

Ciclos de Vida del Desarrollo de Software CVDS

Enrolment Key:
CVDS1

Ivan Dario Sanchez Fonseca



- Ingeniero de sistemas
- Esp. Ingeniería de software
- DevOps Leader (DOL) Certification
- SM, M3.0, Safe Agilist 4
- +15 años en proyectos de ingeniería de software
- + Arquitectura de software
- + Prácticas y cultura DevOps
- + Agilidad y equipos productivos
- + DevOps Days Lover

Cine | Running | Calistenia | Futbol | Tecnología | Agilista

 idsanchezf@gmail.com

 linkedin.com/in/idsanchezf

Ciclos de vida de desarrollo de Software - SDLC

- Proceso usado por la industria del software para diseñar, desarrollar y probar software de **alta calidad**. Tiene como objetivo producir software de alta calidad que alcance o supere las **expectativas del cliente**, y que sea finalizado en los **tiempos** y **costos** estimados.

La ingeniería de software y
la calidad de software nos
resultan temas
apasionantes...

Ser
humanamente responsable
nos resulta un reto

Intentar hacer las dos
primeras sin la última es
claramente un **despropósito**



¿Dónde hay software?



Calidad?

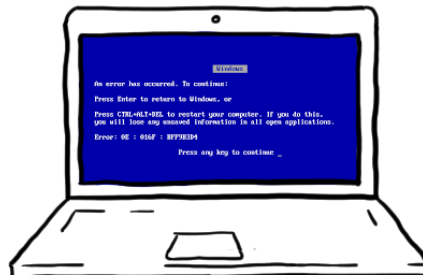
When the developer tests



When the quality team tests



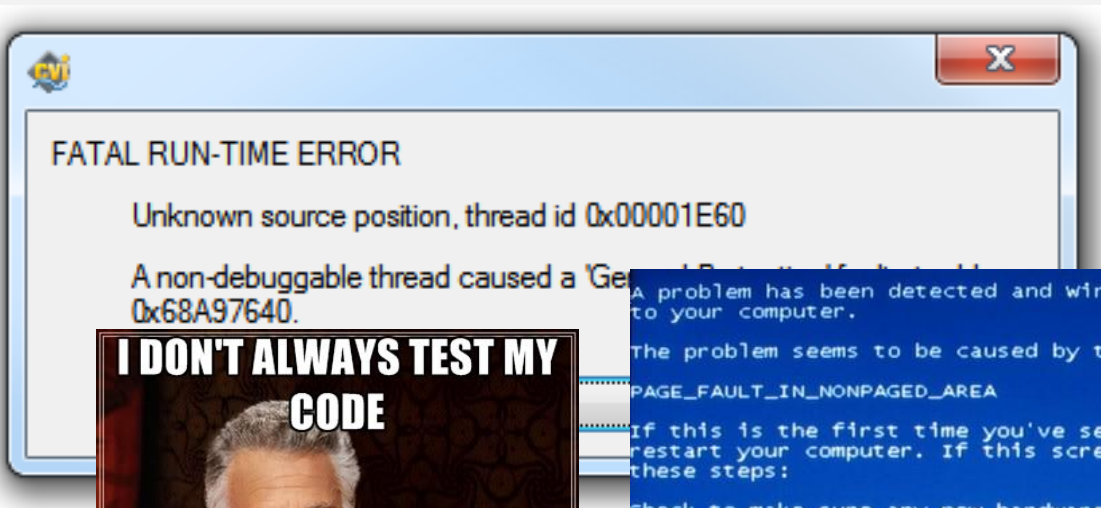
When the project manager tests



When the customer tests



Calidad?



```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

The problem seems to be caused by the following file: vsdatant.sys

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFFE00007,0x00000000,0xF379B038,0x00000000)

*** vsdatant.sys - Address F379B038 base at F376E000, DateStamp 49990c8f

Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
assistance.
```

¿Acaso en software no importa es
basicamente que funcione?

Veamos algunas respuestas a esa
pregunta....

