

Trabajo de Campo I



UNIDAD 1

Procesos para la creación de Software

SOFTWARE COMO PRODUCTO

- ¿Que es?
- ¿Quién lo hace?
- ¿Por qué es importante?
- ¿Cuáles son los pasos para obtenerlo?
- ¿Qué se obtiene como producto final?
 - Programas, documentos y datos que configuran el SW
- Garantías de corrección.

SOFTWARE COMO PRODUCTO

¿Qué es?

- ✧ Es el producto del trabajo de los programadores.
- ✧ Se sustenta en otros productos generados por los Ingenieros en sistemas, los arquitectos de software, los diseñadores y los analistas.
- ✧ Como producto podemos afirmar que luego de su puesta en marcha necesita que se lo mantenga.

SOFTWARE COMO PRODUCTO

¿Quién lo hace?

- ✧ Ingenieros de Software / Ingenieros en Sistemas
- ✧ Arquitectos de Software
- ✧ Diseñadores
- ✧ Analistas
- ✧ Programadores

SOFTWARE COMO PRODUCTO

¿Por qué es importante?

- ✧ Producto transversal a casi todas las actividades cotidianas tanto en el ámbito personal como organizacional.
- ✧ La IS permite que cuando estos productos son complejos se puedan desarrollar con calidad.

SOFTWARE COMO PRODUCTO

¿Cuáles son los pasos para obtenerlo?

- ✧ Los pasos están definidos por las metodologías
- ✧ Existen métodos variados:
 - ✧ Tradicionales
 - ✧ Iterativos y evolutivos
 - ✧ Ágiles

SOFTWARE COMO PRODUCTO

¿Cuál es el producto final?
Depende de la visión

- Programas y Datos
- Organizados
Constituyen
“EL SISTEMA”

Programador

- La Información
- El valor agregado a
los procesos y
negocios de la
organización

Usuario

SOFTWARE COMO PRODUCTO

BUENO + BARATO = LENTO

BUENO + RAPIDO = CARO

RAPIDO + BARATO = MALO

SOFTWARE COMO PRODUCTO

Definición

1. Instrucciones que cuando se ejecutan proporcionan las características, funciones y desempeño buscado.
2. Estructuras de datos que permiten que los programadores manipulen en forma adecuada la información.

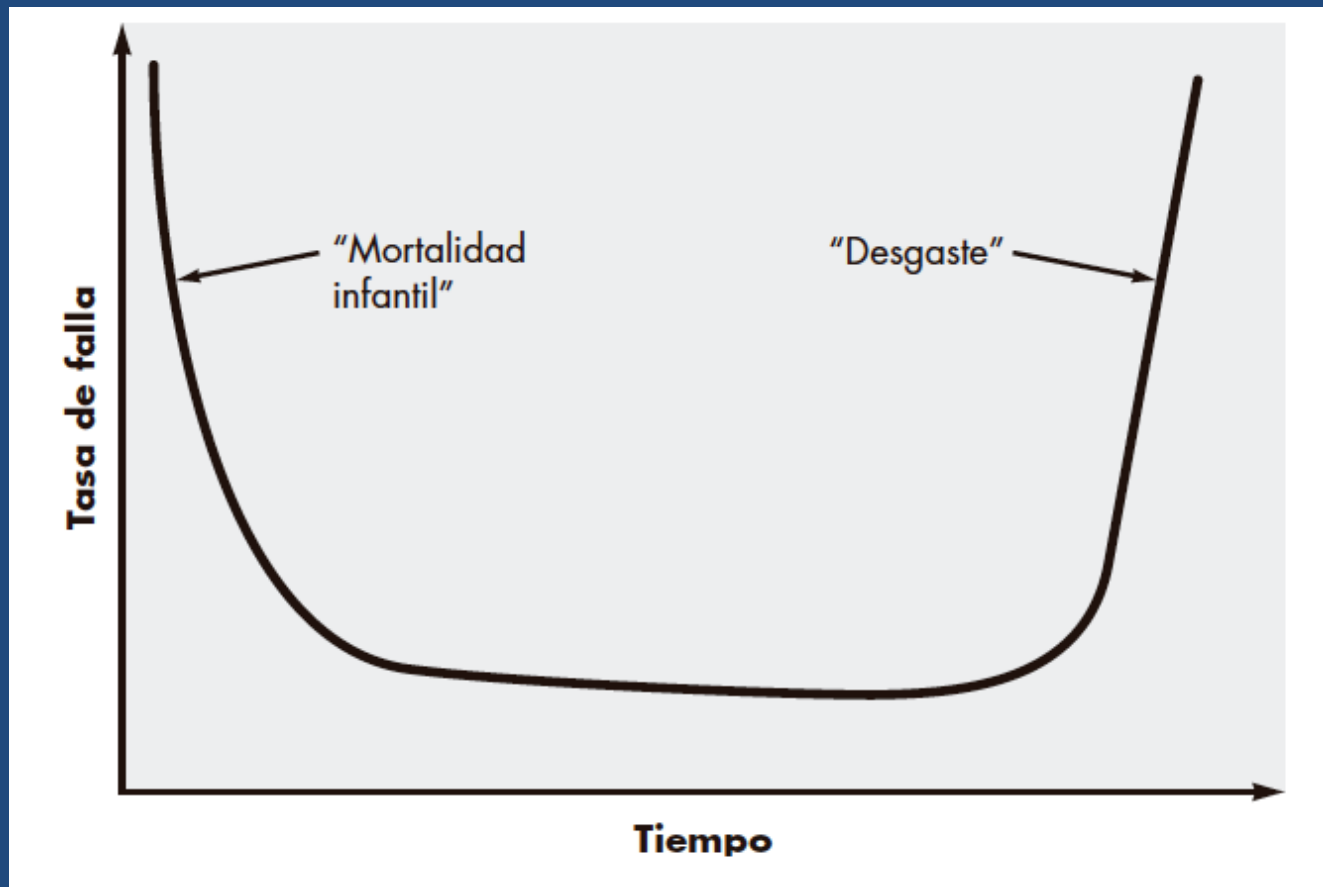
...

