Ciclos de vida

Analisis de Sistemas - FRBA

Agenda

- Definición
- Historia – Antecedentes
- Planificación - Decisión
- Ciclos de vida
  - Waterfall (Cascada)
  - Espiral
  - Iterativo e incremental
  - Agile
  - Code & Fix

Agenda

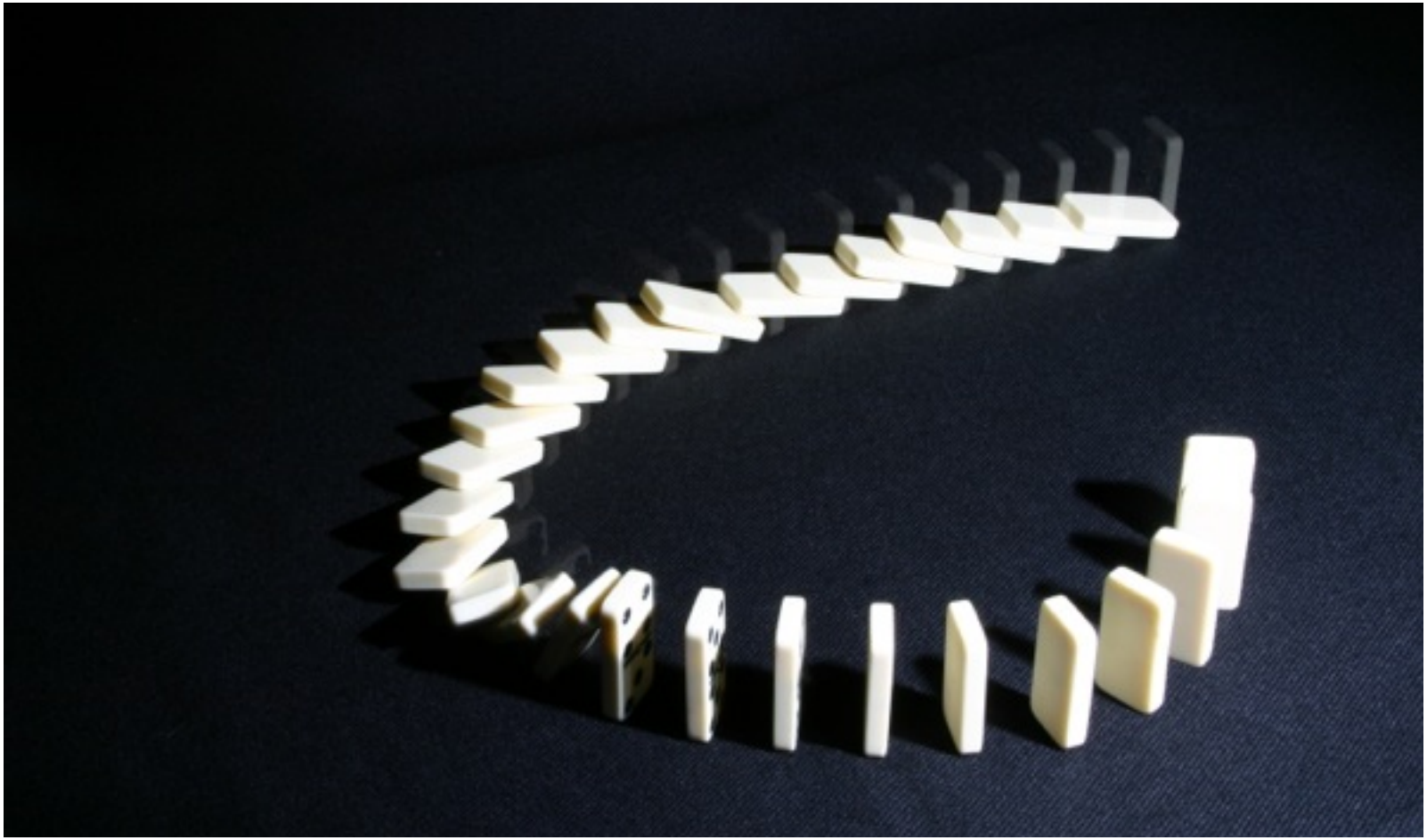
