



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

INGENIERÍA DE SOFTWARE I

Semana 2

SABERES PREVIOS

¿Qué es el proceso del software?

¿Cuáles son las actividades generales que están presentes en todo modelo de software?

¿Cuáles son los modelos prescriptivos del proceso y cuales son sus fortalezas y debilidades?

¿Por qué la agilidad es un modelo imperativo en la ingeniería de software moderna?

¿Qué es un desarrollo ágil de software y en que se diferencia en los modelos más tradicionales del proceso?



- Al término de la sesión, el estudiante diferenciará los diferentes modelos de procesos de desarrollo de software, comprendiendo sus ventajas y desventajas, lo que plasmará en un cuadro comparativo.



TEMARIO



- El proceso de software
- Modelos de proceso prescriptivos
 - Modelo de la cascada
 - Modelos de proceso incremental
 - Modelos de proceso evolutivo
 - Modelo espiral
- Proceso Unificado



Procesos del software



Proceso de software

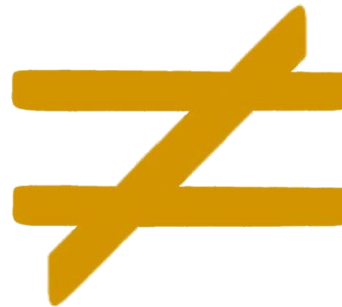
- El **desarrollo de software** es un proceso de **aprendizaje social**, donde interactúa el **usuario** y el **diseñador**. (Howard Baetjer)
- **Capital de software**, es el conjunto de conocimiento recabado, depurado y organizado a medida que se realiza el proceso.
- El **proceso de software** es una estructura para las actividades, acciones y tareas que se requiere a fin de construir un software de calidad.

Proceso de software =? Ingeniería de software

¿Cuáles son los pasos?

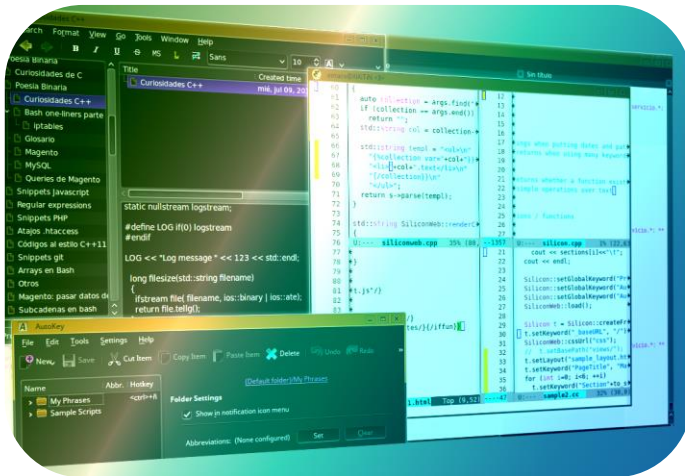
- El proceso que se adopte depende del software que se esté elaborando.

Sistema de control
electrónico de un
aeroplano



Sitio web

¿Cuál es el producto final?



Programas



Documentos



Datos

Estructura de un proceso del software



Proceso del software

Estructura del proceso

Actividades sombrilla

actividad estructural # 1

acción de ingeniería de software # 1.1

Conjuntos
de tareas
⋮

tareas del trabajo
productos del trabajo
puntos de aseguramiento de la calidad
puntos de referencia del proyecto

acción de ingeniería de software # 1.k

Conjuntos
de tareas

tareas del trabajo
productos del trabajo
puntos de aseguramiento de la calidad
puntos de referencia del proyecto

⋮

actividad estructural # n

acción de ingeniería de software # n.1

Conjuntos
de tareas
⋮

tareas del trabajo
productos del trabajo
puntos de aseguramiento de la calidad
puntos de referencia del proyecto

acción de ingeniería de software # n.m

Conjuntos
de tareas

tareas del trabajo
productos del trabajo
puntos de aseguramiento de la calidad
puntos de referencia del proyecto



Modelo general de proceso

Actividades generales/sombrilla

Seguimiento y control del proyecto,

Gestión de riesgos,

Aseguramiento de la calidad,

Gestión de la configuración,

Revisiones técnicas,

Entre otras.

Actividades estructurales

Comunicación



Planeación



Modelado



Construcción



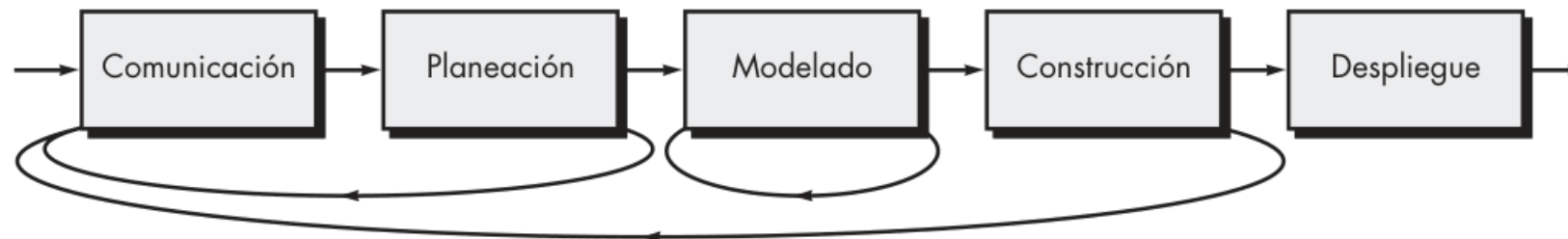
Despliegue

Flujos de procesos

El **flujo de procesos**: describe la manera en que están organizadas las **actividades estructurales** y las acciones y tareas que ocurren dentro de cada una con respecto de la secuencia y el tiempo.

