3 STATIC TESTING

Geoff Thompson

INTRODUCTION

This chapter provides an introduction to an important area of software testing – static testing. Static testing techniques test software without executing it. They are important because they can find errors and defects before code is built/executed and therefore earlier in the life cycle of a project, making corrections easier and cheaper to achieve than for the same defects found during test execution. Review types and techniques are central to the static testing approach, and in this chapter we will look at the alternative types of review and how they fit with the test process that was defined in **Chapter 1**.

Learning objectives

The learning objectives for this chapter are listed below. You can confirm that you have achieved these by using the self-assessment questions that follow the ‘Check of understanding’ boxes distributed throughout the text and the example examination questions provided at the end of the chapter. The chapter summary will remind you of the key ideas.

The sections are allocated a K number to represent the level of understanding required for that section; where an individual topic has a lower K number than the section as a whole, this is indicated for that topic; for an explanation of the K numbers, see the **Introduction**.

Static testing basics (K2)

- FL-3.1.1 Recognize types of software work product that can be examined by the different static testing techniques. (K1)
- FL-3.1.2 Use examples to describe the value of static testing.
- FL-3.1.3 Explain the difference between static and dynamic techniques, considering objectives, types of defects to be identified, and the role of these techniques within the software lifecycle.

Review process

- FL-3.2.1 Summarize the activities of the work product review process.
- FL-3.2.2 Recognize the different roles and responsibilities in a formal review. (K1)
- FL-3.2.3 Explain the differences between different review types: informal review, walkthrough technical review, and inspection. (K2)
- FL-3.2.4 Apply a review technique to a work product to find defects. (K3)
- FL-3.2.5 Explain the factors that contribute to a successful review. (K2)

Self-assessment questions

The following questions have been designed to enable you to check your current level of understanding for the topics in this chapter. The answers are at the end of the chapter.

Question SA1 (K1)

One of the roles in a review is that of facilitator. Which of the following best describes this role?

a. Ensures the effective running of the review meetings, where it is decided that a meeting is required.

b. Allocates time in the plan, decides which reviews will take place and ensures that the benefits are delivered.

c. Writes the document to be reviewed, agrees that the document can be reviewed and updates the document with any changes.

d. Documents all issues raised in the review meeting, records problems and open points.

Question SA2 (K2)

Which of the following statements are correct for walkthroughs?

i. Often led by the author.

ii. Documented and defined results.

iii. All participants have defined roles.

iv. Used to aid learning.

v. Main purpose is to find defects.

a. i, ii and v are correct.

b. ii, iii and iv are correct.

c. i, iv and v are correct.

d. iii, iv and v are correct.

Question SA3 (K1)

Which of the following is an activity in the review process?

a. Design of the document.
b. Booking meeting rooms.
c. Writing program code.
d. Fixing and reporting.

BACKGROUND TO STATIC TESTING

Static testing tests software and work products without executing the code. Typically, this includes requirements or specification documents, and the testing of code without actually executing it. The first of these is known as a review and is typically used to find and remove errors and ambiguities in documents before they are used in the development process, thus reducing one source of defects in the code; the second is known as static analysis, and it enables code to be analysed for structural defects or systematic programming weaknesses that may lead to defects.

Static techniques find the causes of failures rather than the failure itself, which is found during test execution.

Reviews are normally completed manually; static analysis is normally completed automatically using tools.

Giving a draft document to a colleague to read is the simplest example of a review, and one that can yield a larger crop of errors than often anticipated (see **Chapter 5** regarding ‘world view’ to understand why).

WORK PRODUCTS THAT CAN BE EXAMINED BY STATIC TESTING

Virtually any work product can be examined using static testing techniques. A work product is anything that is written down and can include:

- specifications such as business requirements, functional requirements, security requirements and non-functional requirements;
- epics, user stories and acceptance criteria;
- architecture and design specifications;
- code;
- testware such as test plans and test cases;
- user guides;
- web pages;
- contracts, plans, schedules and budgets;
- design models.