- Conclusiones
- Preguntas

Definición

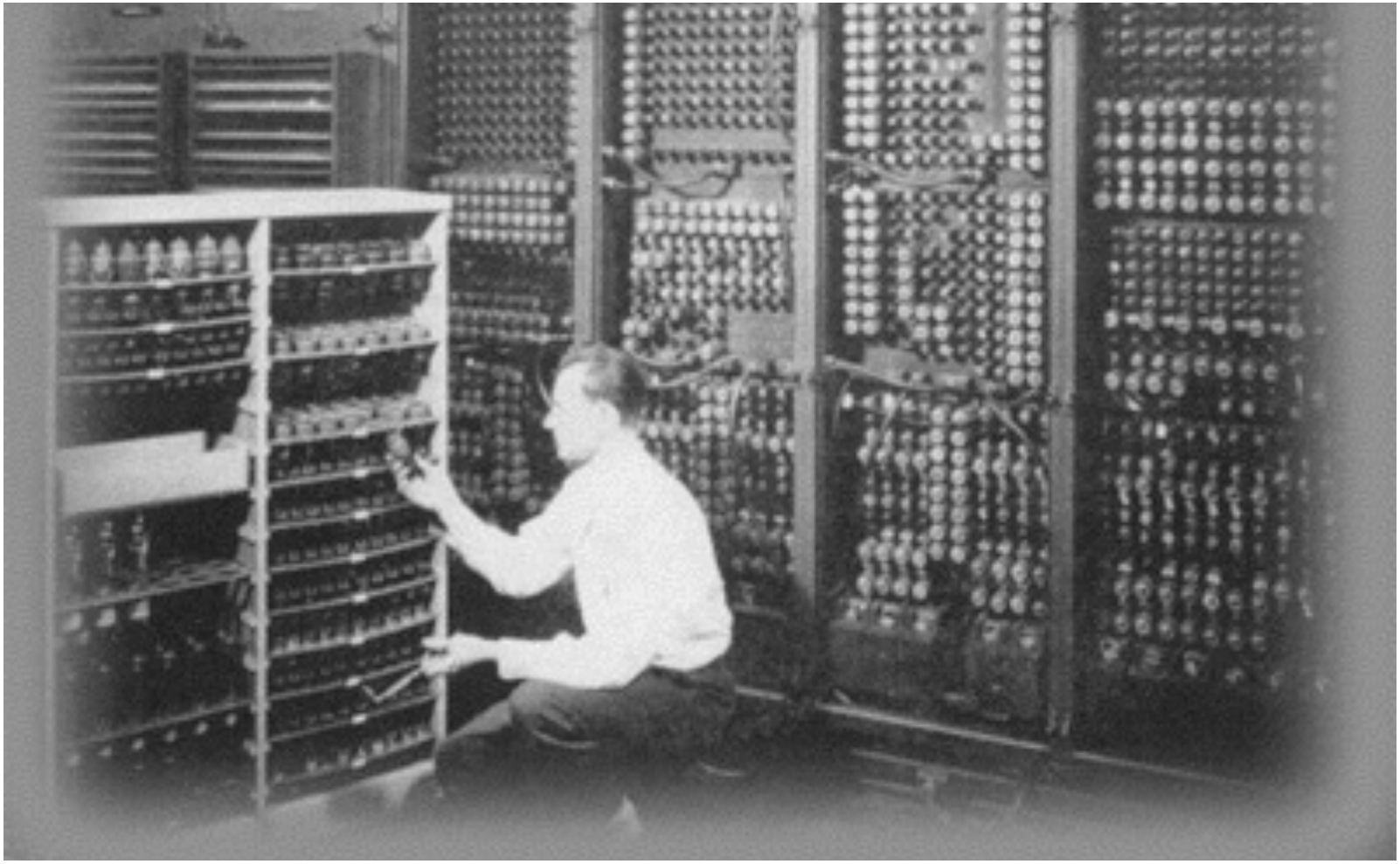
- ¿A qué llamamos Ciclo de vida?

Es el conjunto de etapas y sus transiciones por las cuales pasa un proyecto de software desde que se manifiesta la necesidad hasta que deja de existir

