



Estrategias de prueba

Universidad Autónoma de Coahuila

Facultad de Sistemas

Calidad y Pruebas de Software

Carlos Nassif Trejo García

Conlleva



Planificación



Diseño



Ejecución



Resultados

Prueba



PEQUEÑO A GRANDE



REVISIONES TÉCNICAS



DESARROLLADORES O
GRUPO
INDEPENDIENTE

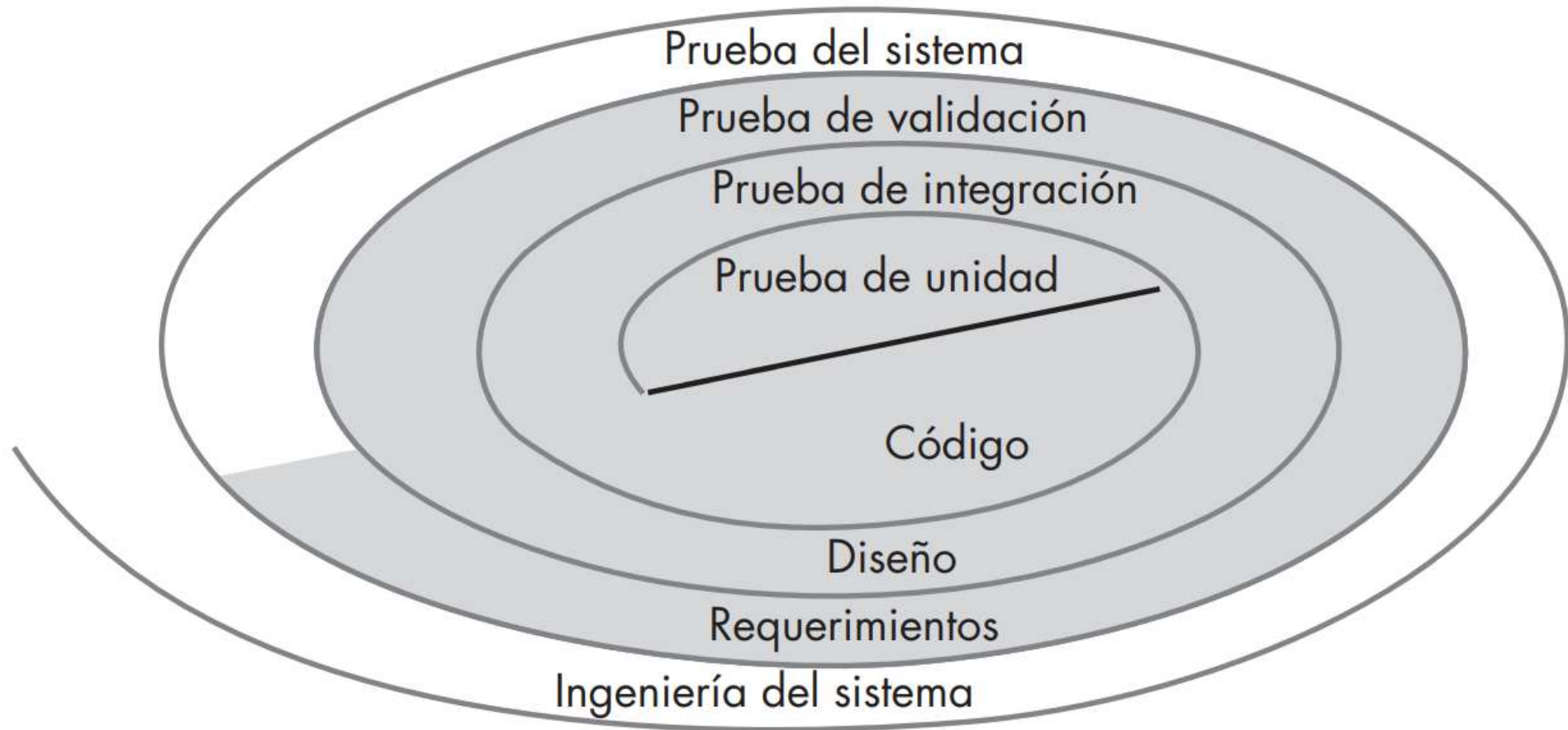


PRUEBA Y
DEPURACIÓN

Psicología



Estrategia



Aspectos a tomar en cuenta

Requerimientos cuantificables

Objetivos de las pruebas

Entender a los usuarios

Software robusto

Revisiones técnicas

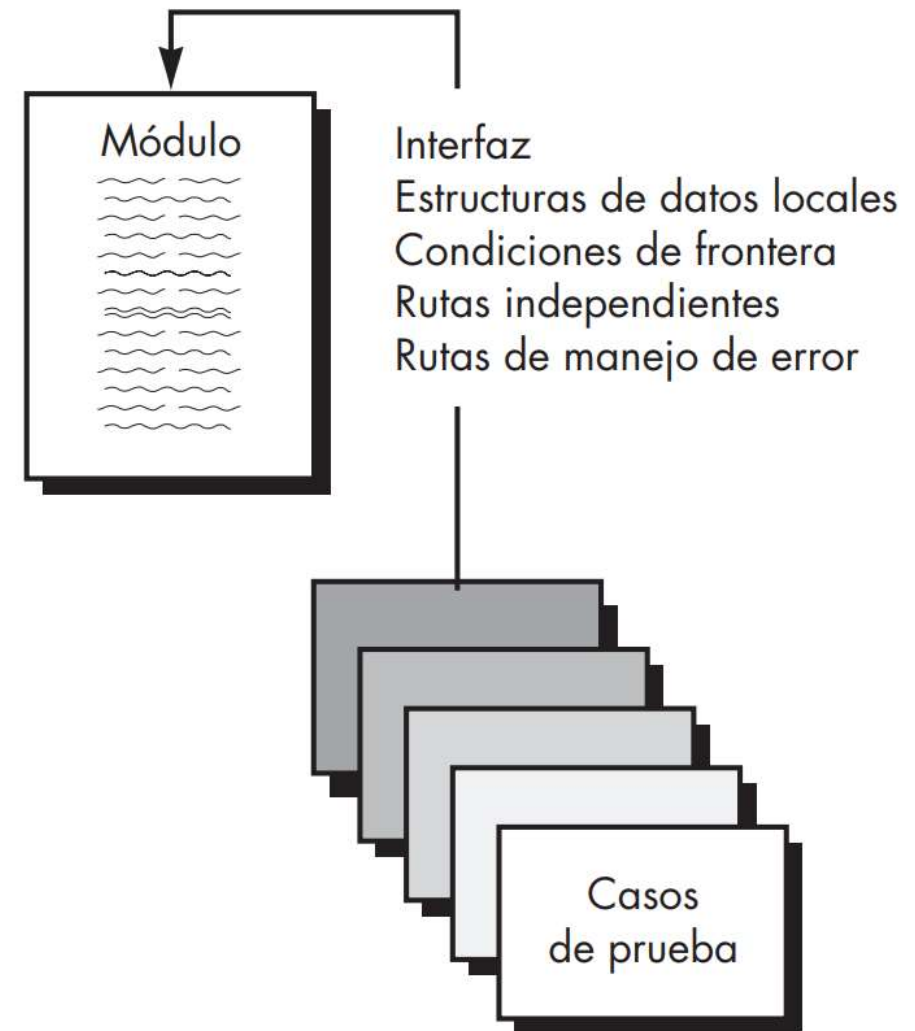
Mejora continua

¿Qué tanto
probar?



