UNIVERSIDAD DE LOS ANDES
VENEZUELA

Métodos de Desarrollo de Software

(o bien, como desarrollar software sin morir en el intento...)

Universidad de los Andes
Demián Gutierrez
Julio 2011

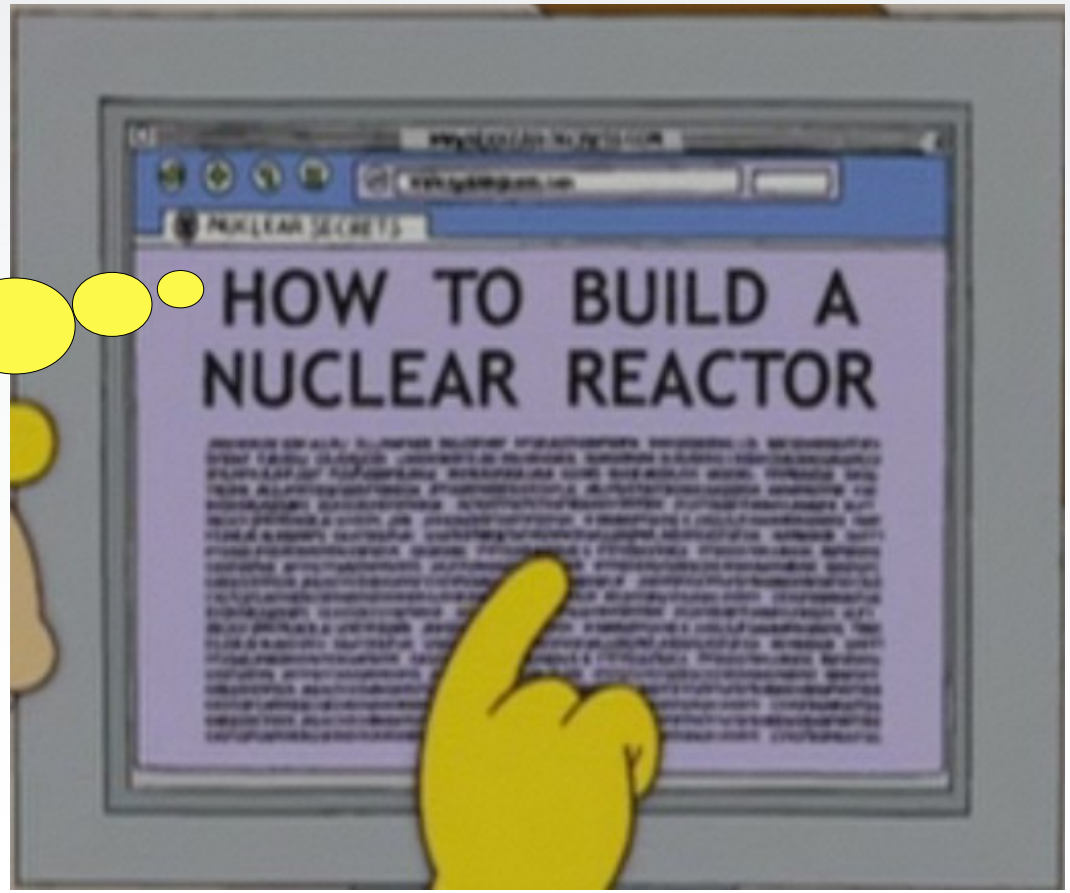
¿método?

Métodos / Metodologías

UNIVERSIDAD DE LOS ANDES VENEZUELA

Método: Es un ***conjunto de herramientas***, ***técnicas*** y ***procesos*** que brindan soporte y facilitan el logro u obtención de ***una meta***

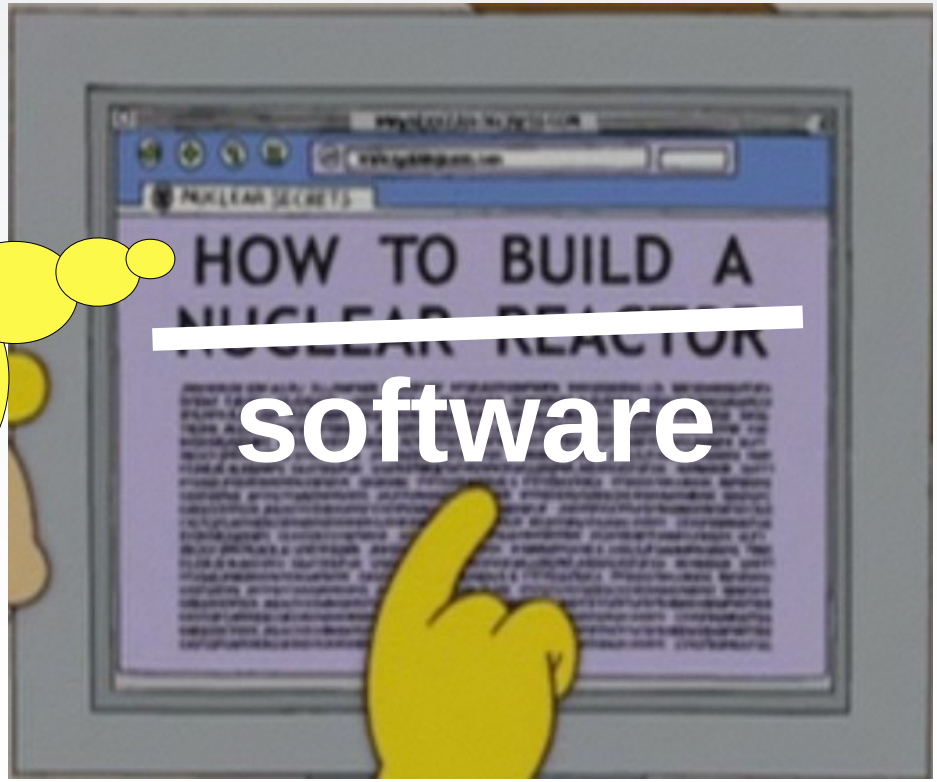
¿Cómo Construir un Reactor Nuclear?



UNIVERSIDAD DE LOS ANDES VENEZUELA

Método: que hacer, *a lo largo de todo el ciclo de vida del software*, para construir un producto bueno, de calidad, dentro del presupuesto y a tiempo

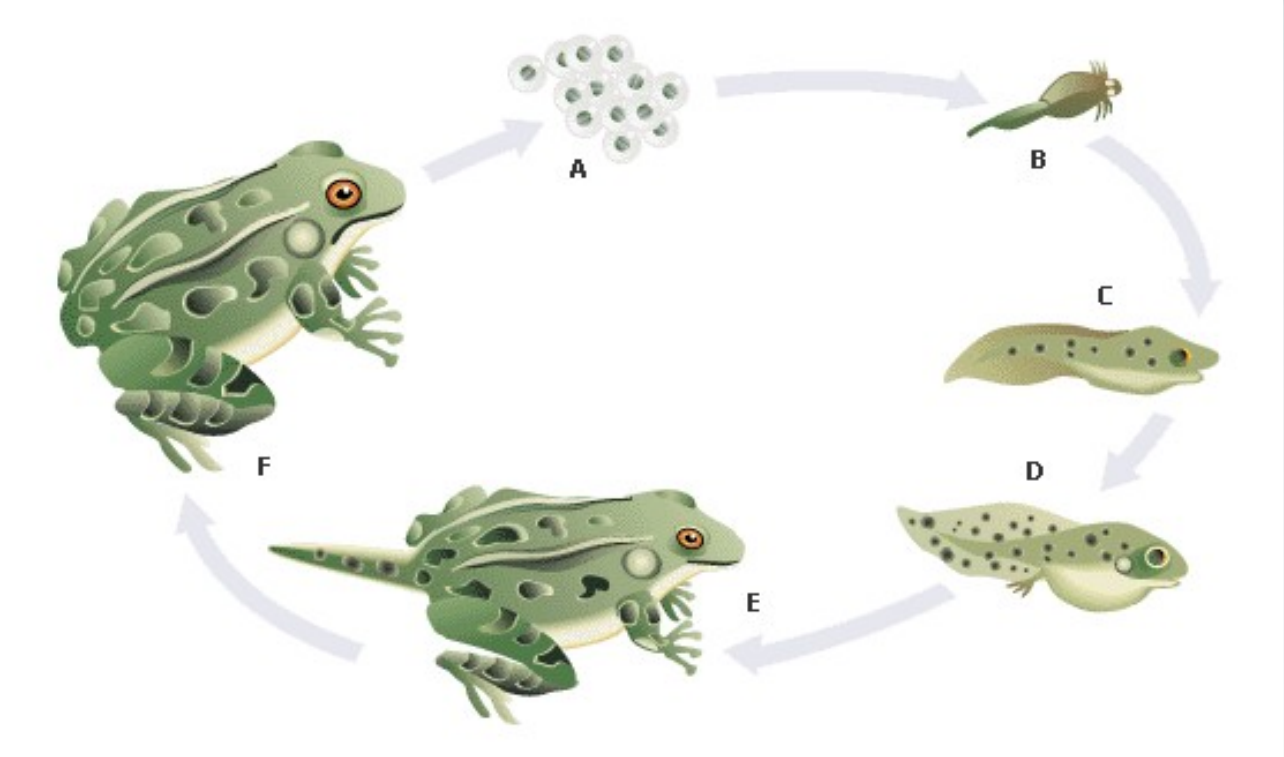
¿Cómo Construir un ~~Reactor Nuclear~~ **software**?



¿ciclo de vida?
¿ciclo de desarrollo?

Ciclo de Vida / Ciclo de Desarrollo

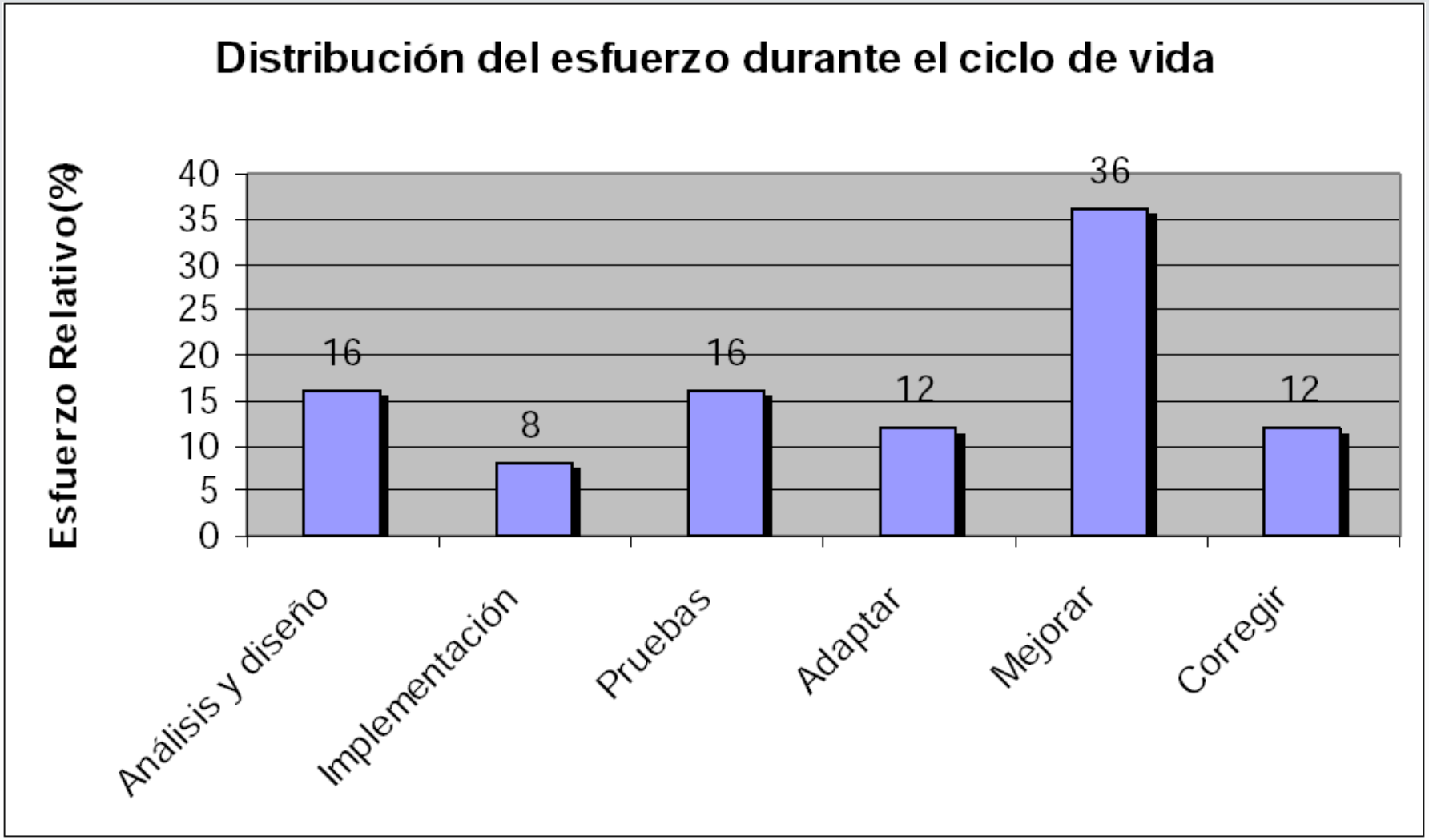
UNIVERSIDAD DE LOS ANDES VENEZUELA



Describe la vida de un producto de software ***desde su definición***, pasando por su diseño, implementación, verificación, validación, entrega, y ***hasta su operación y mantenimiento***

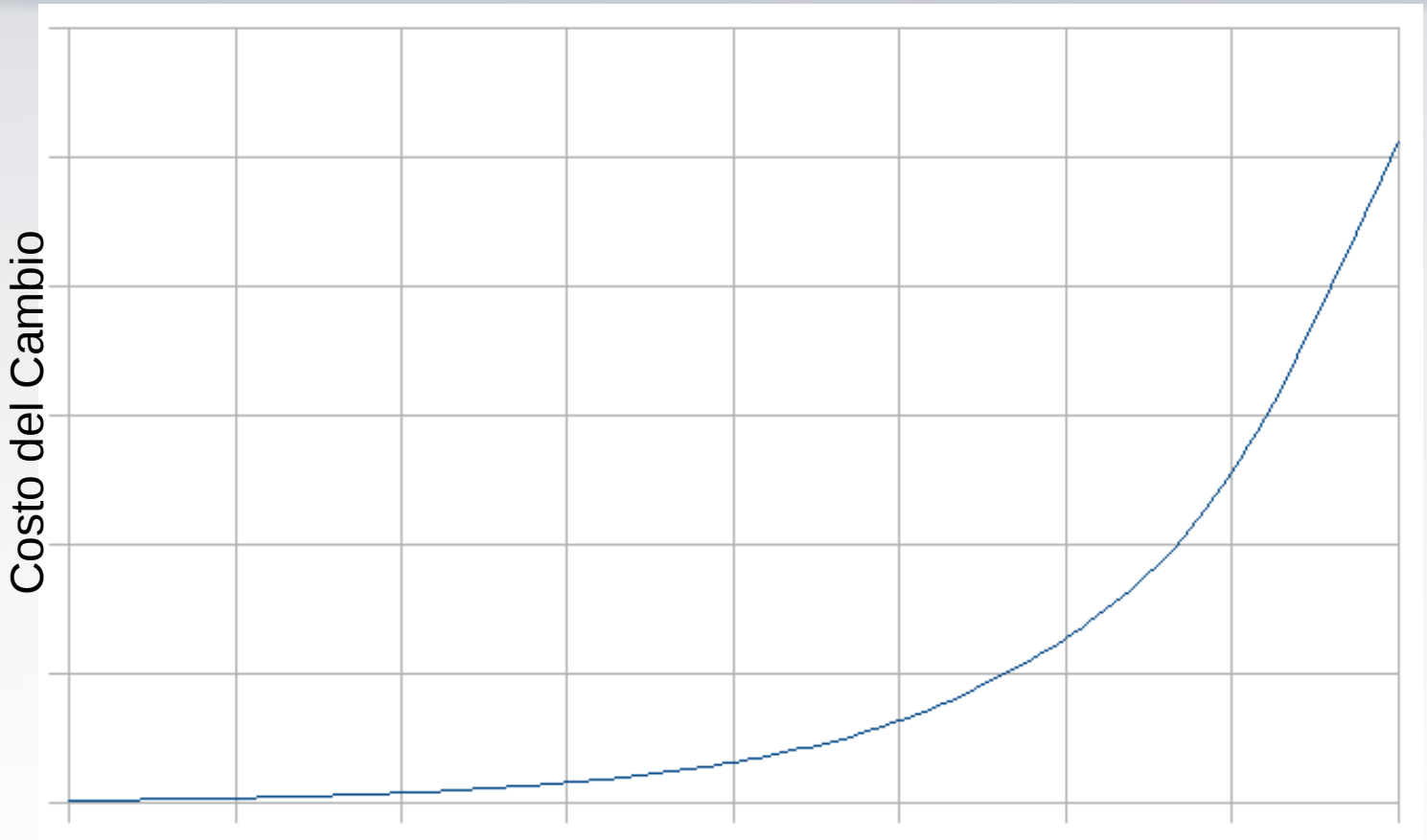
¿por qué es necesario un método para desarrollar software?

Ciclo de Vida / Ciclo de Desarrollo



por lo complejo que resulta desarrollar software

Ciclo de Vida / Ciclo de Desarrollo



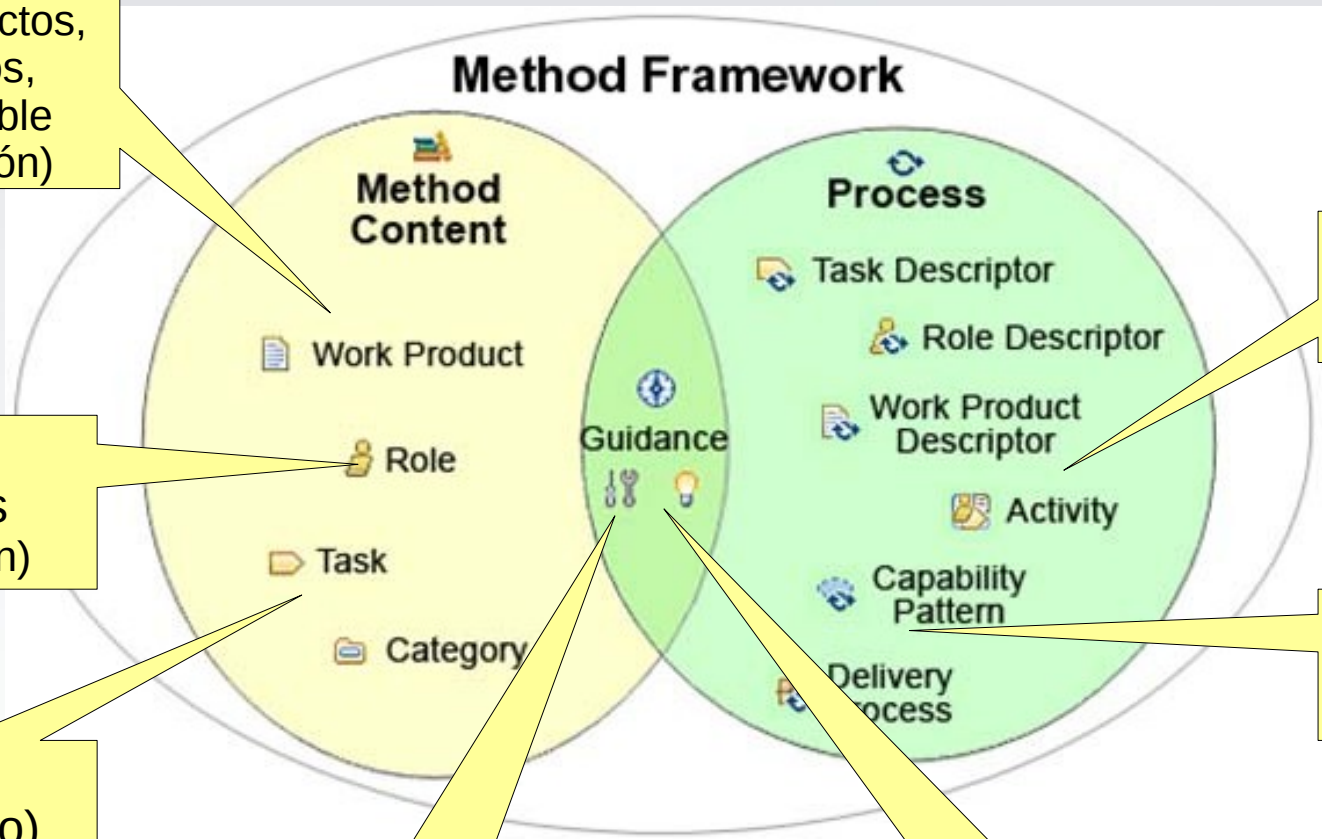
(aunque los métodos ágiles pueden cambiar esta visión)

Fuente: Adaptado de Kent Beck / Extreme Programming Explained, Embrace the Change

por el costo del cambio, la naturaleza del software y otras razones

¿qué aporta un método?

Métodos / Metodologías



Productos, Subproductos, Insumos, Entregable (definición)

Roles, Actores (definición)

Tareas (Genérico) (definición)

Guías, Herramientas

Buenas Prácticas

Actividades

Procesos

y otros elementos adicionales...

Fuente: Eclipse Process Framework Composer / April 2007 / Peter Haumer / IBM Rational Software

Métodos / Metodologías (Herramientas)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Casos de Uso, Plantillas de Documentos, UML: Diagramas de Clases, de Casos de Uso, de Actividades, de Secuencia, etcétera.

Grafos de navegación, lenguajes de programación, bibliotecas, armazones de aplicación (frameworks), entornos integrados de desarrollo (IDEs), armazones de pruebas, etcétera.

Software de gestión, herramientas de gestión, etcétera

y muchas otras...

Métodos / Metodologías (Buenas Prácticas)

UNIVERSIDAD DE LOS ANDES VENEZUELA

¿Su empresa usa control de código fuente? ¿Control de versiones?

¿Se hacen "compilaciones" (builds) e integraciones diarias?

¿Se tiene algún tipo de base de datos de defectos (bugs)?

¿Arreglan los defectos existentes antes de escribir código nuevo?

¿Se mantiene un calendario de proyecto actualizado?

¿Trabajan en base a especificaciones de algún tipo?

¿Los programadores tienen condiciones adecuadas y tranquilas de trabajo?

¿Se utilizan las mejores herramientas que el dinero puede comprar?

¿Se tienen probadores? ¿Se tienen probadores dedicados sólo a las pruebas?

¿Los nuevos candidatos a programadores escriben código durante su entrevista de trabajo?

¿Se realizan pruebas de usabilidad?

entre otras, y no necesariamente en este orden...

Fuente: Joel on Software / http://www.joelonsoftware.com/articles/fog0000000043.html

¿proceso?
¿modelo de proceso?

Métodos / Metodologías
¿Qué es el Proceso?

