

Proceso

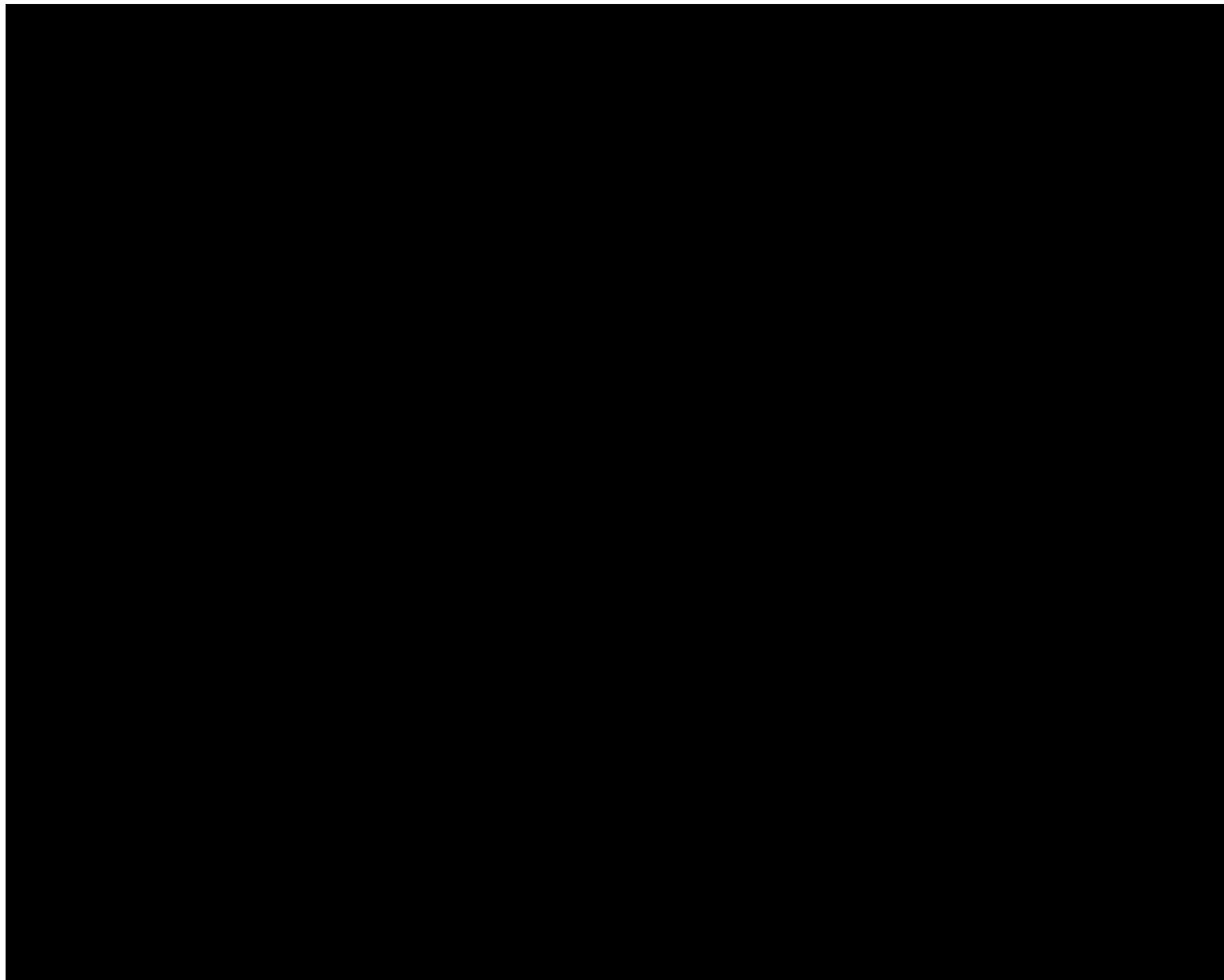
Proceso de Desarrollo

- ▶ La estrategia usada para llevar a cabo las actividades de desarrollo hasta llegar al producto
- ▶ Cada actividad tiene sus insumos y entregables, puede requerir herramientas
- ▶ Una actividad puede aparecer más de una vez en el proceso
- ▶ Existen diversos *modelos de proceso* que pueden ser usados

Proceso de Construcción de un Auto



Proceso: Neumáticos Michelin



Proceso de Construcción de un Edificio

Desarrollo de Software involucra muchas actividades distintas

- Estudio de factibilidad (puede ser muy caro, muy riesgoso o innecesario)
- Ingeniería de requisitos (especificación de requisitos)
- Diseño de Interfaz de Usuario
- Diseño de la Arquitectura
- Diseño detallado (modules)
- Programación (coding)
- Integración (build)
- Verificación (unit testing, system testing)
- Validación (acceptance testing)
- Poner en producción
- Mantención
- Documentación
- Gestión del Proyecto

¿ Se puede desarrollar sin un modelo de proceso ?

- ▶ Sí, se llama *hacking*
- ▶ programador usa habilidades extrañas y no sigue ningún método visible para lograr la meta .
- ▶ dependencia muy alta de habilidades
- ▶ riesgo muy alto hasta el final



Finalmente ...

