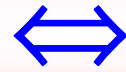


Gerencia de Proyectos



Proceso de Software

Victor Manuel Toro C.

VictorToro@cincosoft.com

CincoSOFT Ltda.

Compañía de Ingenieros Constructores de Software

Tel. (+57)(1) 6230180 * Fax (+57)(1) 2566774

Carrera 15 # 80-48 (ofc. 402)

Apartado Aéreo 350762

Bogotá - Colombia

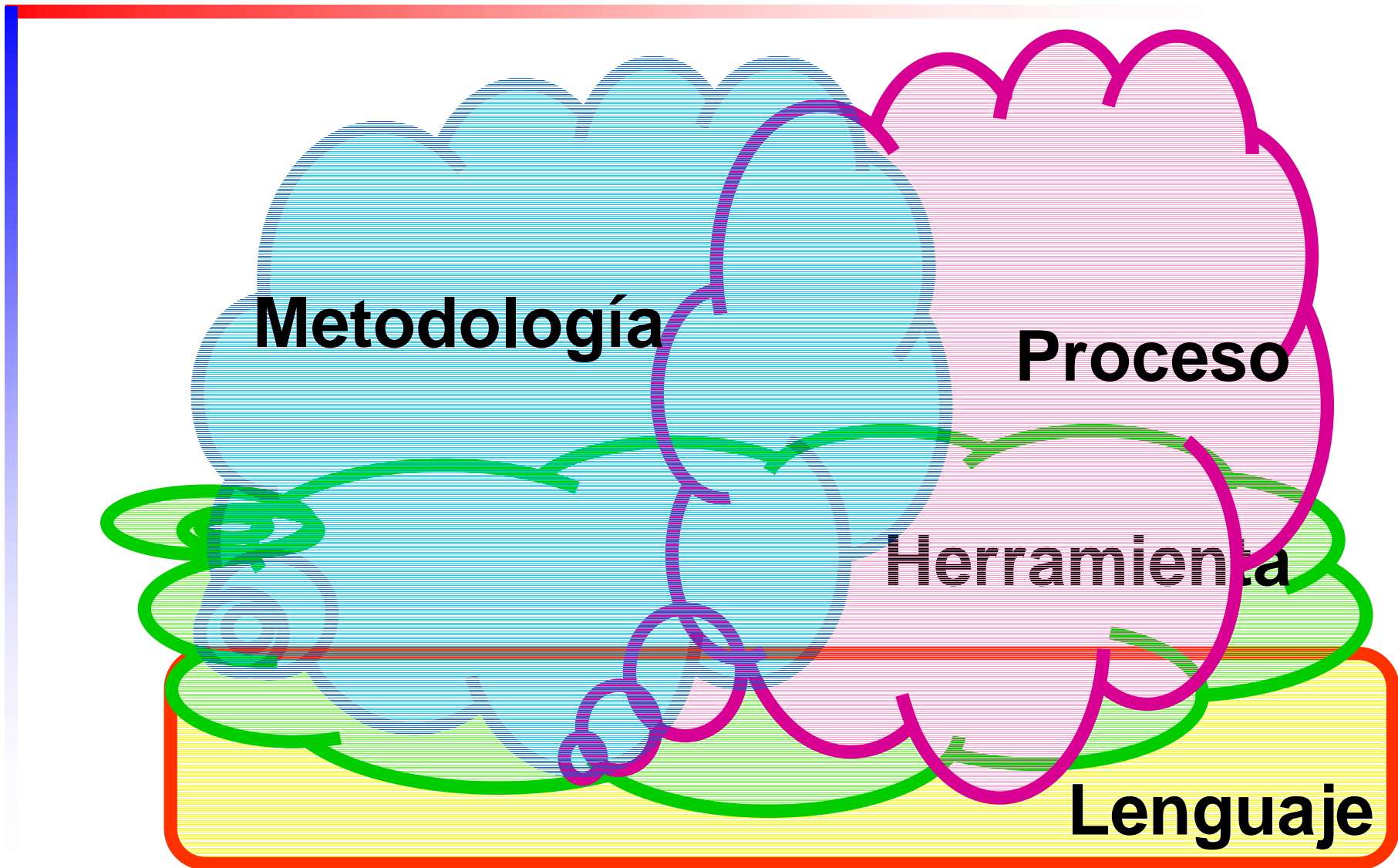
<http://www.cincosoft.com>

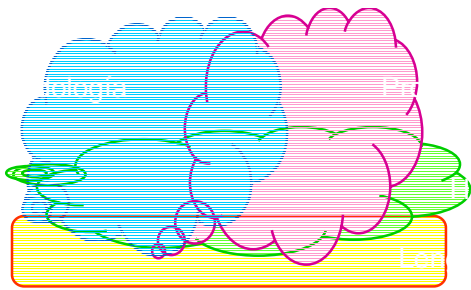




Conceptos Preliminares

Cuatro conceptos relacionados pero distintos:

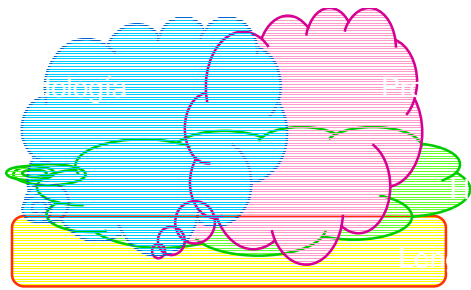




Aclarando algunos conceptos: **Lenguaje**

- **Lenguaje** (de programación, especificación, ...):
 - Sintaxis (elementos del lenguaje)
 - + Gramática (reglas para combinar dichos elementos)
 - + Semántica (significado de las frases del lenguaje).
- **Ejemplos de Lenguajes:**
 - Programación: Java, C, VisualBasic, RPG, Cobol, C#, ...
 - Análisis/Diseño:
 - ➔ Diagramas Entidad-Relación, Diagramas de Flujo de Datos DFD, ...
 - Especificación: Z, B, VDM, ...
 - Ingeniería de Software: UML





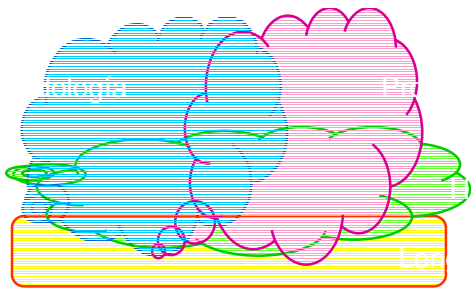
Aclarando algunos conceptos: **Herramienta**

- **Herramienta:**

- Editor de un(os) lenguaje(s)
- + Librería de elementos reutilizables
- + Ayudas “inteligentes”
- + Compilador ó Traductor hacia/desde otro(s) lenguaje(s)

- **Ejemplos de Herramientas:**

- JBuilder, JDeveloper, JEdit, Forté, ...
- VisualStudio.NET
- PowerBuilder, OracleForms, Designer-2000,...
- ...