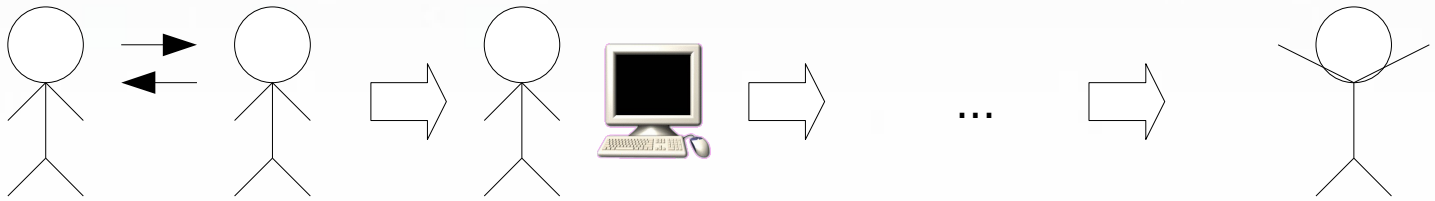
UNIVERSIDAD DE LOS ANDES VENEZUELA

Un proceso define quien está haciendo qué, cuándo y cómo lograr cierta meta.

The three "Amigos"

Un proceso es "una serie de pasos que involucra actividades, restricciones y recursos que producen una salida de algún tipo"

Pfleeger



Los *"procesos de desarrollo de software"* poseen ***reglas preestablecidas***, y deben ser aplicados en la creación del software de mediano y gran porte, ya que en caso contrario lo más seguro es que el proyecto o no logre concluir o termine sin cumplir los objetivos previstos, y con variedad de fallos inaceptables (fracasan, en pocas palabras).

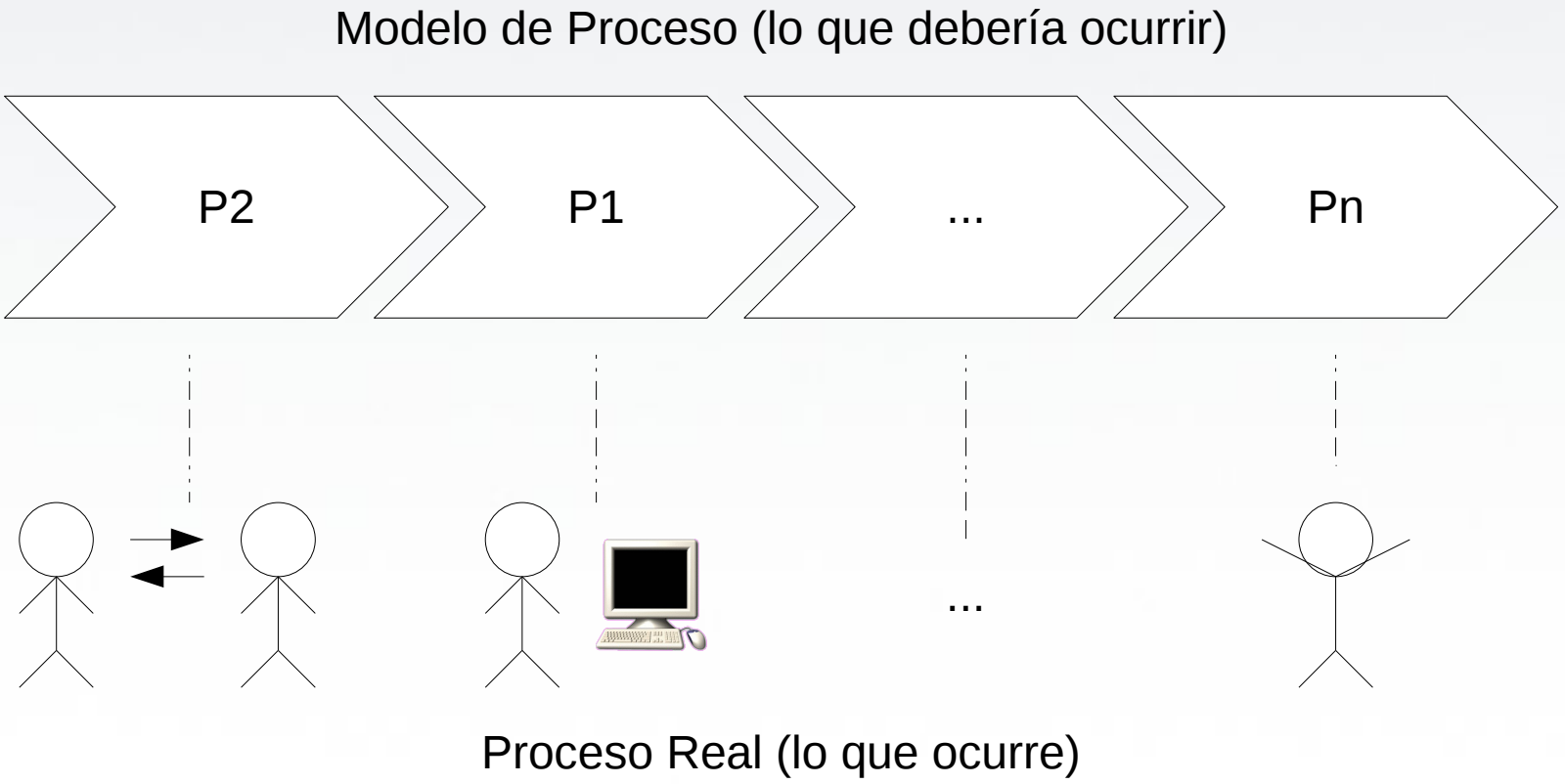
Tomado de:http://es.wikipedia.org/wiki/Software

...en realidad, esta definición se refiere a un "modelo de proceso"...

Métodos / Metodologías
(Diferencia entre Proceso y Modelo de Proceso)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Un modelo de proceso de software es una ***representación abstracta*** de un proceso de software.



Algunas Características de los Procesos (Modelos de...)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Claridad: ¿Es fácil de comprender?

Visibilidad: ¿Puedo Ver lo que Ocurre en el Proceso?

Fiabilidad: Probabilidad de Buen Funcionamiento

Robustez: ¿Es Difícil de Perturbar?

Facilidad de Soporte

Facilidad de Mantenimiento

Aceptación: ¿Se vende? ¿Los "Usuarios" lo Consideran Viable?

Rapidez: ¿Permite Entregar Rápido el Producto?

Conveniencia: ¿Es el método conveniente para lo que vamos a hacer?

Adaptabilidad: ¿Lo puedo cambiar según las necesidades?

Procesos Livianos vs Pesados

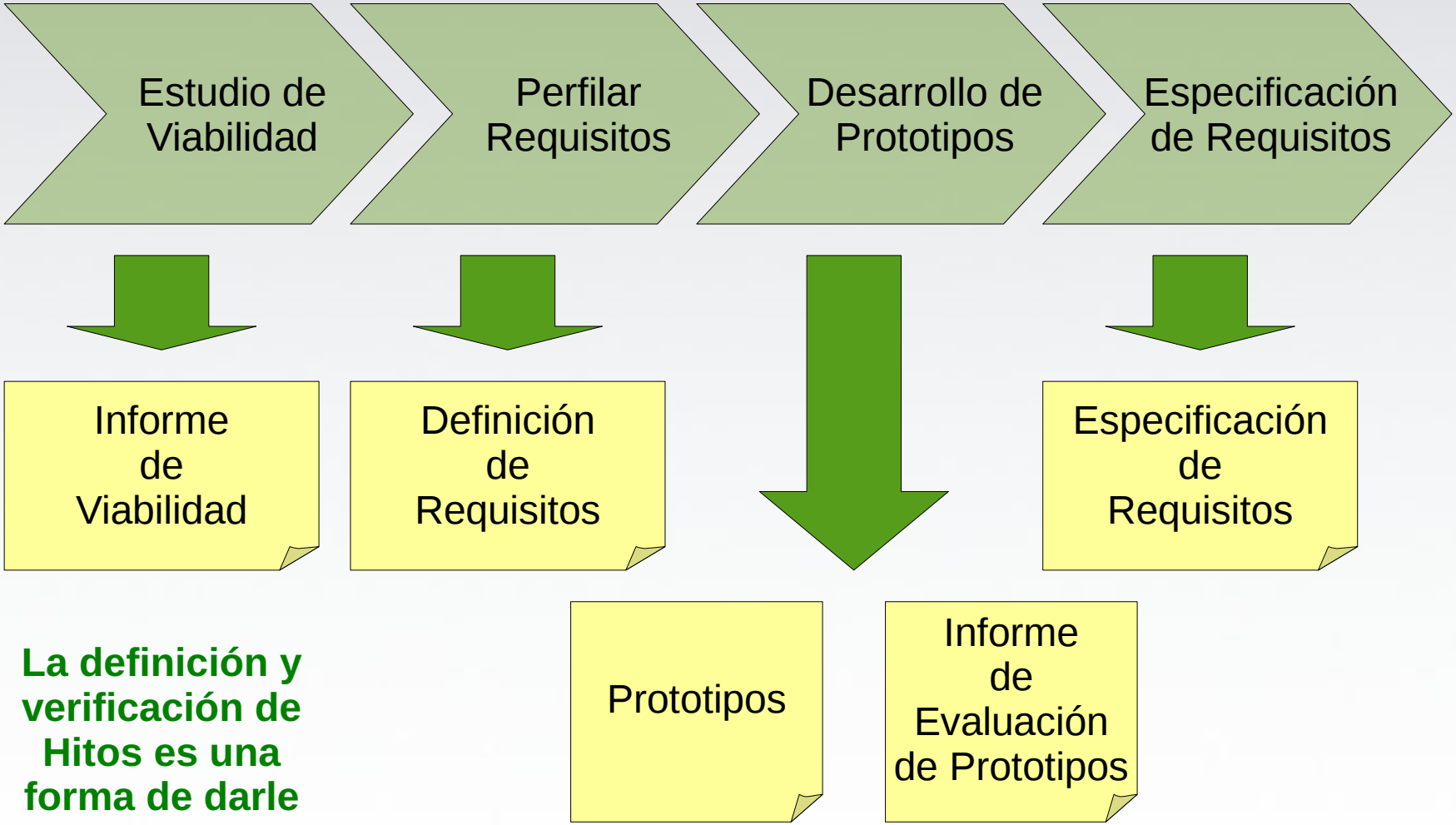
UNIVERSIDAD DE LOS ANDES VENEZUELA

Procesos Livianos
(O de “peso liviano”)

Procesos Pesados
(O de “peso pesado”)

entregables, subproductos, hitos, etc

Métodos / Metodologías (Hitos)



La definición y verificación de Hitos es una forma de darle "visibilidad" al proceso

Fuente: Adaptado de EPS Informática UAM (T3: 18/19)

Métodos / Metodologías (Hitos)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Producto intermedio “enseñable”

Se consigue un hito cuando se ha revisado la calidad de uno o más productos y se han aceptado

Tras cada hito se debería generar un informe de progreso del proyecto

Definir Qué, Quién, Cuándo y Cómo se va a evaluar

Coincidiendo con el final de una fase (al menos)

Definir los productos correspondientes a cada hito

Fuente EPS Informática UAM (T3: 18/19)

roles
actores

UNIVERSIDAD DE LOS ANDES VENEZUELA

Los roles sirven para definir quién hace que (y probablemente cuando), son una forma de asignar y definir responsabilidades a personas, sin tener que nombrar a las personas en particular

Métodos / Metodologías (Roles)

Un cerdo y un pollo van caminando por la carretera. El pollo le dice al cerdo:*

-Oye, ¿por qué no abrimos un restaurante?

El cerdo se vuelve y le responde:

-Buena idea, ¿cómo quieres que lo llamemos?

El pollo se lo piensa y propone:

-¿Por qué no lo llamamos "Huevos con jamón".

Rol: Las acciones o actividades asignadas o requeridas de una persona o grupo ("La función del maestro", "El gobierno debe de...")

Rol: Un personaje o parte escenificada por un actor; El comportamiento esperado de un individuo en la sociedad. La función o posición de algo.

*-No cuentes conmigo -responde el cerdo-. En ese caso, tú sólo estarías **IMPLICADO**, mientras que yo estaría realmente **COMPROMETIDO**.*

**** Fuente:*** Tomado de SCRUM

UNIVERSIDAD DE LOS ANDES VENEZUELA

importante

No confundir los roles en los procesos de desarrollo con los *actores o roles* del sistema o con los *interesados o "stakeholders"*

¡Son dos cosas totalmente distintas!

Ej: Describir al "desarrollador" como actor del sistema en el documento de casos de uso probablemente será un error en la mayoría de los casos

UNIVERSIDAD DE LOS ANDES VENEZUELA

¿modelos básicos de procesos?

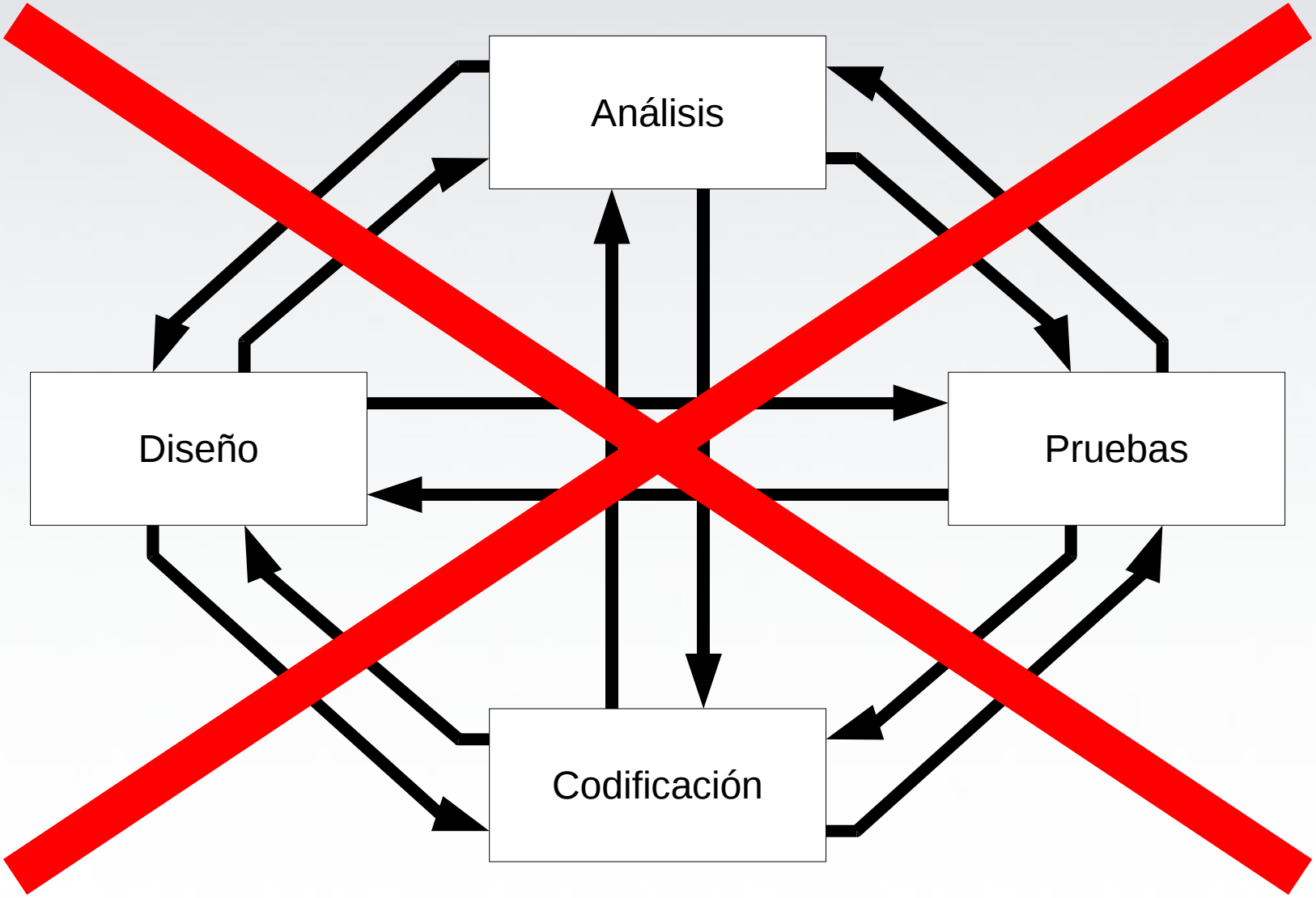
...modelos de procesos muy generales (algunas veces llamados paradigmas de proceso) ... Esto es, vemos el marco de trabajo del proceso, pero no los detalles de actividades especìficas. Estos modelos generales ***no son descripciones definitivas*** de los procesos del software. Más bien, ***son abstracciones de los procesos*** que se pueden usar para explicar diferentes enfoques del desarrollo de software...

Ian Sommerville

lo que algunas veces pasa (y no debería)

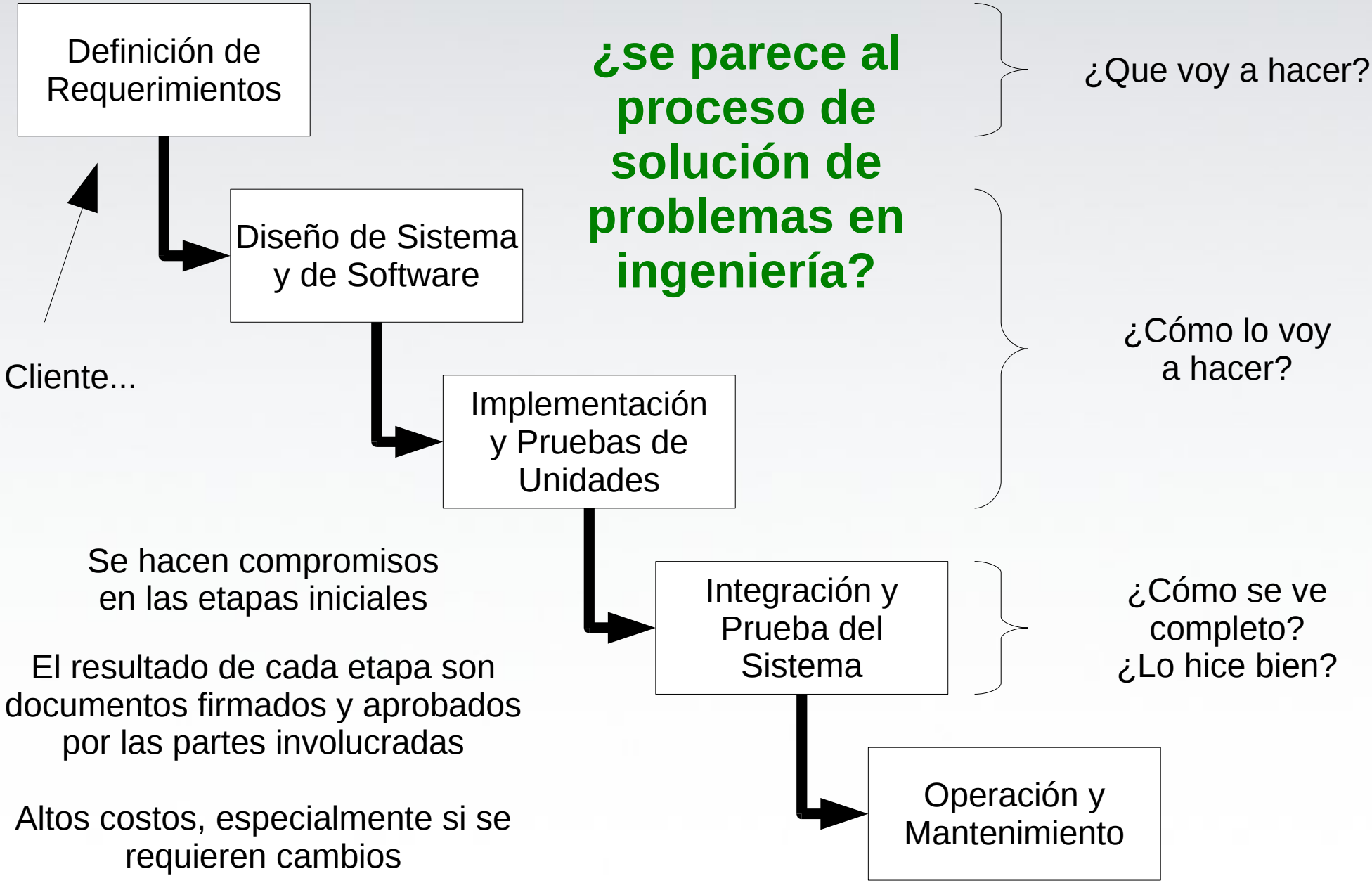
Ciclo de Vida / Ciclo de Desarrollo
(Lo que usualmente pasa, y no debe pasar)

UNIVERSIDAD DE LOS ANDES VENEZUELA

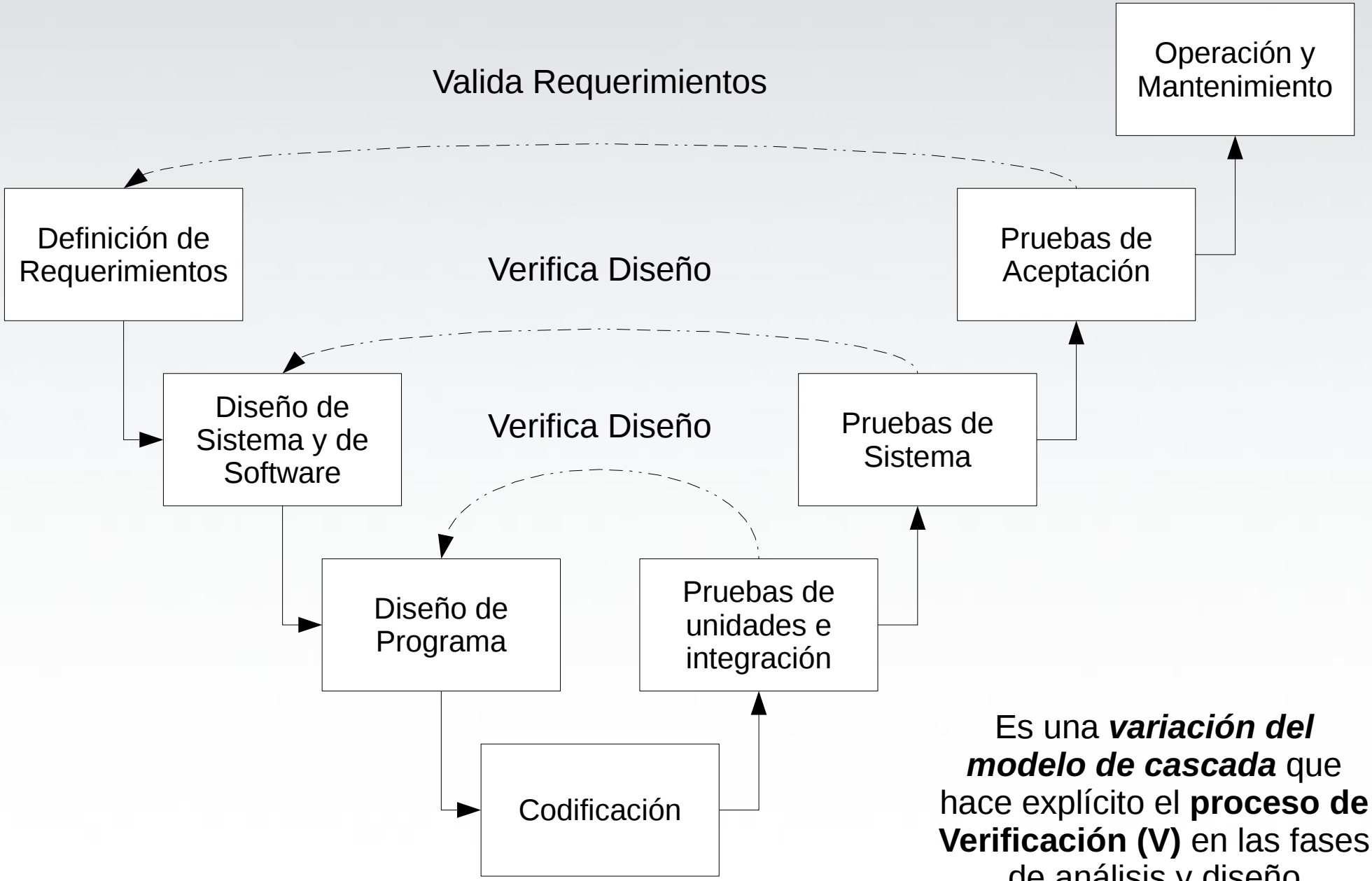


proceso en cascada

¿Proceso en Cascada?

