



# Técnicas de revisión

Universidad Autónoma de Coahuila

Facultad de Sistemas

Calidad y Pruebas de Software

# Pruebas

Es el proceso de evaluar un producto, aprendiendo a través de la exploración y experimentación, lo cual incluye:

- Cuestionar
- Estudiar
- Modelar
- Observar
- Inferir
- Checar salidas de datos

# Principios de la prueba

Glen Myers:

- La prueba es un proceso que ejecuta un programa con objeto de encontrar un error
- Un buen caso de prueba es el que tiene alta probabilidad de encontrar un error que no se ha detectado hasta el momento
- Una prueba exitosa es la que descubre un error no detectado hasta el momento

# Principios de las pruebas

## Davis

- Todas las pruebas deben poder rastrearse hasta los requerimientos del cliente.
- Las pruebas deben planearse mucho antes de que den comienzo
- El principio de Pareto se aplica a las pruebas de software
- Las pruebas deben comenzar “en lo pequeño” y avanzar hacia “lo grande”
- No son posibles las pruebas exhaustivas

# Principios de las pruebas

Alan Page & Veren Genzen

1. Nuestra prioridad es mejorar el negocio
2. Nosotros aceleramos al equipo, usamos modelos como Lean Thinking y Teoria de las Restricciones para ayudar a identificar, priorizar y mitigar cuellos de botella en el sistema
3. Somos la fuerza para la mejora continua, ayudando al equipo a adaptarse y optimizar para tener éxito, en lugar de proporcionar una red de seguridad para detectar fallas
4. Nos preocupamos profundamente acerca de la cultura de calidad en el equipo, y asesoramos, lideramos y nutrimos el equipo para llevarlos a una cultura de calidad más madura
5. Nosotros creemos que el cliente es el único capaz de juzgar y evaluar la calidad de nuestro producto
6. Nosotros usamos datos de manera extensa y profunda para entender los casos de uso del cliente y entonces cerrar huecos entre hipótesis del producto e impacto del negocio
7. Expandimos las habilidades de testing y el conocimiento en todo el equipo; entendemos que esto reduce o elimina la necesidad de un especialista dedicado al testing



# Principios de validación

Terminando la codificación:

1. Realizar el recorrido de código
2. Ejecutar pruebas unitarias
3. Rediseñar el código

# Failure, Error, Fault, and Defect

---

**Anomalía:** la manifestación de un error en el software.

---

**Error:** una acción **humana** que produce un resultado incorrecto.

---

**Defecto:** imperfección o deficiencia en un producto, el cual no cumple sus requerimientos o especificaciones y necesita ser reparado o remplazado.


---

**Fallo:** el cese de la habilidad de un producto de cumplir una función requerida o su inhabilidad de funcionar dentro de márgenes previamente especificados.


---

**Problema:** dificultad o incertidumbre experimentada por una o más personas, como resultado de un encuentro insatisfactorio con el sistema usado.





"El **error** humano cometido inyecta un **defecto** en el software que, ocasionalmente, se observa como una **anomalía** a causa de un comportamiento incorrecto, no acorde a lo especificado, que finalmente provoca el **fallo** del sistema software"



