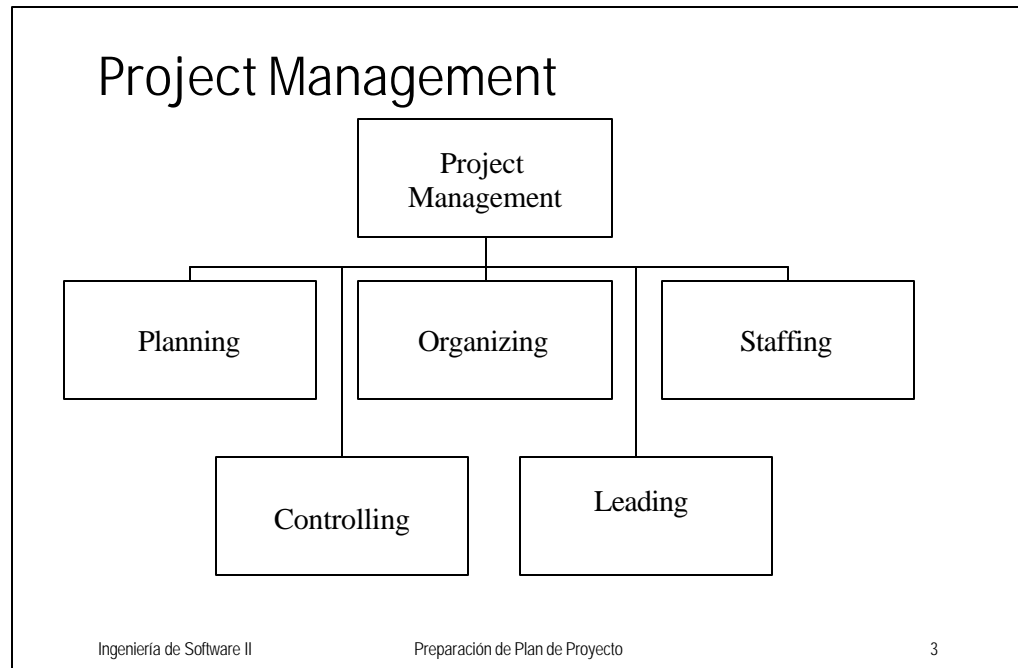
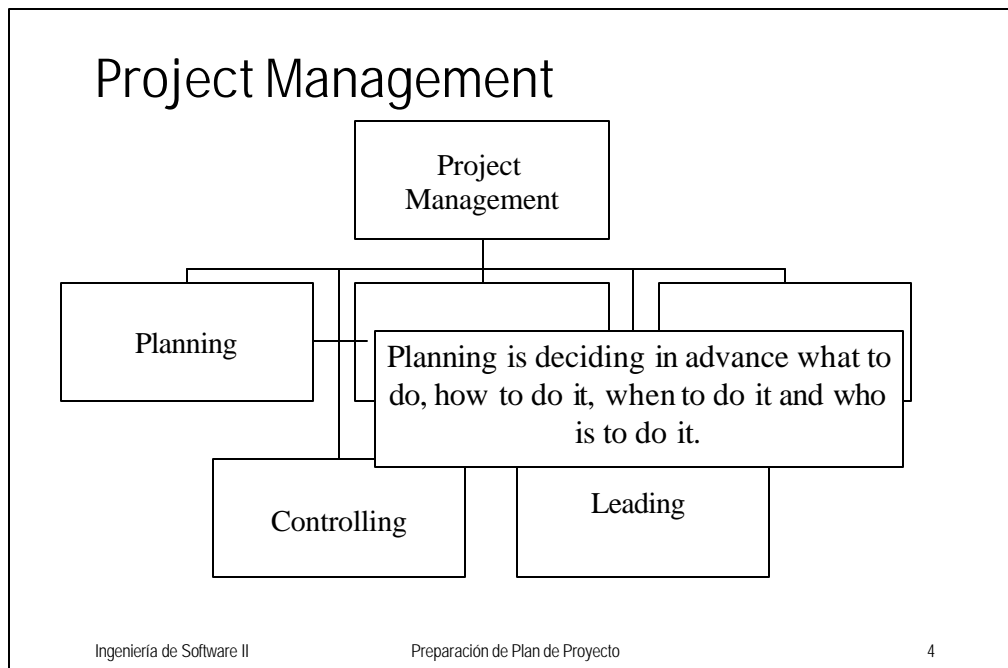


Preparación de  
Plan de Proyecto

## Contenido

- ➔ Etapas en la Preparación
  - Plan de Proyecto
  - Estructura del Equipo de Proyecto
  - Pasos en la Preparación del Work-Plan
  - Seguimiento y Supervisión
- ➔ Planificación del Ciclo de Vida
  - Ciclo de Vida
  - Algunos Modelos
  - Conclusiones
- ➔ Conclusiones





### Desafíos

- Minimizar Retrabajo
- Estabilizar Requerimientos
- Poder seguir el estado
- Perfecto balance entre Tiempo, Costo, Funcionalidades, Calidad y Recursos contra las Espectativas del cliente.
- Poder medir impacto de cambios.

Ingeniería de Software II

Preparación de Plan de Proyecto

5

#### Desafíos

##### -Minimizar Retrabajo

Los errores de fases "previas" encontrados en fases "siguientes" que deben ser corregidos, generan tareas no previstas. El volumen de estas tareas depende de la distancia entre fases y de la complejidad del proyecto.

##### -Estabilizar Requerimientos

Los cambios en los requerimientos fuera de la etapa de blueprint, necesariamente afectan la agenda del proyecto por no haber sido este esfuerzo contemplado en la planificación.

Las estrategias a seguir para controlar esta variable contemplan en ciclos de vida como Waterfall el congelar requerimientos (lo cual es al menos muy difícil en gran parte de proyectos que deben acompañar la dinámica del negocio), o en el otro extremo, XP propone acompañar la dinámica de requerimientos fragmentando el mismo en pequeñas porciones autocontenidas que se implementan en ciclos de no más de 2 semanas.

##### -Poder seguir el estado

Métricas sobre parámetros de interés (issues). Al menos un milestone por fase.

##### -Perfecto balance entre Tiempo, Costo, Funcionalidades, Calidad y Recursos contra las Espectativas del cliente.

El mejor líder de proyecto será quien consiga el resultado más parecido a la negociación acordada en project agreement.

##### -Poder medir impacto de cambios.

Cada vez que el requerimiento cambia, debemos poder medir el costo del cambio para replanificar y renegociar pautas.

Desde el punto de vista de la planificación, es importante conocer de antemano todas las variables y que sea decisión del equipo de proyecto los puntos del requerimiento que no serán alcanzados, o hasta qué punto será aceptable niveles de calidad inferiores a los definidos inicialmente.

## Contenido

- ➔ Etapas en la Preparación
  - Plan de Proyecto
  - Estructura del Equipo de Proyecto
  - Pasos en la Preparación del Work-Plan
  - Seguimiento y Supervisión
- ➔ Planificación del Ciclo de Vida
  - Ciclo de Vida
  - Algunos Modelos
  - Conclusiones
- ➔ Conclusiones

## Plan de Proyecto

### ➤ ¿Qué es un Plan de Proyecto?

Es la fotografía de las acciones que se deben tomar para alcanzar un objetivo determinado.

### Etapas en la preparación – Plan de Proyecto

- ✓ La palabra "Blueprint" (sin traducción breve) refleja lo que un plan de proyecto es:  
"Una fotografía de las tareas a ser realizadas para alcanzar un objetivo bien definido"
- ✓ Debemos llegar a un acuerdo con los directivos y el patrocinador del proyecto a fin de alinear los beneficios clave del proyecto con los objetivos del negocio. (Business Case)
- ✓ La elección del ciclo de vida se realiza también en esta etapa.

## Equipo del Proyecto

### ➔ Roles

- ▣ Directive Board
- ▣ Configuration Control Board
- ▣ Sponsor
- ▣ Stakeholders
- ▣ Project Manager
- ▣ Staff

Ingeniería de Software II

Preparación de Plan de Proyecto

8

### Etapas en la preparación – Equipo del Proyecto

#### ✓ Directive Board

Compuesto por los directivos más altos de la corporación. Son los responsables de aprobar el Business Case y funciones principales y su justificación en el contexto del negocio.

#### ✓ Configuration Control Board

Equipo de expertos (técnicos o funcionales) en las áreas afectadas del negocio. Deben tener un nivel de decisión acorde a la envergadura del proyecto. Puede estar solo formado por los stakeholders.

