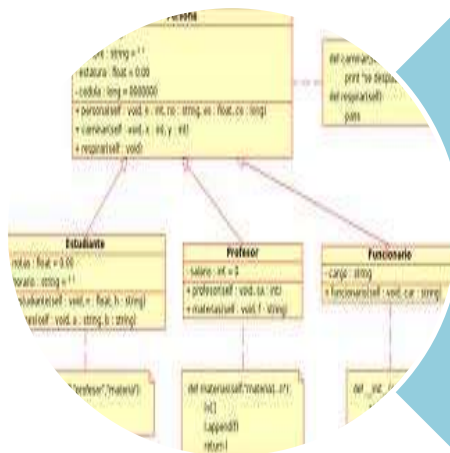


Ingeniería de Software I (semana 2)

EL PROCESO UNIFICADO

Agenda



El Proceso Unificado (RUP)

Síntomas de los Problemas de Desarrollo de Software

- ✓ Las necesidades del usuario o la empresa no se cumplen
- ✓ Los requerimientos cambian rápidamente
- ✓ Los módulos no se integran
- ✓ Difícil de mantener
- ✓ Descubrimiento tardío de fallas
- ✓ Baja calidad o experiencia del usuario final
- ✓ Desempeño deficiente bajo presión
- ✓ Falta de coordinación del esfuerzo en equipo
- ✓ Problemas de construcción y liberación



Rastrear los síntomas hasta sus Causas Raíces

Síntomas

Neces. incumplidas
Requer. cambiantes
Módulos no Integran
Difícil de mantener
Hallazgos tardíos
Calidad pobre
Desempeño pobre
Analistas sin info.
Construc. – Liberac.

Causas

Requer. insuficientes
Comunicación ambigua
Arquitectura frágil
Excesiva complejidad
Inconsistenc. no detect.
Pruebas pobres
Evaluación subjetiva
Desarrollo en cascada
Cambio no controlado
Automatización. insuficiente

Mejores Prácticas

Desarrollo Iterativo
Administrar los Requerim.
Modelado Visual (UML)
Arquitect. de Componentes
Administrar el Cambio
Verificar Calidad Continua

¿Qué es Rational Unified Process?

RUP es un proceso de desarrollo de Software:

- Forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo).

Objetivos:

- Asegurar la producción de software de calidad dentro de plazos y presupuesto predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones).



¿Qué es Rational Unified Process?

Es también un producto:

- Desarrollado y mantenido por Rational.
- Actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo con la experiencia.

Aumenta la productividad de los desarrolladores mediante acceso a:

- Base de conocimiento, plantillas y herramientas.

¿Qué es Rational Unified Process?

Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos.

RUP es una guía de cómo usar UML de la forma más efectiva.

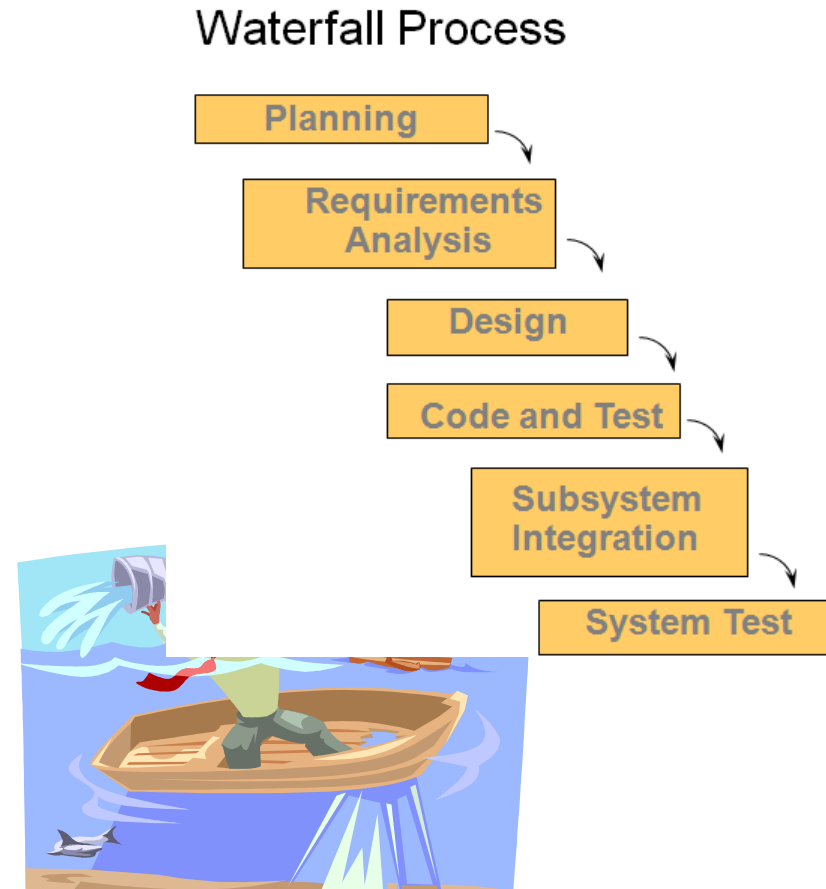
Existen herramientas de apoyo a todo el proceso:

- Modelado Visual, programación, pruebas, etc.

Características del desarrollo en cascada

Recordemos:

- Demora en la confirmación de la solución, con riesgos críticos
- Mide el progreso evaluando productos que no predicen bien el tiempo faltante
- Acumulación de trabajo de integración y prueba
- Impide el despliegue temprano de funcionalidades
- Con frecuencia trae iteraciones mayores no planificadas



P-R-D-C-I-T

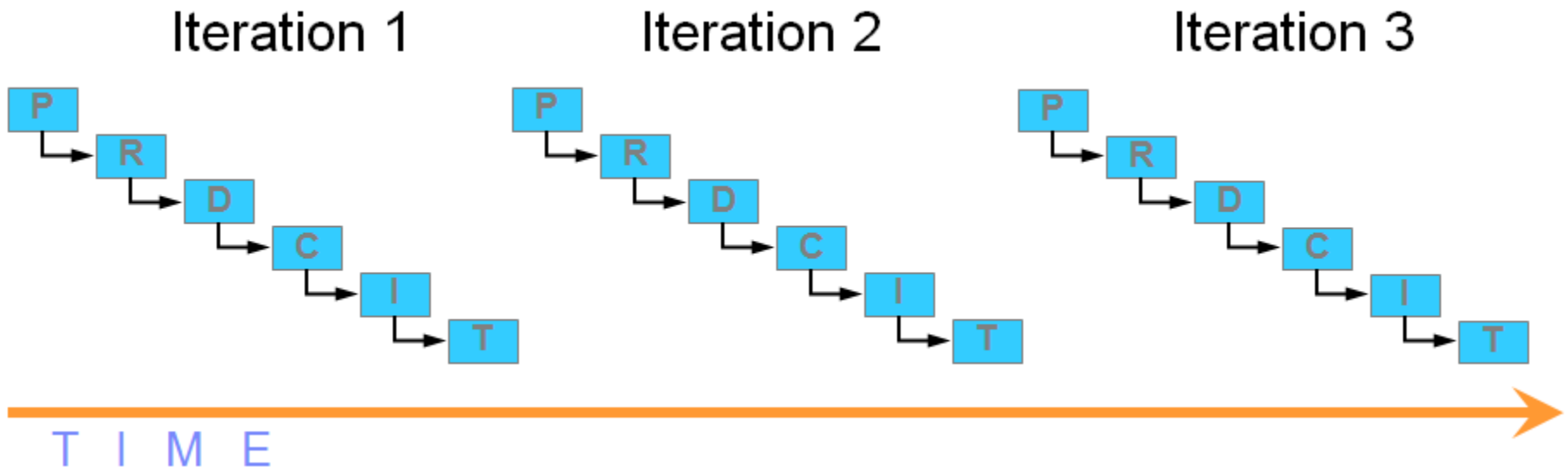
Las 6 mejores prácticas

- Desarrollo Iterativo
- Gestión de Requerimientos
- Arquitectura basada en Componentes
- Modelado Visual (UML)
- Verificar continuamente la calidad
- Administrar el cambio