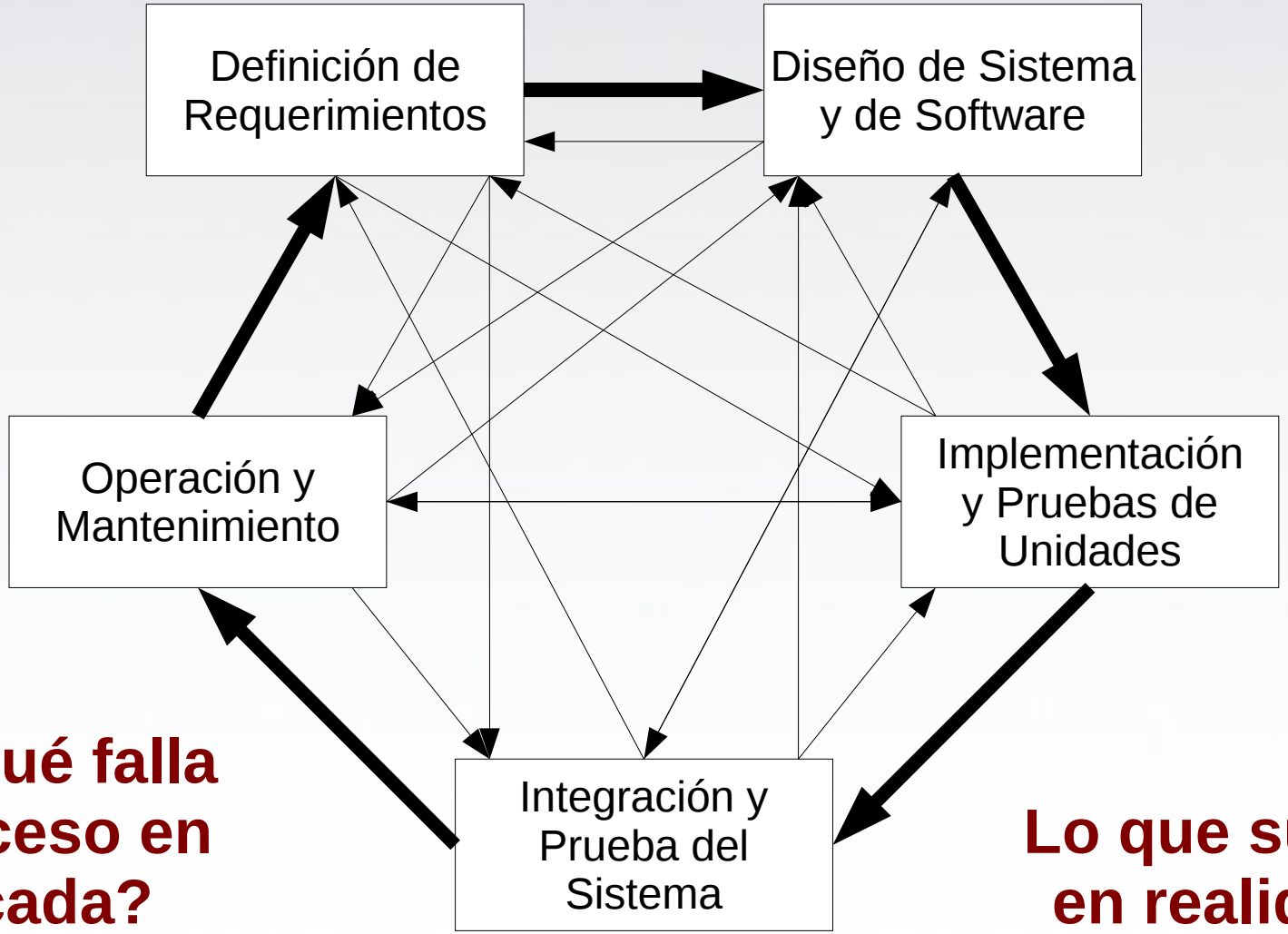


Modelo en V



Es una ***variación del modelo de cascada*** que hace explícito el **proceso de Verificación (V)** en las fases de análisis y diseño

¿Proceso en Cascada?



¿Por qué falla el proceso en cascada?

Lo que sucede en realidad...

¿Proceso en Cascada?

UNIVERSIDAD DE LOS ANDES VENEZUELA

Es el modelo más simple, conocido

El producto / resultado sólo se ve al final (para el cliente). Si existe algún error (diferencia) ésto tiene un resultado catastrófico

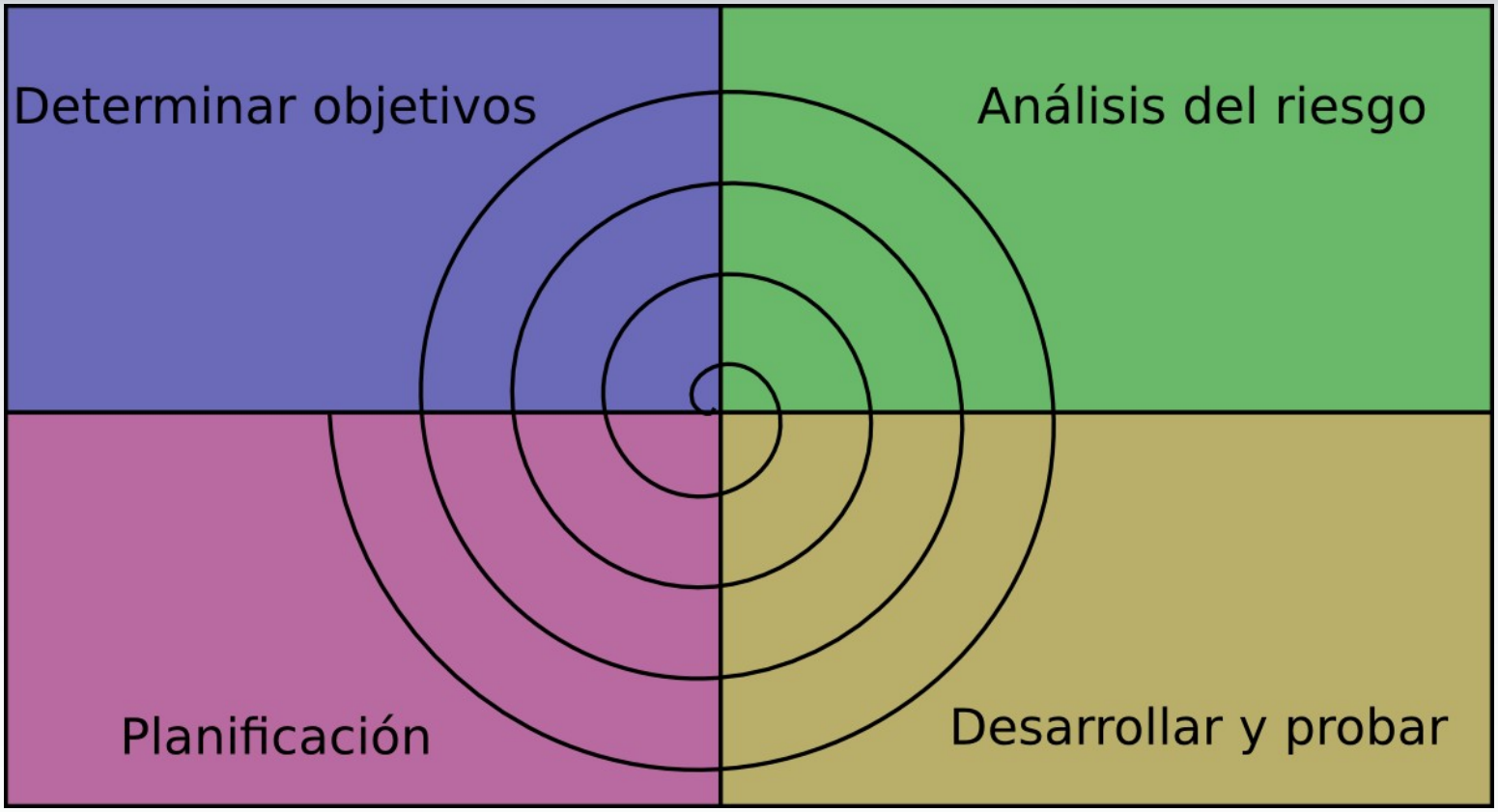
Suele ser difícil para el cliente establecer ***TODOS*** los requisitos de manera explicita (y al principio del proceso). Naturaleza del software: Cambio

Suele ser difícil para el cliente establecer ***TODA*** las arquitectura del software de manera explicita al principio del proceso.

Se producen estados de bloqueo, en los que algunos miembros del equipo deben esperar a otros para terminar tareas dependientes

proceso / modelo en espiral

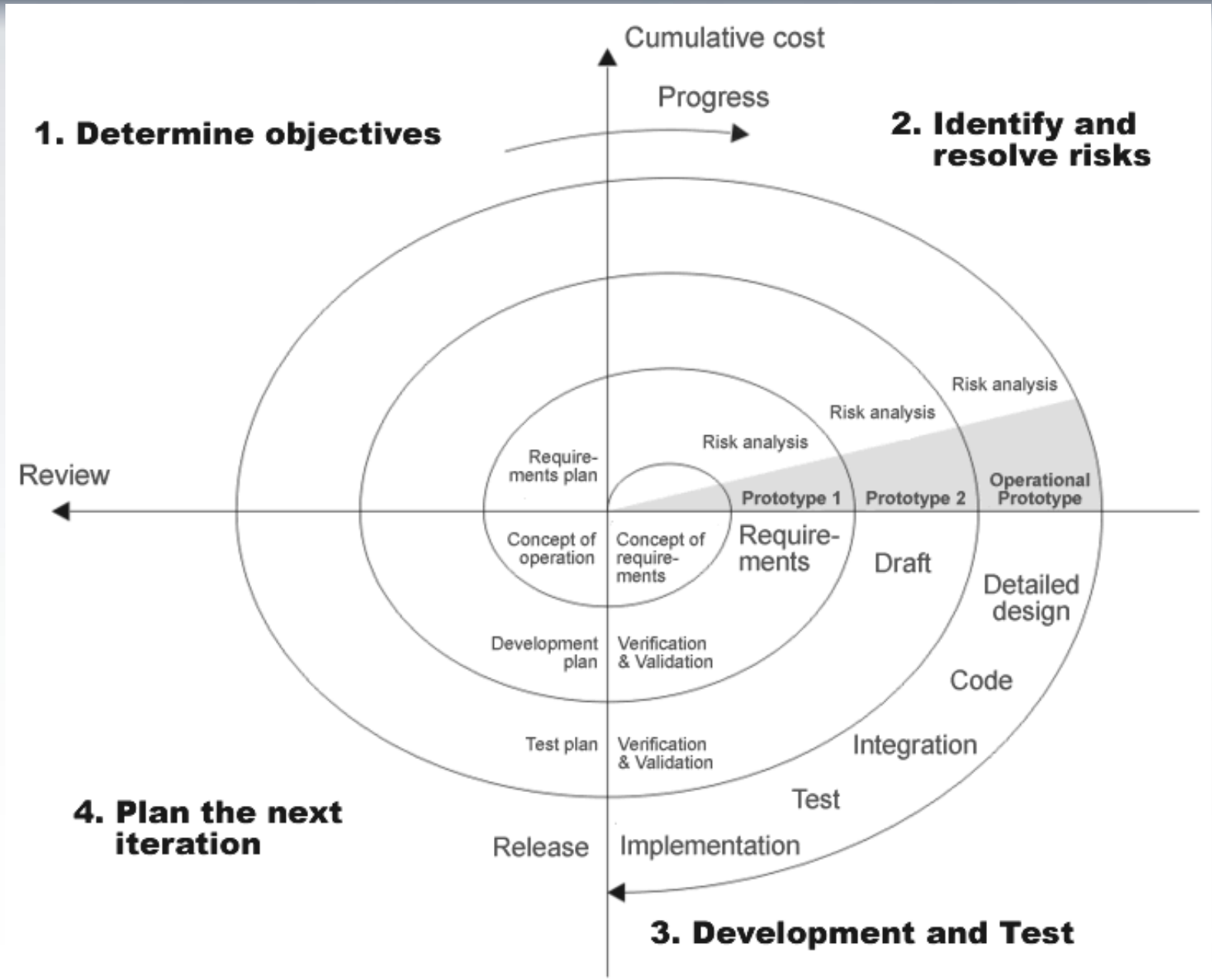
Modelo de Riesgos o de Espiral



En general se puede asociar cada giro a una fase del proceso de desarrollo. Ej. 1er giro: objetivos, alternativas restricciones; 2do giro: especificación de requisitos; 3er giro: diseño; 4to giro: implementación, etcétera.

Fuente; http://es.wikipedia.org/wiki/Espiral_de_Boehm

Modelo de Riesgos o de Espiral



Fuente; http://en.wikipedia.org/wiki/Spiral_model

Modelo de Riesgos o de Espiral

UNIVERSIDAD DE LOS ANDES VENEZUELA

Incluye de forma explícita en cada giro la especificación de objetivos, definición de alternativas y restricciones y evaluación de riesgos (verdaderamente importante)

En cada giro se construye un nuevo modelo del sistema.

Hasta los momentos, se considera el mejor modelo para el desarrollo de sistemas grandes (El más fiable)

No es aconsejable para sistemas pequeños debido a su alta complejidad

procesos

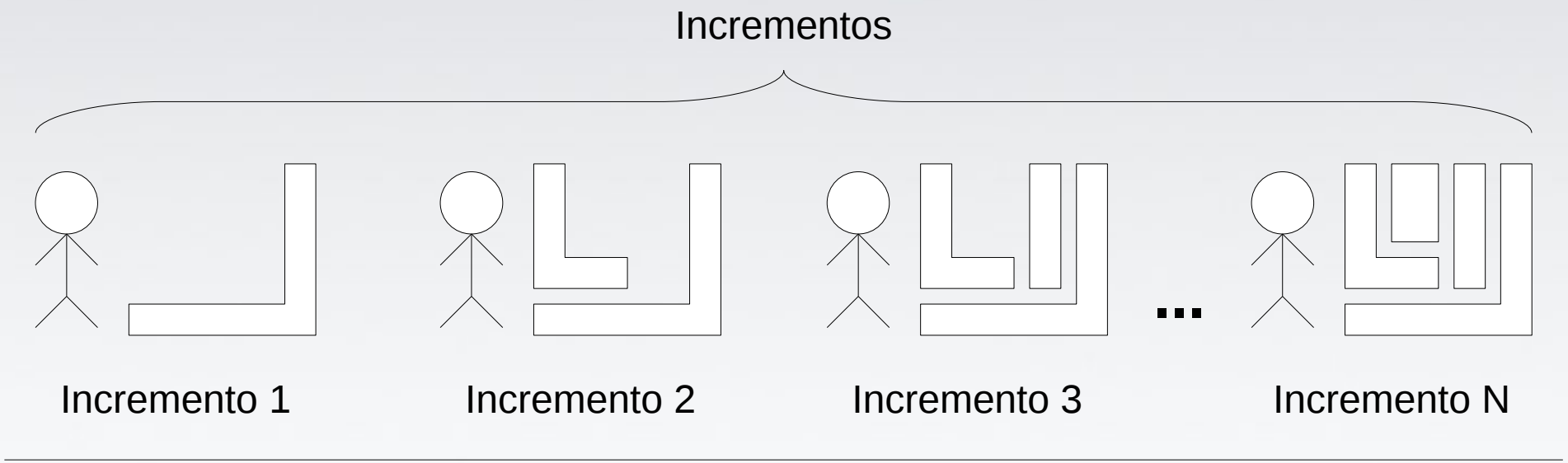
iterativos

incrementales

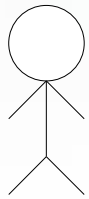
UNIVERSIDAD DE LOS ANDES VENEZUELA

¿qué es un incremento?
(incrementar algo)
¿qué es una iteración?
(iterar)

Modelos Incrementales (Modelo Incremental)

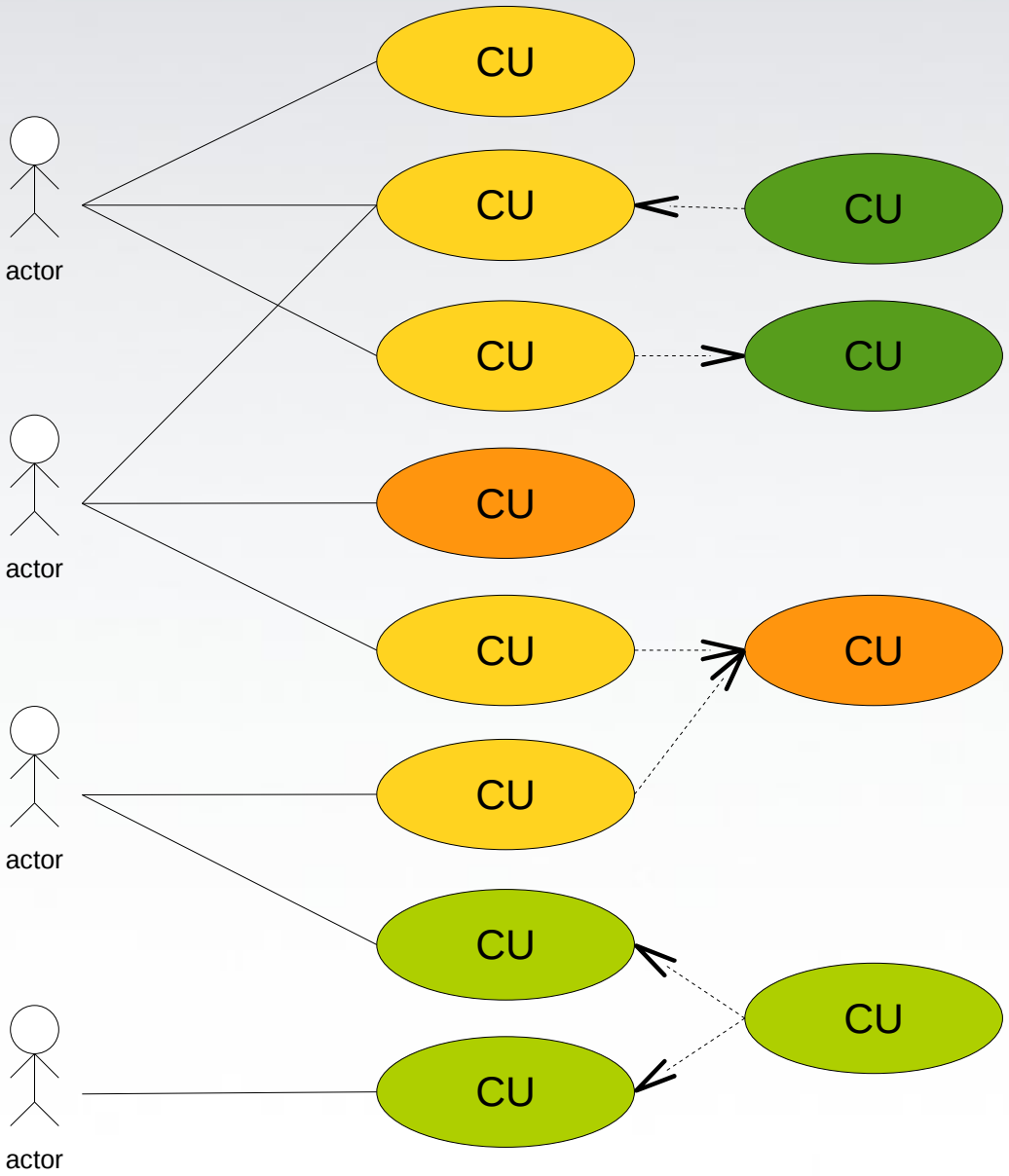


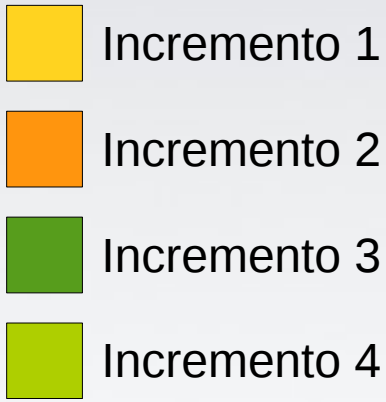
Las fases se dividen en incrementos, en cada incremento se desarrolla una parte de la funcionalidad y se validan los productos resultantes



En general, una vez los productos de una fase se consideran listos, estos no se modifican más a lo largo de las siguientes fases

Modelos Incrementales (Modelo Incremental)





En general, cada incremento añade funcionalidad nueva al sistema, de manera que el usuario puede ir utilizando (validando) la funcionalidad antes de terminar el sistema completo

Modelos Incrementales (Modelo Incremental)

UNIVERSIDAD DE LOS ANDES VENEZUELA

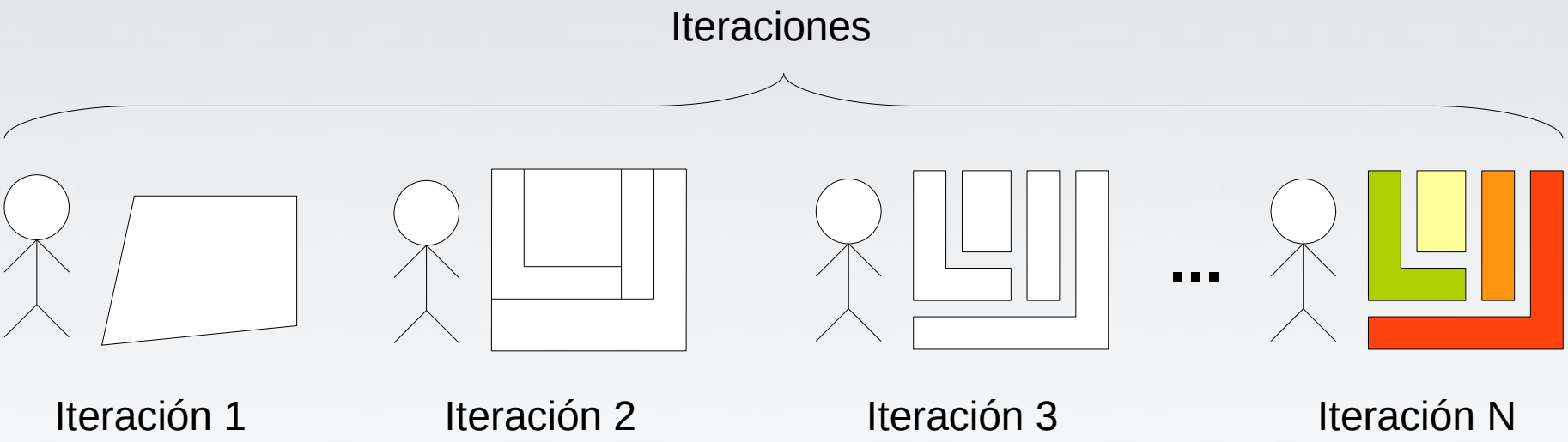
El sistema se desarrolla como una secuencia de pasos e iteraciones una vez establecida la arquitectura global

