

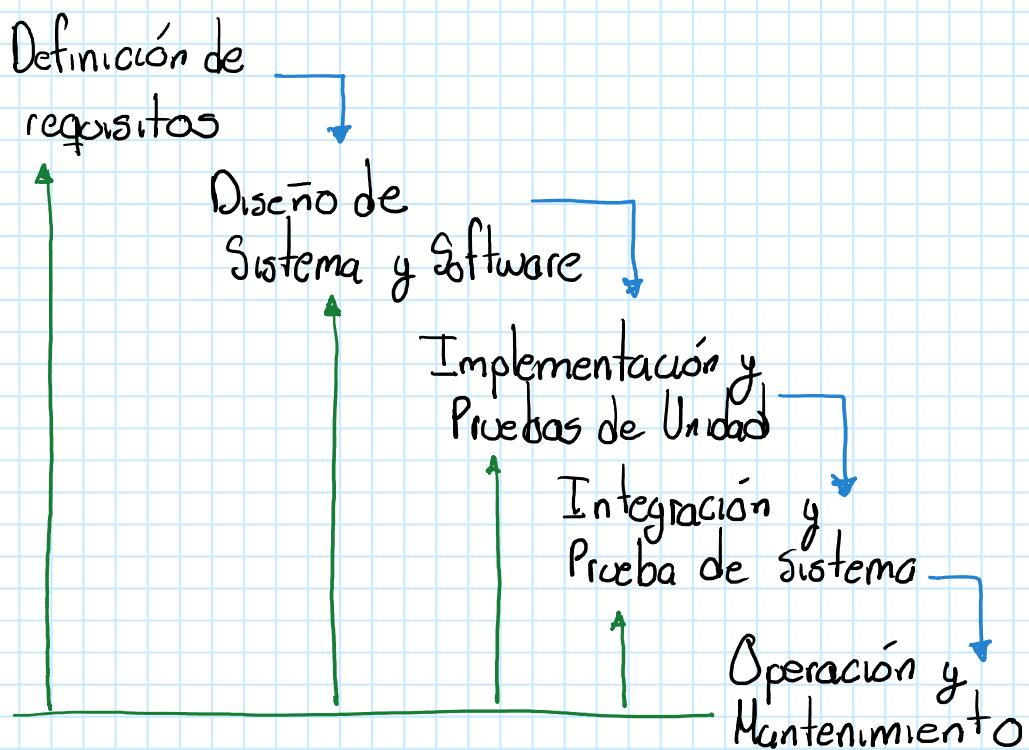
## 2.1\_Modelos

Thursday, September 5, 2019 7:27 PM

### ► Modelos para desarrollo de sistemas

- **Modelo:** Es una representación de un sistema

### ► Modelo en Cascada



// Cada documento tiene como resultado documentos que deben ser aprobados por usuario