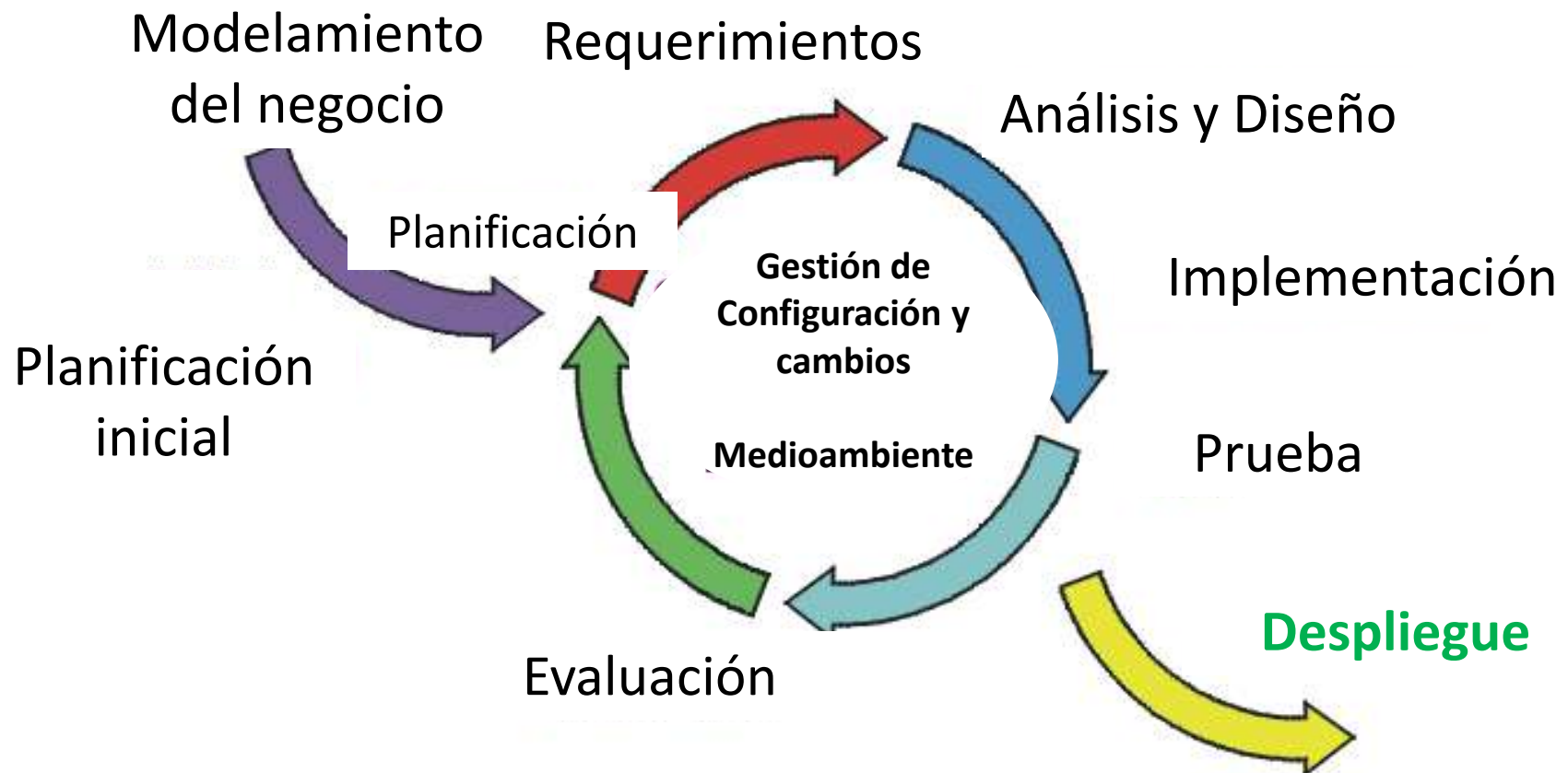


Práctica 1: Desarrollo Iterativo

- El software moderno es complejo y novedoso. No es realista usar un modelo lineal de desarrollo como el de cascada.
- Un proceso iterativo permite una comprensión creciente de los requisitos a la vez que se va haciendo crecer el sistema.
- RUP sigue un modelo iterativo que aborda las tareas más riesgosas primero.
- Se logra reducir los riesgos del proyecto y tener un subsistema ejecutable de forma temprana.

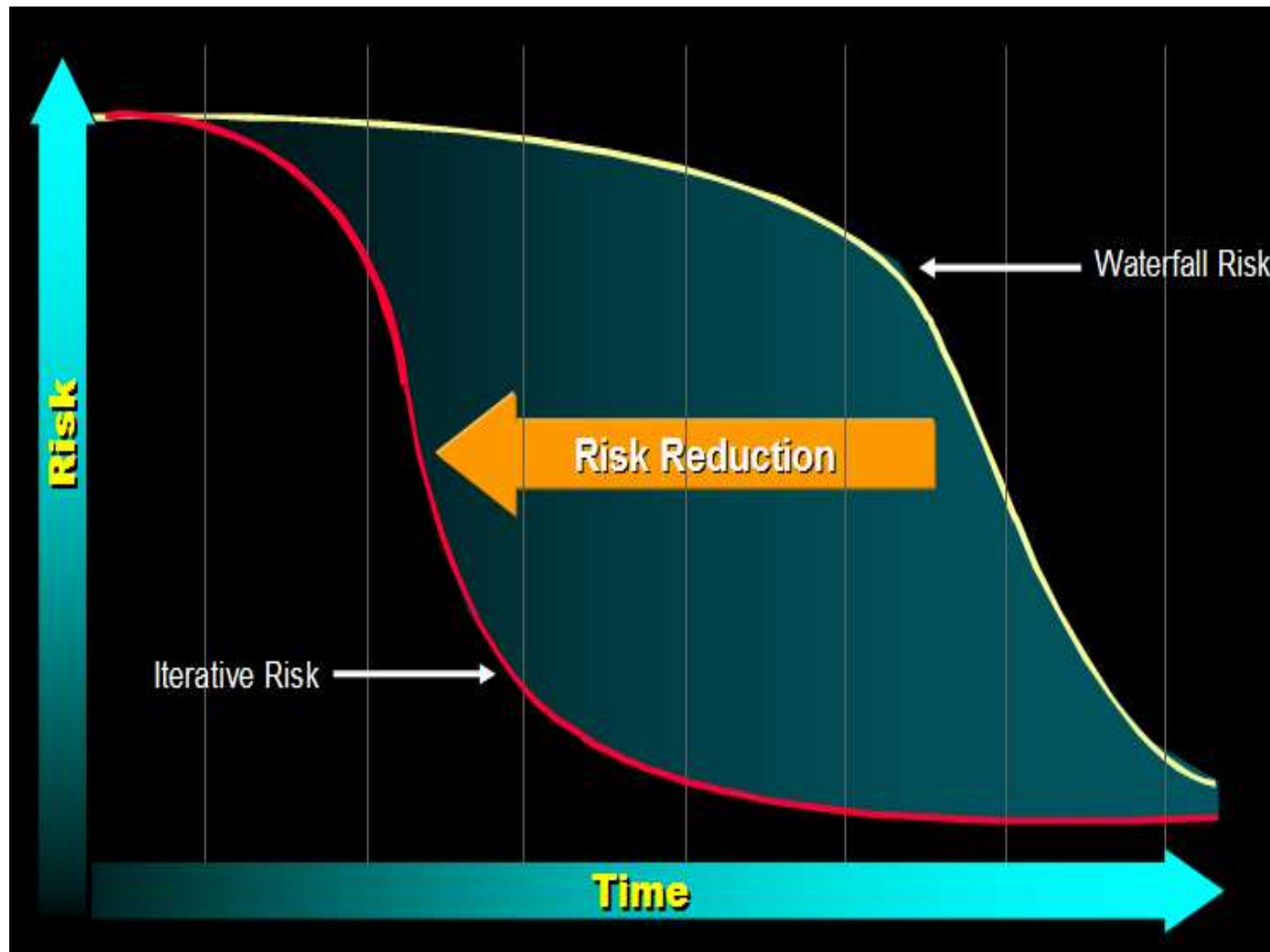


El desarrollo iterativo produce un ejecutable



Riesgo

Una meta importante en el desarrollo iterativo es la detección de errores a tiempo. Esto se reduce analizando, priorizando, y atacando los riesgos altos en cada iteración.