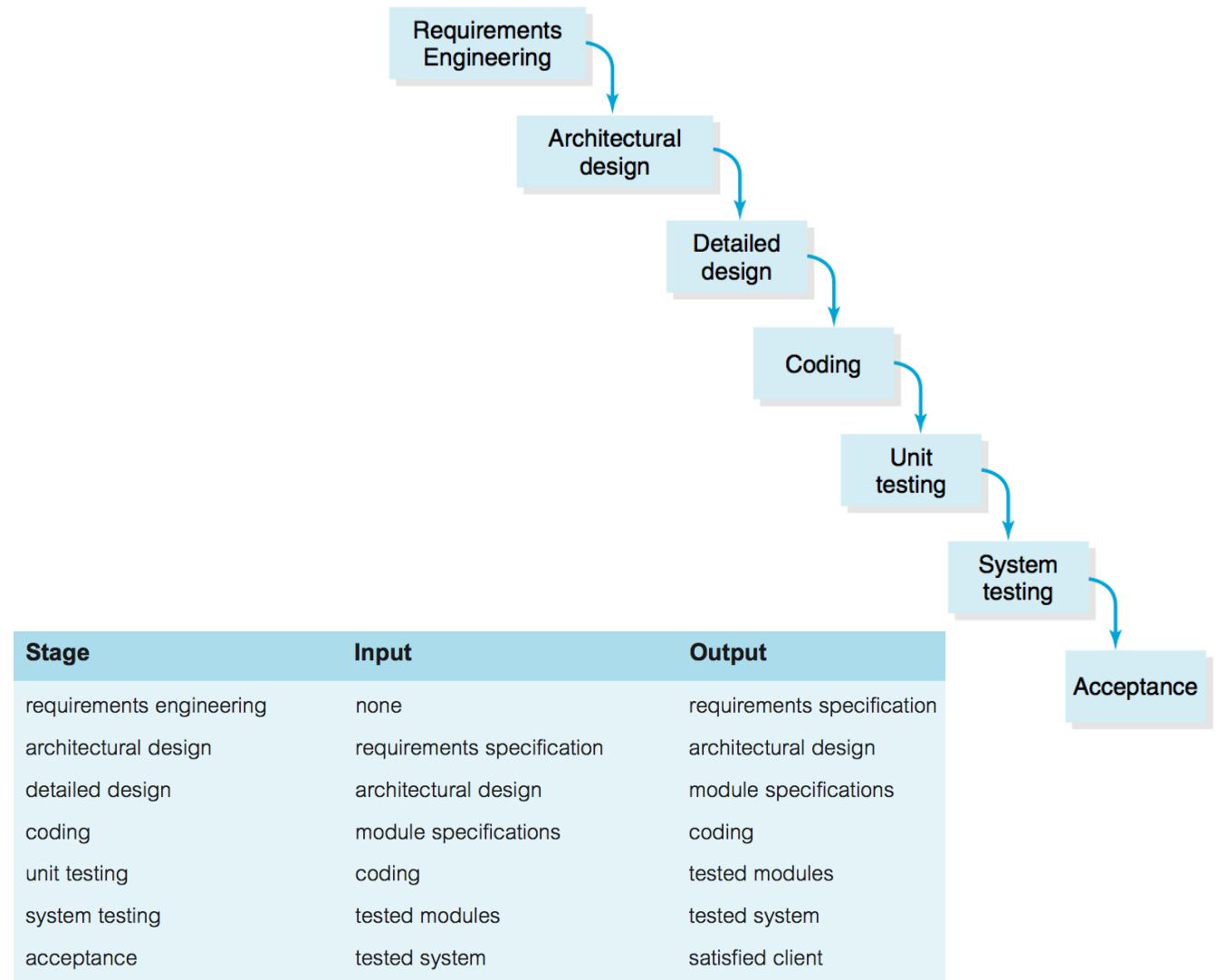


Pero mas frecuentemente ...



El Modelo de Cascada (Waterfall)

► El Modelo de Cascada (Waterfall)