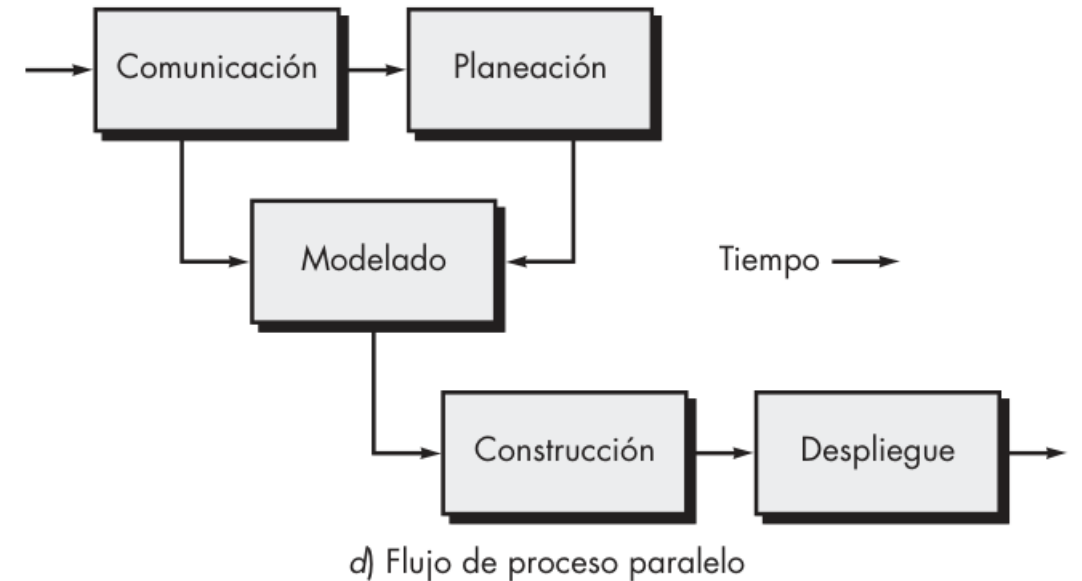
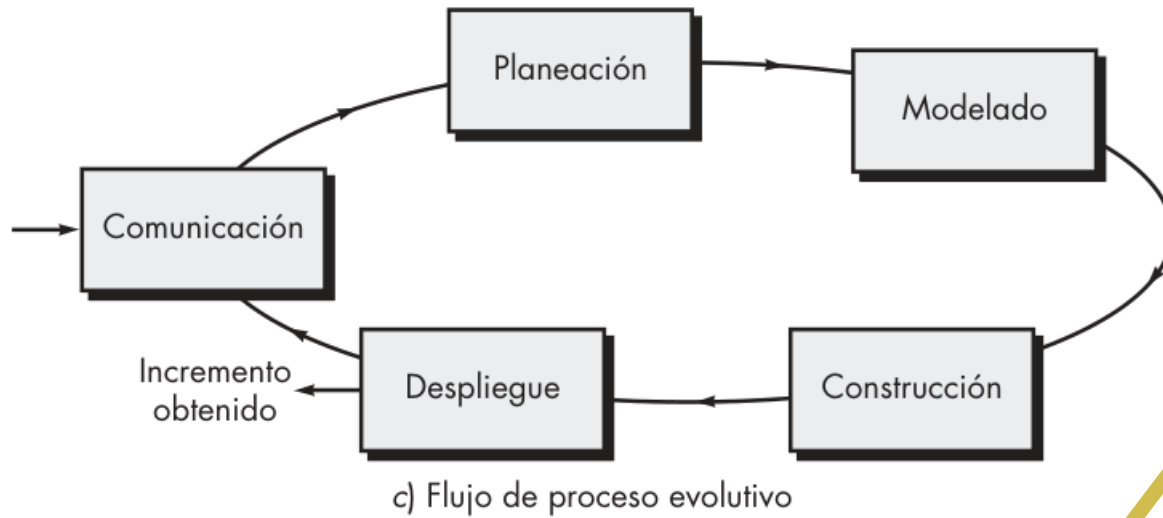


a) Flujo de proceso lineal



b) Flujo de proceso iterativo

Flujos de procesos



Definición de actividad estructural

- ¿Qué acciones son apropiadas para una actividad estructural, dados la naturaleza del problema por resolver, las características de las personas que hacen el trabajo y los participantes que patrocinan el proyecto?

Ejemplo



- Para un proyecto de **software pequeño** solicitado por **una persona** (en una **ubicación remota**) con requerimientos sencillos y directos.

ACTIVIDAD:

Comunicación

ACCIÓN:

Una conversación telefónica

TAREAS:

1. Hacer contacto con el participante por vía telefónica.
2. Analizar los requerimientos y tomar notas.
3. Organizar las notas por escrito en una formulación breve de los requerimientos.
4. Enviar correo electrónico al participante para que revise y apruebe.



Ejemplo 2:

- Si el proyecto fuera considerablemente más complejo, con muchos participantes y cada uno con un distinto conjunto de requerimientos (a veces en conflicto)

ACTIVIDAD:

Comunicación

ACCIÓN:

- Concepción,
- Indagación,
- Elaboración,
- Negociación,
- Especificación
- Validación.

tareas de trabajo y un número grande de diferentes productos finales.



