Software Reviews – Good Practice

Stuart Reid

Introduction

Anything can be reviewed - and reviews can be applied at any stage in the life cycle. Reviews applied early in the life cycle detect defects early and so can help prevent large amounts of rework, arguably making them the most cost effective software engineering practice available to developers and testers[1]. After an overview, this paper focuses on review effectiveness before considering where to invest future effort.

Definition: Review: an evaluation of an artefact, which often has a number of complementary objectives (see below), one of which is to determine its fitness for purpose.

Objectives

The main objectives of performing reviews include:

- Find defects (ideally as early as possible)
- Measure quality (often to determine if entry & exit criteria of the development and testing process have been satisfied)
- Educate reviewers (in both formal and cultural) organizational standards and in the artefact under review
- Gain consensus on product (technical) and project decisions (e.g. resource allocation)
- Generate new ideas
- Motivate authors to improve their practices

Review Types

There are numerous review types described in the literature. IEEE 1028-2008 defines five types of software reviews and audits. These are, in this standard's order of formality - audits, management reviews, technical reviews, inspections, and, least formal, walkthroughs - although many test practitioners would argue that inspections are often more formal than technical reviews. Another term that is often used is 'peer review' (similar to the technical review), where the reviewers are typically colleagues on the project (they do not usually all have to perform the same role, such as business analysis or design, to qualify as 'peers'). Informal reviews are normally considered to be those performed by a single peer reviewer (sometimes with the author) where no formal documentation is produced as a result.

Some review types have specific attributes, such as inspections normally including process improvement and walkthroughs being led by the artefact author, but all follow the same basic process of:

- Planning
- Individual Reviewing (preparation)

[1] Defect removal effectiveness measures of up to 95% have been reported where inspections have been performed at all stages of the development and test process. In reality far lower levels are typically achieved, although these are still typically higher than those achievable by dynamic test execution and debugging.

- Meeting (and logging) – but see below on 'Effectiveness'
- Fixing and Reporting

In practice, it doesn't matter what the reviews are called - the attributes that are appropriate for the current situation are those that should be used. For instance, if 'walkthroughs' are nominally selected due to their ability to provide an education to the reviewers, it should not mean that because process improvement (and associated metrics) is not explicitly part of the traditional walkthrough approach that you should not include them in your review process.

Roles

Reviews involve a variety of different stakeholders, who may take on a number of different roles (not all the roles are used in all review types):

- Review Leader (called a Moderator in inspections) – takes overall responsibility for the review and decides who will be involved, organizes when and where it will take place, etc.
- Author – creates (and fixes) the artefact under review
- Reviewer – reviews the artefact, typically identifying issues with it
- Reader – reads aloud from the artefact in the review meeting
- Recorder/Scribe – records information at the review meeting
- Management – such as the project and programme managers, who may decide what is to be reviewed
- Customer – to provide their unique viewpoint
- User Representative – to explain what the users expect of the final product

Review Effectiveness

Meetings

Traditionally reviews have been centred around the review meeting, where defects are found, issues are raised, discussion performed and decisions are made and logged. In the last 15 years, the necessity of the review meeting has been considered and research performed to determine if it adds value, or not. The conclusion is that generally the review meeting does not increase the effectiveness of defect detection, typically the primary reason for holding the review. Arguments for including the review meeting are that it is a good forum for learning, sharing knowledge and gaining consensus and it is shown to reduce the number of false positives (issues raised by reviewers that are subsequently found to be 'features'). However, it does contribute a major cost to the review process and can often be difficult to organize. It also forces participants to review at the same pace and follow the same review sequence through the artefact, when it is known that the optimal pace differs for different reviewers and not everyone wants to review documents in the same order.

Where it is decided to hold a review meeting, it is the expertise of the individuals that decides the group performance (although the idea of synergy is attractive, research shows that it does not help in this situation) – so choose good reviewers. Although as each new reviewer is added the team's expertise *should* increase, at some point (typically with the fifth reviewer) the overhead of group interaction has a larger negative effect than the positive contribution of additional expertise. When the group needs to make decisions, if the group can identify the best member of the group then using that individual to make group decisions will normally be the most effective approach (giving equal weight to all reviewers' input will results in worse decision making).