Los usuarios pueden experimentar con los productos resultantes de cada iteración, y usualmente el equipo de desarrollo puede continuar con el trabajo mientras que los usuarios experimentan con el sistema

En general, la idea es combinar lo mejor de las estrategias orientadas a prototipos con una buena gestión

En general, luego de que se valida y se termina un componente, este no se cambia (o se procura no cambiarlo) a menos que se encuentren errores (Bugs)

Modelos Incrementales (Modelo Iterativo)



Cada iteración ***refina*** lo realizado en la iteración anterior. De esta forma se produce una dinámica en la que se van mejorando los productos (entregables) obtenidos en la iteración anterior. Eventualmente se realizarán todas las iteraciones planificadas, o se llegará al nivel de refinamiento deseado

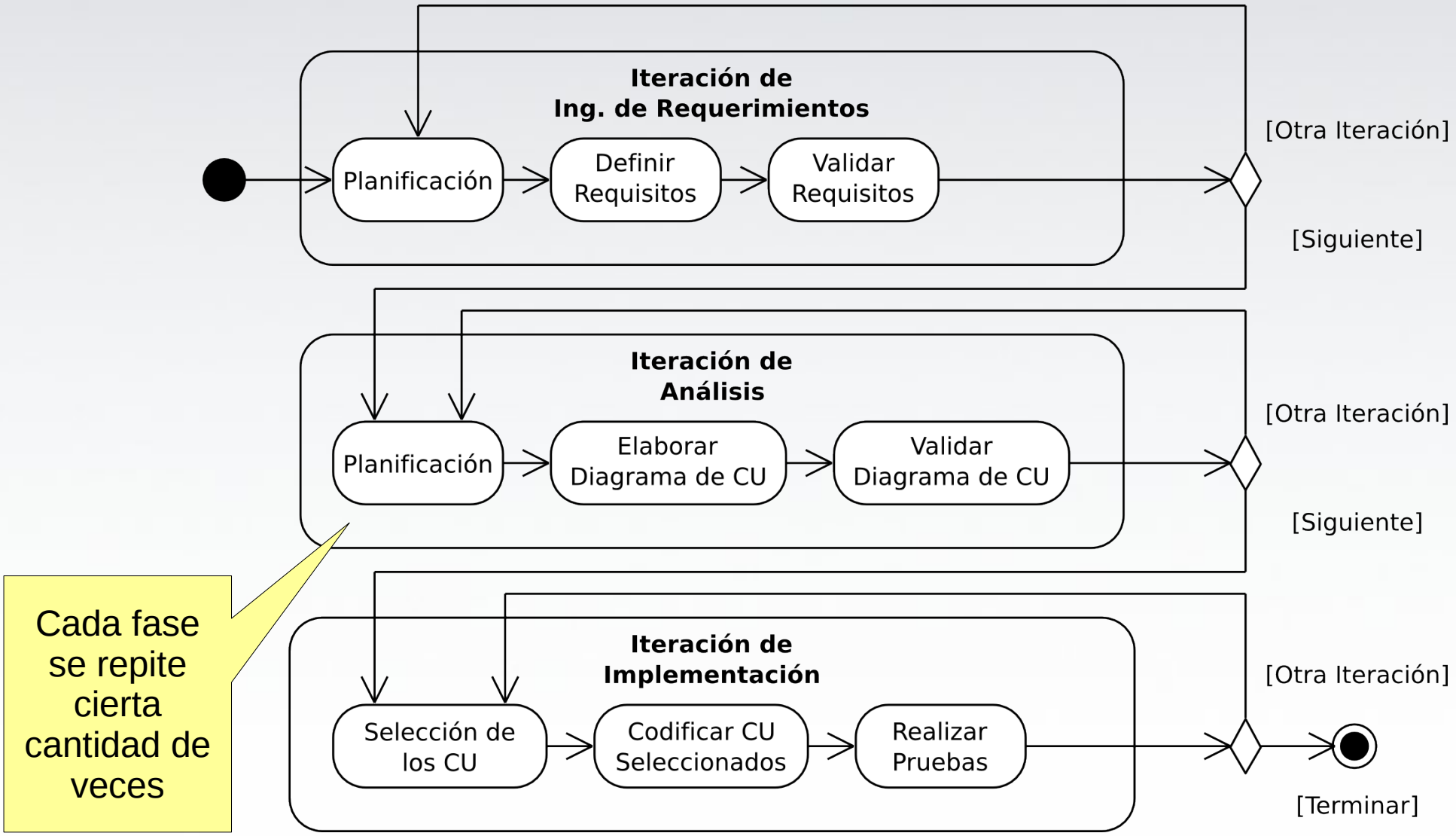
UNIVERSIDAD DE LOS ANDES VENEZUELA

¿un proceso puede ser iterativo e incremental?

...

generalmente si

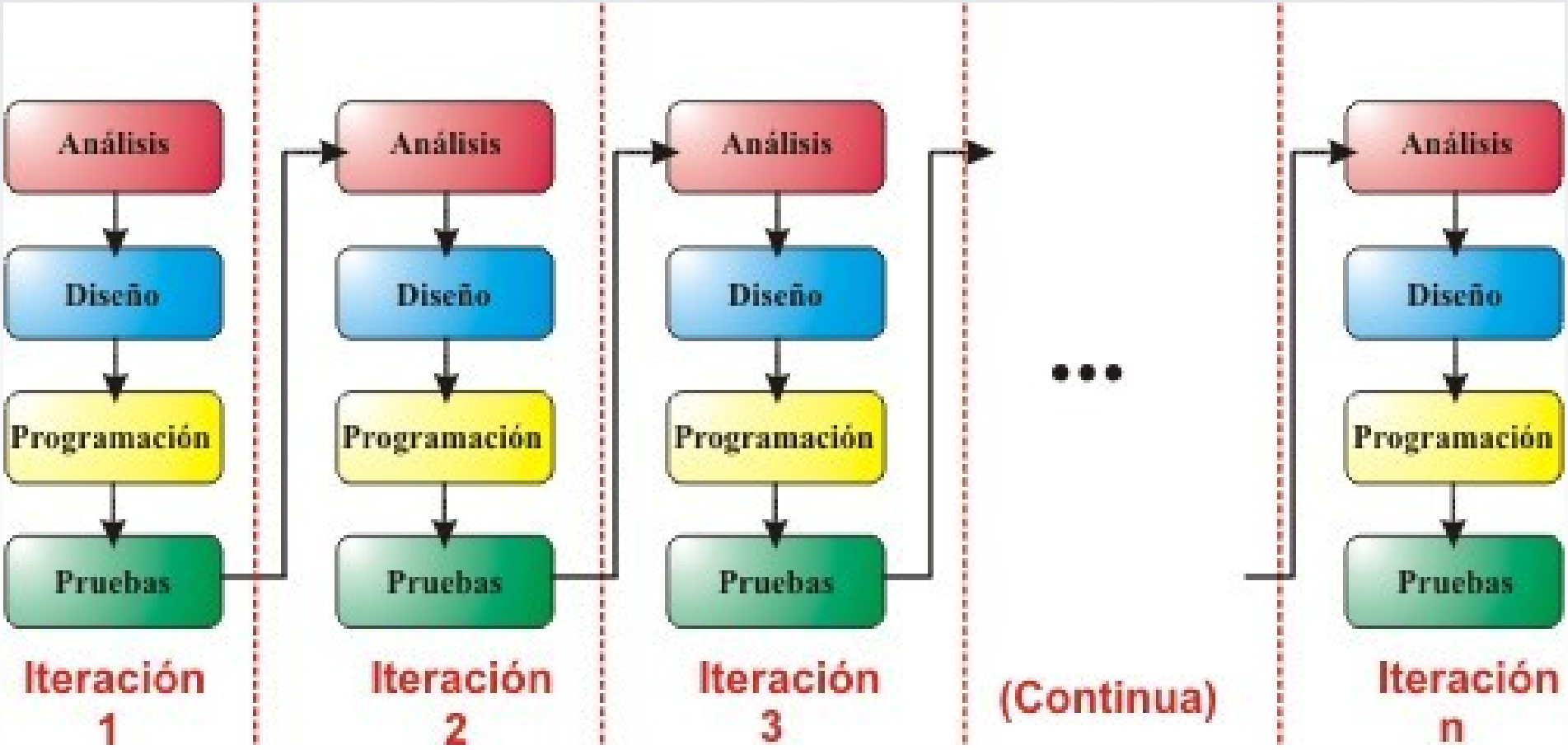
Modelos Incrementales (Modelo Iterativo)



Modelo Iterativo-Incremental

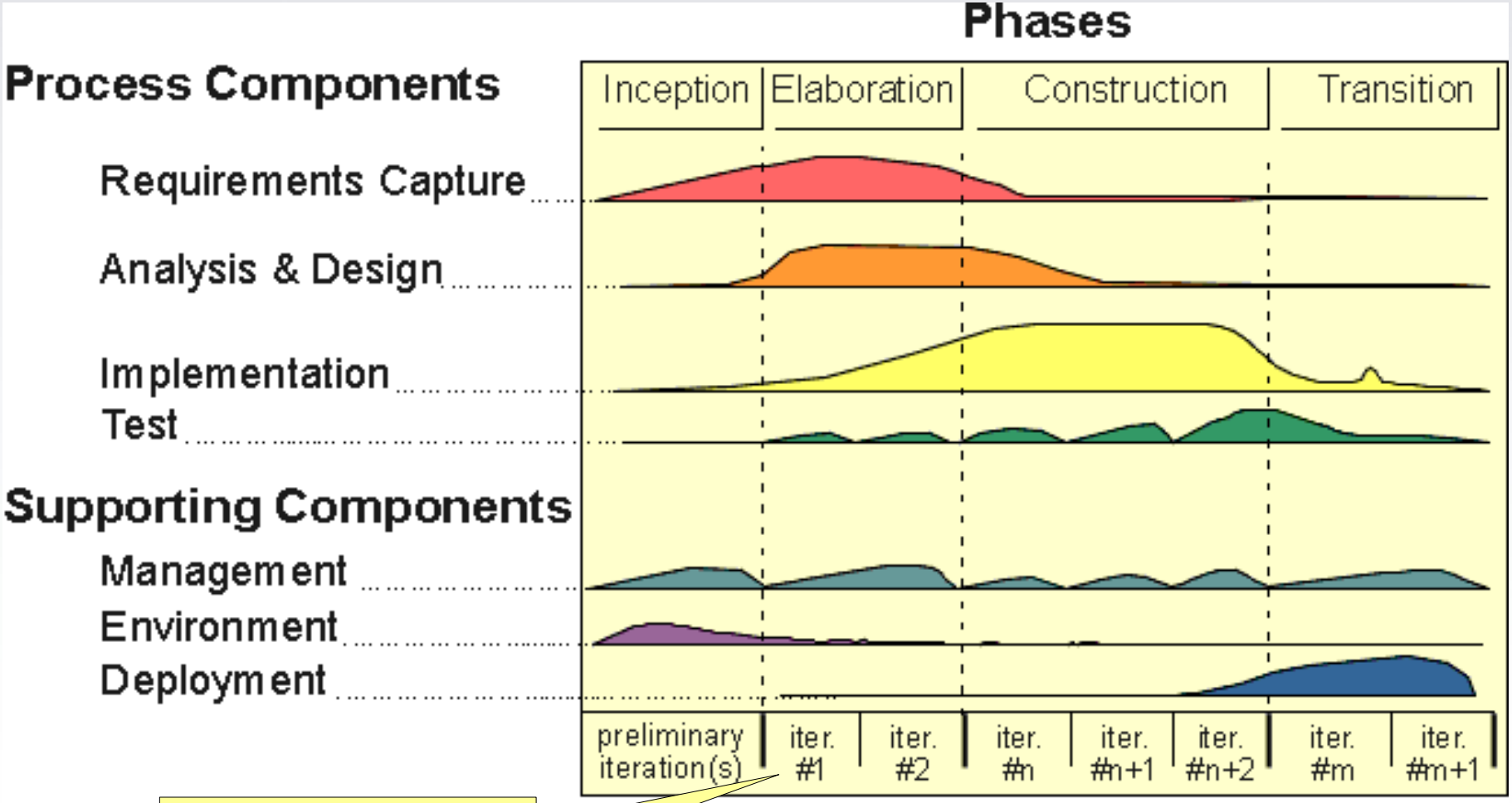
UNIVERSIDAD DE LOS ANDES VENEZUELA

Ojo con los términos y las confusiones en relación a la concepción de los modelos iterativos, incremetales y evolutivos



También se puede iterar sobre todo el proceso de desarrollo, aunque esto está mas asociado a los procesos evolutivos que a los procesos iterativos

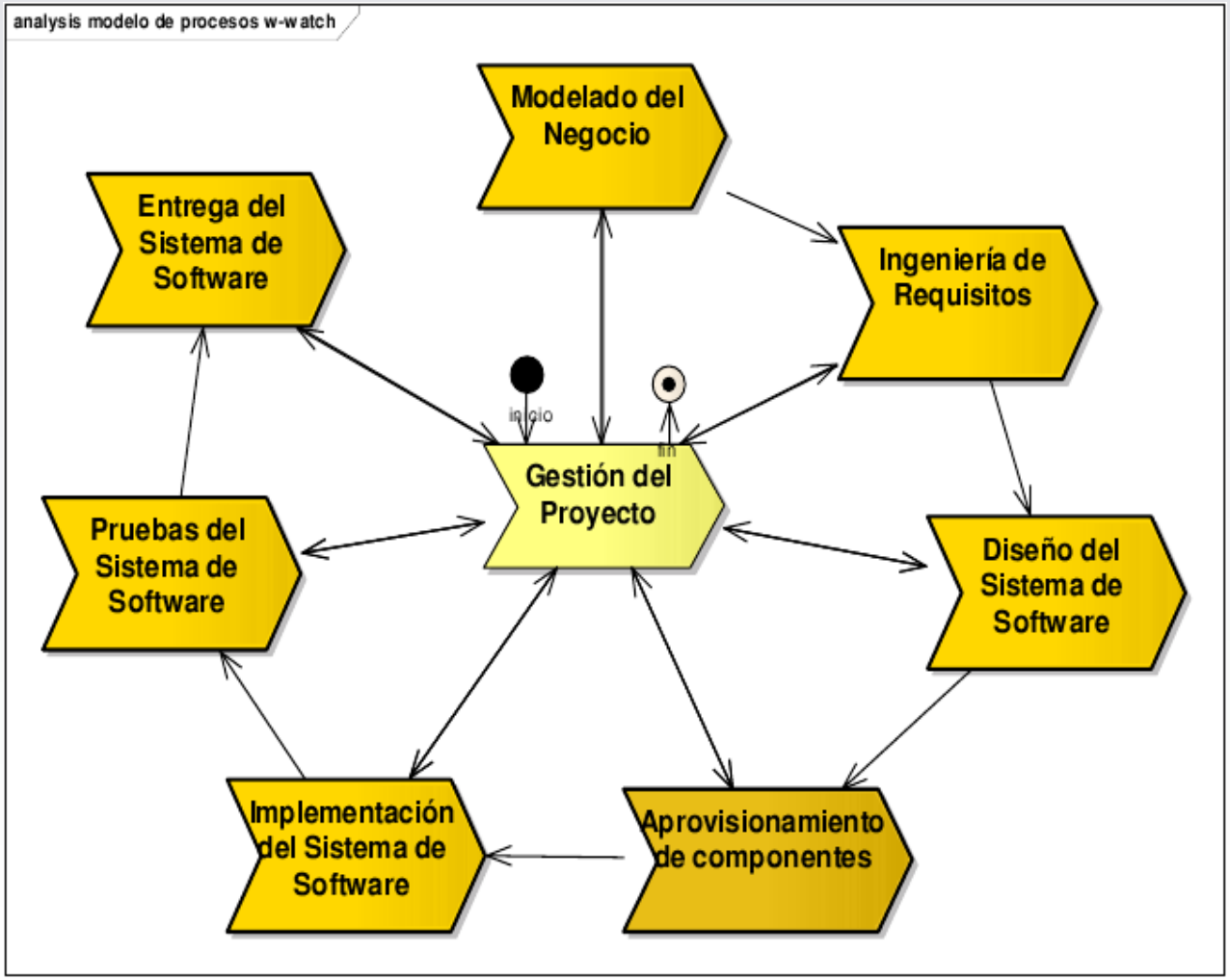
Modelos Incrementales (Modelo Incremental / UP)



Iteraciones para cada una de las ***fases***

La visión de UP (Unified Process)

White Watch, ¿Iterativo o Incremental?



Excelente artículo de Alistair Cockburn sobre desarrollo incremental y desarrollo iterativo:

http://alistair.cockburn.us/Incremental+versus+iterative+development

Básicamente aclara bastante la confusión existente entre los dos términos

UNIVERSIDAD DE LOS ANDES VENEZUELA

Incremental development defined

‘’By “incremental development”, I specifically mean a’’

staging and scheduling strategy ‘’in which the *various parts of the system are developed at different times or rates, and integrated as they are completed.*’’

‘’***It neither implies, requires nor precludes iterative development or waterfall development*** – both of those are rework strategies. The alternative to incremental development is to develop the entire system with a “big bang” integration.‘‘

—

‘’Incremental’’ development helps you improve your process. Each time around the process, you get to change and improve your work habits.

Alistair Cockburn:
http://alistair.cockburn.us/Incremental+versus+iterative+development

UNIVERSIDAD DE LOS ANDES VENEZUELA

Iterative development defined

‘’By “iterative development”, I specifically wish to mean a’’

rework scheduling strategy ‘’in which time is set aside to revise and improve parts of the system.’’

‘’***It does not presuppose incremental development, but works very well with it.*** As shown in the figures, the difference is that an increment may actually ship, whereas an iteration is examined for modification.‘‘

—

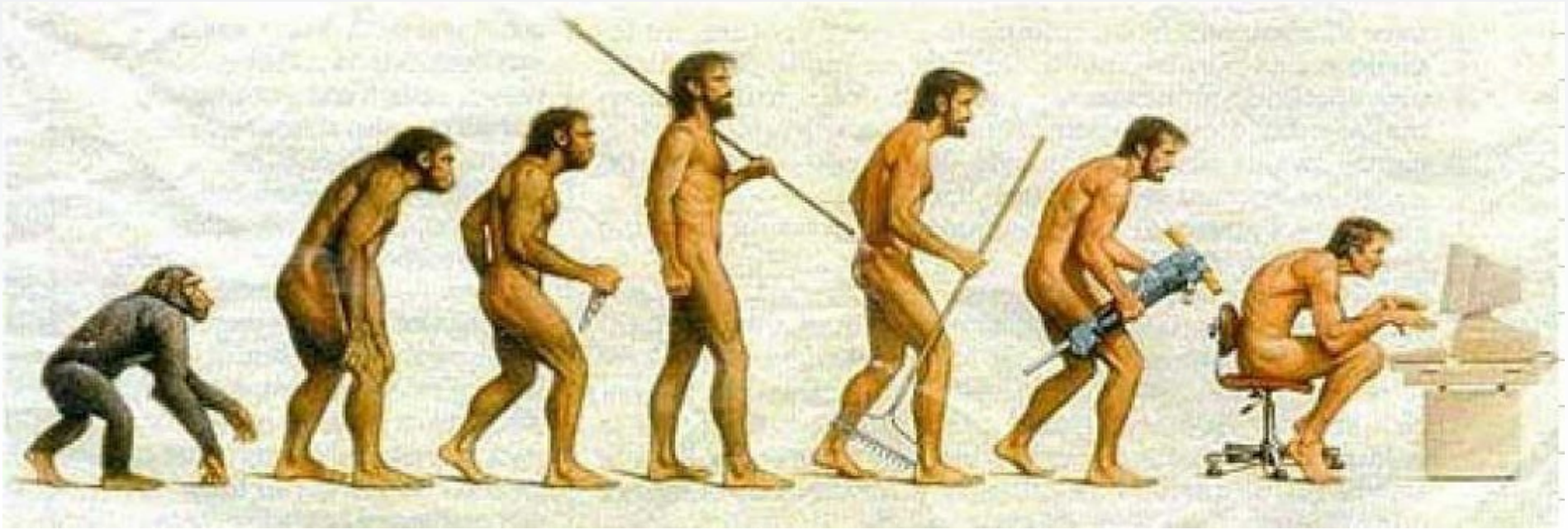
‘’Iterative’’ development helps you improve your product. Each time around the process you get to change and improve the product itself (and maybe some of your work habits)

Alistair Cockburn:
http://alistair.cockburn.us/Incremental+versus+iterative+development

procesos evolutivos y basados en prototipos

Modelos Evolutivos

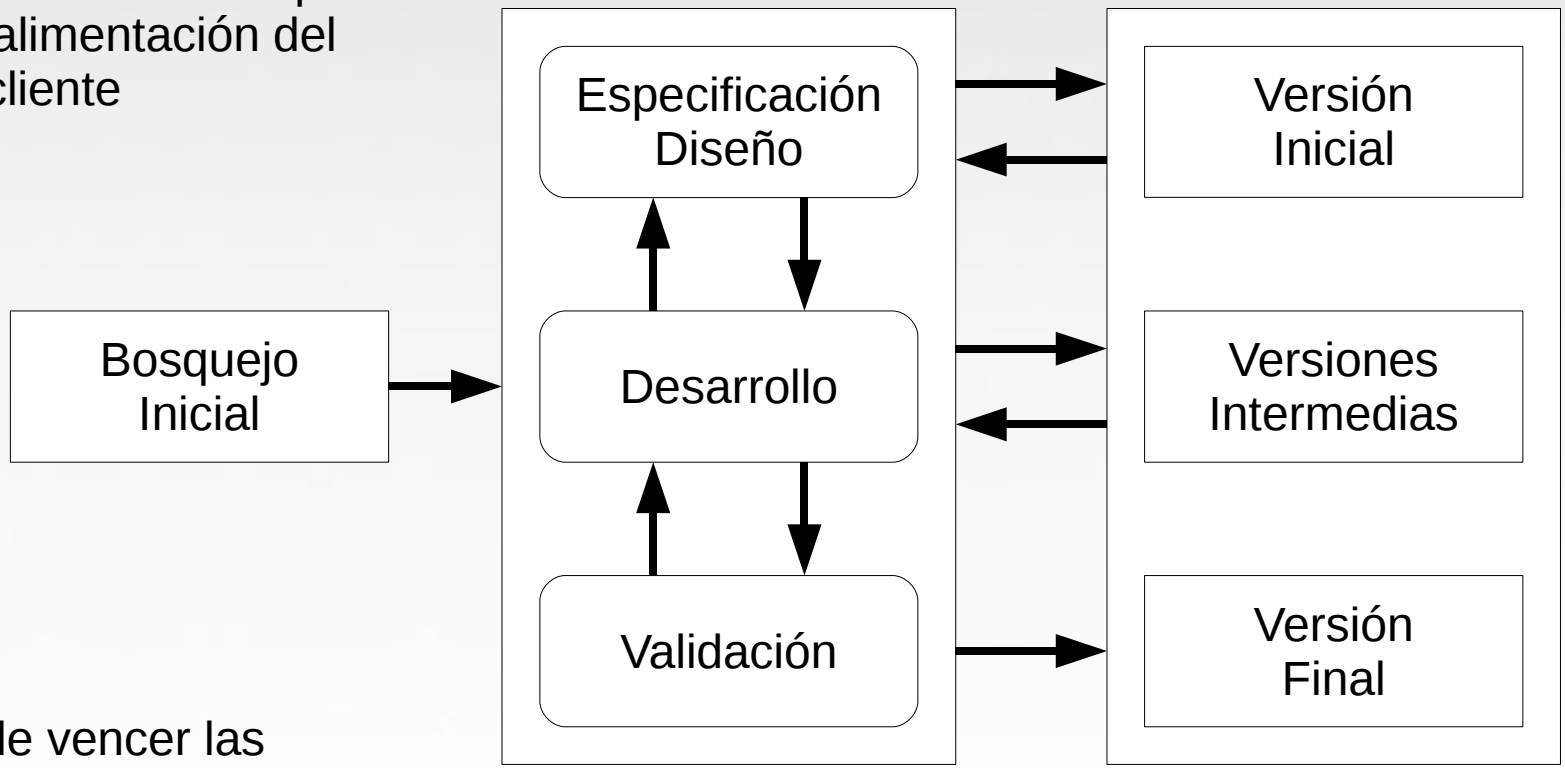
La definición y especificación de requerimientos y el desarrollo de software es un proceso evolutivo que demanda la experimentación previa con algún componente (o la totalidad) del Sistema Programado (Ej. Interfaz U-S, función o subsistema) antes de desarrollar la totalidad del sistema.