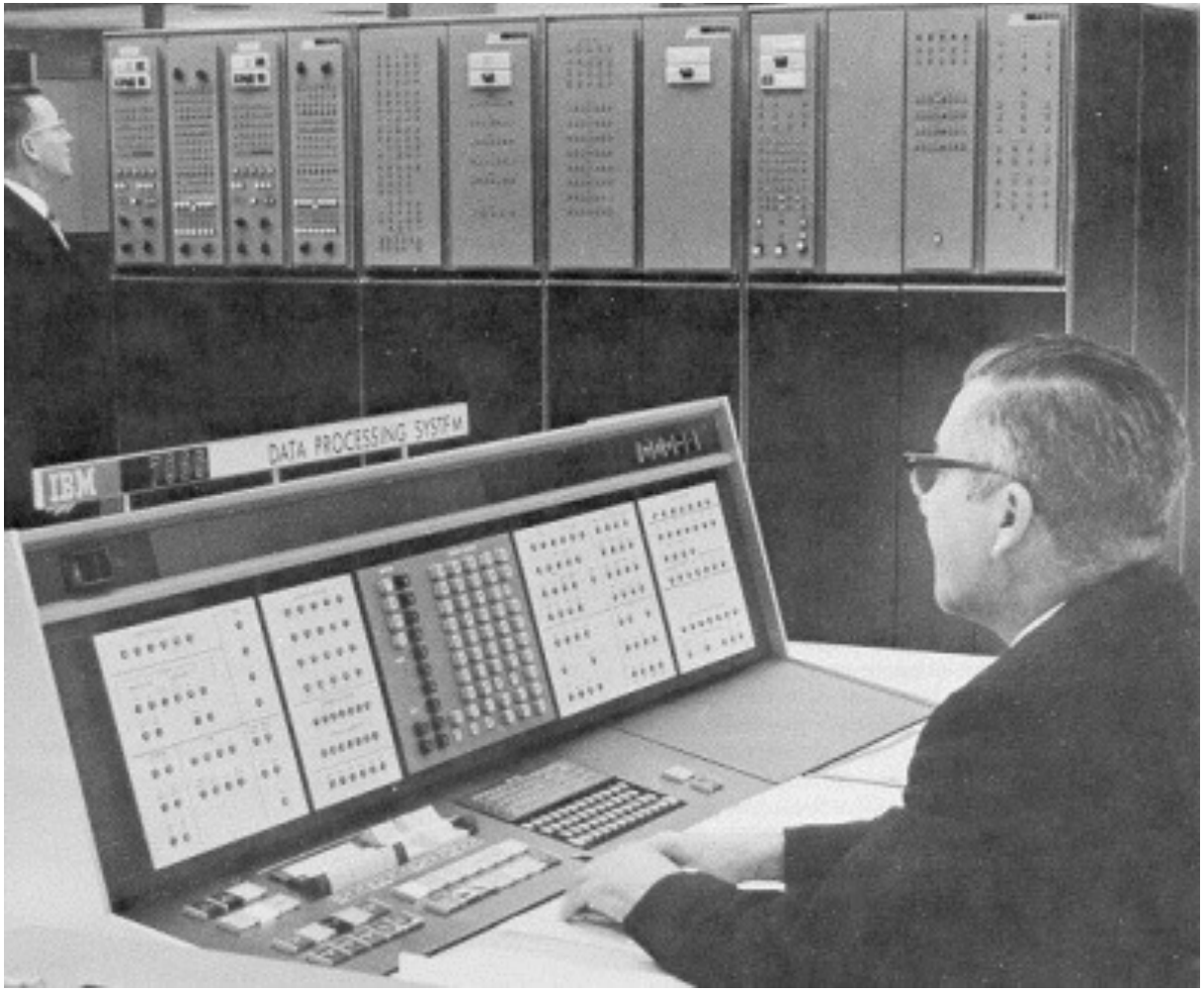
Definicion



Historia - Antecedentes



Historia - Antecedentes



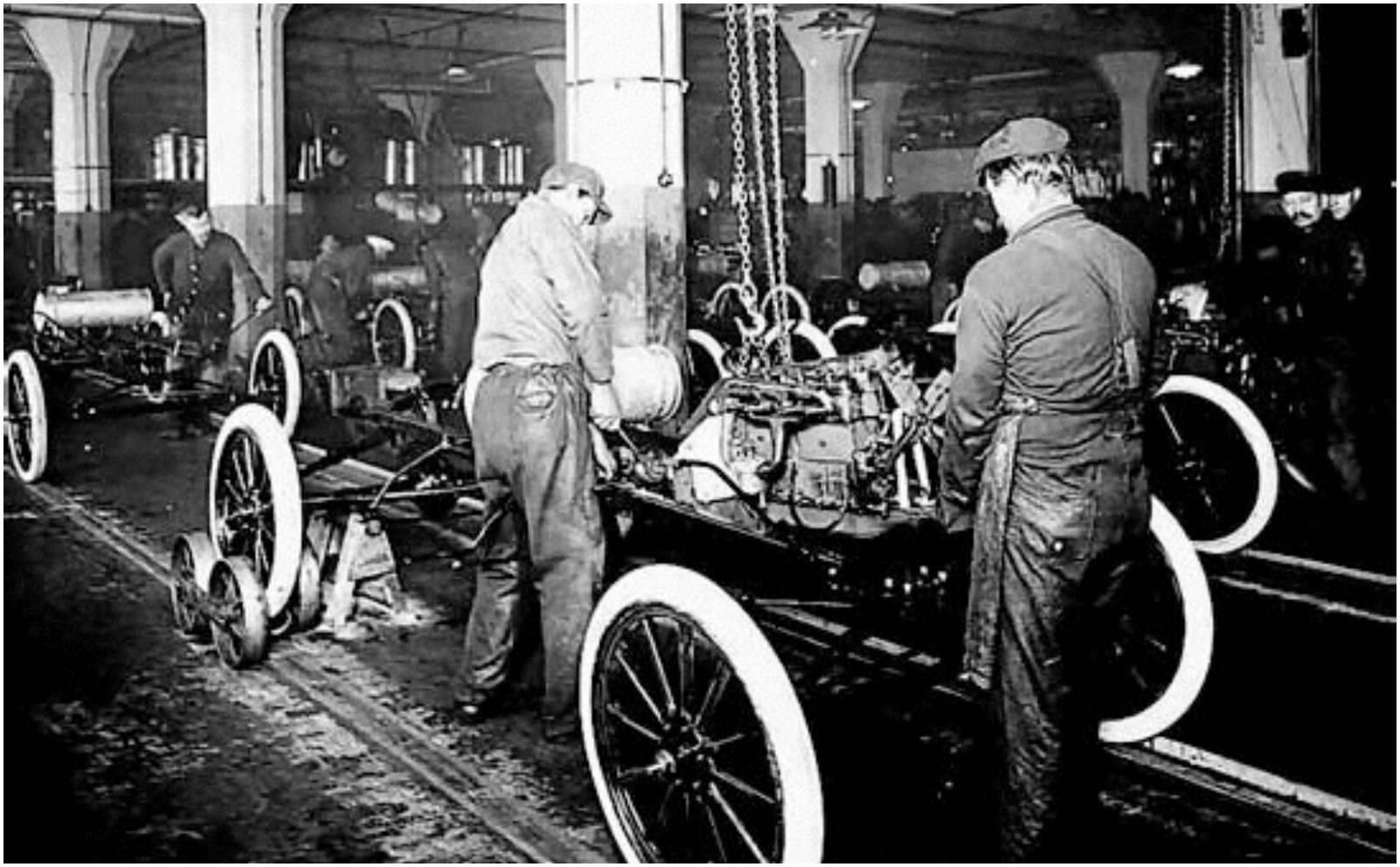
Historia - Antecedentes

Historia - Antecedentes

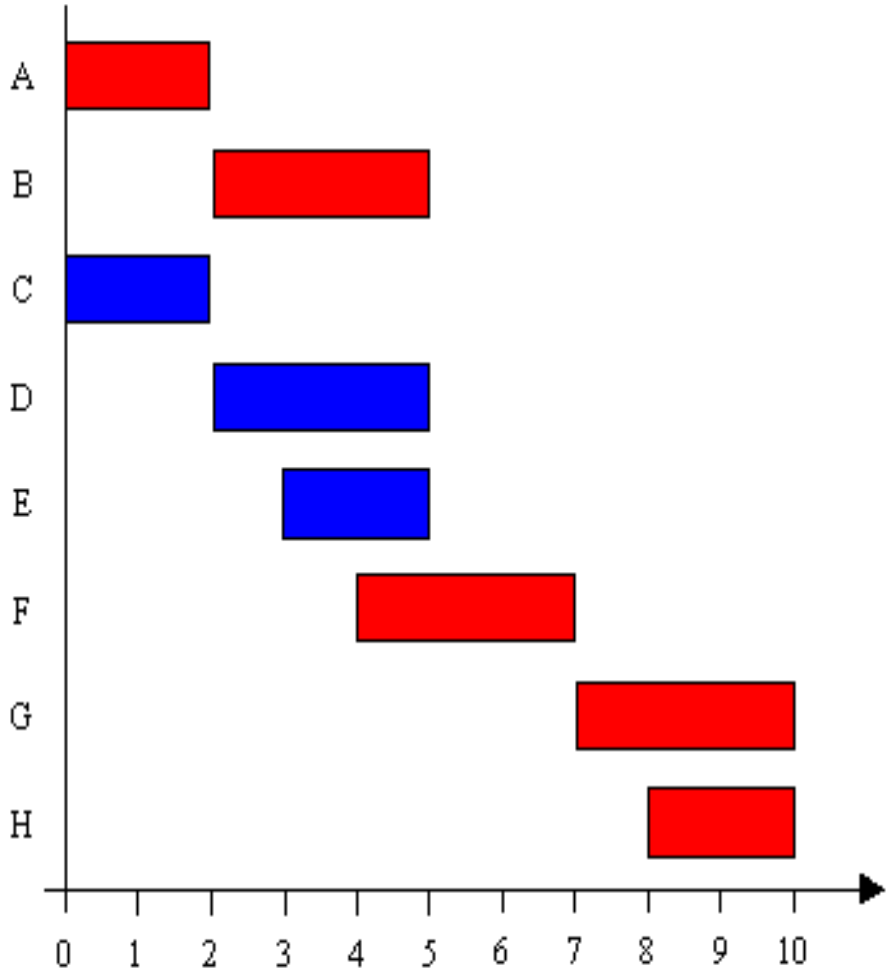
- La arquitectura de Von Neumann da nacimiento al Software como algo separado del Hardware (1945/49)
- Los primeros programas eran muy sencillos
  - Algoritmos matemáticos, estadísticos
  - Problemas de lógica y optimización
- ¿Como era el proceso de desarrollo de esos programas?

