SENG 265

Testing

Testing

- How would you answer the following questions (posted by R. Baber in a 1982 textbook):
  - Would you trust a completely-automated nuclear power plant?
  - Would you trust an anti-lock braking system whose software was written by yourself? What if it was written by one of your colleagues?
  - Would you dare to write an **expert system** that helps busy oncologists diagnose for cancer? What if you are personally held liable in a case where a patient suffers due to malfunction of the software?
- We may find it difficult to answer “yes” to these questions, yet some of us board modern fly-by-wire aircraft with (relatively) little worry

State of the software art

- We are not yet able to deliver fault-free software
  - Various studies report that for every 1000 lines of code written, there are 30 to 85 errors
  - Testing catches many of these errors
  - In extensively tested software, there are still 0.5—3 errors per 1000 lines of code.
- Errors can have serious consequences for enterprises using the software
  - Example: airline reservation system fault (consequence is that are seats sold far below cost of provisioning them)

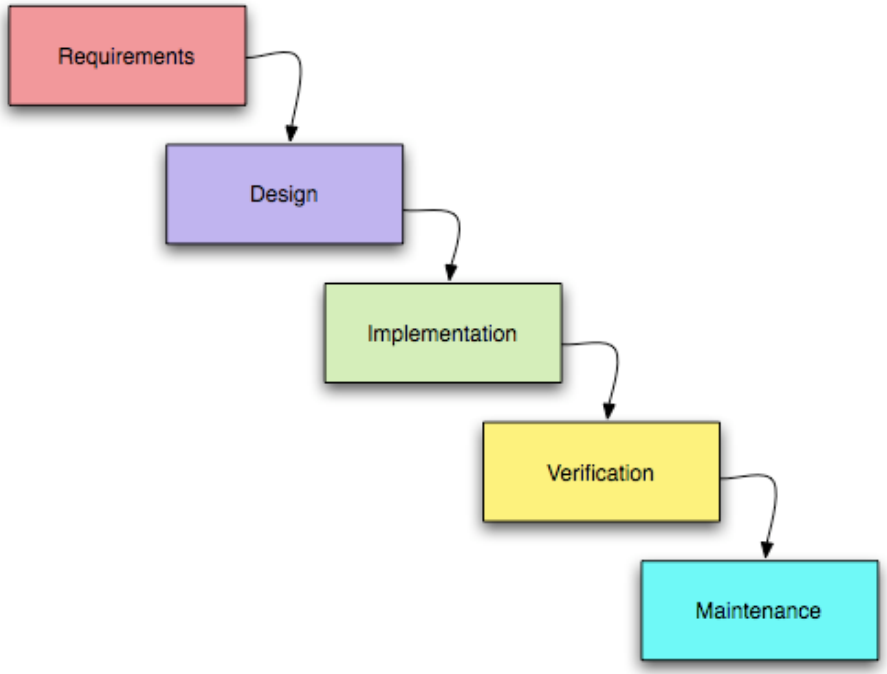
Challenge – when to start testing

- Postponing test activities is often a serious mistake
  - Yet such decisions to postpone are often made
  - There is a relation between **when** a mistake is found and **how much it costs** to repair it
- Most errors are made in the early phase of the software development cycle
  - Barry Boehm: 60% of errors introduced during design phase, 40% during implementation phase
  - in his study, two-thirds of the errors introduced during design were not discovered until software was operational!

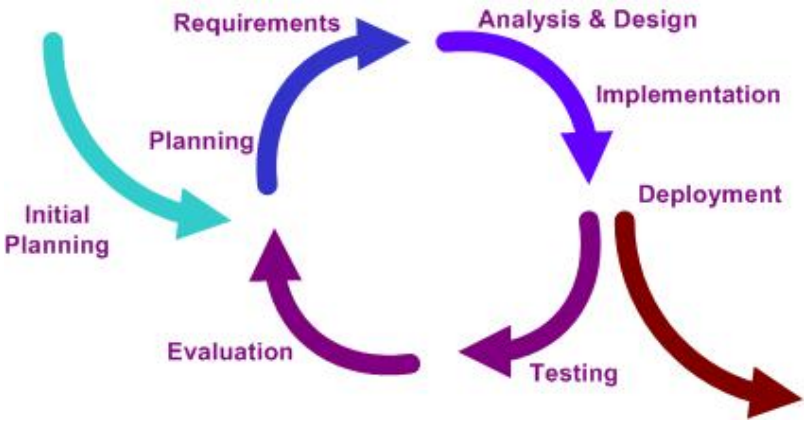
Change of view

- Many **software-development process models** separate testing into its own phase.
- We can instead view testing as an integral part of the whole software-development process
- This means some "testing" is carried out in each of the phases
  - Note that testing does not necessarily require running code
  - The degree to which “manual” tests succeed in catch bugs depends upon the clarity of relevant documents
- Our approach to testing also determines objectives.
- What are these phases?

Software-Development Process Models



model 1: "Waterfall"



model 2: "Iterative"

Each major group of activities is often referred to as a "phase".

Testing objectives: relating to phases

- Requirements-phase testing
  - Requirements are reviewed with the client
  - **Rapid prototyping refines requirements**
  - Also aids in accommodating changing requirements
- Specification-phase testing
  - Specification document checked for feasibility, traceability, completeness, absence of contradictions (i.e., consistency) and ambiguities
  - **Specification reviews (walkthroughs, inspections) are very effective**
- Design-phase testing
  - Similar to specification reviews, but more technical.
  - Design checked for **logic faults**, **interface faults**, lack of **exception handling**, and **nonconformance** to specifications.

Testing objectives: relating to more phases

- Implementation
  - Code modules are informally test by programming while they are implemented (desk checking)
  - More formal testing done methodically by a testing team.
  - **Non-execution-based** methods (code inspections, walkthroughs)
  - **Execution-based** (black-box testing, integration testing)
- Integration
  - Ensure modules correctly combine into product meeting specification
  - **Focus on interfaces between modules.**
  - Top-down, bottom-up
- Product Testing
  - Functionality of whole product compared against specifications.
  - **Test cases derived from specification document**.
  - Some testing for robustness (error-handling capabilities, stress tests)
  - Checks for **completeness** and **consistency**

Testing objectives: even more phases

- Acceptance testing
  - Software is delivered to client, who tests it on site hardware
  - **Uses actual data (not test data) and infrastructure**
  - Product doesn't "satisfy its specifications" until acceptance test is passed
  - (alpha, beta testing)
- Maintenance
  - modified version of originally product tested to ensure changes are correctly implemented
  - **Testing against previous cases (regression testing)**
- Software Process Management
  - **actual management plan is scrutinized**
  - cost and duration estimates checked against actual numbers

Summary so far

- Every day we depend upon systems that operate with the assistance of software…
- … and we do so even though we know it is hard to convince ourselves such systems are correctly built.
- "Testing" describes a critical activity
  - What constitutes testing depends upon which phase of the software-development process we are in
- In it's broadest sense, testing is far more than just finding bugs in code.