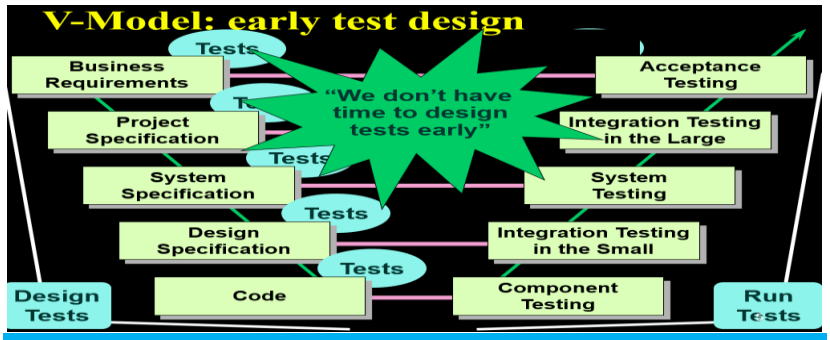
# Ch 6




## Early test design

- test design finds faults
- faults found early are cheaper to fix
- most significant faults found first
- faults prevented, not built in

Early test design helps to build quality, stops fault multiplication

## Verification

checks that the work-product meets the requirements set out for it.

## Validation

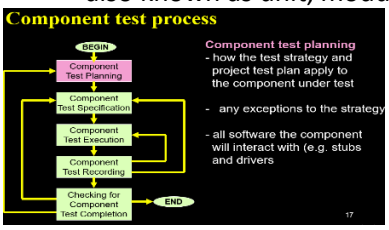
changes the focus of work-product evaluation to evaluation against user needs. This means ensuring that the behaviour of the work-product matches the customer needs as defined for the project

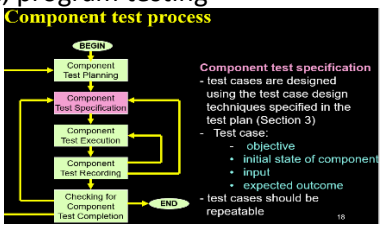
## Testing

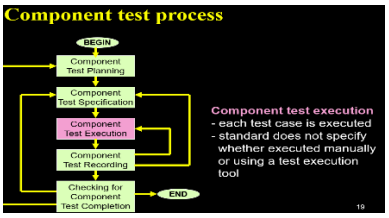
the process of exercising software to verify that it satisfies specified requirements and to detect faults

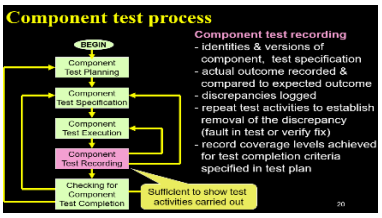
## Component testing

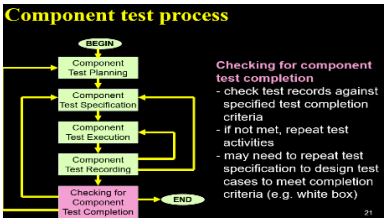
- lowest level
- tested in isolation
- most thorough look at detail
  - error handling
  - interfaces
- usually done by programmer
- also known as unit, module, program testing

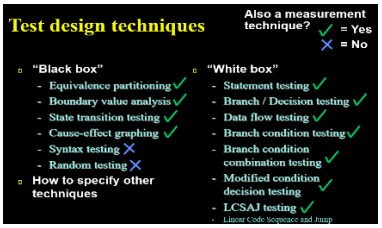



## Integration testing in the small

- more than one (tested) component
- communication between components
- what the set can perform that is not possible individually
- non-functional aspects if possible
- integration strategy:
  - ➢ big-bang vs incremental (top-down, bottom-up, functional)
- done by designers, analysts, or independent testers

## Big-Bang Integration

- In practice:
  - takes longer to locate and fix faults
  - re-testing after fixes more extensive
  - end result? takes more time

## Incremental Integration

- **Advantages:**
  - easier fault location and fix
  - easier recovery from disaster / problems
  - interfaces should have been tested in component testing

## Top-Down Integration

- Need to call to lower level components not yet integrated
- Stubs: simulate missing components

## Bottom-up Integration

- Needs drivers to call the baseline configuration
- Also needs stubs for some baselines

## Stubs

- Stub replaces a called component for integration testing
- Keep it Simple

## Drivers

- Driver: test harness: scaffolding
- specially written or general purpose (commercial tools)
  - invoke baseline
  - send any data baseline expects
  - receive any data baseline produces(print)

## Pros & cons of top-down approach

- **Advantages:**
  - critical control structure tested first and most often
  - can demonstrate system early (show working menus)
- **Disadvantages:**
  - needs stubs
  - may be difficult to "see" detailed output (but should have been tested in component test)