SOFTWARE COMO PRODUCTO

- Características
 - No es físico
 - El software se desarrolla o modifica haciendo uso del intelecto; no se manufactura en el sentido clásico.
 - No se Desgasta. (Cuando es el fin del ciclo de vida?)
- Evolución
 - Crisis del Software (1968)
 - Mejoras en el rendimiento del HW

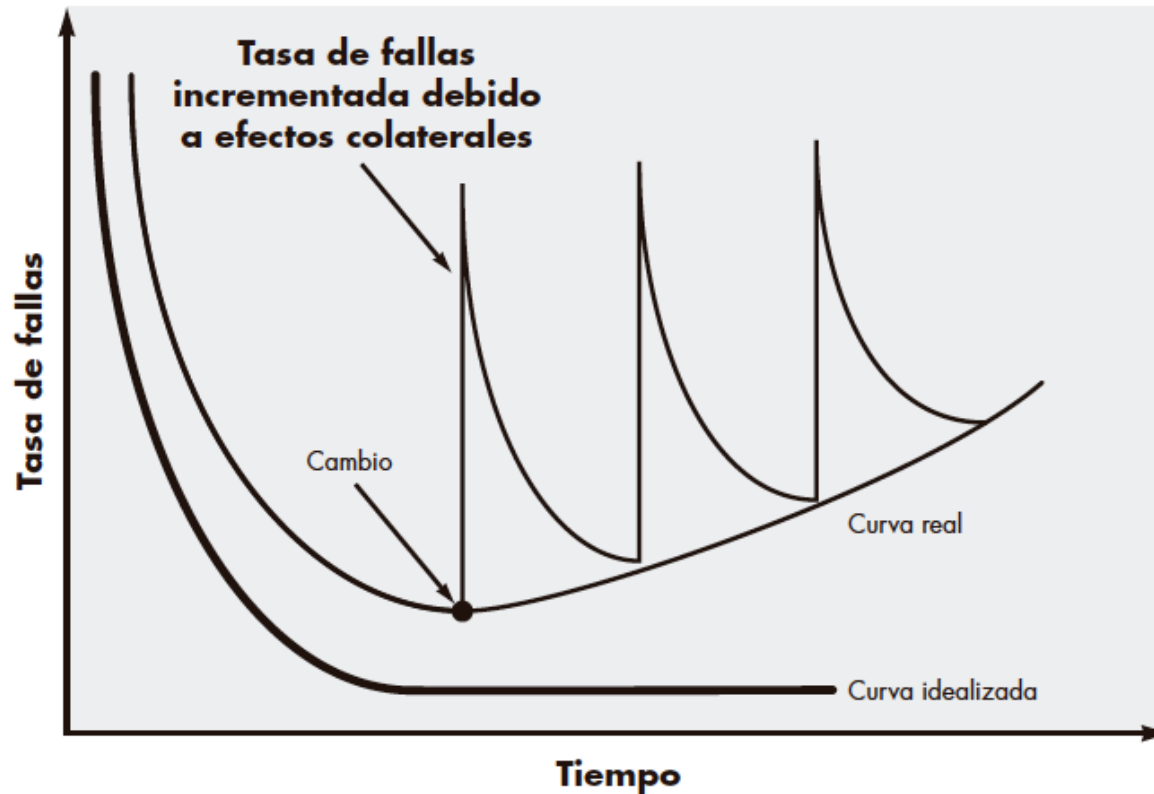
SOFTWARE COMO PRODUCTO

El software no se desgasta; sufre y se vuelve obsoleto



SOFTWARE COMO PRODUCTO

Impacto de las fallas por errores en la IS



SOFTWARE COMO PRODUCTO

El software debe:

- Adaptarse a las necesidades de los nuevos ambientes de cómputo y de la tecnología
- Ser mejorado para implementar nuevos requerimientos del negocio
- Ampliarse para operar con otros sistemas
- Rediseñarse de acuerdo a las pautas de las nuevas arquitecturas

PRODUCTO – Aplicaciones del software

- Software de Sistemas (para servir a otros programas)
- Software de Tiempo-Real
- Software de Gestión (mayor área de aplicación)
- Software de Ingeniería y científico (manejo de números)
- Software empotrado o embebido (smart, microcontrol)
- Software de computadoras personales (ofimática, etc)
- Software basado en Web
- Software de Inteligencia Artificial (AI)

PRODUCTO

Ciclo de vida

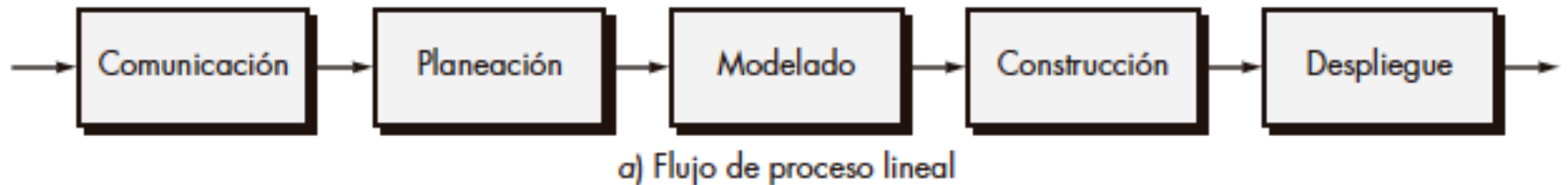
- Elemento clave en la evolución de los sistemas y productos informáticos. Siendo factor limitante en su evolución.
- Analisis: no se vence ni se estropea, pero se deterior [Curva-
>Tiempo = F(falla)]
- Reconvertir Vs. Mantener

¿Qué es exactamente un proceso del software?

Se define proceso del software como una estructura para las actividades, acciones y tareas que se requieren a fin de construir software de alta calidad.

SOFTWARE - EL PROCESO

Flujo del proceso



Un flujo de proceso lineal ejecuta cada una de las cinco actividades estructurales en secuencia, comenzando por la comunicación y terminando con el despliegue.

SOFTWARE - EL PROCESO

PATRONES DE PROCESO

Un patrón del proceso describe un problema relacionado con el proceso que se encuentra durante el trabajo de ingeniería de software, identifica el ambiente en el que surge el problema y sugiere una o más soluciones para el mismo.

REUTILIZACIÓN DE LA EXPERIENCIA ADQUIRIDA

Ingeniería de Software: EL PROCESO

- ✧ Se debe hacer un esfuerzo por comprender el problema antes de desarrollar una aplicación de software.
- ✧ El diseño se ha vuelto una actividad crucial.
- ✧ La calidad del software impacta directamente en el producto o servicio que se brinda.
- ✧ El software actual debe ser fácil de mantener.