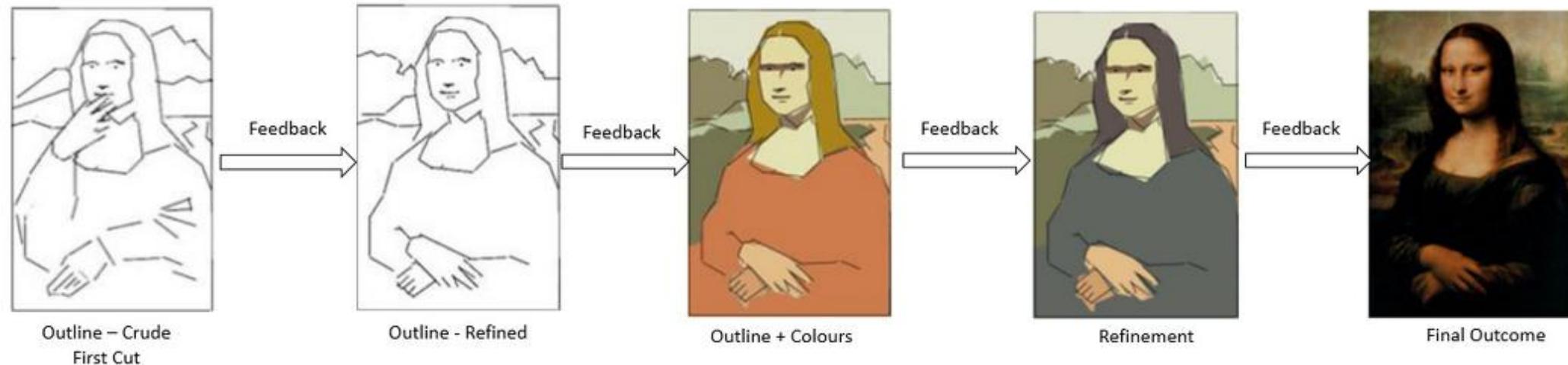
¿ Por qué es popular ?

- ▶ Conceptualmente simple y fácil de entender
- ▶ Similar a procesos de manufactura o construcción
- ▶ Cada fase tiene un input y un output definido
- ▶ Se facilita asignar recursos
- ▶ Se puede saber cuanto llevamos y cuanto falta

Tiene un problemita ...

- ▶ No funciona en la práctica !! Salvo proyectos muy cortos
- ▶ no se conocen todos los requisitos al comienzo
- ▶ se conocen pero cambian durante el proyecto
- ▶ Riesgo alto hasta etapas muy avanzadas del proyecto

Procesos Iterativos



- ▶ Ejemplos de procesos iterativos: modelo espiral, modelo de prototipos. modelo unificado (RUP)

Modelo Espiral

Construct

Evaluate

Analyze
requirements

Analyze risks
and plan

- ▶ Enfoque Iterativo
- ▶ Evaluación periódica del riesgo
- ▶ No hay número definido de iteraciones
- ▶ Construcción puede ser diseño, programación, verificación o validación

Riesgos

- cliente cambia requerimientos
- no se consideraron algunos requerimientos
- salida de gente del equipo de desarrollo
- una o más de las tareas fuera de plazos
- plataforma de software es muy lenta
- software requiere demasiada memoria
- aparecen nuevas tecnologías de desarrollo de software
- se malentendió un requerimiento
- cambio en el hardware planificado
- un bug que no puede ser reparado
- competidor lanza un producto alternativo



Prototipos



- ▶ Foco en asegurar que el usuario obtenga lo que realmente necesita (validación)
- ▶ Se le entrega tempranamente una versión que funciona (puede tener muchas limitaciones)
- ▶ Técnicas de desarrollo rápido
- ▶ El usuario solo se da cuenta de lo que quiere al ver el producto
- ▶ Puede ser evolucionario o desechable

Tipo de Prototipo

- ▶ Desechable
 - ▶ prototipos se desarrollan solo para definir los requisitos y validar el producto
 - ▶ una vez terminado se procede a construir el producto real
- ▶ Evolucionario
 - ▶ prototipo evoluciona hasta convertirse en el producto final

Prototipado Rápido

- lenguaje de alto nivel (no se hace en C !)
- reuso de componentes
- stand alone (una sola máquina)
- sin manejo de errores de entrada o excepciones
- omitir features no esenciales (seguridad, autenticación, etc)
- ignorar parte de la funcionalidad

Ventajas

- ▶ permite enfrentar falta de claridad en requerimientos
- ▶ menor tiempo de desarrollo porque no hay grandes sorpresas al final
- ▶ involucramiento del usuario es más fácil de lograr
- ▶ menor frustración y ansiedad durante el desarrollo
- ▶ permite introducir nuevo sistema gradualmente
- ▶ facilita entrenamiento de usuarios
- ▶ aumento de satisfacción del usuario

Desventajas

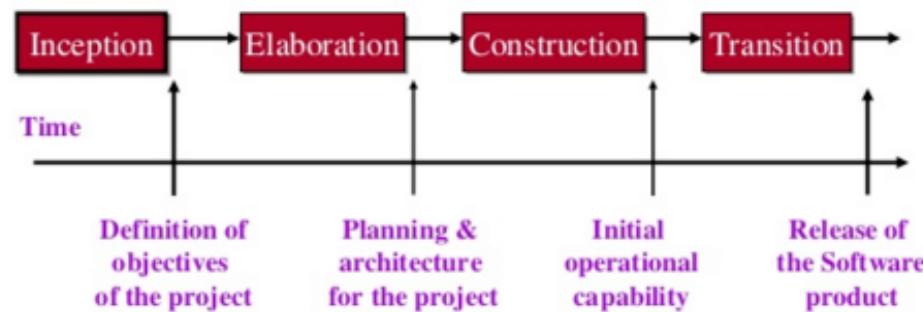
- ▶ puede generar expectativas de que el software está “casi listo”
- ▶ producto final no queda igual que prototipo que se desecha
- ▶ usuario puede no estar nunca satisfecho (proyecto no tiene fin)
- ▶ análisis incompleto de requisitos (superficialmente entendido)
- ▶ omisión de requerimientos no funcionales
- ▶ difícil de planificar (cuantos prototipos, cuanto tiempo)
- ▶ no hay entregables claros
- ▶ poco apropiado para sistemas embebidos, sistemas de tiempo real, software científico, etc