Identificación de un conjunto de tareas

- Cada **acción** de la ingeniería de software se representa por cierto número de distintos conjuntos de **tareas**:
 - Productos del trabajo
 - Puntos de aseguramiento de la calidad
 - Hitos proyecto

Debe **escogerse** el conjunto de **tareas** que se adapte mejor a las:

- Las necesidades del proyecto
- Las características del equipo

Patrones del proceso

- Un patrón del proceso, identifica el **ambiente** en el que surge el **problema** y sugiere una o más **soluciones** para el mismo.
- Al combinar patrones, un equipo de software resuelve problemas y construye el proceso que mejor satisfaga las necesidades de un proyecto.

Ejemplo de patrón de proceso



Ejemplo de patrón de proceso

El siguiente patrón de proceso abreviado describe un enfoque aplicable en el caso en el que los participantes tienen una idea general de lo que debe hacerse, pero no están seguros de los requerimientos específicos de software.

Nombre del patrón. Requerimientos Poco Claros

Intención. Este patrón describe un enfoque para construir un modelo (un prototipo) que los participantes pueden evaluar en forma iterativa, en un esfuerzo por identificar o solidificar los requerimientos de software.

Tipo. Patrón de fase.

Contexto inicial. Antes de iniciar este patrón deben cumplirse las siguientes condiciones: 1) se ha identificado a los participantes; 2) se ha establecido un modo de comunicación entre los participantes y el equipo de software; 3) los participantes han identificado el problema general de software que se va a resolver; 4) se ha obtenido el entendimiento inicial del alcance del proyecto, los requerimientos básicos del negocio y las restricciones del proyecto.

Problema. Los requerimientos son confusos o inexistentes, pero hay un reconocimiento claro de que existe un problema por resolver y que

debe hacerse con una solución de software. Los participantes no están seguros de lo que quieren, es decir, no pueden describir con detalle los requerimientos del software.

Solución. Aquí se presentaría una descripción del proceso prototipo, que se describirá más adelante, en la sección 2.3.3.

Contexto resultante. Los participantes aprueban un prototipo de software que identifica los requerimientos básicos (por ejemplo, modos de interacción, características computacionales, funciones de procesamiento). Después de esto, 1) el prototipo quizá evolucione a través de una serie de incrementos para convertirse en el software de producción, o 2) tal vez se descarte el prototipo y el software de producción se elabore con el empleo de otro proceso de patrón.

Patrones relacionados. Los patrones siguientes están relacionados con este patrón: **Comunicación Con Clientes, Diseño Iterativo, Desarrollo Iterativo, Evaluación Del Cliente, Obtención De Requerimientos.**

Usos y ejemplos conocidos. Cuando los requerimientos sean inciertos, es recomendable hacer prototipos.

INFORMACIÓN

Tareas para indagar REQUERIMIENTOS

... En un proyecto pequeño

1. Elaborar la lista de participantes del proyecto.
2. Invitar a todos los participantes a una reunión informal.
3. Pedir a cada participante que haga una relación de las características y funciones que requiere.
4. Analizar los requerimientos y construir la lista definitiva.
5. Ordenar los requerimientos según su prioridad.
6. Identificar las áreas de incertidumbre.

... en un proyecto más grande y complejo

1. Hacer la lista de participantes del proyecto.
2. Entrevistar a cada participante por separado a fin de determinar los deseos y necesidades generales.
3. Formar la lista preliminar de las funciones y características con base en las aportaciones del participante.
4. Programar una serie de reuniones para facilitar la elaboración de las especificaciones de la aplicación.
5. Celebrar las reuniones.
6. Producir en cada reunión escenarios informales de usuario.
7. Afinar los escenarios del usuario con base en la retroalimentación de los participantes.
8. Formar una lista revisada de los requerimientos de los participantes.
9. Usar técnicas de despliegue de la función de calidad para asignar prioridades a los requerimientos.
10. Agrupar los requerimientos de modo que puedan entregarse en forma paulatina y creciente.
11. Resaltar las limitantes y restricciones que se introducirán al sistema.
12. Analizar métodos para validar el sistema.

(Patrones de diseño)

- <https://refactoring.guru/es/design-patterns>



Video disponible en: <https://youtu.be/6BHOeDL8vls?si=RRPcqskfsaHxiOc2>

Evaluación y mejoras del proceso

- La existencia de un proceso del software **no es garantía** de que el software se **entregue a tiempo**, que **satisfaga las necesidades** de los consumidores o que tenga las **características técnicas** que conducirán a características de **calidad de largo plazo**.
- **El proceso en sí** puede **evaluarse** para garantizar que cumple con ciertos criterios de proceso básicos que se haya demostrado que son esenciales para el éxito de la ingeniería de software

Enfoques para la evaluación y mejora de un proceso del software



Método de evaluación del estándar CMMI para el proceso de mejora (SCAMPI)

Evaluación basada en CMM para la mejora del proceso interno (CBA IPI)

SPICE (ISO/IEC 15504)

ISO9001:2000 para software

http://fcaenlinea.unam.mx/anexos/1728/Unidad_2/u2_act2_1.pdf



Modelos de proceso prescriptivo

- Los modelos de procesos prescriptivos luchan por lograr estructura y orden en el desarrollo de software.



¿Son los modelos prescriptivos apropiados para un mundo del software que prospera con el cambio?