Mariner 1

LA NASA 1962:

- El cohete **calculo mal la trayectoria** que debía seguir.
- Tuvo que ser destruido segundos después de ser lanzado **antes que se estrellara sobre el océano Pacífico.**
- el problema estaba en una fórmula escrita a lápiz que luego fue "**inadecuadamente**" trasladada al lenguaje informático

1996 –Ariane 5

- Estalló a 3700 mts de altura después de su lanzamiento.
- Error: “operand error” no controlado del código ADA (conversión de un número flotante de 64 bits a un entero de 16 bits).
- Subrutina re-utilizada del Ariane4.
- Costó más de US\$500 millones
- Transportaba cuatro satélites
- El desarrollo requirió cerca de 10 años de trabajo
- Pérdidas totales: US\$1.8 billones

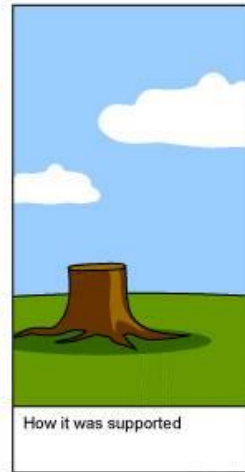
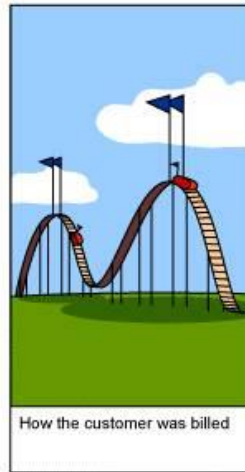
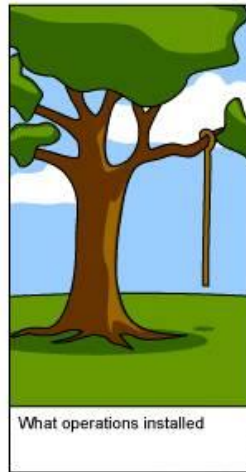
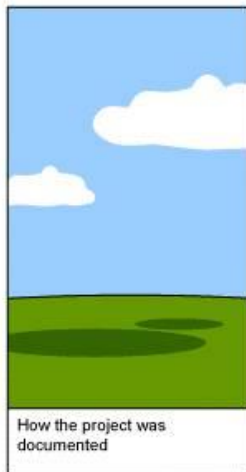
¡Si, ya sé!