#### ✓ Sponsor

#### ✓ Stakeholders

Cualquier persona con conocimiento requerido para la realización del proyecto.

En general el stakeholder es seleccionado de entre los usuarios porque no solo tiene el peso "político" como para contribuir al éxito, sino que en general tienen una visión más global de los procesos de negocio y los beneficios que el cambio propuesto conlleva.

Entonces, por cada perfil de usuario diferente voy a necesitar un stakeholder para representar concretamente ese perfil y contrastar mi análisis contra la realidad.

#### ✓ Project Manager

#### ✓ Staff

Personas pertenecientes al equipo de trabajo del proyecto que se encuentran altamente comprometidas.

## Pasos en la Preparación del Plan

### ➤ Procesos Clave

- ▣ Factor Analysis
- ▣ Project Agreement
- ▣ Change Control Procedures
- ▣ Work Breakdown Structures
- ▣ Estimating Tasks
- ▣ Schedule Creation
- ▣ Risk Assessment

Ingeniería de Software II

Preparación de Plan de Proyecto

9

### Etapas en la preparación – Pasos en la Preparación del Plan de Proyecto

Una posible metodología para la construcción de un Work Plan es explicada en "Creating a Project Plan" de Joseph Launi.

De acuerdo al objetivo y nivel de visibilidad, se distinguen las siguientes etapas:

- ✓ Factor Analysis
- ✓ Project Agreement
- ✓ Change Control Procedures
- ✓ Work Breakdown Structures
- ✓ Estimating Tasks
- ✓ Schedule Creation
- ✓ Risk Assessment

## Pasos en la preparación del Plan

### ➤ Factor Analysis

- Detecto aspectos en donde es requerido mayor conocimiento.
- El éxito de un proyecto es subjetivo, entonces...

Antes de contraer compromisos debemos investigar, analizar y entender :

¿Cuál es real alcance?, ¿Están los verdaderos clientes involucrados?, ¿Cuál es el criterio de aceptación?, ¿ Cuáles son los entregables adecuados para seguir el proyecto?

### Etapas en la preparación – Pasos en la Preparación del Plan de Proyecto

#### Factor Analysis

*"To be understood, you must seek to understand"* – Steve Covey.

Lo primero que debo hacer es invertir esfuerzo en detectar aquellos aspectos en donde mayor conocimiento es requerido.

Antes de negociar con el cliente debemos asegurarnos de entender el requerimiento y su entorno.

Dado que el éxito de un proyecto es subjetivo a los stakeholders, **analizar y entender** cuales son los factores determinantes de que un proyecto sea considerado exitoso es fundamental.

A partir del análisis de factores identificamos también cuales son los entregables adecuados para el correcto seguimiento de los puntos relevantes del proyecto.

## Pasos en la preparación del Plan

### ➤ Factor Analysis

- Definición y Alcance
- Recursos
- Cronograma
- Procedimientos
- Entorno
- Cambios Permitidos
- Líneas de Comunicación
- Compromiso
- Expectativas
- Riesgos

Ingeniería de Software II

Preparación de Plan de Proyecto

11

### **Etapas en la preparación – Pasos en la Preparación del Plan de Proyecto**

**Definición y Alcance:** Se refiere al objetivo primario del proyecto, está definido a través de sus funciones principales y entregables. Debemos encontrar dentro de estas funciones los motivos por los cuales se alinea el proyecto con los objetivos del negocio.

**Recursos:** Recursos requeridos. Recordar que las estimaciones se hacen sobre los mismos y... No podemos medir lo que no podemos ver.

Los recursos son asignados de acuerdo a la visión subjetiva de los stakeholders y sponsor... Debemos alinear nuestro proyecto a los intereses de la compañía a fin de conseguir dichos recursos.

**Cronograma:** Tiempo estimado para finalizar el proyecto y ajuste con el cronograma global del negocio.

**Procedimientos:** Estándares, (políticas y procedimientos), metodologías, programas de calidad, revisión financiera, requerimiento directivo (sponsor)

**Entorno:** Contexto dentro del cual se desarrollará el proyecto.

**Cambios Permitidos:** Pronosticar futuras condiciones que afecten el requerimiento o el dominio de problema.

**Comunicación:** Reuniones, reportes de estado, presentaciones y complejidades que puedan afectar a la comunicación.

**Compromiso:** Soporte del patrocinador y de los correspondientes stakeholders.

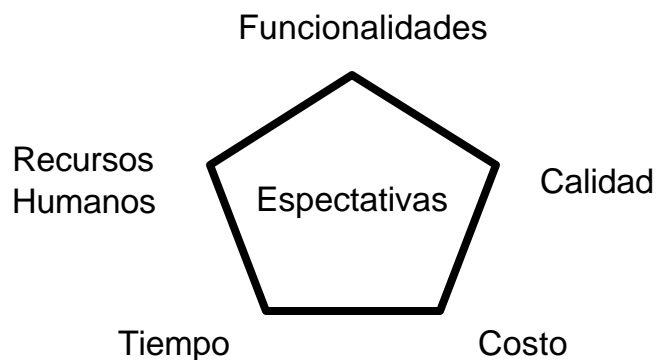
**Expectativas:** Identificar los beneficios del proyecto y asegurar que se condicen con la realidad.

**Riesgos:** Análisis de Riesgos.

## Pasos en la preparación del Plan

### ⇒ Project Agreement

#### ▣ Project Diamond: Administrando Expectativas



Ingeniería de Software II

Preparación de Plan de Proyecto

12

### Pasos en la Preparación del Plan de Proyecto – Project Agreement

- ✓ Con muy pocas excepciones, el estimado inicial de recursos y agenda de proyecto es inaceptable. Esto no ocurre porque el equipo de proyecto sea ineficiente sino porque usualmente el usuario pide más de lo que puede enfrentar. También ocurre que normalmente las nuevas funciones requeridas son solicitadas tardíamente; es extraño que un usuario pueda prevenir una necesidad con la suficiente antelación como para poder construirla a tiempo. Es una buena práctica identificar las funciones esenciales y dejar las secundarias para futuras entregas; es una buena regla general que la primera versión será extremadamente cara si contiene el 100% de las funciones del producto.
- ✓ El Administrador del Proyecto debe especificar y entender en conjunto con sus stakeholders, la dimensión de cada uno de los vértices del "Project Diamond".
- ✓ Este acuerdo es un contrato entre el administrador y el patrocinador del proyecto. Muchas veces se expresa a través de un "Business Case".
- ✓ J. Launi fija sólo cuatro dimensiones, hemos adaptado la presentación de acuerdo a la visión de la cátedra presentada en la clase "Basarse en Principios"

## Pasos en la preparación del Plan

*Development from a requirement is as easy as walking on the water...*

*If both are frozen*

### ⇒ Change Control Procedures

#### ■ Actividades

- Administración de Requerimientos
- Administración de Cambios
- Project Agreement Supuestos y dependencias

- Debo administrar los cambios en los requisitos de modo que sean permitidos sólo previos al diseño de cada iteración.

Ingeniería de Software II

Preparación de Plan de Proyecto

13

### Pasos en la Preparación del Plan de Proyecto – Change Control Procedures

- ✓ Los cambios son inevitables y por consiguiente debemos administrarlos:
  - ✓ Administración de requerimientos
  - ✓ Administración de cambios sobre el entorno.
- ✓ Debemos documentar en el acuerdo del proyecto aquellos supuestos y dependencias sobre los cuales realizamos nuestra construcción... Cualquier cambio en ellos implica un impacto en el proyecto.
- ✓ Los procedimientos de cambio incluyen la aprobación del CCB (Configuration Control Board)
- ✓ Normalmente se analiza el requerimiento de una parte del problema y se comienza con el diseño acotado a dicha fracción del sistema, una vez comenzado esto, no es una buena práctica permitir cambios en el requerimiento durante la implementación de dicho módulo; los cambios efectuados "en medio" del proceso de diseño e implementación son normalmente destructivos.

## Pasos en la preparación del Plan

### ⇒ Work Breakdown Structures

¿Qué es?

- ⇒ Método para representar jerárquicamente las partes de un proceso o producto
- ⇒ Se basa en factorizar el entregable principal y medir las piezas resultantes.

### Pasos en la Preparación del Plan de Proyecto – Work Breakdown Structures

**Objetivo:** Determinar las fases, tareas y dependencias, y entregables a ser completados para considerar que el proyecto a cumplido con el objetivo.

