Software Quality

- Software engineering is the production of quality software
- External quality
  - May be detected by its users
- Internal quality
  - Perceptible only to computer professionals who have access to the actual software text
- What matters is the external quality, but it can only be achieved through internal quality

External Quality

- Correctness
  - Perform the exact tasks as specified
- Robustness
  - Ability to react appropriately to abnormal conditions
- Extendibility
  - Ease of adapting program to changes of specification
- Reusability
  - Ability of software elements to serve for the construction of many different applications.
External Quality

• Other Quality
  - Compatibility, Portability, Ease of Use, Efficiency, Timeliness, Integrity
• Tradeoffs
  - Integrity v.s. Ease of Use
  - Efficiency v.s. Portability
  - Timeliness v.s. Extendibility

Software Life Cycle

Phase 1: Specification

• Answers “What do we build?”
• Define clearly all aspects of the problem
  - What is input (valid/invalid) data?
  - What assumptions are possible?
  - Are there special cases?
  - What enhancements are likely in the future?
Software Life Cycle
Phase 2: Design
• Divide into manageable parts - modules
• Specify each module’s purpose, assumptions, input, and output
• Develop algorithms
• Look for existing software components

Software Life Cycle
Phase 3: Risk Analysis
• Attempts to answer “What can go wrong, and how bad can it be?”
• Predict and manage what risks you can
• Risks to timetable, cost, human health, etc. should be taken into account
• Risks can greatly influence the direction of a project

Software Life Cycle
Phase 4: Verification
• Answers “Are the algorithms correct?”
• Some algorithms can be proven correct
  - assertion: condition at a certain point
  - invariant: condition that is always true
**Software Life Cycle**

**Phase 5: Coding**

- Translate the algorithms into a particular programming language
- Minor phase in the software life cycle

**Software Life Cycle**

**Phase 6: Testing**

- Answers “Did we build it correctly?”
- Try to make the software fail
- Develop as many test cases as possible
- Both a science and an art

**Software Life Cycle**

**Phase 7: Refining a Solution**

- Add bells and whistles
- Retest after any changes are made
Software Life Cycle

Phase 8: Production
- Distribute it
- Install it

Phase 9: Maintenance
- Fix previously undiscovered bugs
- Add new features
- Enhance old features
- Generally the most costly phase (80% of total cost by some estimates)

Documentation
- Performed extensively in every phase
- In modern software development, there are typically different people working on each phase of the software life cycle
- Novice programmers usually undervalue documentation
Continuing Effort