**Usted no hace software para
aviones o para la NASA**

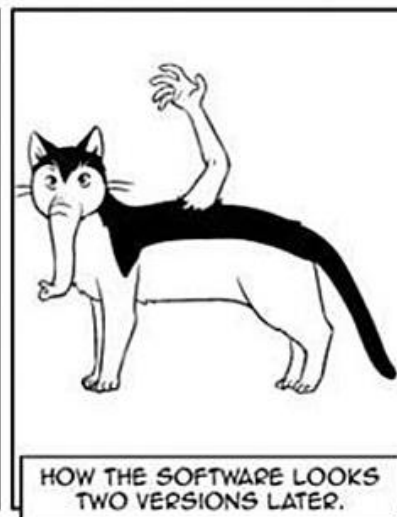
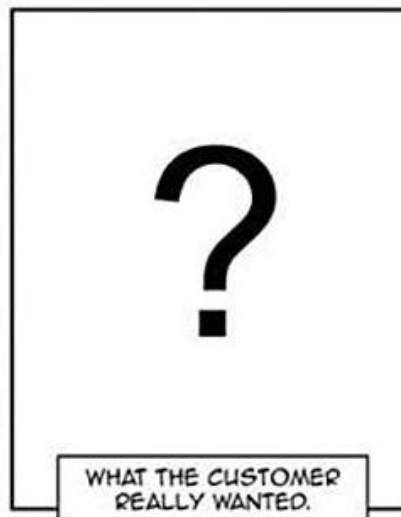
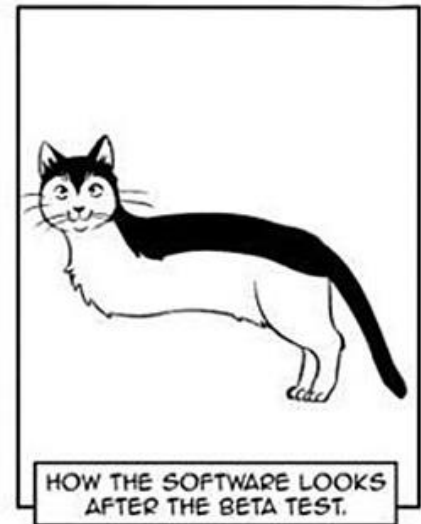
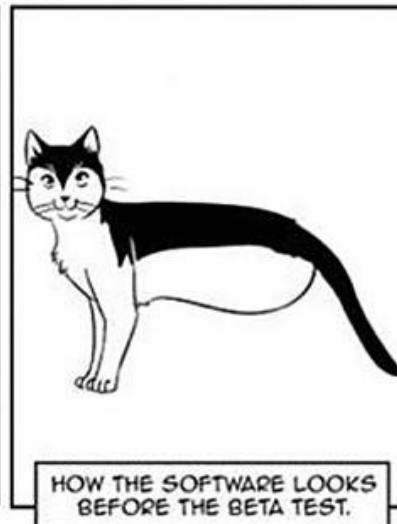
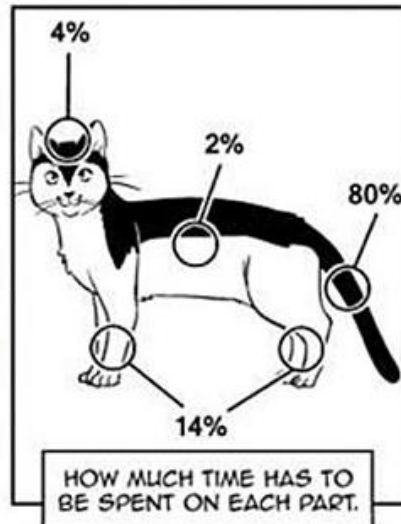
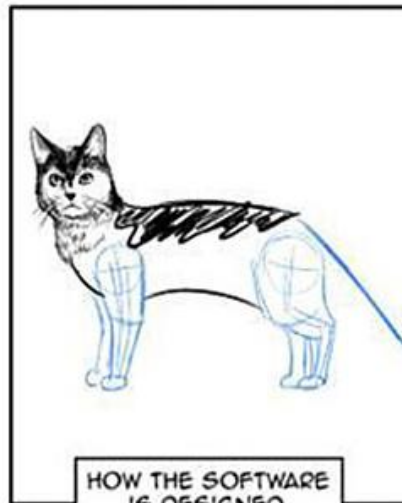
Lo que los estoy invitando a
cuestionarse es ...

**Es común ver software hecho
para quienes lo desarrollaron y
no para quienes lo van a usar...**

Expectativas/Necesidades del cliente?



Expectativas/Necesidades del cliente?