# Ch 6

## Pros & cons of bottom-up approach

- **Advantages:**
  - lowest levels tested first and most thoroughly (but should have been tested in unit testing)
  - good for testing interfaces to external environment (hardware, network)
- **Disadvantages**
  - no working system until last baseline
  - needs both drivers and stubs
  - major control problems found last

## Thread Integration (also called functional)

- order of processing some event determines integration order
- interrupt, user transaction
- **advantages:**
  - critical processing first
- **disadvantages:**
  - may need complex drivers and stubs

## System testing

- **functional**
  - functional requirements and requirements-based testing
  - business process-based testing
- **non-functional**
  - as important as functional requirements
  - often poorly specified
  - must be tested
- **often done by independent test group**

## Non-functional system testing

- usability
- security
- documentation
- storage
- volume
- configuration / installation
- reliability / qualities
- back-up / recovery
- performance, load, stress

## Performance Tests

- **Timing Tests**
  - response and service time
  - database back-up times
- **Capacity & Volume Tests**
  - maximum amount or processing rate
  - number of records on the system
  - graceful degradation
- **Endurance Tests (24-hr operation?)**
  - robustness of the system
  - memory allocation

## Multi-User Tests

- **Concurrency Tests**

-small numbers, large benefits - detect record locking problems

- **Stress Tests**
  - go beyond limits for the system - know what will happen
  - particular relevance for e-commerce

## Stress Tests

- The goal of stress testing is to evaluate and determine the behavior of a software component while the offered load is in excess of its designed capacity
- **Stress tests are targeted to bring out the problems associated with one or more of the following**:
  - Memory leak
  - Buffer allocation and memory carving

## Security Tests

- Security tests are designed to verify that the system meets the security requirements
- **Confidentiality**
  - It is the requirement that data and the processes be protected from unauthorized disclosure
- **Integrity**
  - It is the requirement that data and process be protected from unauthorized modification
- **Availability**
  - It is the requirement that data and processes be protected from the denial of service to authorized users

## Configuration and Installation

- **Configuration Tests**
  - different hardware or software environment
  - configuration of the system itself
- **Installation Tests**

- distribution (CD, network, etc.) and timings

-physical aspects: electromagnetic fields, heat, humidity, motion, chemicals, power supplies

- uninstall (removing installation)

## Reliability Tests

- Reliability tests are designed to measure the ability of the system to remain operational for long periods of time.
- The reliability of a system is typically expressed in terms of mean time to failure (MTTF)
- The average of all the time intervals between successive failures is called the MTTF

## Back-up and Recovery

- **Back-ups**
  - computer functions
  - manual procedures (where are tapes stored)
- **Recovery**
  - real test of back-up
  - manual procedures unfamiliar
  - should be regularly rehearsed
  - documentation should be detailed, clear and thorough

## Documentation Testing

- **Documentation review**
  - check for accuracy against other documents
  - gain consensus about content
  - documentation exists, in right format
- **Documentation tests**
  - is it usable? does it work?
  - user manual
  - maintenance documentation

# Ch 6

## User acceptance testing

- **Final stage of validation**
  - customer (user) should perform or be closely involved
  - customer can perform any test they wish, usually based on their business processes
  - final user sign-off

## Contract acceptance testing

- Based on the results, the customer considers whether the software system is free of (major) deficiencies and whether the service defined by the development contract has been accomplished and is acceptable.
- The test criteria are the acceptance criteria determined in the development contract. Therefore, these criteria must be stated as unambiguously as possible.
- Additionally, conformance to any governmental, legal, or safety regulations must be addressed here.

## Alpha and Beta tests: similarities

- Testing by [potential] customers or representatives of your market
  - not suitable for bespoke software
- When software is stable
- Use the product in a realistic way in its operational environment
- Give comments back on the product
  - faults found
  - how the product meets their expectations
  - improvement / enhancement suggestions?

## Alpha and Beta tests: differences

- Alpha tests are carried out at the producer's location, while beta tests are carried out at the customer's site.
- The idea is that "if you make dogfood, try it yourself first."
- Large suppliers of software like Microsoft and Google advocate this approach before beta testing.

## Acceptance testing motto

- If you don't have patience to test the system  the system will surely test your patience

## Maintenance testing

- Testing to preserve quality after software maintenance
- when a product is adapted to new operational conditions (adaptive maintenance, updates of operating systems, databases, middleware) or
- when defects are corrected (corrective maintenance).
- Testing changes made during maintenance can be difficult because the system's specifications are often out of date or missing.

## What to test in maintenance testing ?

- answer: "It depends"