Three Stage Formal Technical Reviews

C. C. Shelley

Oxford Software Engineering
9 Spinners Court, 53 West End, Witney, Oxfordshire, England, OX28 1NH
shelley@osel.netkonect.co.uk

ABSTRACT

Technical reviews of software and software documentation are well known as the most effective quality control. However the application of rigorous quality control can have a demotivating effect to those whose software and documents are being reviewed, and this can ultimately undermine the review process itself. A staged approach to reviews that builds the confidence of those producing items and those that will use them, and that recognizes the positive points as well as the defects in the reviewed items is described.

1. Introduction

Quality controls performed early in the software project lifecycle, and early in the development of each of the project's software artefacts, both documents and code, are one of the most valuable development activities. As well as being a defect detection method and way of sharing ideas they are one of the most effective means of 'instrumenting' development work to provide both management and development process information. However despite these advantages there are problems. The scrutiny of technical artefacts if managed badly, not from a technical perspective, but from a human perspective, can work against the introduction or continued use of reviews and inspections. These artefacts are the intellectual products of people who have invested time and effort in their development. Technically sound reviews can, if mismanaged or poorly timed, be seen as intrusive, obstructive and demotivating. However when well timed and correctly scoped they can fulfil their potential.

A staged approach to Formal Technical Reviews (FTRs), and Inspections can improve the acceptability of these quality controls, as well as encouraging very early defect detection.

2. Formal Technical Reviews

Formal[1] Technical Reviews[2] and Inspections of documents or software are performed to identify and remove defects. They are distinct from management reviews that take a broader view of the performance of technical work: its conduct so far, and the need, prospects and risks of future work; they are reviews of *context.* Technical reviews evaluate an artefact's quality and fitness for purpose; they are reviews of *content*. The two should not be confused or combined.

Performed early in the software development lifecycle FTRs can remove potentially expensive requirements or design defects. Performed later, on code, they can indicate the defect densities to be expected during testing and help shape testing strategies. An advantage of technical reviews performed on code is that they reveal defects directly. Testing often only reveals the symptoms; the defect needing to be found by 'debugging'.

As well as acting as quality controls and communications mechanisms FTRs also provide surprising amounts of management information:

- When an FTR is completed it signals the completion of the work undertaken to produce the reviewed item.

[1] Formal means the review is planned and scheduled, the reviewers use checklists during the review, and the outcome of the review is a binary decision to accept the item (albeit with some rework) or reject it, and there is a record of the review.

[2] As described in IEEE 1028 as technical reviews when these are performed as a quality control; effectively 'inspection-lite'.

- And the availability of the item for use. (FTR's should only be performed on completed items).

- Software development plans will indicate the estimated effort needed to produce the reviewed item, With the review completed (and the item accepted) the actual work to produce the item can be recorded: the cost deviation (the difference between the planned effort and the actual effort) can be calculated.

- And the 'earned value' of the project will increase by the planned effort to produce the item.

This is a large amount of management information produced as a side effect of a simple quality control activity.

In addition 'process' information can be collected during the review: numbers of defects, their types and modes, when they were introduced (sometimes), and, implicit in the timing of the review, when they were detected. This too is valuable information for improving understanding of the software development process.

There can however be problems. The effectiveness of FTRs and Inspections as quality controls can work against them. The items submitted for review are often complex and novel technical artefacts that have received a considerable investment of intellectual effort. These artefacts are to be used by software developers or other technical staff later in the development process. From both the authors' and the users' perspective the value of these artefacts resides in their usefulness. While the quality of the items is undoubtedly increased by the removal of defects, the increase in usefulness can be moot. FTRs' focus on quality control, finding defects, can be seen as having limited value if users perceive that the item can fulfil its purpose of informing and setting technical direction, by acting as a jumping off point for future technical work, as a 'working draft', even with some defects[3].

Perhaps more importantly the review process can be an emotionally bruising affair for the author who sees 'their' item criticized, and by implication themselves. This is made worse because the criticism is objective and constructive – there are no emotional 'work-arounds' – it really was full of defects.

The value of these quality controls can be further undermined by review and Inspection enthusiasts who can identify extraordinary numbers of defects by flagging the almost inevitable ambiguities inherent in the use of natural language[4]. This legitimate defect detection can be seen as academic 'nit picking' and work against the acceptance of reviews as a valued technical practice. Over time reviews can be seen as an intrusive drain on resources that add little real value to the items reviewed but implicitly criticize the authors. More informal walkthroughs and design reviews, the circulation of drafts for constructive comment by peers, and iterative development can seem more useful, quicker and altogether more attractive.