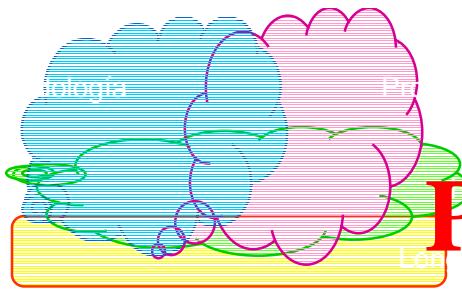
Aclarando algunos conceptos: **Metodología**

● **Metodología:**

- Lenguaje
- + forma de plantear problemas
- + forma de construir poco a poco una solución
- + forma de verificar que una solución es correcta
- + conjunto de patrones (paradigmas deseables)

● **Ejemplos de Metodologías:**

- Programación Estructurada {Pre y Post condiciones, invariantes, ...}
- Metodología Entidad-Relación para diseño de B. de D.
- Metodología Orientada a Objetos
- ...



Aclarando algunos conceptos:

Proceso de desarrollo de Software

● Proceso de desarrollo de Software:

- Una secuencia de actividades
- + una asignación de responsabilidades
- + un conjunto de **Entregables**
(documentos, código, pantallas, manuales, ...) estandarizados
- + dinámica de interacción entre los miembros del grupo
- + un conjunto de herramientas y una forma de usarlas
- + ...

que usa un grupo para el desarrollo de un sistema de software

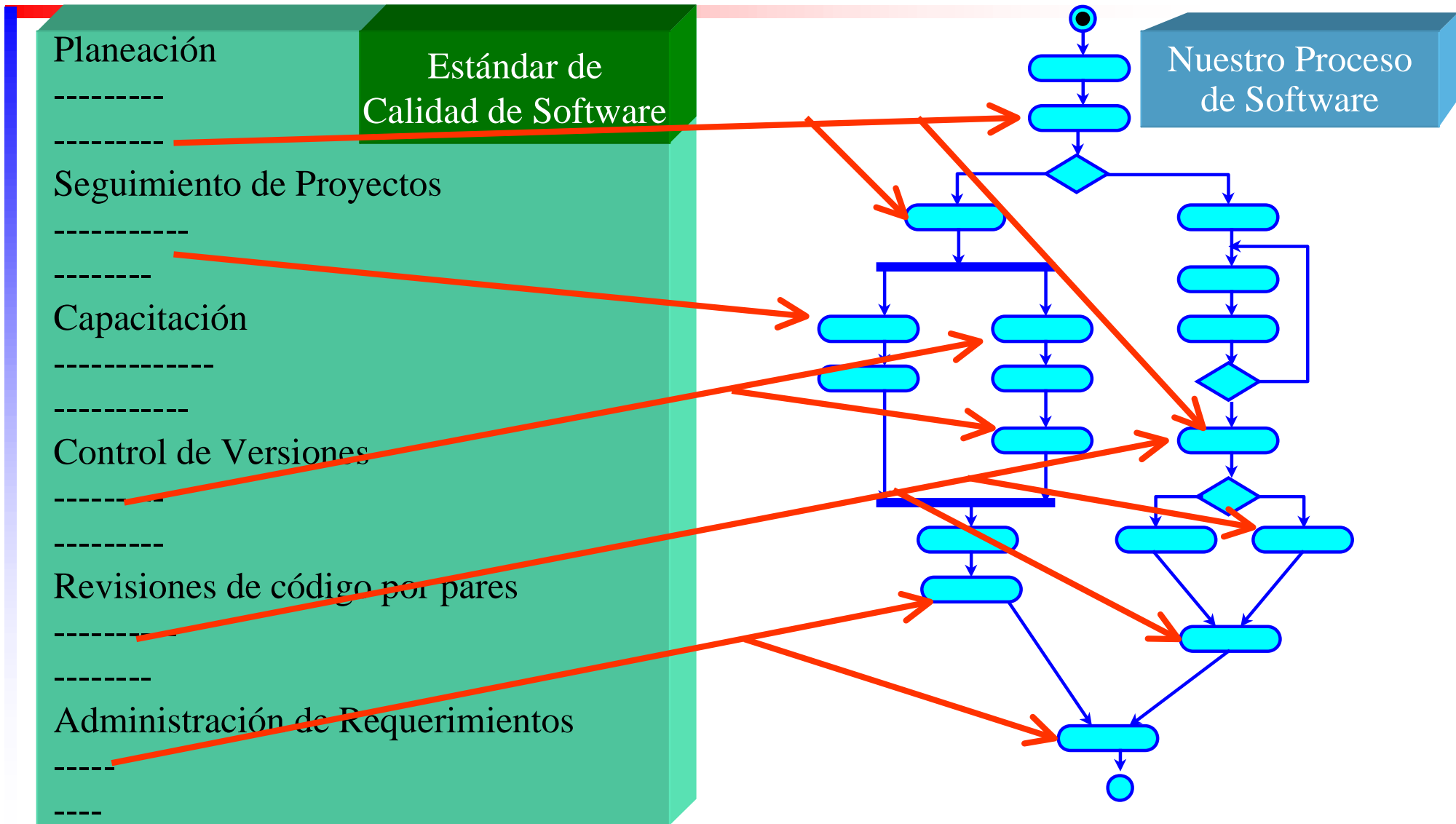
Madurez de un Proceso de Desarrollo


- Un Proceso Desarrollo de Software es **MADURO** en la medida que:
 - Está escrito
 - Es usado por TODOS los miembros del grupo
 - Cubre todas las etapas del desarrollo
 - Es efectivo
 - Es eficiente
 - Es repetible, medible, optimizable, ...
- Ver *Software Process Capability Maturity Model CMM*
[<http://www.sei.cmu.edu>]

¿ Qué es una compañía de software certificada ?

- Explicita su proceso de desarrollo de software
- Documenta su proceso de desarrollo (por escrito)
- Estandariza sus entregables (i.e., crea plantillas)
- Sintoniza sus herramientas para apoyar el proceso
- ...
- Incorpora en su proceso de software las recomendaciones de algún Estándar de Calidad (CMM, ISO)
- Contrata Inspectores autorizados para que revisen, y eventualmente certifiquen, que su proceso de desarrollo cumple con el Estándar de Calidad.

¿Qué es implantar un estándar de Calidad de Software?