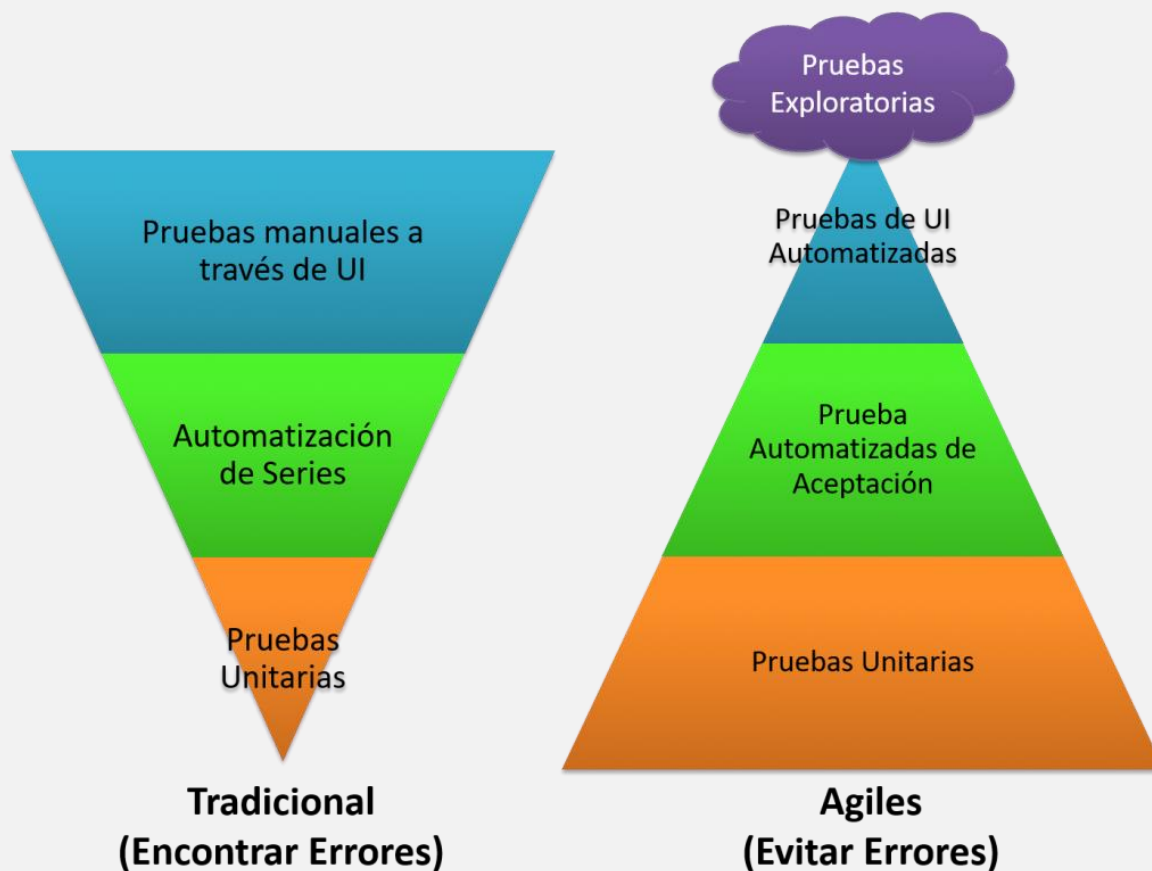


Tiempos y costos?

you can't have failing unit tests
if you do not write unit tests



Tiempos y costos?



Tiempos y costos?










REFUCTORING

purposefully making your program's code so convoluted and ludicrous that only you can understand it.