El Proceso Unifica



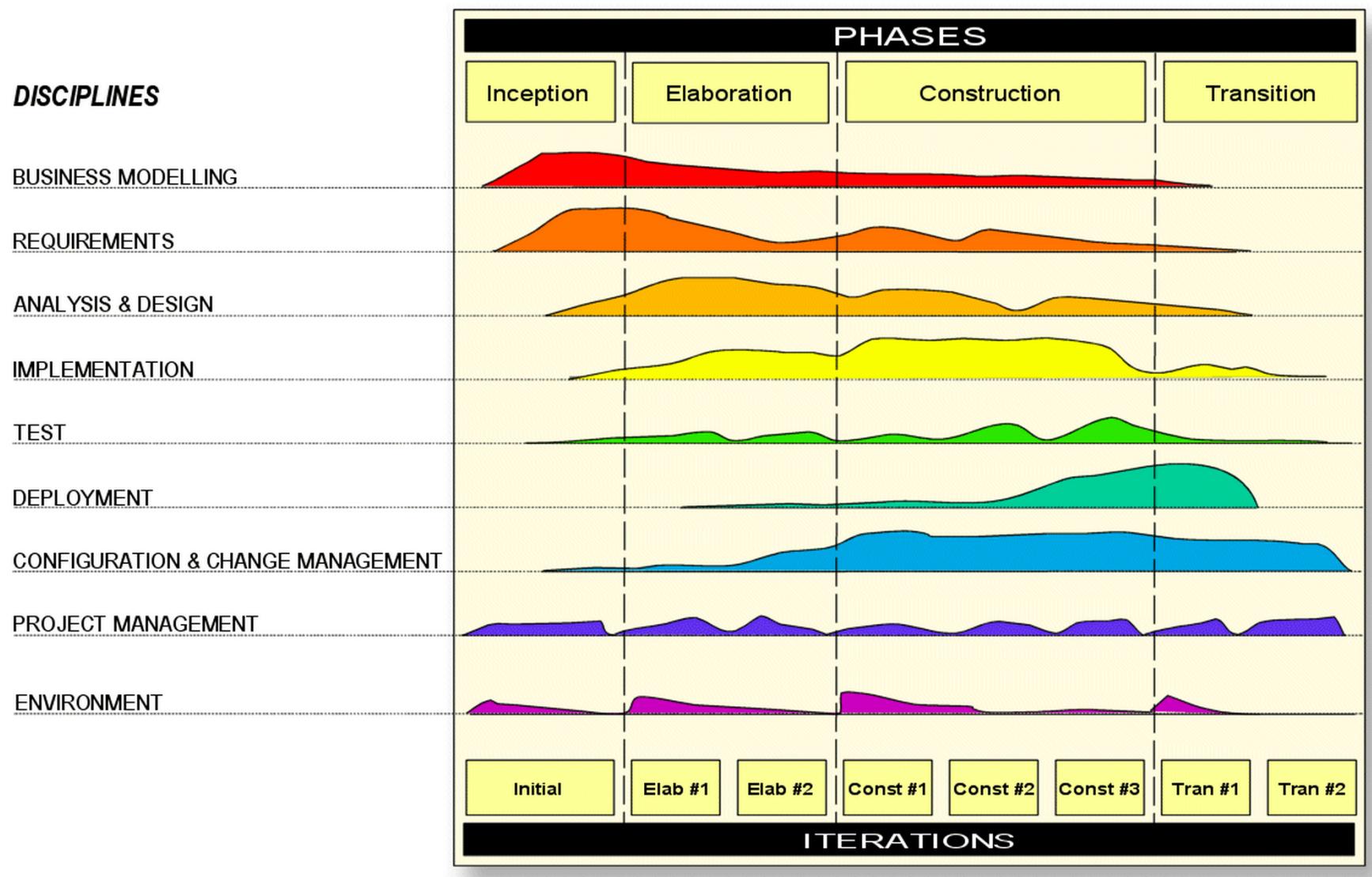
- ▶ Anteriormente conocido como RUP
- ▶ Desarrollo en 4 fases: Incepción, elaboración, construcción y transición
- ▶ En cada fase pueden haber varias iteraciones
- ▶ Se hacen distintas tareas en paralelo