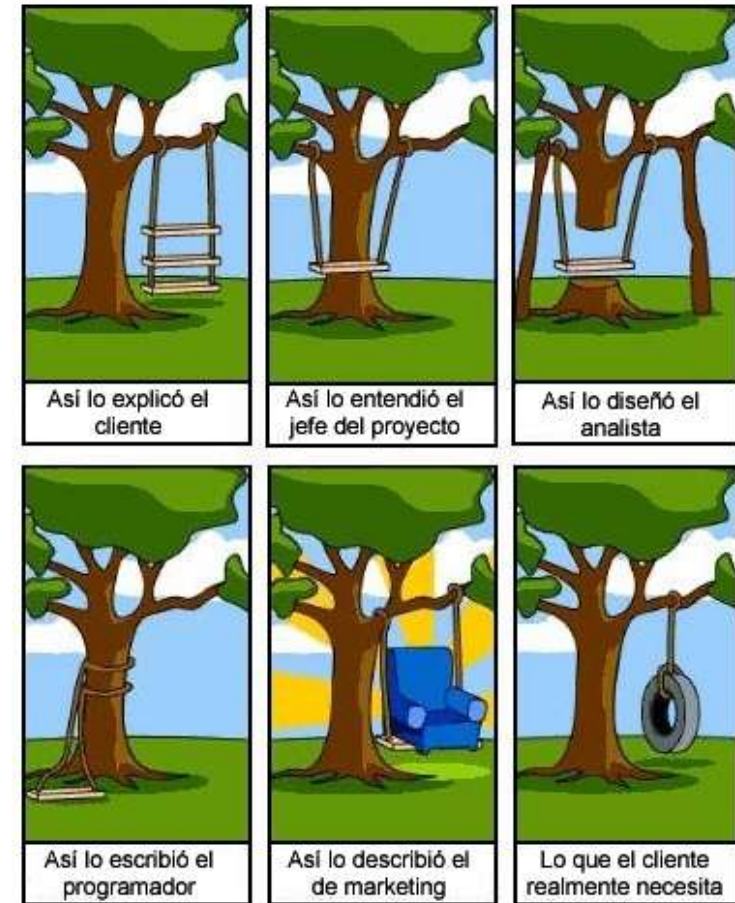
Práctica 2: Gestión de Requerimientos

- Los casos de uso y los escenarios indicados por el proceso han probado ser una buena forma de captar requisitos y guiar el diseño, la implementación y las pruebas.
- RUP describe cómo:
 - Obtener los requisitos (requerimientos)
 - Organizarlos
 - Documentar requisitos funcionales, no funcionales y restricciones
 - Rastrear y documentar decisiones
 - Captar y comunicar requisitos (requerimientos) del negocio



Gestión de Requerimientos

- Asegúrese que usted
 - Resuelve el problema correcto
 - Construye el sistema correcto
- Tomar un enfoque sistemático para:
 - Recolectar
 - Organizar
 - Documentar
 - Administrar



Aspectos de la Gestión de Requerimientos

- Analizar el problema
- Entender las necesidades del usuario
- Definir el sistema
- Administrar el alcance
- Refinar la definición del sistema
- Construir el sistema correcto

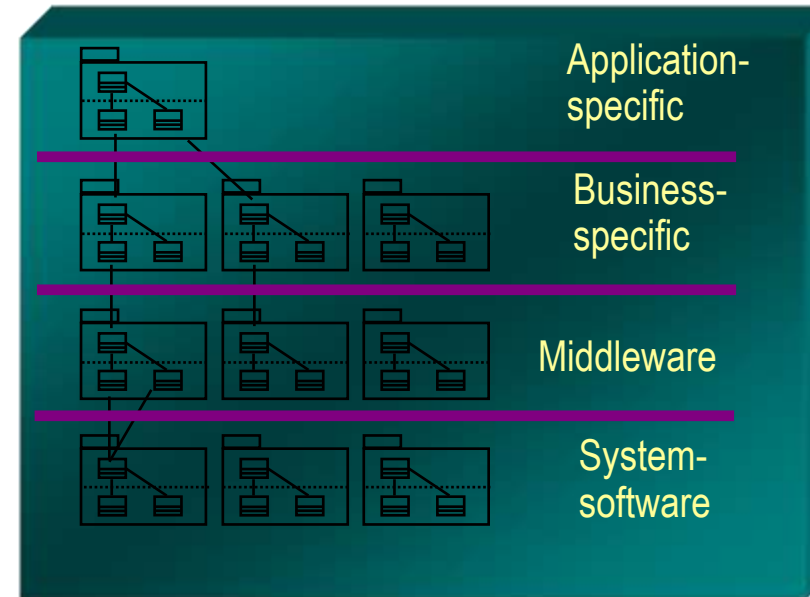


Práctica 3: Arquitectura basada en componentes

- El proceso se basa en diseñar tempranamente una arquitectura basada en ejecutable.
- La Arquitectura debe ser:
 - Flexible
 - Fácil de modificar
 - Intuitivamente comprensible
 - Promueve la reutilización de componentes
- RUP apoya el desarrollo basado en componentes, tanto nuevos como preexistentes.

Propósito de la arquitectura basada en componentes

- Base en la reutilización
 - Reutilización de componentes
 - Reutilización de arquitecturas
- Base en la gerencia del proyecto
 - Planeación
 - Asignar recursos
 - Entrega
- Control intelectual
 - Administrar la complejidad
 - Manejar la integridad



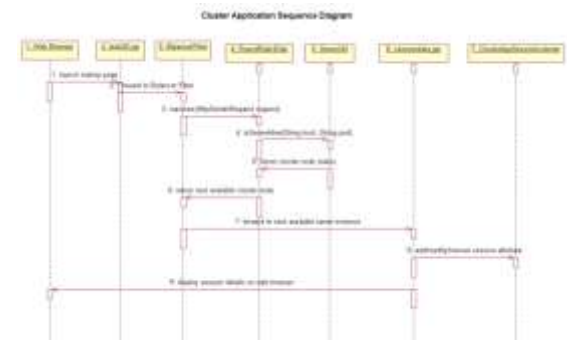
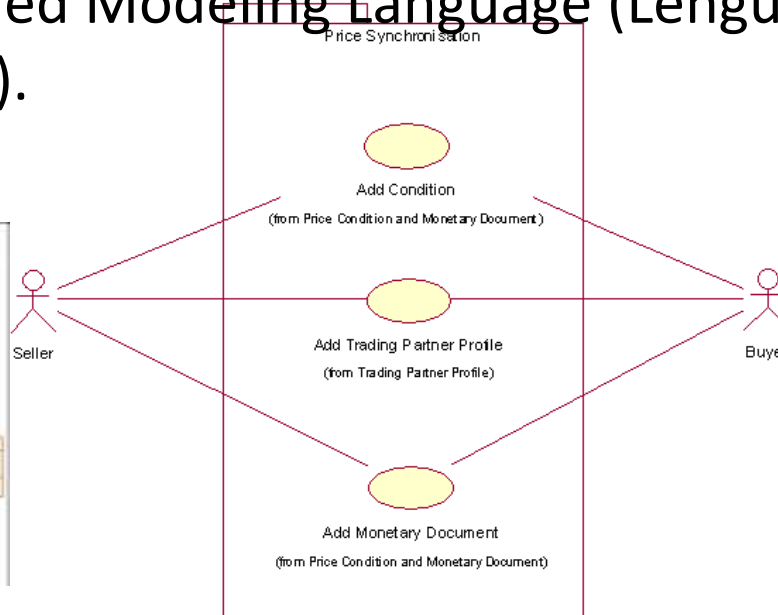
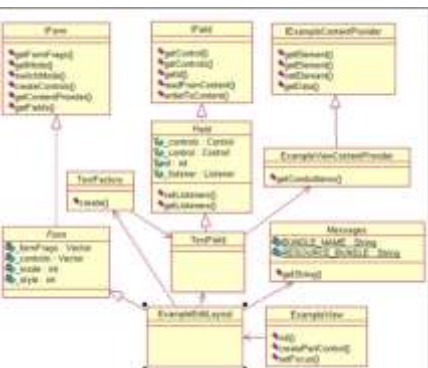
Práctica 4: Modelado Visual

- Modelado visual de la estructura y el comportamiento de la arquitectura y los componentes.
- Bloques de construcción:
 - Ocultan detalles
 - Permiten comunicación en el equipo de desarrollo
 - Permite analizar la consistencia:
 - Entre los componentes
 - Entre diseño e implementación
- UML es la base del modelado visual de RUP.

