CSC444 Software Engineering I Fall 2002

Assignment 3: quality assessment

last updated: Oct 13 2002

Content

The assignment is to conduct an inspection of part of your phase 1 module (use Fagan's guidelines to determine a suitable sized unit to inspect). You will conduct the inspection during your a tutorial. Prior to that tutorial you should prepare for the inspection, by selecting a suitable section of your program to inspect. Choose a coherent section of your program, using Fagan's length guidelines. Each member of the team should familiarize themselves with the code and fill out a typo log *prior to* the tutorial. You should also choose inspection roles before the tutorial.

Note: If, after the inspection in the tutorial, you are unhappy with how this inspection went, you may choose to conduct a second inspection in your own time, and report on that).

You should submit the following:

1) A description of the code you chose to inspect, with some rationale for why you selected it. Include details of any testing already performed on this code, and your assessment of its quality *before* the inspection.
2) A description of the inspection process you used. What roles did your inspectors take on? How did you structure the inspection meeting?
3) The results of the inspection. Use the forms on the course website to document your inspection results: you should end up with the following:
   - One inspection Typo log for each inspector
   - One inspection Issue log
   - One inspection Summary Report form

   Use continuation sheets if you cannot fit everything on a form.
4) A discussion of the lessons learnt from your inspection meeting. Use the "Inspection Lessons Learnt Questionnaire" as a guide to help you think about what you learnt.

You should also hand in a listing of the program code that you inspected, in exactly the form you used for the inspection meeting.

Background Information

Informal reviews happen all the time in software engineering, e.g. any time when two or more people chat about or comment on various aspect of the software. A formal inspection is a scheduled meeting, with an agenda, and a written output. Furthermore, it is 'technical' in the sense that it concentrated on the technical aspects of a product, rather than scheduling, budgeting, or other management concerns of the development process. There are a variety of inspection types described in the literature, with different names: Formal technical reviews, Fagan inspections, etc. Many of the principles are the same, no matter which variety of inspection you use.

Objectives of a formal inspection process:

- to uncover errors in function, logic or implementation
- to verify that requirements are met
- to ensure that standards are complied with
- to achieve uniformity (of style, quality, etc.) across a project
- to collect data on error profiles so that these errors can be avoided on future projects
- to train junior software engineers (by allowing them to review other's work)
- to promote continuity across teams

Formal inspection has been shown to be more effective than testing in ensuring that programs are error-free. Furthermore, it can be used on products which cannot be tested, such as specifications, designs, documentation, manuals, test plans, etc.

Formal inspections generally have two main parts: a scheduled review meeting, and individual inspection by each member of the team *prior to* the scheduled meeting. An inspection team should consist of between 3 and 7 people, depending on how experienced the review leader is, and how well the necessary types of expertise can be covered. The review meeting should last for no more than 2 hours. It should focus on a small manageable portion of a product, not on the whole thing. It should be held only after the author of that product has finished it, but in plenty of time to take action on the results of the review. All reviewers should agree on the outcome, which may be to accept or reject the product, or to recommend specific modifications.

The following roles (at least) will be needed during the review meeting:

- leader - chairs the review, ensuring it remains focused, and that everyone contributes
- reader – (possibly the author) steps through the product inviting comments from the review team.
- recorder - keeps a public record of issues raised, preferably visible to the reviewers (e.g. on a whiteboard)

The material to be inspected should be circulated to all reviewers in plenty of time for them to prepare for the review. Each reviewer should spend around two hours preparing, by familiarizing themselves with the product, and noting any issues they wish to raise in the review meeting. Unprepared reviewers are not much use.

- Choose the reviewers carefully, from amongst the development team, from other teams, from review specialists, from interested parties, from visiting experts, etc.
- Exclude anyone who creates a conflict of interest, e.g. the author's line manager.
- Don't proceed with the inspection if some of the reviewers are not present, or have not prepared properly.
- Review the product, not the person
- Stick to the agenda
- Limit debate and rebuttal – defer contentious issues to be discussed after the meeting.
- Identify problems but don't try to solve them
- Use checklists where appropriate
- Schedule inspections into the project plan
- Train all reviewers
- Review the inspection process itself occasionally.