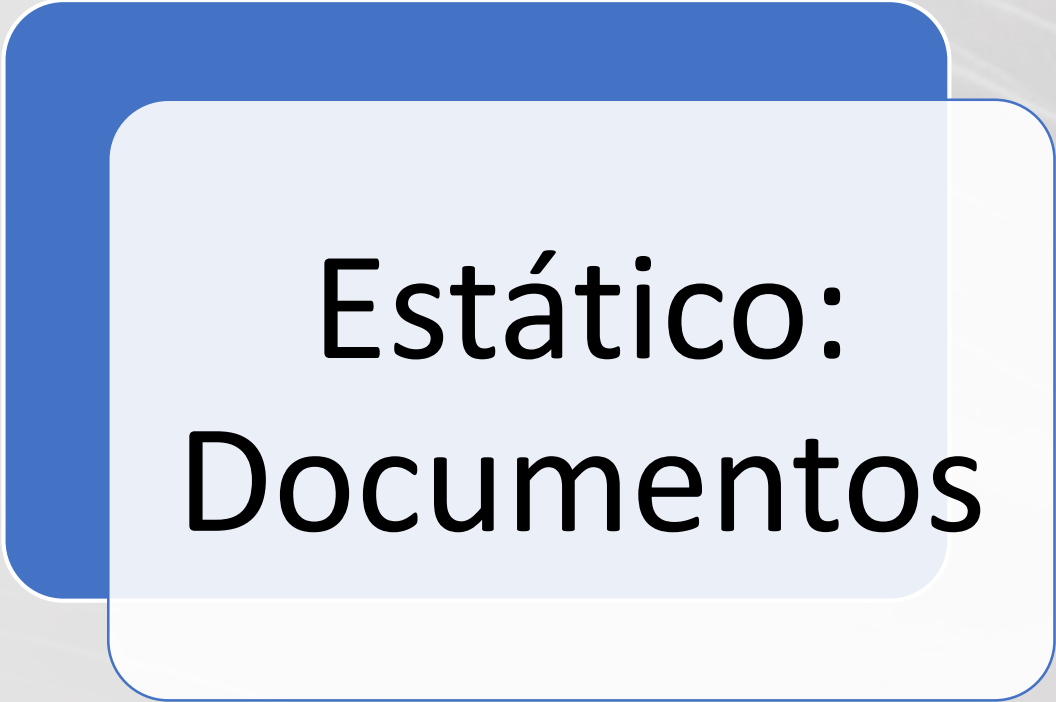


# Verificación y validación

**Verificación:** Asegurarse que en cada etapa se cumpla los requerimientos del cliente

**Validación:** Al final del proceso, comprobar con el cliente que todo esta bien

# Estático y Dinámico

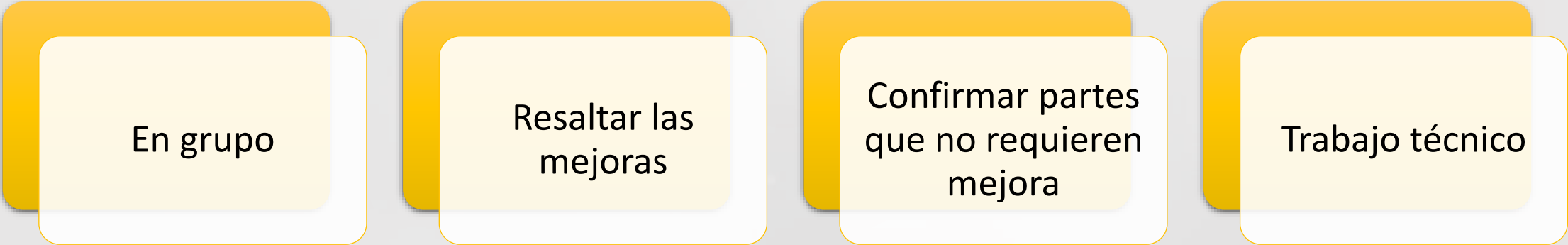


**Estático:  
Documentos**



**Dinámico:  
Código**

# Revisión



En grupo

Resaltar las  
mejoras

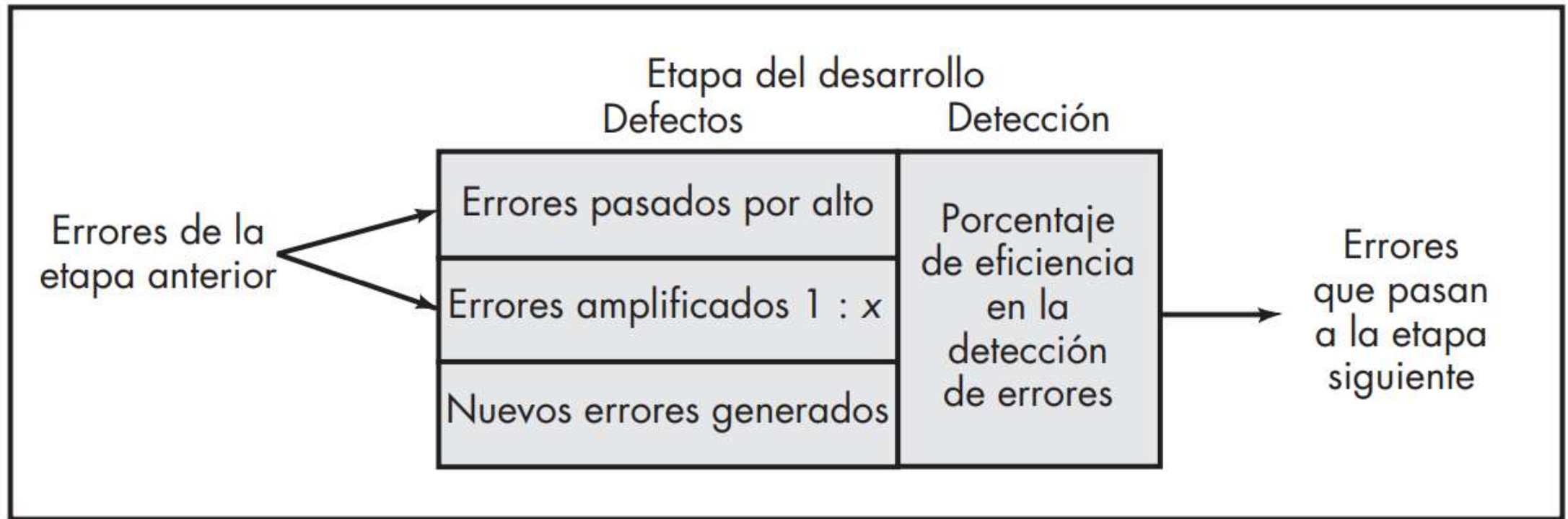
Confirmar partes  
que no requieren  
mejora