Historia - Antecedentes

Historia - Antecedentes



Historia - Antecedentes



GANTT CHART

Historia - Antecedentes

- En los comienzos la separación entre Hardware y Software era debil
- Los profesionales de software eran electrónicos, físicos o matemáticos
- Trataban a los productos de software como un producto industrial
- Las metodologias de construcción eran analogias de la producción en serie

Historia - Antecedentes

- Con el paso del tiempo los software y los sistemas se hicieron cada vez más complejos (ley de Moore)
- ¿Que paso con las viejas metodologias entonces?

Historia - Antecedentes

Historia - Antecedentes

Historia - Antecedentes

Historia - Antecedentes

- Nuestro objetivo hoy es conocer los ciclos de vida modernos y utilizados actualmente
- Pero antes tenemos que entender algo muy importante y fundamental
- There is no silver bullet!

Planificación - Decisión

FRED BROOKS

- La construcción de SW es un proceso complejo por naturaleza
- Dos tipos de tareas
  - Esenciales
  - Accidentales

Planificación - Decisión

- Esenciales
  - Inherentes a la naturaleza del sistema a construir
  - Relacionado con las necesidades del usuario
  - Modelo abstracto de solución
- Accidentales
  - Relacionadas con la concresión del producto
  - Como es llevado el sistema a la práctica

Planificación - Decisión

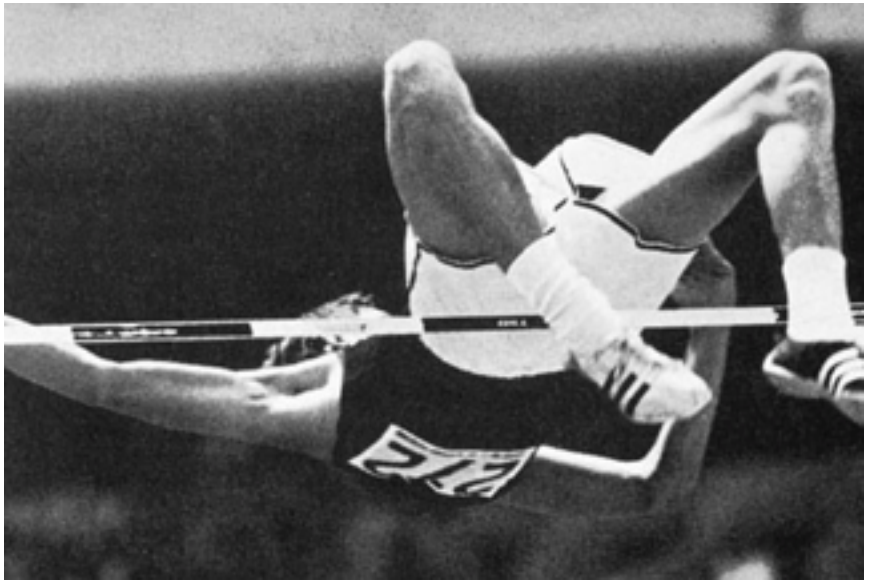
- ¿Que dice Brooks entonces?
- Enfocarse en las tareas accidentales nunca va a traer una mejora de un orden de magnitud
- Es en las tareas esenciales donde uno debe enfocarse para optimizar el proceso
  - Pero no es lo que se hizo en el pasado

Planificación - Decisión

- Ejemplo en cambios Esenciales

Valeri Brumel
poseedor del record
mundial 1968 altura 2,28m

Dick Fosbury
Medalla olimpíca 1968 marca 2,24 m



Planificación - Decisión

- Ejemplo en cambios Esenciales

Javier Sotomayor
actual record mundial
Salto en alto
marca 2,45 m

Planificación - Decisión

- ¿Que hace a un sistema tan complejo?
- Un producto de software tiene 4 caracteristicas que lo hacen complejo:

Planificación - Decisión

- Complejidad:
  - no hay 2 partes iguales, no existen elementos repetitivos
- Conformidad:
  - el producto tiene que adherirse a estandares y normas
- Cambiabilidad:
  - los productos raramente no sufren cambios, esto se debe a que existe la sensación de que el SW es mas fácil de cambiar que un producto físico
- Invisibilidad:
  - es una construcción abstracta, no puede examinarse el producto final en el plano físico con métodos de validación

Planificación - Decisión

- ¿Entonces?
  - No hay una solución que siempre funcione en un contexto (pensamiento sistémico)
  - Hay que analizar la esencia de cada problema
  - Como lo implementamos es accidental
  - La esencia se refiere a entender el problema, planificar la solución e implementarla
  - Siempre teniendo en cuenta el contexto del usuario
  - ¿Como lo resume Brooks?