BENEFITS OF STATIC TESTING

The earlier in the life cycle that static testing is applied, the larger the benefits are. If work products are statically tested before any code is written, this will remove defects from the work product and ensure these defects are not built into the code. As has been shown in **Chapter 1** the earlier a defect is found, the cheaper it is to fix. If a defect in the design documentation is built into code and even deployed to the user base, the fix can be enormously expensive. How much more expensive a defect found in live use could be, can be seen in the following example: a bank develops a letter to inform their clients of a change in interest rates. A small defect (spelling mistake), if found in the initial letter design before the letter is coded, costs no more than a few seconds of the author's time to correct. However, if that spelling mistake moves from the design, into the coded letter and goes into live use without being spotted, the cost to fix it increases considerably. The letter needs to be taken out of live use, corrected and retested, which could take up to a week to complete but could have been fixed in a few seconds if found earlier.

Static testing benefits may include:

- detecting and correcting defects more efficiently, and prior to test execution;
- identification of defects not easily found by dynamic testing;
- preventing defects in design or coding by uncovering inconsistencies, ambiguities, contradictions, omissions, inaccuracies and redundancies in requirements;
- increasing development productivity;
- reducing development cost and time;
- reducing dynamic testing cost and time;
- reducing total cost of quality over the software's lifetime, due to fewer failures later in the life cycle or after delivery into live operation;
- improving communication between team members in the course of participating in reviews.

Benefits recognition is key to ensuring that static testing remains a focus, as when no one recognises any benefits there is a better than average chance that static testing will be removed from the project plans. It is important, therefore, that results of static testing are made clear to all stakeholders regularly.

DIFFERENCES BETWEEN STATIC AND DYNAMIC TESTING

Static and dynamic testing have the same objective: to find defects as soon as possible. The main difference is that static testing is carried out against work products without actually executing any code, whereas dynamic testing is carried out by executing actual code or the final software or hardware product.

Often, code may not be exercised that often, or is deeply embedded, so building a dynamic test to find defects is sometimes too hard or impossible to do. Static testing is most effective when it is used to find defects before any code is written, thereby ensuring that the code that is written is not based on wrong or faulty specifications and so on. Static testing can often be used in this situation to find defects quicker and more easily than in dynamic testing.

Static testing improves the consistency and quality of work products, while dynamic testing focuses on externally visible behaviours; perhaps the next screen that is displayed, or the customer details displayed on the screen.

Typical defects, that are easier and cheaper to find during static testing, include:

- requirements defects (e.g. inconsistencies, ambiguities, contradictions, omissions, inaccuracies and redundancies);
- design defects (inefficient algorithms or database structures, high coupling (the lack of interdependence between software modules) and low cohesion (associated with undesirable traits such as being difficult to maintain, test, reuse or even understand));
- coding defects (e.g. variables with undefined values, variables that are declared but never used, unreachable code and duplicate code);
- deviations from standards (e.g. lack of adherence to coding standards);
- incorrect interface specifications (e.g. different units of measurement used by the calling system than by the called system);
- security vulnerabilities (e.g. susceptibility to buffer overflows);
- gaps or inaccuracies in test basis traceability or coverage (e.g. missing tests for an acceptance criterion).

Maintainability defects, such as poor reuse of components, code that is difficult to analyse and modify without introducing new defects, improper modularisation or no documentation at all, can all be found using static testing techniques.

REVIEW PROCESS

There are two types of review, informal and formal.

An informal review is identified by the principle that it is not following a defined process and has no formal documented output.

A formal review is identified by team participation, documented results and documented procedures are followed. Often there will also be defined roles.

The decision on the appropriate level of formality for a review is usually based on combinations of the following factors:

- The type of Software Development Life Cycle (there are different approaches to reviews in an Agile project compared to a waterfall project.

- The maturity of the development process. The more mature the process is, the more formal reviews tend to be.
- The complexity of the work product.
- Legal or regulatory requirements. These are used to govern the software development activities in certain industries, notably in safety-critical areas such as railway signalling, and determine what kinds of review should take place.
- The need for an audit trail. Formal review processes ensure that it is possible to trace backwards throughout the Software Development Life Cycle. The level of formality in the types of review used can help to increase how much is contained within the audit trail.

Reviews can also have a variety of objectives, where the term ‘review objective’ identifies the main focus for a review. Typical review objectives include:

- finding defects;
- gaining understanding;
- generating discussion;
- education;
- decision making by consensus.

The way a review is conducted will depend on what its specific objective is, so a review aimed primarily at finding defects will be quite different from one that is aimed at gaining understanding of a document.