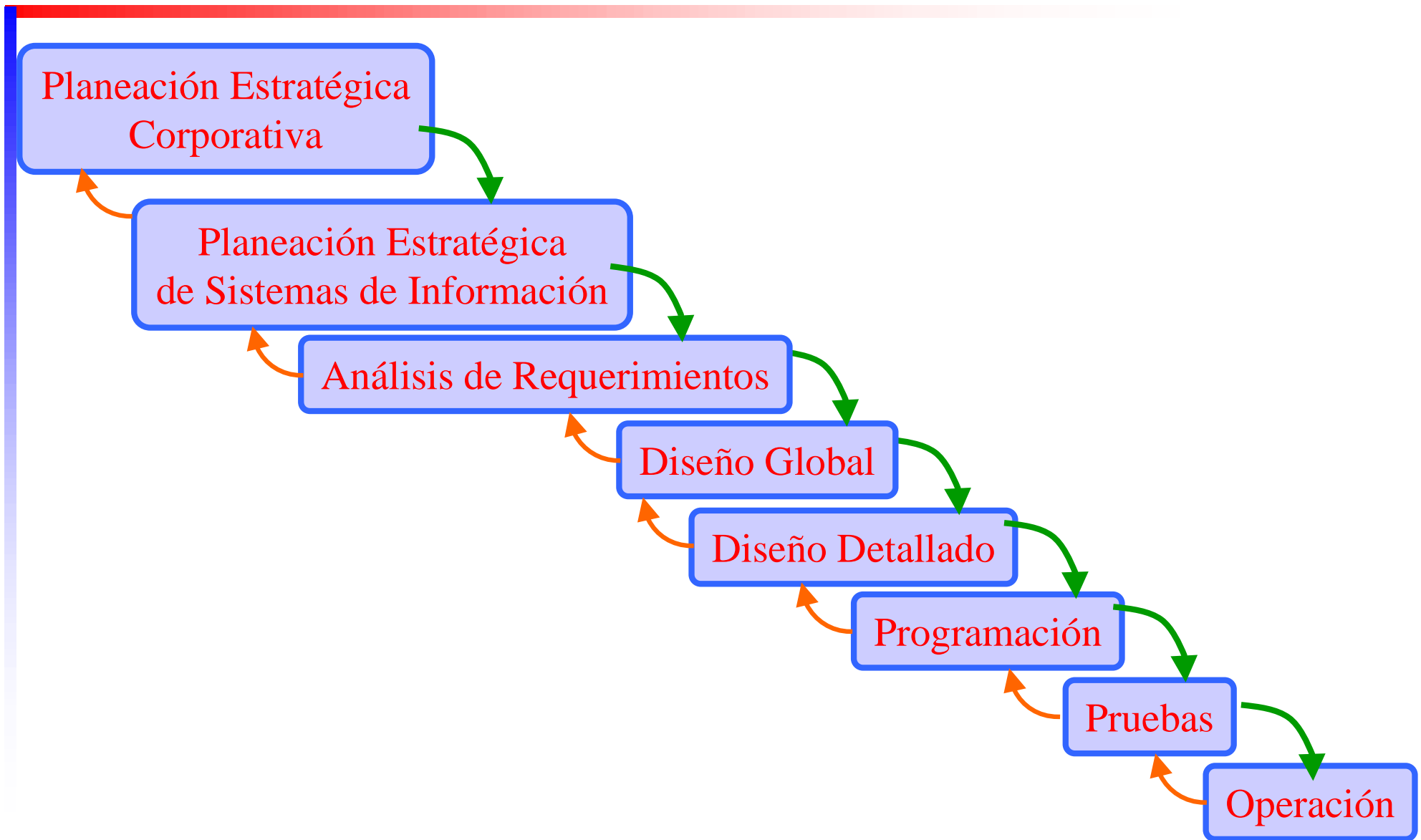
Evolución de los Procesos de Desarrollo de Software

¿Cómo nace un Proceso de Desarrollo de Software ?

Nace de la visión/enfoque que se tenga sobre:

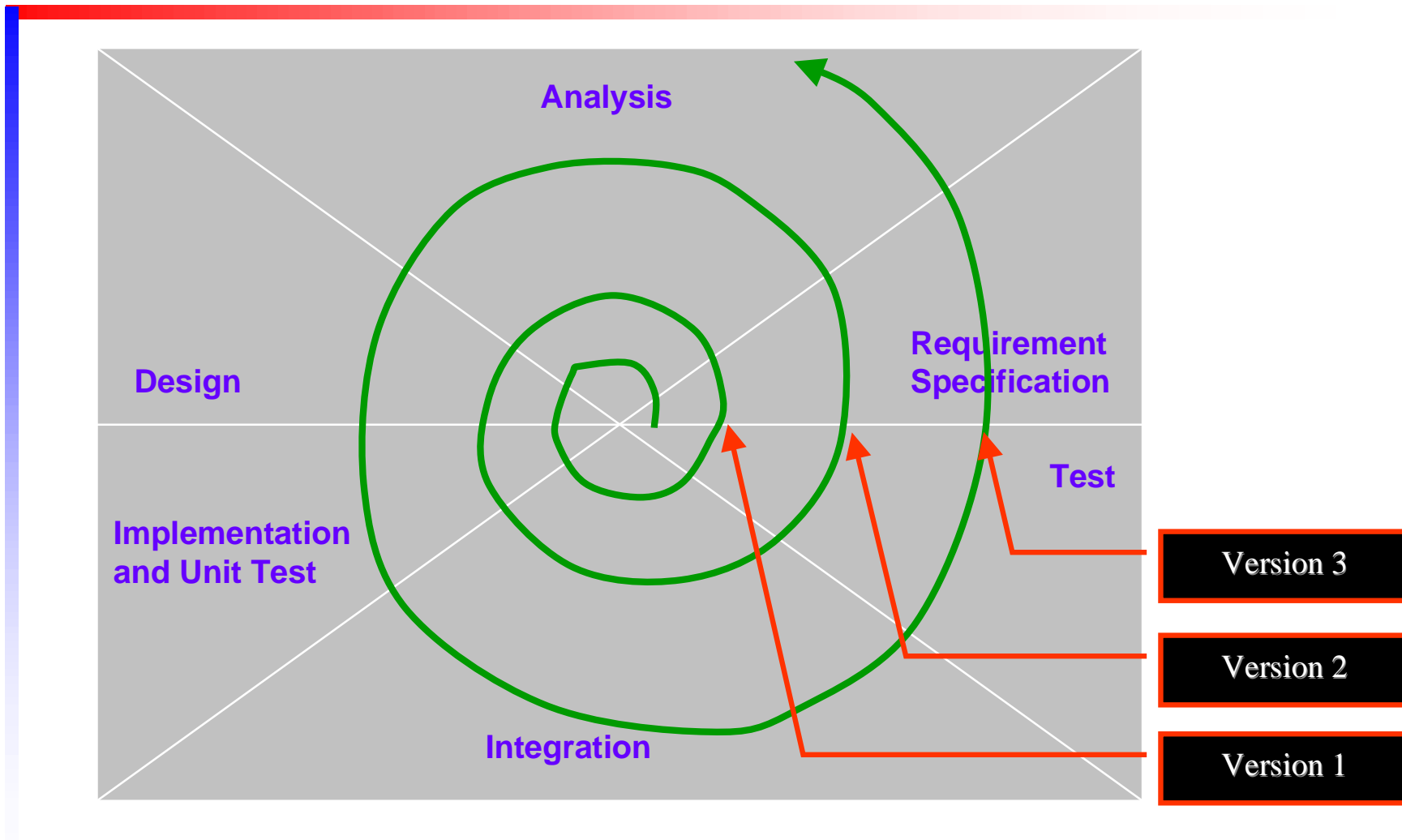
- Cómo se debe producir el software
- Cómo se debe entregar el software