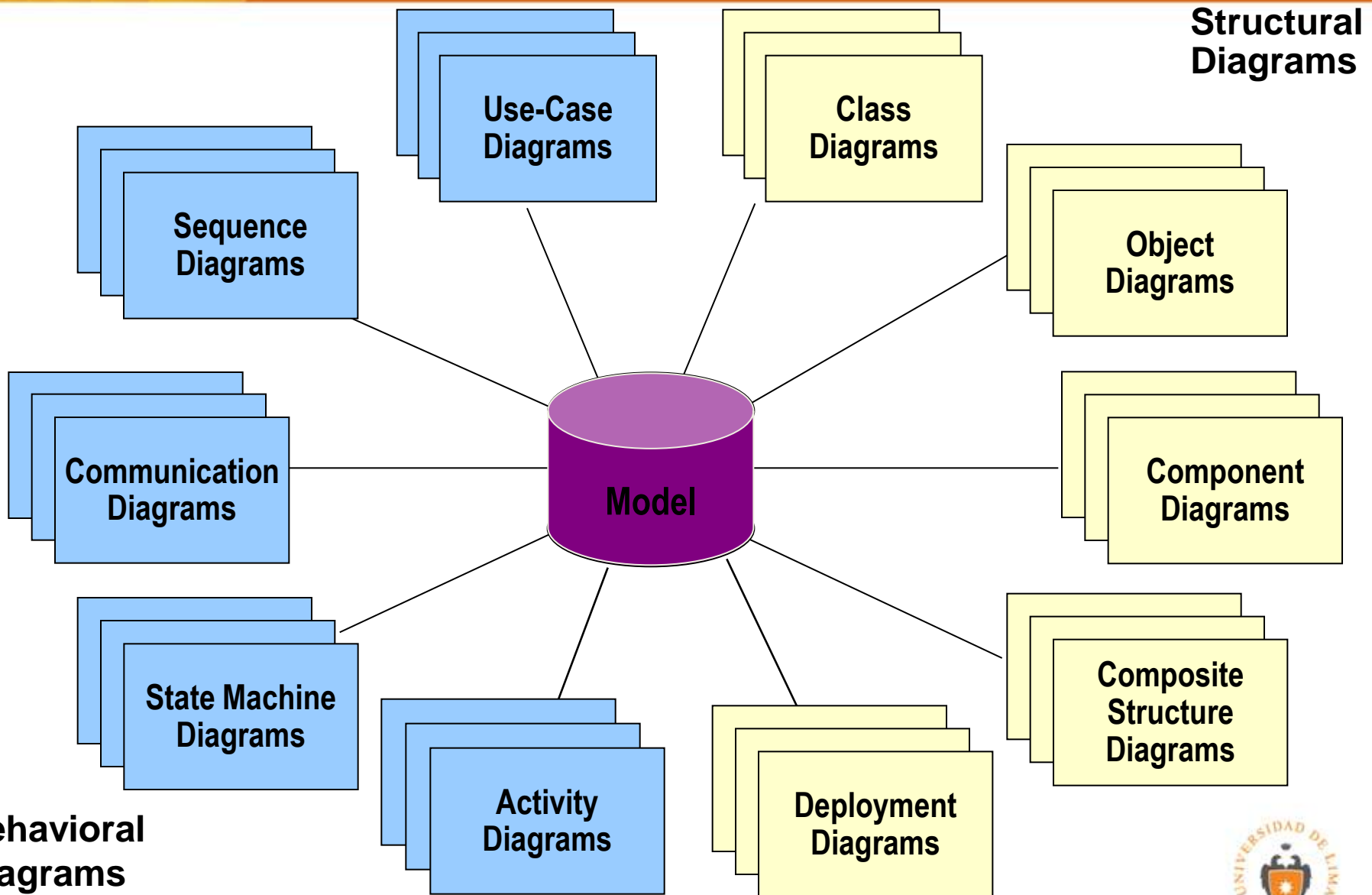


¿Por qué modelar visualmente?

- Capturar la estructura y el comportamiento
- Mostrar como los elementos del sistema calzan entre si
- Mantener consistentes el diseño y la implementación
- Ocultar o exponer los detalles según sea conveniente
- Promover una clara comunicación
 - UML: Unified Modeling Language (Lenguaje Unificado de Modelado).



Modelado Visual con el Unified Modeling Language



Práctica 5: Verificar continuamente la calidad

- No sólo la funcionalidad es esencial, también el rendimiento y la confiabilidad.
- RUP ayuda a planificar, diseñar, implementar, ejecutar y evaluar pruebas que verifiquen estas cualidades.
- El **aseguramiento de la calidad** es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente.



Verifique continuamente la calidad del software

- Los problemas de software son de 100 a 1000 más costosos de encontrar y reparar después de la liberación.
 - Costos de reparación
 - Costos de oportunidades perdidas
 - Costo de clientes perdidos



Dimensiones de las Pruebas de Calidad

- Facilidad de Uso (Usabilidad)
 - Probar la aplicación desde la perspectiva de la conveniencia del usuario final.
- Confiabilidad (Fiabilidad)
 - Probar que la aplicación se comporta predecible y consistentemente.
- Desempeño (Eficiencia)
 - Probar la respuesta en línea bajo cargas promedio y máximas.



Dimensiones de las Pruebas de Calidad

- Funcionalidad (Fiabilidad)
 - Probar la habilidad para mantener y soportar la aplicación bajo uso en producción.
- Compatibilidad (Portabilidad)
 - Comprobar el soporte a diferentes entornos, ser extensible y permitir portabilidad.

Práctica 6: Control de Cambios

- Los cambios son inevitables, pero es necesario **evaluar** si éstos son **necesarios** y rastrear su **impacto**.
- RUP indica como controlar, rastrear y monitorear los cambios dentro del proceso iterativo de desarrollo.



¿Qué se desea controlar?

- Cambios que se incluyan durante el desarrollo iterativo
- Espacios de trabajo seguros para cada desarrollador
- Administración automatizada de integración y construcción
- Desarrollo en paralelo

