

1) Software Faults and Failures

(1) أخطاء البرنامج

Why Does Software Fail?

لماذا يفشل البرنامج؟

Wrong requirement: not what the customer wants

متطلبات خاطئة: ليس ما يريده العميل

Missing requirement

متطلب مفقود

Requirement impossible to implement

متطلب مستحيل تنفيذه

Faulty design

خطأ بالتصميم

Faulty code

خطأ بكتابة البرنامج

Improperly implemented design

تصميم غير صحيح

Objective of Testing

الهدف من الاختبار

Discover faults

اكتشاف الأخطاء

A test is successful only when a fault is discovered

الاختبار يكون ناجح فقط عند اكتشاف خطأ

Fault identification : the process of determining what fault caused the failure

تحديد الأعطال : عملية تحديد الخطأ الذي تسبب في الفشل

Fault correction : the process of making changes to the system so that the faults are removed

تصحيح الخطأ: عملية إجراء تغييرات على النظام بحيث تتم إزالة الأخطاء

Types of Faults

أنواع الأخطاء

Algorithmic fault

خطأ حسابي

Computation and precision fault

خطأ بدقة الحساب

a formula's implementation is wrong

تطبيق الصيغة خاطئ

Documentation fault

خطأ في التوثيق

Documentation does not match what program does

الوثائق لا تتطابق مع ما يفعله البرنامج

Capacity or boundary faults

أخطاء القدرة أو الحدود

System's performance not acceptable when certain limits are reached

أداء النظام غير مقبول عند الوصول إلى حدود معينة

Timing or coordination faults

أخطاء التوقيت أو التنسيق

Performance faults

أخطاء الأداء

System does not perform at the speed prescribed

لا يعمل النظام في السرعة المقررة

Standard and procedure faults

أخطاء معيارية وإجرائية

Typical Algorithmic Faults

An algorithmic fault occurs when a component's algorithm or logic does not produce proper output

- Branching too soon
- Branching too late
- Testing for the wrong condition
- Forgetting to initialize variable or set loop invariants
- Forgetting to test for a particular condition
- Comparing variables of inappropriate data types

Syntax faults

Orthogonal Defect Classification

Fault Type - Meaning

Function

Fault that affects capability, end-user interface, product interface with hardware architecture, or global data structure

Interface

Fault in interacting with other component or drivers via calls, macros, control, blocks or parameter lists

Checking

Fault in program logic that fails to validate data and values properly before they are used

Assignment

Fault in data structure or code block initialization

Timing/serialization

Fault in timing of shared and real-time resources

Build/package/merge

Fault that occurs because of problems in repositories management changes, or version control

Documentation

Fault that affects publications and maintenance notes

Algorithm

Fault involving efficiency or correctness of algorithm or data structure but not design

أخطاء حسابية نموذجية

يحدث خطأ حسابي عندما لا ينتج عن الخوارزمية أو المنطق الخاص بالمكون مخرجًا مناسبًا

المتفرعة في وقت مبكر جدا

متفرع بعد فوات الأوان

اختبار لحالة خاطئة

نسيان تهيئة متغيرات حلقة أو متغير

نسيان الاختبار لحالة معينة

مقارنة المتغيرات لأنواع البيانات غير المناسبة

أخطاء بكتابة الكود

تصنيف الخطأ ؟

نوع الخطأ - معناه

الخطأ الذي يؤثر على القدرة، واجهة المستخدم النهائي، واجهة المنتج مع بنية الأجهزة أو بنية البيانات العامة

عطل في التفاعل مع المكونات أو برامج التشغيل الأخرى عبر المكالمات أو (macros, control, blocks or parameter lists)

خطأ في منطق البرنامج الذي فشل في التحقق من صحة البيانات وقيمها بشكل صحيح قبل استخدامها

