ACTUALIZACIÓN INFORMÁTICA

CPAP

CURSO DE POSGRADO

# Técnicas y Gestión de las Pruebas de Software

Darío Macchi

DOCENTE (invitado)

ACTUALIZACIÓN INFORMÁTICA

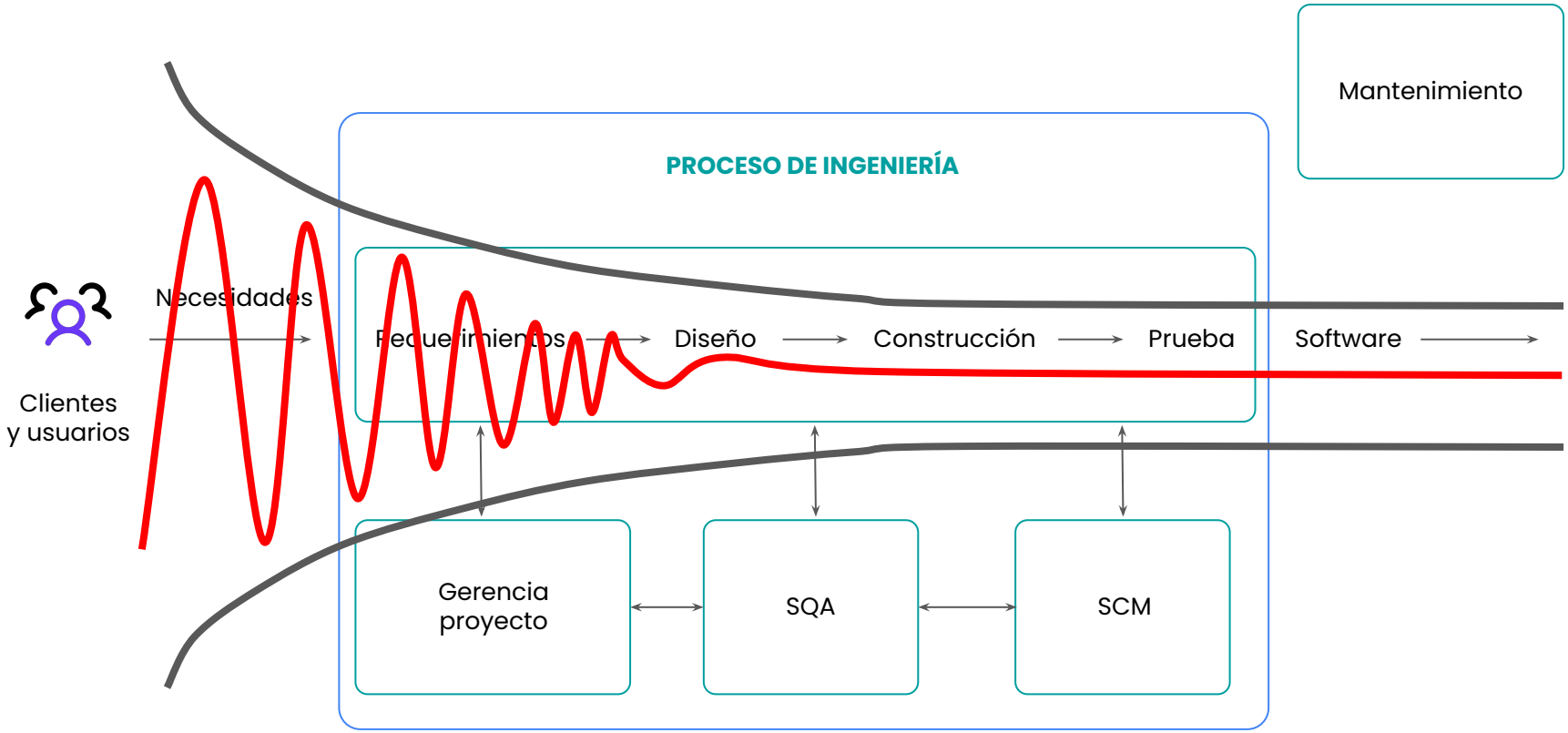
CPAP

# Gestión de proyectos de prueba

## Plan de pruebas

# Plan de pruebas

- Describe las **actividades** de prueba para un proyecto.
- Se ajusta conforme avanza el proyecto.