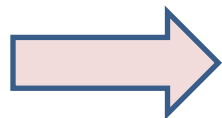


Aspectos de un sistema de control de configuraciones

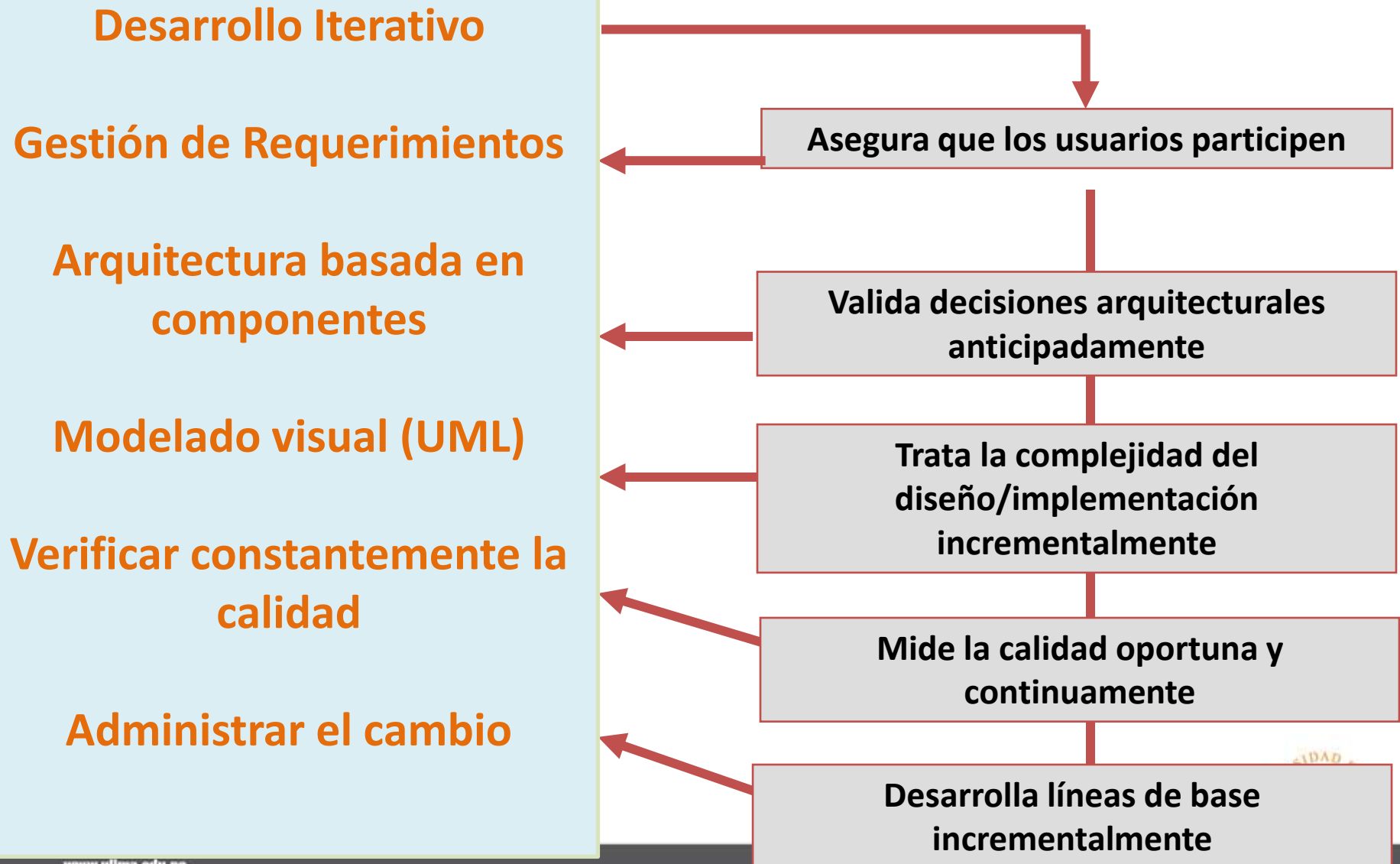
- Gestión de requerimientos de cambio (Change request management)
- Informes de estado de configuración (Configuration status reporting)
- Gestión de Configuraciones (Configuration management) (CM)
- Selección de versiones (Version selection)
- Manufactura de software (Software manufacture)

La Administración del Cambio Unificado involucra

- Administrar a lo largo del ciclo de desarrollo
 - Sistema
 - Gerencia del proyecto
- Gerencia basada en actividades
 - Tareas
 - Defectos
 - Mejoras
- Haciendo seguimiento al progreso
 - Gráficos
 - Informes

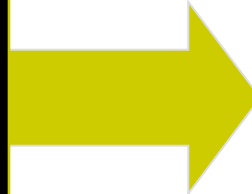


Relación entre las Mejores Prácticas



RUP IMPLEMENTA LAS MEJORES PRÁCTICAS

Mejores prácticas



Desarrollo Iterativo

Gestión de Requerimientos

**Arquitectura basada en
componentes**

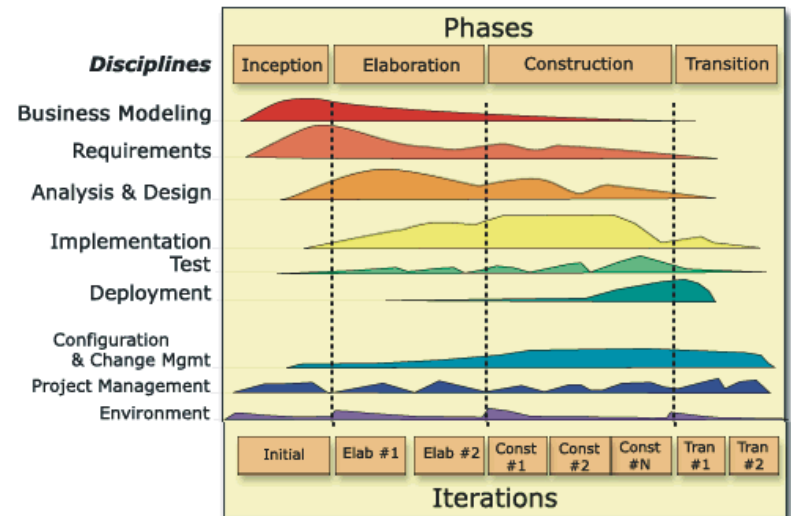
Modelado visual (UML)

**Verificar constantemente la
calidad**

Administrar el cambio

Logra las mejores prácticas

- Enfoque iterativo
- Guía para las actividades y artefactos
- Proceso dirigido hacia la arquitectura
- Casos de uso que manejan el diseño e implementación
- Modelos que abstraen el sistema



Un equipo basado en la definición del proceso

Un proceso define **QUIÉN** está haciendo **QUÉ**, **CUÁNDO Y CÓMO** con la finalidad de alcanzar una cierta meta.



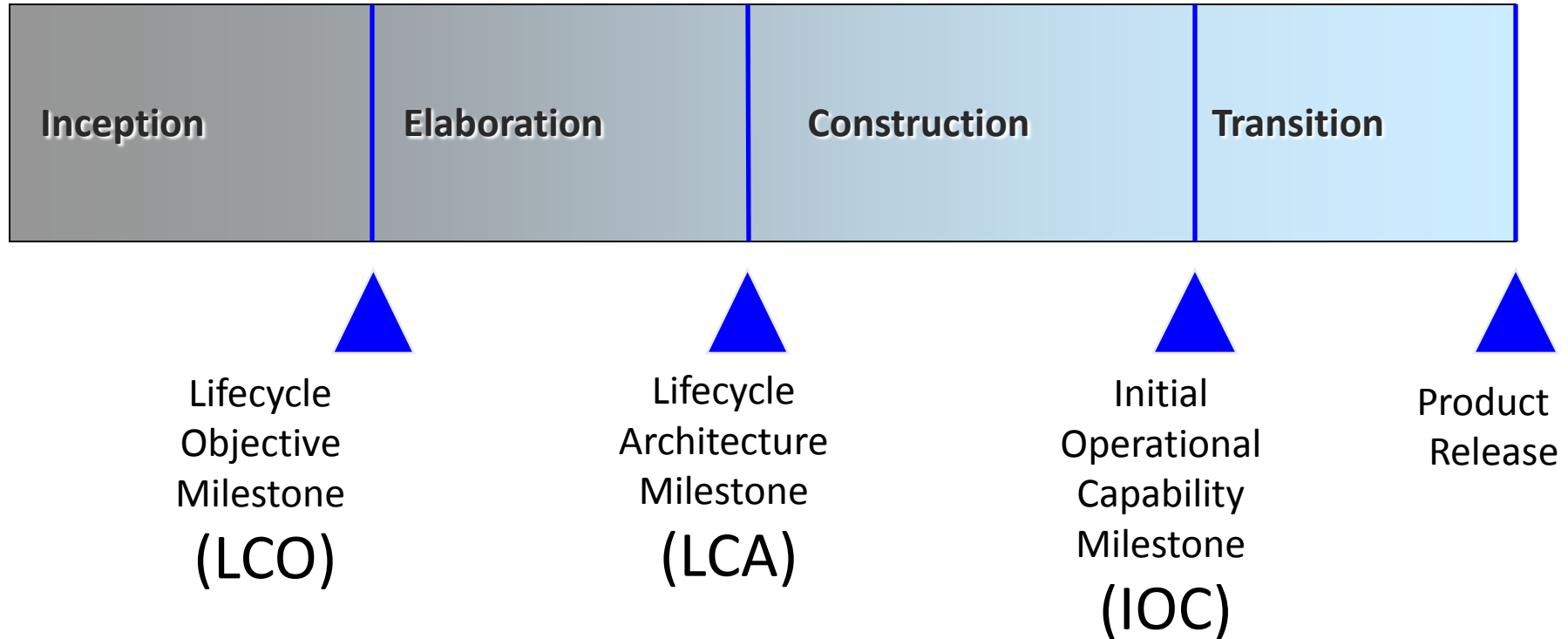
Estructura del Proceso – Fases del ciclo de vida

