




1	 <h2 style="text-align: center;">Software Quality Reviews</h2> <h3 style="text-align: center;">Informal Walkthrough</h3>	
2	 <h2 style="text-align: center;">Informal Walkthrough</h2> <div style="display: flex; align-items: center;">  <div style="border: 1px solid black; padding: 5px; background-color: #ffffcc;"> <p style="text-align: center; margin: 0;"><b>Informal Walkthrough</b></p> <p style="margin: 0;">A review process in which the author of a work product leads one or members of the project team through the work product, while other members ask questions, make comments about the style, technique, possible errors, violations of standards, discuss alternative approaches and other issues. Solutions to issues may also be discussed.</p> </div> </div> <p style="font-size: small; margin-top: 10px;">SOURCE: Adapted from IEEE Standard Glossary of Software Engineering Terminology (IEEE 729)</p>	<p>In this lecture, I'll describe the informal walkthrough quality review model.</p> <p>The walkthrough is a very popular software quality review model that has numerous variations.</p> <p>This model is considered to be a relatively informal review model...and it can be very effective if certain things, which I will describe, are incorporated.</p> <p>Used as a software quality review, the objectives of a walkthrough are to detect and remove defects in a work product, and to ensure that the work product meets its requirements. Some organizations expand the scope of the walkthrough to include additional things as described in the definition you see here, but experience has shown that it is best to limit the objectives when this model is applied as a quality review.</p> <p>The review is implemented as a meeting, either in-person or virtual. The work product author leads the review team through the work product, usually summarizing and paraphrasing chunks of material.</p> <p>Reviewers, who ideally have looked through the work product as part of their pre-review meeting preparation, can ask questions, make comments, and raise defect issues.</p> <p>The issues should be logged and resolved after the walkthrough meeting.</p> <p>In this model, the work product author acts as both a presenter, literally walking the review team through the work product, and also typically serves as the meeting facilitator.</p>

### Walkthrough Guidelines

- 1 Minimize the number of objectives for the walkthrough
- 2 Only invite those who can contribute
- 3 Clearly communicate purpose & expectations
- 4 Use a checklist to help reviewers focus attention
- 5 Limit review meeting time
- 6 Produce an issues list
- 7 Include a follow-up step
- 8 Log issues but limit discussion of solutions

If certain guidelines are followed, the informal walkthrough can be a very effective quality review model.

In practice, one commonly occurring problem is that the objectives for the walkthrough are too broad and can become unrelated. As an example, sometimes people are invited for orientation purposes or because they think they have a need to know where the project team is at. By itself, this is not detrimental, but what often happens is that the focus of the meeting drifts to accommodate questions and comments from this group, the meeting drags on, and the defect detection objective is set aside.

So...one important guideline is to minimize the objectives to include only those mentioned earlier, and to invite only those who can directly contribute to the review's objectives.

Another commonly occurring problem is that people are unclear as to what the objectives are, so it is important to communicate objectives and expectations in advance. I usually recommend to my clients that an email be sent in advance specifying the review objectives, scope, and expectations. This can help pre-condition reviewers and help to keep things focused on the correct objectives.

Some organizations use checklists to help focus reviewer attention on important things to look for. A sample checklist for a requirements review is illustrated later on in this lecture.

It's also important to limit the meeting time for the walkthrough. Ninety minute to two hours is an industry rule of thumb. Done correctly, walkthroughs are energy-intensive, and people tire and lose focus when meetings drag on too long...and this increases the likelihood that defects will slip through the cracks and not be identified. Another reason for limiting meeting time, in my experience, is that the longer the meeting the more time you are asking reviewers to invest...and this may cause some who would be important contributors to opt out because the walkthrough is taking up a good part of their work day.