Reviewer numbers

If more reviewers are involved in the individual reviewing activity then more defects will be found, and reviewer effectiveness at defect detection is largely dependent on their expertise (typical reviewers will find about one in three defects). If reviewers with high levels of expertise are available then two reviewers is often optimal. In terms of cost effectiveness it has been shown that pair reviews (an author and a reviewer – also known as a 'peer deskcheck') are most effective in some situations.

Whenever more than one reviewer is chosen, care should be taken to focus the reviewers on distinct aspects of the artefact under review as otherwise duplication of effort reduces cost effectiveness.

Multiple reviews

It is not cost effective to plan multiple formal reviews on a single artefact. It is, however, often useful to ensure that review artefacts are in a reviewable state by performing an informal review of the artefact ahead of the more expensive formal review.

Individual preparation

Given that the inclusion of a review meeting often adds little or no value to the overall review process, it is not in the meetings where improvements can best be made. Instead, in practice, it is in review preparation (where defects are detected using reading techniques) where improvements can most efficiently be made (see 'Reading Techniques').

Reading Techniques

Ad hoc

The traditional approach to defect detection by reviewers is completely unstructured; each reviewer is expected to find as many defects as possible of any type. This approach is highly-dependent on reviewer skills and leads to the same issues being identified by the numerous reviewers.

Checklist-based

A more systematic approach to identifying defects is based on defect checklists – if different reviewers are assigned different checklists then this ensures wider coverage overall and helps prevent the duplication inherent in the ad hoc approach. One danger of checklists is that some reviewers mistakenly limit themselves to only considering the checklist entries and ignore other potential issues with the artefact under review, so care should be taken to ensure reviewers are made aware that they have a wider responsibility than simply following the checklist.

Typically review checklists take the form of a set of questions based on potential defects, which may derived from experience within the project, the organization or across the industry as a whole. Checklists should be specific to the artefact under review, so a checklist for a requirements document will be different to one for a design document or a test plan, and may be specific to the methodology used to develop the artefact (e.g. may have different checklist questions for requirements in the form of plain text to those in the form of use cases or user stories). Checklists may also be specific to the application domain of the artefact, e. g. a checklist for a banking artefact may be based on banking regulations while a checklist for an avionics artefact would be based on avionics standards).

Typical problems with checklists are that they are too long and never change. The ideal checklist should be constrained to about 10 entries and regularly updated – as entries become stale and find

fewer issues (hopefully the authors have learned and improved) then they should be replaced with newer entries reflecting issues missed in the recent past. It is possible to enhance the checklist-based approach by using risk information to ensure that those defects that could have the highest impact on the business are included in the checklists and so explicitly checked during the reviews.

Scenario-based

Where requirements, designs (or tests) are documented in a suitable format (e.g. use cases) then a scenario-based approach to defect detection may be the most appropriate. With this approach the reviewers perform 'dry runs' on the artefact to check whether the correct functionality is described and typical error conditions are handled suitably. There is, of course, the danger that these reviews will be constrained to the documented scenarios and will miss defects of omission where required functionality is not included in the artefact under review.

As with the checklist-based approach, it is possible to enhance the scenario-based approach with risk information to ensure that the most important scenarios to the business in addition to the most used scenarios are reviewed in more depth.

Perspective-Based Reading/Reviews (PBR)

According to the available research, the most generally effective (and efficient) form of defect detection for reviews is perspective-based reading. Some may know this as role-based reviewing, and it uses the idea that different reviewers take on different stakeholder viewpoints and review the artefact from that stakeholder's perspective. The idea is that if all stakeholders are happy with an artefact it should be of high quality, and it means that each reviewer can look into their stakeholder's view in more depth with less duplication of effort between reviewers.

Typical stakeholder viewpoints used in PBR are:

- User
- Business Analyst
- Designer
- Tester
- Operations
- Maintainer
- Regulator

It is important that the correct balance of viewpoints is included in the review. For instance, if reviewing a requirements document, then the user, designer and tester viewpoints would normally be the most important to cover. If a system is being built within a highly-regulated area then the regulator viewpoint should be included and if the system is to be long-lived, then the maintainer viewpoint becomes more important.