Pruebas de unidad

- Unidad mas pequeña
- Lógica interna



Manejar errores

Descripción de error

El sistema debe manejar el error

Procesamiento
excepción-condición

Inexistente

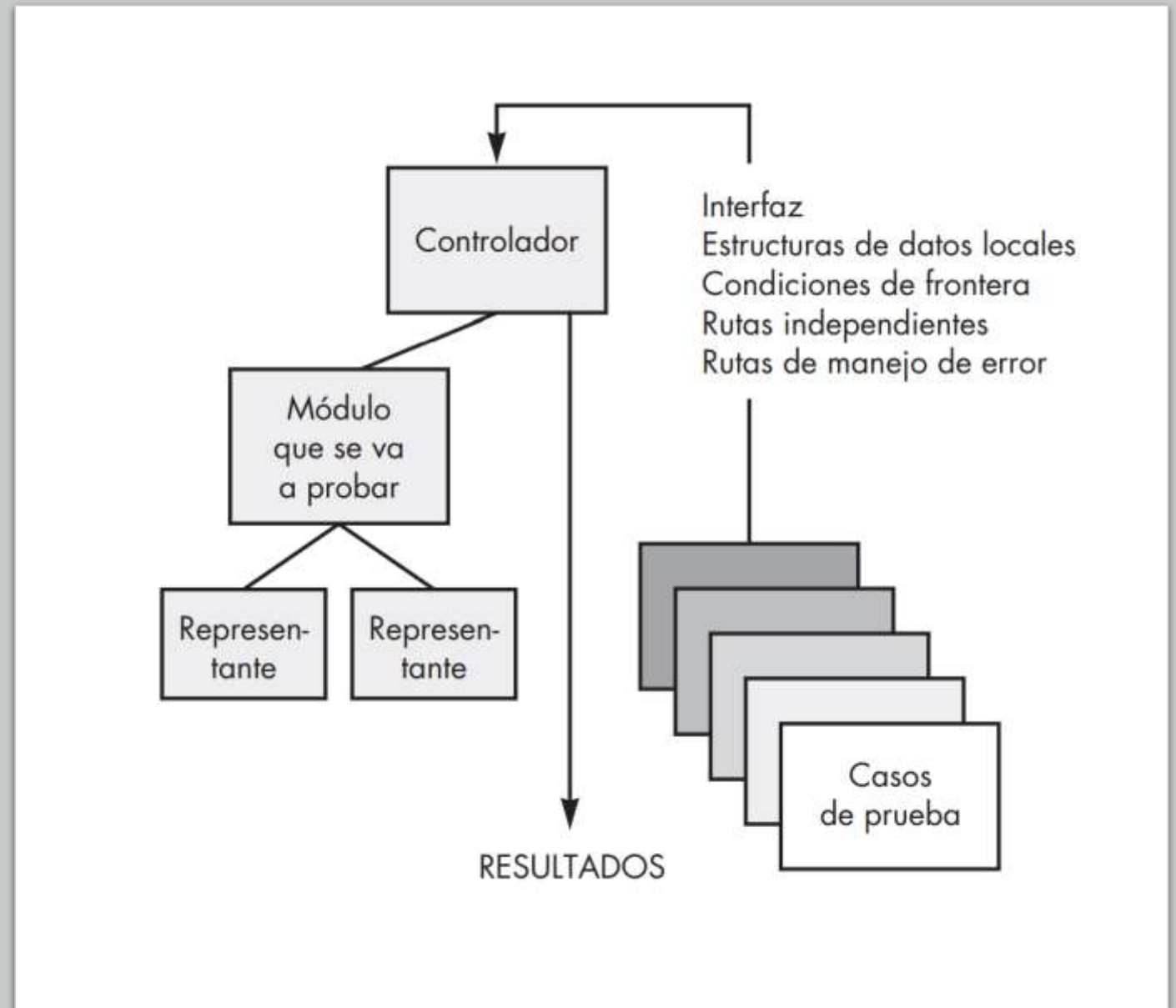
Erróneo

Causa

Auxiliar

Procedimientos

- Pruebas de unidad junto con el código
- Resultados esperados



Estático

Código no se ejecuta

Problemas potenciales

Inspección: Revisión grupal paso a paso donde cada uno es revisado a través de un criterio definido

Walkthrough: El autor explica al equipo la ejecución de su código usando escenario definidos