Las 4 fases

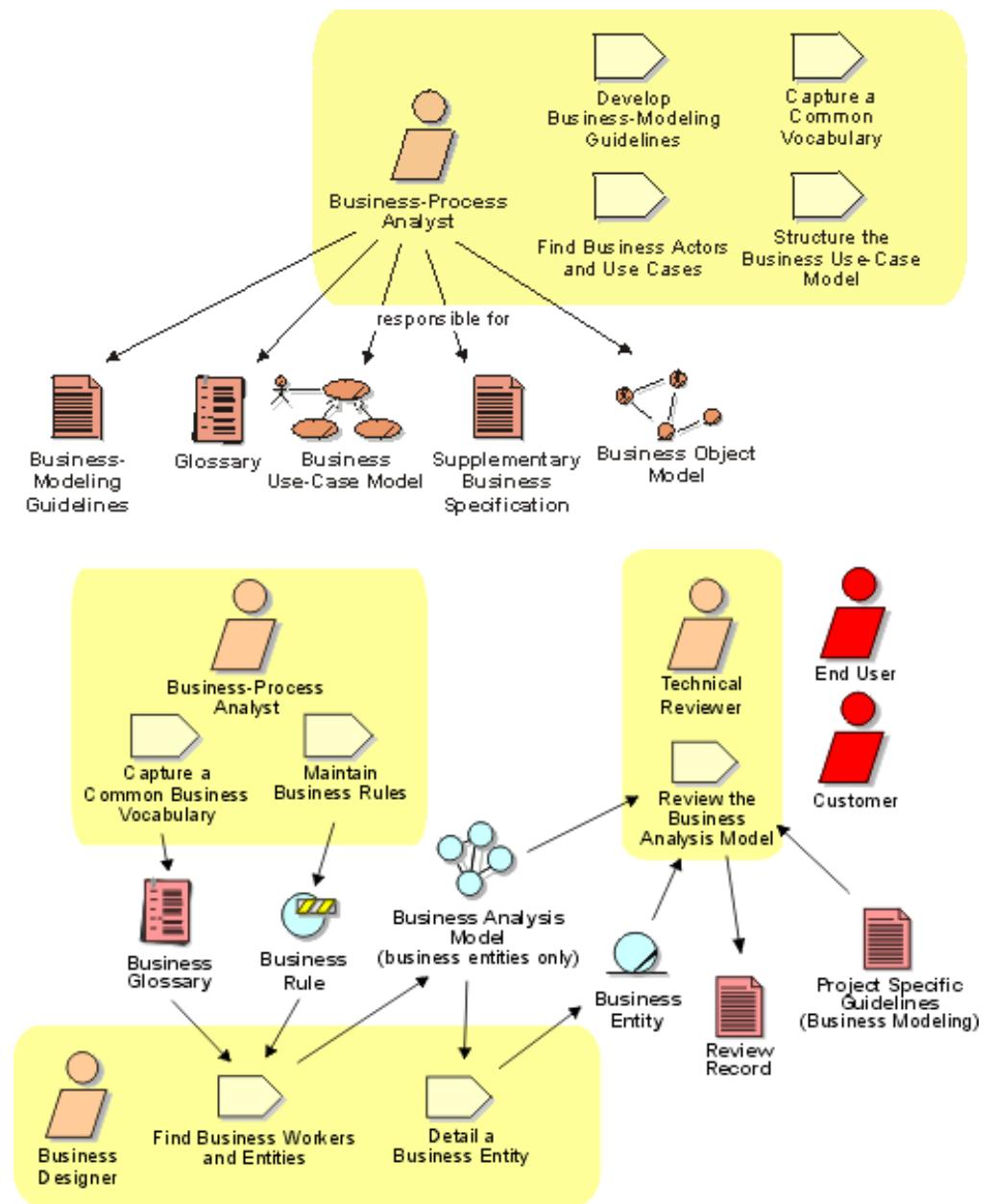
