Software Reviews using ISO/IEC 20246

Dr Stuart Reid

Introduction

Reviews have long been acknowledged as potentially the most effective approach to ensuring software quality. But are you aware of the options available to make your reviews more efficient and cost effective? Did you know that a new international reviews standard was published in February 2017? This article introduces software reviews and gives an overview of the new ISO/IEC 20246 standard [1].

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.

Most, if not all, safety-related standards require reviews to be performed, but none of them explain how. For instance, ISO 26262-6, 2011 for software developed for production road vehicles [2] states that walkthroughs must be performed for the lowest risk-level software (ASIL A), and that inspections must be performed for the more safety-critical software (ASILs B to D). However, ISO 26262 [2] gives no guidance on how to implement them in practice, meaning that each supplier and OEM can select quite different ways of meeting these critical requirements. The new ISO/IEC 20246 standard on work product reviews [1] fills this gap by defining a generic review process, while also showing how this process can be tailored to satisfy the requirements of any other standards that require specific review types to be performed.

Review Objectives

Reviews are performed for a variety of reasons, but the main objectives include:

- Finding defects (ideally as early as possible in the life cycle to reduce any necessary rework)
- Measuring quality (often to determine if entry & exit criteria of the development and testing process have been satisfied)
- Educating reviewers (in both formal and cultural organizational standards, and in the artefact under review)
- Gaining consensus on technical issues and project decisions (e.g. resource allocation)
- Generating new ideas
- Motivating authors to improve their practices
- Reducing risks and the overall time taken to deliver software

In practice, most reviews are performed to address several of these potential objectives.

Review Types

There are many different types of review described in software engineering textbooks and mandated by various standards. Most, if not all, safety-related standards for software require the use of different types of review (e.g. inspections, walkthroughs, code reviews), but these standards do not describe the differences between the different review types, and assume that practitioners know how to best perform the different types of reviews they mandate. Practitioners could refer to textbooks that describe reviews, but these typically disagree on the details of how they are performed (and commercial drivers could encourage practitioners to choose the textbook description that is the cheapest to implement but that still allows them to state they have performed the required type of review).

Since February 2017 there is now an international standard that formally defines the different review types - ISO/IEC 20246 [1] defines ten review types, as follows:

- Milestone Reviews
- Inspections
- Technical Reviews
- Peer Reviews
- Walkthroughs
- Informal Group Reviews
- Pair Review
- Peer Desk Check
- Buddy Check
- Author Check

The above review types are shown in order of rigour from most formal to least formal (the informal review types towards the bottom of the list normally require no formal documentation of the review).

IEEE 1028-2008 [3], an older reviews standard, 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. As can be seen, there is some overlap between IEEE 1028 and ISO/IEC 20246. However, one notable difference between the standards is that IEEE 1028 defines only four (formal) review types and does not allow any deviation from their formal descriptions, whereas ISO/IEC 20246 defines nine formal and informal review types, but also allows users to tailor the generic test process to fit their specific needs. IEEE 1028 also describes inspections as one of the least formal review types, when many test practitioners would argue that inspections are normally the most formal review type.

A Generic Review Process

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 generic review process, as shown in figure 1. Each of the activities within the generic review process are briefly described in the following subsections.

Planning

In this first step of the generic review process, the scope of the review is agreed and specific review characteristics, such as the roles of the review participants (e.g. facilitator, review leader, author, technical lead, customer, etc.) and the use of checklists are decided. During the planning, the people performing the review are selected, agreed and assigned to specific roles.

Initiate Review

When initiating the review, the review materials are distributed to the review participants (as early as possible), and review training may be arranged, if it is considered necessary (for reviews to work well all the participants need to be trained and know their responsibilities).

If the review is more formal, then an overview meeting may be arranged to explain the scope and individual roles and responsibilities to the reviewers.