- Los llamamos “prescriptivos” porque **fijan** un conjunto de elementos del proceso.
- Los modelos de procesos prescriptivos a veces se denominan modelos de procesos “tradicionales”

Modelos de proceso prescriptivo

El modelo de cascada

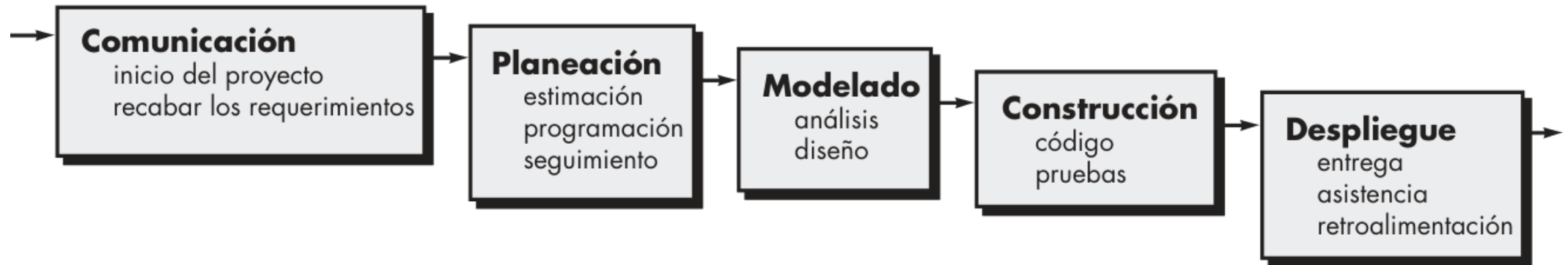
Modelo de proceso de creación de prototipos

Modelo de proceso evolutivo

Modelo de proceso unificado

Modelo de la cascada

El modelo en cascada, a veces llamado **modelo secuencial lineal**, sugiere un enfoque sistemático y secuencial para el desarrollo de software que comienza con el cliente.



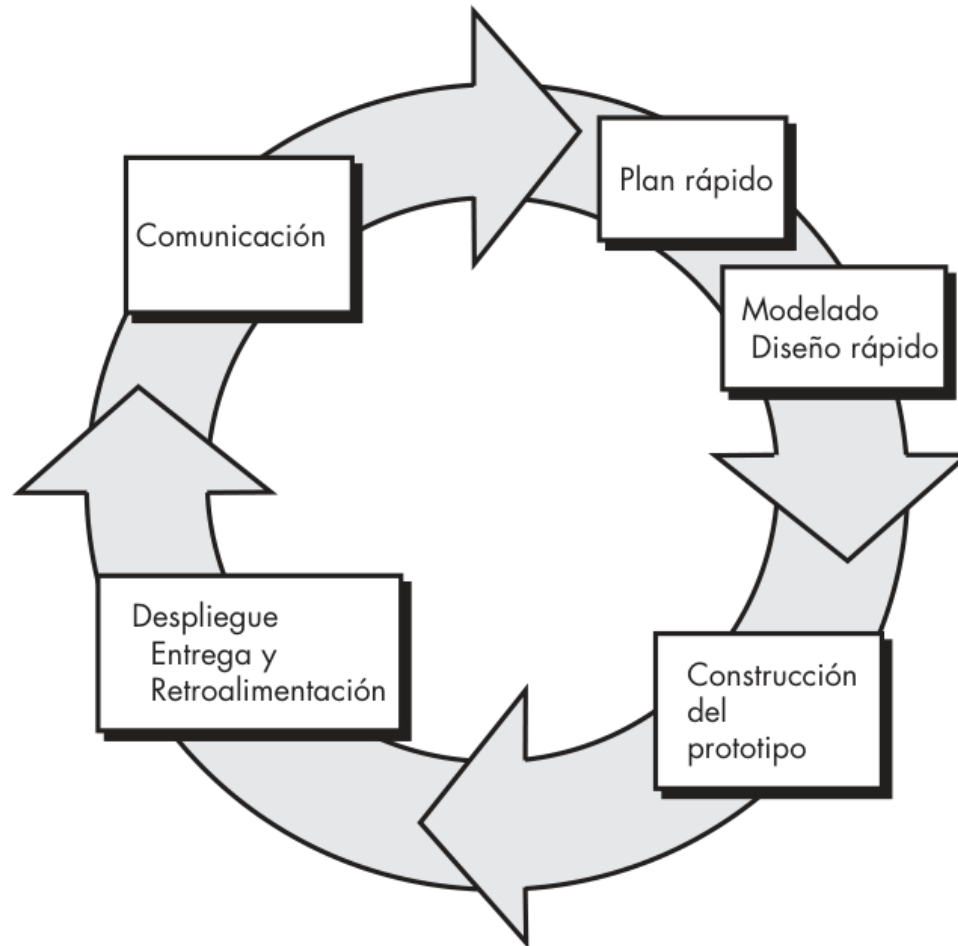
El modelo en cascada **es el paradigma más antiguo** de la ingeniería de software.

Problemas del modelo de la cascada

- Los proyectos reales **rara vez siguen el flujo de trabajo secuencial** que propone el modelo.
- A menudo resulta **difícil** para el cliente **indicar todos los requisitos** explícitamente en el comienzo de la mayoría de los proyectos
- El **cliente debe tener paciencia** porque una versión funcional del programa no estará disponible hasta finales del proyecto.
- Los **errores graves no podrán detectarse** hasta que se revise el programa de trabajo

Modelo de proceso de creación de prototipos

El paradigma de **creación de prototipos** le ayuda a usted y a otras partes interesadas a **comprender mejor lo que se debe construir** cuando los requisitos son confusos.



