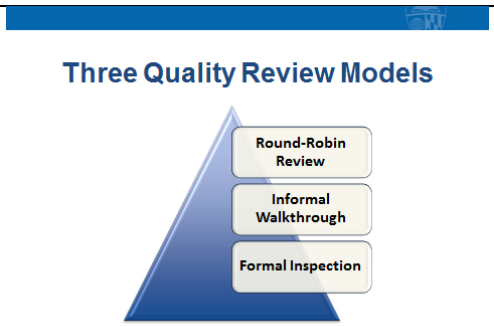
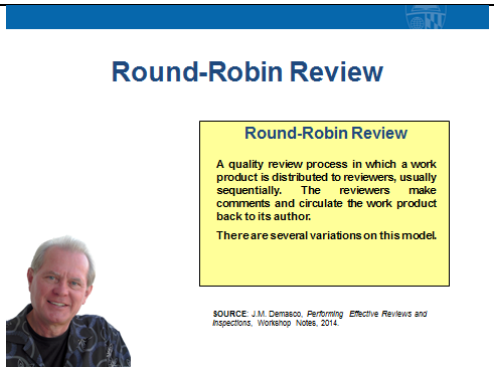
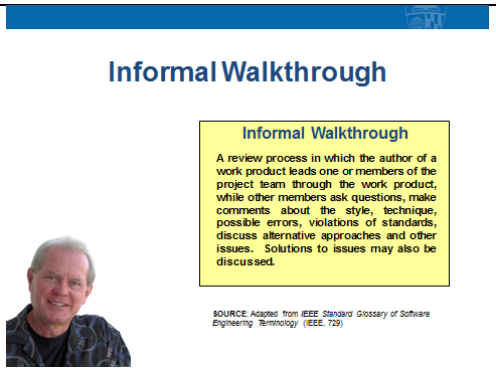
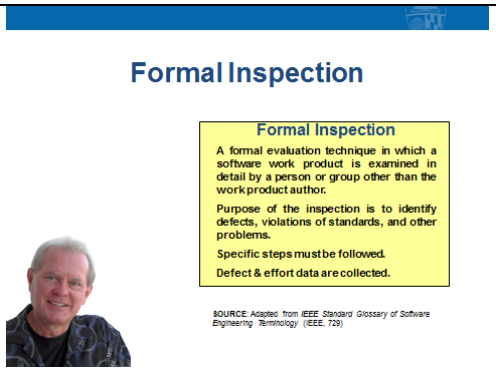
1	**Software Quality Reviews** Objectives	
2	**Lecture Topics** • Discuss objectives for software quality reviews • Discuss common review problems • Introduce three commonly used software review models 	In this lecture we'll discuss the key objectives of software quality reviews, some of the common problems organizations run into when they perform quality reviews, and we'll define three common types of software quality reviews.
3	**What's a Review?** **Review** A formal meeting at which a product or document is presented to the user, customer, or other interested parties for comment & approval. It can be a review of the management & technical progress of a project as well. SOURCE: Adapted from *IEEE Standard Glossary of Software Engineering Terminology* (IEEE, 729) 	There are many different types of reviews that can be conducted during the course of a software project. A review is basically some type of meeting…as you can see from this definition. A meeting can take place face-to-face, or be virtual and be conducted over the Internet. It can be synchronous or asynchronuous.
4	**Common Review Objectives** Share Knowledge Orientation, Training Get Consensus, Approval Discuss Issues Look for Better Ways Identify Problems Discuss Solutions Report Status Make Decisions	Because the term "review" has a very generic meaning, it is very important to define and communicate the objectives of each type of review…or there is a good chance that the review will not be effective. Here's a list of common review objectives that I've compiled from working with my clients. This list is not all-inclusive. It just contains some of the objectives that come up most frequently. I've had a number of clients list all of these as objectives for a single review…and that is problematic. Those same clients also reported that their reviews were mostly ineffective. Let's have a look at why.