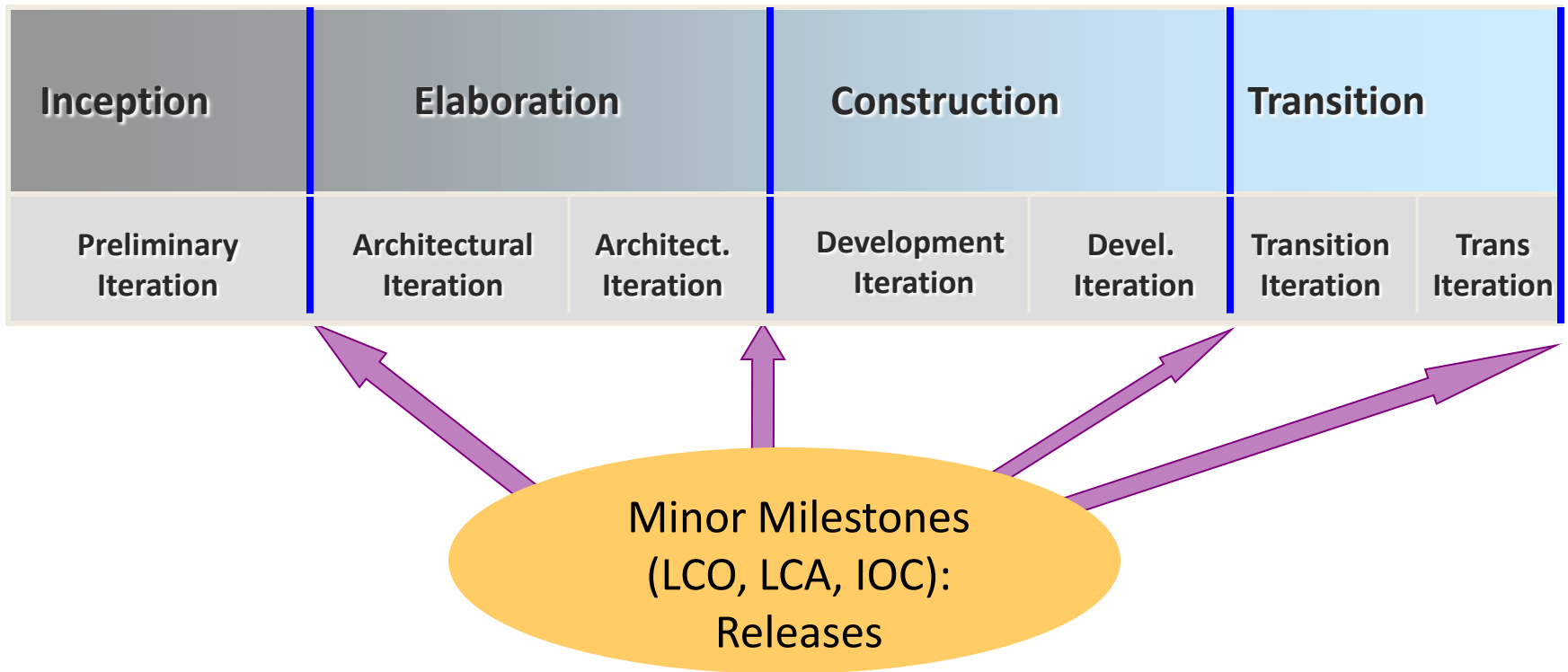
Inception (Inicio)	Elaboration (Elaboración)	Construction (Construcción)	Transition (Transición)
Handle risks related to business case. (financial worthiness of the project)	Handle risks related to the technical risks of the project.	Handle risks related to “getting the mass of work done.”	Handle risks related to logistics of deploying the application to its user base.

Time



Hitos principales de medición

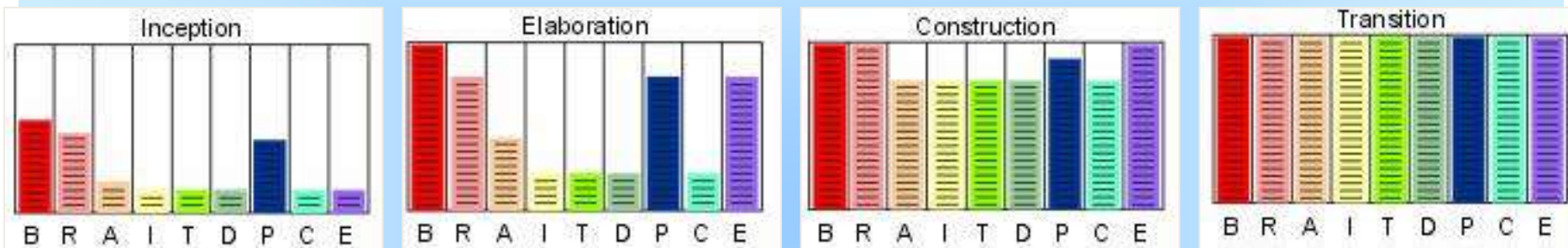




Una **iteración** es una secuencia distinta de actividades basada en un plan establecido y criterios de evaluación, obteniéndose una versión ejecutable (interna o externa)

Artefactos a través del ciclo de vida

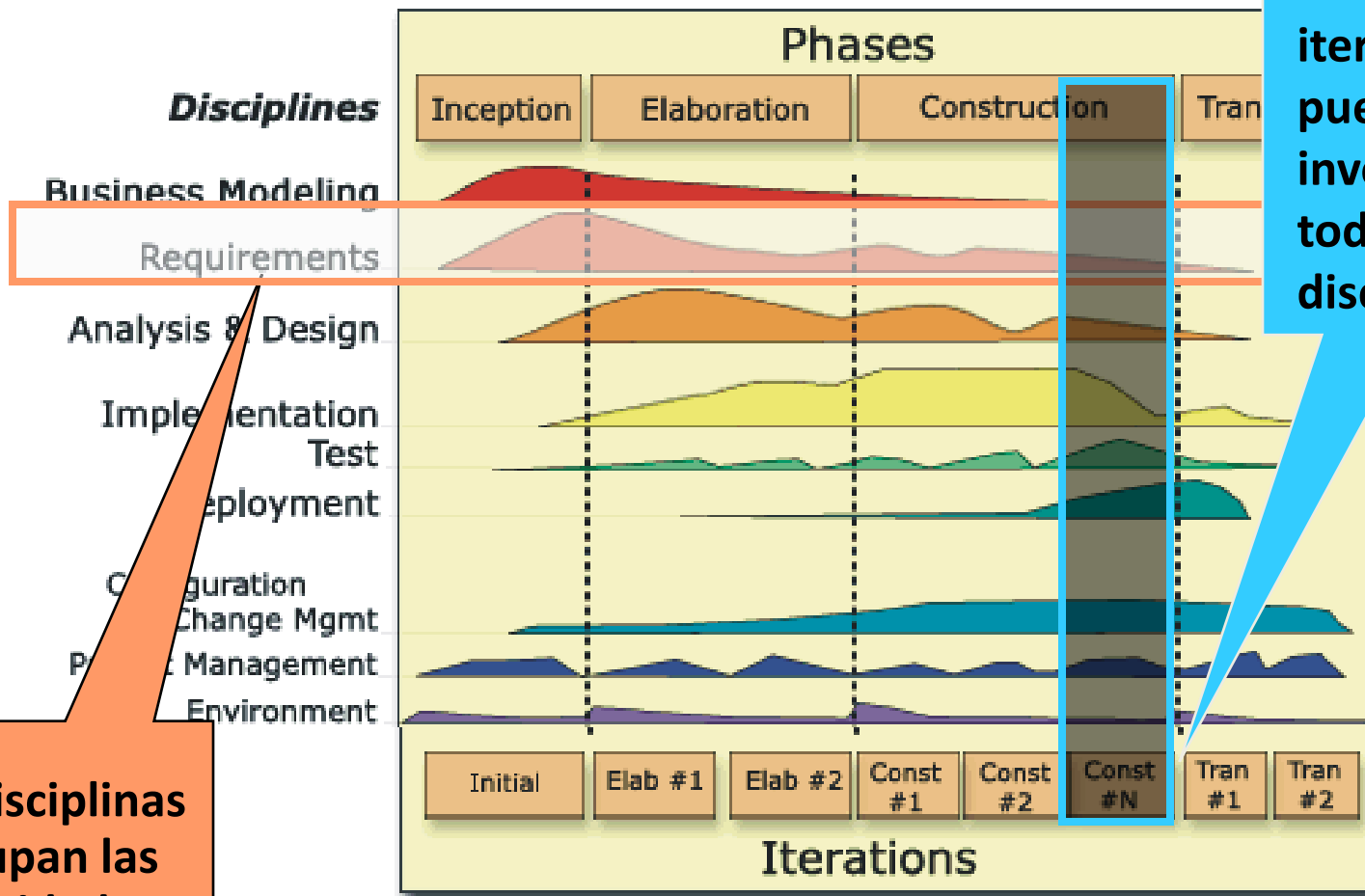
Artefactos del proyecto son desarrollados iterativamente.