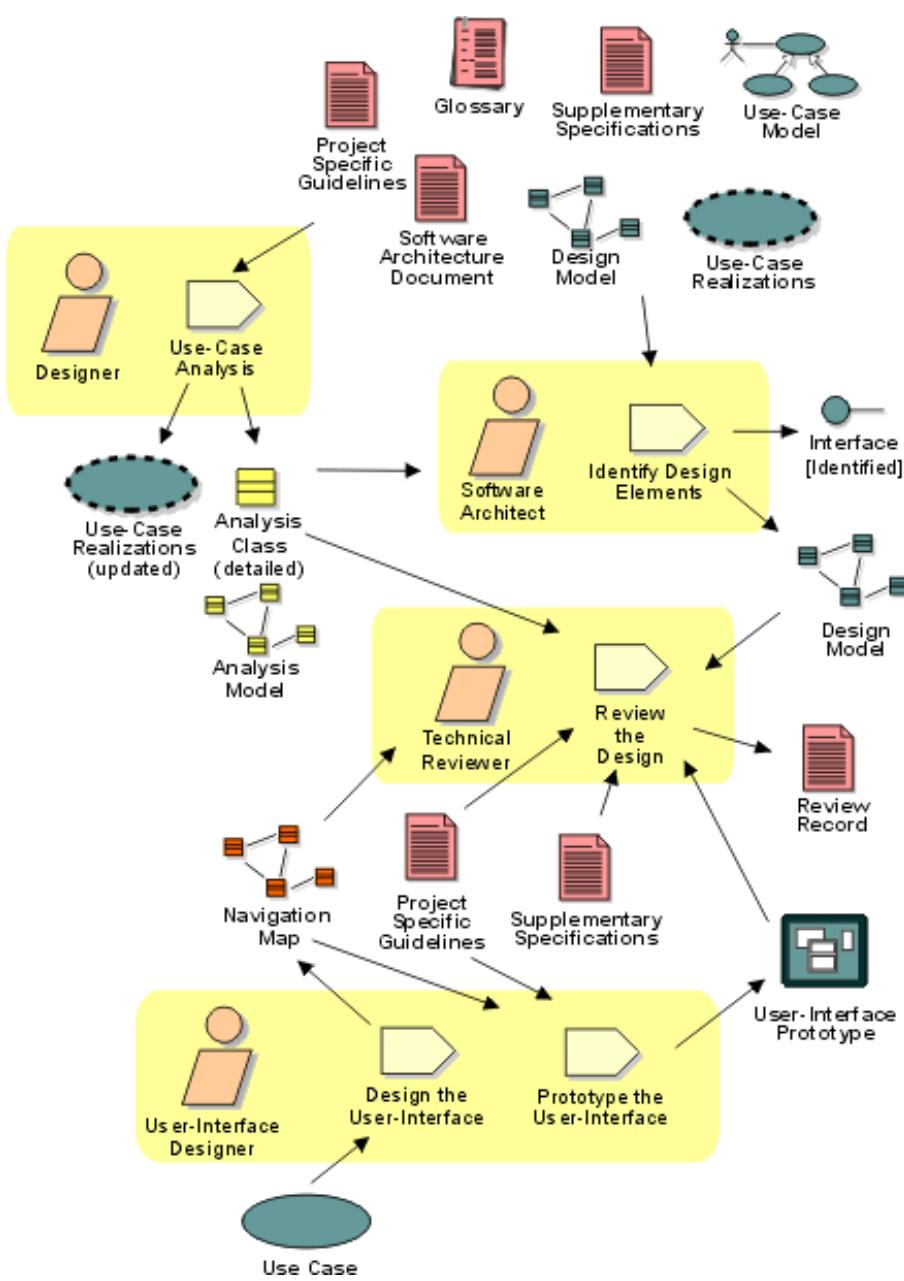


