultado? El modelo más conocido actualmente: CMMI.



CMMI DEV / 2SM
Exp. 2020-03-10 / Appraisal #28791

¿Por qué es prioritario el uso de un modelo para desarrollar un software?

Hacer uso de un modelo de desarrollo de software es fundamental porque es precisamente lo que permite comprender cuáles son los elementos específicos de una organización, a la vez que ayuda a formular y hablar de qué es lo que se debe mejorar dentro de la misma y de cómo se pueden lograr dichas mejoras. Dicho esto, mencionamos las grandes ventajas del uso de un modelo para el desarrollo de proyectos de software:

- Proporciona un marco y un lenguaje común, lo que se traduce en la ruptura de las barreras de la comunicación en el interior de las organizaciones.
- Permite que los usuarios puedan enfocarse específicamente en la mejora, ya que ayudan a que no pierdan la idea global.
- Aporta años de experiencia.
- Ayudan a mejorar la satisfacción del cliente.
- Permiten producir productos y servicios de alta calidad.

Beneficios del modelo CMMI para el desarrollo de software

Hacer uso del modelo CMMI para el desarrollo de software, no solo permite optimizar procesos de negocios, sino que también trae consigo una serie de grandes beneficios que son indispensables para obtener el mejor producto posible, entre ellos los siguientes:

- La gestión y la ingeniería de las actividades se encuentran entrelazadas de una manera explícita, tan es así que facilita el reconocimiento de los objetivos del negocio.
- Permite hacer la incorporación de la experiencia adquirida en otras zonas de las mejores prácticas. Algunos ejemplos serían la medición, gestión de riesgos y de proveedores.
- Poder aplicar prácticas de alta madurez mucho más robustas.
- Cumplir de forma mucho más completa con las normas ISO.