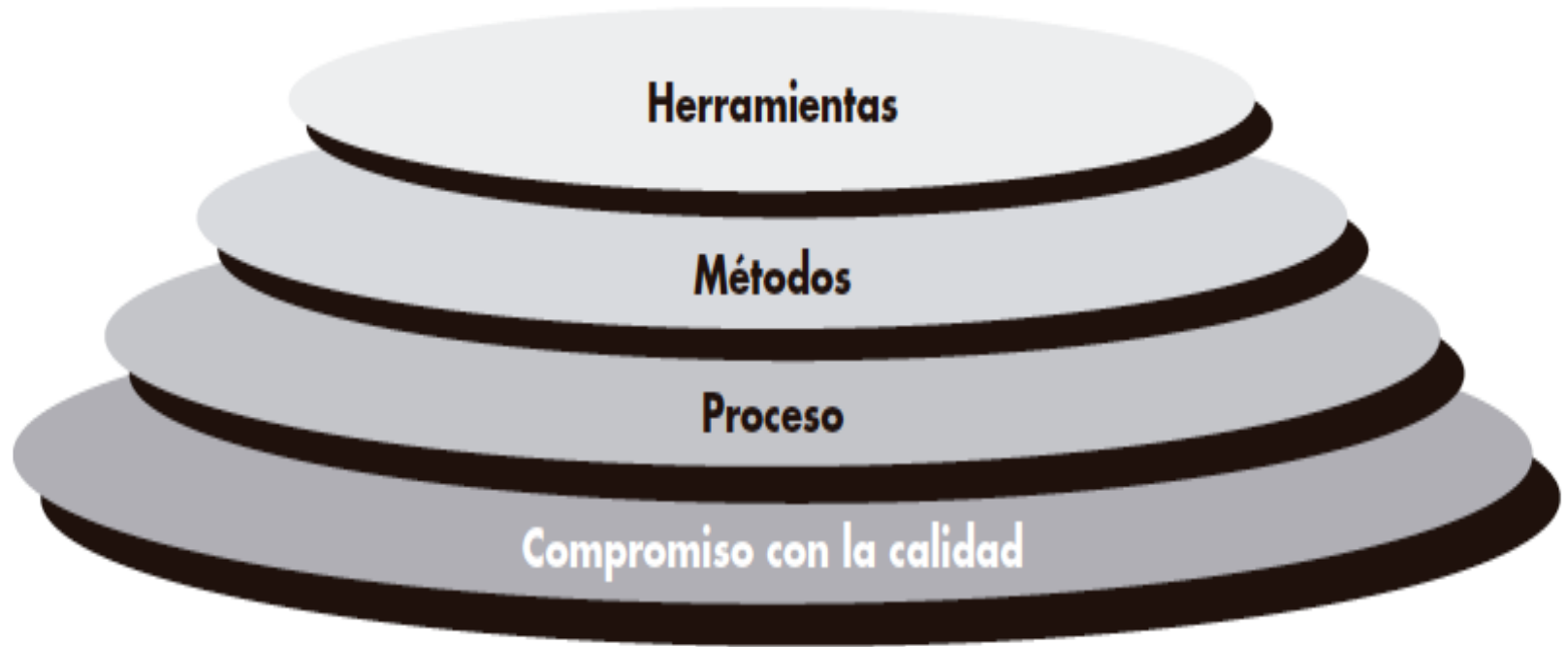
Ingeniería de Software

- Un proceso de software es el marco de trabajo de las tareas que se requieren para construir software de calidad.
- Un proceso de software define el enfoque que se toma cuando el software es tratado por la ingeniería.

Ingeniería del software

- Establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales.
- Según el IEEE:
 - La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software
 - El estudio de enfoques.

Ingeniería de Software



PROCESO

Proceso, métodos y herramientas

- Proceso: define el marco de trabajo para un conjunto de áreas claves de proceso (ACPs).
- Métodos: indican “como” construir técnicamente el software (Ej.: análisis de requisitos, diseño, construcción, pruebas, mantenimiento).
- Herramientas: proporcionan un enfoque automático o semi-automático para el proceso y los métodos.

PROCESO

Fases del proceso

- Definición: se centra en el “Qué”. Permite identificar los requisitos claves del sistema y del software.
- Desarrollo: se centra en el “Cómo”. Definición de estructuras de datos, arquitectura y detalles procedimentales. Se debe elegir un lenguaje.
- Mantenimiento: se centra en el “Cambio” que va asociado a la corrección de errores y adaptaciones requeridas.

PROCESO – CMMI

Modelo de madurez de la Capacidad Integrado

- El CMMI es un enfoque de mejora de procesos que provee a las organizaciones de los elementos esenciales para un proceso efectivo.
- El CMMI es el Modelo de Madurez de Capacidades Integrado.
- Fue desarrollado por el SEI (Software Engineering Institute).
- Mide la madurez del desarrollo del software en una escala del 1 al 5.

PROCESO – Nivel de madurez - CMMI

1. Inicial:

El proceso de software está caracterizado como *ad hoc*, y a veces caótico. Pocos procesos están definidos y el éxito depende de esfuerzos individuales y heroicos.

2. Repetible:

Se establecen procesos administrativos básicos en los proyectos para el seguimiento de costos, tiempos y funcionalidades. La disciplina necesaria en los procesos es acorde para repetir éxitos anteriores de proyectos con aplicaciones similares.

3. Definido:

El proceso de software, t está documentado, estandarizado e integrado en un proceso estándar para la organización. Todos los proyectos usan una versión aprobada y ajustada del proceso estándar de software de la organización para desarrollar y mantener software.

4. Administrado:

Se toman decisiones del proceso de software y de la calidad del producto. Tanto el proceso de software como los productos se conocen cuantitativamente y se controlan.

5. Optimizado:

Existe retroalimentación cuantitativa del proceso, lo que permite una mejora continua del proceso. Al mismo tiempo se manejan ideas y tecnologías innovadoras.

PROCESO – Modelos de procesos

- El modelado es una técnica para tratar con la complejidad inherente a los sistemas.
- El uso de modelos ayuda al ingeniero de software a visualizar el sistema a construir.
- Los modelos se pueden utilizar para validar los requerimientos o funcionalidad esperada del sistema.

PROCESO – Modelos de procesos

- Todas las etapas (status quo, definición de problemas, desarrollo técnico e integración de soluciones) coexisten simultáneamente en algún nivel de detalle.
- Los usuarios de software requieren un mantenimiento continuo.

¿Qué es un Modelo Prescriptivo?

- Conjunto de actividades, acciones, tareas, fundamentos y productos de trabajo que se requieren para desarrollar software de alta calidad.

Marco de Trabajo Genérico

- Actividades
 - Comunicación
 - Planeación
 - Modelado
 - Construcción
 - Desarrollo

¿Cómo se organizan estas Actividades?

