

Modelos de ciclo de vida del software

Omar David Jaimes Molina - 1152263

Bryan Alejandro Vera Osorio - 1152277

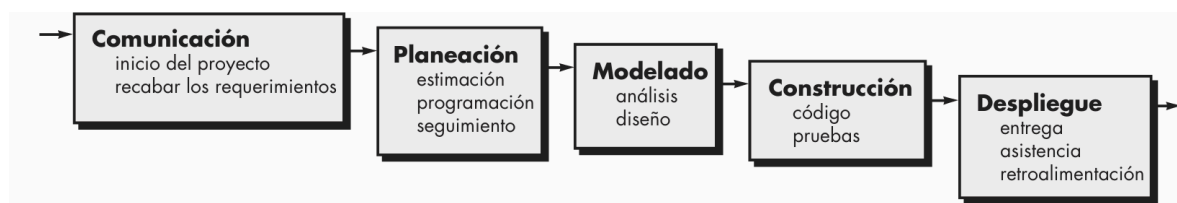
Saimer Adrián Saavedra Rojas - 1152263

Modelos prescriptivos:

Los modelos de proceso prescriptivo fueron propuestos originalmente para poner orden en el caos del desarrollo de software (Pressman, 2010).

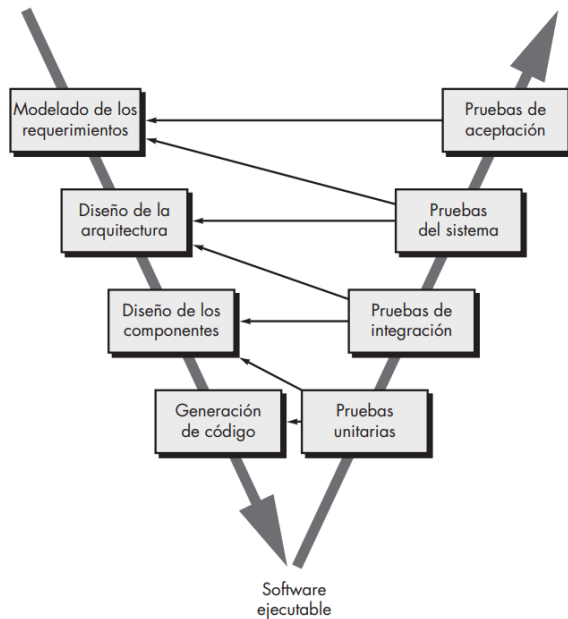
- Modelo de cascada:

El modelo de la cascada, a veces llamado ciclo de vida clásico, sugiere un **enfoque sistemático y secuencial para el desarrollo del software**, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado.



- Modelo en V:

Una variante de la representación del modelo de la cascada se denomina modelo en V. Se aprecia la relación entre las **acciones para el aseguramiento de la calidad y aquellas asociadas con la comunicación, modelado y construcción temprana**. A medida que el equipo de software avanza hacia abajo desde el lado izquierdo de la V, los **requerimientos básicos del problema mejoran hacia representaciones técnicas cada vez más detalladas del problema y de su solución**.

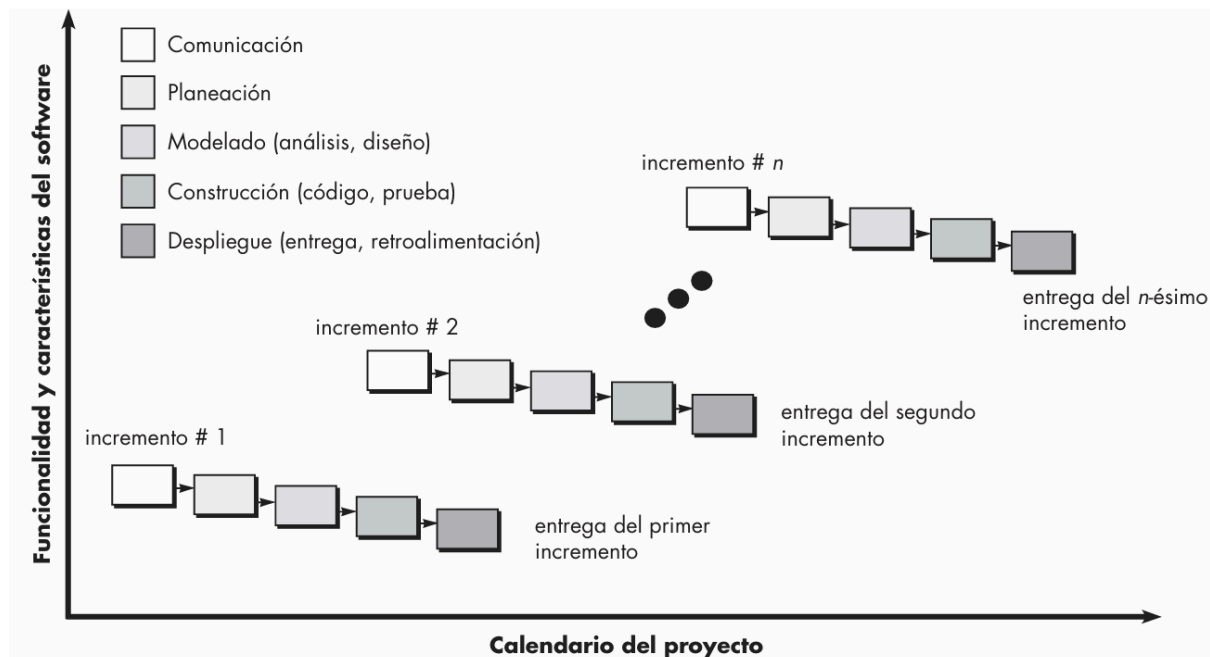


En cada fase de desarrollo se incluyen niveles de pruebas asociados. En este modelo las pruebas comienzan lo más pronto posible, para descubrir rápidamente los posibles errores y no esperar al final para mejorarlo.