**Idea:** El entregable que represente el objetivo es colocado como raíz de un arbol donde se va abriendo en  $7 \pm 2$  "hijos" hacia abajo hasta alcanzar el nivel mínimo al cual es posible medir en forma eficiente y certera.

## Work Breakdown Structures

### ➔ ¿Cómo se construye un WBS?

1. Definir el propósito del WBS
2. Identificar el nodo raíz (nombre del proyecto/producto)
3. Dividir cada componente en subcomponentes (hasta 7 +/- 2 elementos)
4. Continuar la división hasta que se cumpla con el objetivo (ej: poder estimar o asignar tareas)
5. Desarrollar un diccionario

### Pasos en la Preparación del Plan de Proyecto – Work Breakdown Structures

- ✓ Referirse a "Work Breakdown Structures" de Richard E. Fairlay.

## Work Breakdown Structures

### ⇒ Tipos de WBS

#### ⇒ WBS de proceso

- Usado por estimadores
- La raíz identifica el nombre del proyecto
- El segundo nivel identifica elementos mayores
  - Planificación, organización, análisis de req., diseño, etc
- Partición de un proceso en subprocesos hasta obtener tareas individuales (1 o 2 personas) a desarrollar en poco tiempo (1 a 2 semanas)

### Pasos en la Preparación del Plan de Proyecto – Work Breakdown Structures

- ✓ Referirse a "Work Breakdown Structures" de Richard E. Fairlay.

## Work Breakdown Structures

### ⇒ Tipos de WBS

#### ⇒ WBS de producto

- Usado por ingenieros de software y sistemas. Altamente relacionado con la arquitectura del producto.
- Identifica componentes e interfaces del producto
- Identifica hardware, software y datos
- La raíz identifica el nombre del producto
- Los otros elementos son ítems discretos e identificables de hardware, software y datos

### Pasos en la Preparación del Plan de Proyecto – Work Breakdown Structures

- ✓ Referirse a "Work Breakdown Structures" de Richard E. Fairlay.

## Work Breakdown Structures

### ⇒ Tipos de WBS

#### ⇒ WBS híbrido

- Combina elementos de los dos tipos anteriores
- La raíz es un proceso, alternando elementos de proceso y producto y termina con elementos de producto
- La idea es que los procesos producen productos y los subproductos requieren procesos para su desarrollo
- Utilizado por managers que quieren priorizar la estimación y control precisos de cada elementos de producto

### Pasos en la Preparación del Plan de Proyecto – Work Breakdown Structures

- ✓ Referirse a "Work Breakdown Structures" de Richard E. Fairlay.

## Ejemplo de WBS

## Pasos en la preparación del Plan

### ⇒ Estimating Tasks

- ⇒ Cada tarea atómica debe ser especificada por:
  - ▣ Nombre, número y descripción
  - ▣ Duración estimada
  - ▣ Recursos que necesita
  - ▣ Entregables y productos del trabajo realizado
  - ▣ Criterio de terminación (incluyendo calidad)
  - ▣ Riesgos asociados a la terminación con éxito

### Pasos en la Preparación del Plan de Proyecto – Work Breakdown Structures

- ✓ Al recorrer el árbol desde la raíz hacia abajo aumentamos el nivel de detalle con lo cual, nuestra estimación es más certera.
- ✓ Cada nodo "medible" (generalmente las hojas) se asocia a un Work Package que tiene una asignación directa a un grupo de miembros staff.
- ✓ Los work package representan unidades medidas y seguidas desde el cronograma.
- ✓ Los milestones serán eventos en los cuales veremos el estado de avance de cada work package.

## Pasos en la preparación del Plan

### ➤ Schedule Creation

- Determinar todas las dependencias entre las tareas
  - Permite optimizar asignación de recursos y paralelización de tareas
- Identificación de Milestones (puntos de revisión)
- Rollbacks y ciclos de tareas por chequeos
  - Ej: el testing puede implicar ajustes en el código
- Dependencias externas
  - Proyectos técnicos o de negocios
  - Otros proyectos en la organización con impacto en el nuestro

## Schedule Creation

### ⇒ Camino Crítico

- ⇒ Es la secuencia de tareas cuyo atraso provoca atrasos en el fin del proyecto
- ⇒ Las herramientas lo calculan automáticamente (grafos PERT y Gantt)
- ⇒ Una tarea no crítica tiene un margen de tiempo a partir del cual se convierte en crítica
- ⇒ Por lo tanto:
  - ▣ El mayor esfuerzo de la estimación debe ser en las tareas críticas
  - ▣ El análisis del cronograma tendiente a comprimir tareas debe comenzar desde las tareas del camino crítico.

### Schedule Creation – Camino Crítico

- ✓ Cada tarea posee un grado de tolerancia denominado "floating days" que no son más que los días que puede excederse la finalización de la misma sin producir un retraso sobre el proyecto en su conjunto.
- ✓ Las tareas del camino crítico son justamente las que se identifican como "0 floating days".

## Schedule Creation

### ⇒ **Alocación de Recursos**

- ⇒ Identificar los recursos del proyecto y asignar disponibilidad de tiempos
- ⇒ OJO: Verificar real disponibilidad y comunicar su asignación antes de asignar el recurso al proyecto
- ⇒ Al comienzo del proyecto se debe estimar la duración de las tareas asumiendo dedicación completa de los recursos
  - ▣ Recursos sobre alocados pero se obtiene una estimación absoluta de los tiempos del proyecto

## Schedule Creation

### ➔ **Alocación de Recursos**

- ➔ Se identifican y resuelven los conflictos de sobrealocación
  - ▣ Con cambio del orden y duración de tareas o asignación de recursos
  - ▣ OJO: no olvidar dependencias con otros proyectos (puede haber recursos compartidos)
- ➔ Uno de los recursos más críticos que en general esta sobrealocado
  - ▣ EL USUARIO

## Schedule Creation

### ⇒ **Alocación de Recursos**

- ⇒ Un error muy común es la sobreestimación de “la dedicación completa”
  - Vacaciones/feriados
  - Enfermedades
  - Embarazos
  - Capacitación
  - Interferencias por actividades de soporte
  - Horas reales de trabajo (almuerzos, descansos)

Ingeniería de Software II

Preparación de Plan de Proyecto

25

### **Interferencias por actividades de soporte**

Se debe buscar organizar los equipos de proyecto de modo que las actividades de soporte no recaigan sobre los recursos asignados al mismo. De todos modos, dado que requerimos de usuarios en el equipo, estos recursos difícilmente puedan desatender la actividad diaria.

## Schedule Creation

### ⇒ Consejos

- ⇒ Definir en forma dinámica los tiempos de tareas en relación a las dependencias
- ⇒ Cada tarea debe tener un entregable que permita la evaluación de su concreción
- ⇒ Establecer hitos (puntos de control) para revisar avance y plan
- ⇒ Planes de contingencia

## Schedule Creation

### ⇒ Consejos

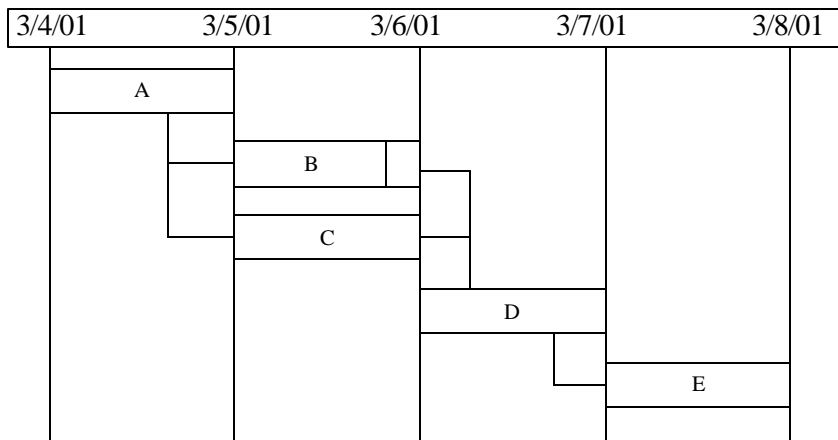
- ⇒ Revisar planes similares y curva de cumplimiento
- ⇒ Integración: de tareas y como una tarea en sí
- ⇒ No olvidarse de las dependencias externas
  - ▣ Otros proyectos
  - ▣ Recursos asignados a otros proyectos
  - ▣ Usuarios
  - ▣ Proveedores
  - ▣ Disponibilidad de hardware y software
  - ▣ Tercerización

## Schedule Creation

### ⇒ Gantt

- ⇒ Técnica de control de proyecto que puede ser utilizada para Scheduling y Plan de recursos
- ⇒ Gráfico de barras
- ⇒ Cada barra representa una actividad
- ⇒ Se dibujan contra una línea de tiempo
- ⇒ La longitud de cada barra representa la longitud de tiempo de esa actividad
- ⇒ Se le puede asignar a las tareas los recursos

## Ejemplo Gantt



Ingeniería de Software II

Preparación de Plan de Proyecto

29

## Risk Assessment

- ➔ El análisis de riesgos
- ➔ La ejecución de Planes de Contingencia.
- ➔ La evaluación de nuevos escenarios y consiguientes nuevos riesgos.