- En un flujo de proceso:
 - Lineal
 - Incremental
 - Evolutivo

Esquema

MODELOS PRESCRIPTIVOS

LINEAL

EN CASCADA

DRA

INCREMENTAL

M. INCREMENTAL

EVOLUTIVO

PROTOTIPOS

D. ESPIRAL

D. CONCURRENTE

M. ESPECIALIZADOS
DE PROCESO

MET. FORMALES

O. A ASPECTOS

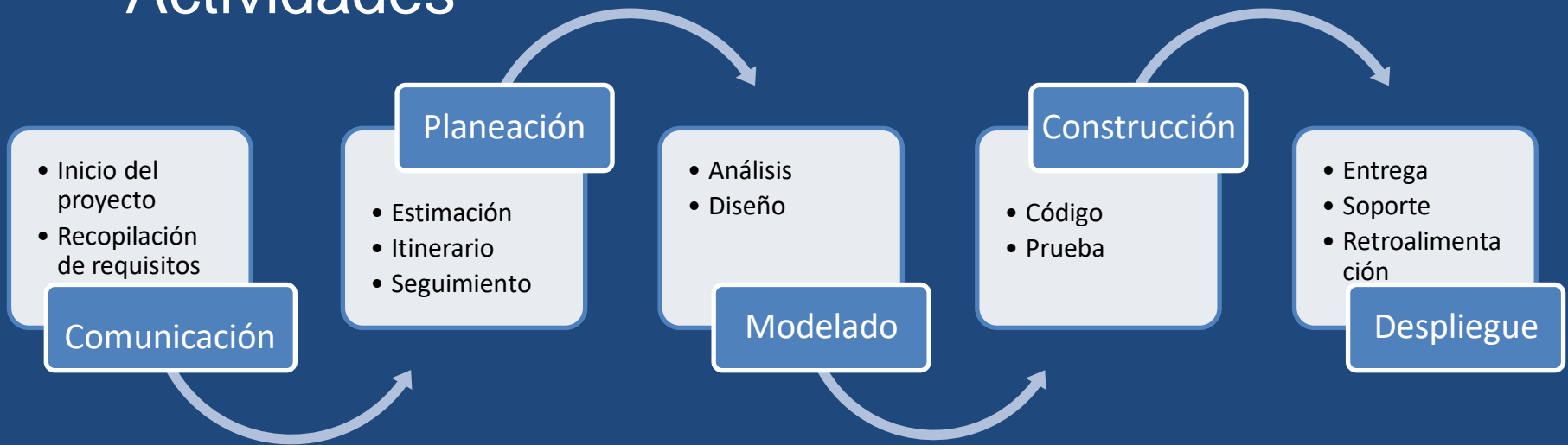
PROCESO UNIFICADO

PROCESO – Modelo Lineal/Secuencial

- También llamado “modelo en cascada” (waterfall).
- Sugiere un enfoque sistemático y secuencial para el desarrollo del software basado en la especificación del cliente.

Modelo en Cascada

- Actividades



Modelo en Cascada

- Principales Problemas
 - No es muy común que los proyectos sigan un flujo secuencial.
 - Es difícil que el Cliente establezca todos los requerimientos de manera explícita.
 - Para obtener el software se debe esperar a que el proyecto este avanzado.

PROCESO

Modelo de Prototipos

- Comienza con la recolección de requisitos, el desarrollador y el cliente definen los objetivos globales, identifican requisitos conocidos y áreas que requieren más definición.
- De esto se crea un diseño rápido, que es construido como un prototipo.
- El prototipo se evalúa y se utiliza para refinar los requerimientos.
- Algunos problemas del modelo:
 - Suele confundirse el prototipo con una versión final.

Construcción de Prototipos

- Actividades



Construcción de Prototipos

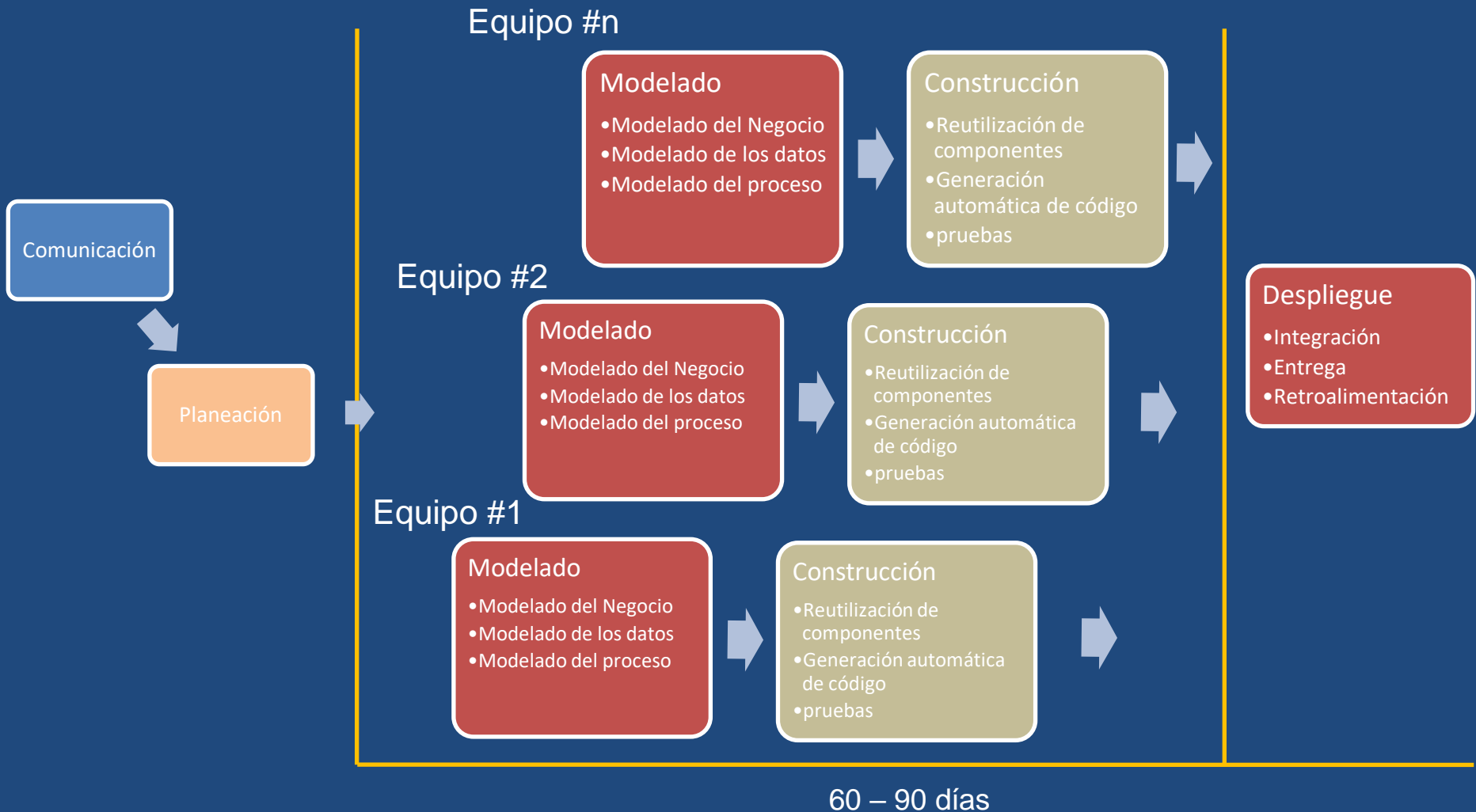
- El diseño rápido se centra en una representación de los aspectos del software visibles por el usuario, interfaz con el usuario y formatos de salida.
- Por la rapidez de sus procesos no se consideran los aspectos de calidad del software, el prototipo debe desecharse y construir una revisión rediseñada.