Planificación - Decisión

"Lo difícil de construir Software es decidir qué decir, no decirlo"

Ciclos de vida

- Debemos decidir al comienzo que ciclo de vida se adecua mejor
- No existe una receta
- Es importante conocer las ventajas y desventajas
- Conocer las fortalezas y limitaciones

Ciclos de vida - Waterfall

Ciclos de vida - Waterfall

Ciclos de vida - Waterfall

- Con este modelo comienza la separación lógica
  - Especificar
  - Construir
  - Verificar y mantener
- Es muy estricto
- **No comienza una etapa hasta que no esta terminada la anterior**
- El usuario debe conocer todos los requerimientos de antemano y con mucho detalle
- No se tiene un producto hasta el final del proceso

Ciclos de vida – Waterfall

- El modelo original funcionó muy bien durante mucho tiempo
- Los proyectos en ese momento (hasta la decada del '80) consistían en informatizar procesos que ya existian
- El usuario sabía de antemano los requerimientos
- Pero con el tiempo los sistemas empezaron a ser más complejos, cambiantes, y lo que se definía al principio podia no ser cierto durante todo el proceso de desarrollo
- ¿Que pasaba si el usuario cambiaba de parecer?

Ciclos de vida – Waterfall

¿Recordé decirte antes que terminaras de codificar , que el usuario cambió las especificaciones?

Ciclos de vida – Waterfall

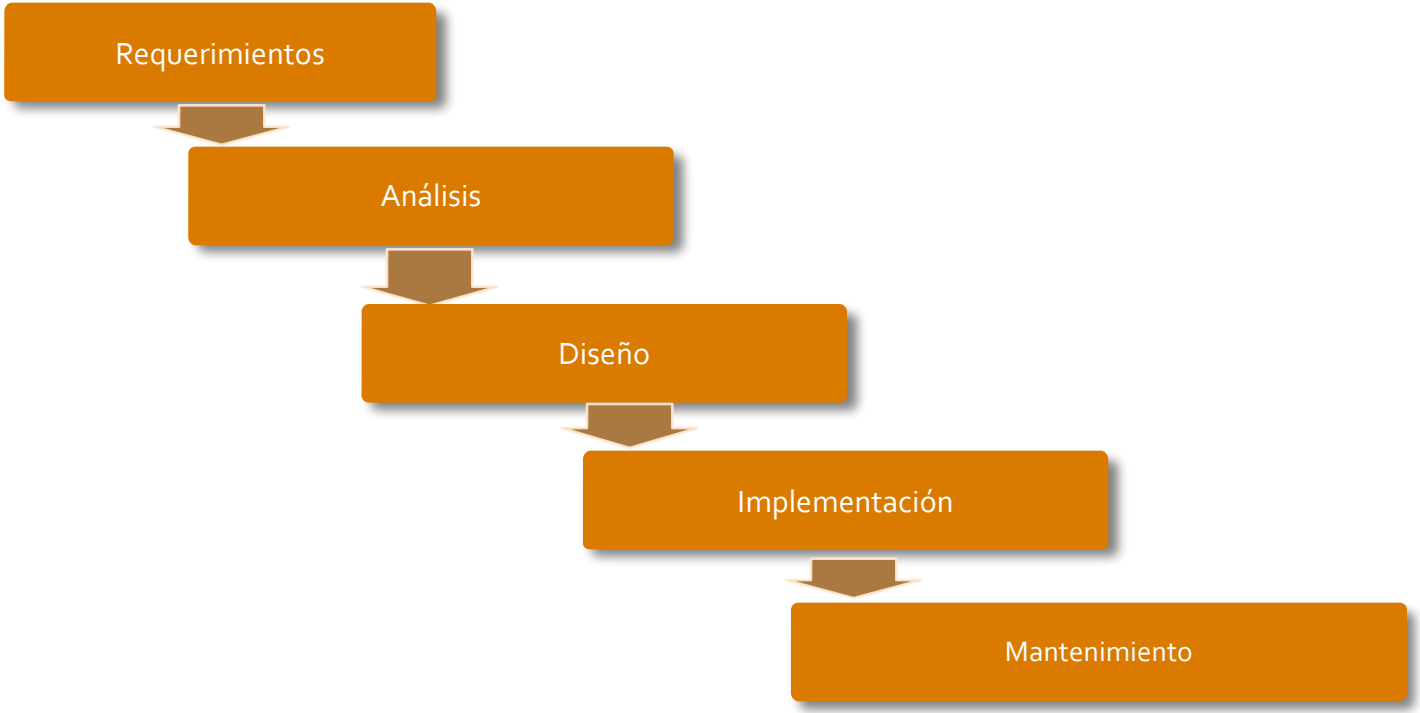
- El modelo original era muy…

Ciclos de vida – Waterfall

INFLEXIBLE

Ciclos de vida – Waterfall (Sashimi)