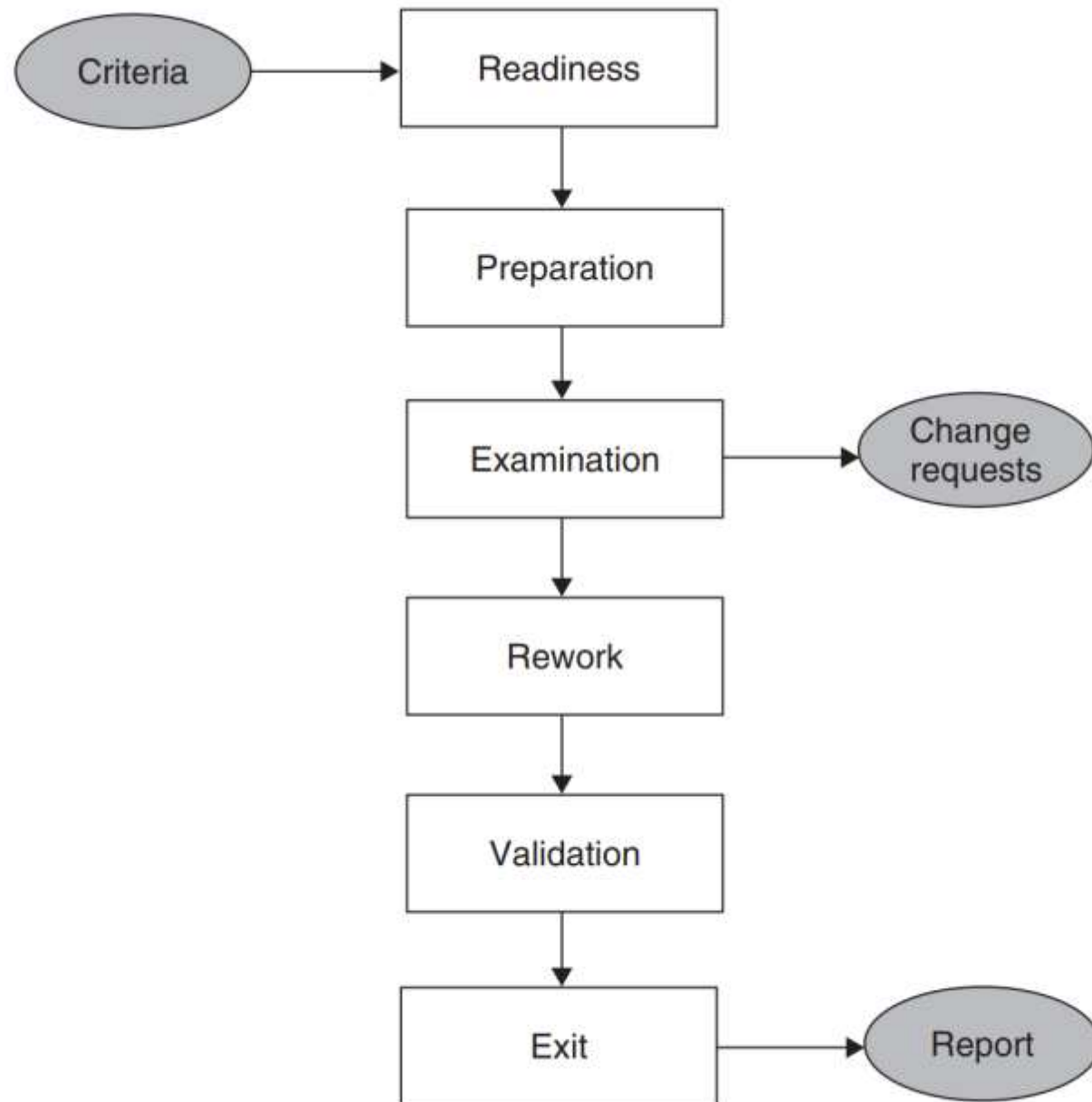


Figure 3.1 Steps in the code review process.

Listo para ser revisado

- Completeness
- Minimal Functionality
- Readability
- Complexity
- Requirements and Design Documents
- 125 lines of code per hour

TABLE 3.1 Hierarchy of System Documents

Requirement: High-level marketing or product proposal.

Functional specification: Software engineering response to the marketing p

High-level design: Overall system architecture.

Low-level design: Detailed specification of the modules within the architect

Programming: Coding of the modules.