## Etapas en la Preparación

### ⇒ Seguimiento y Supervisión.

- ▣ Monitorear medibles del Proyecto.
- ▣ Reaccionar en forma temprana a desvíos.
- ▣ Identificar recursos con retrasos y posibles extensiones en el equipo.
- ▣ Monitorear riesgos.

## Contenido

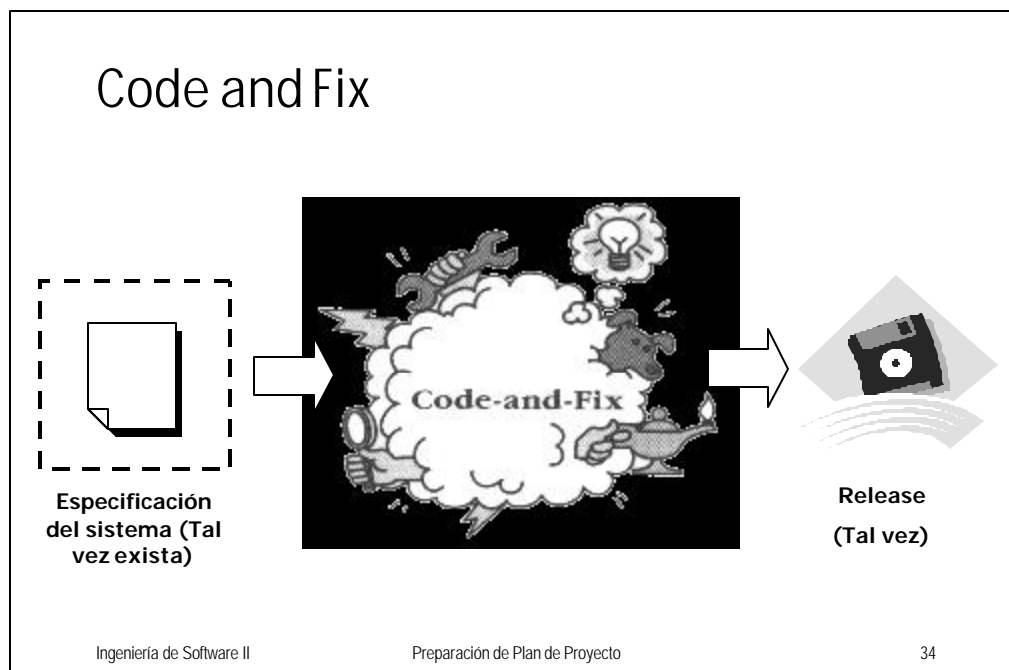
- Etapas en la Preparación
  - ▣ Plan de Proyecto
  - ▣ Estructura del Equipo de Proyecto
  - ▣ Pasos en la Preparación del Work-Plan
  - ▣ Seguimiento y Supervisión
- Planificación del Ciclo de Vida
  - ▣ Ciclo de Vida
  - ▣ Algunos Modelos
  - ▣ Conclusiones
- EUP
  - ▣ Beneficios Clave
  - ▣ Actividades
  - ▣ Relación entre Core Workflows y Fases del Plan
- Conclusiones

## Ciclo de Vida

- Es el conjunto y la disposición de las actividades que suceden desde la concepción del sistema hasta que el mismo es desinstalado de la última máquina del usuario.
- Tiene como función establecer el criterio bajo el cual proceder de un tipo de tareas a otro.

### **Ciclo de Vida**

Dependiendo del ciclo de vida que uno elija, es posible mejorar la velocidad del desarrollo, mejorar la calidad, facilitar el seguimiento, reducir la exposición a riesgos o mejorar el contacto con el cliente. La elección errónea puede producir reducción en la productividad, re-trabajo y frustración.



### Ciclo de Vida – Code and Fix

Este ciclo de vida es raramente útil pero sin embargo altamente utilizado. Si no se ha explícitamente elegido un ciclo de vida probablemente sea éste el que está siendo usado.

El ciclo comienza con una idea general sobre lo que debe ser construido. Es posible que exista una especificación, sin ser la misma necesaria para comenzar a construir. Luego se comienza a usar cualquier combinación de metodologías de diseño informal, codificación y testing hasta que el producto está listo para ser liberado.

## Code and Fix

➤ Puede llegar a ser útil para pequeños proyectos donde se sabe de antemano que el producto será rápidamente descartado.

➤ Dos ventajas:

- ▣ No posee overhead.
- ▣ No requiere ningún tipo de conocimiento.

➤ Muchas desventajas, entre otras:

- ▣ No tenemos forma de asegurar que existe progreso alguno.
- ▣ No tenemos forma de controlar la calidad ni de identificar riesgos.
- ▣ Es muy probable que se alcancen puntos en el proyecto donde sea necesario descartar absolutamente todo lo realizado.

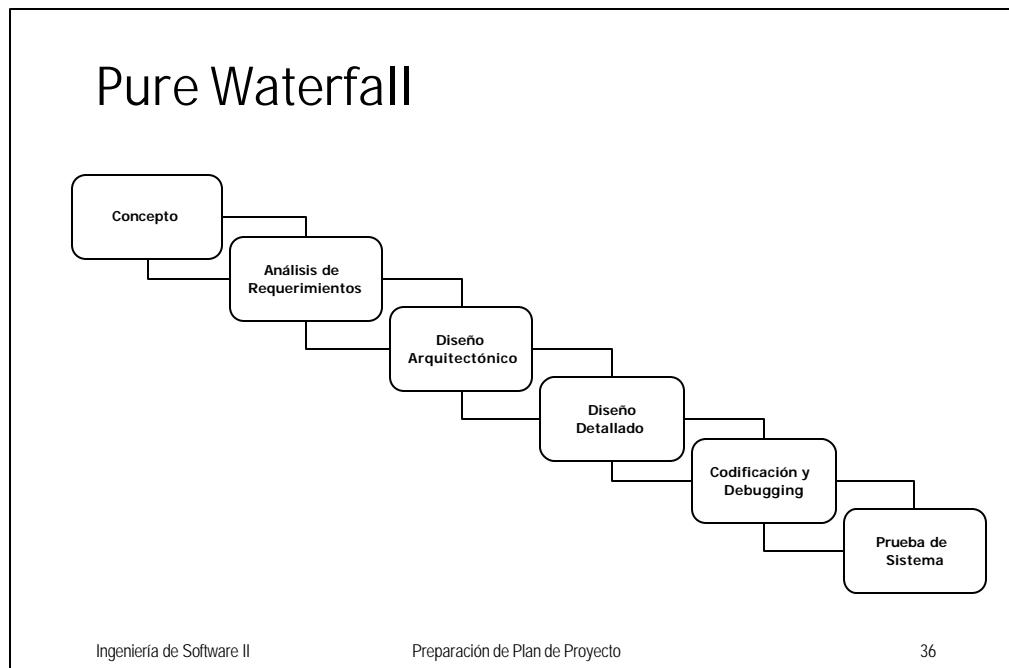
Ingeniería de Software II
Preparación de Plan de Proyecto
35

### Ciclo de Vida – Code and Fix

El modelo tiene dos ventajas. Primero, no posee overhead: uno salta directamente a codificar, uno cree que posee inmediatamente signos de progreso. Segundo, no requiere ningún tipo de conocimiento excepto saber programar: cualquiera puede usarlo.

Para pequeños proyectos que uno sabe que serán descartados rápidamente luego de ser usados, este modelo puede llegar a ser útil (programas de conversión de datos para una migración, pruebas de concepto en general, etc.).

Para cualquier otro proyecto este modelo es peligroso. Puede ser que no tenga overhead pero tampoco tenemos forma de asegurar que existe progreso alguno. Uno codifica hasta que el producto se considera listo (“listo” no tiene métrica asociada tampoco). No tenemos forma de controlar la calidad ni de identificar riesgos. Es muy probable aquí que se alcancen puntos en el proyecto donde sea necesario descartar absolutamente todo lo realizado justamente por no estar el objetivo bien definido y comprendido.