	<p>Guidelines six and 7 are absolutely crucial to ensuring that walkthroughs are effective. Issues need to be logged, issue owners need to be identified, and a follow-up activity needs to be included to be sure that all issues have, in fact, been reconciled. In my own experience with clients, I have found that unless a follow-up step is included there's about a 50-50 chance that issues will actually be resolved...and...unless they are resolved they are still present. It surprises me how often I see walkthroughs that do a good job of defect detection, but then fail to actually eliminate defects due to lack of a follow-up step. Recall the error amplification model from an earlier lecture. Unless defects are actually removed they will be amplified. Failure to follow-up can make the walkthrough activity a waste of time and resources...and it will add to the project cost and increase its schedule.</p> <p>The last guideline can substantially reduce the overall time spent doing walkthroughs and can also increase its effectiveness. It is best to log issues but limit discussion of how to resolve the issues during the walkthrough meeting. How come? There are three primary reasons. One, history has shown that typically only a single individual, or a small subset of individuals, are responsible for resolving an issue...so it is not productive to take up everyone's time ironing out a resolution. Second, people are under a bit of time pressure during the meeting because they have to get through the work product, and this sometimes leads to issue owners jumping prematurely to a resolution that, upon further reflection, turns out to be incorrect. And third, if issue owners are not present at the walkthrough and the review team attempts a resolution, it can often be wrong...so now we have the possibility of the walkthrough being a source of error.</p> <p>In practice, this last guideline may be difficult to achieve...because in many business meetings we are used to solving issues as well as identifying issues. But...the payoff for including this guideline can be significant. I helped one of my airline clients implement this and they reduced the time spent in walkthrough meetings by 40 percent...and also increased the walkthrough effectiveness by eliminating the problems I just mentioned.</p>
--	--

4	<div><div></div><h3>Sample Walkthrough Checklist</h3><table><thead><tr><th>Requirements Document</th><th>Individual Requirements</th></tr></thead><tbody><tr><td>✓ Does the documentation format comply with standards?</td><td>✓ Is the requirement correct?</td></tr><tr><td>✓ Is the documentation complete?</td><td>✓ Is the requirement complete?</td></tr><tr><td>✓ Are all relevant attachments &amp; appendices included?</td><td>✓ Is the requirement unambiguous?</td></tr><tr><td>✓ Is the document easy to understand?</td><td>✓ Is the requirement testable?</td></tr><tr><td>✓ Are both functional and applicable non-functional requirements included?</td><td>✓ Is the requirement necessary?</td></tr><tr><td>✓ Are terms and acronyms defined in a glossary?</td><td>✓ Is the requirement traceable?</td></tr><tr><td>✓ Are there any missing requirements?</td><td>✓ Is the requirement feasible?</td></tr><tr><td></td><td>✓ Does the requirement express "what" and not "how"?</td></tr><tr><td></td><td>✓ Is the requirement concise?</td></tr></tbody></table></div>	Requirements Document	Individual Requirements	✓ Does the documentation format comply with standards?	✓ Is the requirement correct?	✓ Is the documentation complete?	✓ Is the requirement complete?	✓ Are all relevant attachments & appendices included?	✓ Is the requirement unambiguous?	✓ Is the document easy to understand?	✓ Is the requirement testable?	✓ Are both functional and applicable non-functional requirements included?	✓ Is the requirement necessary?	✓ Are terms and acronyms defined in a glossary?	✓ Is the requirement traceable?	✓ Are there any missing requirements?	✓ Is the requirement feasible?		✓ Does the requirement express "what" and not "how"?		✓ Is the requirement concise?	<p>As I mentioned earlier, it's often useful to incorporate checklists into walkthroughs to help reviewer's focus on what's important.</p> <p>If reviewers don't know what to look for the effectiveness of the review may be compromised.</p> <p>Here's a sample checklist that could be used in a requirements walkthrough. I've split it into two parts...one focuses on the content of the document itself, and one focuses on the quality of the actual requirements.</p> <p>As an aside...too often, people focus on style, formatting, and grammar...and nitpick at those...and that does not add value in my experience. To help avoid this for one of my clients we made spelling and grammar checking a pre-requisite for doing the walkthrough...and it helped immensely.</p>
Requirements Document	Individual Requirements																					
✓ Does the documentation format comply with standards?	✓ Is the requirement correct?																					
✓ Is the documentation complete?	✓ Is the requirement complete?																					
✓ Are all relevant attachments & appendices included?	✓ Is the requirement unambiguous?																					
✓ Is the document easy to understand?	✓ Is the requirement testable?																					
✓ Are both functional and applicable non-functional requirements included?	✓ Is the requirement necessary?																					
✓ Are terms and acronyms defined in a glossary?	✓ Is the requirement traceable?																					
✓ Are there any missing requirements?	✓ Is the requirement feasible?																					
	✓ Does the requirement express "what" and not "how"?																					
	✓ Is the requirement concise?																					