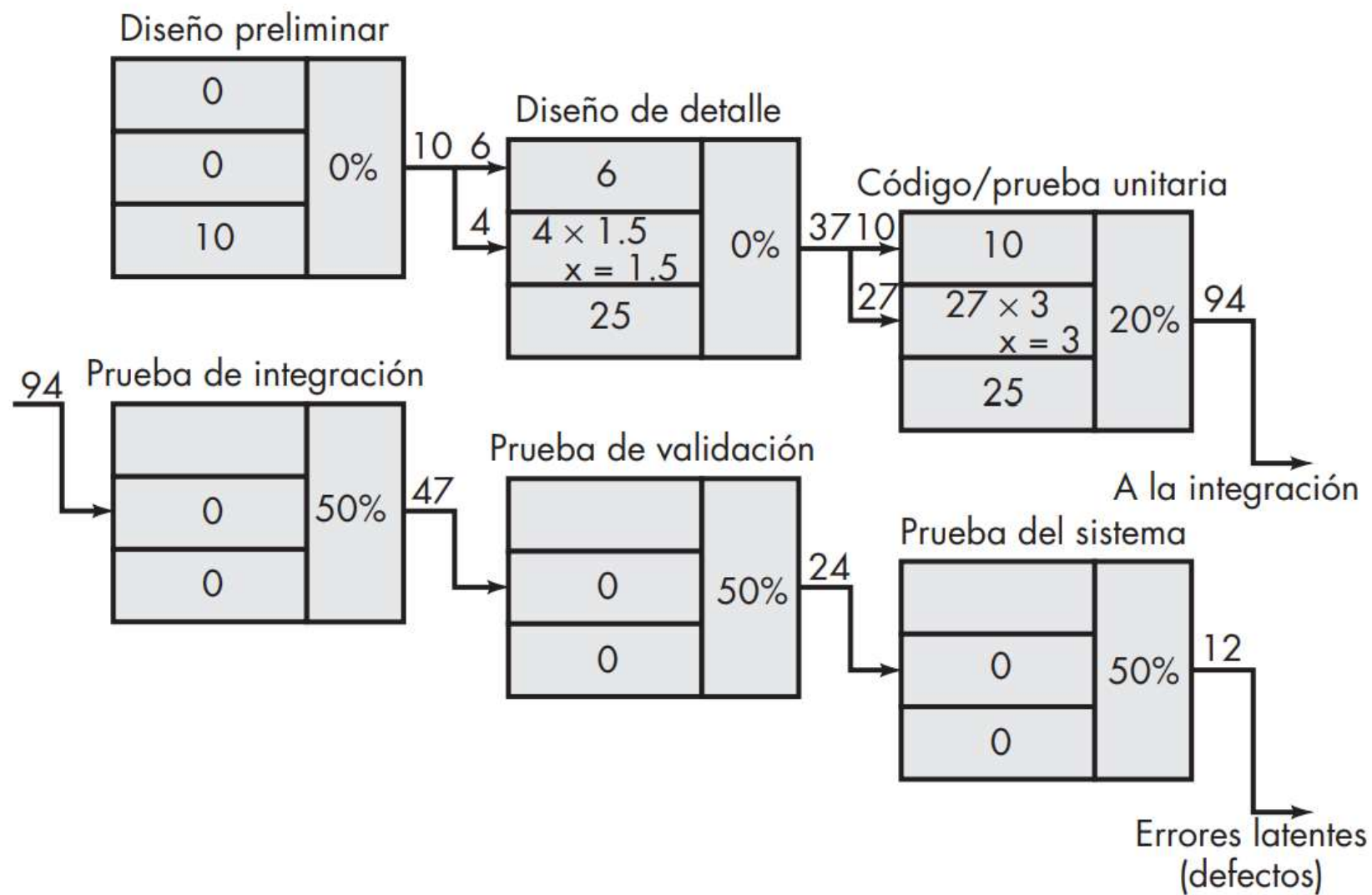
Trabajo técnico

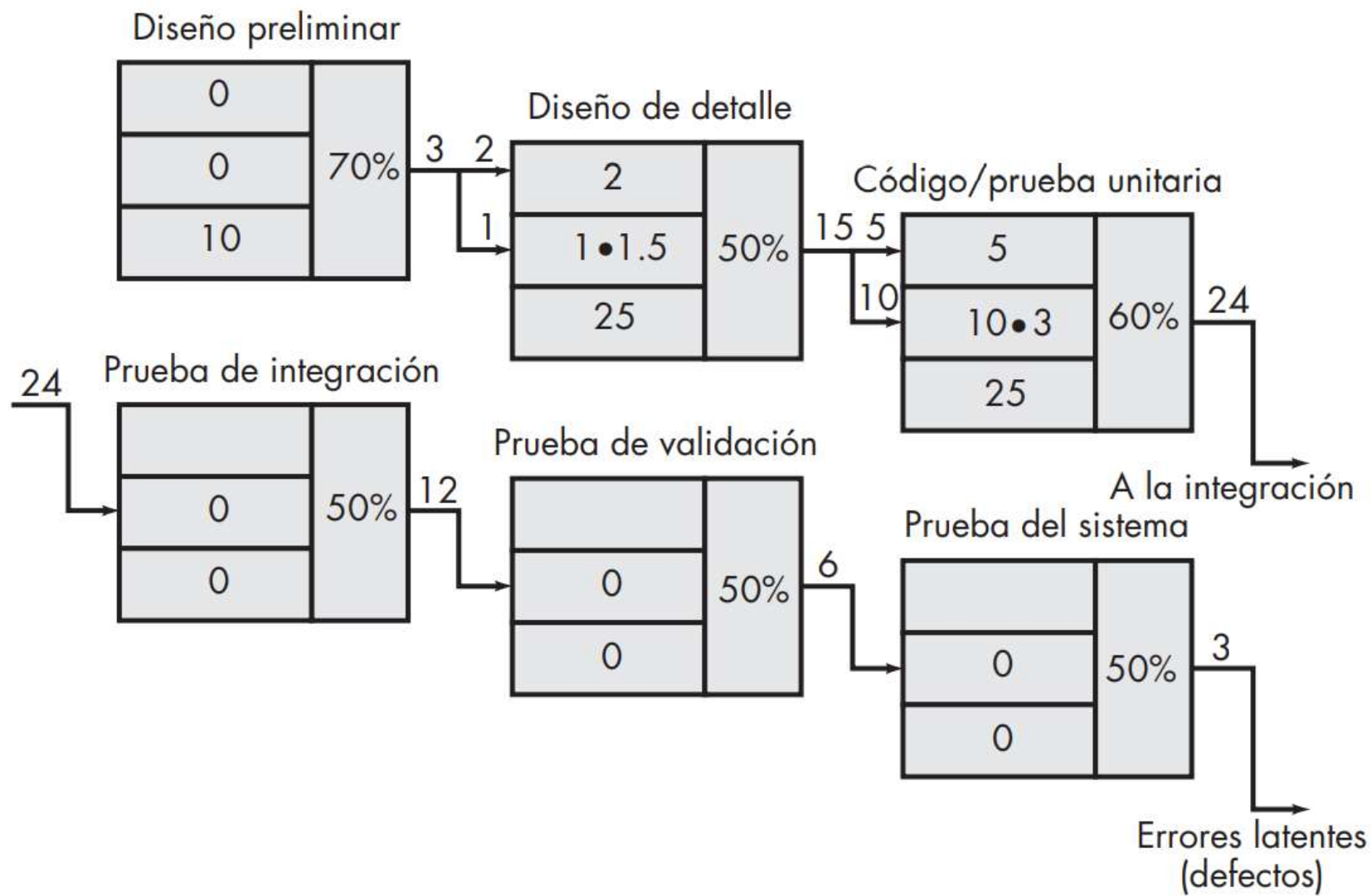
# Revisión técnica

- Descubrimiento temprano de bugs de modo que no se propagan en las siguientes etapas
- Diseño: 50% - 65%
- Revisión: 75% eficaz