Desarrollo por etapas secuenciales: “Modelo de Cascada”



Desarrollo iterativo:

“Modelo en Espiral” (B.W. Boehm)



“A Spiral Model of Software Development and Enhancement”

Tutorial: Software Engineering Project Management: Computer Society of the IEEE, 1988

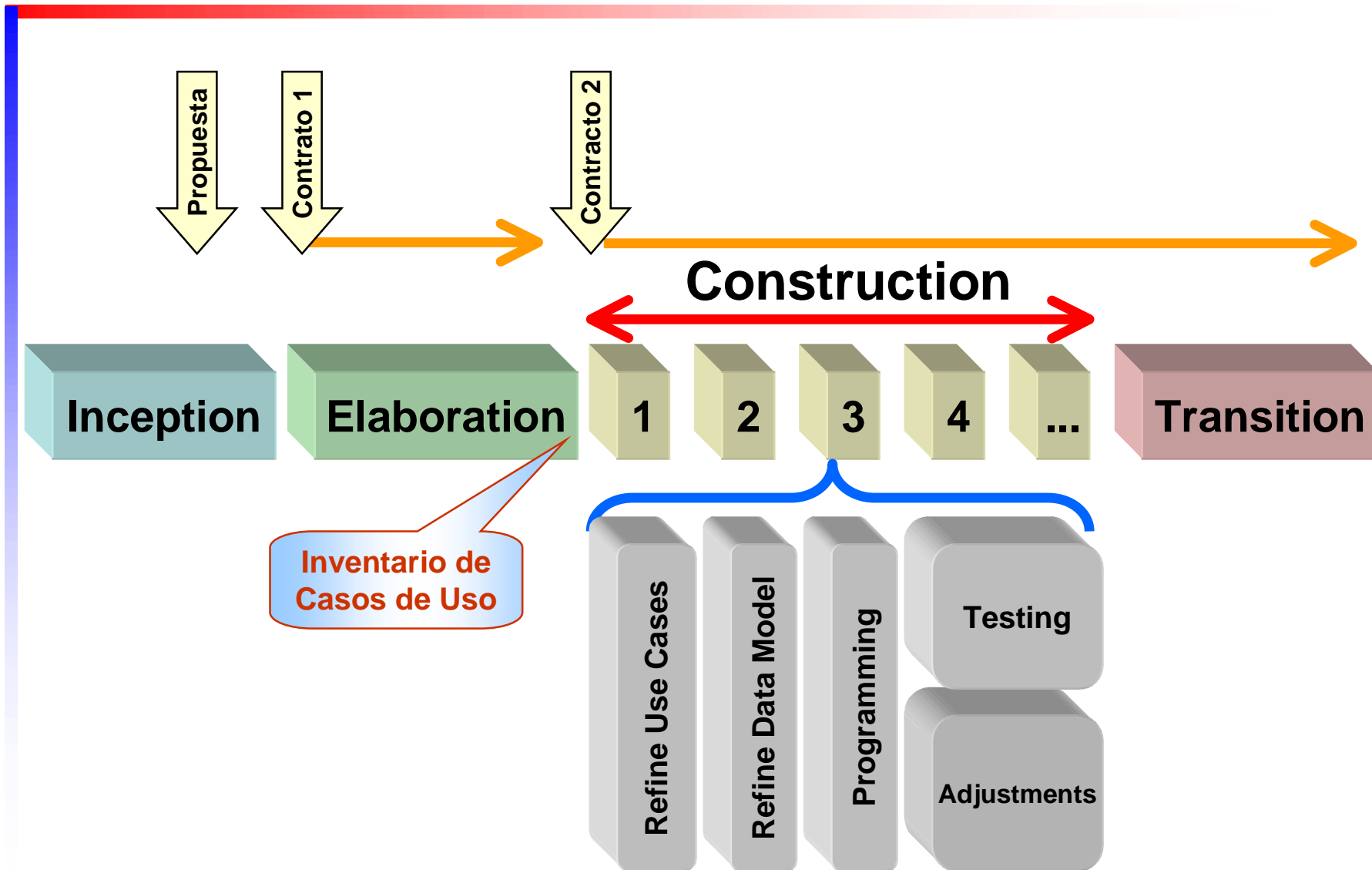
Desarrollo Incremental e Iterativo: “RUP Rational Unified Process”

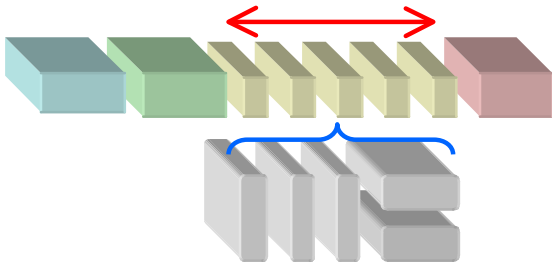




RUP Rational Unified Process

RUP Rational Unified Process





Dinámica del “Rational Unified Process RUP”