All reviews, formal and informal alike, exhibit the same basic elements of process:

- The document under review is studied by the reviewers.
- Reviewers identify issues or problems and inform the author either verbally or in a documented form, which might be as formal as raising a defect report or as informal as annotating the document under review.
- The author decides on any action to take in response to the comments, and updates the document accordingly.

This basic process is always present, but in the more formal reviews it is elaborated to include additional stages and more attention to documentation and measurement.

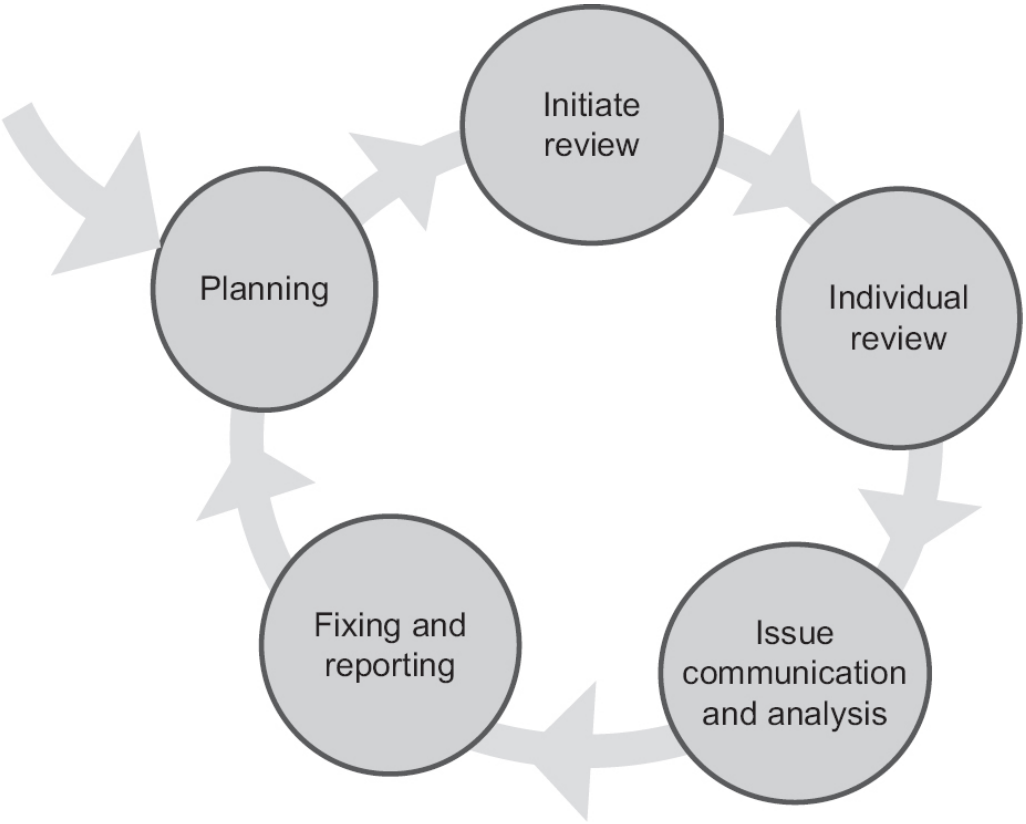
WORK PRODUCT REVIEW PROCESS

Reviews all follow the same basic process: the more formal the review, the more formal the process surrounding the review is. The following list explains the key stages in more detail. See also Figure 3.1 the work product review process flow.

Planning

- Defining the scope – this includes the purpose of the review; for example finding defects, what documents or parts of documents are to be reviewed and the quality characteristics.
- Estimating the effort required and the time frame that the review will be undertaken in – much like the estimating required to fund the test activity, reviews need to be estimated so they can be costed in a project budget.

Figure 3.1 Stages of a formal review



- Identifying the review characteristics such as review type, roles, activities and checklists – as described above, there are many factors that help the decision; for example, what type of review characteristics will be used.
- Selecting the people to undertake the review – ensuring that those selected can and will add value to the process. There is little point in selecting a reviewer who will agree with everything written by the author without question. As a rule of thumb, it is best to include some reviewers who are from a different part of the organisation, who are known to be ‘picky’ and known to be dissenters. Reviews, like weddings, are enhanced by including ‘something old, something new, something borrowed, something blue’. In this case ‘something old’ would be an experienced practitioner; ‘something new’ would be a new or inexperienced team member; ‘something borrowed’ would be someone from a different team; and ‘something blue’ would be the dissenter who is hard to please. At the earliest stage of the process, a moderator (review leader) must be identified. This is the person who will coordinate all of the review activity. This also includes allocating roles.