If early quality control - FTRs and Inspections - are to be valued and used they must be a) effective and b) acceptable. To be effective the quality control should be formally managed, distinct from other collaborative working or consensus building activity and focus on finding defects. To be acceptable they must be perceived as supporting the development of complex technical artefacts and not seen as criticising or emabarrassing the artefact's authors. Reviews that fail to meet both these needs will be devalued or disliked and fall out of use It is necessary to find a formal review process that is both effective and perceived as acceptable and useful. Three stage formal technical reviews meet this need.

3. Staged Reviews

Staged reviews are not novel. They are well established and in use by many developers. Iterative development of artefacts (not to be confused with iterative software development lifecycles) is a well known precursor to staged reviews: documents are drafted, circulated for comment, refined and circulated again until the document converges on an acceptable draft. This is a constructive and collaborative process that has one of the secondary benefits of FTRs: shared understanding. But it is not formal quality control, although this may be an aspect of this activity. Staged reviews of increasing rigour build on this iterative development. It is well known in the more sophisticated software development organizations: A draft is circulated for informal review and comment by peers - desk-checking say. This is followed be further refinement and a more formal review, perhaps a walkthrough. When the item has been further developed a formal

[3] Which of course the users will expect to pick up and correct later when they use the item.

[4] These defect numbers can be reduced by structuring natural language. Tables and matrices are one of the more effective means for structuring this type of information: the tables revealing what is not known as well as what is in the case of sparsely populated tables.

technical review or inspection is conducted. At each stage the quality of the item is improved.

This type of approach has been well described by Michielson, 2005 (1).

Fig. 1 Staged Reviews

Here four levels of review take place giving an increasing level of product quality - as perceived by a balancing of three contexts - the '3C's:

- Craftsmanship – engineer's perspective;
- Conformance – standards' perspective;
- Customer Satisfaction – customer perspective;

each of which needs little justification. The increase in quality is not perceived solely as a defect detection and counting exercise. Typically the focus of these staged reviews is on design considerations in the early stages with the emphasis shifting to the identification and removal of defects (quality control) towards the final review. Many of the benefits of shared understanding and quality control are achieved.

However some disadvantages remain. The development of the item is not directed. It may be developed increment by increment and each increment reviewed in isolation, or not reviewed until every increment is complete, or it may evolve as a whole. The underlying soundness of the item may not be tested as reviewers, at the early stages, strive to understand the item rather than scrutinize its properties, and will tend to encourage the producer of the item rather than criticize an undeveloped product by identifying defects. This early informality militates against the consistent use of rules and checklists (standards) either for guidance or conformance. And the informality can mean that the earlier reviews may happen either too early when there is little to comment upon, or late in the development of an item, with the first review taking place when the originator believes it is complete, or they may not happen at all.

4. Staged Formal Technical Reviews

Staged FTRs, occasionally encountered in the engineering and defence sectors, eliminate these problems while retaining the advantages of iterative development and informal staged reviews. In addition they provide guidance on the development of complex or novel technical artefacts and allow quality control from the very beginning, rather than delaying it until the author believes the document is complete, then identifying large numbers of defects, and perhaps major defects that may require large amounts of rework.

The critical distinction is that staged *Formal* Technical Reviews have the same level of formality at each stage, but the rules and checklists vary. This is in contrast to most staged reviewed where the formality increases at each stage but the rules and checklists – so far as they are used at all in the early stages – remaining the same.

The three stages are:

- Outline review;
- Draft review or walkthrough;
- Final review or inspection.

Each of these reviews is formal in the sense described earlier. This formality and the application of checklists ensures that the effort to produce item is directed well, fundamental misunderstandings are identified very early, accountable decisions to proceed are made and recorded, and peers and managers have visibility of the work prior to its delivery for final review and approval.

The first outline review ensures that the right artefact is to be produced and it will be fit for its intended purpose. It ensures that its requirements are known, understood and complete and that the proposed structure will result in an item that fully meets the requirements. To do this the structure of the artefact is presented at the outline review together with statement of its intended purpose and scope. For technical documents the outline could be a complete, detailed and annotated table of contents – with its supporting purpose and scope statements. Where the item is code then the outline may include programme specifications and outlines and (if they exist) pointers to tests. The material submitted for review should be well presented with all technical problems solved - perhaps by collaborative working or iterative development - but that collaborative

design work does not happen in the review process. The material submitted should not be thought of as preliminary notes or a rough draft, but as a well considered structure and context for further development and refinement.