// Incluye la corrección de los problemas de las fases previas

// Deben usarse si se entienden los requisitos

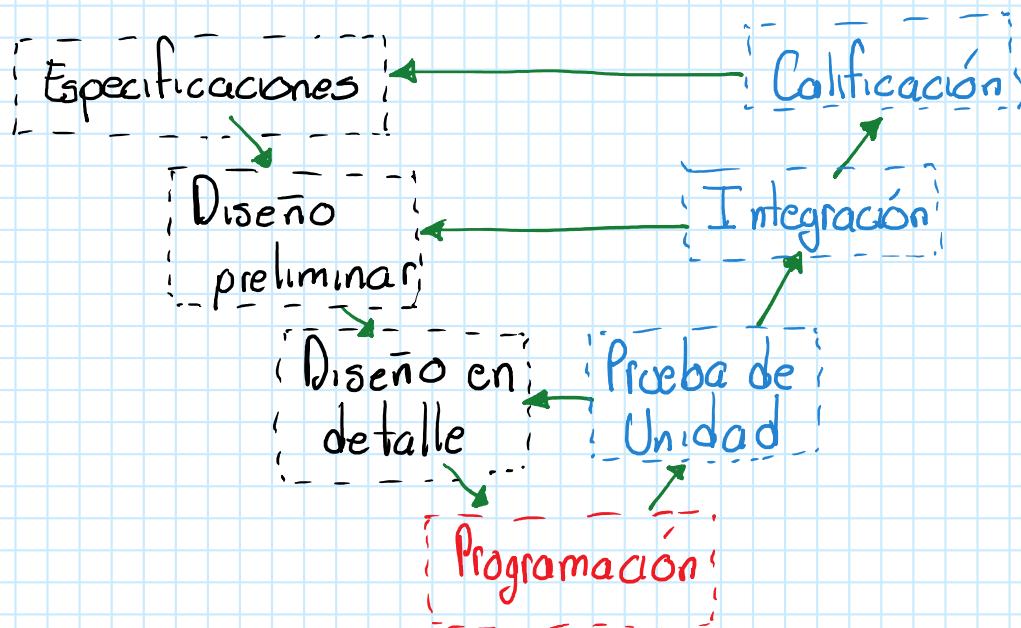
- **Ventajas**

- Planificación sencilla

- Calidad del producto alta

- Planificación sencilla
- Es conocida por desarrolladores
- Desventaja
- Complicado el cambio
- Se genera mucho mantenimiento inicial (Período de congelación de requisitos)
- Calidad del producto alta
- Usuarios comprenden fácilmente
- Errores de análisis y diseño difíciles de identificar y eliminar

## ► Modelo en V



// Los procedimientos para probar las especificaciones de la aplicación deben haberse creado en la fase de diseño

### • Ventajas

- Hace explícita la tarea en cada iteración
- Las pruebas en cada fase ayudan a corregir posibles errores
- Con pruebas unitarias y de integración se consigue

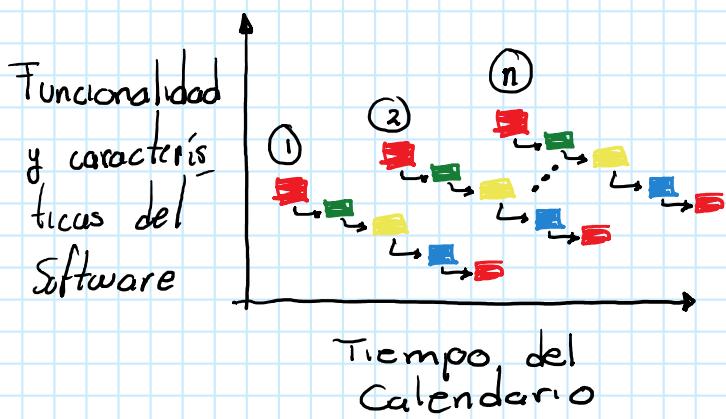
POSITIVOS - NUEVE

- Con pruebas unitarias y de integración se consigue obtener exactitud

### • Desventajas

- Al encontrarse errores luego de realizar las pruebas se pierde tiempo

## ► Modelo Incremental



• Se incrementan las funcionalidades respectivamente a como pasa el tiempo.

- Comunicación
- Planeación
- Modelado
- Construcción
- Despliegue

### • Ventajas

- Se reduce el tiempo de desarrollo inicial (func parcial)
- Se da énfasis en la planeación tanto administrativo y técnico
- Resalta más sencillo acomodar cambios al acotar el tamaño en los incrementos
- Es sencillo acomodar cambios al acomodar cambios al acotar el tamaño de los incrementos.