- Allocating roles – each reviewer is given a role to provide them with a unique focus on the document under review. Someone in a tester role might be checking for testability and clarity of definition, while someone in a user role might look for simplicity and a clear link to business values. This approach ensures that, although all reviewers are working on the same document, each individual is looking at it from a different perspective.
- Defining the entry and exit criteria, especially for the most formal review types (e.g. inspections) – before a review can start certain criteria have to be met. These are defined by the moderator, and could include: all reviewers must have received the review papers; all reviewers have a kick-off meeting booked in their diaries; any training of reviewers has been completed.
- Checking entry criteria (mainly used for more formal review types such as inspections) – this stage is where the entry criteria agreed earlier are checked to ensure that they have been met so that the review can continue.

Initiate review

- Distributing the work products – the facilitator distributes all of the required documents to all reviewers.
- Explaining the scope, objectives, process, roles and work products to the participants – this can be run as a meeting or simply by sending out the details to the reviewers. The method used will depend on timescales and the volume of information to pass on. A lot of information can be disseminated better by a meeting rather than expecting reviewers to read pages of text.
- Answering any questions raised by the review team.

Individual review

- Reviewing all parts of the work product including the source documents.
- Noting potential defects, questions and comments – in this stage the potential defects, questions and comments found during individual preparation are logged.

Issue communication and analysis

- Communicating potential defects either in a review meeting or via a defect log.
- Analysing potential defects; if agreed that there is a defect, assigning ownership of the repair work.
- Evaluating and documenting the required quality criteria, identifying whether they have been met or not.
- Evaluating the review results against the review exit criteria to decide whether the work product is to be rejected for major change or simply updated with minor changes.

Fixing and reporting

- Creating defect reports for those findings that require changes, which may include making recommendations regarding handling the defects, making decisions about the defects and so on.
- Fixing defects found – here, typically, the author will be fixing defects that were found and agreed as requiring a fix.
- Communicating defects to the appropriate person or team.
- Recording updated status of defects (in formal reviews) – always done during more formal review techniques; optional for others.
- Gathering metrics, such as how much time was spent on the review and how many defects were found, to show notionally how the quality of the reviewed item has increased, but also the value added by the review to later stages of the life cycle; for example, how much the defect would have cost if it hadn’t been found until user acceptance testing.
- Checking on exit criteria – the moderator will also check the exit criteria (for more formal review types such as inspections) to ensure that they have been met so that the review can be officially closed.
- Accepting the work product when the exit criteria has been met.

ROLES AND RESPONSIBILITIES

The role of each reviewer is to look at the documents under review (and any appropriate source documents) from their assigned perspective; this may include the use of checklists. For example, a checklist based on a particular perspective (such as user, maintainer, tester or operations) may be used, or a more general checklist (such as typical requirements problems) may be used to identify defects.

In addition to these assigned review roles, the review process itself defines specific roles and responsibilities that should be fulfilled for formal reviews. They are:

- Author: the author is the writer or person with chief responsibility for the development of the document(s) to be reviewed. The author will in most instances also take responsibility for fixing any agreed defects.
- Manager: the manager decides on what is to be reviewed (if not already defined), assigns staff, ensures that there is sufficient time allocated in the project plan for all of the required review activities, monitors ongoing cost-effectiveness of the review, determines if the review objectives have been met, and executes control decisions if objectives are not met. Managers do not normally get involved in the actual review process unless they can add real value; for example, they have technical knowledge key to the review.
- Facilitator (often called a moderator): ensures effective running of the review. The moderator may mediate between the various points of view and is often the person on whom the success of the review rests. The facilitator will also make the final decision as to whether to release an updated document.

- Review leader: the review leader is the person who leads the review of the document or set of documents, including planning the review, running the meeting, and follow-ups after the meeting.
- Reviewers: these are individuals with a specific technical or business background (also called subject matter experts, checkers or inspectors) who, after the necessary preparation, identify and describe findings (e.g. defects) in the product under review. As discussed above, reviewers should be chosen to represent different perspectives and roles in the review process and take part in any review meetings.
- Scribe (or recorder): the scribe attends the review meeting and documents all of the issues and defects, problems and open points that were identified during the meeting.