خطأ في بنية البيانات أو بالكود البرمجي

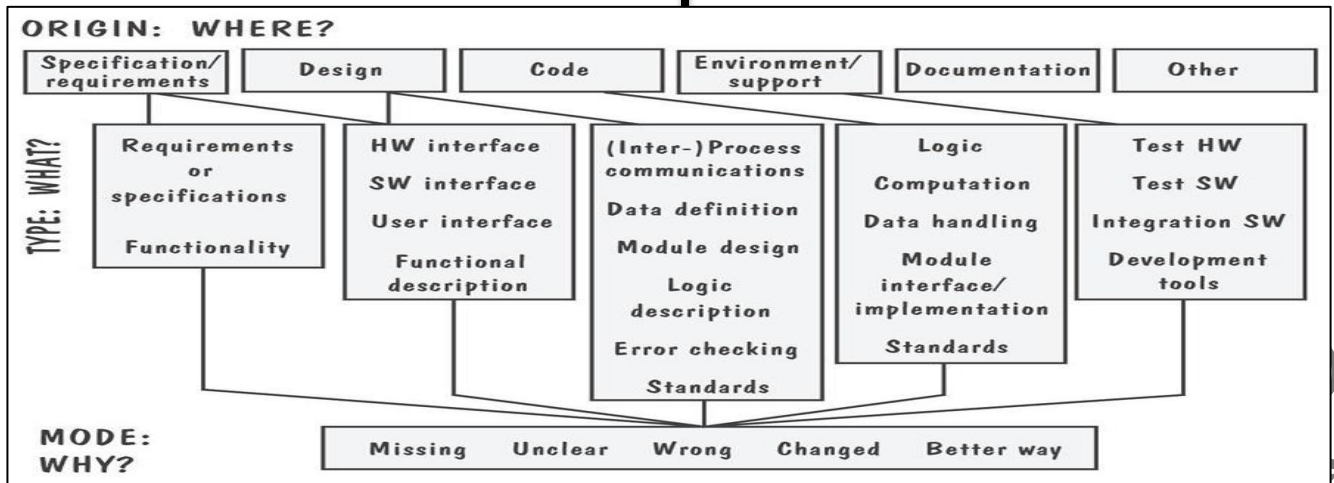
خطأ في توقيت الموارد المشتركة في الوقت الفعلي