Modelo de proceso de creación de prototipos

- El paradigma de creación de prototipos le **ayuda** a usted y a **otras partes** interesadas a comprender mejor lo que se debe construir cuando los requisitos son confusos.
- El prototipo es implementado y evaluado por las partes interesadas, quienes brindan comentarios que se utilizan para perfeccionar aún más los requisitos.
- La iteración se produce a medida que el prototipo se ajusta para satisfacer las necesidades de varias partes interesadas y, al mismo tiempo, le permite comprender mejor lo que hay que hacer.

- Si se va a construir un prototipo funcional, puede utilizar fragmentos de programas existentes o aplicar herramientas que permitan generar programas funcionales rápidamente.
- Los usuarios tienen una idea del sistema real y los desarrolladores pueden crear algo de inmediato.

El modelo evolutivo.

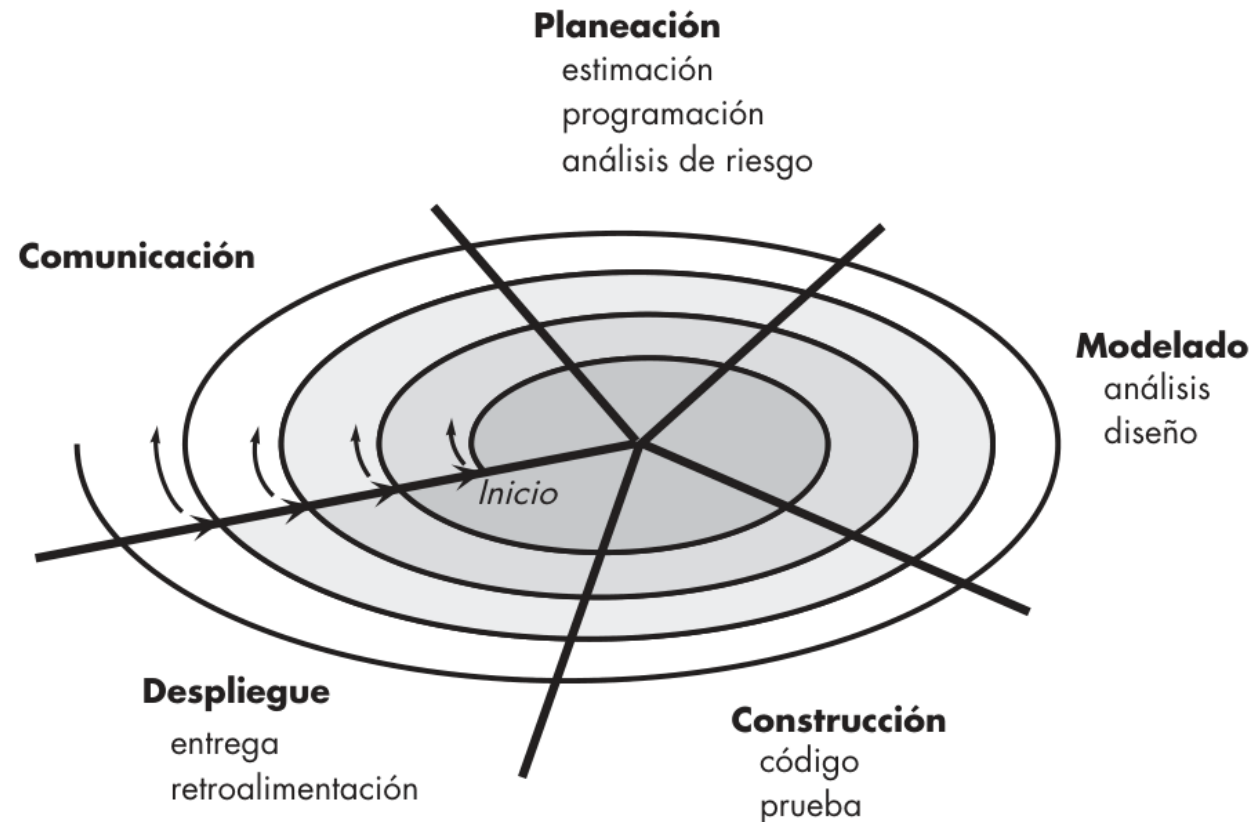


- Crear una versión limitada de un producto para satisfacer la presión competitiva o empresarial y lanzar una versión refinada una vez que se comprendan mejor todas las características del sistema