- ▶ Inception - Establecer el caso y la viabilidad del sistema propuesto
- ▶ Elaboration - Establecer la habilidad de construir el sistema dadas las restricciones existentes
- ▶ Construction - Construir un sistema que opere exitosamente (beta)
- ▶ Transition - Entregar un sistema totalmente funcional a los usuarios

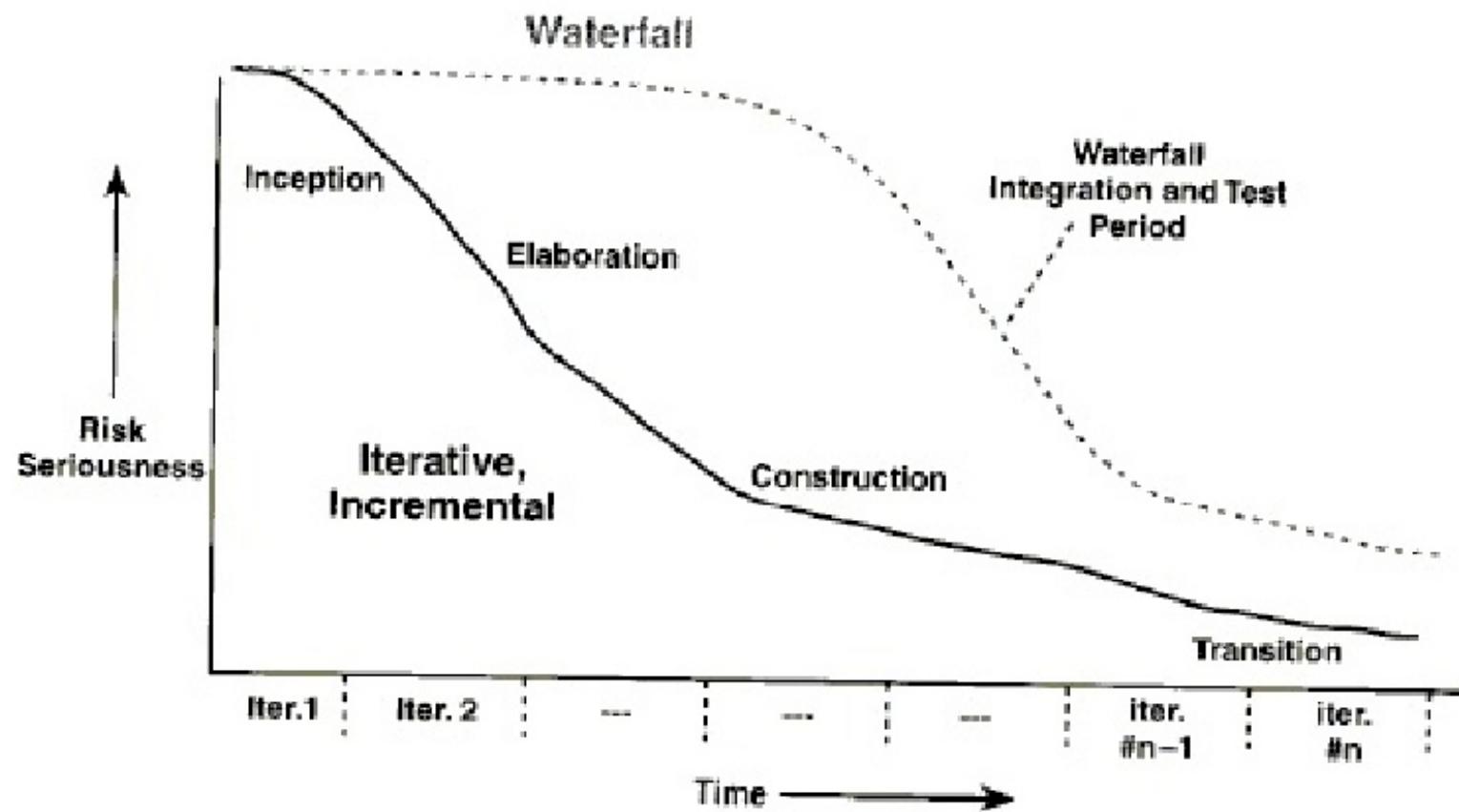
Fases, Iteraciones y Tareas



RUP Workflows



Reducción temprana del riesgo



Noticia Reciente ...

Auditor General for Scotland

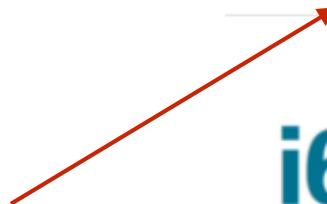
The Auditor General's role is to:

- appoint auditors to Scotland's central government and NHS bodies
- examine how public bodies spend public money
- help them to manage their finances to the highest standards
- check whether they achieve value for money.

A classic “waterfall” IT project disaster – yet officials went by the book

Posted on [March 9, 2017](#) | [2 Comments](#)

i6: a review



Lecciones (del reporte)

The “waterfall” systems development contributed to the fact that Police Scotland “only discovered the true extent of problems with the system when it was delivered for testing”. Waterfall meant that Accenture produced the software in distinct phases, in a sequence resembling a waterfall. Once a phase was complete, the process moved to the next phase – and no turning back. “It meant that all of the design, coding and construction of i6 would be completed before Accenture released it to Police Scotland for testing. Police Scotland would pay for each phase when it was completed.” [Agile, on the other hand, is a “test and see” approach and is far more flexible. It can adapted according to what the end-user needs and wants, and changes in those needs and wants.]

to complete i6 was around eight times greater than the resources Accenture had estimated when signing the original contract. “The i6 programme team believed that the functionality of Accenture’s solution did not meet the requirements it had agreed in the contract. Accenture maintained that Police Scotland had not specified a detailed description of business requirements. This issue had not emerged during months of pre-award dialogue. Accenture also believed that it had set out clearly what its solution would do and maintained that Police Scotland, as part of procurement process, had accepted its qualified solution. A dispute followed about the interpretation of the contract requirements. Police Scotland argued that, after months of competitive dialogue, the requirements of the i6 system were well-defined, and that in line with the contract, these took precedence. Accenture argued its solution had precedence and that Police Scotland was trying to extend the scope of the programme. Accenture stated that, to meet Police Scotland’s interpretation of requirements, it would require more time and money.”