## • Desventajas

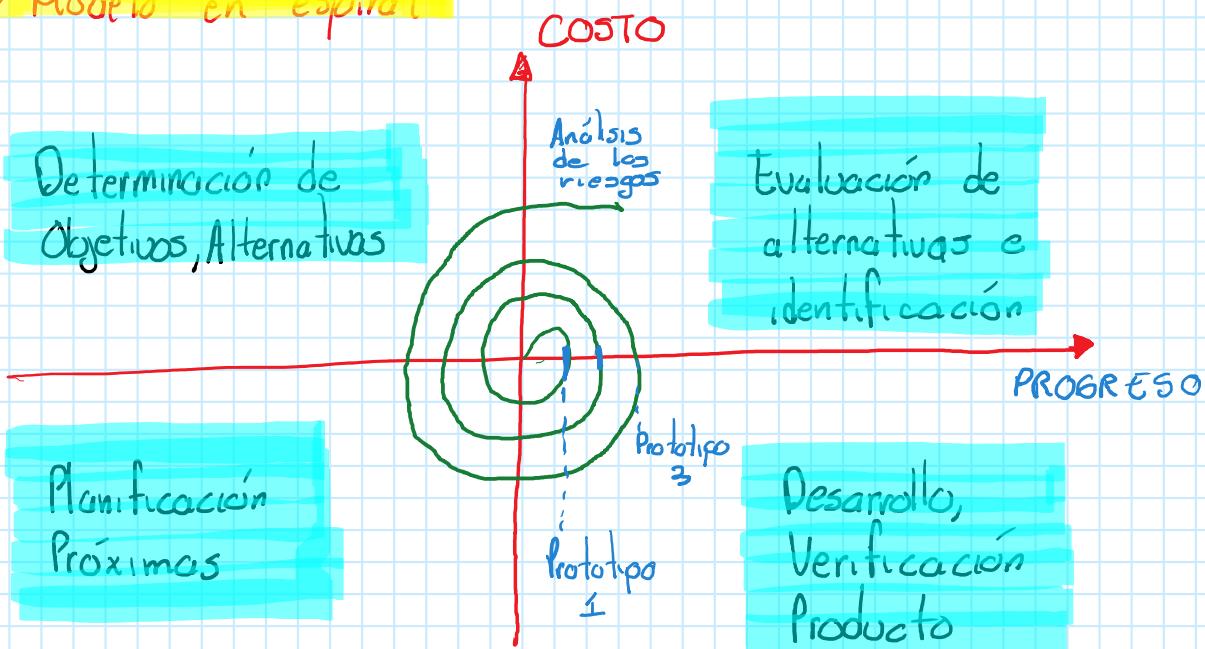
- Requiere mucha planeación
- Requiere metas claras para conocer el estado del proyecto
- No es recomendable para sistemas de tiempo real, de alto nivel de seguridad, de procedimiento distribuido y/o alto índice de riesgo.
- Se necesitan pruebas de regresión y su coste puede aumentar

## 2.2 Modelos

Tuesday, September 10, 2019

8:49 PM

### ► Modelo en espiral



#### • Ventajas

- Modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software
- Software evolucionará a medida que progrese el proyecto
- Aplica un enfoque de construcción de prototipos en cualquier proyecto para reducir riesgos

#### • Desventajas

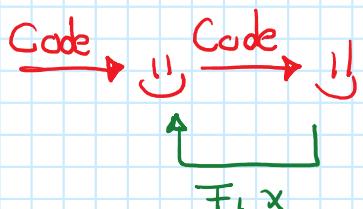
- Difícil convencimiento a grandes clientes
- Debido a su complejidad no se hace en sistemas pequeños
- Genera demasiado tiempo en el desarrollo del sistema
- C... I... I... - ... L... L... , ... , ...

Genera demasiados "tempo en el desarrollo del sistema

- Sin grupo de trabajo no se puede trabajar con este método.

### ► Code & Fix

- Es el modelo básico para desarrollo de software
  - ↳ Escribir código y corregir problemas en el código
- No conlleva ninguna gestión, no se pierde tiempo en la planificación
- Como se pasa directamente a codificar se puede mostrar directamente indicio de progreso



### ► Ventajas

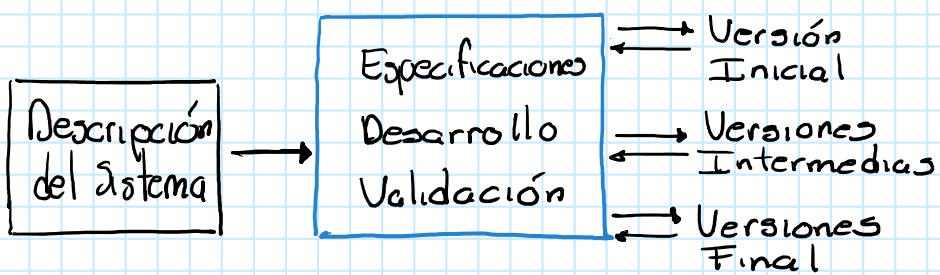
- No conlleva ninguna gestión
- Se muestran indicios de progreso rápido
- Requiere poca experiencia
- Para pequeños proyectos

### ► Desventajas

- Peligroso para proyectos no pequeños
- No ofrece evaluación del progreso
- No hay evaluación de calidad
- Probable "Diseño incorrecto".