- Modelos de proceso incremental:

El desarrollo incremental se basa en la idea de diseñar una **implementación inicial, exponer ésta al comentario del usuario**, y luego desarrollarla en sus diversas versiones hasta producir un sistema adecuado (Sommerville, 2011).

El modelo de proceso incremental se centra en que en cada incremento se entrega un producto que ya opera. Los primeros incrementos son versiones desnudas del producto final, pero proporcionan capacidad que sirve al usuario y también le dan una **plataforma de evaluación** (Pressman, 2010).



- Modelos de proceso evolutivo:

Modelo de proceso diseñado explícitamente para adaptarse a un producto que evoluciona con el tiempo. Los modelos evolutivos son iterativos. Se caracterizan por la manera en la que permiten desarrollar versiones cada vez más completas del software (Pressman, 2010).

Hacer prototipos.

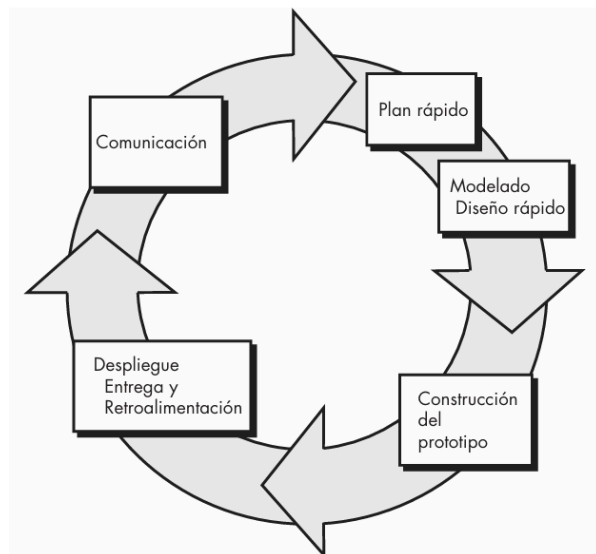
El paradigma de hacer prototipos comienza con la comunicación. Usted se reúne con otros participantes para definir los objetivos generales del software, identifica cualesquiera requerimientos que conozca y detecta las áreas en las que es imprescindible una mayor definición. Se planea rápidamente una iteración para hacer el prototipo, y se lleva a cabo el modelado. Éste se centra en la representación de aquellos aspectos del software que serán visibles para los usuarios finales. El **diseño rápido lleva a la construcción de un prototipo**. Éste se entrega y es evaluado por los participantes, que dan retroalimentación para mejorar los requerimientos. La iteración ocurre a medida de que **el prototipo es afinado para satisfacer las necesidades de distintos participantes, y al mismo tiempo le permite a usted entender mejor lo que se necesita hacer**.

- Modelo de espiral:

El modelo de espiral combina los pequeños **ciclos repetidos** del modelo iterativo con el **flujo secuencial y lineal del modelo de cascada** para dar prioridad al análisis de riesgos. Puede usar el modelo de espiral para garantizar la actualización y mejora graduales del software mediante la creación de prototipos en cada fase (Amazon Web Services, Inc., s.f.).

Ventajas y desventajas

El modelo de espiral es adecuado para proyectos grandes y complejos que requieren cambios frecuentes. Sin embargo, puede ser costoso para proyectos pequeños con objetivos muy concretos.



- Modelo Ágil:

El modelo ágil dispone las fases del SDLC (Ciclo de vida de desarrollo de software) en varios ciclos de desarrollo. El equipo **itera a través de las fases rápidamente y solo se hacen pequeños cambios progresivos de software en cada ciclo**. Los requisitos, planes y resultados se evalúan continuamente para responder con rapidez a los cambios. El modelo ágil es iterativo y progresivo, por lo que es más eficiente que otros modelos de procesos (Amazon Web Services, Inc., s.f.).

Ventajas y desventajas

Los ciclos rápidos de desarrollo permiten a los equipos identificar y abordar problemas en proyectos complejos desde el principio y antes de que se conviertan en problemas graves. También promueven la participación de los clientes y las partes interesadas para que den su opinión en todo el ciclo de vida del proyecto. Sin embargo, depender en exceso de la opinión de los clientes puede hacer que los objetivos cambien drásticamente o dejar el proyecto a medias.

Referencias Bibliográficas:

Amazon Web Services, Inc. (s.f.). ¿Qué es el SDLC? - Explicación del ciclo de vida del desarrollo de software - AWS. <https://aws.amazon.com/es/what-is/sdlc/>

Pressman, R. S. (2010). Ingeniería del Software - Un enfoque práctico (7ª ed.). McGraw - Hill.

Sommerville, I. (2011). Ingeniería de software (9ª ed.). Pearson Educación de México.