An additional role not normally associated with reviews is that of the tester. Testers have a particular role to play in relation to document reviews. In their test analysis role, they will be required to analyse a document to enable the development of tests. In analysing the document, they will also review it; for example, in starting to build end-to-end scenarios they will notice if there is a ‘hole’ in the requirements that will stop the business functioning, such as a process that is missing or some data that is not available at a given point. So, effectively a tester can either be formally invited to review a document or may do so by default in carrying out the tester’s normal test analysis role.

CHECK OF UNDERSTANDING

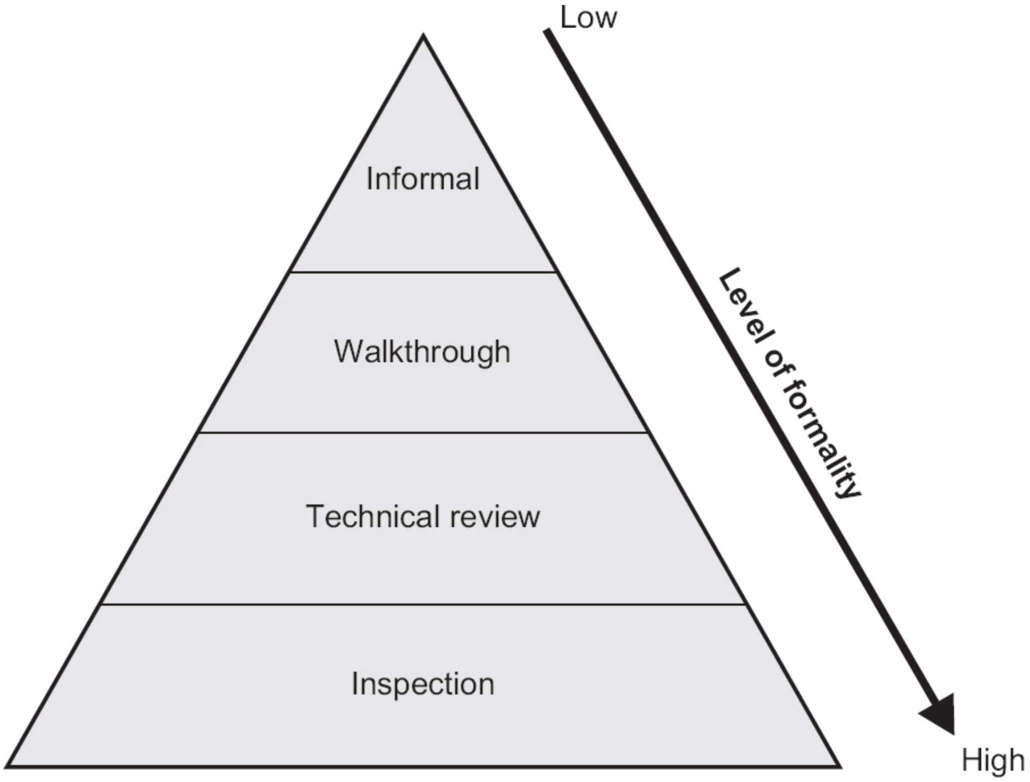
1. Identify three benefits of reviews.
2. What happens during the planning phase of a review?
3. Who manages the review process?

TYPES OF REVIEW

A single document or related work product may be subject to many different review types: for example, an informal review may be carried out before the document is subjected to a technical review or, depending on the level of risk, a technical review or inspection may take place before a walkthrough with a customer.

Figure 3.2 shows the different levels of formality by review type.

Figure 3.2 Formality of reviews



Each type of review has its own defining characteristics. We identify four review types to cover the spectrum of formality. These are usually known as:

1. Informal review (least formal) (e.g. buddy check, pairing, pair review). Key characteristics:

   - The main purpose is detecting potential defects.
   - A possible additional purpose could be to generate new ideas, or to quickly resolve a problem.
   - There is no formal process underpinning the review.
   - It may not involve a review meeting.
   - It may be performed by a colleague of the author or opened up to many people.
   - The review may be documented but this is not required; many informal reviews are not documented.
   - There may be some variations in the usefulness of the review depending on the reviewer; for example, the reviewer does not have the technical skills but is just available to check quickly and ensure that the document makes sense.
   - Use of a checklist is optional.
   - This is a common review in Agile development.