¿qué es evolucionar?

Modelos Evolutivos

Logran su objetivo por medio del desarrollo de una serie de prototipos que van evolucionando a medida que se tiene realimentación del cliente



Pretende vencer las limitaciones del modelo en cascada debidas a la deficiente realimentación entre sus fases

Fuente; Sommerville / Ingeniería del Software

¿qué es un prototipo?

Modelos Basados en Prototipos

UNIVERSIDAD DE LOS ANDES VENEZUELA

Prototipos Evolutivos

Poner un sistema a disposición de los usuarios finales. El proceso comienza con una serie de requisitos, se desarrollan una serie de prototipos, se exponen al usuario y se van refinando paso a paso

Prototipos Experimentales

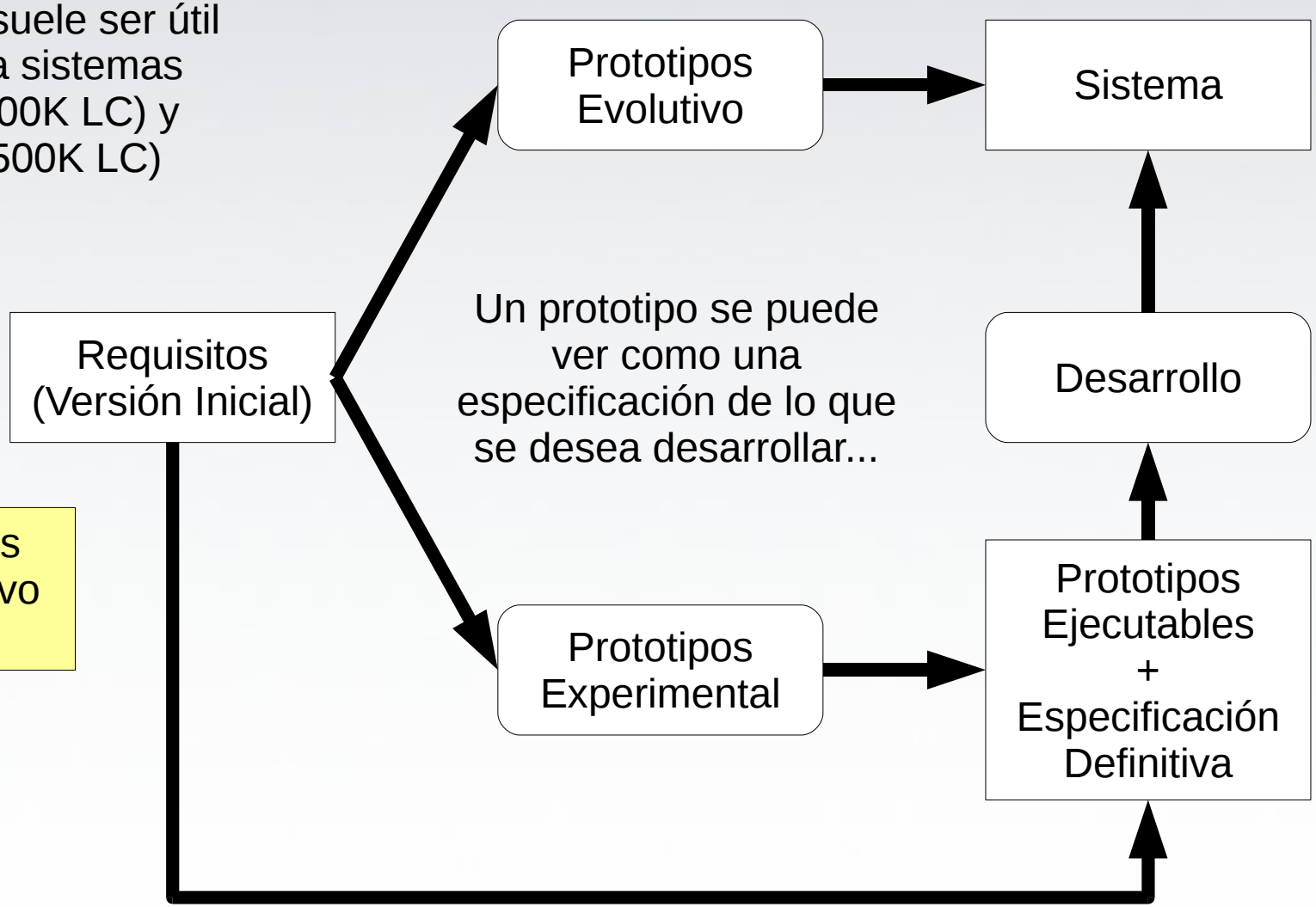
Prototipos Desechables / Exploratorios

Se desarrollan prototipos (que luego se desecharan) para aclarar aspectos particulares de los requerimientos del usuario. Este conocimiento se utilizará para especificar/diseñar/desarrollar la aplicación

Modelos Basados en Prototipos

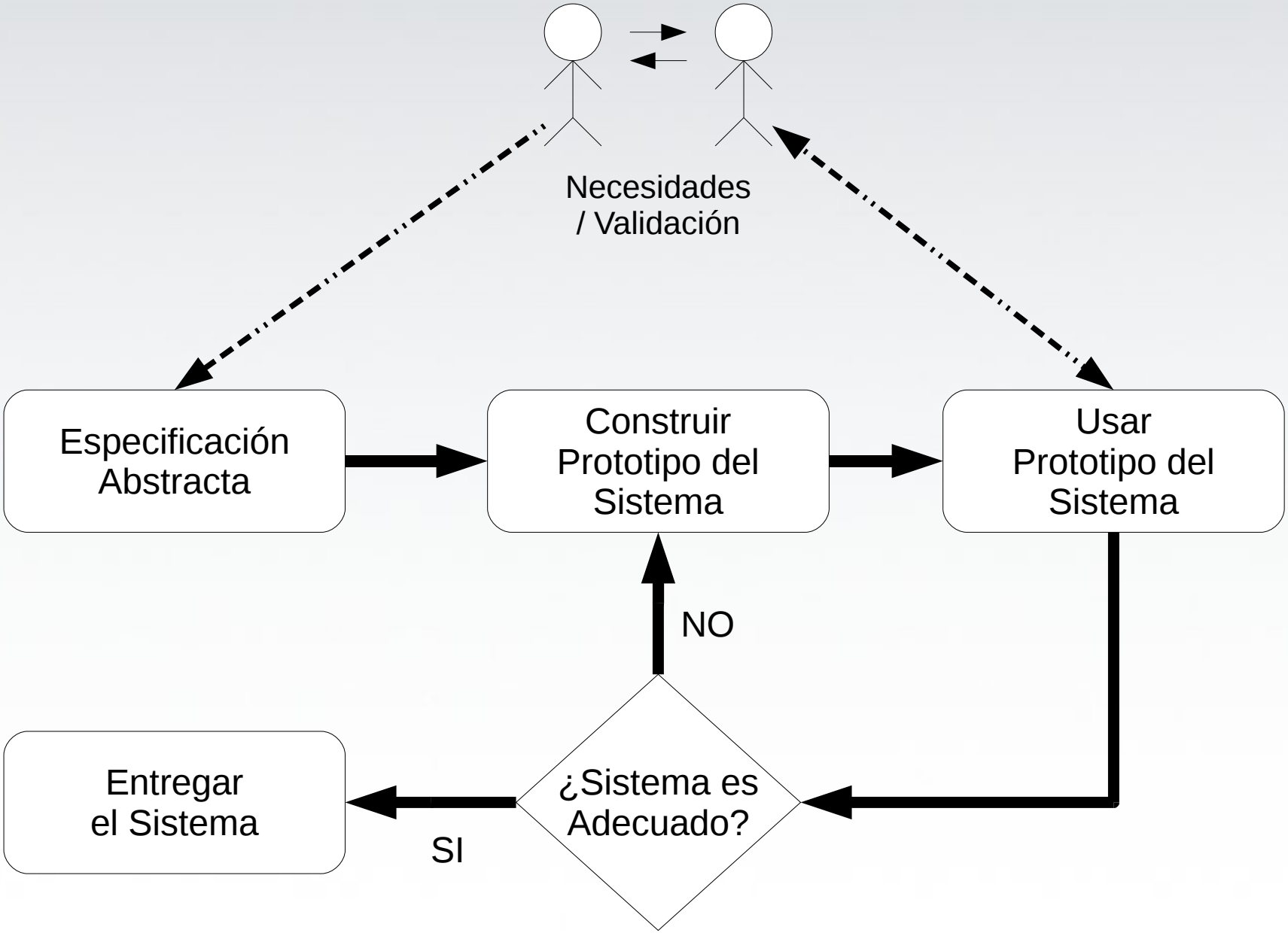
Esta estrategia suele ser útil y práctica para sistemas pequeños (<100K LC) y medianos (<500K LC)

aunque esto es muy, muy relativo también



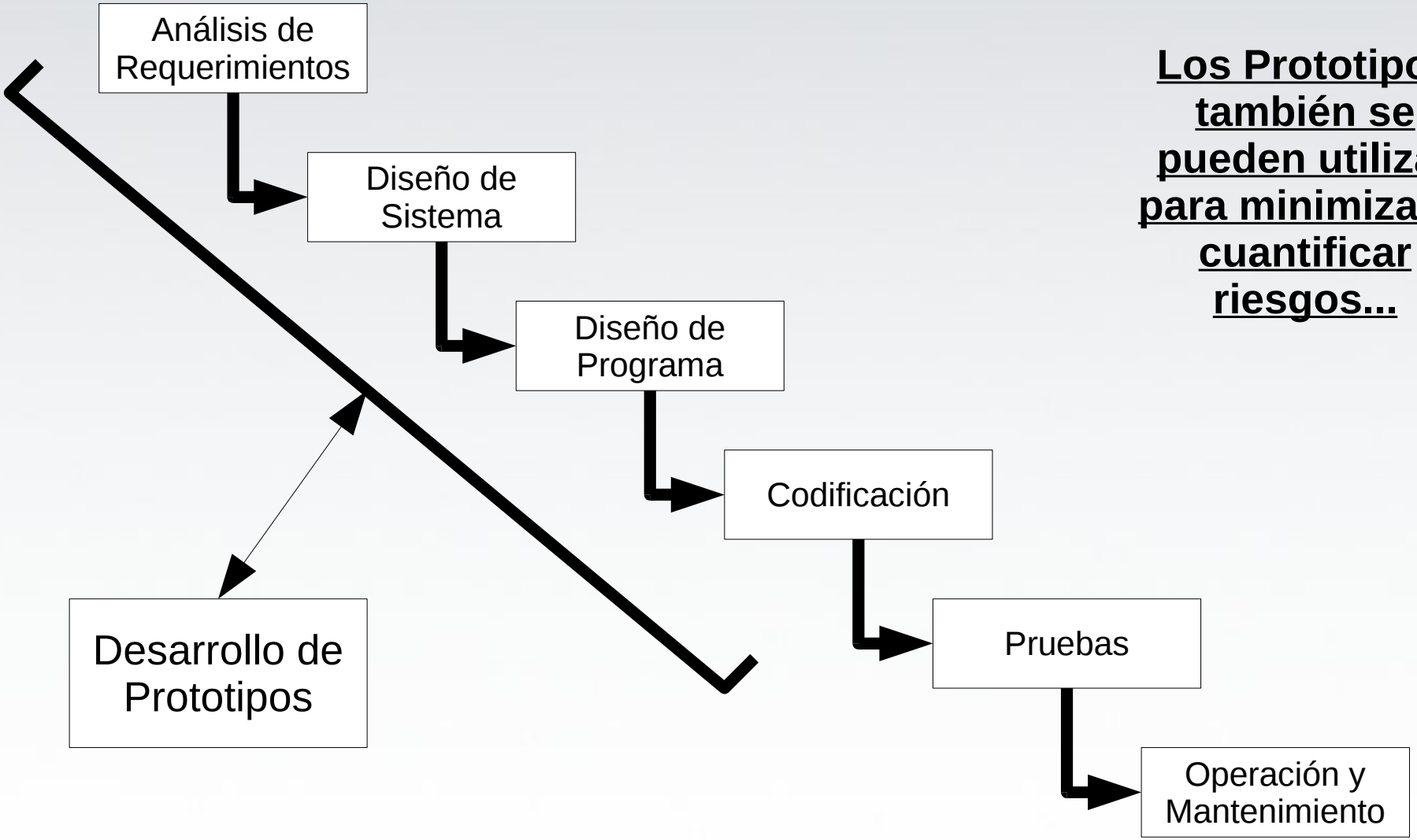
Un prototipo se puede ver como una especificación de lo que se desea desarrollar...

Modelos Basados en Prototipos (No desechable)



Integración del Proceso en Cascada y el Desarrollo Basado en Prototipos

UNIVERSIDAD DE LOS ANDES VENEZUELA



Los Prototipos también se pueden utilizar para minimizar o cuantificar riesgos...

Integración del modelo de prototipos con el modelo de cascada

(Adaptado de Pfleeger, 1998)

Modelos Basados en Prototipos

UNIVERSIDAD DE LOS ANDES VENEZUELA

Útiles en sistemas (O aspectos de un sistema) en los que no es posible inicialmente (o es difícil) desarrollar una especificación. Ej: Sistemas de Inteligencia artificial, Interfaces de Usuario, etcétera

Usualmente se basan en técnicas que permiten obtener versiones rápidas del sistema, que se pueden poner en operación tan rápido como sea posible: Rapid Application Development (RAD)

Los requisitos no funcionales no se prueban / determinan adecuadamente en el prototipo (Ej. seguridad, rendimiento, u otros

Práctica generalmente para sistemas pequeños / medianos (<100K o <500K líneas de código)

Modelos Basados en Prototipos

UNIVERSIDAD DE LOS ANDES VENEZUELA

El Proceso, la evolución, el avance, es difícil de medir. No siempre es fácil responder a la pregunta: ¿Cuanto falta para terminar el sistema? (*El proceso no siempre es visible*)

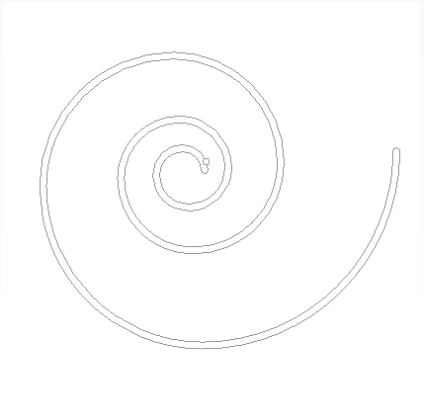
Los cambios continuos pueden provocar la destrucción de la estructura del sistema

Es posible que no se puedan realizar verificaciones formales, debido a que es posible que no exista una especificación formal y/o documentación bien definida

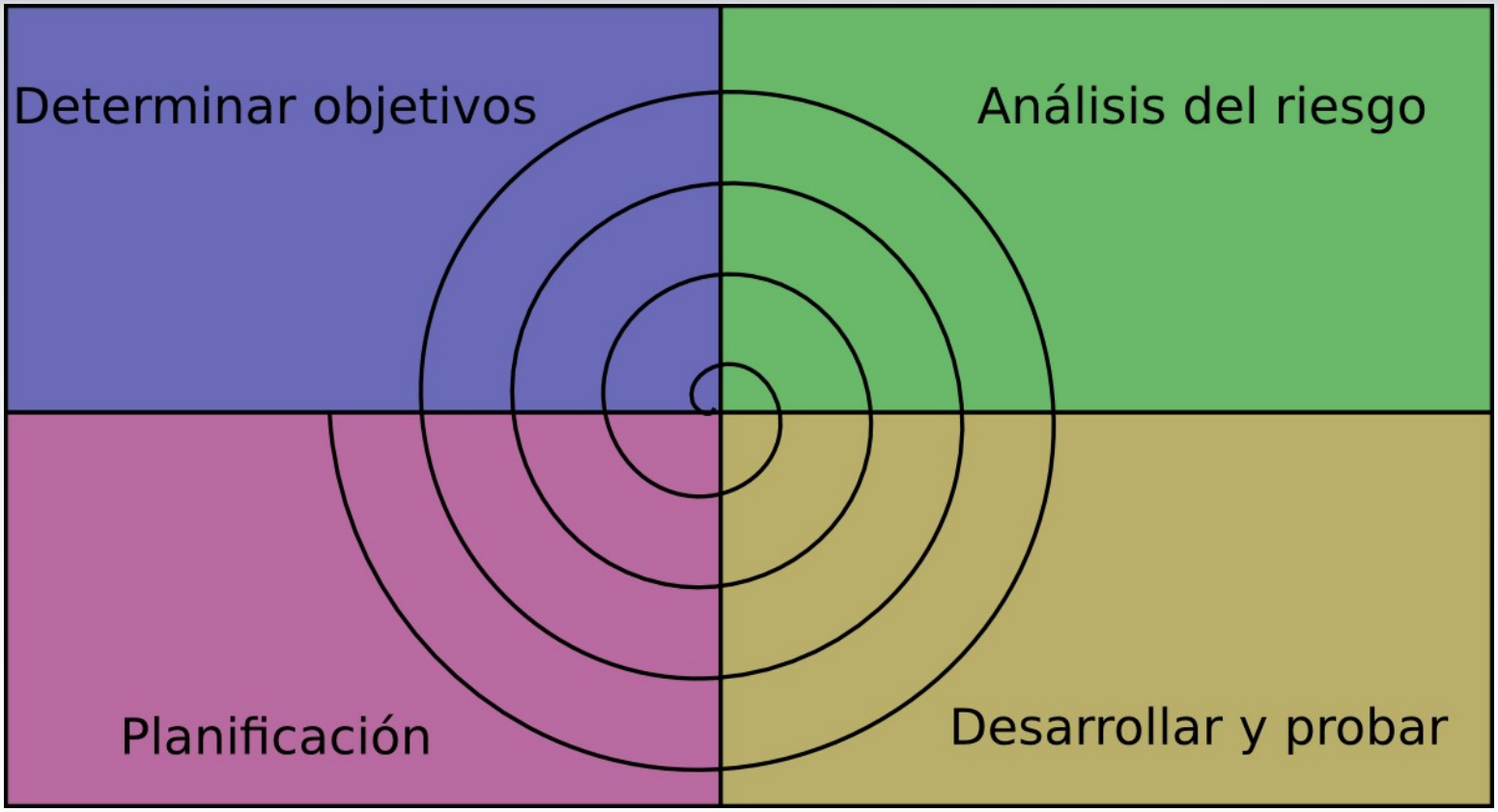
Es difícil establecer contratos, una implementación no tiene carácter de contrato

¡Excelentes para mitigar riesgos y hacer una negociación justa con el cliente!

...por cierto...