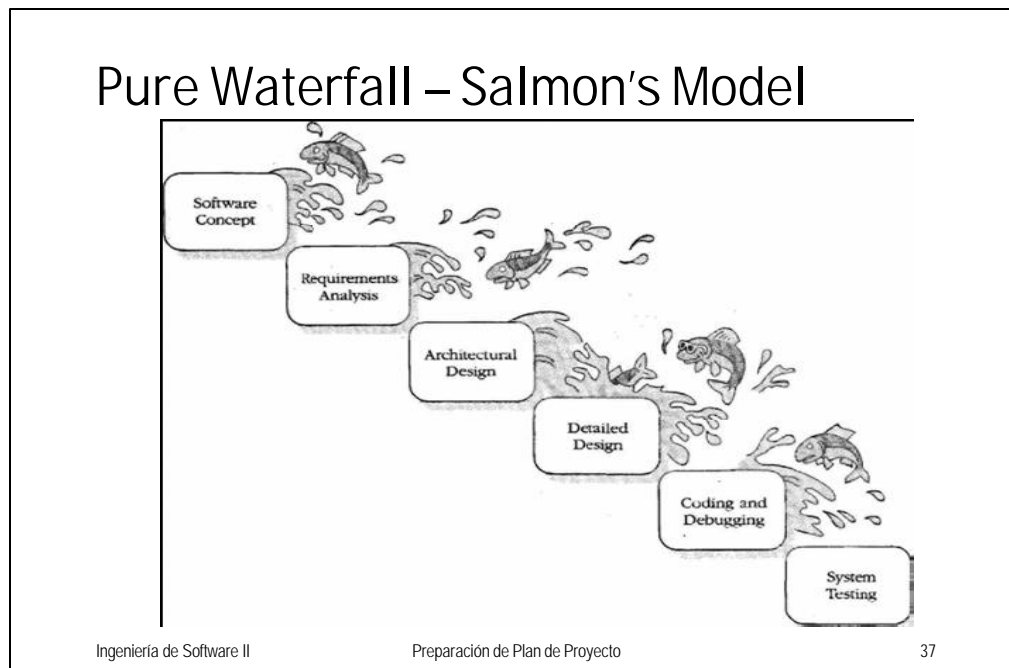


#### Ciclo de Vida – Pure Waterfall

Bajo este ciclo de vida el proyecto progresa a través de una secuencia ordenada de pasos desde la concepción inicial del Software.

El proyecto contempla una revisión al final de cada paso para determinar si se pasará al siguiente. Esto hace que sea posible permanecer por un tiempo indeterminado sobre uno de los pasos si dicha revisión no es alcanzada.

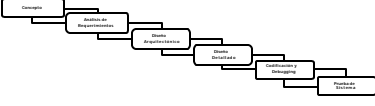
El modelo waterfall es orientado a la documentación lo que significa que la gran mayoría de los productos de software que son intercambiados entre etapas son documentación. Las etapas no se solapan en este modelo.



#### Ciclo de Vida – Pure Waterfall – Salmon's Model

*"Es posible retroceder en las fases pero esto puede costarnos el proyecto"*

## Pure Waterfall



- ⇒ Muy útil cuando los requerimientos de calidad dominan el costo y el cronograma.
- ⇒ Debo conocer muy bien los requerimientos y los métodos a ser usados.

⇒ **Ventajas**

- ▣ Produce Sistemas Confiables y de alta calidad.
- ▣ Minimiza el overhead de planning.

⇒ **Desventajas**

- ▣ Dificultad de obtener requerimientos completamente especificados al inicio del proyecto.
- ▣ La visualización de resultados se presenta al finalizar el proyecto.
- ▣ No es flexible.
- ▣ No apropiado para desarrollos rápidos por cantidad de documentación.

Ingeniería de Software II Preparación de Plan de Proyecto 38

### Ciclo de Vida – Pure Waterfall

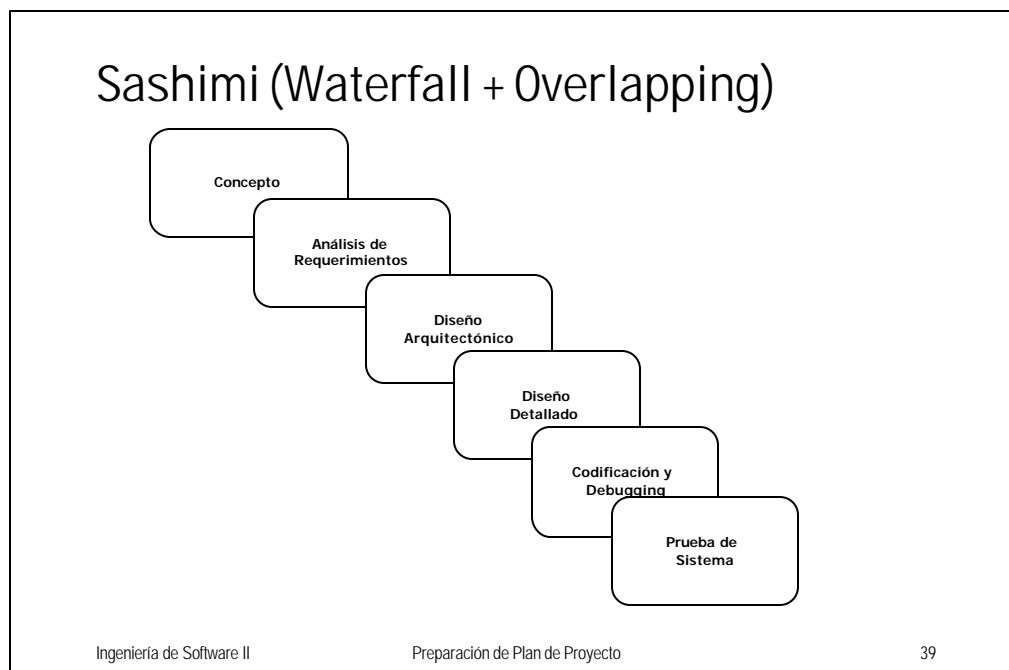
En proyectos donde hay alta estabilidad funciona bien. La estabilidad se relaciona con un gran conocimiento de los requerimientos y de los métodos que serán usados.

En estos casos no sólo es un modelo eficiente sino que es el correcto a usar puesto que tenemos la oportunidad de encontrar errores de alto impacto en etapas tempranas y de bajo costo.

El modelo contribuye a a minimizar el overhead en la planificación porque es posible hacer la actividad de planificación al inicio. Por otra parte, no tenemos resultados tangibles hasta el fin del proyecto. El avance debe ser entonces medido a partir de la documentación generada durante las etapas.

Funciona bien cuando los requerimientos de calidad dominan el costo y el cronograma.

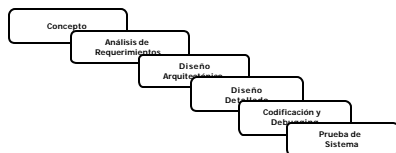
Las desventajas del modelo radican en la dificultad de conocer los requerimientos al 100% antes de comenzar a diseñar.



#### Ciclo de Vida – Sashimi

En el modelo puro Waterfall, la documentación ideal es la que es pasada “de mano en mano” entre los equipos de trabajo que operan en las diferentes fases. La pregunta es “¿Por qué?”. Si el equipo de trabajo es el mismo que opera en cada una de las fases, entonces no es necesaria la documentación creada para transmitir el conocimiento entre fases. Este cambio reduce la documentación a la necesaria para el posterior mantenimiento del Software.

## Sashimi (Waterfall + Overlapping)



- ⇒ Aplicable en condiciones similares al pure waterfall pero con equipos más pequeños.
- ⇒ Asumimos que el mismo equipo realiza actividades en más de una fase.

### ⇒ Ventajas

- Reduce la documentación necesaria en el purewaterfall.
- Mismas ventajas que el purewaterfall.

### ⇒ Desventajas

- Mismas dificultades que el pure waterfall.
- Adicionalmente el solapamiento puede ocasionar conflictos relacionados con la comunicación entre fases solapadas.