Roles

Moderador

Autor

Presentador

Registrador

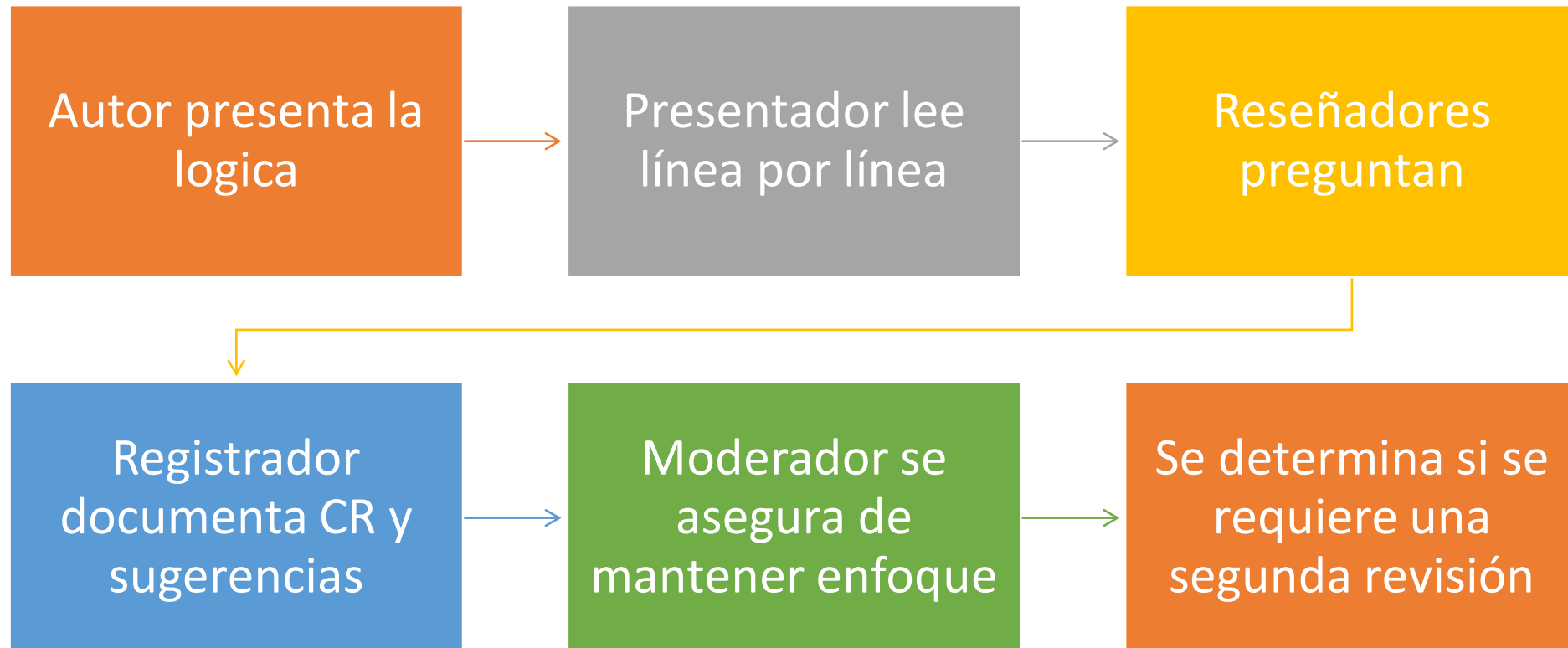
Reseñadores

Observadores

Preparación

- Reseñador:
 - Preguntas
 - CR
 - Oportunidades de mejora

Examinacion



Retrabajo



Validación



Salida

Cada línea fue revisada

Consenso

Cambios revisados

Reporte enviado a todos

Métricas

- LOC: Lines of Code per hour
- KLOC: CR generados por 1000 lineas de código
- CR per hour
- CR per Project
- Num. of hours per Project