At the conclusion of the outline review the originator and reviewers will be confident (or not, of course) that the requirements are well understood and that the overall structure of the item and the approach to developing it are correct.

Work can then proceed to produce a complete draft based upon the agreed outline. When this is nearly complete a draft review is scheduled. The draft review is conducted to assess its 'goodness', ensure that the artefact conforms to the structure agreed at the outline review, that the material is correct, complete and meets the requirements. It also ensures that the artefact is coherent, and complete and that it is consistent both internally and externally. If standards are used then they may, but not necessarily, have been applied by this stage. Any issues raised at the outline review will have been addressed at this stage. Where the item is software it should be complete, adhere to coding standards, compile correctly and may have demonstrated functionality, but not have been tested.

During the draft review it is permissible to consider design options, to explore the opportunities being presented by the artefact and to recognize and record exceptional aspects, novel design solutions, elegant refinements - 'profects[5]'. This recognizes that the development of design artefacts - documents or code - is a creative act, not just a mechanical production process, the draft review acknowledges this by not being limited solely to the identification of defects and issues like a productionl quality control process. A structured 'walkthrough' may be appropriate here. When the draft review is completed the originator and the reviewers will be confident that the development of a technically sound, potentially excellent, item is well under way. The reviewers will have had insights into the nature of the item being produced and its likely quality when presented for final review or inspection.

For items of modest size or standard structure and format the outline and draft review may be combined.

Work then proceeds to develop the draft to issue status and release quality. This work should be a matter of refinement and elaboration by the author/originator. The final technical review or inspection applies checklists to ensure that the product is entirely acceptable. It:

- is concise, complete, and correct;
- is unambiguous[6];
- meets its requirements;
- complies with applicable standards;
- will create a favourable impression upon its recipients;
- and is *now* error free.

It is reasonable to expect that an item submitted for final review will be error free. With an acceptable draft already available as the basis for the final product the final work can focus on removing all remaining defects. It should be expected that the product will be of high quality with no trivial errors, spelling mistakes, or elementary formatting errors. If the devil is in the detail then prior reviews of the outline and draft enables the devil to be engaged during this final stage of production. If the work is unusually difficult, novel or complex and the final review does uncover defects then the prior reviews should have eliminated most of the major defects and limited rework only should be required, not a rejection and extensive rework and rescheduling of the review. The development schedule should not be placed at risk and the originator should not feel too abused by the review process – unlike when the final review is the only formal review where the first stringent application of review checklists can reveal devastating numbers of defects and misunderstandings, resulting in the rejection of the artefact, rescheduling and the demoralization of the originator.

[5] Profects are the opposite of defects; those bits of software design that surprise and please: incidents of new design excellence and microinvention that cause ripples in the design aether. Consider establishing a software 'coolhunter' role in reviews to detect these fragments of design serendipity with a view to exploiting them more widely, and finding the source of these innovations.

[6] Natural language is necessarily ambiguous. It is possible to identify more ambiguities than words when undertaking an inspection. Some of these ambiguities may reveal real issues or defects, most will not. Some inspections processes, following Gilb, 1993 (2), will require the recording of these large numbers of defects. Others filter out the trivial ambiguities only recording what are believed to be real defects. One approach risks jeopardizing the credibility of the inspection process by claiming very high defect numbers, the other risks letting real major defects through the quality control filter by attempting to detect only 'real defects'.

5. Checklists and Rules

The success of staged reviews depends on concise, meaningful checklists supported or justified by agreed rules. It is unreasonable to check artefacts against rules, however sound, without communicating these to the authors or originators of artefacts by means of these checklists.

Rules, described by Gilb (2), specify the content of an artefact, its format and relationships with other artefacts. Checklists are derived from these rules and are a fundamental part of FTRs and Inspections. They are usually stated as questions about an artefact, a negative answer indicating a defect or issue. For example a rule for the format of an artefact may state that 'every element of the artefact is uniquely identified'. The derived checklist item may state 'is every paragraph and figure in the SRS sequentially numbered and listed in the table of contents?'

Rules (and corresponding checklists) should cover three principal areas: The content of the artefact, the structure of the artefact, and the relationship of its contents to predecessor documents – traceability.