Ingeniería de Software II

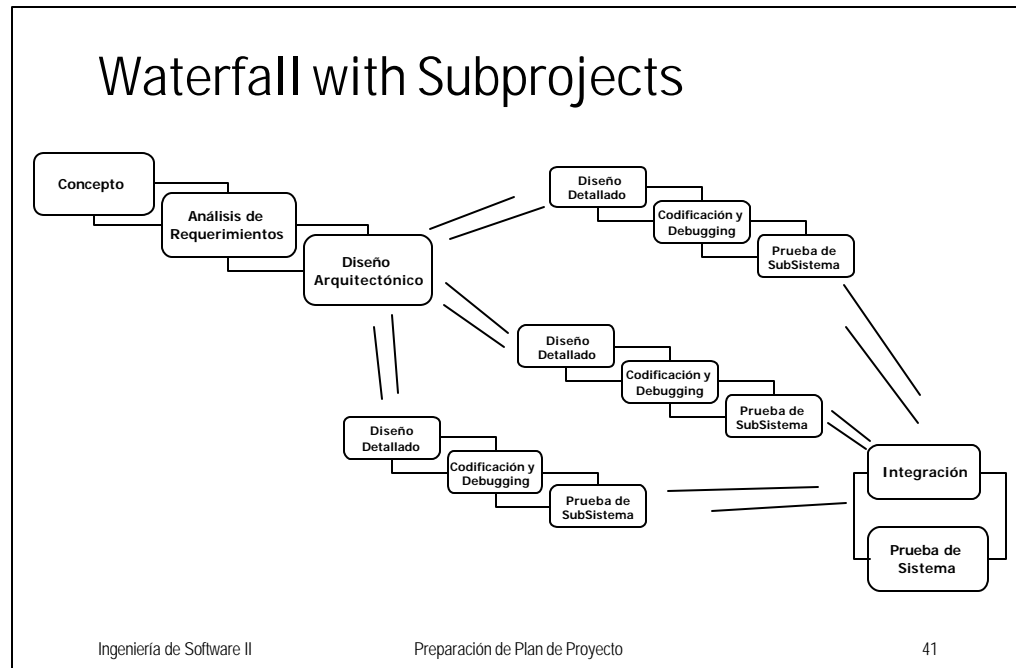
Preparación de Plan de Proyecto

40

### Ciclo de Vida – Sashimi

Como existe solapamiento entre fases, los hitos son más ambiguos y esto hace más difícil realizar el seguimiento con cierto nivel de seguridad.

Efectuar actividades en paralelo requiere que las actividades solapadas estén bien comunicadas, estamos expuestos a que cambios o novedades en actividades “viejas” no sean comunicadas a las actividades “nuevas” y por ende exista inconsistencia a lo largo del ciclo.



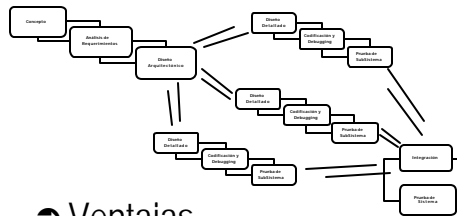
### Ciclo de Vida – Waterfall with Subprojects

Otro problema con el modelo waterfall puro es que se supone que debemos completar el 100% del diseño arquitectónico antes de comenzar con el diseño detallado, y que a su vez debemos completar el 100% del diseño detallado antes de comenzar a codificar. .

Los sistemas tienen ciertas áreas donde existen sorpresas de diseño, pero también existen áreas donde la tarea es simple e incluso en algunos casos conocida. ¿Por qué entonces retrasar el comienzo de los subsistemas simples a causa de la complejidad de parte de la arquitectura?

Si es posible partir la arquitectura en subsistemas lógicamente independientes se pueden obtener subproyectos que sean tratados independientemente y en paralelo hasta llegar al momento de integrarlos.

## Waterfall with Subprojects



- ⇒ Aplicable en condiciones similares al pure waterfall pero donde es posible identificar subsistemas independientes.
- ⇒ El equipo de trabajo es suficientemente grande como para paralelizar el trabajo.

### ⇒ Ventajas

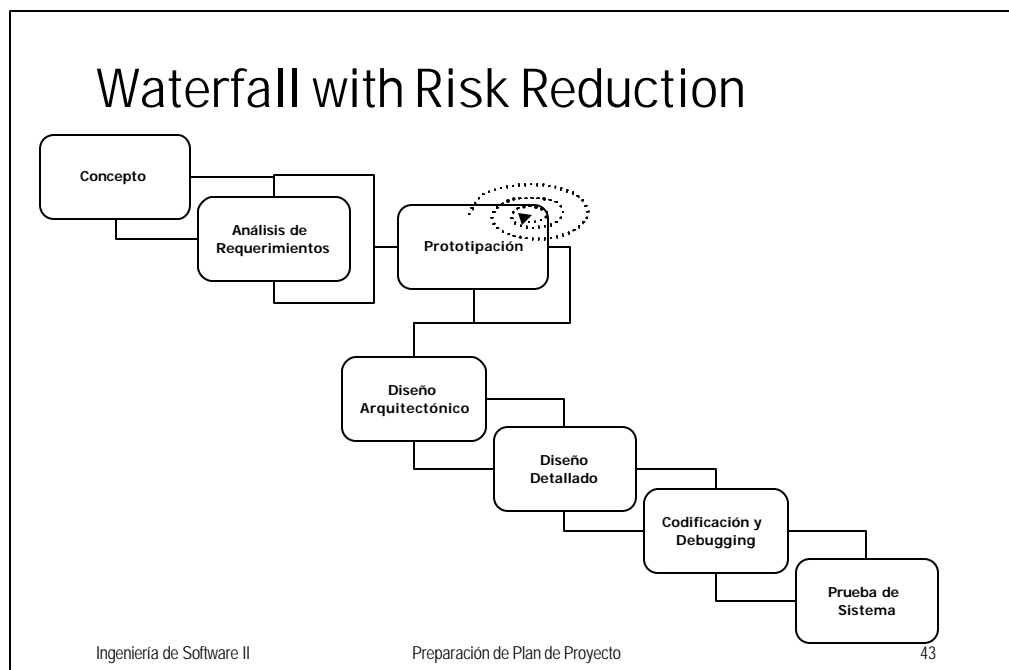
- Permite construir los subsistemas en paralelo.
- Mismas ventajas que el purewaterfall.

### ⇒ Desventajas

- Posibles problemas de integración por interdependencias no identificadas.

### Ciclo de Vida – Waterfall with Subprojects

El mayor riesgo de esto es que existan interdependencias no previstas que generen problemas de integración. Este riesgo es posible mitigarlo con una actividad de arquitectura más intenso que en el caso del pure waterfall, pero nunca podremos eliminarlo por completo.

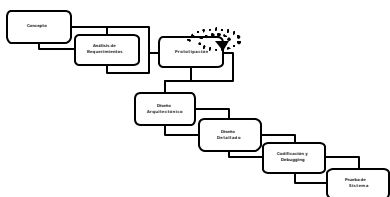


### Ciclo de Vida – Waterfall with Risk Reduction

Otro de los problemas del waterfall puro es que es necesario tener una precisa definición de los requerimientos antes de comenzar con el diseño arquitectónico. Esta necesidad no solo se encuentra en los requerimientos sino que es necesario para ingresar en un fase contar con gran estabilidad en la anterior y con el total de los productos requeridos para ingresar en la nueva fase.

Una posible modificación leve al pure waterfall es la propuesta por "Risk Reduction", donde se incorpora una actividad de prototipado de interfaz o de aquellos componentes de mayor riesgo por la incertidumbre natural al desarrollo de Software. Esta actividad no necesariamente debe centrarse en los requerimientos sino que es posible extenderla al diseño e incluso a la codificación.

## Waterfall with Risk Reduction



⇒ Lo aplico en los casos donde es posible identificar aquellas áreas donde hace falta mayor conocimiento.

(\*) Esto no siempre es posible.

### ⇒ Ventajas

- ▣ Reduce el riesgo con respecto al pure waterfall proveniente de los requerimientos incompletos o mal definidos.
- ▣ Mismas ventajas que el purewaterfall.