Types of Inspection:

1) Checklist - use a checklist of questions or issues appropriate to the type of product being reviewed. The meeting is structured around the checklist – consider each item on the list in turn during the meeting.
2) Walkthrough - One person presents the product step by step, with reviewers raising issues when necessary. The structure of the meeting reflects the structure of the product.
3) Round Robin - Each reviewer in turn gets to raise an issue. The structure of the meeting reflects the composition of the review team.
4) Speed Review - Each reviewer gets a short time (e.g. 3 minutes) to review a small chunk of the product, before passing it on to the next person. This combines a walkthrough with a round robin, and is good for assessing comprehensibility, as the reviewers only get a limited time to understand each chunk.

Marking scheme

Listed below are some of the things your TA will look for when marking this assignment. Use this list to check your work before you hand it in.

Did they select an appropriate piece of program to inspect, and give some rationale for their choice?

- Length should be appropriate: anywhere from 200-400 lines.
- It should be a coherent chunk of code (i.e. not just the first 200 lines of a module, etc).
- Did they give a sensible explanation for how they made the choice?
- The explanation should consider timing of the inspection vs development stage of the chunk
- The explanation should include comments about the possible benefits of inspecting this particular chunk.

Did they give a brief description the expected quality of the chosen chunk? This only needs to be brief and informal.

- Is it clear how much testing has been done prior to the inspection?
- Did they give sensible estimates of how many remaining errors they expected there to be?
- Did they give sensible predictions of types of errors the inspection might uncover?

Did they describe the process they used for the inspection?

- Did they describe the preparation they did prior to the inspection?

- Did they describe how the inspection meeting itself was structured (Not just "walkthrough" or a "round robin" - they need to give some detail about what they actually did).
- Did they describe a follow-up process?
- Did they give sensible reasons for the various choices of process?
- Did they show insights into what will work in their particular context when making these choices?

Did they describe the roles they each played in the meeting?

- Was there a designated leader and a designated recorder?
- Did everyone have a clearly defined role (even if that role is just “inspector”)
- Is it clear who played which role?
- Is it clear that they understood the duties of each role?
- Did they describe some sensible rationale for how they allocated the roles to members of the team?

Have the forms been used correctly?

- is there 1 typo list for each participant, and 1 Issue log and 1 inspection summary?
- Are the forms filled out properly?
- Did they keep the information concise? (Penalize teams that include large amounts of superfluous information - the forms are meant to capture all the important info collected during the inspection).

Are the issues/defects that were found during inspection clearly described?

- Is there enough description to guide the author for when it comes to fixing each defect?
- Is there a clear indication of the location of each problem?

Have they used the classification of different types of defect properly, and are there a range of different types of defect?

- Did they correctly identify the “origin” of each issue?
- Did they correctly identify the “type” of each issue?
- Did they sensibly distinguish between major and minor issues?
- Have they found a reasonable range of different types of issue? Even for code that was thoroughly tested beforehand, I would expect a range of different types of programming style, efficiency, dead code, etc, problems. *If they claim the code really was close to perfect, then they must provide a convincing explanation of how this could be to get credit for this point.*
- Based on the issues discovered, does it seem that the inspection was thorough?

In the lessons learnt section, did they comment on how well the inspection worked?

- Did they provide a good critique of whether their chunk of code inspected at the right point in its development?
- Did they give a good critique of the structure of the inspection meeting itself (preparation, roles played, organization of the meeting)
- Did they comment on the outcome (e.g. was the list of issues generated useful? If not why not?).

In the lessons learnt section, did they come up with some sensible suggestions for how they would improve the inspection process (or what they would do differently) next time?

- Give credit for any sensible suggestions.

Did they include a listing of the code they inspected?

- Are the locations of the defects documented on the forms easy to identify in the code listing?.