Week 13: Implementing a Software Quality Program
Organizational opportunities and obstacles

COMP 370 -- Fall. 2014
Mr. Weisert

Quality plan & quality assurance

- Quality is not:
  - a separate attribute of software
  - a separate set of activities in development

- Every step in analysis, design, coding, testing, etc. must emphasize quality in the broad sense of:
  - internal quality (maintainability)
  - external quality (reliability, freedom from defects)

Infrastructure for a software development organization

- Role definitions
- Written methodology
- Staff education & training
- Tools
- Support (consulting and "help")

Roles

- What is:
  - the development team?
  - the quality team?

- When are they formed?

- What roles are required?
Role definitions
- More detailed than personnel department's typical job description.
- Specifies exactly what the incumbent does, not what he or she assists with or participates in.
- One enlightened 3-part structure:
  - Responsibilities (or activities)
  - Accountability (or measures of performance)
  - Authority (what he or she may do without specific authorization)
- If personnel (or "human resources") won't support them, I.T. must prepare their own.
- Incumbent and manager must agree, esp. on accountability.

Levels of software development methodology
- Mandatory standards
- Conventions
- Guidelines
- Helpful information
  Why do we have each one?
  How are they enforced?

Two Methodology approaches
- **Military** approach
  - Lots of rules, emphasis on mandatory standards (easier to enforce)
  - Rules specify detailed steps.
  - We don't trust the staff, and assume they'll cheat whenever they get the chance.
  - A committee or expert prepares the standards.
- **Professional** approach
  - Emphasis on quality of the end product.
  - We trust the staff to use good judgment and assume they want to cooperate.
  - Mandatory standards only where there a global or future impact.
  - Staff contributes content (participative)

There is (or should be) no such thing as
- Analysis & design
- Waterfall approach
- Quality strategy
Analysis & design

- Two entirely separate disciplines!
- Both are critically important to the success of a development project.
- Systems Analysis:
  - Defines what the proposed application will do (detailed user requirements or external design)
  - Usually done by a systems analyst.
- System design:
  - Defines how the proposed application will be built ("architecture:"
  - Usually done by a senior programmer.

Developer role

- We're seeing more and more recruiting for "developers", who can presumably do both analysis and design (!) as well as coding and testing.
- But very few people have the skills, interest, and temperament for both roles.
- What happens when an organization "promotes" its best programmer to become a systems analyst?
- COMP 320 is about systems analysis

Approaches to testing

- Top-down versus bottom-up versus big-bang

- Black-box versus white-box
  - Also called Functional versus Structural

- Test-driven development
  - How does each work?
  - What are advantages & disadvantages of each?

Testing phases

- Review (or walkthrough)
- Unit test
- Integration test
- System test
- Volume (or stress) test
- Alpha test
- Beta test
- Acceptance test

Do we always do all of them?
Test driven development

- Many "agile" experts favor TDD, especially for unit testing.
- We develop the test driver code before we develop (or even design, except for interfaces) the MUT!
- Test cases may serve as a substitute for rigorous specifications; may help guide implementation.

What could possibly go wrong with that?

Developing and maintaining an organization's methodology

- Does every software-development organization really need to specify and disseminate its own standards and conventions?
  - Develop in-house or buy from an expert?
- What if everyone on the programming staff is a highly experienced senior practitioner?
  - Can we just trust them to do superior work?

Capability Maturity Model for an organization

- A software development organization's written methodology provides a framework to assure repeatability, predictability, and consistency.
  - In-house education supports the methodology
- An organization without such methodology documentation must depend upon the experience, professionalism, and biases of each project team.

Where does an organization's methodology come from?

CMM framework (from SEI)

- A software development organization is at one of these capability maturity levels:
  1. Initial (chaotic, ad hoc)
  2. Repeatable
  3. Defined
  4. Managed
  5. Optimizing

What do those mean?

Why do we care?

Which should we aim for?

http://en.wikipedia.org/wiki/Capability_Maturity_Model