El modelo evolutivo → Modelo espiral

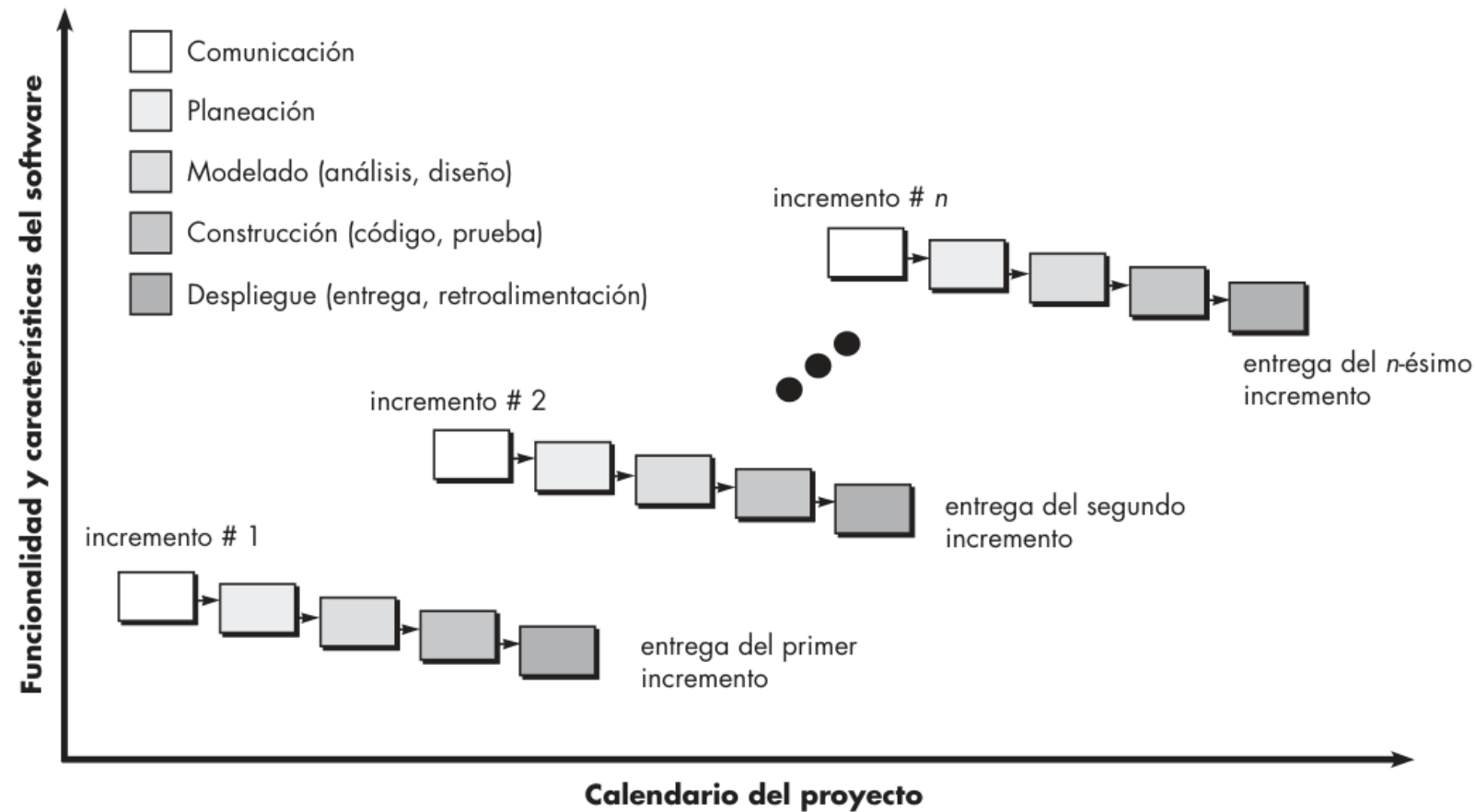
- el modelo espiral es un modelo evolutivo del proceso del software



El modelo espiral.

- Propuesto originalmente por Barry Boehm [Boe88], el modelo en espiral es un modelo de proceso de software evolutivo que **combina** la naturaleza iterativa de la creación de **prototipos** con los aspectos controlados y sistemáticos del modelo en **cascada**.
- La intención de los modelos evolutivos es desarrollar software de alta calidad de manera iterativa o incremental.

Modelos de proceso incremental



Modelos de proceso incremental

- El modelo incremental aplica **secuencias lineales en forma escalonada** a medida que avanza el calendario de actividades.

Ejemplo:

Un software
para
procesar
textos

El **primer** incremento las funciones básicas de administración de archivos, edición y producción del documento;

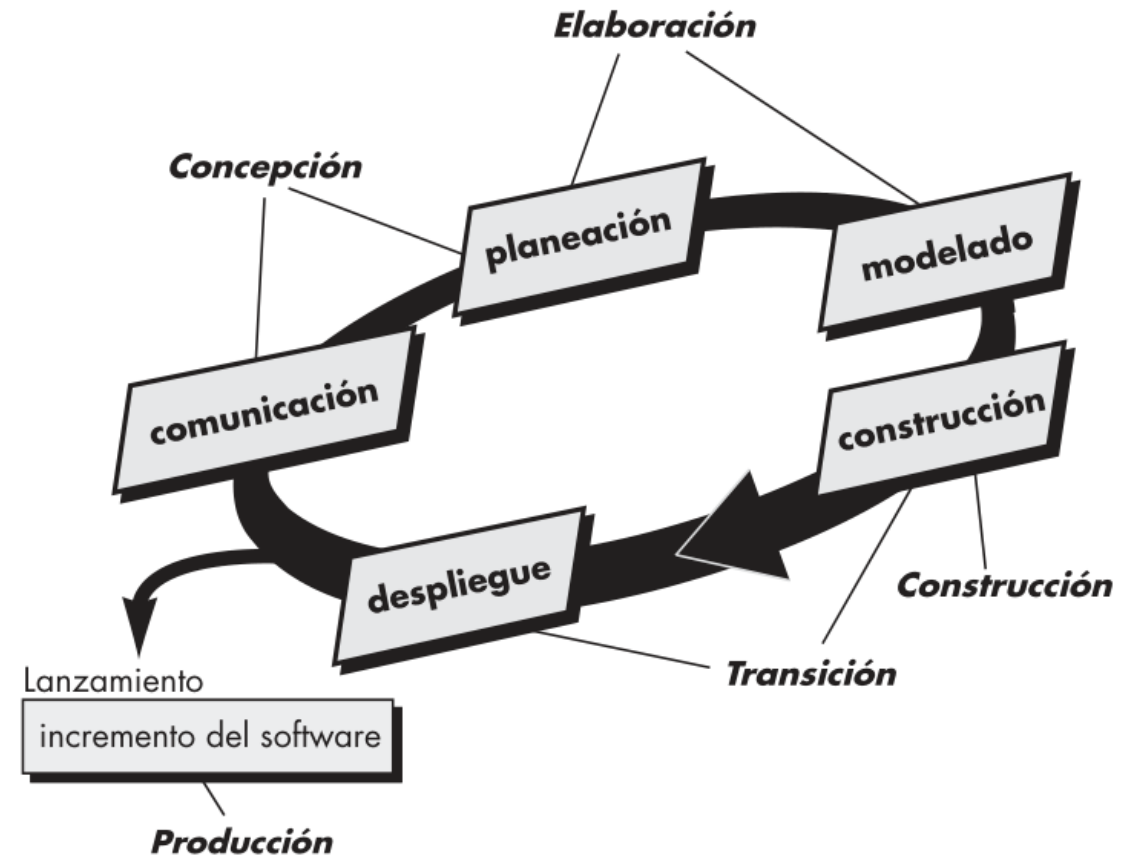
El **segundo** dará herramientas más sofisticadas de edición y producción de documentos;