B : Business Modeling Set
R : Requirements Set
A : Analysis & Design Set
I : Implementation Set
T : Test Set

D : Deployment Set
P : Project Management Set
C : Configuration & Change Management Set
E : Environment Set

El enfoque iterativo



En una iteración, se puede involucrar todas las disciplinas.

Las disciplinas agrupan las actividades lógicamente

Disciplines

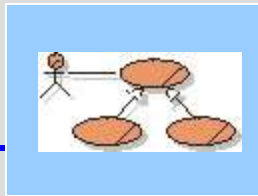
Business
Modeling

Requirements

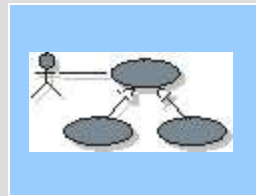
Analysis &
Design

Implementation

Models

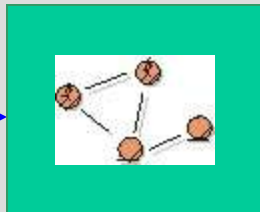


**Business Use-
Case Model**



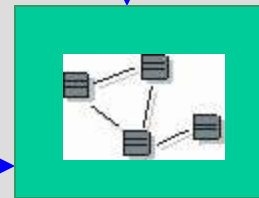
**Use-Case
Model**

Realized By



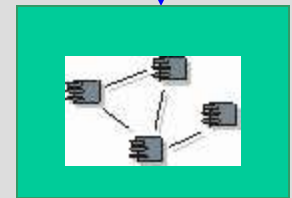
**Business
Object Model**

Automated By



**Design
Model**

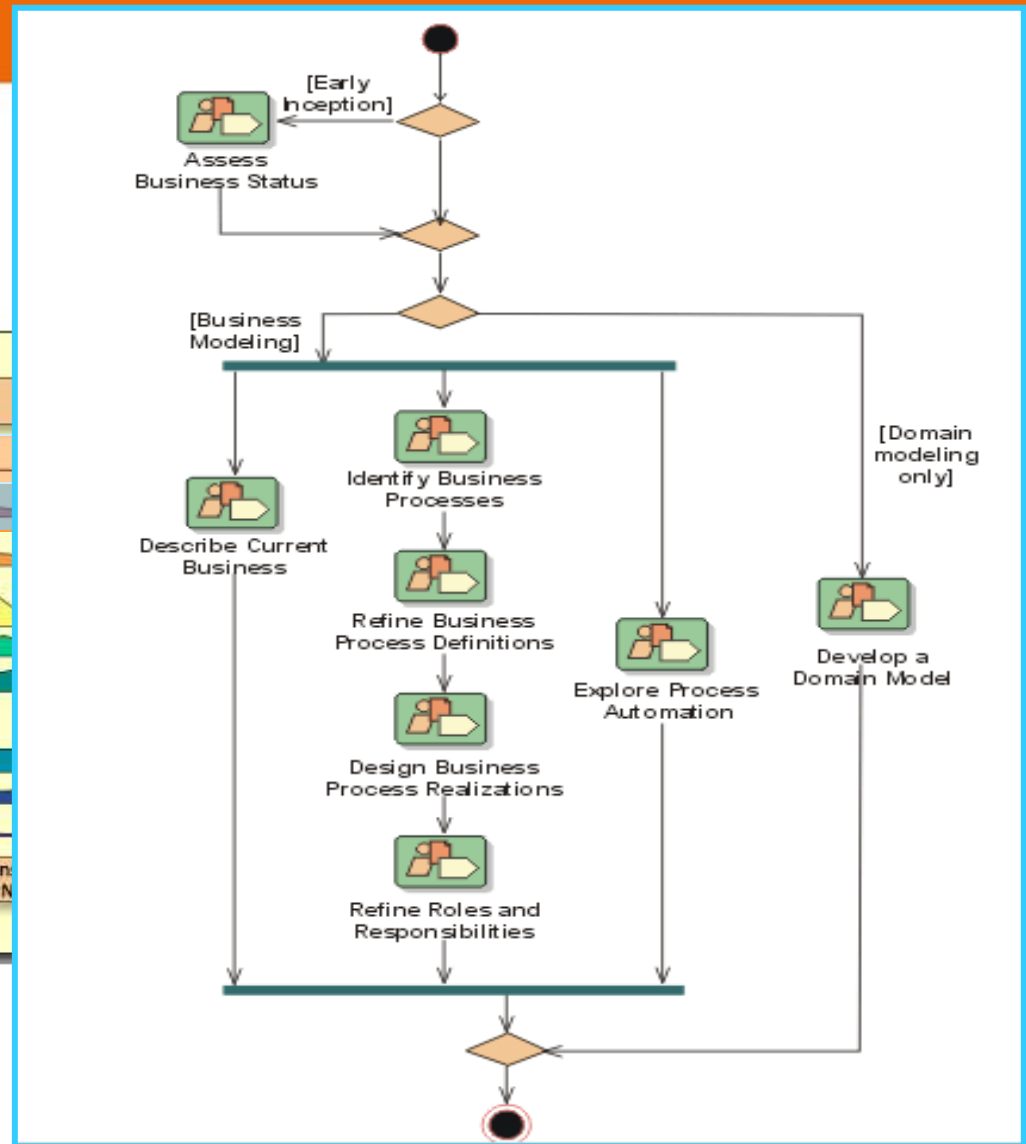
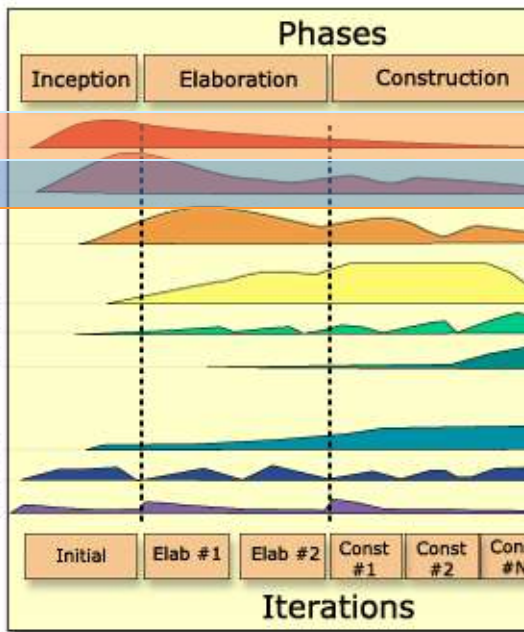
Realized By