2. Walkthrough. Key characteristics:

   - The main purpose is to find defects, improve the software product, consider alternative implementations, and/or evaluate conformance to standards and specifications.

- Possible additional purposes include exchanging ideas about techniques or style variations, the training of participants and achieving consensus.
- Preparation by reviewers before the walkthrough meeting, production of a review report or a list of findings, and appointment of a scribe who is not the author are all optional components.
- The meeting is led by the author of the document under review and attended by members of the author’s peer group.
- A scribe is mandatory.
- Use of a checklist is optional.
- Review sessions are open-ended and may vary in practice from quite informal to very formal.
- Walkthroughs typically explore scenarios, or conduct dry runs of code or processes.
- Potential defect logs and review reports may be produced.
- It may vary in practice from quite informal to very formal.

3. Technical review. Key characteristics:

   - The main purpose is gaining consensus and detecting potential defects.
   - Possible additional purposes are evaluating quality and building confidence in the work product, generating new ideas, motivating authors to improve future work products and to consider alternative implementations.
   - Technical reviews are documented and use a well-defined defect detection process that includes peers and technical experts.
   - Individual preparation before the review meeting is required.
   - A review meeting is optional; if one takes place it is ideally led by a trained facilitator.
   - A scribe is mandatory, ideally not the author.
   - Reviewers using checklists is optional.
   - Potential defect logs and review reports are typically produced.

4. Inspection (most formal). Key characteristics:

   - The main purpose is to detect potential defects, evaluate quality and build confidence in the work product, preventing future similar defects through author learning and root cause analysis.
   - A possible further purpose includes authors improving future work products and the software development process, achieving consensus.
   - The inspection process is formal, based on rules and checklists, and uses entry and exit criteria.

- Pre-meeting preparation is essential, which includes the reading of any source documents to ensure consistency.
- Reviewers are either peers of the author or experts in other disciplines that are relevant to the work product.
- Specific entry and exit criteria are used.
- A scribe is mandatory.
- Review meetings are led by a trained facilitator who is not the author and usually involve peer examination of a document; individual inspectors work within defined roles.
- The author cannot lead the review, or scribe.
- All potential defects are logged and a review report is produced.
- Metrics are collected and used to improve the entire software development process, including the inspection process.

In reality, the lines between the review types often get blurred and what is seen as a technical review in one company may be seen as an inspection in another. The above is the ‘classic view’ of reviews. The key for each company is to agree the objectives and benefits of the reviews that they plan to carry out.

A single work product during its development may be subject to many different types of review.

APPLYING REVIEW TECHNIQUES

As with testing, there are specific techniques that can be used with all of the aforementioned types of review. The effectiveness of the techniques will vary dependant on what type of review they are used on. Examples of review techniques are as follows.

Ad hoc

Used mainly in the less formal review types, in an ad hoc review those involved are provided with little or no guidance on what is expected of them and how the review should be performed. This technique is very dependent on the reviewer’s skills and leads to issues like duplication of potential defects identified.

Checklist based

Uses a checklist to deliver a systematic approach to a review, ensuring that the reviewers detect potential defects based on the checklist rules. A review checklist documents the activities to be undertaken, or the types of defects to be identified; for example typos, or differences to source documents or technical content.

Scenarios and dry runs

A reviewer of a scenario and dry runs is given structured guidelines on how to read the document under review. The scenario allows dry runs of products to take place based on expected usage of the work product. A scenario provides better guidelines on how to identify specific defect types and is sometimes seen as better than the checklist approach.

Role based

In a role-based review, the work product is examined from a specific role perspective. Typical roles include end user administrator, system administrator and so on.

Perspective based

A perspective-based review is a little like a role-based review in that the reviewer will take on a different stakeholder's viewpoint. Perspectives such as end user, marketing, designer, tester or operations may be required. This approach leads to more depth in the review as well as a reduction in the duplication of potential defects found.

In addition, this type of review requires the reviewer to attempt to deliver the result of the work product. For example, a tester may have to produce draft acceptance tests if reviewing a requirement specification.

This approach has been shown to be the most effective general review technique.

Success factors for reviews

A successful review will depend on using the right type of review and techniques. There are other factors that can influence the successful outcome of a review. These fall into two types of factors:

1. organisational;
2. people related.