### ► Desarrollo Evolutivo





- La idea detrás es hacer un sistema funcional desde el inicio
- Existen 2 tipos de variantes:

### Desarrollo Exploratorio

- Enfoque es explorar con el usuario los requisitos hasta llegar a un sistema final
- Sistema evoluciona conforme se añade nuevas características

#### ► Ventajas

- Especificaciones pueden desarrollarse de forma creciente
- Mejor entendimiento del sistema
- Cumple necesidades del cliente

#### ► Desventajas

### Enfoque utilizando prototipos

- Entender los requisitos del usuarios y trabajar para mejorar la calidad.
- Definir requisitos que no están claros y prototipos para experimentar
  - ↳ Prototipo - Definir requisitos

#### ► Ventajas

- Se crea con rapidez
- Reduce costos
- Aumenta probabilidad de éxito

#### ► Desventajas

- Administración difícil
- Adaptarlo como sistema

## ► Desventajas

- Proceso no visible, debido a que se hacen entregas no documentadas
- Pobremente estructurado
- Cambios continuos → Software Afecta

Adaptarlos como sistema final

- Surgen cambios imprevistos que retrasan el progreso

## ► Desarrollo Orientado a la Reutilización

- Consiste de 4 fases:

① Análisis de componentes

- Se determina qué componentes pueden ser utilizadas

② Modificación de requisitos

- Se adaptan en lo posible los requisitos para concordar con los componentes.

③ Diseño del sistema con Reutilización

- Se diseña o reutiliza el marco de trabajo para el sistema

④ Desarrollo e integración

- El software que no puede comprarse, se desarrolla.
- Integración como parte del desarrollo.

# METODOLOGÍAS

Tuesday, September 17, 2019

7:15 PM

## ► Lean

- Demanda de Ahorro de costos
  - Eliminación de Residuos
  - Organización →  $\xrightarrow{\text{Participación Colectiva}}$  Confianza → Menor Costo
- Optimizar todo  
- Se basa en una comprensión profunda del proyecto

## ► Open Up

← Me / No problem

## ► D.S.D.M

- Método de Desarrollo de Sistemas Dinámicos es un método que provee un framework para el desarrollo ágil de software
- Sensible a los requerimientos cambiantes
- RAD → "Desarrollo rápido de aplicaciones"
- Se creó con el objetivo de desarrollar en conjunto al cliente.

## // Principios de la metodología

- ① DSDM entrega frecuente de productos
- ② Todos los cambios durante el desarrollo son controlados

② Todos los cambios durante el desarrollo son reversibles

## // Fases del Diagrama

- ① Pre - proyecto
  - ② Estudio de viabilidad
  - ③ Estudio de negocio
  - ④ Iteración de modelo funcional
  - ⑤ Iteración de diseño y desarrollo
  - ⑥ Implementación
  - ⑦ Post - desarrollo
- 
- 
- 
- 

## ► Ciclo de Vida del software

① Requerimientos | • Minutas | Resultados de las entrevistas con el clientes

② Análisis de requerimientos | • Identificar elementos con los que realizaremos el software  
| • Matriz de requerimientos  
|   ↳ Funcional o no funcional  
| • Determinar la arquitectura del proyecto y la metodología

③ Diseño | • Minutas | Planificación del software tanto para reuniones con clientes como con equipo de trabajo  
| • UML | • Diagrama de flujo  
|   · Diagrama F/I

- UML
- Diagrama de flujo
- Diagrama E/R
- Casos de uso
- Diagrama de Gantt  
(Dependencia de actividades)

④ Desarrollo

- Escribir códigos a través de las tecnologías puntuadas

⑤ Pruebas

- Matriz de pruebas
- Pruebas
  - Unitarias Cada pequeño elemento cumpla su función
  - Interfaz Cuidar la comunicación con los elementos
  - Sistema

⑥ Mantenimiento

- Manual técnico
- Manual de usuario

---

### ▶ Crystal Clear

• Creada por "Alistair Cockburn"

- Propiedades
  - Entrega frecuente
  - Comunicación osmótico → "En el mismo cuarto"
  - Mejora reflexiva
  - ...

- Mejora reflexiva
  - Seguridad
  - Estrategias
    - Exploración 360°
    - Rearquitectura incremental
    - Radiadores de información
    - Victoria temprana
    - Esqueleto andante
- 
- 
- 
- 
- 

### ► RUP

- Rotational Unified Process

### // Previo e Historia

- Se utilizan elementos auxiliares como UML y Casos de Uso.
- Se lanza hasta 1998

### // Características principales

- ① Casos de uso
- ② Centrado en la arquitectura
- ③ Iterativo e incremental

### ► XP → Xtreme Programming

- Es una metodología ágil
- Principios
 

- Humanidad	- Flow
- Económicos	- Failure
- Self-Similarity	- Calidad
- Mejorar	- Baby steps
- Diversidad	