ZDNet

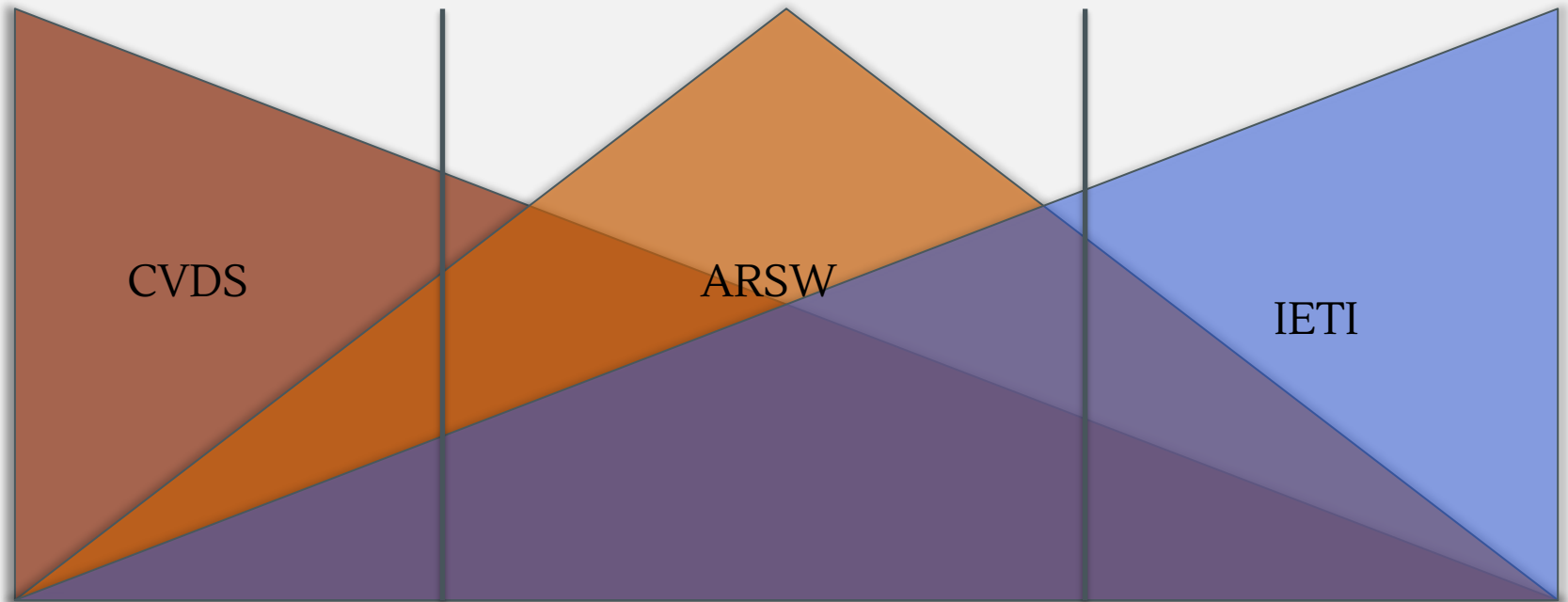
Tiempos y costos?

Name
 120525_document_updated.txt
 120604_document.txt
 120605_document_amended.txt
 120605_document_John.txt
 120605_document_latest.txt
 120605_document_latestcopy.txt
 120605_document.txt
 1200602_document.txt
 document_meeting.txt



This is only going to
end in tears.

Línea de Ingeniería de Software



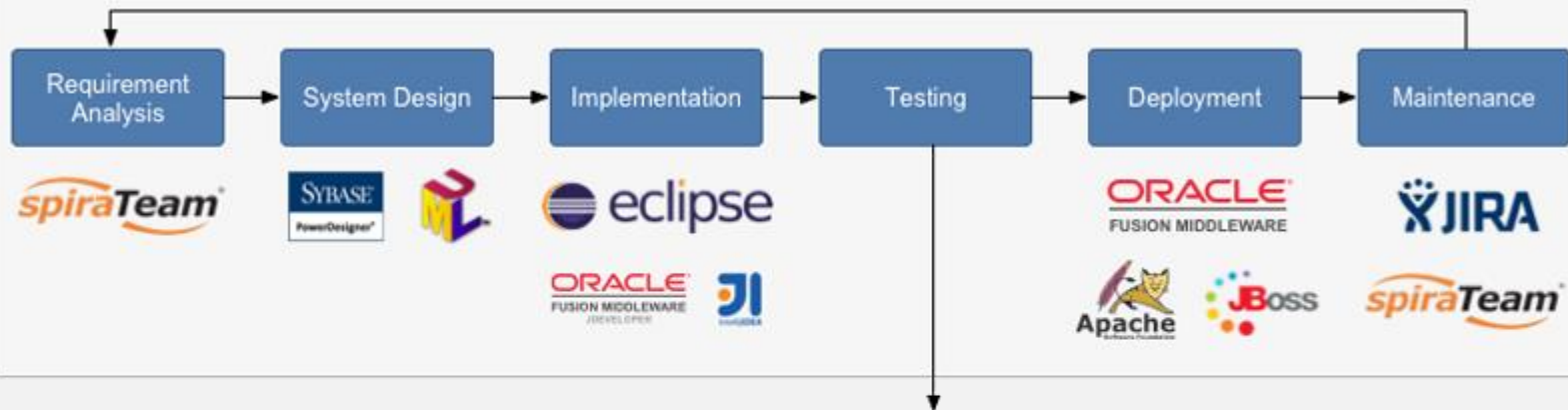
Procesos/PM/Entornos/Herram.

