

Deeper 📌

- Why do reviewers find problems the author can't find?
 - The author, in studying the problem, has come to conclusions (and assumptions) that the reviewers have not (yet) made.
 - The author may review something and see what (s)he intended, rather than what (s)he actually wrote.
 - Different people draw on different perspectives and experiences .

9/29/2012

Review Processes

32

Deeper 📌

- How do reviews encourage better work?
 - If we know that our work will be reviewed by people we respect, we will be strongly motivated to present them with our best work.
 - If we observe problems in other peoples' work we will try to avoid them in our own.
 - If we observe things that have been done very well in other peoples' work, we are likely to try to emulate their example.

9/29/2012

Review Processes

33

Deeper 📌

- Why do we call out these particular work products for review?
 - Each of these work products serves as the basis for significant amounts of future work. This means that errors in these work products will experience defect amplification, and be much more expensive to fix if found later.
 - Find mistakes before you base substantial additional work on them.

9/29/2012

Review Processes

34

Deeper 📌

- Why should a work product be “complete” before it is submitted to formal review?
 - It is a waste of time to ask other people to find problems that the author already knows about.*
- Does this mean you can't get help from your peers until you are mostly done?
 - You should feel free to discuss problems with others throughout the process. The purpose of a formal review is to ensure that a work product is complete and correct.*

9/29/2012

Review Processes

35

Deeper 📌

- Why is testing only 35% to 55% effective at finding defects?
 - Because testing only covers specifically enumerated situations ... and we don't think of everything, and we don't develop test cases for everything we think of.
 - Because test cases don't cover missing requirements and specifications.
 - Because test plans and test cases are developed by people who are just as fallible as the people who develop the code.

9/29/2012

Review Processes

36

Deeper 📌

- How do reviews find problems sooner than testing?
 - Reviews find many problems before the code is even written.*
- Why are reviews more efficient at defect elimination than testing?
 - We save the time we would spend writing the code and test cases, running the test cases, debugging the problem, and fixing the code.*

9/29/2012

Review Processes

37

Deeper

- If Pair Programming was not an effective training tool, why are reviews better?

The problem with putting untrained people in programming pairs is that they are unable to substantially contribute to the work (thus defeating the purpose).

In reviews they are able to learn process, methodology, and technology without slowing down the review process. Even if they only observe, they do not reduce the group's productivity.

Also, studying code is a great way to learn.

9/29/2012

Review Processes

38

Deeper

- Are there problems with having the author explain the code?
 - (a) *The code should speak for itself. If it doesn't, it needs to be better commented.*
 - (b) *If the author explains his/her intent for the code, reviewers may buy in to the approach without determining its correctness for themselves.*

9/29/2012

Review Processes

39

Deeper

- Why should the scope and goals of the review be explicitly stated?
 - (a) *So reviewers will know what kinds of issues they are and are not looking for.*
 - (b) *To avoid wasting time preparing or discussing inappropriate comments.*
- Why might interesting topics be designated out of scope?
 - (a) *Because they have already been resolved.*
 - (b) *Because they will be discussed elsewhere.*

9/29/2012

Review Processes

40

Deeper

- Why is it important to lay out the order in which pieces will be reviewed?
 - (a) *So reviewers will know when it is appropriate to raise which issues.*
 - (b) *Because it may be easier to discuss some parts after other parts have already been reviewed.*

9/29/2012

Review Processes

41

Deeper

- Why should the moderator be someone other than the author?
 - The moderator's primary responsibilities include ensuring that all points of view are heard, and that the focus stays on defects in the code rather than ego issues.*
 - This requires that moderator to be neutral and dispassionate ... which would be difficult for the author.*

9/29/2012

Review Processes

42

Deeper

- Why should the reader be someone other than the author?
 - The reader is supposed to move the discussion through the code, giving reviewers an opportunity to make comments about each section.*
 - This author might feel compelled to make his/her own observations about the code during the reading, and to respond to comments raised by other reviewers.*

9/29/2012

Review Processes

43

Deeper

- Why is the author not included in the reviewers?

The author has already had ample opportunity to make observations about, and to make changes to the code. The purpose of the review is to get input from others.

9/29/2012

Review Processes

44

Deeper

- Why should the recorder/scribe be someone other than the author?

The scribe keeps a record of all issues raised, all action items assigned, and writes the report. This should be done by someone who has no ego in the code or responsibility for its delivery.

9/29/2012

Review Processes

45

Deeper

- How do we recognize and deal with digressions, rat-holes and rehashing old decisions?

It is the moderator's responsibility to keep the discussion on-target and moving.

- *If the discussion seems to be digressing, the moderator should bring the discussion back to the components under review.*
- *When the discussion gets into a rat-hole, the moderator should stop the discussion, note the issue for the report, and move on to the next topic.*
- *When old issues are being rehashed, the moderator should remind the reviewers what the scope of this review is.*

9/29/2012

Review Processes

46

Deeper

- What should we do if reviewers have difficulty understanding the materials?

Confusing materials are supposed to be caught by the moderator before the review is scheduled.

If, however, it becomes clear that the materials are too confusing (or that the wrong reviewers have been selected) the moderator should immediately end the review. The materials should be revised and/or new reviewers should be selected, and the review should be rescheduled.

9/29/2012

Review Processes

47

Deeper

- What should we do if it becomes clear that one or more reviewers have not done the necessary study and preparation?

The moderator should ascertain the preparedness of the reviewers at the start of the review.

Unprepared reviewers should be immediately dismissed from the review. If there are not enough reviewers to complete the review, the session should be cancelled and rescheduled.

9/29/2012

Review Processes

48

Deeper

- What should we do if the reviewers get burned out from too long a review sessions?

Review sessions should be limited to 1-2 hours maximum. If a component is too complex to be reviewed in that time, it should be broken up into smaller sub-units for review purposes.

9/29/2012

Review Processes

49

Deeper

- What should we do if the reviewers get into arguments about issues and approaches to a problem?

The moderator should remind the reviewers that the purpose of the review is not to solve problems, but to identify them. Fixing problems is the responsibility of the author, and discussions about how to fix them should be conducted outside of the review meeting.

9/29/2012

Review Processes

50

Deeper

- Why do we limit discussion to defects, and not allow discussion of solutions?
 - (a) *Because defects are relatively objective, whereas approaches may be more matters of opinion.*
 - (b) *Because responsibility for solving the problem is best left with the person who is responsible for solving the problem. Second guessing them is highly counter-productive.*
 - (c) *Because discussions of approaches to solutions can easily turn into long arguments.*
 - (d) *Because a code review is not an efficient format for brain-storming about how to solve problems.*

9/29/2012

Review Processes

51

Deeper

- The author gets defensive about his/her code?

The moderator should keep the review focused on objective issues and criteria, rather than on opinions about how things should be done.

The presence of management in reviews can increase defensiveness ... hence the recommendation that they be excluded.

If the author cannot help but be defensive, they do not have to be present for the review ... since they are not an active participant.

9/29/2012

Review Processes

52