Prevención de defectos

Instrumentation code

Controles estándares de errores

Codigo termina para todos los valores de entrada

Overflow y underflow

Mensajes de error

Validar entradas

Assertions

Comparar entrada y salidas

Loops infinitos

Prueba dinámica

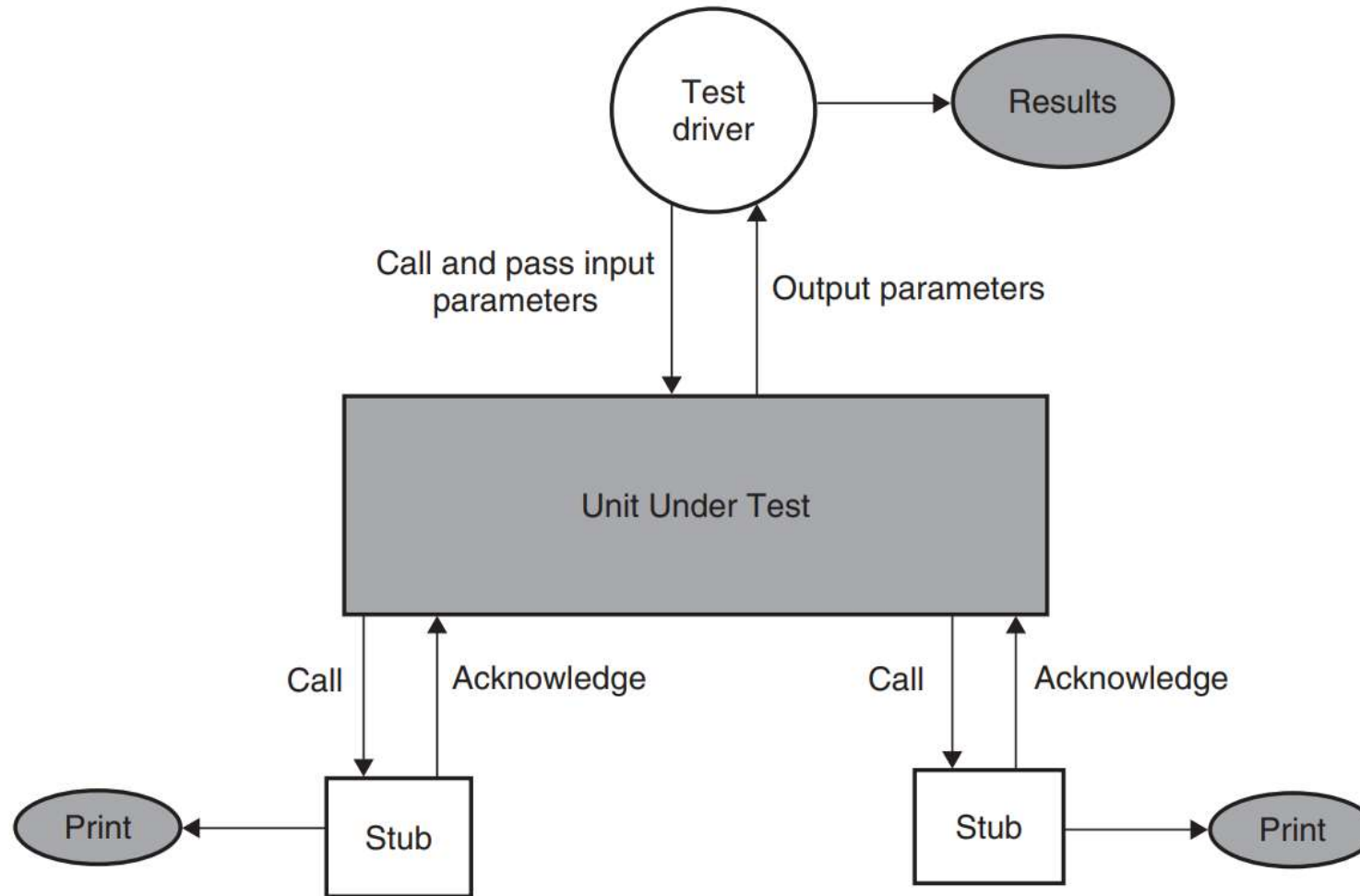


Figure 3.2 Dynamic unit test environment.

Ejemplo...

- A4 Unit Testing
- A5 TDD

Mutation Testing

TDD: Test Driven Development

Test Data

- To be continued