El **tercero** habrá separación de palabras y revisión de la ortografía;

El **cuarto** se proporcionará la capacidad para dar formato avanzado a las páginas.

El proceso unificado

Es un intento de aprovechar las mejores características de los modelos de proceso de software tradicionales, pero caracterizarlos de una manera que implemente muchos de los mejores principios del desarrollo de software ágil.



Fase inicial

- La **fase inicial** del PROCESO UNIFICADO es donde tiene lugar la **comunicación** y la **planificación** con el cliente.
- Los requisitos empresariales fundamentales se describen a través de un conjunto de **casos de uso** preliminares que describen qué características y funciones desea cada clase principal de usuarios que se realizarán en la arquitectura del software. La planificación identifica recursos, evalúa los riesgos principales y define un cronograma preliminar para los incrementos de software

Fase de elaboración

- La fase de **elaboración** abarca las actividades de **planificación** y **modelado** del modelo de proceso genérico.
- La elaboración refina y amplía los casos de uso preliminares y crea una línea base arquitectónica que incluye cinco vistas diferentes del software: el modelo de caso de uso, el modelo de análisis, el modelo de diseño, el modelo de implementación y el modelo de implementación

Fase de construcción

- La fase de construcción de la UP es idéntica a la actividad de **construcción** definida para el proceso de software genérico.
- Todas las características y funciones necesarias y requeridas para el incremento del software (es decir, el lanzamiento) se implementan en el código fuente. A medida que se implementan los componentes, se diseñan y ejecutan pruebas unitarias para cada uno. Además, se llevan a cabo actividades de integración (montaje de componentes y pruebas de integración). Los casos de uso se utilizan para derivar un conjunto de pruebas de aceptación que se ejecutan antes del inicio de la siguiente fase UP.