PROCESO – Modelo DRA

- El modelo de Desarrollo Rápido de Aplicaciones es lineal y secuencial y enfatiza los ciclos de desarrollo extremadamente corto, utilizando la construcción basada en componentes (fases muy cortas)
- Fases:
 - Modelado de la gestión
 - Modelado de datos
 - Modelado del proceso
 - Generación de aplicaciones
 - Pruebas y entrega

El Modelo DRA

- Actividades



El Modelo DRA

- Características Principales
 - Gran consumo de Recursos Humanos.
 - Insume alto grado de compromiso por parte del equipo para cumplir con los tiempos breves.
 - Inapropiado cuando los riesgos técnicos son altos.

PROCESO – Modelos Evolutivos

- Los modelos evolutivos son iterativos.
- Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez mas completas del software.
- Modelos evolutivos:
 - Modelo Incremental
 - Modelo Espiral
 - Modelo Espiral Win-Win (Victoria-Victoria)
 - Modelo de Desarrollo Concurrente

Modelo Incremental

- Combina elementos del modelo lineal secuencial con la filosofía iterativa de construcción de prototipos.
- Aplica secuencias líneas de forma escalonada mientras progresa el tiempo en el calendario.
- Entrega un producto operacional al final de cada incremento.

El Modelo Incremental

- Actividades



El Modelo Incremental

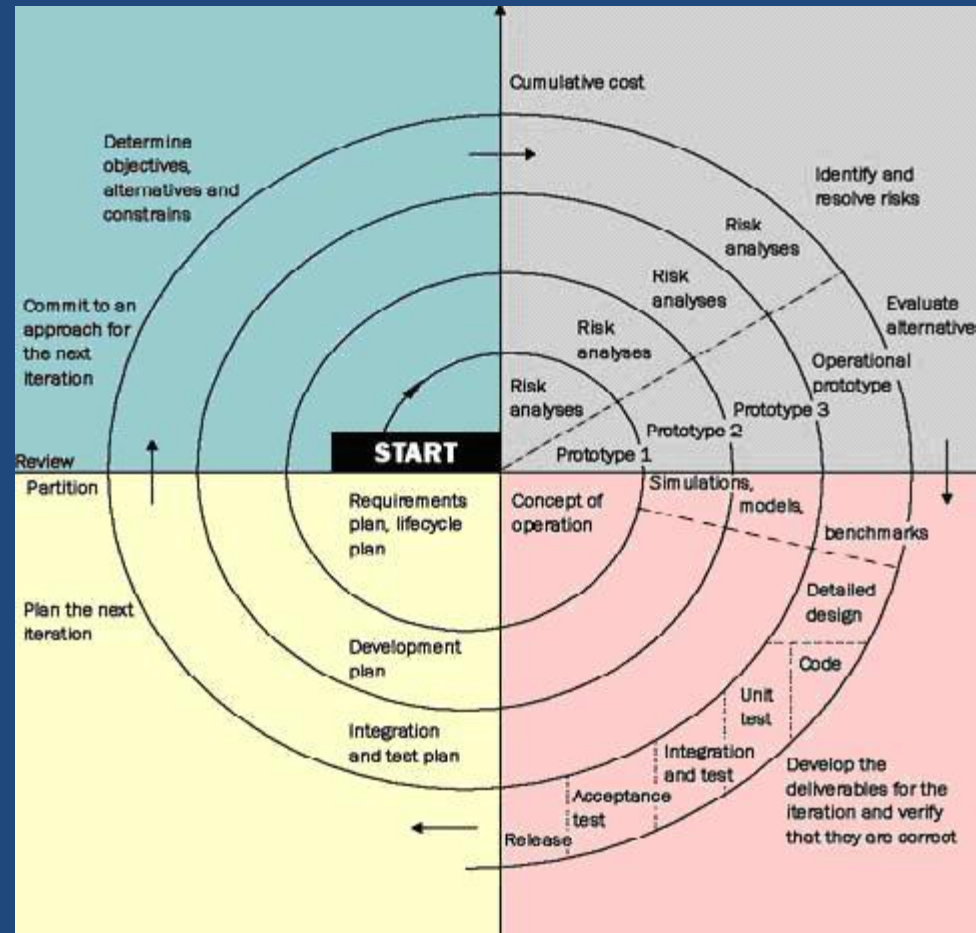
- El primer incremento es un producto esencial, se incorporan los requisitos básicos.
- El producto esencial es evaluado por el cliente.
- Los primeros incrementos son versiones “incompletas” del producto final.

Modelo Espiral

- Es incremental y evolutivo, conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial (cascada).
- El software se desarrolla en una serie de versiones incrementales.

El Modelo Espiral

- Actividades



El Modelo Espiral

- Características distintivas
 - Enfoque cíclico para el crecimiento incremental del grado de definición e implementación, mientras disminuye su grado de riesgo.
 - Conjunto de puntos de fijación para asegurar el compromiso del usuario con soluciones factibles y satisfactorias.