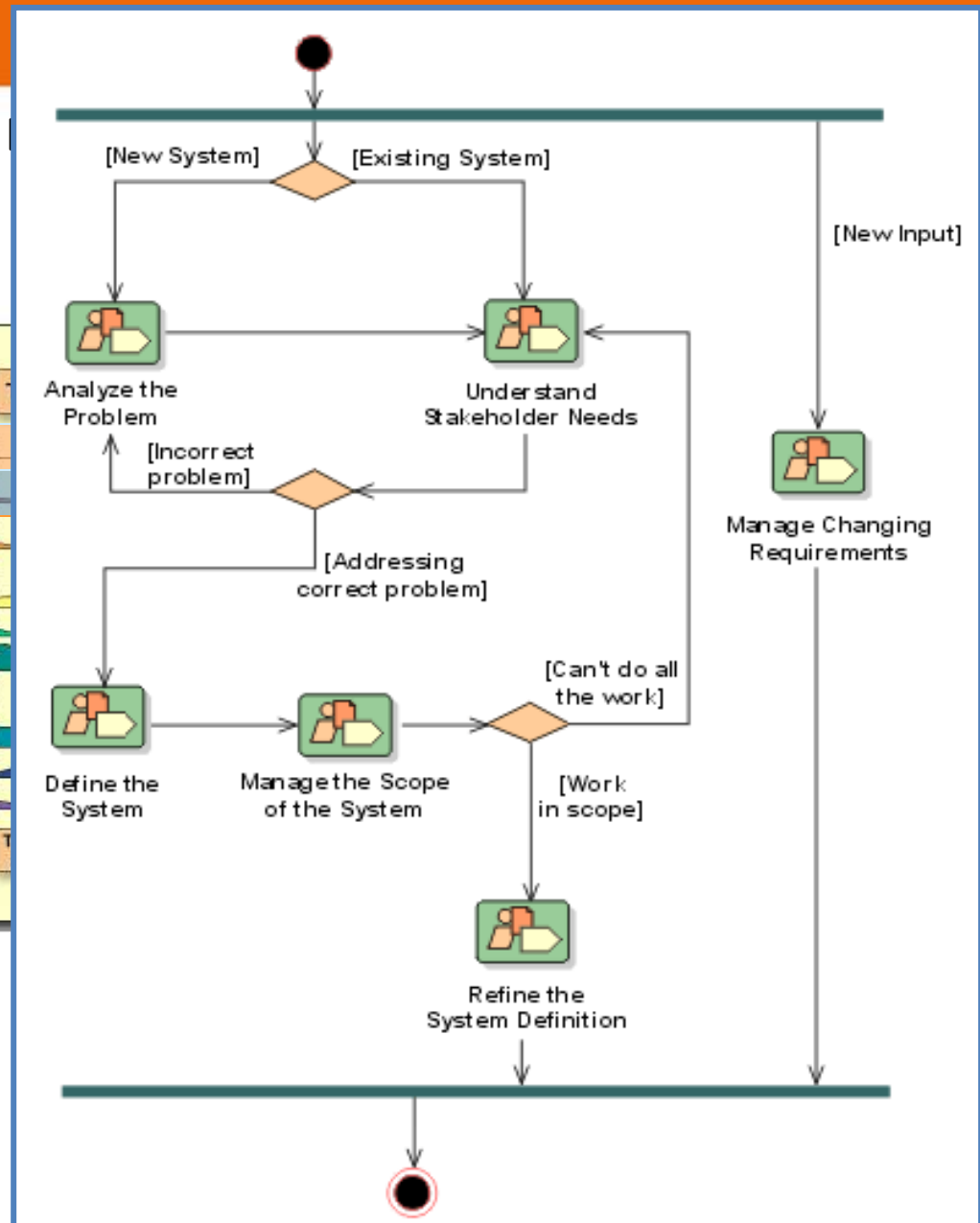
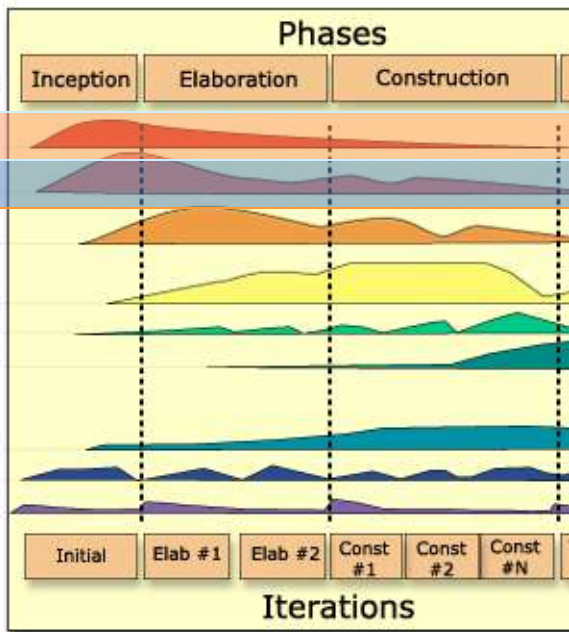
**Implementation
Model**

Implemented By

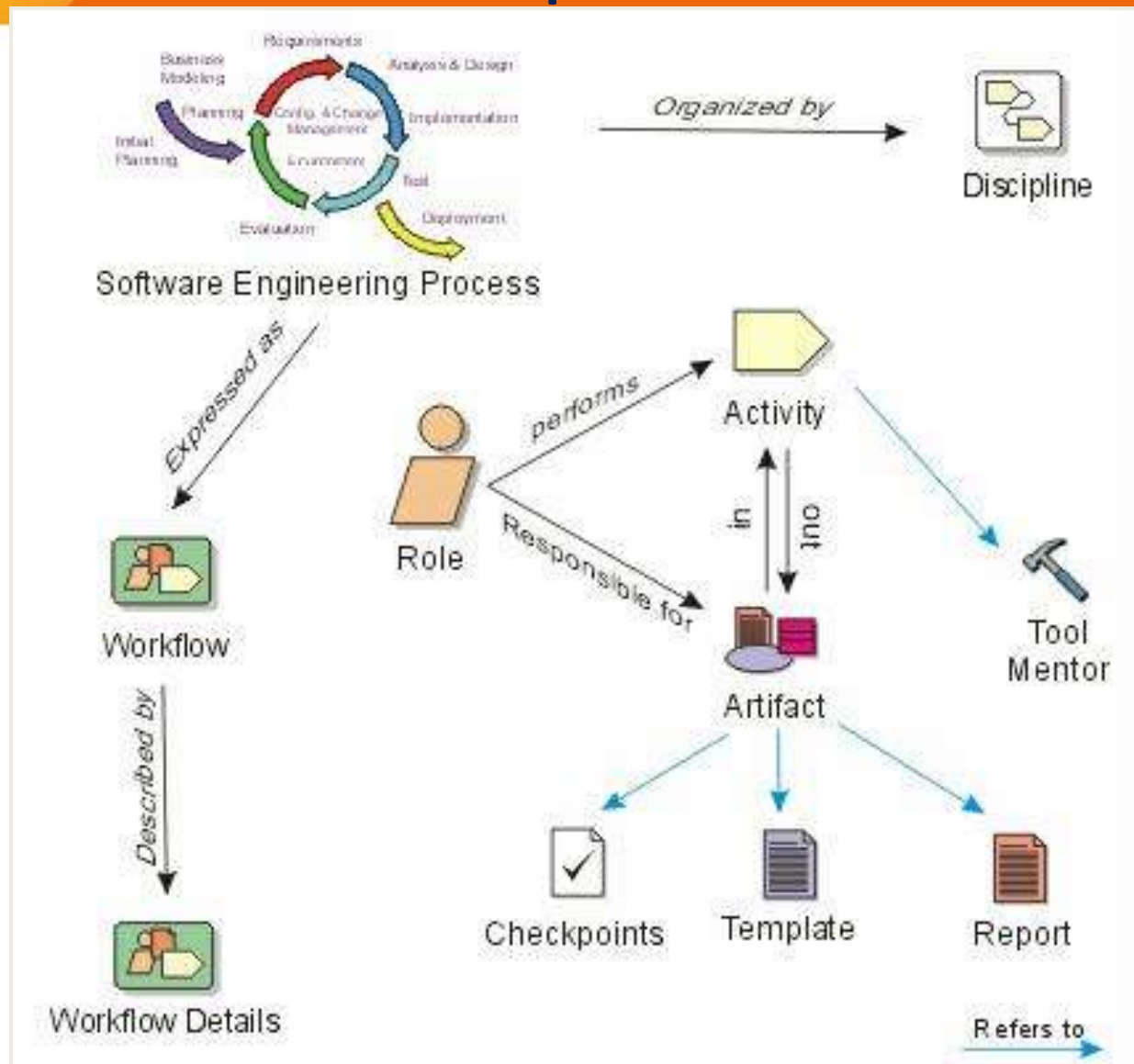
Guía de disciplinas de desarrollo iterativo



Guía de disciplinas de desarrollo iterativo



Resumen de conceptos de RUP



Resumen

- Las mejores prácticas guían a la ingeniería de software a identificar las causas raíz.
- Las mejores prácticas se relacionan entre sí.
- Los procesos guían al equipo a quién hace qué, cuándo y cómo.
- RUP es un medio para lograr la aplicación de las mejores prácticas.

Referencias

- <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesRUP.pdf>
- <http://www.redbooks.ibm.com/redbooks/pdfs/sg247362.pdf>

¿Consultas?