Estilos arquitectónicos

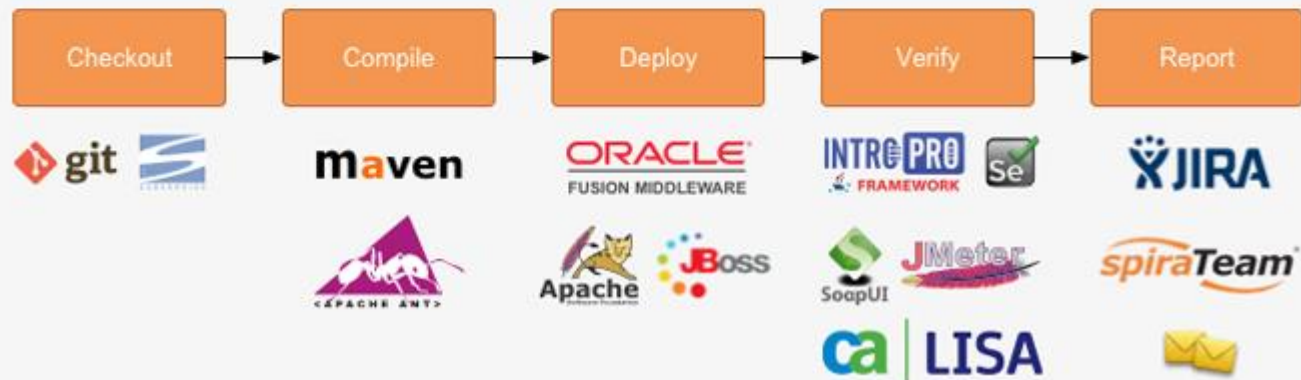
Diseño de productos/Innovación

Flujo – desarrollo de software

Software Development Life Cycle



Continuous Integration and Test Automation



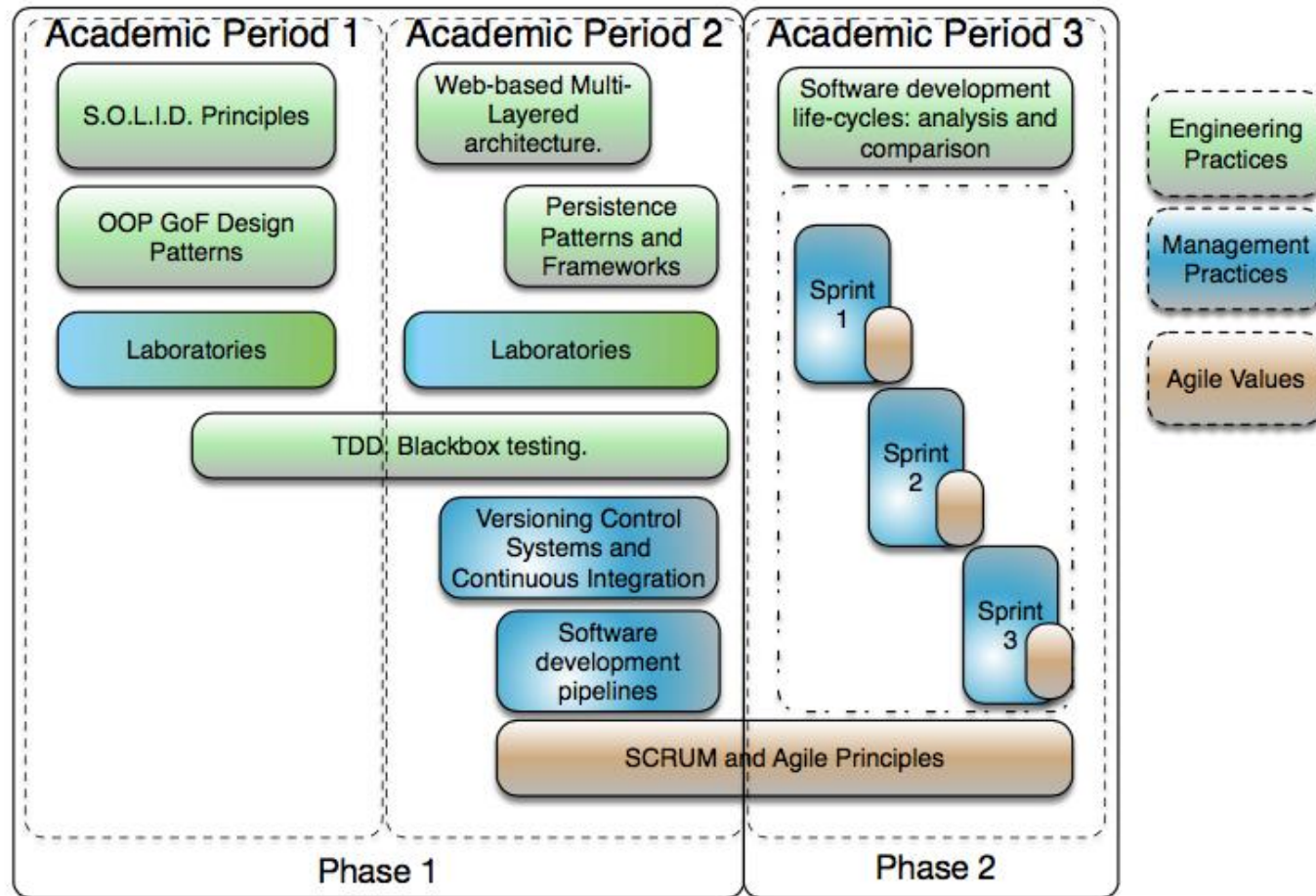
Contenido

SEMANA	Sesión teórica	Laboratorio
1	Introducción al curso	Introducción a GIT
2	Principios SOLI(D) - DiP - O/C	Caso de estudio SOLID - Ejercicios simples -
3	Diseño de pruebas. Ejercicio en grupos - diseño de pruebas caso X	TDD - Implementación de pruebas y análisis de cubrimiento - caso X. Maven - fase de pruebas.
4	Principios SOLI(D) - Contenedores livianos e Inyección de dependencias vs Patrones Creacionales.	Ejercicio Contenedor IoC + TDD. Caso para
5	(En laboratorio) Introducción - Desarrollo Web.	Introducción al Desarrollo Web. Arquitecturas
6	Recapitulación	Parcial teórico/práctico (Límite entrega
7	Introducción a la integración continua.	Despliegue de aplicaciones. Integración y
8	Patrones de persistencia. Principios ACID.	Patrones de persistencia. Mapeo de modelos
9	Introducción metodologías ágiles y SCRUM.	Integración de capas: presentación, lógica y
SEMANA DE RECESO		SEMANA DE RECESO

Contenido

10	Metodologías ágiles y SCRUM. Juegos ágiles	PARCIAL teórico/práctico (Límite entrega
11	Introducción al proyecto. Backlog de producto. Backlog de sprint: estimación y planeación.	
12	*Seguridad en aplicaciones Web (por definirse)	
13	*Tiempo para el proyecto + tiempo de soporte opcional.	Daily Scrum - seguridad Web
14	Modelos de ciclo de vida y metodologías tradicionales	Sprint review/Sprint retrospective/Sprint
15	Modelos de ciclo de vida y metodologías alternas (cont): UP	Daily scrum - soporte proyectos
16	Modelos de ciclo de vida y metodologías alternas (cont): Roles RUP, RUP vs Scrum	
exámenes		

Metodología



Metodología

- Primero y segundo tercio.
 - Apropiación de conceptos básicos:
 - Patrones de diseño.
 - Principios de diseño.
 - Metodos ágiles.
 - Desarrollo dirigido por pruebas.
 - Aspectos técnicos de desarrollo de aplicaciones empresariales básicas: Desarrollo Web, persistencia, etc.
 - Implementación de líneas de producción de software (construcción, verificación y despliegue automático).