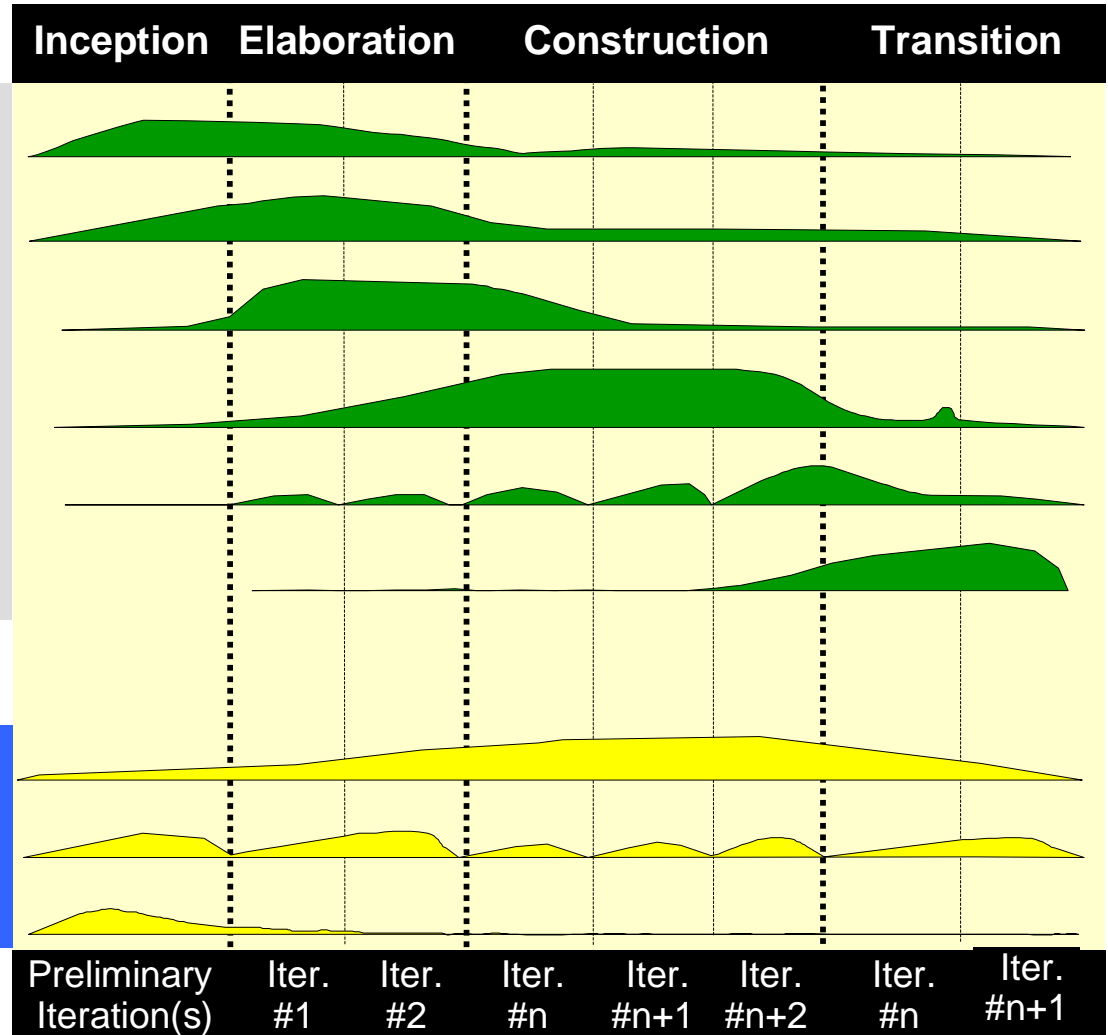
Process Workflows

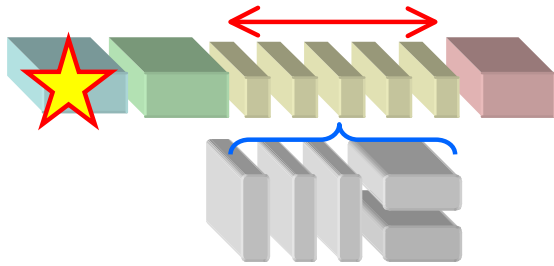
- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

Supporting Workflows

- Configuration Mgmt
- Management Environment

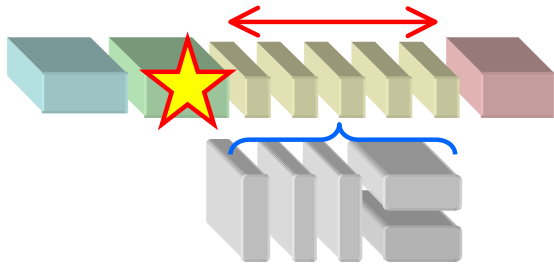
Phases





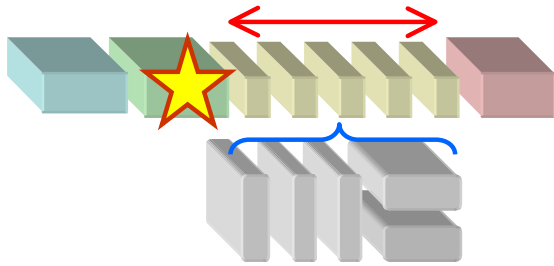
Etapas del RUP:
Inicio (Inception)

- Empezar a conocer el problema.
- Identificar impacto y beneficios del nuevo sistema.
- Identificar relación con otros sistemas.
- Plantear arquitectura.
- Definir el alcance del nuevo sistema.
- Primera estimación de **Costo y Tiempo**.
- **Obtener/Elaborar Propuestas**
- Obtener la decisión y compromiso de arrancar.
- Promover un **dueño** (“sponsor”) de proyecto.



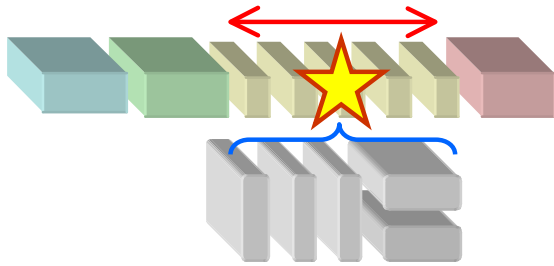
Etapas del RUP: Elaboración (1)

- Entender / Formalizar la “Lógica del Negocio”
- Obtener requerimientos detallados
- Elaborar el Inventario de todos los “Casos de Uso”
- Agrupar los Casos de Uso en **módulos**, de acuerdo a:
 - Áreas del Negocio
 - Interdependencia funcional
- Establecer un orden para implementar los módulos:
 - Dependencia entre Módulos
 - Prioridades (organizacionales, políticas, de negocio, ...)
 - Requerimientos tecnológicos



Etapas del RUP: Elaboración(2)

- Identificar (y enfrentar!) los **riesgos del proyecto**:
 - Lógica del Negocio
 - Requerimientos
 - Tecnológicos
 - Recursos humanos
 - Políticos
- Definir un **Plan Detallado** de construcción.