Content related checklists may consider

a) Clarity – is it clear and unambiguous?

b) Level of detail required – is it sufficient, but no more?

c) Technical feasibility – can it be done, is it the best way?

d) Verifiability – can it be tested?

e) Standards - does the content conform to applicable product standards?

The Structure related checklists may consider:

a) Internally consistency – are there any contradictions?

b) Uniqueness – is there repetition or unnecessary redundancy?

c) Organization – is the material is appropriately structured[7]?

d) Identity - is the artefact uniquely identified?

e) Format – does the artefact conform to applicable documentation templates and standards?

Traceability checklists should address:

a) Completeness – is every requirement of predecessor documents addressed, is it sufficient?

b) Externally consistent – is each requirement of predecessor artefacts addressed or interpreted correctly?

c) Necessary – does every element of the artefact addresses a requirement of predecessor documents (the complement to a) above).

For outline reviews checklists addressing the content elements a) and b) should be applied to the purpose and scoping statements; the structure related elements c), d), and e) applied to the document outline as a whole; and possibly the traceability element a) where this can be evaluated.

For a draft review where the document is substantially complete and correct the following additional elements should be included in the checklists used:

Content: c), d), and perhaps e). Structure: a), b), c). Traceability b).

For a final review all elements should be addressed, together with a conscious appreciation of the artefacts overall excellence – not readily captured in any checklist.

6. Costs and Benefits

A legitimate concern of those not familiar with three stage formal reviews is the increase in cost: review costs are increased three fold. This is to make the same error as those concerned with the cost of conventional technical reviews. These activities are not in addition to existing development activities, they are a refinement of them. During the course of the development of an artefact its development may be checked, perhaps by informal reviews, or staged reviews, or by means of iterative development, each of which have costs associated with them. Or perhaps the artefact is not checked until the final formal review and then the project risks accruing costs due to rework and delay.

[7] A non trivial issue that could be considered as a content related matter. The organization of material can have a major impact on the way subsequent work is performed and on the design of subsequent artefacts. The organization of material in, for example, an SRS is a design issue in its own right. See IEEE 830 for a range of examples of ways of organizing the content of an artefact.

Formal staged reviews cost no more than less formal checking during development – and can cost less if well managed. They cost less than the cost of not reviewing until the artefact is believed complete.

The benefits include those of other forms of staged reviews – acceptable scrutiny of artefacts. Acceptable because it is early, before too much intellectual investment has been made. And early scrutiny providing the opportunity to correct misunderstanding and error, and sharing of design ideas. In addition staged formal reviews are acceptable because the scrutiny is fact based – enabled by checklists – not just opinions. And the checklists provide support and direction for the development of the artefacts through each of their stages: support not provided by any other approach to reviews. Finally the formality of the reviews provides a degree of predictability and visibility not always provided by other means.

7. Approval

It is customary to approve a document that has been successfully reviewed. The review provides confidence that the document's quality is sufficient and the approver, a decision maker or manager, uses the record of a successful review to make the decision to release the document for use, by placing the document in a repository, perhaps as part of a baseline, or, less common now, by physically signing the document and distributing copies. The review is a technical process and the approval is a management or control process. Occasionally the role of the review and the approval are combined; the successful review acting, by default as the approval as well, but the distinction should be maintained. If it is not the approval can become another faux-technical review by decision makers or management: this is a symptom of poor reviewing; where there is insufficient confidence in the review process or the reviewers.

For formal staged reviews approval is required for each stage. The successful review of an outline provides confidence that the outline is good. A decision is required to determine whether the development of the outline of the draft, from the outline is to begin, and similarly for the draft. The approval is a necessary control point, whether granted automatically upon successful completion of the review, or a distinct 'sign off' approval.

8. Closing Remarks

Three stage reviews have the potential to act as an aid to the development of software artefacts, and to be perceived as such, as well as providing very early defect detection. To realize these benefits clear rules and checklists are required to set expectations and establish agreed criteria for these quality controls.

Where a reasonably well defined development process and stable software artefacts definitions are in place the three stage formal review process, can also act as an effective framework for consistent, low risk artefact design.

9. References

1. Cees Michielson, 'Can You Afford to be Good Enough?'. Gilb Symposium, London 2005.

2. Gilb, T. & Graham, D., 'Software Inspection', pub. Addison Wesley, 1993, ISBN 0-201-63181-4