- Modelo en cascada modificado (moderno)
- Las etapas se pueden solapar

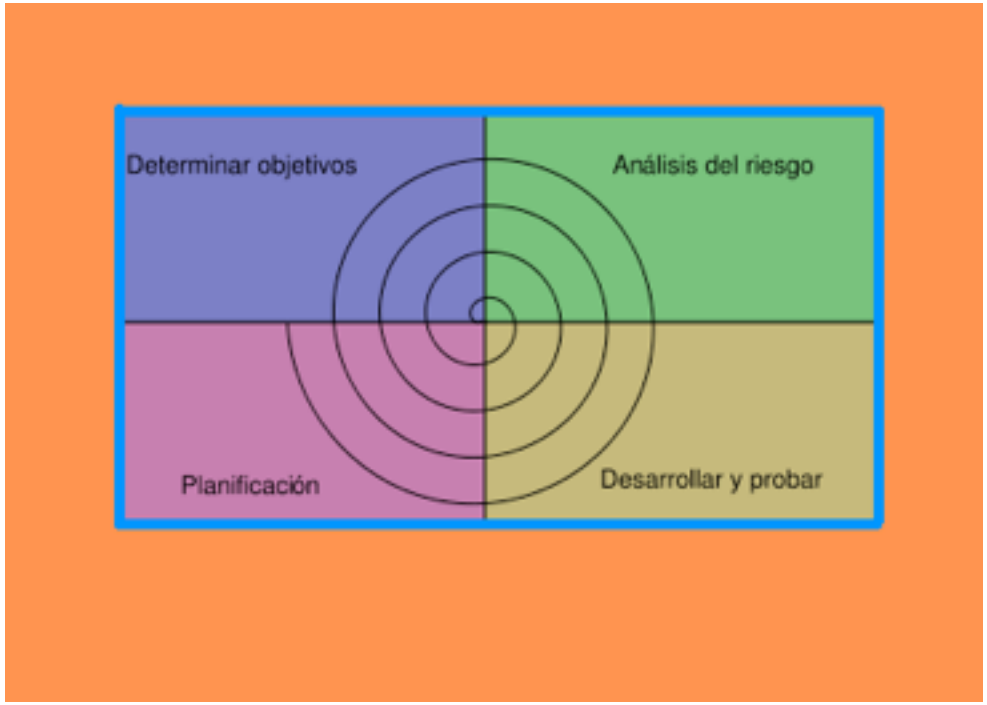


Ciclos de vida – Waterfall (Sashimi)

- ¿Y si Da Vinci hubiera usado esta metodologia?

Ciclos de vida - Espiral

- Desarrollado por Boehm en 1988
- Las actividades se conforman en una espiral



Ciclos de vida - Espiral

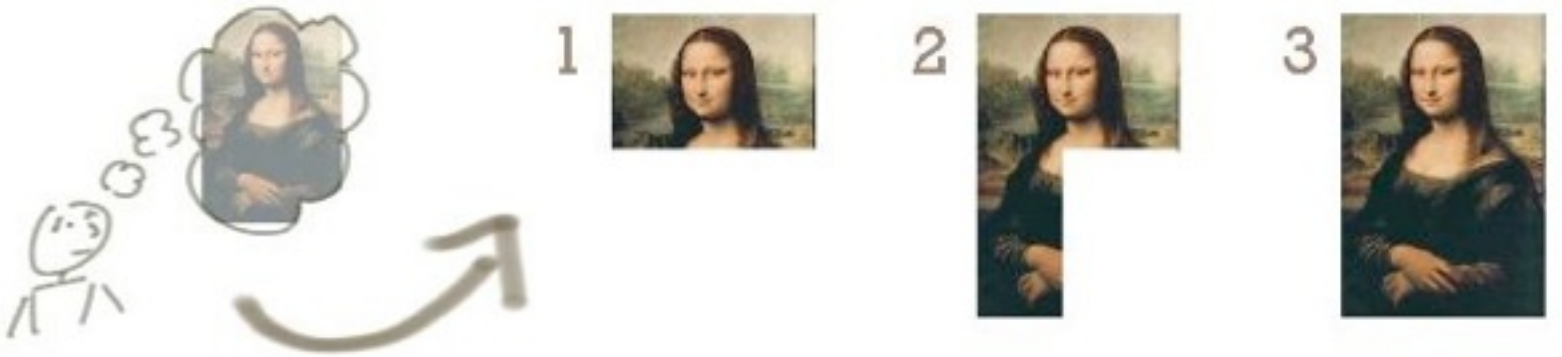
- Se divide el sistema en subproyectos
- Cada ciclo de la espiral incrementa el tamaño del sistema
- El núcleo de cada iteración es el **manejo de riesgos**
- En cada iteración se elige la alternativa de menor riesgo
- Se utiliza para tener una especificación y un producto de mucha calidad
- Es muy costoso de implementar y complicado de gestionar

Ciclos de vida – Iterativo e incremental

- Surge para paliar las debilidades de Waterfall
- Iterativo e incremental no son sinónimos

Ciclos de vida – Iterativo e incremental

- Incremental
  - Se divide el problema en partes
  - Se planifican
  - Se integran todas al final del proceso