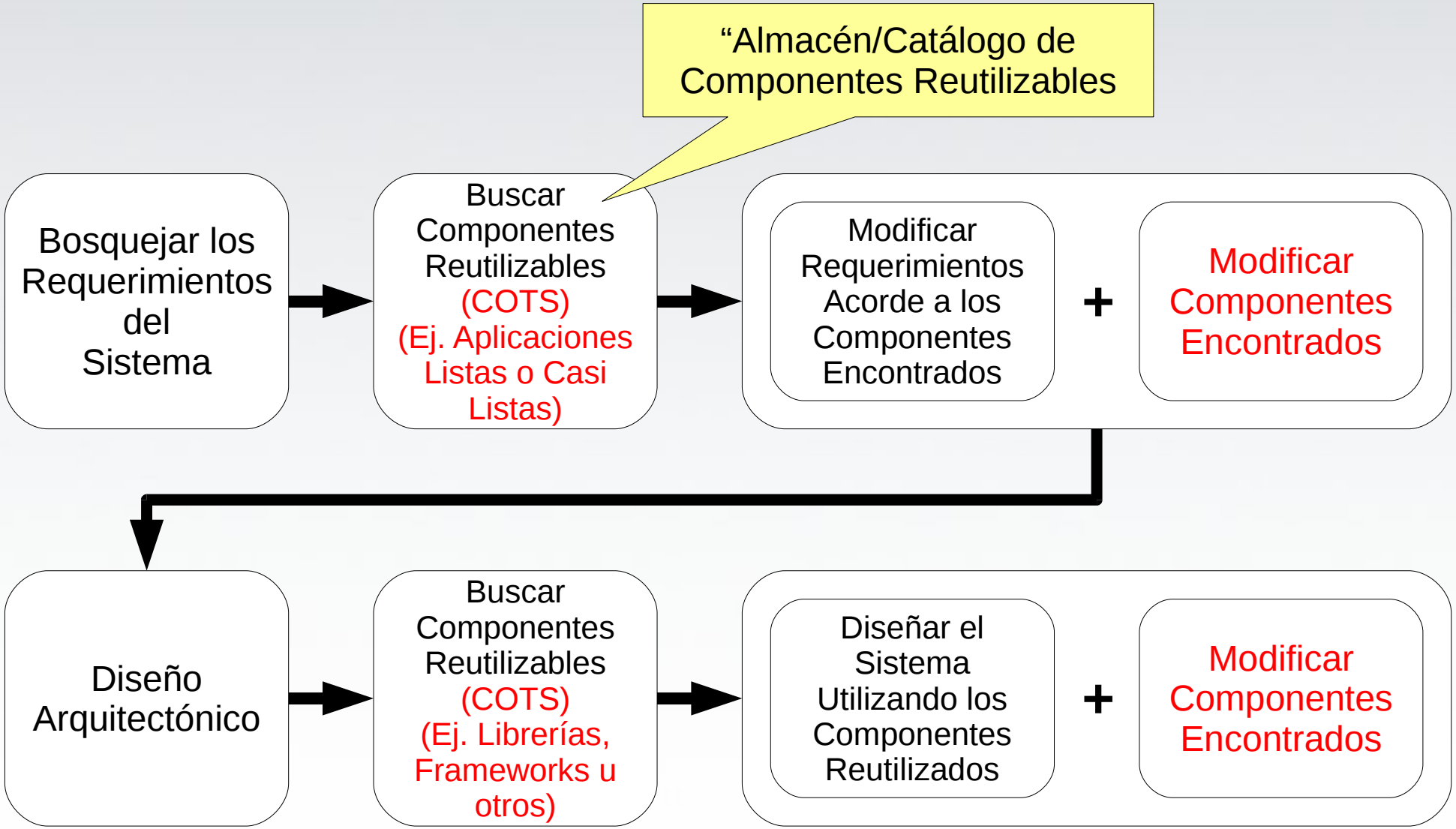
Modelo de Riesgos o de Espiral



¿Será el proceso en espiral incremental, iterativo, evolutivo, etc?

procesos basados en la reutilización de componentes

Desarrollo Basado en Reutilización (Componentes)



COTS: Commercial Off the Shelf

Fuente; Sommerville / Ingeniería del Software (Excepto lo rojo)

Desarrollo Basado en Reutilización (Componentes)

UNIVERSIDAD DE LOS ANDES VENEZUELA

El costo del sistema se puede reducir notablemente debido a la reutilización

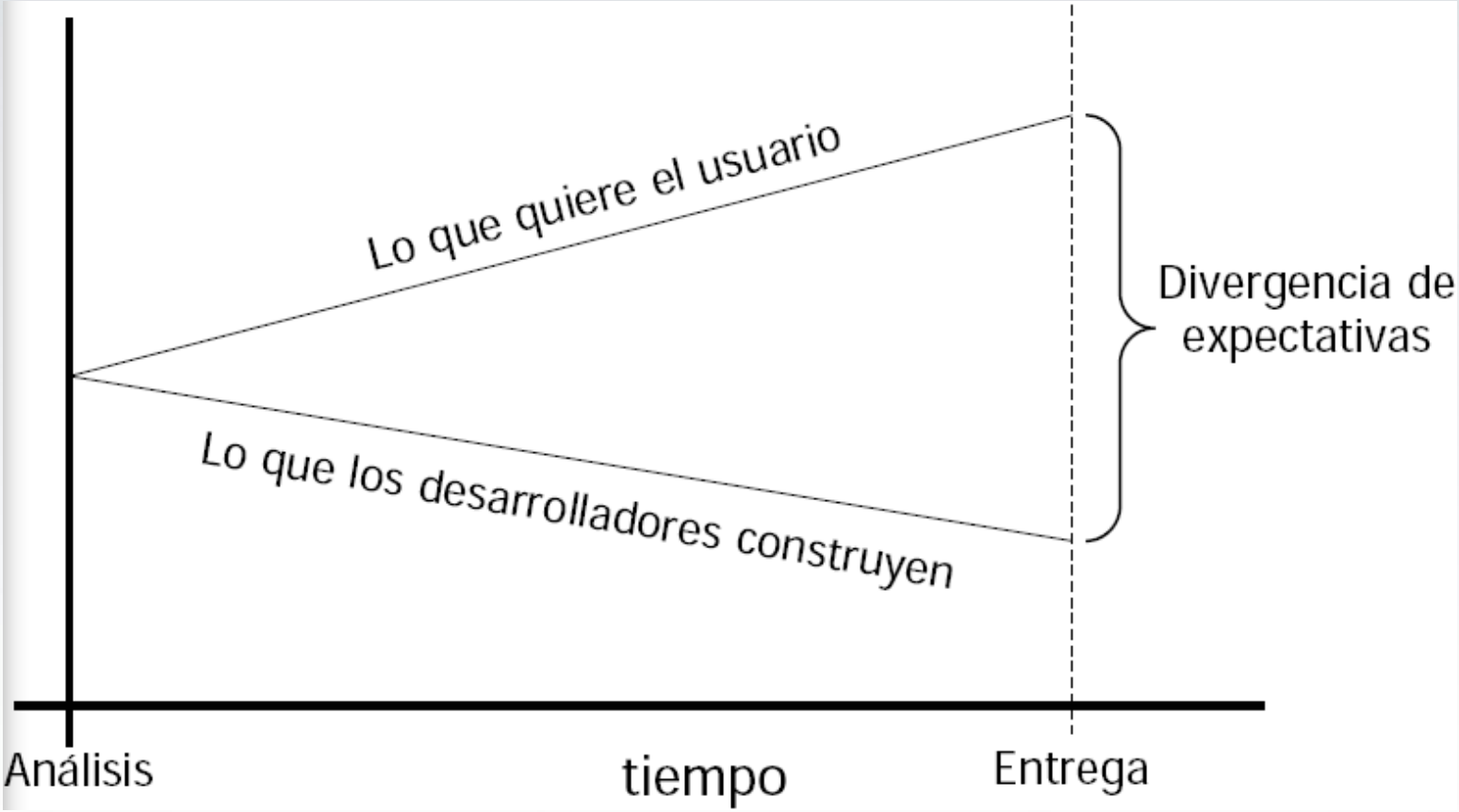
El sistema se construye “uniendo” componentes existentes

Se está limitado a los componentes existentes, es necesario negociar los requerimientos en base a estos, o modificar los componentes (lo que no siempre es fácil) para lograr satisfacerlos (o ambas cosas)

Se necesita todo un “armazón” o un “lenguaje” para poder unir los componentes

importancia de involucrar al usuario / cliente en el proceso de desarrollo

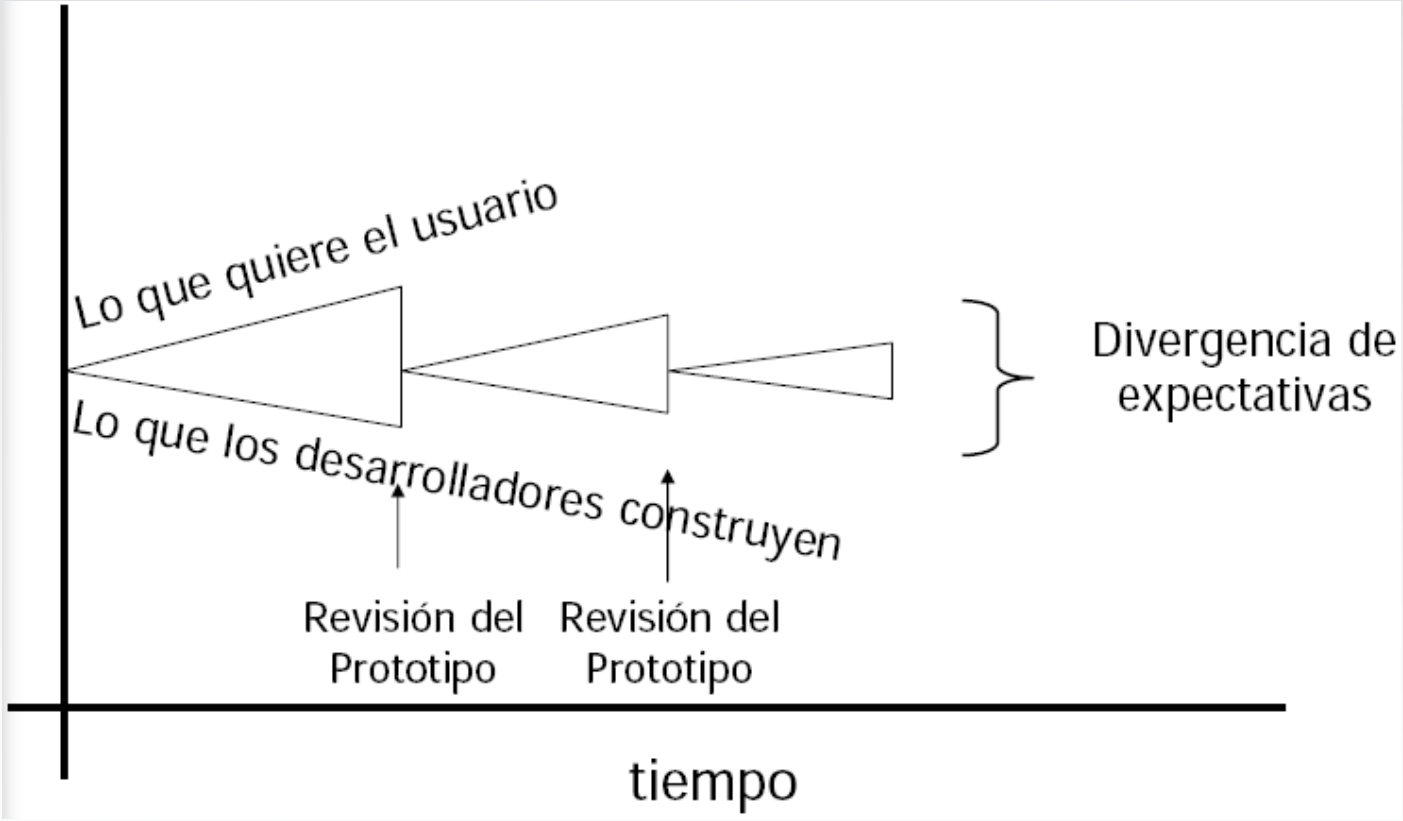
¿Proceso en Cascada?



Diferencia entre las expectativas del usuario y los resultados producidos

el precio que se paga si no se involucra al usuario en el proceso de desarrollo

Modelos Basados en Prototipos (Y en general, modelos iterativos)



Diferencia entre las expectativas del usuario y los resultados producidos

(muchos métodos ágiles logran esto también involucrando al cliente lo más que sea posible)

UNIVERSIDAD DE LOS ANDES VENEZUELA

modelos ágiles (XP)

http://www.agile-software-development.com/2007/05/beauty-of-not-doing-agile-development.html

Modelos ágiles (XP)

UNIVERSIDAD DE LOS ANDES VENEZUELA

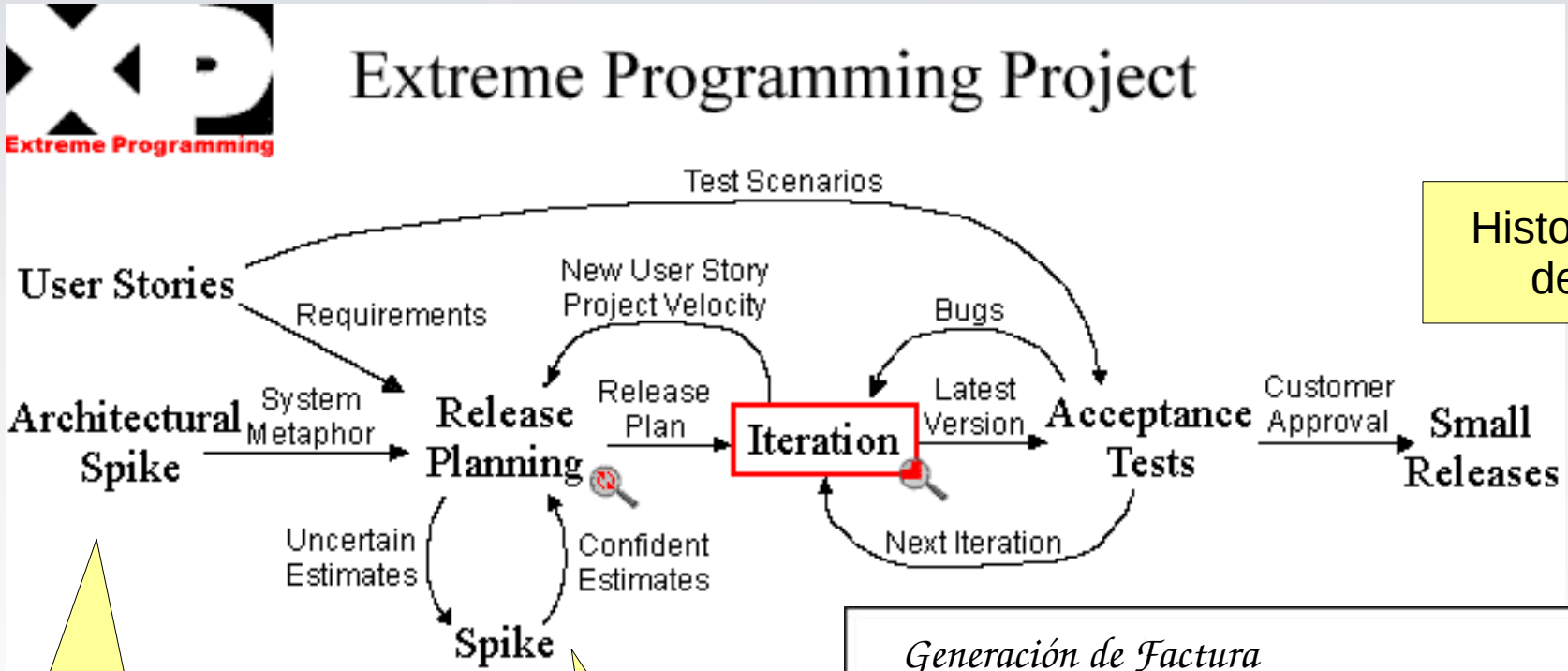
XP (eXtreme Programing): Es una estratégia de desarrollo de software creada hace aproximadamente unos diez años que ha causado un gran revuelo entre el colectivo de programadores del mundo

Kent Beck, su autor, es un programador que ha trabajado en múltiples empresas. Actualmente trabaja en la conocida empresa automovilística DaimlerChrysler

Con sus teorías ha conseguido el ***respaldo*** de gran parte de la industria del software y el ***rechazo*** de otra parte

http://www.extremeprogramming.org

Modelos ágiles (XP)



Historia (cuento) de usuario

Versión inicial de la arquitectura

Prototipo, prueba o experimento rápido pensado para reducir el riesgo

Generación de Factura
El usuario introduce la información del cliente. Si el cliente ya está registrado con sólo introducir la cédula se deben cargar sus datos. Luego se ingresan los elementos a facturar y las cantidades de cada elemento. Finalmente el sistema registra la factura y es capaz de imprimirla en la impresora local asociada al terminal del usuario

Modelos ágiles (XP / Características)

UNIVERSIDAD DE LOS ANDES VENEZUELA

1) ***El desarrollo del plan:*** Determinar rápidamente el alcance de la siguiente iteración / entrega en base a las prioridades del negocio (cliente) y los estimados técnicos. Estar dispuestos a cambiar el plan a medida que es necesario.

2) ***Liberar mucho, en incrementos pequeños:*** Poner el sistema en producción los más rápido posible (el mínimo necesario) y desarrollar las siguientes versiones con el ciclo lo mas corto posible.

3) ***Diseño simple:*** Mantener el diseño lo más simple posible (KISS: Keep it Simple Stup$%#id), concentrarse en el presente y no en el futuro
(YAGNI: You ain't going to need it)

Modelos ágiles (XP / Características)

UNIVERSIDAD DE LOS ANDES VENEZUELA

4) ***Pruebas unitarias continuas:*** Sirven para evitar que los programadores se equivoquen, para evitar las “parcelas” de código y para validar constantemente la aplicación. Los clientes también pueden escribir pruebas para validar / demostrar ciertas características del sistema.

5) ***Programación en parejas:*** Todo el código a ponerse en producción es escrito en parejas. ***¿Sabe usted por que?***

6) ***Propiedad colectiva:*** Nadie es dueño de ninguna clase, de ningún artefacto, de ninguna parte del código.

7) ***Integración continua:*** Las características del sistema se desarrollan y se integran a diario. Luego se corren las pruebas y se verifica que la aplicación corra correctamente.

Modelos ágiles (XP / Características)

UNIVERSIDAD DE LOS ANDES VENEZUELA

8) ***40 horas a la semana:*** Nadie. ¡NADIE! Trabaja horas extra. ***¿Sabe usted porque?***

9) ***El cliente involucrado en el ambiente de desarrollo:*** El cliente (o un representante) es un miembro más del equipo de desarrollo.

10) ***Estándares de codificación:*** Se definen estándares adecuados de codificación y se respetan. Sobre todo aquellos que enfatizan la “auto-documentación” y adecuada documentación del código.

Modelos ágiles (XP / Características)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Planning

- User stories are written.
- Release planning creates the release schedule.
- Make frequent small releases.
- The project is divided into iterations.
- Iteration planning starts each iteration.

Testing

- All code must have unit tests.
- All code must pass all unit tests before it can be released.
- When a bug is found tests are created.
- Acceptance tests are run often and the score is published.

Managing

- Give the team a dedicated open work space.
- Set a sustainable pace.
- A stand up meeting starts each day.
- The Project Velocity is measured.
- Move people around.
- Fix XP when it breaks.

Designing

- Simplicity.
- Choose a system metaphor.
- Use CRC cards for design sessions.
- Create spike solutions to reduce risk.
- No functionality is added early.
- Refactor whenever and wherever possible.

Coding

- The customer is always available.
- Code must be written to agreed standards.
- Code the unit test first.
- All production code is pair programmed.
- Only one pair integrates code at a time.
- Integrate often.
- Set up a dedicated integration computer.
- Use collective ownership.

Modelos ágiles (XP / Valores)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Simplicidad: Es la base de la programación extrema. La idea es simplificar el diseño lo más posible para agilizar el desarrollo y facilitar el mantenimiento.

Comunicación: Se realiza de diferentes formas:

1) Para los programadores el código comunica mejor. La ***simplicidad del código*** hace que este sea legible. Es mejor tener ***"código autodocumentado"*** que código con grandes cantidades de documentación, ya que la ***documentación*** corre el riesgo de quedar ***desfasada*** con el código a medida que este es modificado.

2) ***Las pruebas unitarias comunican***, ya que describen el diseño de las clases y métodos al mostrar ejemplos concretos de como usar su funcionalidad.

3) Los programadores se comunican constantemente gracias a la programación en parejas.

4) La comunicación con el cliente es fluida ya que el cliente es parte del equipo.

Modelos ágiles (XP / Valores)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Retroalimentación (feedback): El cliente está integrado en el proyecto de modo que su opinión sobre el estado del proyecto se conoce en tiempo real. Como las iteraciones son muy cortas (2-4 semanas) se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante

Respeto: Todo el mundo recibe y siente el respeto que merece como miembro valioso del equipo de desarrollo. Todos en el equipo aportan valor, aun si es simple entusiasmo. Los desarrolladores respetan la experiencia de sus clientes y viceversa. La gerencia respeta el derecho del equipo de aceptar la responsabilidad y recibir la autoridad sobre su propio trabajo

Coraje o Valentía: Para los gerentes, muchas de las prácticas de XP pueden parecer poco intuitivas o inclusive erradas (programación en parejas, simplicidad, no pensar en la flexibilidad a futuro, entre otras). Es necesario mucho “coraje” para aceptarlas y vencer este prejuicio

advertencia

La Programación Extrema (XP) y otros métodos no significan desarrollar "sin método"

Los métodos ágiles requieren en el fondo mucha disciplina para poder ejecutarlos y mantener el orden de forma satisfactoria

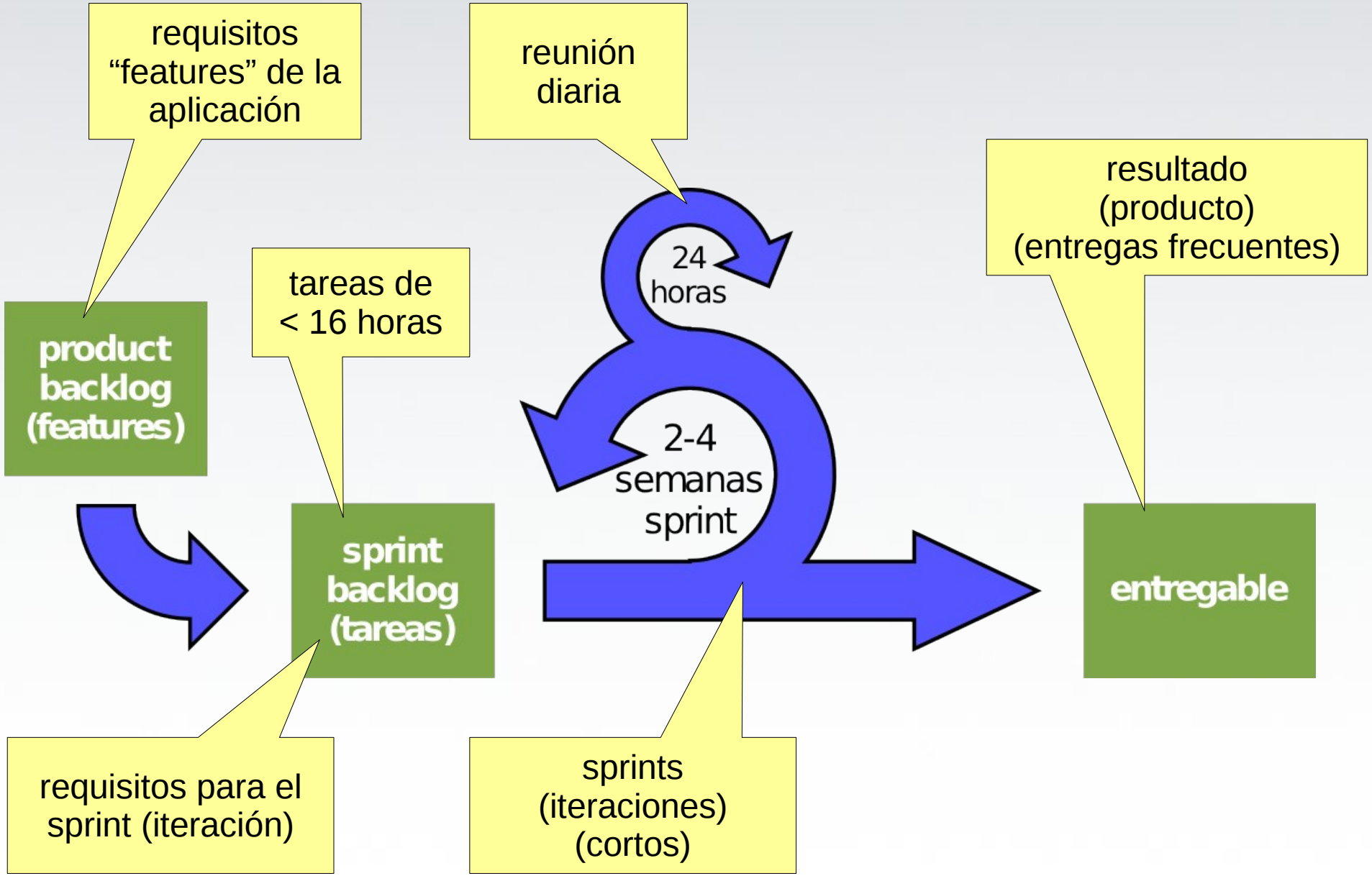
modelos ágiles (scrum)

http://www.agile-software-development.com/2007/05/beauty-of-not-doing-agile-development.html

advertencia

las siguientes transparencias han sido tomadas (¿mutiladas?) de la clase de scrum que vimos al comienzo del semestre, lo que viene es un simple resumen