Modelo Espiral Win-Win

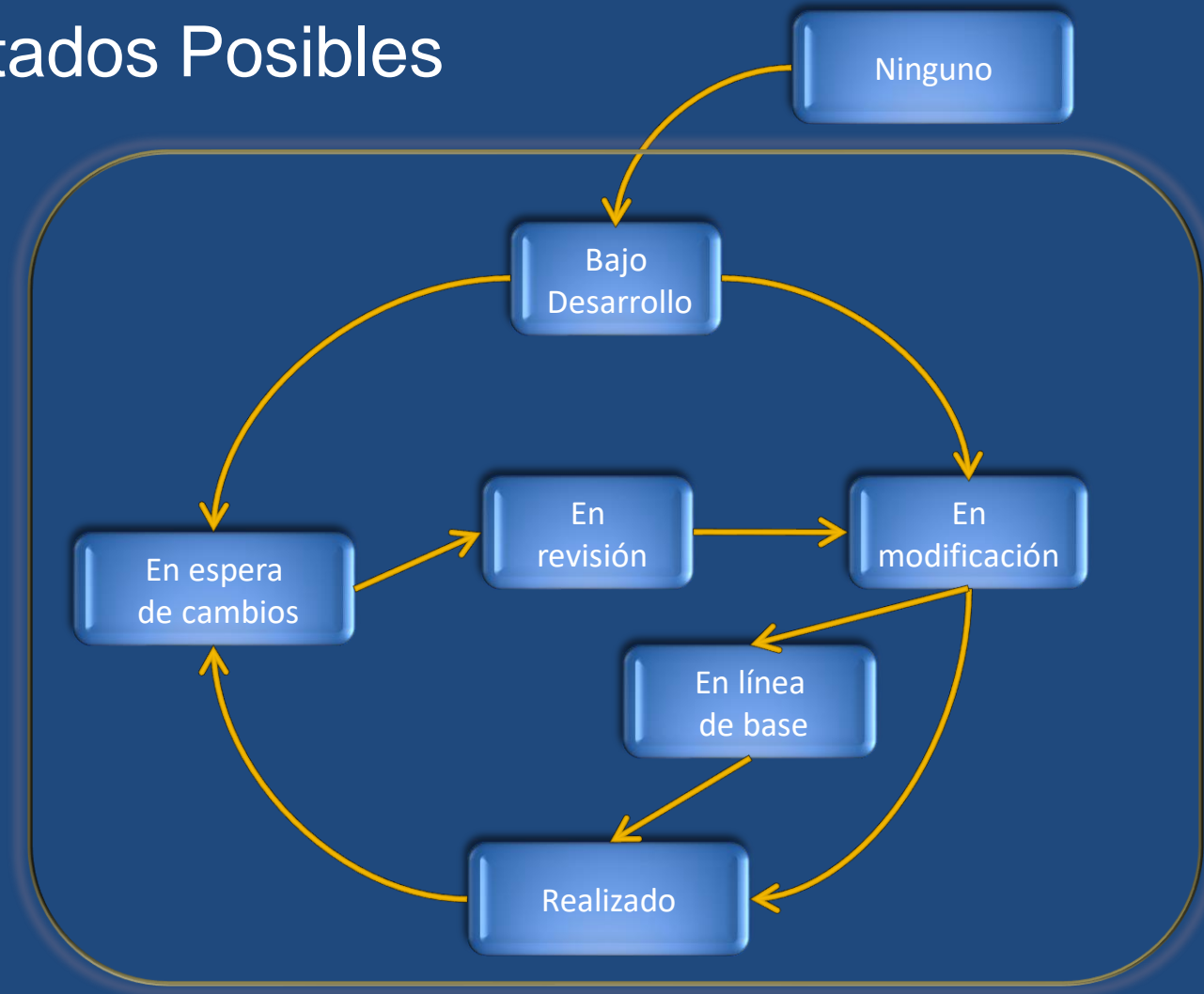
- Define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral:
 - Identificación del sistema clave de los directivos
 - Determinación de las condiciones de victoria de los directivos
 - Negociación de las condiciones de victoria de los directivos para reunirlos en un conjunto de condiciones de victoria-victoria para todos los afectados

Modelo Desarrollo Concurrente

- Se puede representar en forma de esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas.
- Todas las actividades existen concurrentemente, pero residen en estados diferentes.
- Se producen cambios de estados basados en acontecimientos en cada una de las actividades de la Ingeniería del Software.
- Se utiliza a menudo como paradigma de desarrollo en aplicaciones Cliente/Servidor

Desarrollo Concurrente

- Estados Posibles



Desarrollo Concurrente

- Se aplica a todos los tipos de desarrollo de software y proporciona una visión exacta del estado actual de un proyecto.

Modelo de Ensamblaje de Componentes

- Incorpora muchas de las características del modelo e espiral.
- La actividad de la ingeniería comienza con la identificación de las clases candidatas.
- Las clases creadas anteriormente se almacenan en una biblioteca de clases.
- Se utiliza el Lenguaje de Modelado Unificado (UML) para modelar.

Modelo de Métodos Formales

- Comprende un conjunto de actividades que conducen a la especificación matemática del software de computadora.
- Problemas del modelo:
 - Es bastante caro y lleva mucho tiempo
 - Se requiere un estudio detallado porque pocos responsables del desarrollo poseen antecedentes en la materia.
 - Es difícil utilizar los modelos como mecanismos de comunicación con clientes.

Orientado a Aspectos

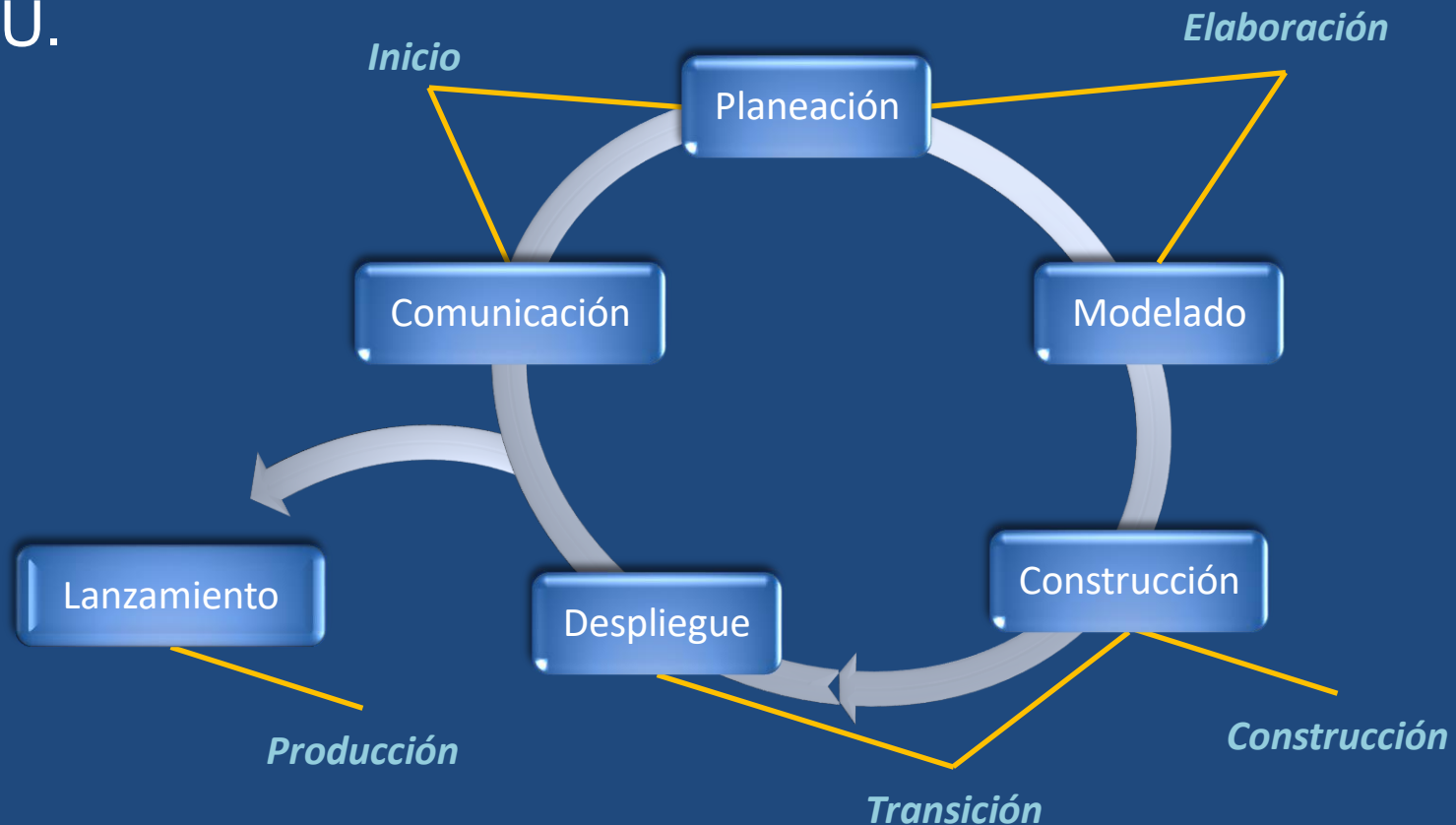
- Paradigma que proporciona un a proceso y enfoque metodológico para definir, especificar, diseñar y construir “Aspectos”.
- Un *aspecto* es una unidad modular que se dispersa por la estructura de otras unidades funcionales.