# Modelo amplificación del defecto



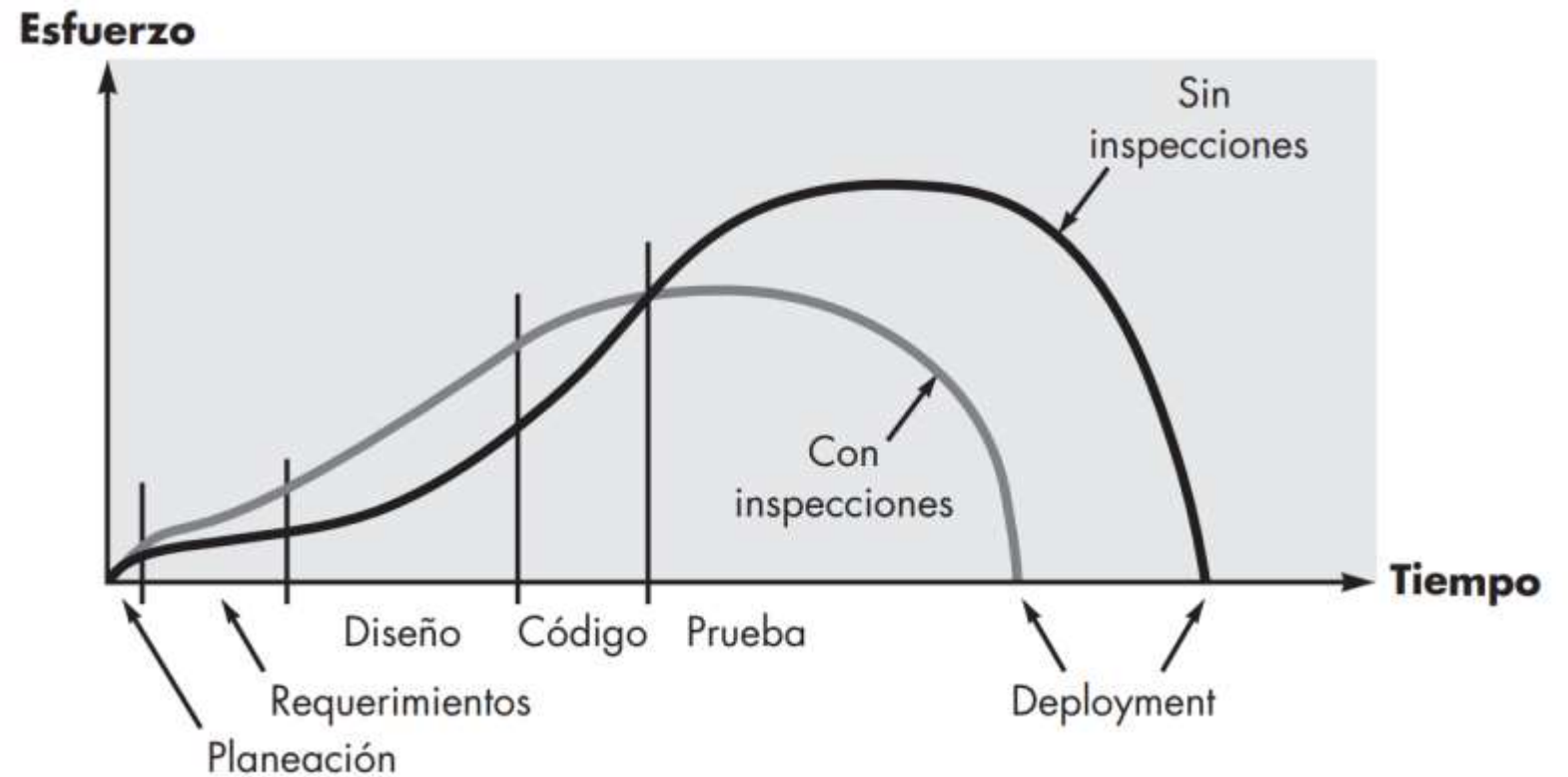






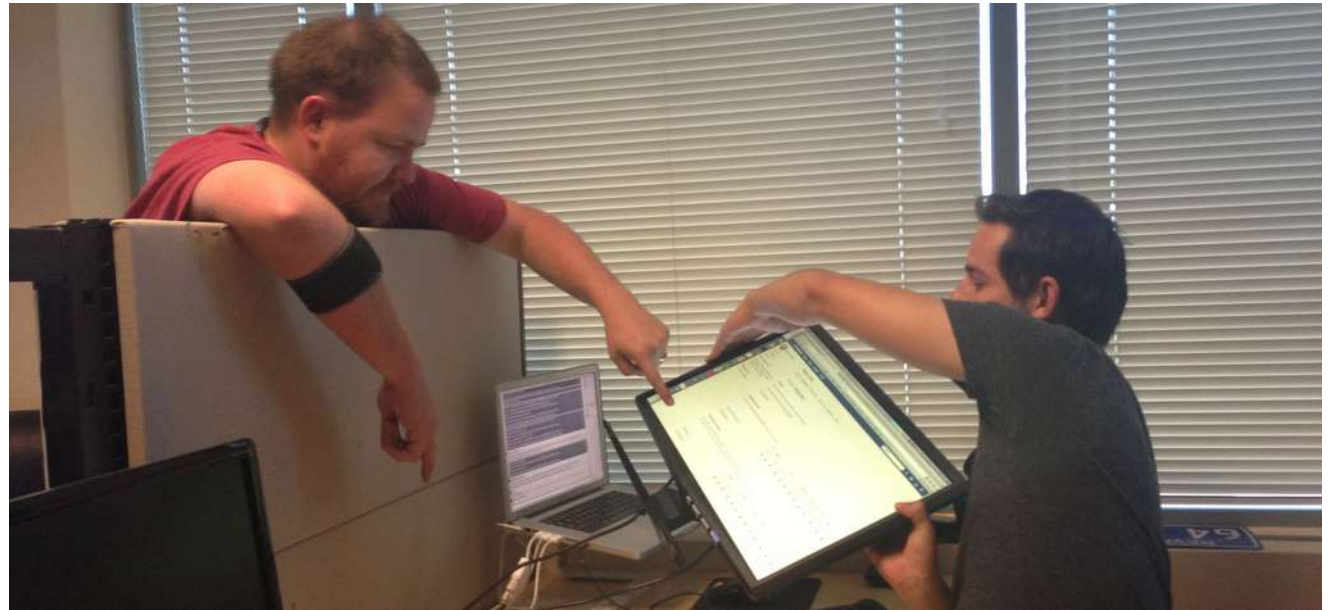
# Nivel de formalidad

- Producto
- Plazo
- Personal



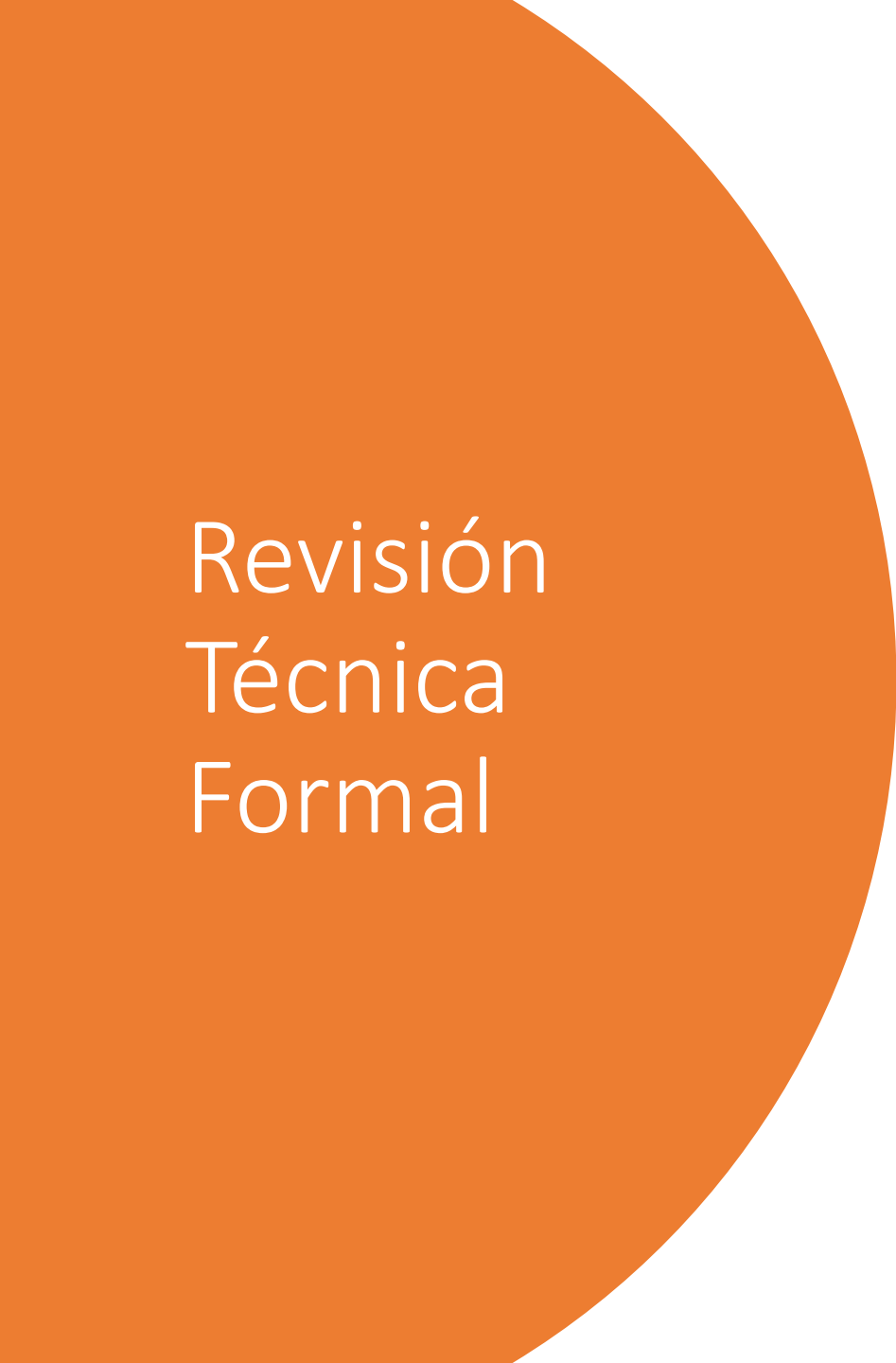
# Revisión técnica informal

- Verificación de escritorio / Reunión Casual
- Listas de verificación\*
- Material pequeño



# Pair programming

- [Agile in Practice: Pair Programming - YouTube](#)

A large orange circle is positioned on the left side of the slide, partially cut off by the edge.