ACTUALIZACIÓN INFORMÁTICA
CPAP

# Plan de pruebas

Actividades:

- Determinar objetivos, alcance, y riesgos de la prueba
- Definir enfoque general de la prueba
- Integrar/coordinar pruebas en modelo de ciclo de vida
- Qué, cómo y cuándo se va a probar y recursos
  - Criterio de preparado y de hecho
- Establecer calendario de pruebas
- Métricas para monitoreo y control
- Presupuesto de actividades de prueba
- Determinar estructura de la doc. de prueba

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Plan de pruebas - Ejemplo

Ver ejemplo Plan de Pruebas mínimo

# Plan de pruebas

¿Cómo se podría adaptar el ejemplo anterior a un contexto ágil?

# Esfuerzo en testing (costos)

Depende de:

- La madurez del proceso de desarrollo
- Calidad y *testeabilidad* del software
- Infraestructura de pruebas
- Equipo y colaboradores
- Objetivos de calidad
- Estrategia de pruebas

¿El Gerente de Pruebas puede influir directamente en alguno de estos factores? ¿En cuáles?

# Estimación de esfuerzo en testing

Dos clásicos:

- Juicio de experto
- Basado en métricas

Regla general: hay que contar con dedicar entre el 25% y el 50% de todo el esfuerzo de desarrollo para pruebas (a todos los niveles).

En ágil, es importante planificar las pruebas y todas las demás tareas al planificar la iteración, incluyendo mantenimiento y la mejora de infraestructura de pruebas.

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Gestión de proyectos de prueba

## Monitoreo / Seguimiento

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo

Seguimiento del progreso de las pruebas → medir número de casos de prueba:

- previstos
- en progreso
- completados

**Software Development Kanban**

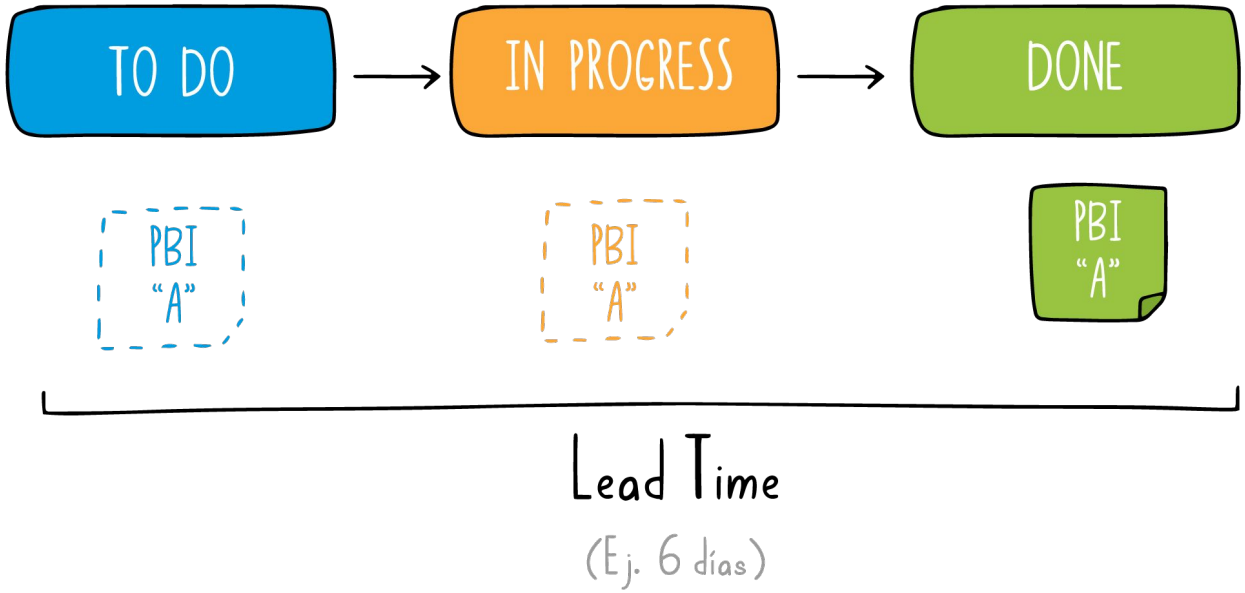
| Backlog | Analysis | | Design | | Development | | QA | | To Deploy |
|---|---|---|---|---|---|---|---|---|---|
| 6 | 3 | | 5 | | 4 | | 3 | | 6 |
| Feature U | Doing | Done | Doing | Done | Doing | Done | Doing | Done | Feature A |
| Feature V | Feature S | Feature R | Feature O | Feature M | Feature K | Feature I | Feature G | Feature F | Feature B |
| Feature W | Feature T | | Feature P | Feature N | Feature L | Feature J | Feature H | | Feature C |
| Feature X | | | Feature Q | | | | | | Feature D |
| | | | | | | | | | Feature E |