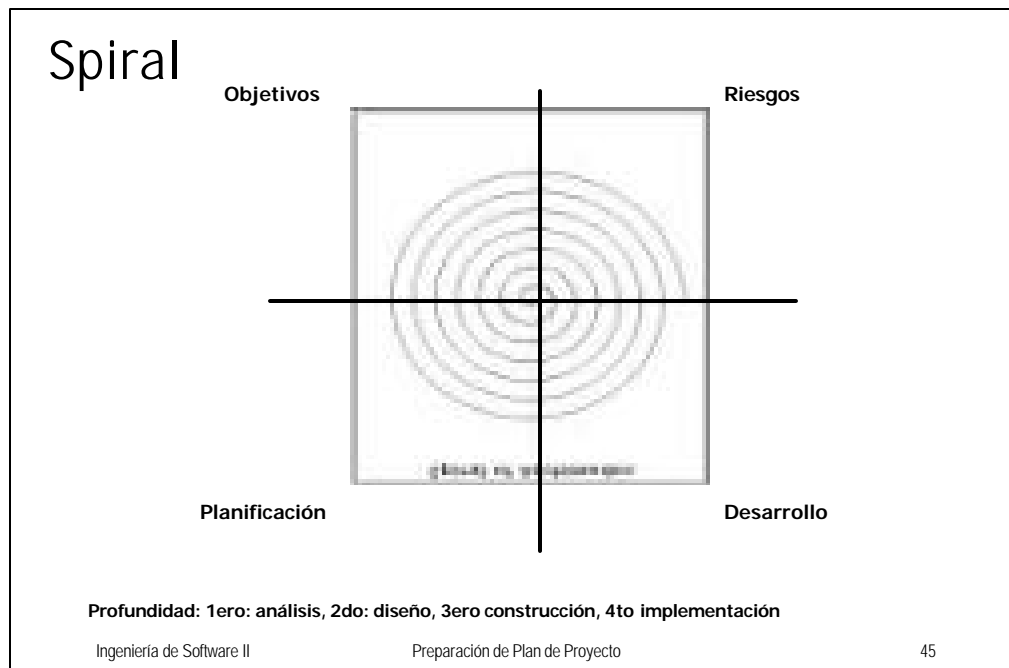
### ⇒ Desventajas

- ▣ Debo poder identificar las áreas donde sea necesaria mayor definición.

### Ciclo de Vida – Waterfall with Risk Reduction

La actividad de prototipado es generalmente beneficiosa aunque existe la posibilidad de no poder recolectar mayor información incurriendo en gastos que no tienen rédito en el proyecto.

Dado que luego de la actividad de prototipado el ciclo es idéntico al pure waterfall estamos frente a las mismas dificultades que en dicho modelo.



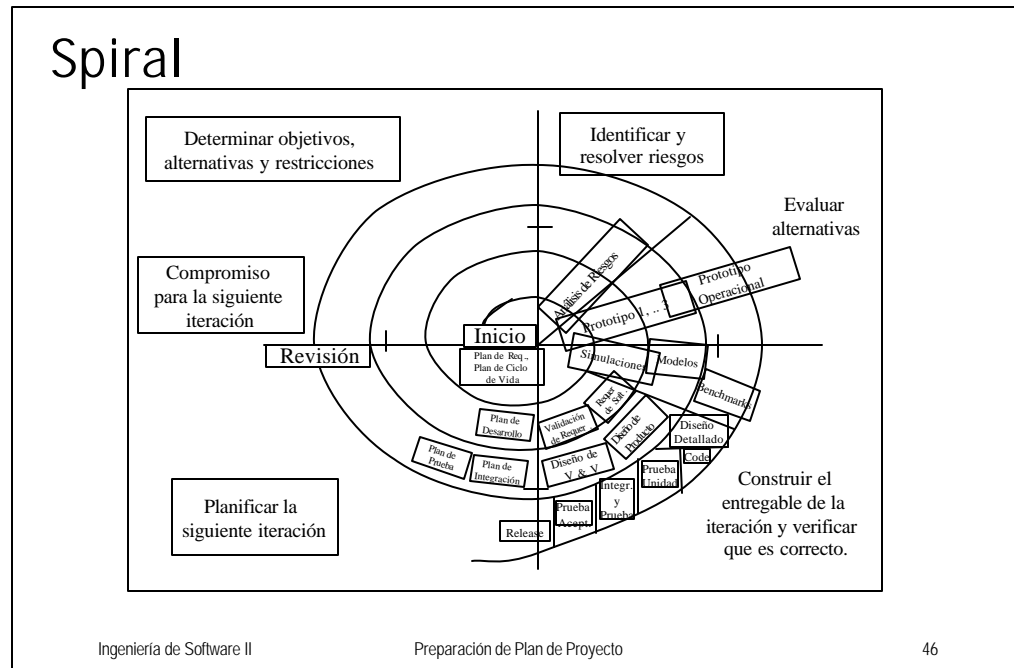
### Ciclo de Vida – Spiral

El ciclo de vida spiral es un modelo orientado a riesgos, donde el proyecto es dividido en mini-proyectos. Cada uno de los mini-proyectos atiende a uno a más riesgos “importantes” hasta que finalmente todos los riesgos con alta exposición han sido atendidos.

El concepto de riesgo es el visto en clase, se refiere a requerimientos pobremente entendidos, a arquitectura pobremente definida o comprendida, problemas potenciales de performance, problemas en la tecnología subyacente, y demás temas del proyecto donde sea requerido mayor conocimiento.

Una vez que los riesgos han sido atendidos el ciclo de vida finaliza como el pure waterefall.

La idea detrás del modelo es que uno comienza con un alcance reducido en el centro de la espiral, explora los riesgos ç, construye el plan para atender los riesgos encontrados, y luego planea el siguiente ciclo. Cada iteración mueve el proyecto hacia un alcance más extenso.

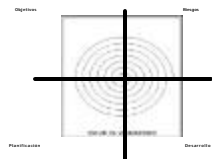


### Ciclo de Vida – Spiral

Cada iteración comprende 6 pasos representados en los cuadros que rodean la espiral del slide.

1. Determinar objetivos, alternativas y restricciones.
2. Identificar y resolver riesgos.
3. Evaluar alternativas.
4. Desarrollar los entregables de la iteración y verificar que son correctos.
5. Planificar la siguiente iteración.
6. Si se decide continuar, obtener compromiso para la siguiente iteración.

## Spiral



⇒ Modelo orientado a manejo de riesgos. A medida que avanza el tiempo y dinero invertidos, la exposición al riesgo disminuye.

### ⇒ Ventajas

- Equilibrio óptimo entre exposición al riesgo e inversión.
- Mayor o equivalente control que en el modelo pure waterfall.

### ⇒ Desventajas

- Requiere gran conocimiento de gestión por parte de quien dirige el proyecto.
- Es posible que si en un momento del proyecto la exposición al riesgo es baja, el modelo se vuelva innecesariamente caro.

### Ciclo de Vida – Spiral

En el modelo espiral las etapas tempranas son las más económicas. Uno gasta menos desarrollando el concepto de operación del Software de los que gasta en la ingeniería de requerimientos, y menos en los requerimientos de lo que gasta en el diseño, desarrollando el producto y probándolo.

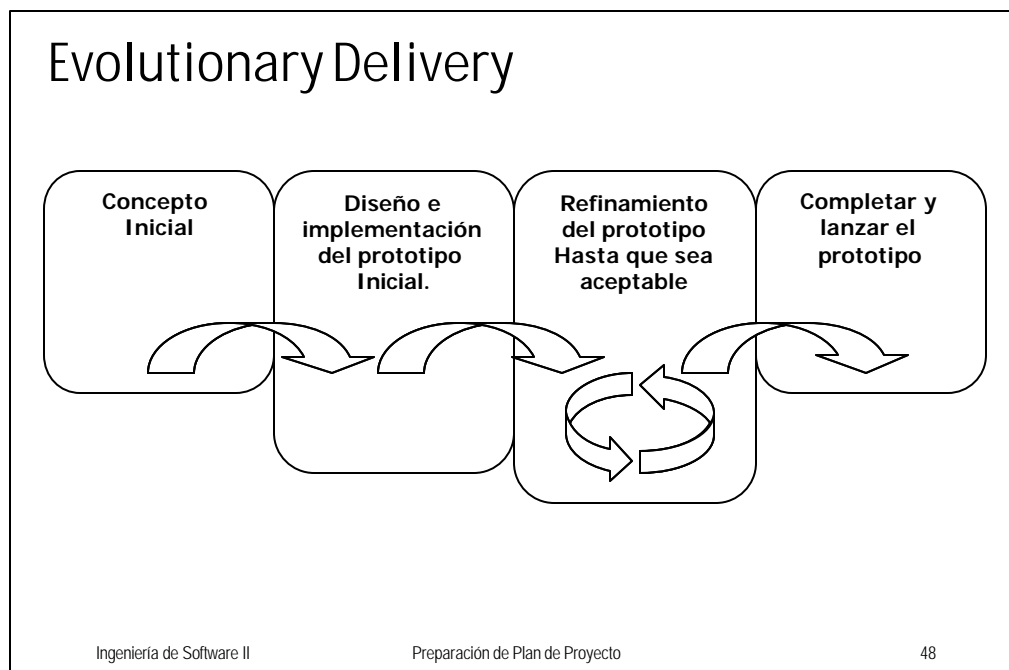
Una de las ventajas más importantes del modelo es que el costo aumenta a medida que el riesgo decrece. A mayor tiempo y dinero invertido, menor la exposición al riesgo. Esto es justamente uno de los atributos más buscados en un proyecto de Software.