الخطأ الذي يحدث بسبب وجود مشاكل في تغييرات إدارة التقارير، أو التحكم في الإصدار

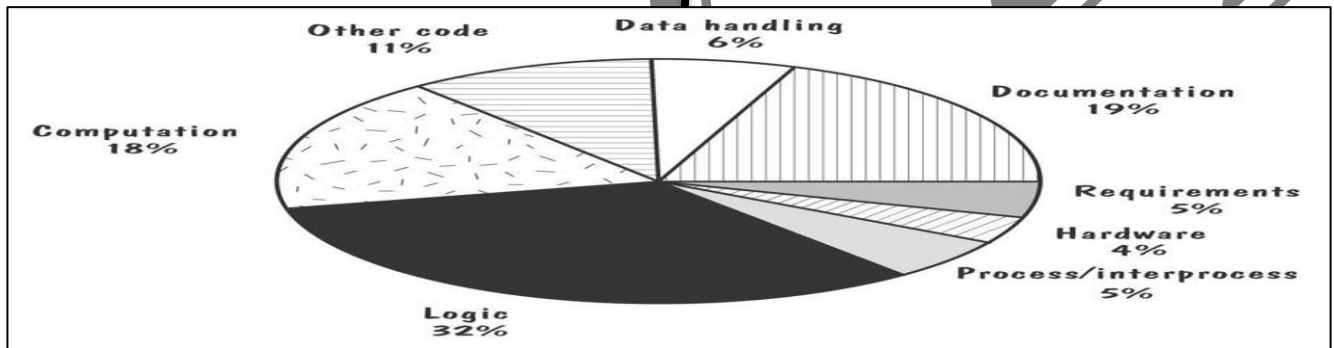
خطأ يؤثر على المنشورات وملاحظات الصيانة

خطأ² يتعلق بكفاءة أو صحة الخوارزمية أو بنية البيانات ولكن ليس التصميم

Hewlett-Packard's Fault Classification



Faults for one Hewlett-Packard Division

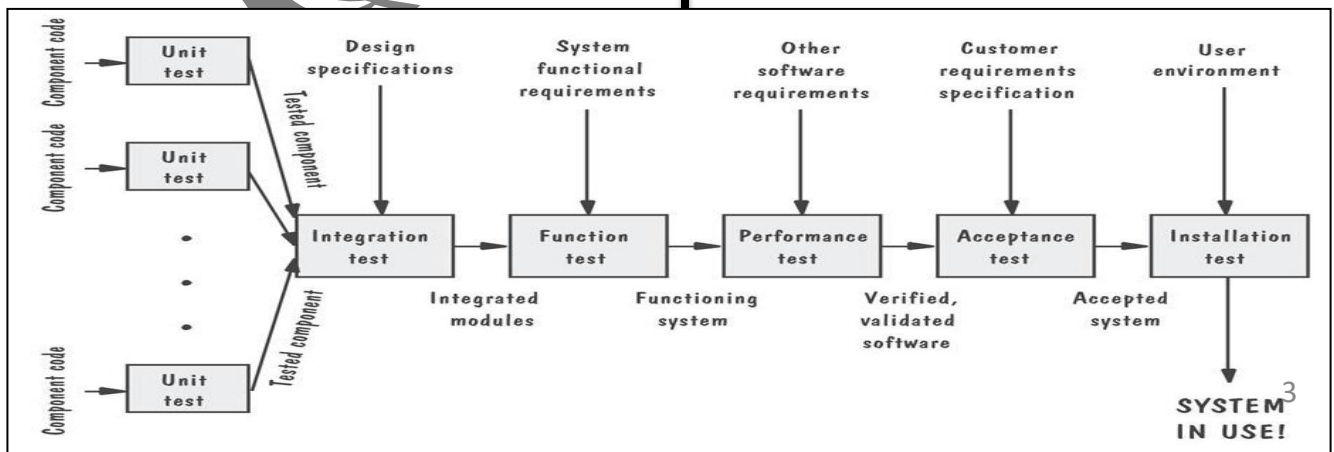


2) Testing Issues

Testing Organization

(2) قضايا الاختبار
اختبار التنظيم

Module testing, component testing, or unit testing
Integration testing
Function testing
Performance testing
Acceptance testing
Installation testing



Attitude Toward Testing

Egoless programming: programs are viewed as components of a larger system, not as the property of those who wrote them

Who Performs the Test?

Independent test team

avoid conflict

improve objectivity

allow testing and coding concurrently

Views of the Test Objects

Closed box or black box: functionality of the test objects

Clear box or white box: structure of the test objects

Box

Advantage

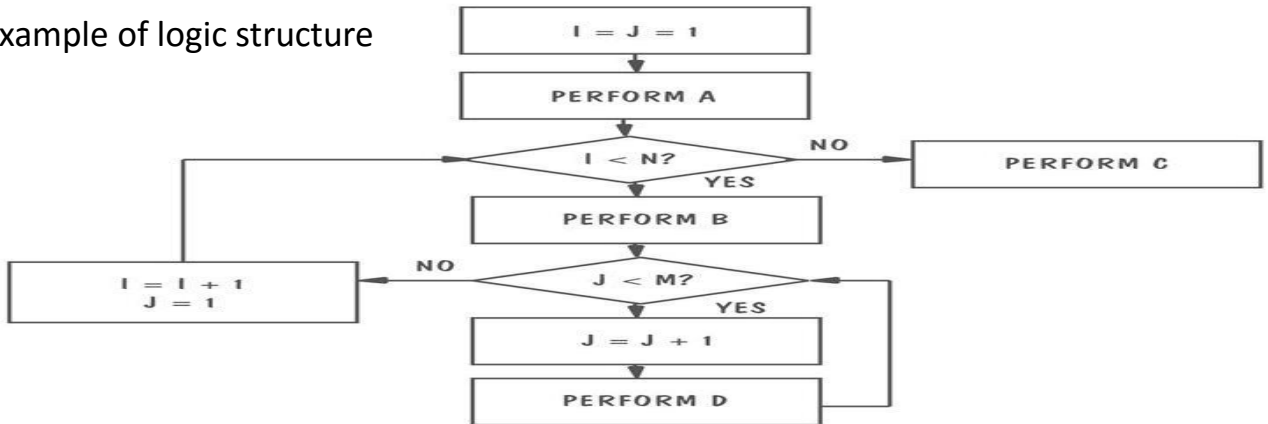
free of internal structure's constraints

Disadvantage

not possible to run a complete test

Clear Box

Example of logic structure



Box Structures

Black box: external behavior description

State box: black box with state information

White box: state box with a procedure

3) Unit Testing

Code Review

Code walkthrough

Code inspection

الموقف نحو الاختبار

برمجة (Egoless) : يتم النظر إلى البرامج كمكونات لنظام أكبر ، وليس كممتلكات لأولئك الذين قاموا بكتابتها