Proceso Unificado

- Marco de trabajo del proceso para la ingeniería de software orientada a objetos, mediante la utilización de UML.
- Sugiere un flujo de proceso con sentido evolutivo.

Proceso Unificado

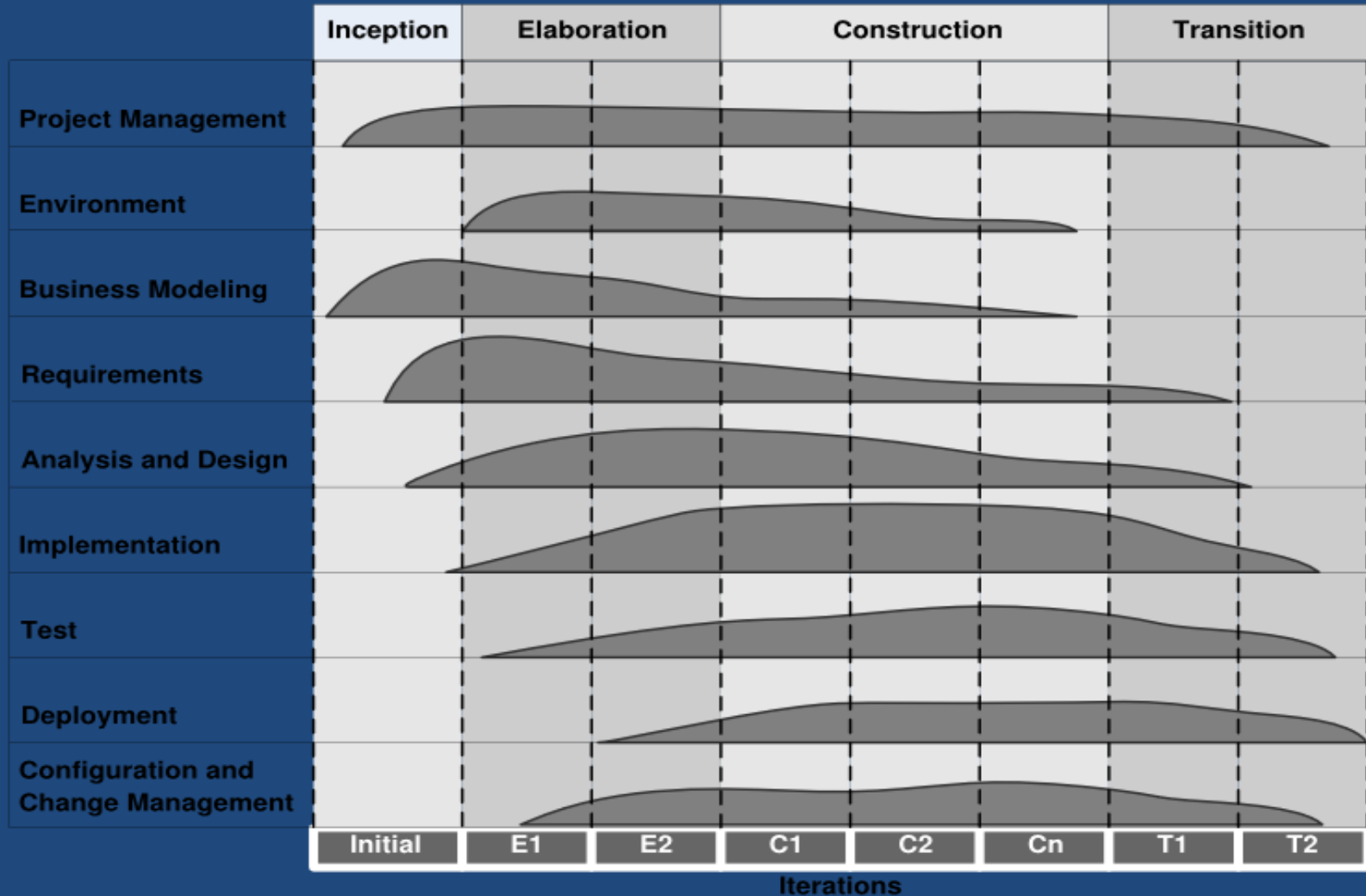
- Actividades genéricas relacionadas con las Fases del PU.