Not all reviewers can easily 'jump' into a new role and so PBR scenarios are used to make this approach more accessible. These scenarios comprise three parts. The first describes the stakeholder view that the reviewer should take for the review. The second part describes the high level product that the stakeholder would be expected to develop from the artefact under review (e.g. a tester viewpoint may well be expected to develop an acceptance test plan based on the requirements specification). In PBR the reviewer is often expected to create a first draft of this product to 'test' whether this is possible from the information provided in the artefact under review (these first drafts may well form the basis of subsequent development and testing). The third part of the PBR scenario typically comprises a checklist of questions specific to the high-level product

developed in the previous part. PBR scenarios are specific to the artefact under review (e.g. a designer PBR scenario for a requirements specification), but once created should be updated as appropriate to keep them useful (e.g. updating questions in part 3) and reused as needed.

Review Measurement & Improvement

Although the main objective of reviews is often perceived to be detecting defects, in the longer term the underlying objective should be to improve the development and test practices. To support both objectives review metrics need to be collected and the following are the most commonly used.

Defect density

This is a measure of the quality of the artefact being reviewed, and will vary widely between organizations, but after sufficient time acceptable benchmarks for the organization (or project) will be understood. A typical measure of defect density is the number of defects per page (or per function point if these are used). Note that this, and later measures also need to define what severity of defect is included, as otherwise high severity defects may be masked by cosmetic defects.

Defect introduction

This is a measure of the quality of the development process for the particular phase of the life cycle to which it applies. A typical measure of defect introduction is the number of defects introduced per page/function point.

Defect detection

This is a measure of the quality of the testing process (including reviews) for the particular phase of the life cycle to which it applies. Typical measures of defect detection are the number of defects detected per review hour or the percentage of defects detected by review in a particular phase. Note that defect detection effectiveness for reviews will decrease as authors learn to make fewer mistakes as a result of the review process (this is a good thing!).

Review rate

This is a measure of the speed of the review process for the particular artefact under review. A typical measure of review rate is the number of pages/function points per hour. As with defect density this will vary widely between organizations and review process used, but after sufficient time acceptable benchmarks for the organization (or project) will be understood.

Sampling

Reviewing is an expensive activity and in some situations (e.g. where the objective is to measure quality) reviewing everything (e.g. all 500 pages of a requirements document, or 1,000 pages of test scripts) is not economically viable. In such situations, sampling is often the only cost effective reviewing approach and some of the above measures, such as defect density, can be used to predict the 'quality' of non-reviewed artefacts. Where possible, it is often extremely useful to review early samples of large important project documents, such as requirements specifications, to provide an early view of the likely quality of the final product (if it is allowed to proceed as it is now).

Review Tools

Reviews are a labour-intensive process; however, they provide an excellent opportunity to leverage tool support, especially when the review process does not include face-to-face meetings. Tools are good for managing the version control and progress of artefacts through the review process, logging

and recording associated comments, as well as the collection of review metrics. These tools can ensure that the traceability between review artefacts, comments and changes is maintained and automatically generate reports. There are a number of tools supporting code inspections, both commercial and open source, and some of these also support document reviews.

The Introduction and Management of Reviews

The introduction of reviews into an organization is a complex exercise, as is any major change, but most organizations are already aware of the effectiveness of reviews and they are included in their documented development and test processes. In many organizations, however, they are not performed as specified and the documented processes describe a review process that is little changed from those first at IBM in the mid-1970s. This is often because it is unclear who is responsible for them, and no continuous improvement initiative was built into the review process. In such situations, the current review process needs to be overhauled and brought in line with current best practice and responsibility for the new process assigned to the appropriate body. Overall responsibility for the review process is typically assigned to the QA/QM or Testing function within the organization, but often responsibility for the review of individual artefacts is distributed to individual authors, which can make it difficult to ensure an optimum sequence of reviews is undertaken and to manage and collate process improvement metrics.

Where the review process is changed or has fallen into disuse, training is normally required to educate review stakeholders in the process and their responsibilities. Many organizations find that a lack of initial training and buy-in by review stakeholders is mitigated by the use of kick-off meetings, which serve to continuously remind reviewers of their responsibilities and the focus for the current review. Typically, however, this is not a cost-effective approach, as the cost of kick-off meeting for each artefact soon outweighs the cost of training (and associated buy-in) that covers the whole review process.