من الذي يقوم بإجراء الاختبار؟

فريق اختبار مستقل

تجنب الخل

تحسين الموضوعية

السماح للاختبار كتابة الكود في وقت واحد

وجهات النظر من مكونات اختبار

إيجابيات

خالية من قيود الهيكل الداخلي

سلبيات

ليس من الممكن تشغيل اختبار كامل

هياكل الصناديق

مراجعة التعليمات البرمجية

مراجعة الكود

فحص الكود

Typical Inspection Preparation and Meeting Times

إعداد التفتيش النموذجي وأوقات الاجتماع

Development Artifact	Preparation Time	Meeting Time
Requirement Document	25 pages per hour	12 pages per hour
Functional specification	45 pages per hour	15 pager per hour
Logic specification	50 pages per hour	20 pages per hour
Source code	150 lines of code per hour	75 lines of code per hour
User documents	35 pages per hour	20 pages per hour

Fault Discovery Rate

معدل اكتشاف الخطأ

Discovery Activity	Fault Found per Thousand Lines of Code
Requirements review	2.5
Design Review	5.0
Code inspection	10.0
Integration test	3.0
Acceptance test	2.0

Steps in Choosing Test Cases

خطوات في اختيار حالات الاختبار

- Determining test objectives
- Selecting test cases
- Defining a test

- تحديد أهداف الاختبار
- اختيار حالات الاختبار
- تحديد الاختبار

Comparing Techniques

مقارنة التقنيات

Fault discovery Percentages by Fault Origin

نسبة اكتشاف الخطأ حسب أصل الخلل

Discovery Techniques	Requirements	Design	Coding	Documentation
Prototyping	40	35	35	15
Requirements review	40	15	0	5
Design Review	15	55	0	15
Code inspection	20	40	65	25
Unit testing	1	5	20	0

Effectiveness of fault-discovery techniques

فعالية تقنيات اكتشاف الخطأ

	Requirements Faults	Design Faults	Code Faults	Documentation Faults
Reviews	Fair	Excellent	Excellent	Good
Prototypes	Good	Fair	Fair	Not applicable
Testing	Poor	Poor	Good	Fair 5
Correctness Proofs	Poor	Poor	Fair	Fair

4) Integration Testing

Bottom-up

Top-down

Big-bang

Sandwich testing

Modified top-down

Modified sandwich

Terminology

Component Driver : a routine that calls a particular component and passes a test case to it

Stub: a special-purpose program to simulate the activity of the missing component

