

Chapter 21: Collaborative Construction

- Basically rubber ducking
- In one way or another, all collaborative construction attempts to formalize the process of showing your work to someone else with the end goal of flushing out errors

Overview of Collaborative Development Practices

- Main idea is two people shouldn't have the exact same blind spots

Collaborative Construction Complements Other Quality-Assurance Techniques

- Primary purpose of collaborative construction is to improve software quality
- The time (and cost associated) with two people reviewing code is factors less than the cost associated with fixing bad code
- Stats
 - o IBM found each hour of inspection prevented 100 hours of testing and defect correction
 - o Raytheon reduced cost of defect correction from 40% to 20% of total project cost
 - o Imperial Chemical Industries found that maintaining a big portfolio of inspected software packages was 10% as expensive as uninspected ones
 - o Each hour spent on inspections prevented 33 hours of maintenance and 20 times more efficient than testing
- Human reviewers can spot unclear error messages, inadequate comments, hard-coded variable values and repeat code patterns that should be consolidated. Testing wont.

Collaborative Construction Provides Mentoring in Corporate Culture and Programming Expertise

- Informal review procedures were passed on from person to person in the general culture of computing for many years before they were acknowledged in print. The need for reviewing was so obvious to the best programmers that they rarely mentioned it in print, while the worst programmers believed they were so good that their work did not need reviewing. – Weinberg

Collective Ownership Applies to All Forms of Collaborative Construction

- All code is owned by the group rather than by individuals
- Benefits
 - o Better code quality arises from multiple sets of eyes seeing the code and multiple programmers working on the code
 - o The impact of someone leaving the project is lessened because multiple people are familiar with each section of code
 - o Defect-correction cycles are shorter overall because any of several programmers can potentially be assigned to fix bugs on an as-available basis

Collaboration Applies as Much Before Construction as After

- Requirements, architecture, design, estimates, testing, maintenance all should be collab tasks

Pair Programming

- Benefits
 - It holds up better under stress than solo development. Pairs encourage each other to keep the code quality high even when there's pressure to write quick and dirty code.
 - It improves code quality. The readability and understandability of the code tends to rise to the level of the best programmer on the team
 - It shortens schedules. Pairs tend to write code faster and with fewer errors. The project team spends less time at the end of the project correcting defects
 - It produces all the other general benefits of collaborative construction, including disseminating corporate culture, mentoring junior programmers, and fostering collective ownership
- Checklist
 - Do you have a coding standard so that pair programmers stay focused on programming rather than on philosophical coding-style discussions?
 - Are both partners participating actively?
 - Are you avoiding pair programming everything and, instead, selecting the assignments that will really benefit from pair programming?
 - Are you rotating pair assignments and work assignments regularly?
 - Are the pairs well matched in terms of pace and personality?
 - Is there a team leader to act as the focal point for management and other people outside the project?

Formal Inspection

- Inspections differ from run of the mill reviews in several ways:
 - Checklists focus the reviewers attention on areas that have been problems in the past
 - The inspection focuses on defect detection, not correction
 - Reviewers prepare for the inspection meeting beforehand and arrive with a list of problems they've discovered
 - Distinct roles are assigned to all participants
 - The moderator of the inspection isn't the author of the work under inspection
 - The moderator has received specific training in moderating inspections
 - The inspection meeting is held only if all participants have adequately prepared
 - Data is collected at each inspection and fed into future inspections to improve them
 - General management doesn't attend the inspection meeting unless you're inspecting a project plan or other management materials. Technical leaders might attend
- Typical to expect catching 60% of defects

Roles during and Inspection

- Moderator
 - Responsible for keeping the inspection moving at a rate that's fast enough to be productive but slow enough to find the most errors possible
 - Needs to be technically competent but not an expert
- Author

- Person who wrote the code
 - Doesn't do much, the idea is the code and the design speaks for itself
- Reviewer
 - Anyone with an interest in the code that's not the author
 - Maybe an architect or designer
- Scribe
 - Person who records errors
 - Neither the moderator nor the author
- Management
 - Usually a bad idea to include people from management
 - Under no circumstance should inspection results be used for performance appraisals

General Procedure for an Inspection

- Planning
 - Author gives design or code to moderator
 - Moderator decides who will review the material and when and where it will occur
 - Moderator distributes design or code and checklist that focuses the attention of the inspectors
- Overview
 - Try to avoid overview of project as it can lead to glossing over unclear points in the design or code being inspected
 - The code/design should speak for itself, the overview should speak for it
- Preparation
 - Each reviewer works alone to scrutinize the design or code for errors
 - Reviewers should use the checklist to simulate and direct their examination of the review materials
 - Keep record of preparation rates for future estimation
 - Some organizations have found that by assigning reviewers a certain role, like customer, maintenance programmer, or designer, yields good results
- Inspection meeting
 - Moderator assigns someone other than the author to design or read the code
 - All logic is explained
 - Scribe takes notes
 - Try not to go too slow, and try not to go too fast
 - DO NOT discuss solutions during meeting
 - Do not let run longer than two hours
- Inspection report
 - Within a day of the inspection meeting, the moderator produces an inspection report (email or the like) that lists each defect with its type and severity
 - These are good for programmers but also great historical data as an organization and on each developer
- Rework
 - The moderator assigns defects to someone, usually the author, for repair
- Follow-up

- The moderator is responsible for seeing all rework assigned is carried out
- Third-Hour meeting
 - After inspection complete, interested parties may want to discuss solutions

Egos in Inspections

- The point of the inspection is to discover defects in the design or code
- NOT to explore alternatives
- NOT to debate over who is right and wrong
- CERTAINLY NOT the criticize the author of the code
- It should NOT convince the author that the team is a bunch of jerks and they should look for a new job
 - If something like this should occur, the moderator should make it unmistakably clear how inappropriate those comments are