There are two basic types of testing:
- Execution-based testing
- Non-execution-based testing

"V & V"
- Verification: Determine if the workflow was completed correctly
- Validation: Determine if the product as a whole satisfies its requirements

Warning
- The term "verify" is also used for all non-execution-based testing

Fault is injected into software
- When a human makes a mistake
- Failure is the observed incorrect behavior
- Error is the amount by which a result is incorrect
- Generic term is defect

Software Quality
- Not "excellence" or "luxurious"
- The extent to which software satisfies its specifications
  - Analogy: Coca-Cola, cars
  - Every software professional is responsible for ensuring that his or her work is correct
    - Quality must be built in from the beginning

Software Quality Assurance
- The members of the SQA group must ensure that the developers are doing high-quality work
  - At the end of each workflow
  - When the product is complete
- In addition, quality assurance must be applied to the process itself
  - Example: Standards

Managerial Independence
- There must be managerial independence between
  - The development group
  - The SQA group
- Neither group should have power over the other
Managerial Independence (2)

- More senior management must decide whether to
  - Deliver the product on time but with faults, or
  - Test further and deliver the product late
- The decision must take into account the interests of the client and the development organization

Non-Execution-Based Testing

- Underlying principles
  - We should not review our own work
  - Group synergy
- Two types of reviews
  - Walkthrough
  - Inspection

Walkthroughs

- A walkthrough team consists of from four to six members
- It includes representatives of
  - The team responsible for the current workflow
  - The team responsible for the next workflow
  - The SQA group
- The walkthrough is preceded by preparation
  - Lists of items
    - Items not understood
    - Items that appear to be incorrect

Managing Walkthroughs

- The walkthrough team is chaired by the SQA representative
- In a walkthrough we detect faults, not correct them
  - A correction produced by a committee is likely to be of low quality
  - The cost of a committee correction is too high
  - Not all items flagged are actually incorrect
  - A walkthrough should not last longer than 2 hours, so there is no time to correct faults as well

Managing Walkthroughs (2)

- A walkthrough must be document-driven, rather than participant-driven
- Verbalization leads to fault finding
- A walkthrough should never be used for performance appraisal

Inspections

- An inspection has five formal steps
  - Overview
  - Preparation, aided by statistics of fault types
  - Inspection
  - Rework
  - Follow-up
Inspections (2)
- An inspection team has four members
  - Moderator
  - A member of the team performing the current workflow
  - A member of the team performing the next workflow
  - A member of the SQA group
- Special roles are played by the
  - Moderator
  - Reader
  - Recorder

Fault Statistics
- Faults are recorded by severity
  - Example:
    - Major or minor
- Faults are recorded by fault type
  - Examples of design faults:
    - Not all specification items have been addressed
    - Actual and formal arguments do not correspond

Fault Statistics (2)
- For a given workflow, we compare current fault rates with those of previous products
- We take action if there are a disproportionate number of faults in an artifact
  - Redesigning from scratch is a good alternative
- We carry forward fault statistics to the next workflow
  - We may not detect all faults of a particular type in the current inspection

Statistics on Inspections
- IBM inspections showed up
  - 82% of all detected faults (1976)
  - 70% of all detected faults (1978)
  - 93% of all detected faults (1986)
- Switching system
  - 90% decrease in the cost of detecting faults (1986)
- JPL
  - Four major faults, 14 minor faults per 2 hours (1990)
  - Savings of $25,000 per inspection
  - The number of faults decreased exponentially by phase (1992)

Statistics on Inspections (2)
- Warning: Fault statistics should never be used for performance appraisal
  - "Killing the goose that lays the golden eggs"

Comparison of Inspections and Walkthroughs
- Walkthrough
  - Two-step, informal process
    - Preparation
    - Analysis
- Inspection
  - Five-step, formal process
    - Overview
    - Preparation
    - Inspection
    - Rework
    - Follow-up
Strengths and Weaknesses of Reviews

- Reviews can be effective
  - Faults are detected early in the process
- Reviews are less effective if the process is inadequate
  - Large-scale software should consist of smaller, largely independent pieces
  - The documentation of the previous workflows must be complete and available online

Metrics for Inspections

- Inspection rate (e.g., design pages inspected per hour)
- Fault density (e.g., faults per KLOC inspected)
- Fault detection rate (e.g., faults detected per hour)
- Fault detection efficiency (e.g., number of major, minor faults detected per hour)

Metrics for Inspections (2)

- Does a 50% increase in the fault detection rate mean that
  - Quality has decreased? Or
  - The inspection process is more efficient?

Execution-Based Testing

- Organizations spend up to 50% of their software budget on testing
  - But delivered software is frequently unreliable
- Dijkstra (1972)
  - "Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence"

Seven Principles of Software Testing

- Read on-line article for Wednesday
- Do you agree or disagree with each of the principles?
- Have you used (or wished you used) any of the principles in your own testing?