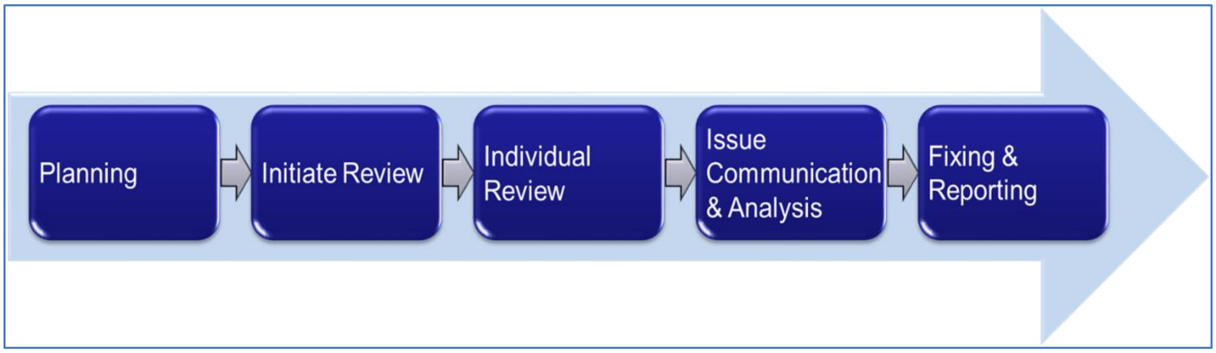

Figure 1: ISO/IEC 20246 Generic Review Process

Individual Review

This step, where individual reviewers work alone, reviewing the work product, is where the clear majority of issues are detected. Identified issues are normally documented in an issue log (which may be supported by a review support tool). According to ISO/IEC 20246, this step is optional, and may not be performed for some types of reviews, such as walkthroughs and informal group reviews, however many (including this author) believe that this is the most important activity in the review process.

Due to the importance of this step, techniques for performing the individual review are described in more detail later in the article.

Issue Communication and Analysis

If there is not going to be a review meeting, then identified issues are communicated to the individual performing the analysis of issues found. Otherwise, the identified issues are reported directly to the review meeting participants, or (probably better) they are collated ahead of the review meeting to help the review meeting go more smoothly.

The issues are analysed, and a decision is made on what to do with them. This analysis and decision-making can be performed by a group of reviewers at a review meeting or by an individual (often the author) without a review meeting.

The result of the issue analysis is typically one of the following:

- ‘issue rejected’
- ‘issue recorded but no action taken’

- ‘issue to be addressed by the product author’
- ‘issue updated due to analysis’
- ‘issue assigned to an external stakeholder’

Based on their status, issues are subsequently assigned to the author of the work product, or another individual (e.g. the author of a specification document).

Next, a decision is made on what to do with the work product. Typically, the work product will be updated based on the identified issues and then released for use. Sometimes it may be necessary to re-review the work product due to the number and scope of the changed needed. Very rarely, the work product may simply be rejected.

Fixing and Reporting

In this final step, we would normally expect the author to be making changes to the work product based on the agreed issues. Some issues will be with other artefacts and these need to be addressed by their respective owners (e.g. the QA department if an issue is found with a company standard).

The changes are confirmed as being completed and then a report on the review is generated.

Review Attributes

For many organizations and projects, the decision to perform reviews is not driven by mandated standards, but, instead, the decision is a pragmatic, commercial one. We know that reviews are probably the most cost-effective means of improving product quality, and so that is the reason we perform them. In these situations, where we do not need to use a particular type of review mandated by a standard, we should select the review attributes that are most appropriate for our unique situation. 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.

ISO/IEC 20246 [1] encourages this flexible approach to reviews and defines and describes the following different review attributes that may be selected to tailor the generic review process, shown in figure 1 and described earlier. Some of these attributes are described in more detail below.

- Purpose (mentioned earlier)
- Roles (see below for more detail)
- Individual review techniques
- Optional activities (see below for more detail)
- Number of reviewers (see below for more detail)
- Planned number of reviews (see below for more detail)
- Work product type
- Work product format
- Formal reporting
- Training required

- Review improvement
- Entry and exit criteria

Roles

A variety of different stakeholders take part in reviews, and these stakeholders may take on several different roles (not all the roles are used in all review types):

- Review Leader – 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
- Facilitator — facilitates the review meetings (called a Moderator in inspections)
- Reader – reads aloud from the artefact in the review meeting
- Recorder/Scribe – records information at the review meeting
- Manager – such as the project manager, who may decide what is to be reviewed
- Customer – to provide their unique viewpoint
- User – to explain what the users expect of the final product

Optional Activities – Review Meetings

Traditionally reviews have been centred around the review meeting, where defects are found, issues are raised, discussion take place and decisions are made and logged. In the last 15 years, the necessity of the review meeting has been considered and much 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, which is often the primary reason for holding the review. Arguments for holding a review meeting are that it is a good forum for learning, sharing knowledge and gaining consensus, and it is also shown to reduce the number of false positives (issues raised by reviewers that are subsequently found to be valid features).

However, the review meeting is a major contributor to the cost of reviews and can often be difficult to organize. It also forces participants to review at the same pace during the meeting and to follow the same review sequence through the artefact, when it is known that the optimal reviewing pace differs for different reviewers and not everyone wants to review documents in the same order.

When a review meeting is held, it is the expertise of the individuals that decides the group performance (although the idea of synergy between review members is an attractive idea, research shows that it rarely helps in this situation) – so it is a good idea to choose expert reviewers.

Logic would suggest that as each new reviewer is added to the review, the team's expertise *should* increase, however, at some point (typically with the fifth reviewer) the overhead of group interaction has a larger negative effect than the positive contribution of the additional expertise.