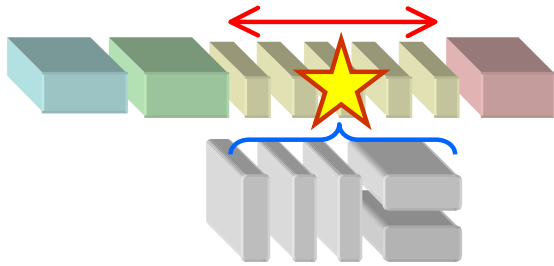


Etapas del RUP: Construcción (1)

INCREMENTAL e ITERATIVO.

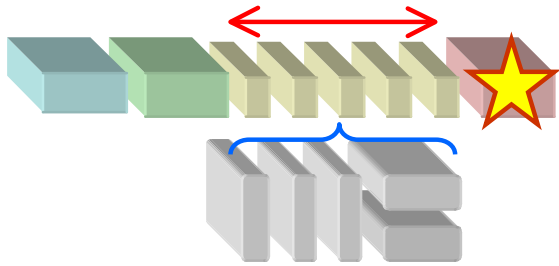
Tomar el siguiente módulo de Casos de Uso:

- Documentar en detalle sus Casos de Uso (empezando por los “pantallazos” y la secuencia de navegación).
- Base de Datos:
 - Refinar / Diseñar las nuevas tablas de la BD
 - Efectuar las modificaciones a las tablas ya existentes (en forma incremental, acumulando los scripts de modificación !!)
- Obtener la aprobación de los Casos de Uso del Módulo.



Etapas del RUP: Construcción (2)

- Programar los Casos de Uso del subsistema
 - Mantener sincronizado el Software y la Documentación de los Casos de Uso
 - Probar los Casos de Uso implementados.
 - Probar interacción con Casos de Uso precedentes.
- Escribir mini-manual de usuario del módulo
- Entregar el módulo para pruebas de los usuarios



Etapas del RUP: Transición

- Homologar todo el software
- Homologar toda la documentación de Casos de Uso
- Pruebas integrales del nuevo sistema (*beta testing*)
- Sintonización fina y optimización de plataforma
- Unificar documentación de instalación.
- Unificar los manuales de usuario final.
- Capacitación de usuarios finales.

Nuestra experiencia en CincoSOFT:

RUP Rational Unified Process “aligerado”

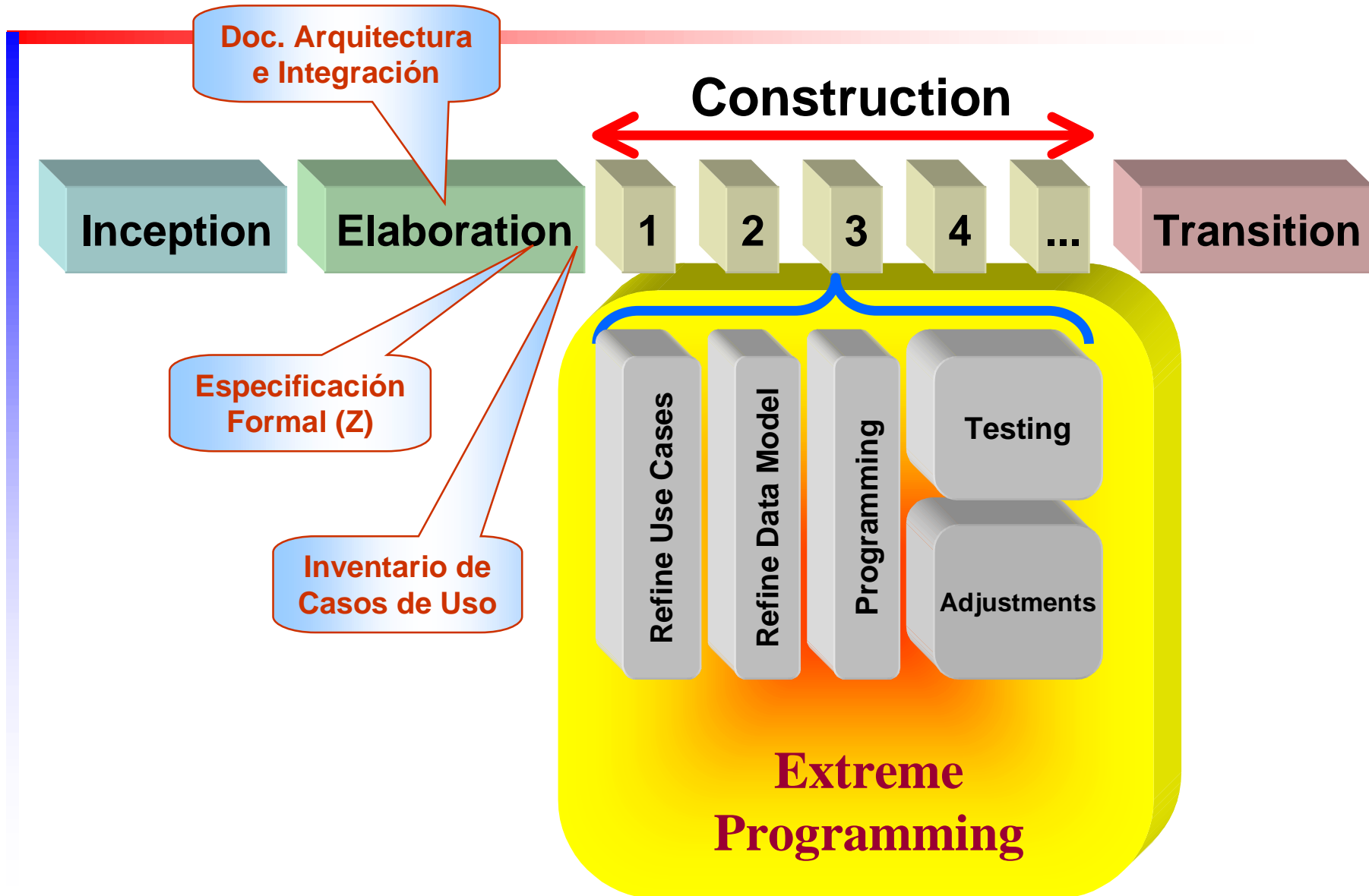
+

“Extreme Programming”