Modelos Ágiles (SCRUM / Proceso)



Historias de Usuarios (SCRUM / Requisitos)

UNIVERSIDAD DE LOS ANDES VENEZUELA

Los requisitos del producto se capturan teniendo en cuenta la visión del cliente y del usuario

Para ello se utilizan ***historias de usuario***, que son unas sencillas tarjetas en las que se recoge de forma esquemática, sencilla y en un lenguaje claro una interacción entre el usuario y el sistema

Generación de Factura
El usuario introduce la información del cliente. Si el cliente ya está registrado con sólo introducir la cédula se deben cargar sus datos. Luego se ingresan los elementos a facturar y las cantidades de cada elemento.
Finalmente el sistema registra la factura y es capaz de imprimirla en la impresora local asociada al terminal del usuario

Historias de Usuarios (SCRUM / Requisitos)

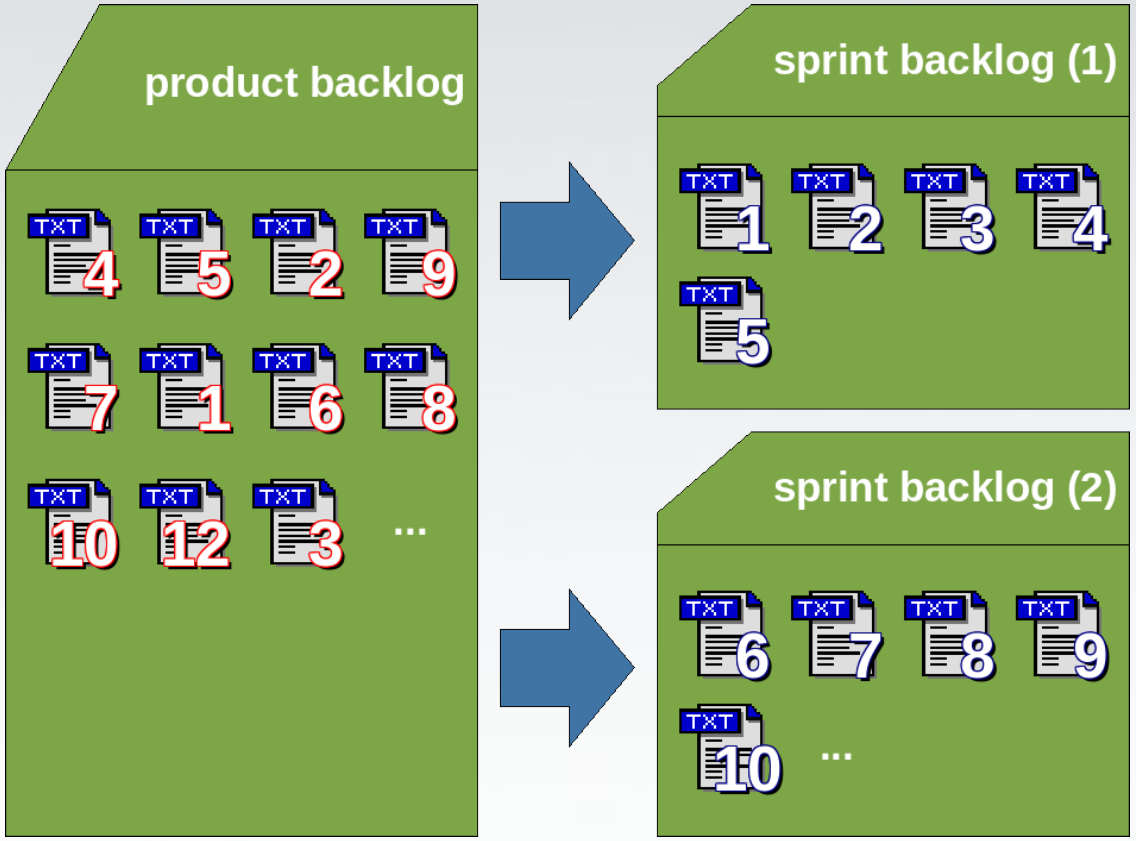
UNIVERSIDAD DE LOS ANDES VENEZUELA

Generación de Factura
El usuario introduce la información del cliente. Si el cliente ya está registrado con sólo introducir la cédula se deben cargar sus datos. Luego se ingresan los elementos a facturar y las cantidades de cada elemento. Finalmente el sistema registra la factura y es capaz de imprimirla en la impresora local asociada al terminal del usuario

Las historias de usuario sirven de "***recordatorio***" de un grupo de características que es necesario implementar en el sistema.

Antes de implementar una característica se produce una discusión con el usuario y ***se refina y extiende la información de la historia de usuario***

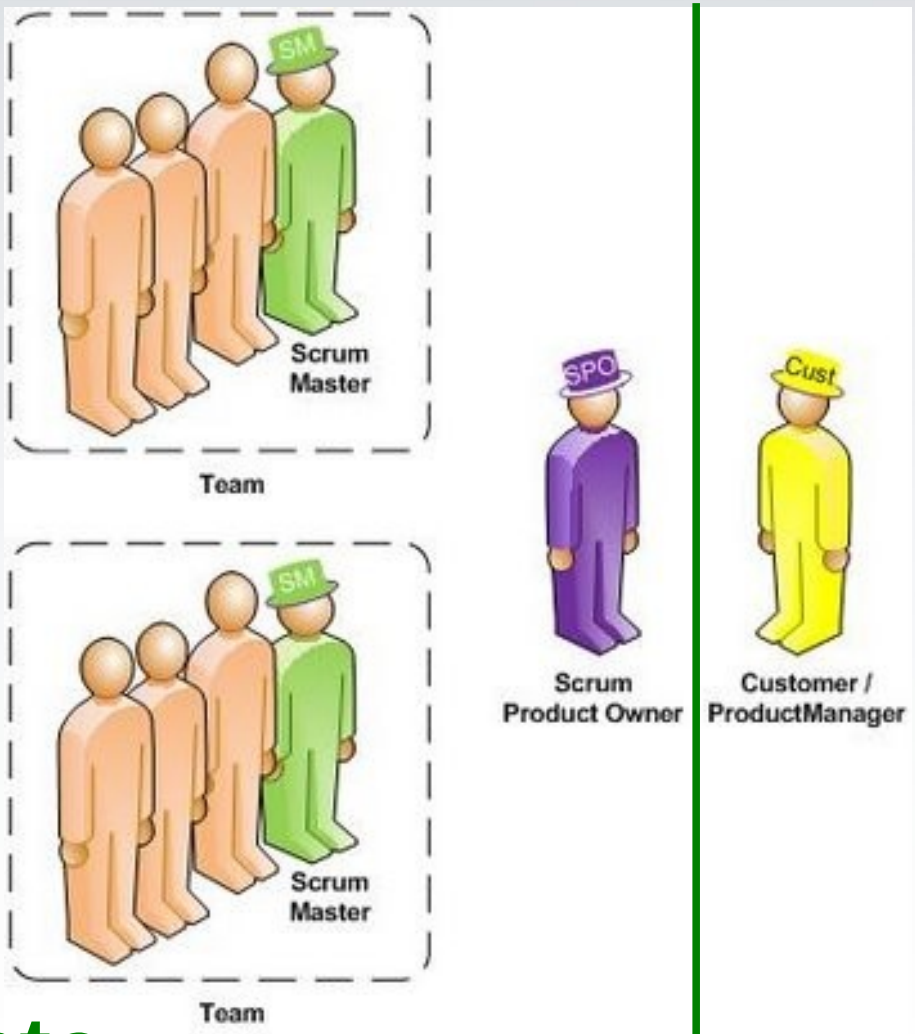
Modelos Ágiles (SCRUM / Requisitos)



Los requisitos del **product backlog** se priorizan y se asignan a los distintos sprints planificados, es decir, al **sprint backlog** de cada sprint

ese cuento ha sido inmortalizado de muchas formas

SCRUM (Roles)



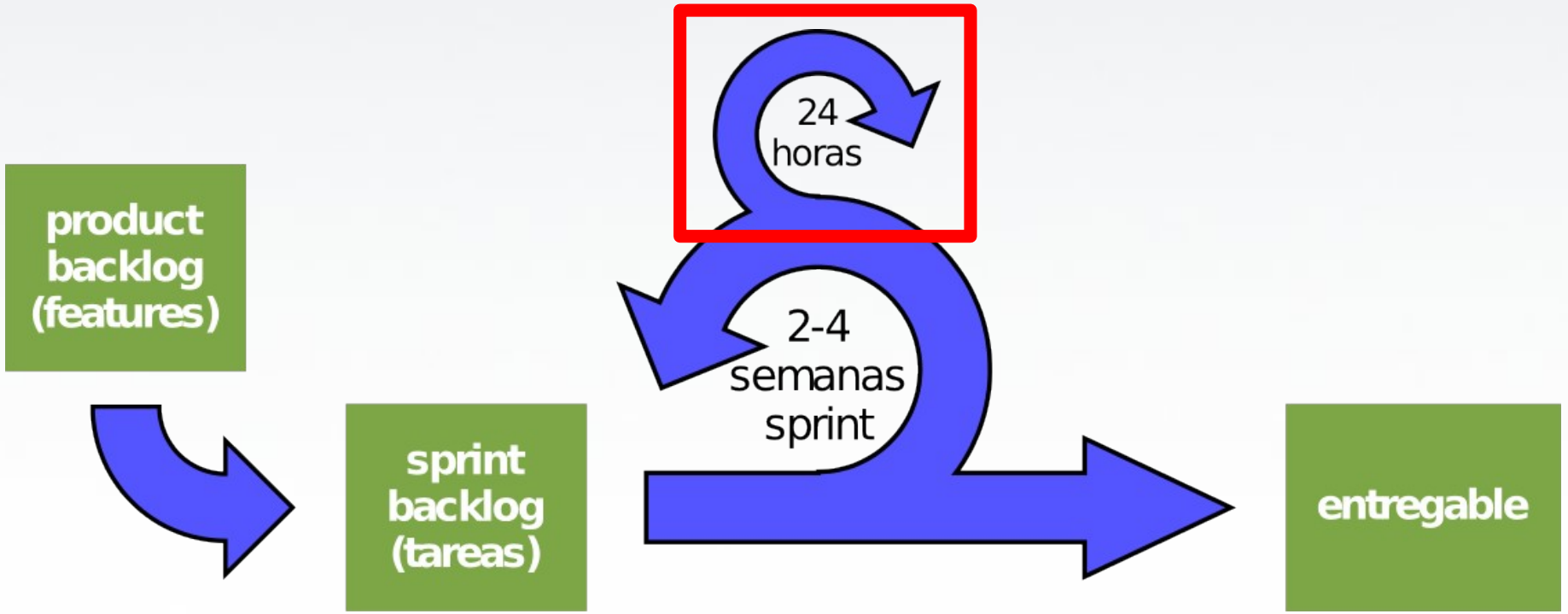
cerdos (realmente comprometidos)

pollos (involucrados)

UNIVERSIDAD DE LOS ANDES VENEZUELA

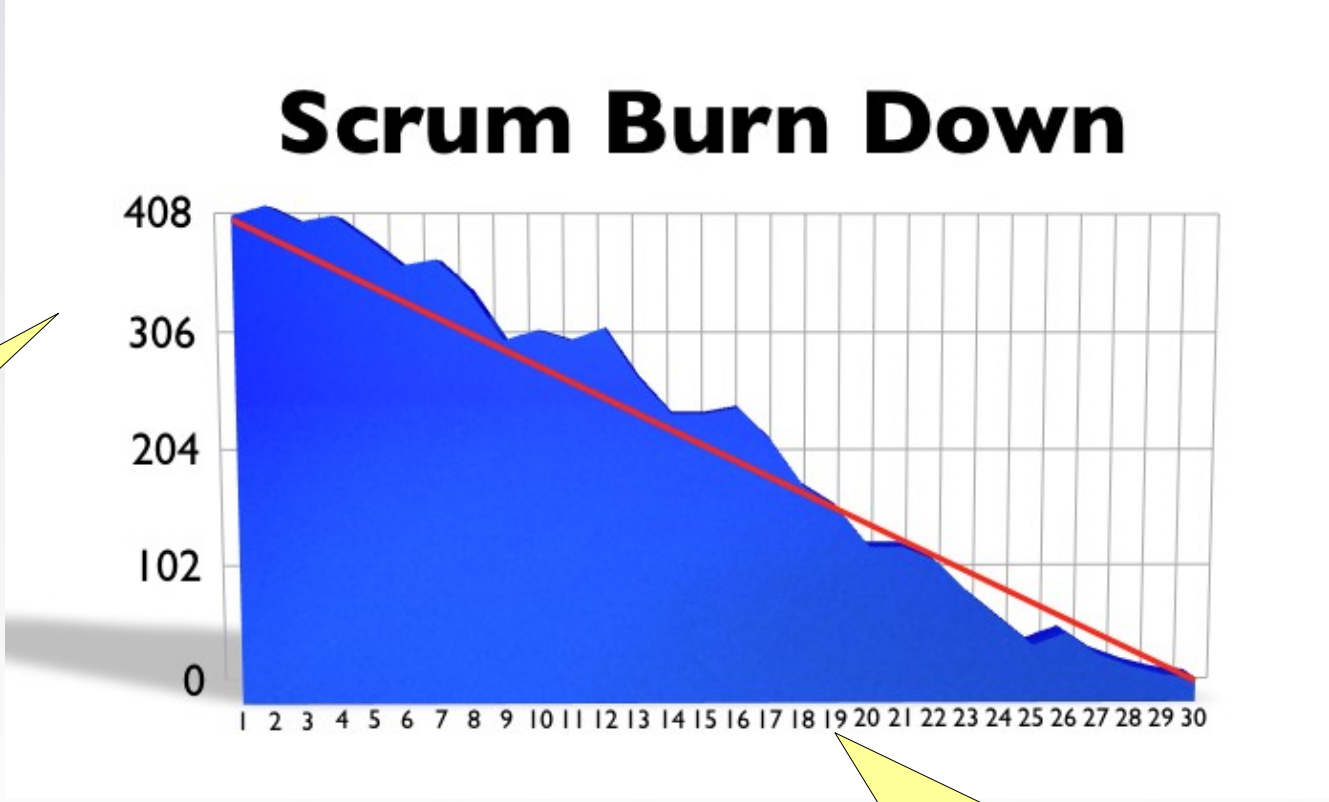
Reunión Diaria:

Es una figura fundamental en SCRUM.
Tiene que reunirse ***TODO*** el equipo y debe hacerse según ciertas reglas



Modelos Ágiles
(Gestión y Seguimiento / Scrum Burn Down)

UNIVERSIDAD DE LOS ANDES VENEZUELA

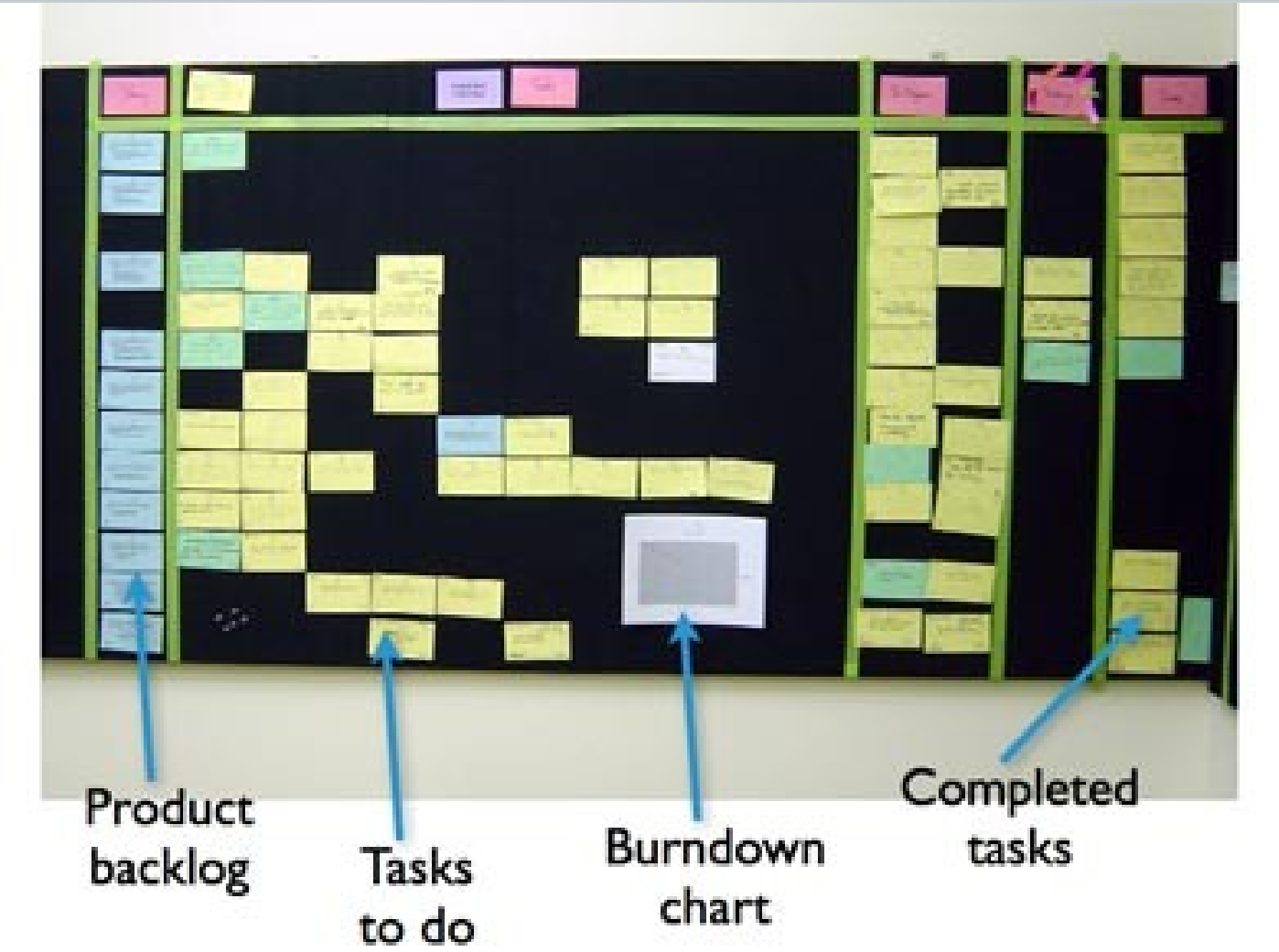


EJE Y
Trabajo restante, horas, puntos de función u otra unidad de medida

EJE X
Día o fecha del sprint

Esto es responsabilidad del Scrum Master

Modelos Ágiles
(Gestión y Seguimiento / Task Boards)



http://www.mountaingoatsoftware.com/scrum/task-boards

bien... pero...

¿qué método o proceso de desarrollo de software debo utilizar?

¿Qué Método o Proceso de desarrollo de software debo utilizar?

UNIVERSIDAD DE LOS ANDES VENEZUELA

Recuerde que no todos los proyectos y tipos de aplicaciones a desarrollar son iguales. Use el método que más se adapte a sus necesidades o al proyecto a enfrentar: No existen métodos perfectos o soluciones universales...

Evite caer en el “fanatismo” de métodos: XP fans vs RUP fans vs Scrum fans etc... (de hecho, evite caer en cualquier tipo de fanatismo)

Si ya ha usado un método anteriormente en proyectos similares a los que debe desarrollar y ha tenido resultados adecuados entonces vuelva a utilizar ese método...

¿Qué Método o Proceso de desarrollo de software debo utilizar?

UNIVERSIDAD DE LOS ANDES VENEZUELA

En proyectos pequeños / medianos los métodos ágiles pueden ser adecuados

En grandes aplicaciones empresariales el proceso unificado puede generar los resultados adecuados

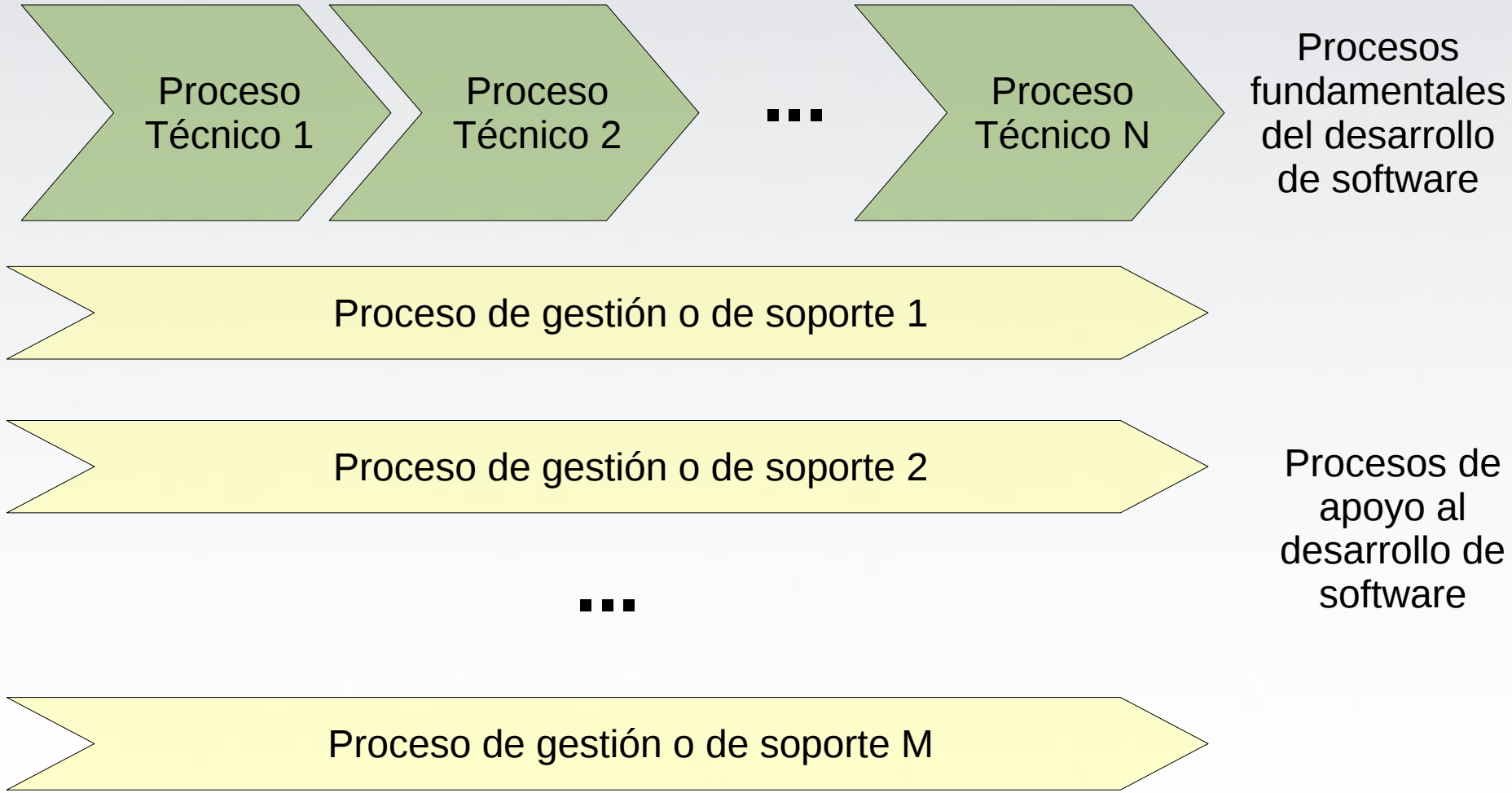
En proyectos con mucho personal los métodos ágiles pueden no ser el enfoque adecuado

Recuerde que en el fondo el método ***NO*** es lo importante, el método no es el fin. El método es sólo una herramienta para lograr el ***verdadero objetivo:*** terminar el proyecto a tiempo, dentro del presupuesto y con las características requeridas. ***¡Mantenga siempre la mira y la concentración en el producto, que es el verdadero objetivo!***

¿cómo se describe un método o proceso?