Fase de transición

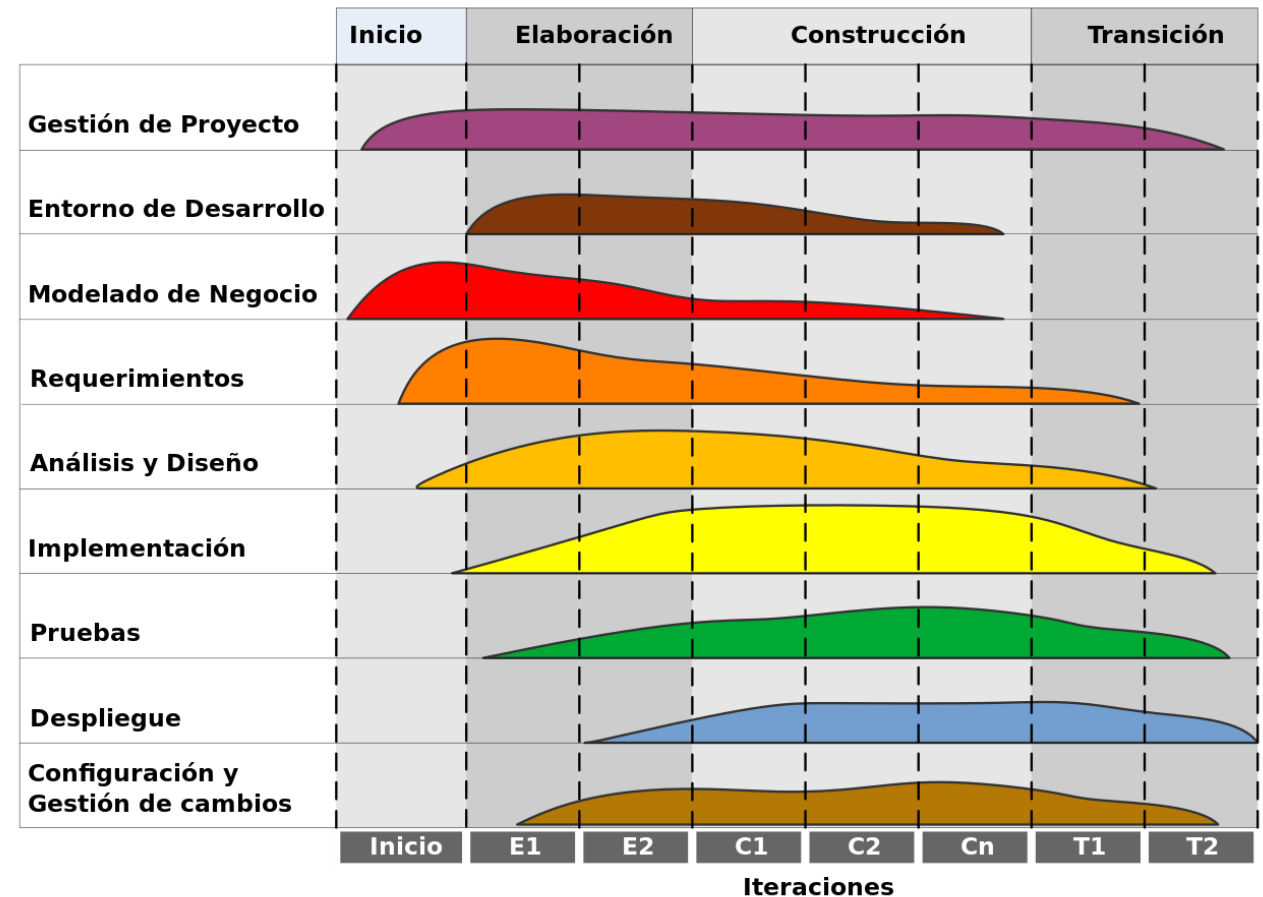
- La fase de transición del UP abarca las últimas etapas de la actividad genérica de construcción y la primera parte de la actividad genérica de **despliegue** (entrega y retroalimentación).
- El software y la documentación de respaldo se entregan a los usuarios finales para realizar pruebas beta, y los comentarios de los usuarios informan tanto de los defectos como de los cambios necesarios.
- Al finalizar la fase de transición, el incremento de software se convierte en una versión de software utilizable.

Fase de producción

- La fase de producción de la UP coincide con la actividad de **despliegue** del proceso genérico.
- Durante esta fase, se monitorea el uso continuo del software, se brinda soporte para el entorno operativo (infraestructura) y se envían y evalúan informes de defectos y solicitudes de cambios.

RUP

- Proceso Unificado de Rational o RUP es una marca registrada por IBM (desde su compra de Rational Software Corporation en 2003).



Producto y proceso

- La realidad es que **ningún proceso es perfecto** para cada proyecto. Por lo general, el equipo de software adapta uno o más de los modelos de procesos ya vistos o los modelos de procesos ágiles para satisfacer las necesidades del proyecto en cuestión.



Producto y proceso

- La realidad es que **ningún proceso es perfecto** para cada proyecto. Por lo general, el equipo de software adapta uno o más de los modelos de procesos ya vistos o los modelos de procesos ágiles para satisfacer las necesidades del proyecto en cuestión.
- Si el proceso es débil, el producto final sin duda se verá afectado. Pero una dependencia excesiva y obsesiva del proceso también es peligrosa.

Resumen



- Un modelo de proceso genérico para la ingeniería de software abarca un conjunto de actividades, acciones y tareas de trabajo marco y generales.
- Los patrones de proceso se pueden utilizar para resolver problemas comunes que se encuentran como parte del proceso de software.



- Los modelos de procesos prescriptivos son un esfuerzo por poner orden y estructura en el desarrollo de software.
- Todos los modelos realizan el mismo conjunto de actividades marco genéricas: **comunicación, planificación, modelado, construcción e implementación.**

- Los modelos de **procesos secuenciales**, como el modelo en cascada, son los paradigmas más antiguos de la ingeniería de software. Sugieren un flujo de proceso lineal que a menudo es inconsistente con las realidades modernas (por ejemplo, cambios continuos, sistemas en evolución, plazos ajustados) en el mundo del software. Sin embargo, tienen aplicabilidad en situaciones donde los requisitos están bien definidos y son estables.

- Los **modelos de procesos incrementales** son de naturaleza iterativa y producen versiones funcionales de software con bastante rapidez.
- Los **modelos de procesos evolutivos** reconocen la naturaleza iterativa e incremental de la mayoría de los proyectos de ingeniería de software y están diseñados para adaptarse al cambio.
- Los modelos evolutivos, como la creación de prototipos y el modelo en espiral, producen productos de trabajo incrementales (o versiones funcionales del software) rápidamente. Estos modelos se pueden adoptar para aplicarlos en todas las actividades de ingeniería de software, desde el desarrollo de conceptos hasta el mantenimiento del sistema a largo plazo.

- El Proceso Unificado es un “caso centrado en la arquitectura, iterativo e incremental”, es un proceso de software diseñado como marco para métodos y herramientas UML

Actividad

- Analice las ventajas y desventajas entre los diversos flujos de proceso y plasmarlo en un cuadro comparativo.

	Ventajas	Desventajas
El modelo de cascada		
Modelo de proceso de creación de prototipos		
Modelo de procesos incremental		
Modelo de proceso evolutivo		
Modelo de proceso unificado		

¿Preguntas o dudas?

- ¿Qué me llevo de la clase?



Cuestionario



GRACIAS

Ingeniería
de Sistemas
Universidad Nacional de Cajamarca