View of a System

System viewed as a hierarchy of components

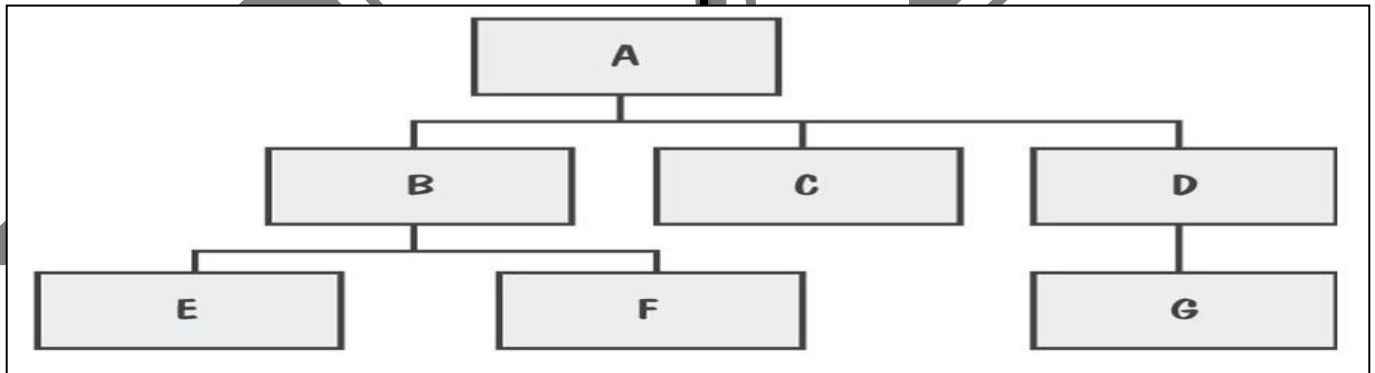
اصطلاحات

برنامج تشغيل المكون : روتين يستدعي مكونًا معينًا ويمرر حالة اختبار عليه

(Stub) برنامج خاص الغرض لمحاكاة عمل المكون المفقود

عرض النظام

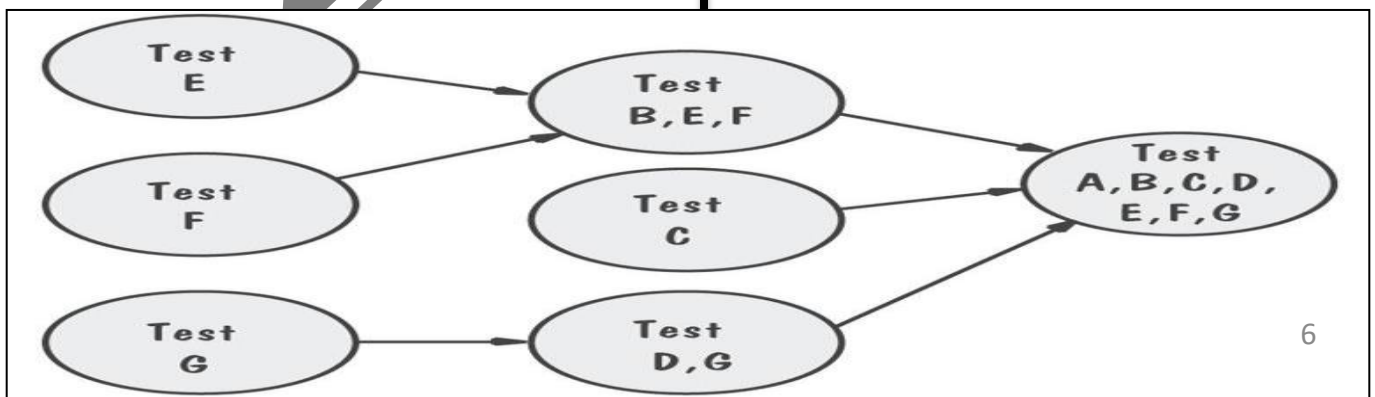
عرض هيكل النظام



Bottom-Up Integration Example

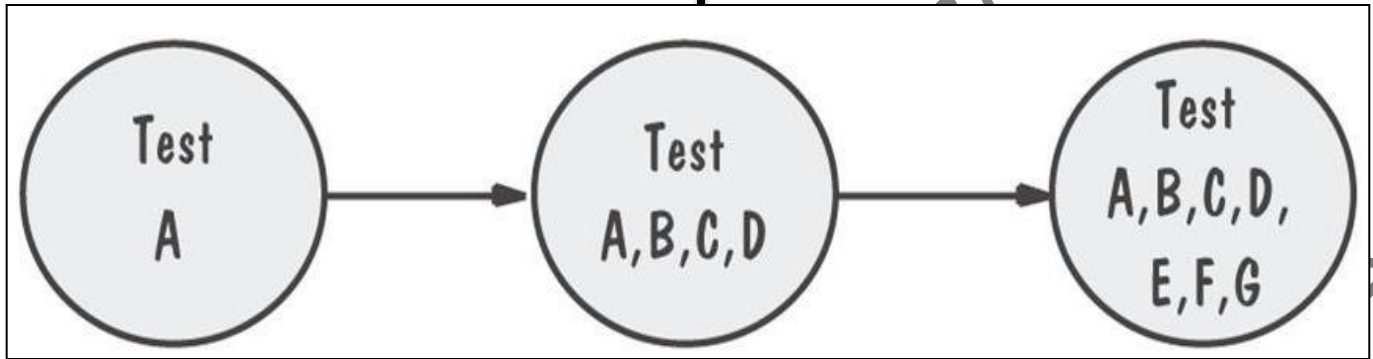
The sequence of tests and their dependencies

تسلسل الاختبارات واعتمادياتها



Top-Down Integration Example

Only A is tested by itself



5) Test Planning

Establish test objectives

Design test cases

Write test cases

Test test cases

Execute tests

Evaluate test results

وضع أهداف الاختبار

تصميم حالات اختبار

اكتب حالات الاختبار

اختبار حالات الاختبار

تنفيذ الاختبار

تقييم نتائج الاختبار