Ciclos de vida – Iterativo e incremental

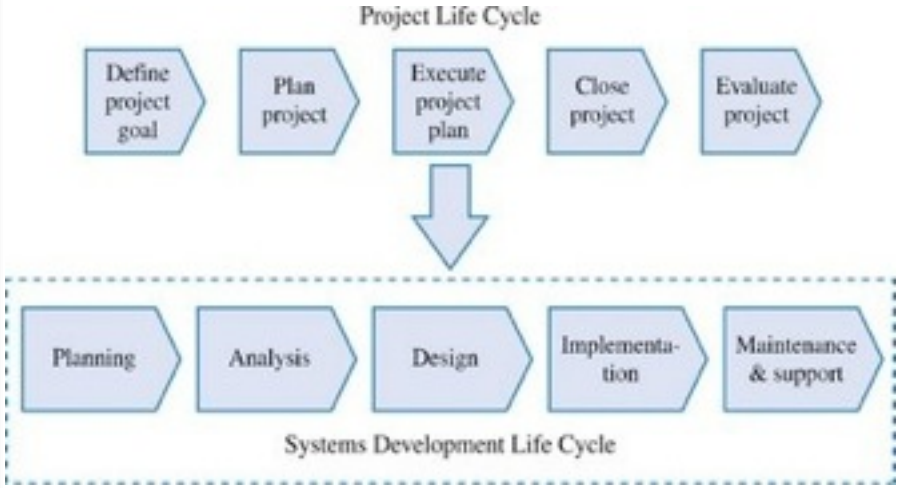
- Iterativo
  - Se utiliza para corregir los errores del incremental, aprender de lo hecho
  - Nos reservamos tiempo para volver sobre lo hecho y mejorarlo

Ciclos de vida - Agile

- Surge en el 2001
- Esta basado en el ciclo de vida iterativo e incremental
- Va más allá y propone un cambio de cultura
- Su piedra fundamental es el "Agile manifesto"

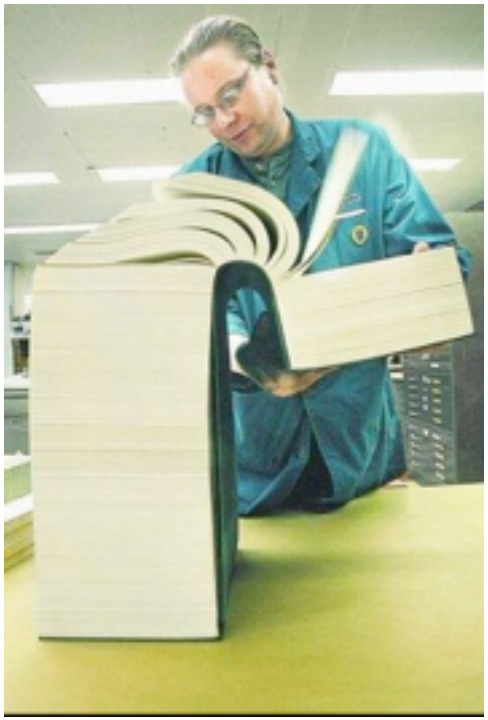
Ciclos de vida - Agile

- Agile Manifesto
  1. Personas e interacciones por sobre procesos y herramientas



Ciclos de vida - Agile

- Agile Manifesto
  2. Software funcionando por sobre documentacion exhaustiva



Ciclos de vida - Agile

- Agile Manifesto
  3. Colaboracion con el cliente por sobre negociacion contractual

Ciclos de vida - Agile

- Agile Manifesto
  4. Responder a los cambios por sobre seguir un plan

Ciclos de vida - Agile

- Tiene 3 grandes valores
  - Integridad: Ser consistente entre lo que se piensa y lo que se hace
  - Transparencia: Decir lo que pasa. Hablar, comunicar. Si hay desvíos, decirlos
  - Compromiso: Surge de tener los otros 2
- Existen distintas técnicas
  - SCRUM
    - planificación sprint
    - sprint (tiempo fijo-"sin cambios"), scrum (reuniones diarias)
    - revisión
    - equipo de desarrollo reducido (3 a 9)

Ciclos de vida - Agile

- XP
  - preferencia por modelo OO
  - etapas
    - planeación
    - diseño
    - [diseño de pruebas] codificación
    - prueba
  - programación en parejas
- TDD -test-driven-development
  - Se diseñan los test antes de desarrollar código
- Otras…

Ciclos de vida – Code & Fix

- Conocido como el "No ciclo de vida"

Ciclos de vida – Code & Fix

- Cuando no pensamos en un ciclo de vida, esto es lo que hacemos
- Produce productos de baja calidad
- No hay previsibilidad
  - No se sabe cuando se va a terminar
  - Porque no se sabe lo que hay que hacer

Conclusiones

- Elegir el ciclo de vida es una decisión del proyecto
- No hay Silver bullets (pensamiento sistémico)
- Entender ventajas y desventajas
- **Comunicar y explicar al cliente, que sea parte de la decision**

Preguntas