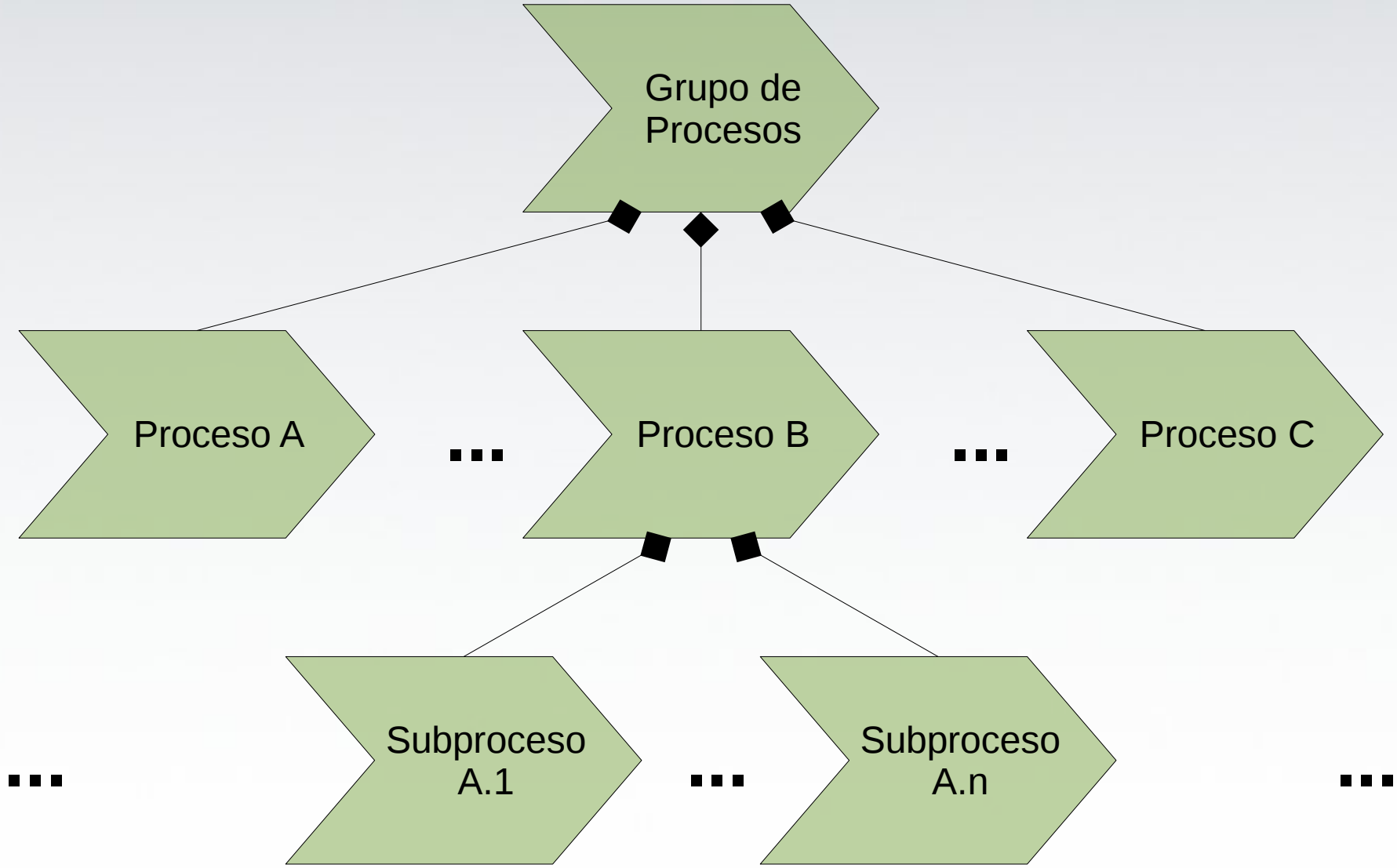
hay muchas formas, pero...

Métodos / Metodologías (Descripción de Procesos)



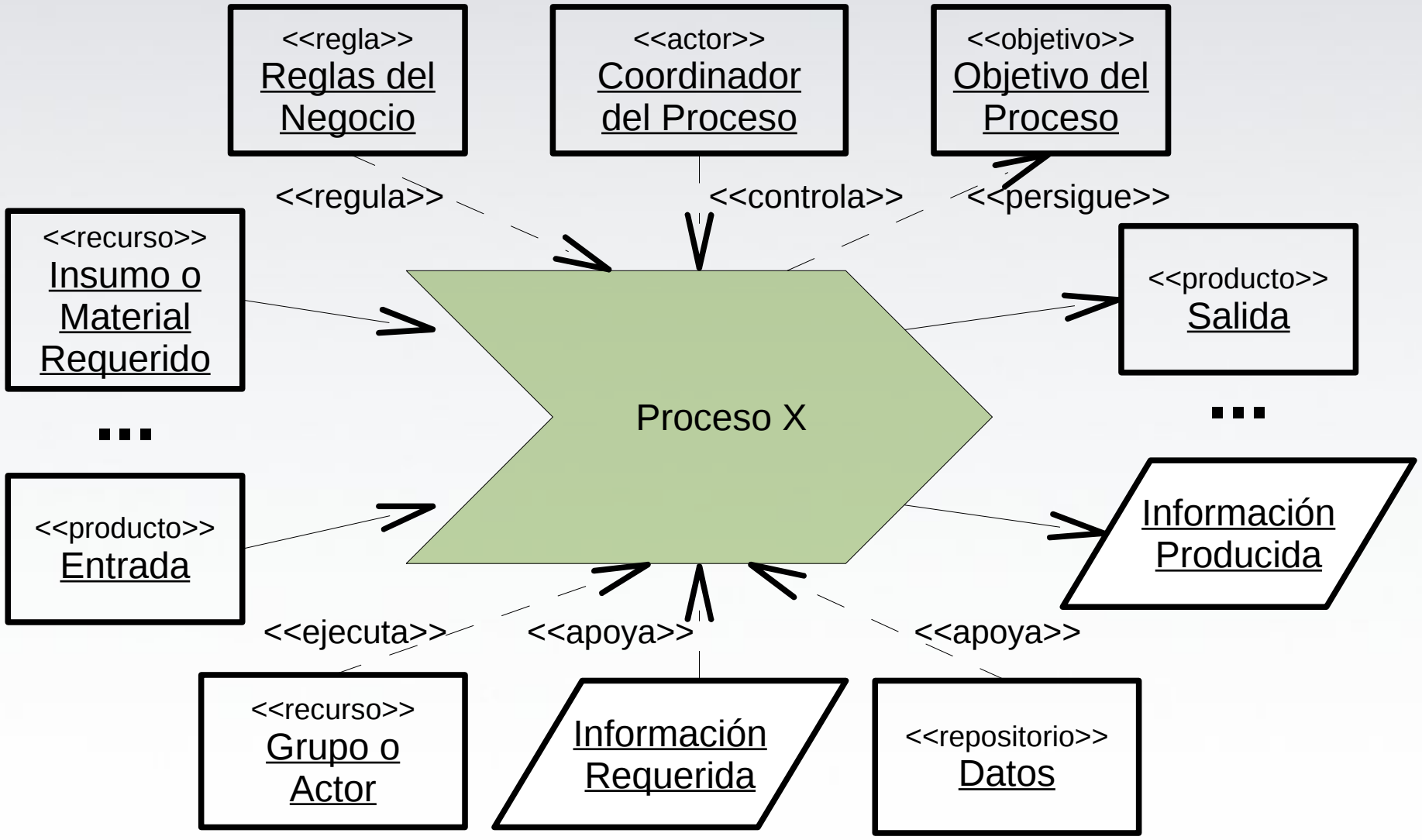
Fuente: GRAY WATCH MÉTODO DE DESARROLLO DE SOFTWARE PARA APLICACIONES EMPRESARIALES / Noviembre 2008

Métodos / Metodologías (Descripción de Procesos)



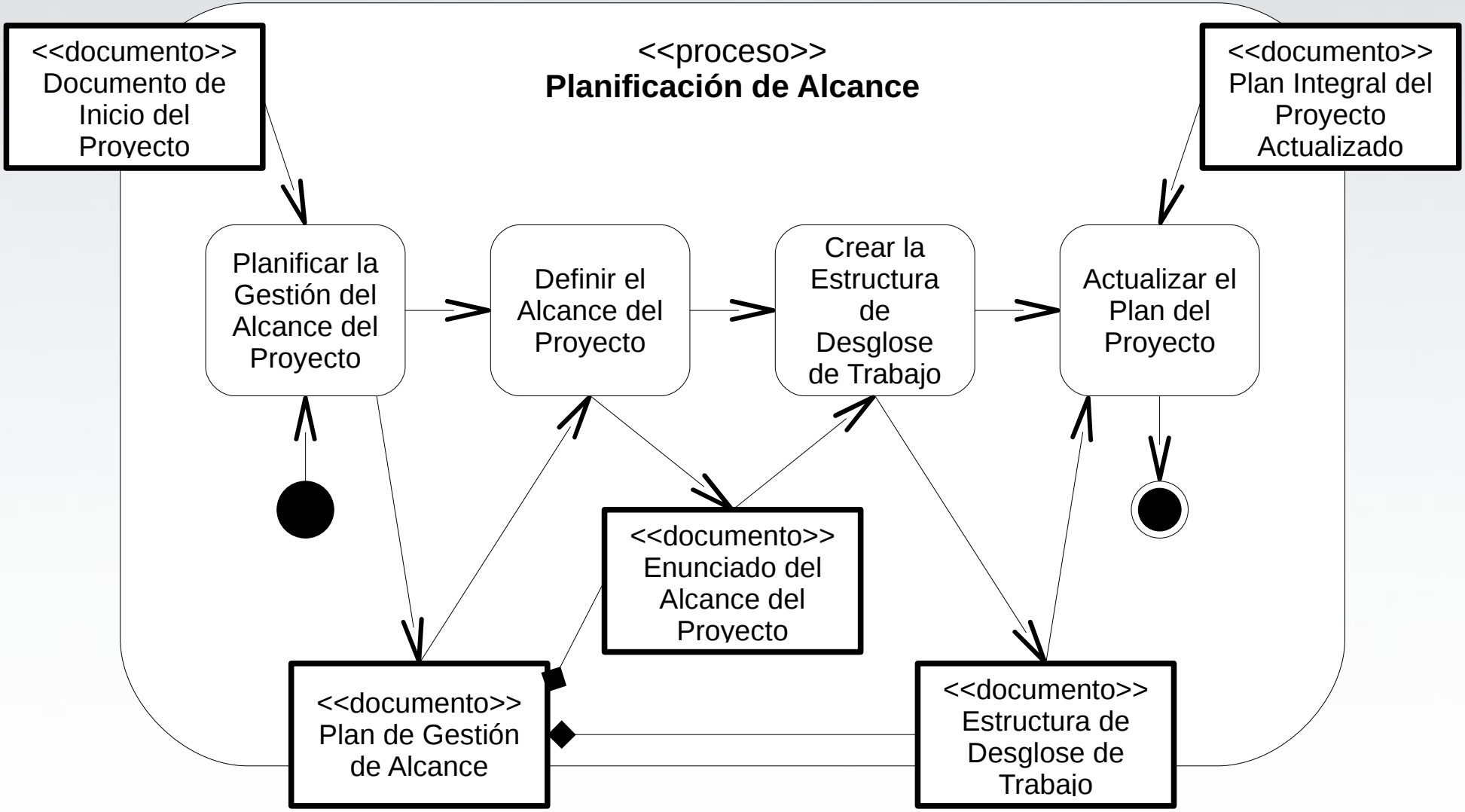
Fuente: Adaptado de GRAY WATCH MÉTODO DE DESARROLLO DE SOFTWARE PARA APLICACIONES EMPRESARIALES / Noviembre 2008

Métodos / Metodologías (Descripción de Procesos)



Fuente: Adaptado de GRAY WATCH MÉTODO DE DESARROLLO DE SOFTWARE PARA APLICACIONES EMPRESARIALES / Noviembre 2008

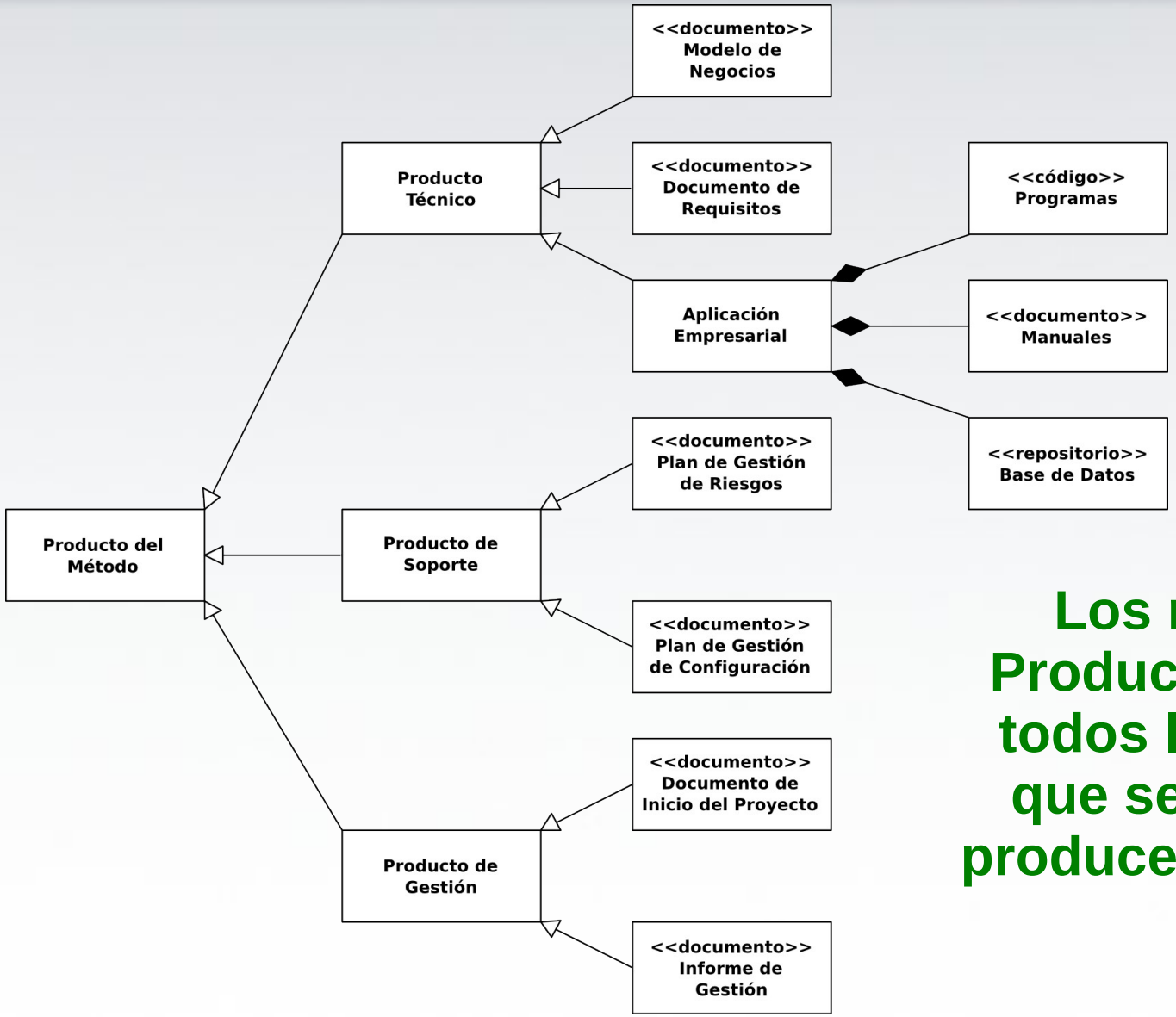
Métodos / Metodologías (Descripción de Procesos)



Fuente: Adaptado de GRAY WATCH MÉTODO DE DESARROLLO DE SOFTWARE PARA APLICACIONES EMPRESARIALES / Noviembre 2008

Métodos / Metodologías (Descripción de Procesos)

UNIVERSIDAD DE LOS ANDES VENEZUELA

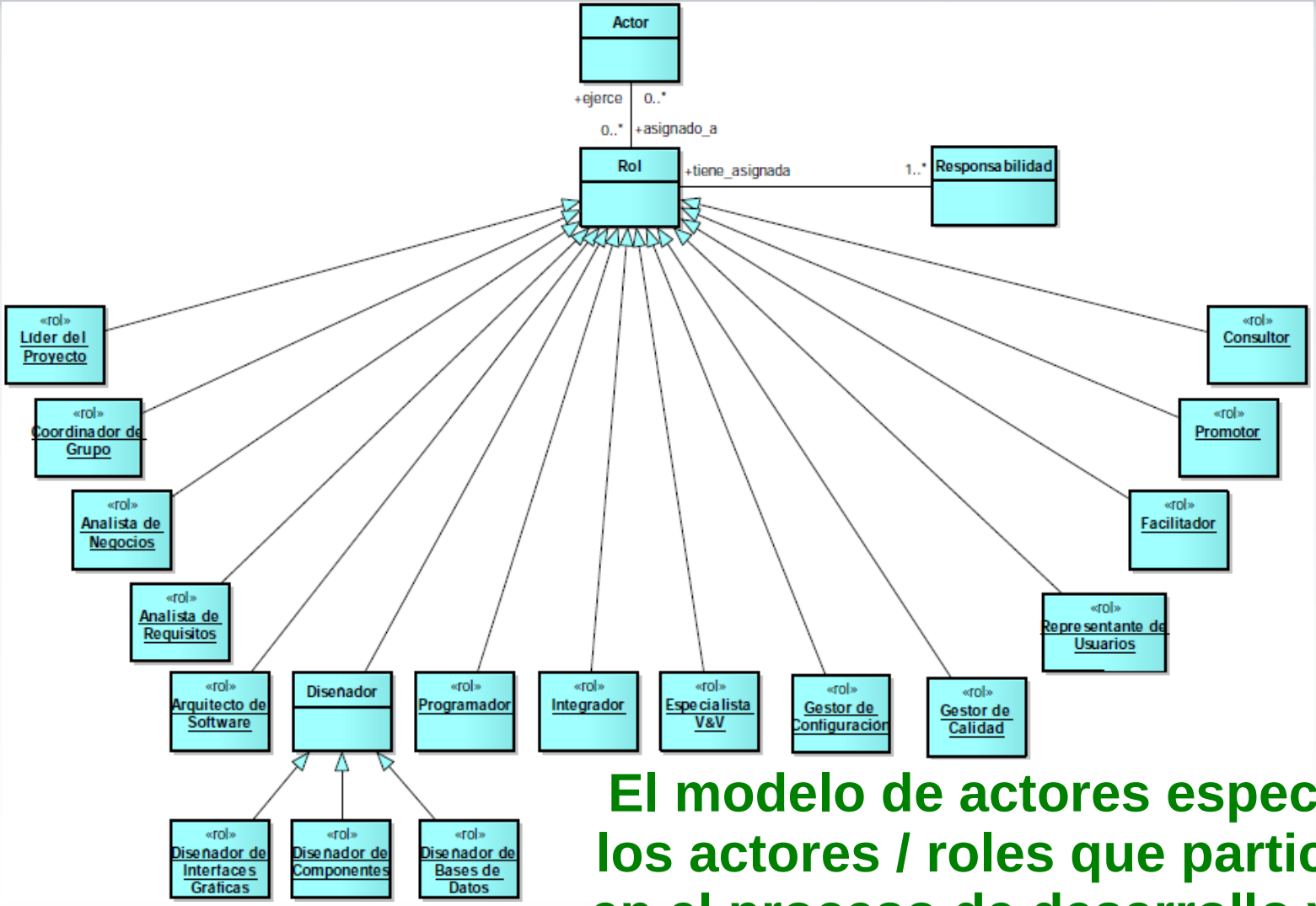


Los modelos de Productos describen todos los productos que se utilizan o se producen en el método

Fuente: Adaptado de GRAY WATCH MÉTODO DE DESARROLLO DE SOFTWARE PARA APLICACIONES EMPRESARIALES / Noviembre 2008

Métodos / Metodologías (Descripción de Procesos)

UNIVERSIDAD DE LOS ANDES VENEZUELA



El modelo de actores especifica los actores / roles que participan en el proceso de desarrollo y sus respectivas responsabilidades

Fuente: Adaptado de GRAY WATCH MÉTODO DE DESARROLLO DE SOFTWARE PARA APLICACIONES EMPRESARIALES / Noviembre 2008

¿algunos métodos de desarrollo?

UNIVERSIDAD DE LOS ANDES VENEZUELA

Extreme Programming (XP)

http://www.extremeprogramming.org/

Scrum

http://www.scrumalliance.org/

UNIVERSIDAD DE LOS ANDES VENEZUELA

RUP (IBM Rational Unified Process)

http://www-01.ibm.com/software/awdtools/rup/

OpenUP (Open Unified Process)

(Muy Buena Referencia)

http://epf.eclipse.org/wikis/openup/

AgileUP (Agile Unified Process)

http://www.ambysoft.com/unifiedprocess/agileUP.html

otros...

UNIVERSIDAD DE LOS ANDES VENEZUELA

Gray Watch

-

White Watch

(desarrollados aquí en la ULA)

UNIVERSIDAD DE LOS ANDES VENEZUELA

reflexión final

Pase lo que pase, siempre procure que el método de desarrollo que use trabaje a su favor. Si el método que está usando trabaja en su contra ¡entonces cambie el método!

Recuerde, el método no es importante, lo importante es el cliente, el producto y las personas involucradas en su desarrollo

UNIVERSIDAD DE LOS ANDES VENEZUELA

¡Gracias!