

## Quality Assurance: Early Work Items

CSE 403  
Lecture 21

Slides derived from a  
talk by Ian King

## Key Points

- Many different characteristics of quality
- Importance of having independent quality assurance from development
- The deliverable of QA is information
- Write it down
- QA is not free

## What is the 'value add' of QA?

- Classic research: fixing a defect is cheaper the earlier you catch it
  - \$: found in spec process
  - \$\$: found during implementation
  - \$\$\$: found during post-implementation testing
  - \$\$\$\$: found in the field
    - Service packs, product recalls, lawsuits
    - Customer confidence

## What is quality?

## 'Make It Didn't Happen'

- The best bug is the one that was never born!
- QA is about process
  - Design review
  - Implementation review
  - Structured testing and evaluation
  - Instrumentation/testability
  - Best practices
- Lesson of History: good process leads to fewer defects

## The deliverable of QA

- QA delivers information
  - What is known about the quality of the code?
  - What are the risks of known defects?
  - What is not known, i.e. untested?
  - What risks may arise from unknown defects?
    - E.g. "We didn't test for malicious use"
- "Bearer of bad news"
- "Validation of the vision"

## Scaling to the project

- Large project
  - Individual design/programming and QA teams
  - Another team to coordinate and administer
- Medium-sized project
  - QA often assumes coordination role
- Small/solo project
  - Develop 'functional schizophrenia'
  - Write it down

## Structure of QA

- Independence
- Authority
- Resources

## What does QA do early in the development cycle?

- Publish (and promote) QA requirements
- Review design work
- Develop testing strategy

## Establish QA Requirements

- Statement of requirements
- Feature specifications
- Implementation specifications
- Design change process
- Development schedule
- Build process
- Developer practice
- Defect process
- Release criteria

## Design Change Process

- How are design changes documented?
  - DCR vs. "bug"
- How are change decisions made?
- When has "the ship sailed"? Design Freeze

## Quality Assurance is about Information

- Specification and requirements
- If it's not written
  - It's forgotten
- Bug database is an asset

## Development Schedule

- When will specs be complete?
- When will code be available?
- When will features be complete?
- When will code be stable?
- Beta releases?
- Leave enough time for the endgame:
  - Complete test pass on Release Candidate
  - Test of final installation media (may include digital signing, release notes)

## Build Process

- Source control
  - Undo the 'oops
- Centralized build
  - Be sure everyone is testing the same bits
  - Avoid platform dependencies
  - How often are new builds generated?
    - Periodic
    - Event-Driven
- Configuration management

## Developer Practice

- Private builds
- Buddy builds
- Code review
- Code analysis tools
- Unit testing

## Defect Process

- Why are defects tracked?
- How are defects tracked?
- What is the lifecycle of a bug?
- How are defects prioritized?
- Controlled check-ins/triage process
- Defect analysis:
  - Defect source analysis
  - Root cause analysis

## Measuring quality

- Is it possible to quantify software quality?

## Release Criteria

- When are we done?
- Indicators of completeness:
  - Quantity of defects being found
  - Severity of defects being found
  - Completeness of testing



## Review Design Work

- Are these documents sufficient to scope the project?
- Are they logically consistent?
- Is the project testable?
  - Test hooks, registry entries, compiler directives
  - Instrumentation
- Does the project address the stated requirements?



## Review Design Work (con't.)

- Evaluate use scenarios
  - Sensible control flows?
  - Features appropriate to use?
- Evaluate failure scenarios
  - Meaningful error feedback
  - Single points of failure
  - Cascading failures
- Understand dependencies



## Costs of quality assurance

- Programmer Productivity
  - 8-20 LOC / day
- Building QA into the schedule