# Revisión Técnica Formal

---

Descubrir errores en funcionamiento, lógica o implementación

---

Software cumple con requerimientos

---

Software cumple con estándares

---

Software desarrollado de manera uniforme

---

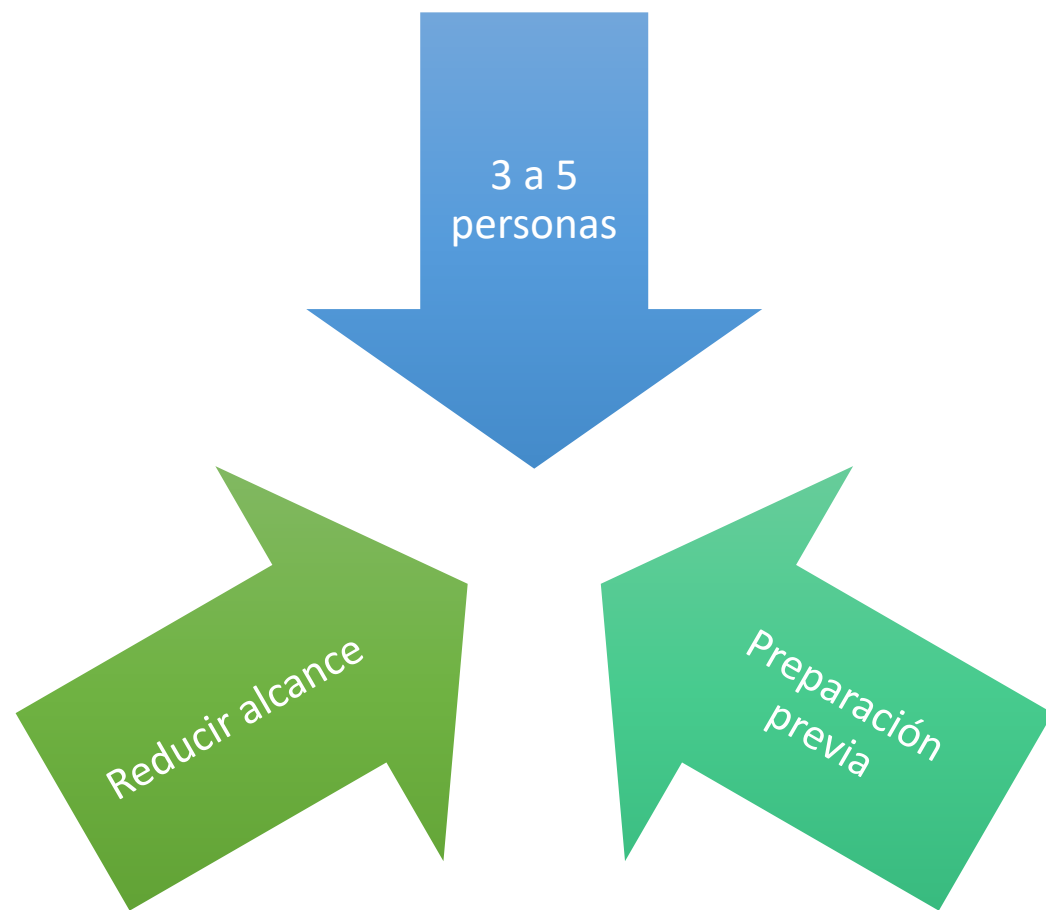
Proyectos mas manejables

---

Capacitación\*

---

# Guia



# Reporte técnico formal de la revisión



¿QUÉ FUE LO QUE SE  
REVISÓ?



¿QUIÉN LO REVISÓ?



¿DESCUBRIMIENTOS Y  
CONCLUSIONES?

# Lista de pendientes de la revisión

Identificar áreas de problemas en el producto

Guia para quien realice los correcciones



# Lineamientos

- Revise el producto, no el productor
- Establezca una agenda y sígala
- Limitar el debate y las contestaciones
- Enunciar áreas de problema, pero no intentar arreglar todos
- Notas por escrito
- Limitar numero de participantes
- Preparación previa
- Lista de verificación
- Asignar recursos y programar tiempo
- Capacitación a supervisores
- Revisar primeras versiones