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo

Métricas de Kanban aplicadas a la gestión de pruebas

## Lead Time

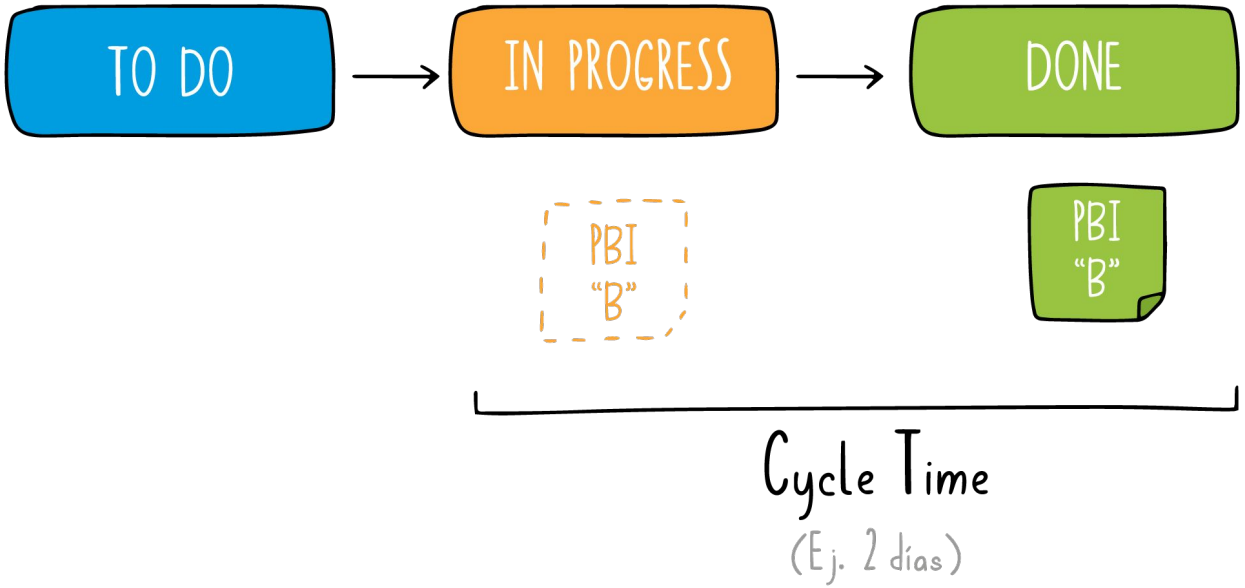


ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo

Métricas de Kanban aplicadas a la gestión de pruebas

## Cycle Time

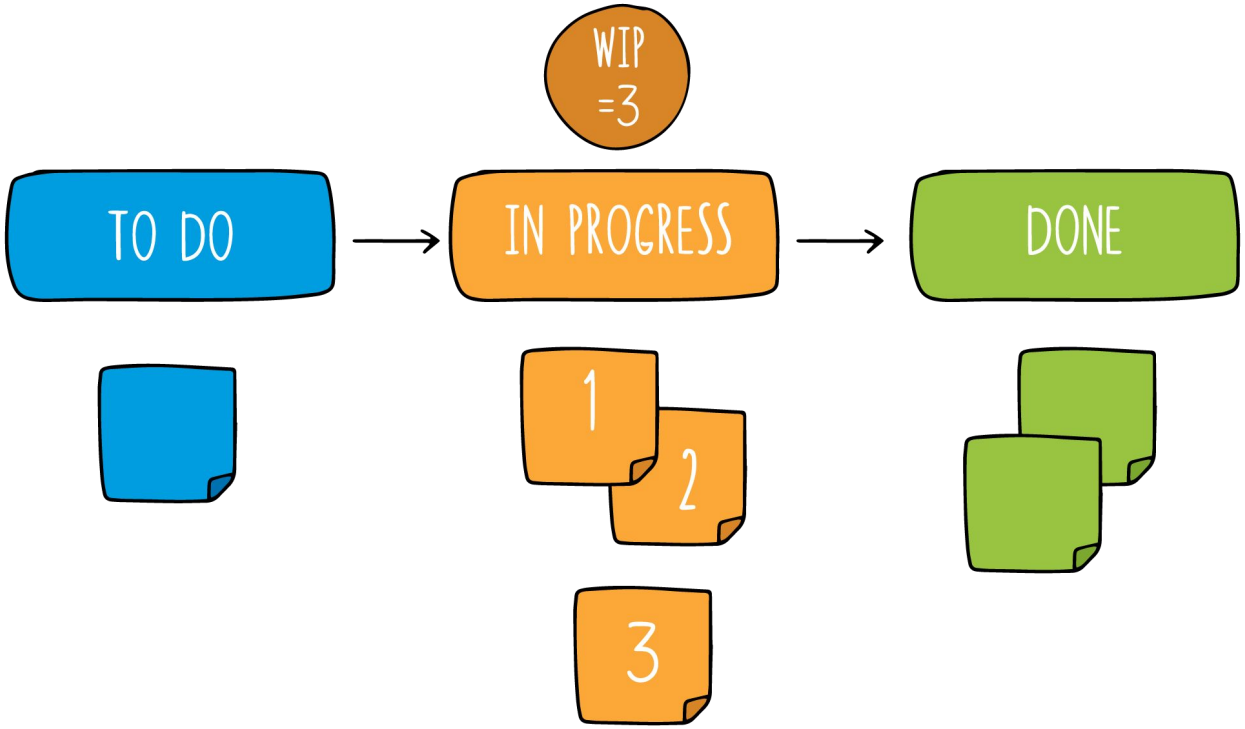


ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo

Métricas de Kanban aplicadas a la gestión de pruebas

**WIP**

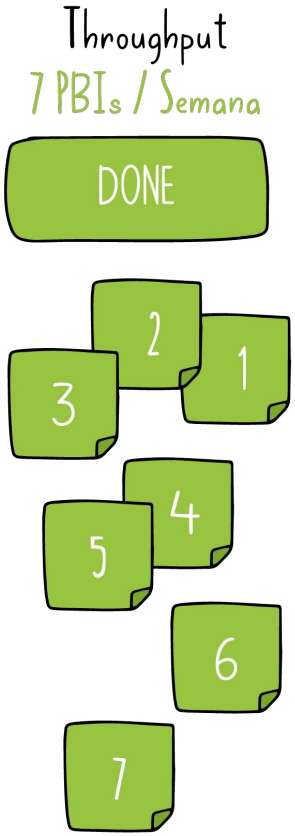


ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo

Métricas de Kanban aplicadas a la gestión de pruebas

## Throughput



ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo - métricas base

- # total de casos de prueba
- # de casos de prueba superados
- # de casos de prueba fallidos
- # de casos de prueba bloqueados

Filtrado por severidad

- # de defectos
  - encontrados
  - aceptados
  - rechazados
  - críticos
- # de horas de prueba previstas
- # de horas de prueba reales
- Tiempo medio de reparación de defectos

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Monitoreo - métricas calculadas

- Cobertura (eficacia)
  - Cobertura de ejecución de pruebas
    *top down* (módulos, funcionalidades, clases, funciones)
- Esfuerzo (eficiencia)
  - # de pruebas ejecutadas por hora