5		Let me start by listing some frequently occurring problems that my clients have experienced with reviews. Again…this is not intended to be a complete list of problems, but I'll bet that if you've ever attended a review at work…or even a general meeting…there's a significant probability that you've personally experienced at least one of these problems. When these problems occur in software quality reviews, the reviews tend to become ineffective, and we risk adding to the cost and increasing the schedule. And when organizations experience quality reviews that add to the cost and increase the schedule, they often discontinue the entire quality review process.
6		In this lecture, we are discussing software quality reviews. Exactly what kind of review is that? A software quality review is a type of review in which we examine a work product that is produced during the course of a project. The work product could be a project plan, a requirements document, a design document, code, a test plan, and so forth. The work product is a key component of a software quality review. There should only be a few objectives for a software quality review…to identify any defects that exist in the work product, to remove the defects…preferably as early as possible in the life cycle, and to ensure the work product satisfies its requirements. That's it. It shouldn't be used to share information, to inform stakeholders about what our strategy and direction are, to challenge project scope, to talk about project schedules or project, or anything else. Many organizations make the mistake of including too many objectives on the agenda…and more often than not objectives that don't have anything to do with quality. And that's a recipe for disaster.

7		I'm going to introduce three different models that organizations in industry commonly use to implement software quality reviews. There are more…but these are very representative of the universe. The first review model is what I like to refer to as a round-robin review. The second model is commonly called a walkthrough, and it's generally an informal type of review. The walkthrough review model has many variations and is very common in industry. The third model is the formal inspection model. As its name implies, this is a more rigorous and formal model than the other two.
8		Here's a definition of what I refer to as a round-robin review. There are several variations, but this definition will suffice for our purposes. A work product is circulated around to a team of reviewers. Each reviewer reviews the work product in isolation and makes comments…either by marking up the document or by attaching an external list of comments. Eventually, the document gets circulated back to the author. The work product may be emailed around or we may use the cloud. The potential advantages of this review model are that reviewers can fit the review into their own schedule for the most part. They don't have to show up on a particular date and time. It's also easy to implement when reviewers are not co-located. The disadvantages of this review model are several: there is often a lack of consistency and rigor. Some people will just go through the motions and rubber stamp the work product or give it a half-hearted look…and you can't control that. Another disadvantage is a lack of information transfer to the entire review team. There is often a synergistic effect when an entire review team is collaborating at the same time. Someone makes a comment and that triggers an idea that otherwise wouldn't have surfaced. Even though review team members can read the comments of others, it does not have anywhere near the same impact. Yet another

		potential disadvantage is that if the review team doesn't provide timely feedback, the project schedule can be delayed. And finally, in many reviews of this type the work product author is under no obligation to implement the reviewer comments. This type of review may be okay for certain types of minor work products, but it is not effective for major work products like requirements, design, code, or testing work products because of the aforementioned disadvantages.
9		A second model that is frequently used for software quality reviews is the informal walkthrough model. Here's an industry definition of the walkthrough…and there are many variations on it. In my experience, when organizations tell me they are performing quality reviews on work products, their review model frequently falls into this category. The walkthrough model is appropriate for all major project work products. It can be very effective, moderately effective, or an absolute waste of time and resources, depending upon how it is structured and carried out. In a later lecture I will describe the characteristics that make this review model effective. This review model is synchronous. The review team will meet in-person or virtually over the Internet.
10		The third type of frequently used quality review model is the formal inspection model. This is more structured, rigorous, and formal than the walkthrough model…and it's considered to be an industry best practice. According to T. Capers Jones, an industry expert on software quality and measurement, all best-in-class software organizations use some variation of the formal inspection model…where "best-in-class" means organizations that achieve a software product defect detection rate of 95 percent or higher. An interesting fact about this model is that it was purposely invented, if you will, to mitigate the problems historically encountered when doing software quality reviews.

		Not every organization needs to use this quality review model. For many…using the more lightweight walkthrough approach may be “good enough”. Earlier in this course you learned about the CMMI process improvement model. The formal inspection review model maps into the CMMI Peer Review process area. We’ll talk about the details of the formal inspection review model in a later lecture.