El modelo espiral provee igual o mayor control en la gestión del proyecto de la provista por el modelo tradicional pure waterfall. Uno cuenta con puntos de control al finalizar cada iteración. Si el proyecto es inviable debido a razones técnicas o funcionales, es descubierto ésto en forma temprana.

La única desventaja del modelo radica en su complejidad.

Requiere de quién gestiona el proyecto conciencia, atención y conocimiento en gestión. Puede llegar a ser difícil definir milestones objetivos y verificables, que indiquen cuando uno está en condiciones de agregar una nueva iteración.

En algunos casos el producto alcanzado es suficiente, y los riesgos a los que estamos expuesto moderados lo suficiente como para que no sea requerido continuar "pensando en riesgos", con lo que el modelo orientado a riesgos propuesto por el ciclo de vida espiral se vuelve redundante.



#### Ciclo de Vida – Evolutionary Delivery

En este modelo uno desarrolla el concepto del sistema a medida que avanza sobre el proyecto.

Usualmente comenzamos desarrollando los aspectos más **visibles** del sistema. El resultado es mostrado al cliente y el producto evoluciona en base a la información obtenido por parte de éste. En determinado momento el cliente indica que el prototipo es "suficientemente bueno", se completan las tareas remanentes, especialmente las relacionadas con performance, y el prototipo se convierte así en el release.

## Evolutionary Delivery

➤ Modelo orientado a proyectos donde no existe una buena definición de requerimientos

- **Ventajas**
  - ▣ Extracción de requerimientos incremental y buena interacción con el cliente.
- **Desventajas**
  - ▣ Peligro de que se convierta en Code & Fix.
  - ▣ Puede no converger a una solución.

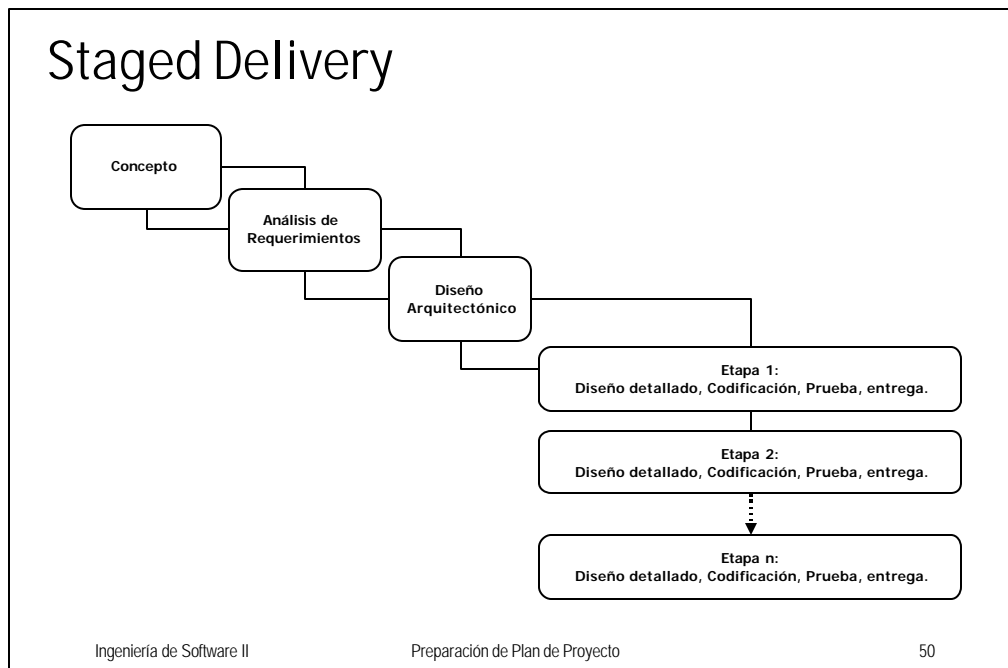
Ingeniería de Software IIPreparación de Plan de Proyecto49

### Ciclo de Vida – Evolutionary Delivery

Este modelo es especialmente útil cuando los requerimientos son dinámicos o pobremente definidos. También es útil cuando el cliente no tiene la capacidad o voluntad para comprometerse con un requerimiento mejor definido, o no existe nadie que conozca bien el dominio de problema.

La mayor desventaja de este modelo radica en que no es posible saber en qué momento el producto convergerá a la solución. Uno desconoce cuantas iteraciones deberán ser necesarias para obtener finalmente el producto.

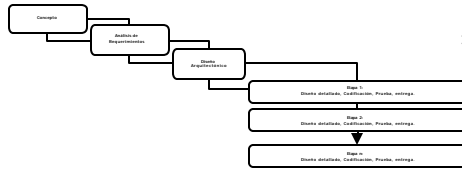
Otra desventaja es que el modelo fácilmente se convierte en Code & Fix. Distinguimos “evolutionary delivery” de “code & fix” principalmente porque la actividad de especificación y diseño existen en el primer modelo.



### Ciclo de Vida – Staged Delivery

El ciclo de vida staged delivery es un modelo en el cual uno muestra el producto al cliente en etapas sucesivas, aumentando en cada una el nivel de refinamiento. En este modelo uno conoce exactamente lo que va a construir desde el comienzo, pero planifica la entrega en etapas (diferencia con el modelo evolutionary delivery).

## Staged Delivery



⇒ Modelo orientado a dividir el requerimiento y realizar un despliegue incremental.

### ⇒ Ventajas

- Las funciones principales se entregan antes.
- El feedback del cliente aumenta a medida que la función es más esencial para el producto...
- Signos tempranos y tangibles de avance.

### ⇒ Desventajas

- Posibles interdependencias entre etapas no identificadas.

Ingeniería de Software II

Preparación de Plan de Proyecto

51

### Ciclo de Vida – Staged Delivery

A nivel de gerenciamiento, se debe estar seguro que las etapas tienen sentido con respecto al uso que el cliente le dará, y que el entregable de cada una de ellas está "autocontenido"

A nivel técnico, se debe asegurar que las dependencias entre los entregables de las etapas no impiden que el producto sea construido independientemente.

La ventaja principal del modelo radica en que nos permite entregar la funcionalidad principal al principio del proyecto. Si las etapas son planeadas con cuidado, podemos reducir el riesgo en los puntos centrales del Software entregándolos primero y pudiendo así obtener feedback operacional antes del fin del proyecto, cuando aún podemos tomar medidas correctivas. Otra ventaja es que provee signos tangibles de avance desde etapas tempranas, lo que facilita el control sobre presión que pueda ejercer el cliente.

## Contenido

- ➔ Etapas en la Preparación
  - Plan de Proyecto
  - Estructura del Equipo de Proyecto
  - Pasos en la Preparación del Work-Plan
  - Seguimiento y Supervisión
- ➔ Planificación del Ciclo de Vida
  - Ciclo de Vida
  - Algunos Modelos
  - Conclusiones
- ➔ Conclusiones

### Contenido

- Etapas en la Preparación
  - ▣ Plan de Proyecto
  - ▣ Estructura del Equipo de Proyecto
  - ▣ Pasos en la Preparación del Work-Plan
  - ▣ Seguimiento y Supervisión
- Planificación del Ciclo de Vida
  - ▣ Ciclo de Vida
  - ▣ Algunos Modelos
  - ▣ Conclusiones
- EUP
  - ▣ Beneficios Clave
  - ▣ Actividades
  - ▣ Relación entre Core Workflows y Fases del Plan
- Conclusiones

Ingeniería de Software II

Preparación de Plan de Proyecto

53

#### Contenido

La clase se centrará sobre los pasos para la preparación del work-plan. Se debe tener una noción acerca del resto de las actividades que desarrollamos durante la creación del Plan.

## Conclusiones

- ⇒ Un Plan de Proyecto requiere gran cantidad de información.
- ⇒ No contamos con toda ella en el momento de la construcción.
- ⇒ No es posible hacer un seguimiento sin un plan de trabajo acorde.