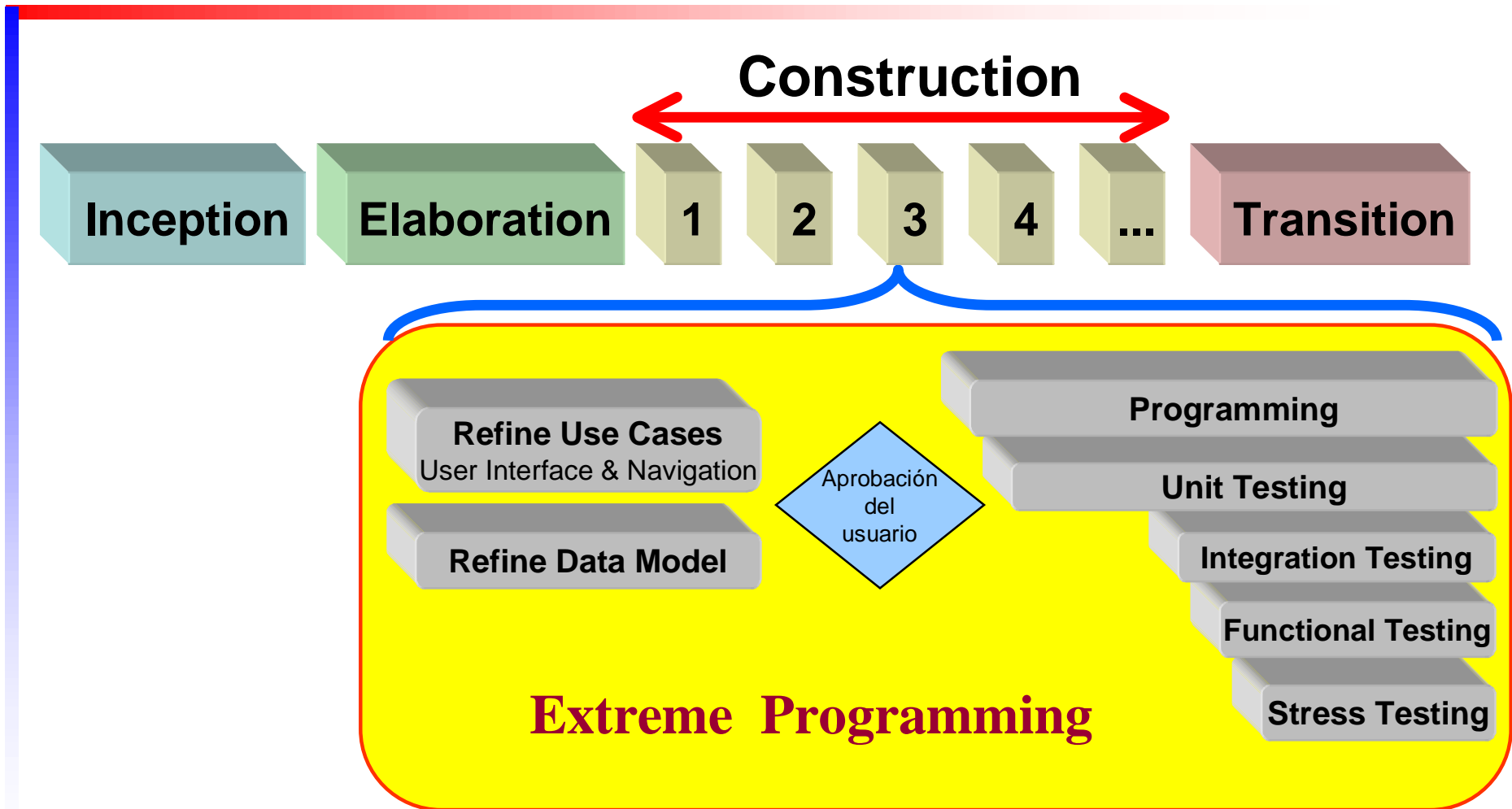
+

apoyo en herramientas libres

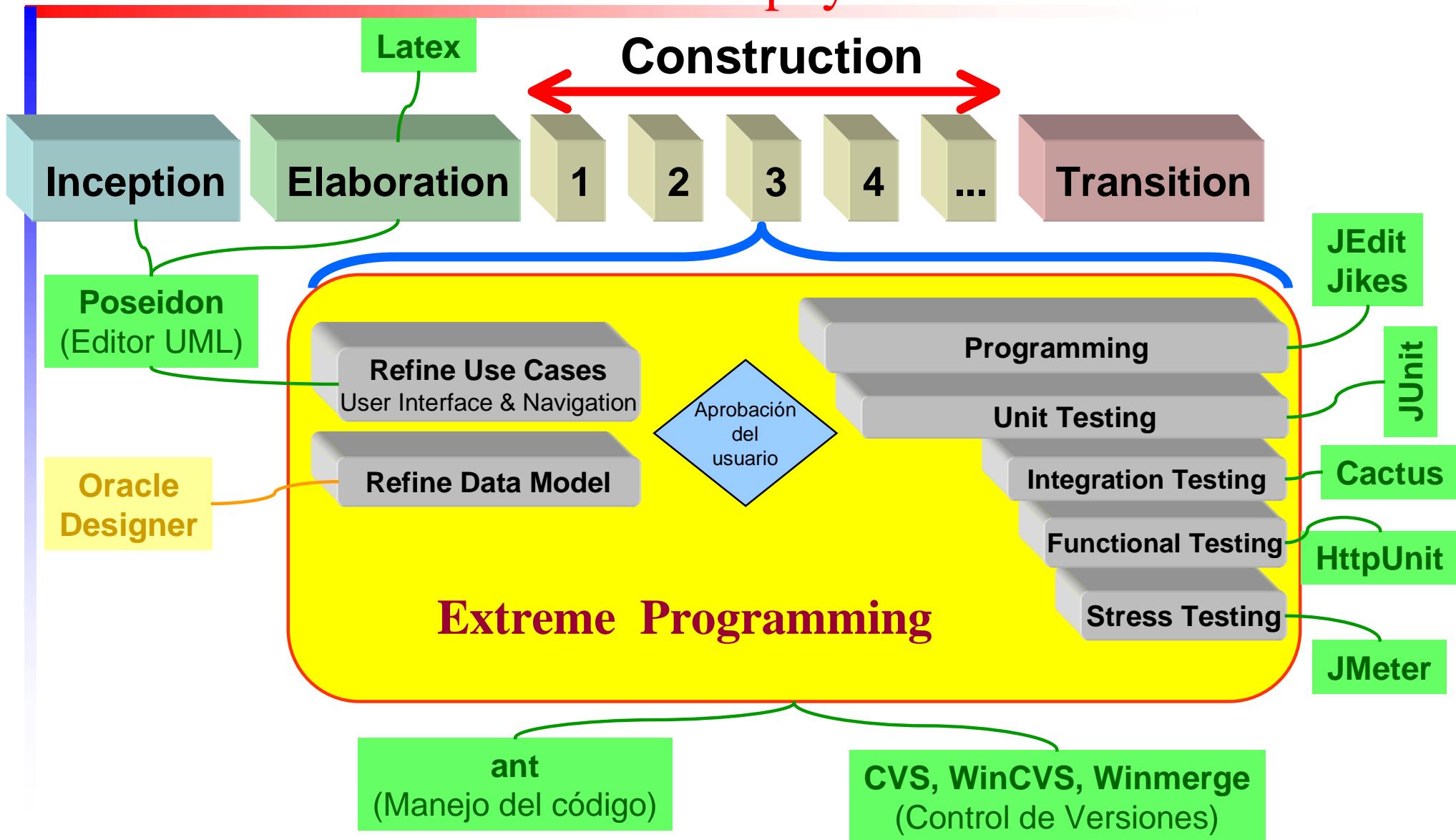
RUP Rational Unified Process “aligerado”