When the group needs to make decisions, if the group can identify its best member, then using that individual to make group decisions will normally be the most effective approach (as giving equal weight to all the reviewers' inputs will normally result in worse decision making). This is a situation where democracy is not always the best option.

Number of Reviewers

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

Whenever multiple reviewers are chosen, care should be taken to focus the different reviewers on distinct aspects of the artefact under review, as otherwise duplication of effort reduces cost effectiveness.

Planned Number of Reviews

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

Individual Reviews – Techniques

Given that the inclusion of a review meeting often adds little or no value to the overall review process, it is not in the review meetings where improvements can most efficiently be made. Instead, in practice, it is in in the individual review activity (where issues are identified by individuals working alone) where improvements can be made most efficiently. ISO/IEC 20246 [1] defines several techniques for individual reviewing, as described below.

Ad hoc Reviewing

The traditional approach to defect detection by individual 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 skill levels and normally leads to the same (most obvious) issues being identified by multiple reviewers.

Checklist-based Reviewing

A more systematic approach to identifying defects is based on defect checklists. Different reviewers should be assigned different checklists as 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 entries on their checklist 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 their own checklist.

Typically, review checklists take the form of a set of questions based on potential defects, which may be 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 even be specific to the methodology used to develop the artefact (e.g. we 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 automotive artefact would be based on ISO 26262 [2].

A typical problem with checklists is that over time the checklists become too long and never change. The ideal checklist should be constrained to about 10 entries and regularly updated. As checklist entries become stale and fewer issues are found (hopefully because the authors have learned and improved) then they should be replaced with newer entries reflecting issues missed in the recent past.

It is also possible to enhance the checklist-based approach by using risk information to ensure that those defects that would have the highest impact on the business are included in the checklists and so are explicitly checked during the reviews.

Scenario-based Reviewing

Where requirements, designs (or tests) are documented in a suitable format (e.g. use cases), then a scenario-based approach to reviewing 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 and the most used scenarios are reviewed in more depth.

Perspective-Based Reading (PBR)

According to the available research, the most generally effective (and efficient) form of defect detection for reviews is perspective-based reading. You 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 by considering all the different stakeholders points of view we will identify more potential issues, and it also means that each stakeholder’s view is reviewed in more depth with less duplication of effort between reviewers.

Typical stakeholder viewpoints used in perspective-based reading are:

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

- Regulator

It is important that a reasonable selection 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 (as the user's requirements are being described, and both the designer and tester are primary users of the requirements). If a system is being built within a highly-regulated area, such as financial or safety-related, then the regulator viewpoint should be included and if the system is to be long-lived, then the maintainer viewpoint becomes more important.

PBRs are not just about the reviewers taking different viewpoints. They also expect reviewers 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). They also require the use of checklists, so are an approach that builds on some of the other approaches.

Not all reviewers can easily 'jump' into taking on a new role for the review and so PBR scenarios are used to make this easier. These PBR scenarios comprise three parts:

- The first part describes the stakeholder view that the reviewer should take for the review (e.g. the reviewer will 'pretend' to be a financial regulator).
- 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).
- 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 role and the artefact under review (e.g. a 'designer' PBR scenario for a 'requirements specification'). They should be updated over time to keep them useful (e.g. updating checklist questions in the third part of the PBR scenario) and reused as needed.

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 should already be aware of the effectiveness of reviews and reviews are often included in many organizations' documented development and test processes. In many organizations, however, they are not performed as specified, and the documented review process describes a process that is little changed from those first used at IBM in the mid-1970s. This is often because it is not clear who is responsible for them, and continuous improvement was not built into the review process. In such situations, the old review process needs to be updated and brought in line with current best practice and the responsibility for the new process must be 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 review best practice is followed and to manage and collate process improvement metrics.

Conclusion

Reviews should be an integral part of all software development and test processes – we know they are probably the most effective means of improving quality that is available to us. However, reviews are not always performed in an effective manner, even when they are mandated by standards, such as ISO 26262 [2]. Organizations may start with a well-defined review process, but, over time, and as development practices have changed, parts of the process can be forgotten or misused, or some projects may even decide that they can do better without a review process at all.

In February 2017, a new international standard on reviews was published – ISO/IEC 20246 – this standard defines a generic review process that is applicable to any type of project and can be used for any artefact at any stage of the life cycle.

The publication of ISO/IEC 20246 should remind us of the universal benefits of reviews. It should be used as the basis of the review process for projects where reviews are mandated by other standards, such as in the financial and safety-related fields. However, we know it is cost-effective for all projects to use reviews, and so it should also be used by any projects where quality and costs are important – it should be used on all projects.

References

[1] ISO/IEC 20246, Software and systems engineering — Work product reviews, 2017

[2] ISO 26262-6, Road vehicles — Functional safety — Part 6: Product development at the software level, 2011

[3] IEEE 1028, IEEE Standard for Software Reviews and Audits, 2008