Metodología

- Tercer tercio:
 - Construcción de una solución requerida por una entidad/persona real.
 - Revisión y comparación de SDLC alternativos.

Evaluación

Evaluación

Primer y segundo tercio

Parcial (teórico+práctico)	.	.	.	50%
Laboratorios	.	.	.	40%
Quices y talleres	.	.	.	10%

Tercer tercio

Parcial (Teórico+documentación del proceso)				50%
Proyecto.	.	.	.	50%

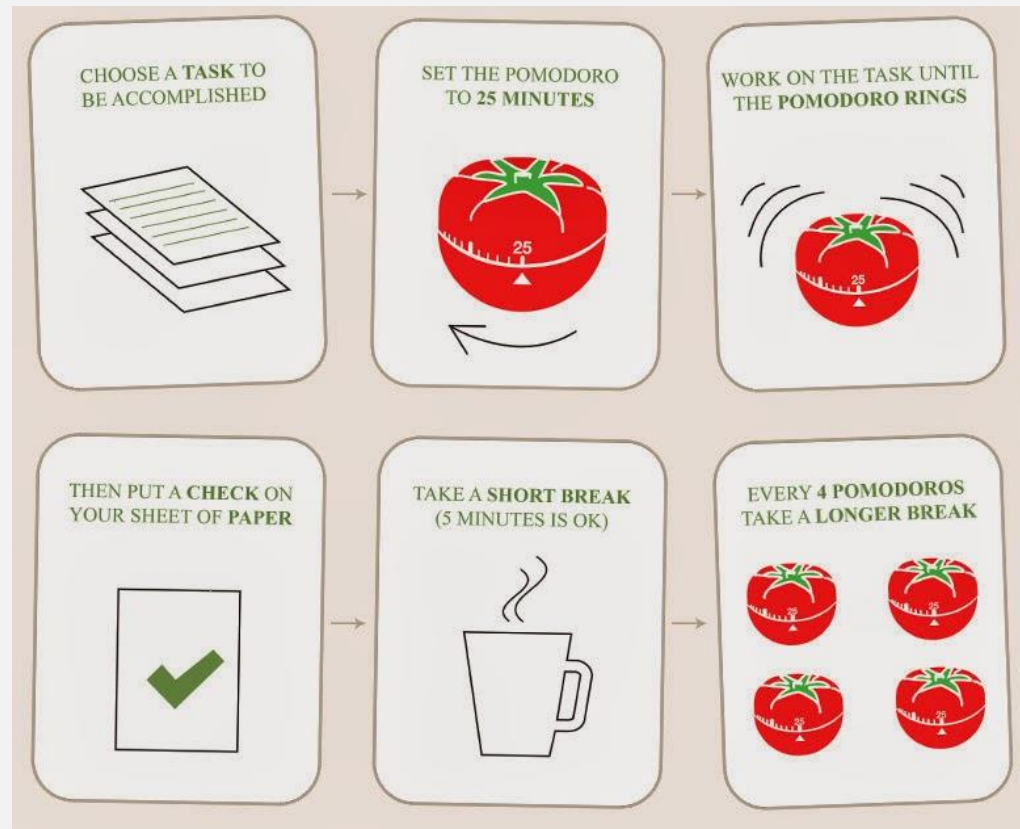
Reglas de juego

- Monitoreo de trabajo individual.
- Colaboración entre grupos.
- Trabajo de laboratorios.
- Evaluación del proyecto vs evaluación individual.

Manejo del tiempo



Técnica pomodoro



Preguntas?

A large, bold, black question mark is centered on the page. It is a simple, sans-serif style question mark with a thick stroke and a solid dot at the bottom.

Para el laboratorio

1. Matricularse en el espacio de Moodle.