Proceso Unificado

- Características Principales
 - Dirigido por Casos de Uso
 - Iterativo e Incremental
 - Centrado en la Arquitectura de software

Proceso Unificado



Ingeniería del software de sala limpia

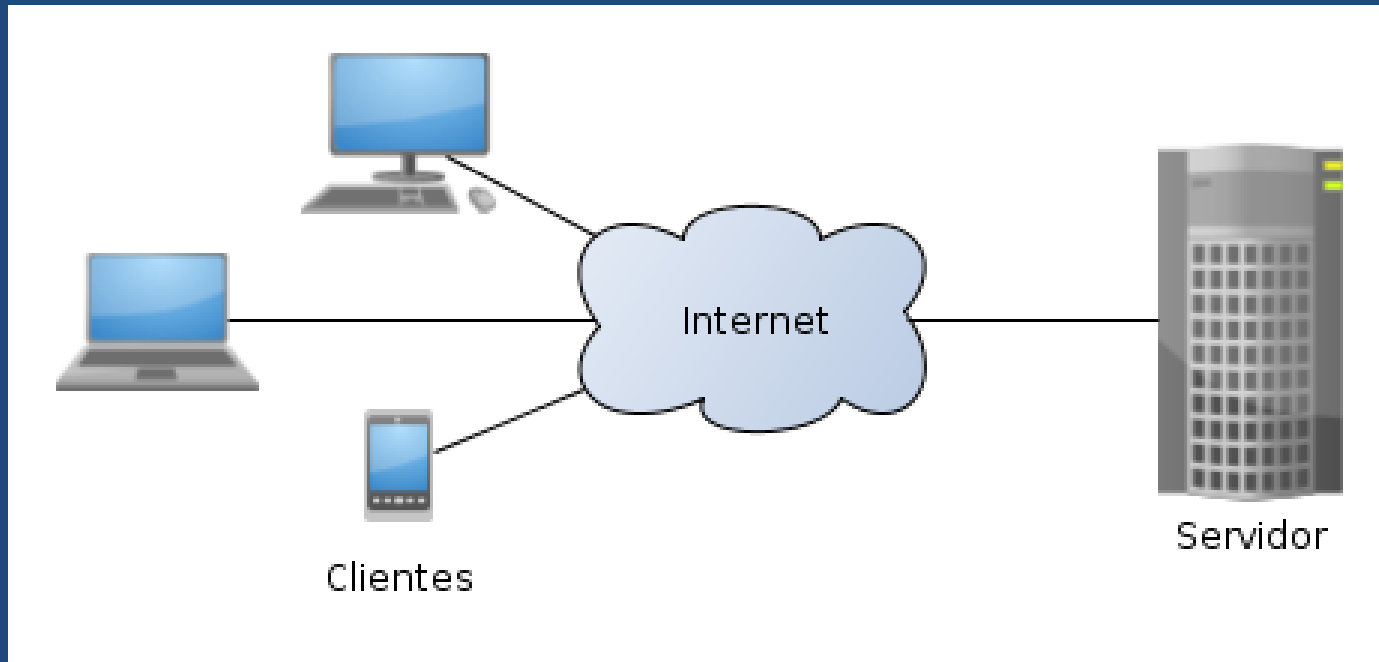
- Este enfoque hace hincapié en la necesidad de incluir la corrección en el software a medida que éste se desarrolla
- Conjunto de técnicas destinadas a la producción de un producto software Es un enfoque formal para el desarrollo del software, que pueda dar lugar a un software con una calidad notablemente alta.
- El proceso de sala limpia hace hincapié en el rigor, en la especificación y en el diseño, y en la verificación formal de cada uno de los elementos del modelo de diseño resultante mediante el uso de pruebas de corrección basadas en fundamentos matemáticos

Ingeniería del software de sala limpia

- Caja negra: especifica el comportamiento del sistema o una parte del mismo. El sistema responde a estímulos específicos mediante la aplicación de un conjunto de reglas de transición que hacen corresponder el estímulo con la respuesta.
- Caja de estado: encapsula los datos de estados y de servicios de forma análoga a los objetos. Se representa las entradas y las salidas . representa la historia de estímulos de la caja negra (los datos encapsulados en la caja de estado que deben ser mantenidos entre las transiciones implicadas..
- Caja transparente: se definen en esta caja las funciones de transición que están implicadas en la caja de estados.

Ingeniería del software cliente-servidor

- En una estructura C/S la computadora que reside por encima de otra computadora se denomina SERVIDOR y las computadoras de nivel inferior se denominan CLIENTES.



Ingeniería del software cliente-servidor

- Servidor de archivos: el cliente solicita registros específicos de un archivo y el servidor transmite estos registros al cliente a través de la red.
- Servidor de Bases de datos: el cliente envía solicitudes en lenguaje de consulta estructurado (SQL) al servidor. Estas se transmiten como mensajes a través de la red. El servidor procesa la solicitud SQL y halla la información solicitada, pasando únicamente los resultados al cliente.
- Servidor WEB?

GESTION - Gestión de Proyectos

- La gestión eficaz de un proyecto de software se centra en las cuatro P's:
 - Personal
 - Producto
 - Proceso
 - Proyecto
 - Planificar y controlar un proyecto es la única manera conocida de gestionar la complejidad.

GESTION – Los jefes de equipo

- Modelo de gestión MOI:
 - Motivación
 - Organización
 - Ideas o Innovación
- Un gestor de proyectos de software debería concentrarse en entender el problema que hay que resolver, gestionando el flujo de ideas y al mismo tiempo haciendo saber a todos los miembros del equipo que la calidad es importante y que no debe verse comprometida.

GESTION – El equipo de software

- Opciones para aplicar a los recursos humanos:
 - n individuos son asignados a m diferentes tareas funcionales, tiene lugar relativamente poco trabajo conjunto; la coordinación es responsabilidad del gestor del software.
 - n individuos son asignados a m diferentes tareas funcionales ($m < n$) de manera que se establecen “equipos” informales; se puede nombrar a un líder al efecto; la coordinación entre los equipos es responsabilidad de un gestor del software.
 - n individuos se organizan en t equipos; a cada equipo se le asignan una estructura específica que se define para todos los equipos que trabajan en el proyecto, la coordinación es controlada por el equipo y por el gestor del proyecto de software.

GESTION – El equipo de software

- Factores de un proyecto que deberían considerarse cuando se planifica el organigrama de equipos de ingeniería del software:
 - La dificultad del problema que hay que resolver.
 - El tamaño del programa(s) resultante(s) en líneas de código o puntos de función.
 - El tiempo que el equipo estará junto.
 - El grado en que el problema puede ser modularizado.
 - La calidad requerida y fiabilidad del sistema que se va a construir.
 - La rigidez de la fecha de entrega.
 - El grado de sociabilidad (comunicación) requerido para el proyecto.

Metodologías ágiles

- ¿Qué son?
- Gestión ágil de proyectos
 - Scrum
- Desarrollo ágil de aplicaciones
 - TDD / ATDD
 - XP

Bibliografía y referencias

- Pressman, Roger. Ingeniería del software: un enfoque práctico. 7ª.ed., México, D.F., McGraw-Hill
 - Capítulos: I, II, III y IV



FIN