# Al terminar

---

Aceptar

---

Rechazar con errores graves:  
Requiere de otra revisión formal

---

Rechazar con errores menores: No  
requiere de otra revisión formal

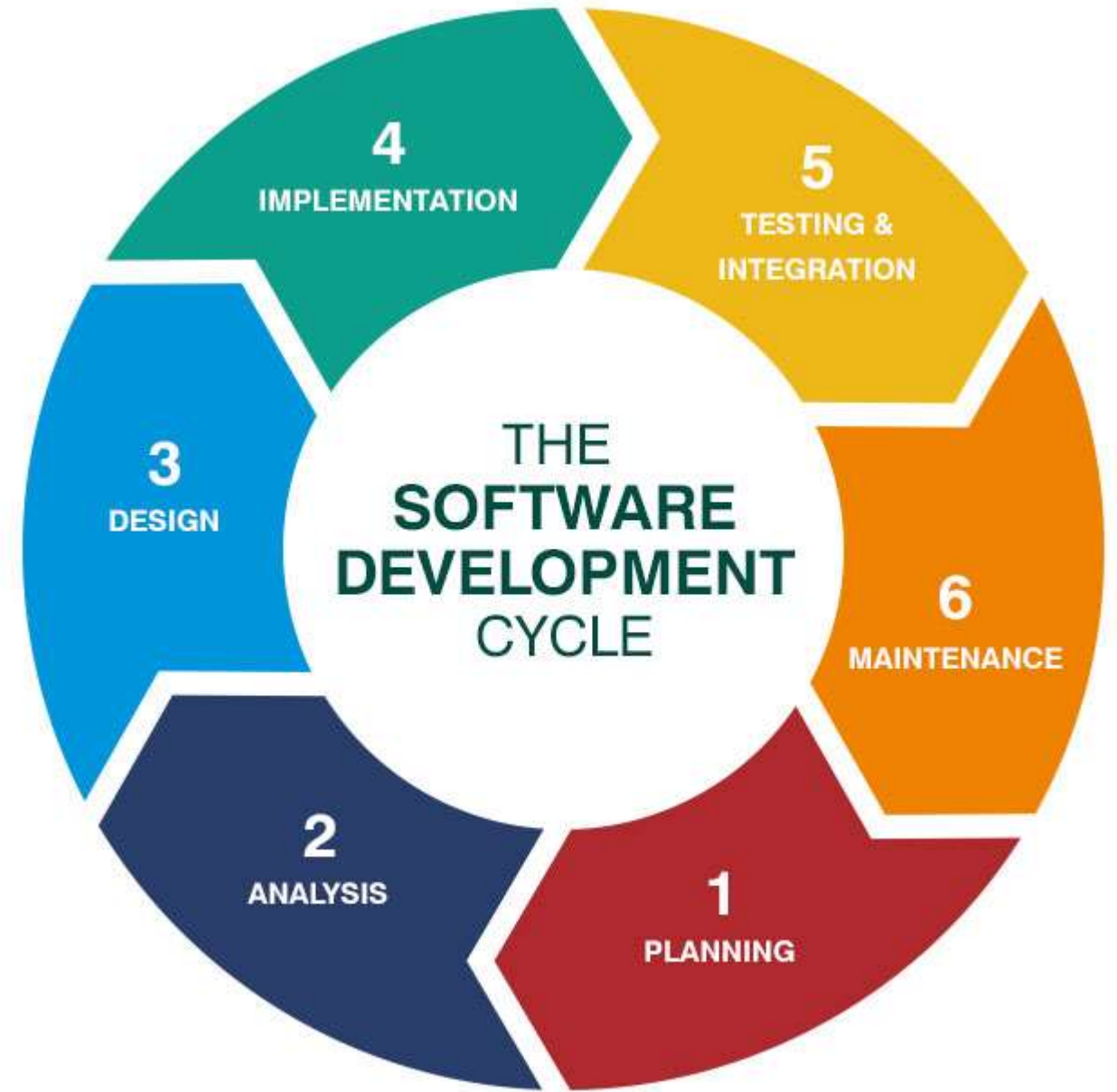


# Revisión orientado al muestreo

- Determinar elementos susceptibles a errores



# Recap ciclo de vida



# Roles en las pruebas de software

QA Engineer

Test Manager

Test Engineer

Test Analyst

Test  
Automation  
Engineer

Security Tester

Data Science  
Tester

Software  
Development  
Engineer in Test

# QA Engineer

Producto y proceso

Monitorea cada fase

Asegura que el producto cumple con estándares

# Test Manager

Equivalente al Project  
manager

Posición gerencial  
dentro del equipo de  
QA



# Test Engineer



MANUAL  
TESTING



EXPLORATORY  
TESTING



PERFORMANCE  
TESTING

# Test Analyst



Enfoca en problemas de negocio



Functional readiness



Diseña, Desarrolla, Ejecuta, y solución problemas en las pruebas

# Test Automation Engineer

Procesos automáticos de prueba

Selenium

Cucumber

Puppeteer

UiPath

# Security Tester



Proteger datos



Asegurar estándares

# Data Science Tester



Datos



Analizar



Agrupar



Limpiar

# Software Development Engineer in Test (SDET)



Desarrollador



Hace pruebas en todo el  
proceso



Pruebas antes de liberar el  
código