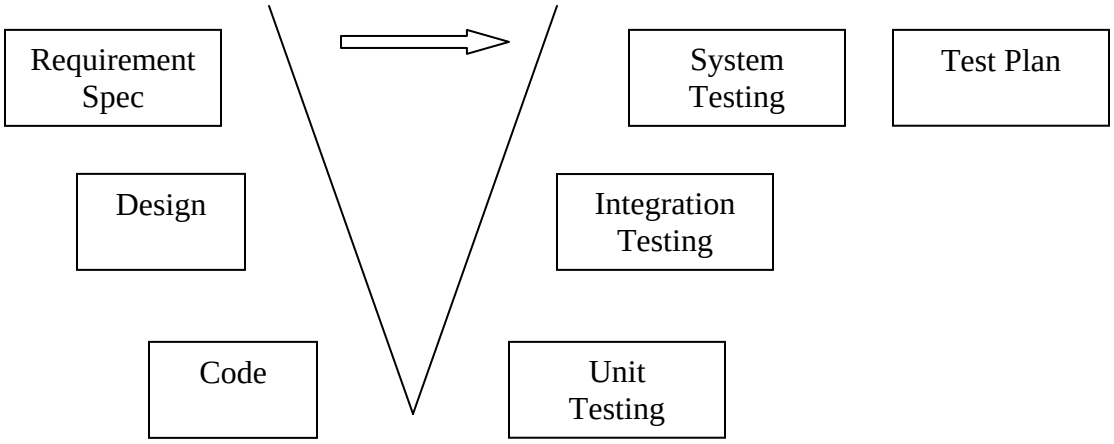
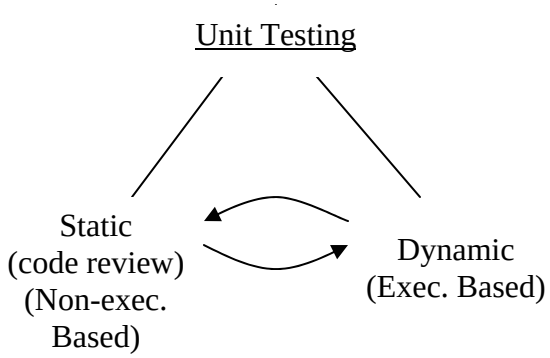
The V-model of Testing and Code Review

Scope of these notes: These notes have not been designed to serve as substitutes for missed classes. Rather, these should be considered as supplementing your own notes taken during lecture hours.





Static and dynamic unit tests should be performed alternatively.

Static Unit Testing

Philosophical view:

Any component/product should undergo a phase of inspection and correction at each milestone in its life cycle.

Idea in inspection:

- Find defects as close to their points of origin as possible.
- Do it with less cost.

In static unit testing, code is reviewed by using two commonly known techniques:

- Inspection [Michael Fagan]
- Walkthrough [Edward Yourdon]

} Code Review

Inspection: It is a step-by-step peer group review of a work product, with each step checked against predetermined criteria.

Walkthrough: The author leads the term through a manual/simulated execution of the product using predefined scenarios.

In our discussion, we will consider *code review*, which is a combination of inspection and walkthrough, instead of its individual components. Code review is more popular than the individual inspection and walkthrough processes.

Code Review

- Performed by a group of programmers and designers
- Performed in 5 steps
  - Readiness for review
  - Preparations
  - Review
  - Re-work
  - Exit

Readiness for review

<u>* The author of the code ensures that the code is ready for review, as explained below</u>:

- Completeness of code
  (The code is not going to be significantly modified. Readers can't understand incomplete code.)
- Minimal functionality: The code must compile and link, and must have been tested to some extent.
- Readability: Because it will be read by other programmers
  - Proper formatting
  - Use of meaningful identifier names
  - Straightforward use of language constructs

    - Appropriate level of abstraction using function calls.
- Complexity:
    - No need to review straightforward code. (TBD by the programmer)
    - The code must be of sufficient complexity described in terms of
        - Conditional statements
        - The # of input data elements
        - The # of output data elements
        - Real-time processing
        - Communication with other units.
- Requirements and low-level design is available.

* Duration and size
- Reviews are done in bursts of 1~2 hrs. (Due to the limited attention span of human beings)
- Rate of review: 125 LOC per hour (high level language)

* Composition of review group
- Moderator:
    - Trained individual
    - Selects reviews
    - Schedules and chairs sessions
- Author of the code to be reviewed
- Presenter: Reads the code beforehand
- Record keeper: Documents the problems found and follow-up actions suggested.
- Reviewers: Experts in the subject area of code.
- Observers: People willing to learn about the code

* Schedule the group review:
- All are informed 2/3 days before the meeting
- They are given a package for perusal

Preparations

- Reviewers carefully review the work package to read and understand the code
- Each reviewer develops a list of the following items:
    - Questions to be asked regarding the code
    - Potential change request (CR) to report bugs
    - Suggested improvement opportunities: How to fix the problems

Review process

- The author presents the procedural logic in the code, paths denoting major computations, and dependency of the unit on others.
- The presenter reads the code line by line.
- Questions are raised
- Potential problems uncovered
- Reviewers may make general suggestions

- Record keeper documents the CR
    - Description of the problem, priority, person assigned for follow-up, target date for fixing.
- The moderator ensures that the meeting remains focused and progress is made
- Decision taken
    - Need for extensive re-work => another meeting

Re-work

At the end of the meeting, the record keeper produces a summary of the meeting.

- List of all CRs
- List of improvement opportunities
- Minutes of the meeting (optional)

The CRs are independently validated by the moderator or another person.

Exit

A review process is said to be complete if

- Every LOC in the module has been inspected
- If too many defects are found. The module is reviewed again.
  > 5% of LOC are contentions => second review
- The author and the reviewers reach a consensus that when corrections are applied, the code will be potentially free of bugs.
- All the CRs are documented + validated
- The author's follow-up actions are documented.
- A summary report of the meeting is distributed to all members of the review group and the SQA.

Code Review Questionnaire

Reviewers may keep the following questions in mind while preparing their lists of items discussed above.

1. Does the code do what has been specified in the design?
2. Does the procedure used in the code solve the problem?
3. Does a module duplicate another module?
4. If used, are the right libraries being used?
5. Is the *Cyclomatic* complexity of the module more than 10?
   (The number of predicates plus one in a routine is a measure of the complexity of the routine.)
6. Can each atomic function be reviewed and understood in 10~15 seconds?
7. Have naming conventions been used?
8. Have variables and constants been correctly initialized?
9. Are data values hard coded in the program?
10. Are memory blocks de-allocated after use?

11. Does the module terminate abnormally?
12. Is code efficient?