RUP Rational Unified Process “aligerado” + Extreme Programming



Nuestra experiencia en CincoSOFT:
RUP Rational Unified Process “aligerado”
 + Extreme Programming
 + apoyo en herramientas libres





Conclusiones

Conclusiones (1)

- El gerente de proyectos de software debe:

- Conocer en detalle, (ó mejor aún)
- Participar en la adaptación (ó mejor aún)
- Participar en la creación

del Proceso de Software que va a orientar el desarrollo

- Velar por el cumplimiento del proceso
- Estar pendiente de cualquier desvío del cronograma (y presupuesto) detallado

Conclusiones (2)

De lo contrario, el papel del gerente será:

- Impotente
- Pasivo
- Su acción será fundamentalmente **represiva**:
(vociferar, amenazar, multar, echar, ...)
- “Hará fuerza” durante todo el proyecto

Conclusiones (3)

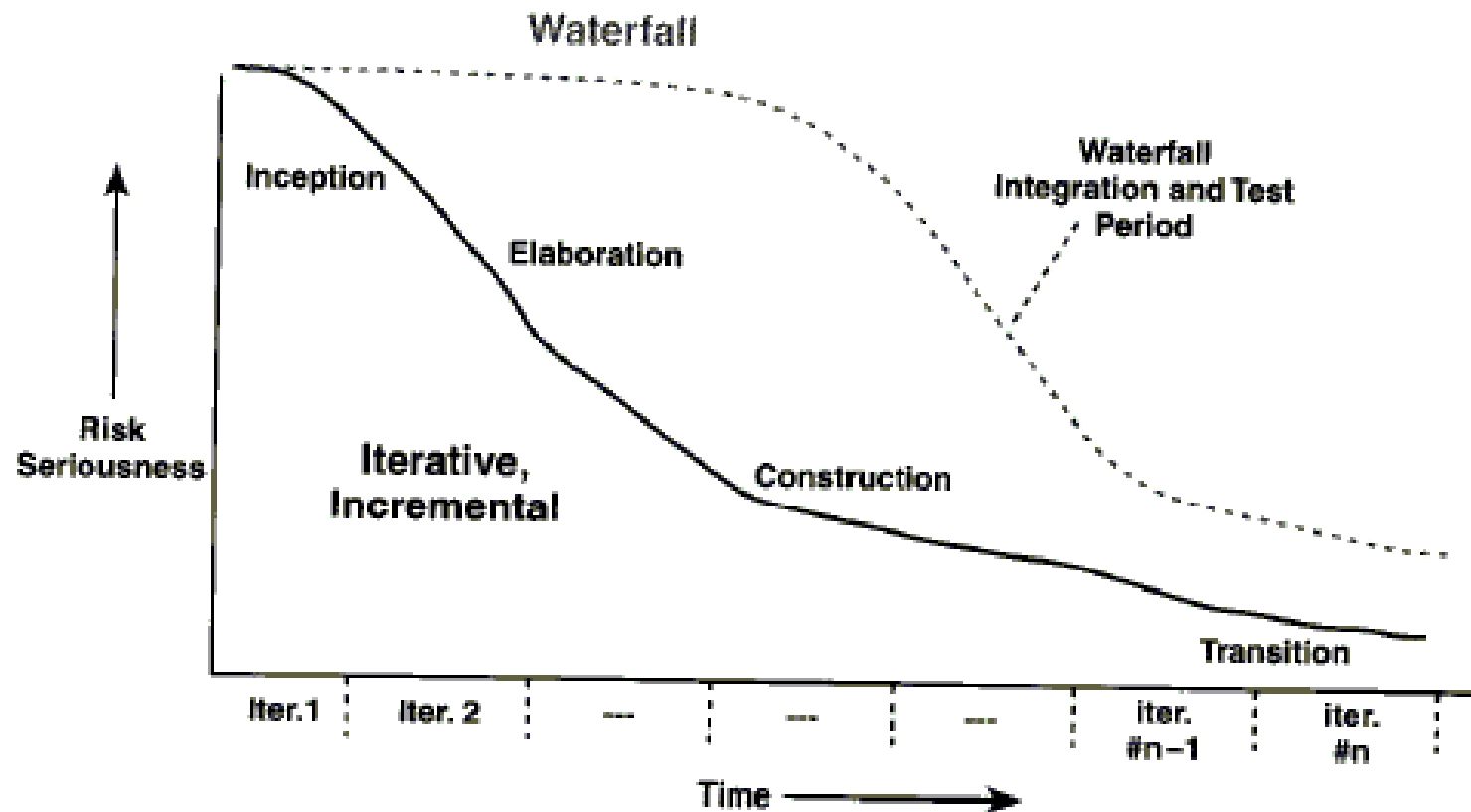
- Los procesos de desarrollo de software modernos deben estar basados en RUP
- RUP es un marco general, no un estándar:
 - Cada compañía debe ajustarlo a sus necesidades.
 - Evitar exageraciones !!!!!
- “Extreme Programming” está teniendo una enorme acogida.
- Hay muchas herramientas libres de alta calidad, que ayudan muchísimo en el proceso

Conclusiones (4)

- Los procesos de desarrollo de software modernos deben estar basados en RUP
- RUP es un marco general, no un estándar:
 - Cada compañía debe ajustarlo a sus necesidades.
 - Evitar exageraciones !!!!!
- “Extreme Programming” está teniendo una enorme acogida.
- Hay muchas herramientas libres de alta calidad, que ayudan enormemente en el proceso de desarrollo

Conclusiones (5)

- Los proyectos de software con las tecnologías actuales (Java J2EE ó Microsoft .NET) son de alto riesgo



Conclusión Final