  - # de defectos por hora
  - # de defectos por prueba
  - Tiempo medio: de resolución de *bugs*, para testear un *bug fix*, etc.
- Calidad
  - Casos de prueba que pasaron
  - Casos de prueba fallidos
  - Casos de prueba críticos
  - Porcentaje de defectos corregidos

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Calidad - Métricas

**Densidad de defectos** = # defectos identificados / unidad

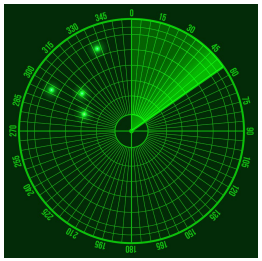
unidad: 100LOC, clase, módulo, paquete, etc.

**Eficacia de eliminación de defectos** = (# de defectos detectados durante las pruebas de control de calidad / (número de defectos detectados durante las pruebas de control de calidad + número de defectos detectados por el usuario final)) * 100

**Siembra de errores de software** (*bebuging)*

ACTUALIZACIÓN INFORMÁTICA
CPAP

# Calidad - Métricas



## Siembra de errores de software (*bebugging)*

Contaminar un software con errores artificiales y ejecutar pruebas. Luego, se calcula la cantidad de errores reales y artificiales descubiertos.

$$\frac{\text{Discovered seeded bugs}}{\text{Seeded bugs}} = \frac{\text{Discovered real bugs}}{\text{Real bugs}}$$

**tasa de detección de errores** = bugs plantados / detectados
**bugs reales** = bugs plantados x (bugs reales-detectados / bugs plantados-detectados)