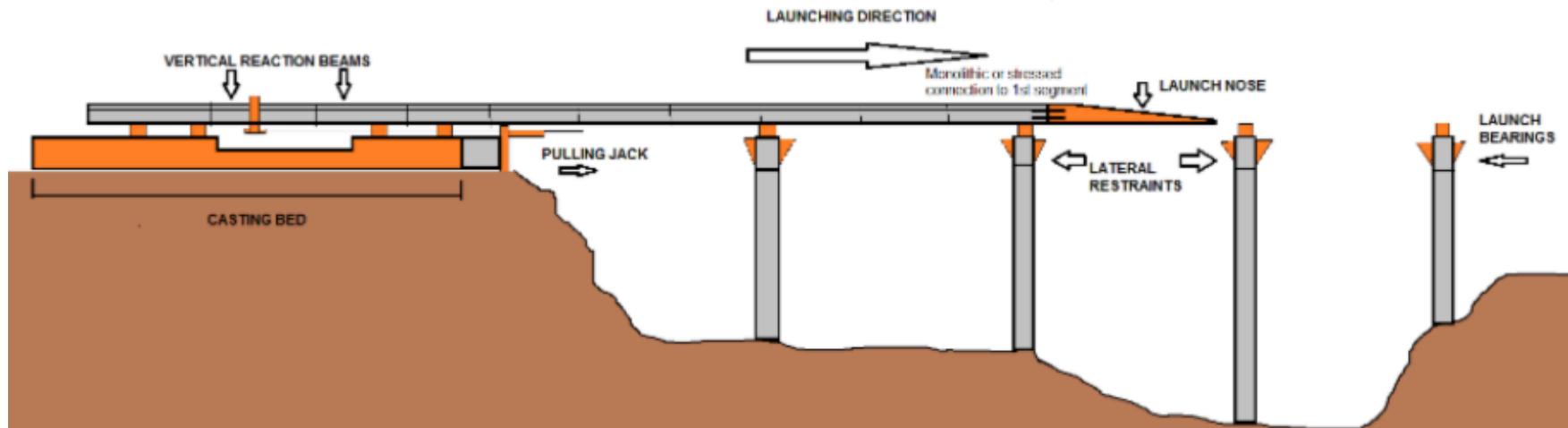
Epílogo

To its credit Accenture refunded all the money the police in Scotland had paid for the i6 system, £11.06m, plus a further £13.56m – but Audit Scotland says the failure of the project ...

11.6 + 13.56 = 25.16 millones de libras

Más de 20.000 millones de pesos !

Procesos Incrementales



- Software se produce en forma de pequeños incrementos
- Cada incremento incluye funcionalidades completas (probadas y validadas)
- Producto se completa con entrega de último incremento
- Incrementos comúnmente asociados a casos de uso (o historias de uso)

Métodos Ágiles

- ▶ iterativo + incremental
- ▶ El Manifiesto Ágil - los valores centrales
 - ▶ individuos e interacciones por sobre procesos y herramientas
 - ▶ software que corre por sobre documentación
 - ▶ colaboración con el cliente por sobre negociación de contratos
 - ▶ responder a los cambios en lugar de apegarse a un plan

Extreme Programming (XP)

- ▶ cliente siempre como parte de equipo de desarrollo
- ▶ casos de uso muy simples (3 frases)
- ▶ cliente especifica casos de prueba
- ▶ software lo más simple posible
- ▶ programación en pares
- ▶ propiedad compartida del código
- ▶ testing permanente

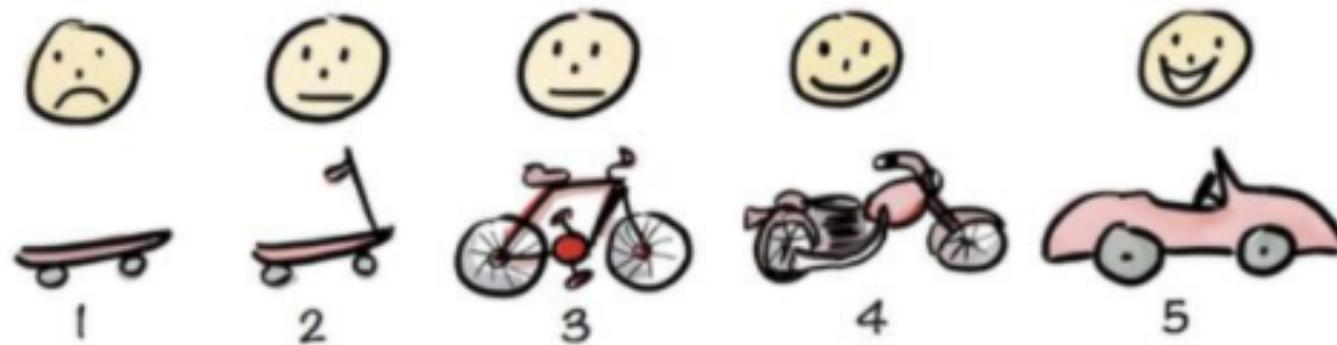
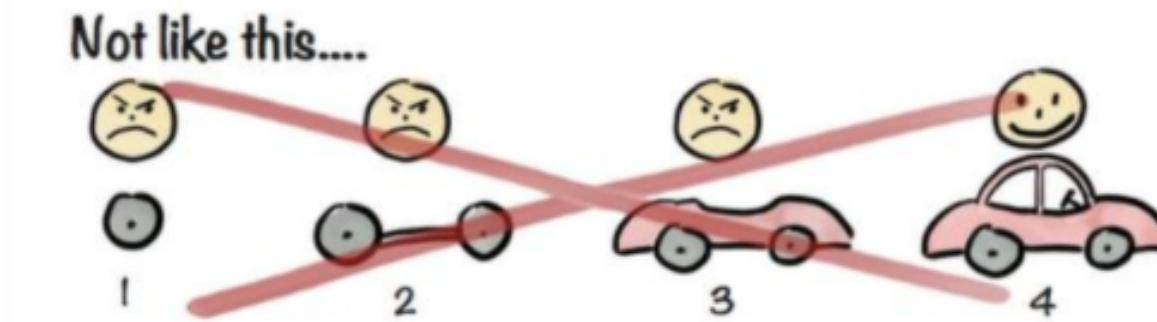


Programación en Pares



Pair programming

MVP: minimum viable product



Cuidado con MVP ...

HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

ALSO HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

HOW TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

Métodos Agiles mas Usados



Agile Usage Statistics

- VersionOne found 84% using agile methods today
- Most are using Scrum with several key XP practices
- Lean-Kanban is a rising practice with a 32% adoption