Examples of organisational success factors are:

- Each review should have a clearly predefined and agreed objective and the right people should be involved to ensure that the measurable outcome is met. For example, in an inspection, if the objective is technically based, each reviewer will have a defined role and have the experience to complete the technical review; this should include testers as valued reviewers. Any defects found are welcomed and expressed objectively.
- Review techniques (both formal and informal) that are suitable to the type and level of software work products and reviewers (this is especially important in inspections).
- Review techniques like checklist-based or role-based reviewing provide effective defect identification of the work product being reviewed.
- Checklists are used to ensure focus on areas of main risk and are up to date.
- Reviewing large documents in small chunks, providing frequent feedback on defects as early as possible to the author.
- Review participants have sufficient time to prepare, which may include reading all supporting work products.

- Reviews are scheduled with adequate notice.
- Management support is essential for a good review process (e.g. by incorporating adequate time for review activities in project schedules).

Examples of people-related success factors are:

- The right people are involved to ensure that the objectives are met; for example, people with different skill sets from the relevant user community.
- The use of testers as valued reviewers so that they can learn about the work product, enabling the development of better and more effective tests, and to be able to develop those tests early.
- Adequate time is allocated to each participant.
- Chunking of the work product into smaller sections enables the reviewer to focus and not lose attention during the actual review meeting.
- Defects found should be acknowledged, appreciated as helping the author and handled objectively.
- The review meeting is well managed, so that it is seen as a valuable use of time.
- There is an atmosphere of trust during the review; there is no evaluation of the person, just the work product.
- The use of negative body language can show boredom, exasperation or hostility and should be avoided.
- Ensuring that adequate training is provided, especially for the formal review processes such as inspections.
- The culture of review is all about learning and process improvement and is promoted.

CHECK OF UNDERSTANDING

1. Compare the differences between a walkthrough and an inspection.
2. Name three characteristics of a walkthrough.
3. Identify at least five success factors for a review.

SUMMARY

In this chapter, we have looked at how review techniques and static analysis fit within the test process defined in **Chapter 1**. We have understood that a review is a static test – that is it is a test carried out without executing any code (by reading and commenting on any work product such as a requirement specification, a piece of code or a test plan/test case). We have also looked at the different types of review techniques available, such as walkthroughs and inspections, as well as spending time understanding the benefits of reviews themselves.

Reviews vary in formality. The formality governs the amount of structure and documentation that surround the review itself.

To obtain the most benefit from reviews, they should be carried out as early in the project life cycle as possible, preferably as soon as the document to be reviewed has been written and definitely, in the case of work products such as requirement specifications and designs, before any code is written or executed. The roles of the participant reviewers need to be defined and, in the more structured review techniques, written output from reviews is expected.

We have learnt that static analysis is checking the developed software code before it is executed, checking for defects such as unreachable (dead code) and the misuse of development standards. We have also learnt that static analysis is best carried out using tools, which are referenced in **Chapter 6**.

Example examination questions with answers

E1. K1 question

Which of the following is most likely to be examined using static testing?

a. User guides.

b. Defect reports.

c. Test logs.

d. Attendance reports.

E2. K2 question

Which of the following has the typical formal review activities in the correct sequence?

a. Kick-off, initiate, review meeting, planning, follow-up.

b. Kick-off, planning, review meeting, issue communications and analysis, rework.

c. Planning, initiate, individual review, issue communications and analysis, fixing and reporting.

d. Planning, individual preparation, initiate, individual review, follow-up, fixing and reporting.

E3. K2 question

Which of the following statements are true?

i. Defects are likely to be found earlier in the development process by using reviews.

ii. Walkthroughs require code but static analysis does not require code.

iii. Informal reviews are used to detect potential defects.

iv. Ad hoc techniques need lots of preparation time.

v. Dynamic testing occurs after reviews have been used to find defects.

a. i, ii, iv.
b. ii, iii, v.
c. i, iv, v.
d. i, iii, v.

E4. K2 question
Which of the following defects could be identified by static testing?

a. Execution defects, coding defects and security vulnerabilities.
b. Coding defects, requirements defects and security vulnerabilities.
c. Security vulnerabilities, test basis issues and environment defects.
d. Design defects, user defects and test basis issues.

E5. K1 question
Which *one* of the following roles is typically used in a review?

a. Champion.
b. Author.
c. Project sponsor.
d. Custodian.

E6. K2 question
Which of the following is a success factor for reviews?

a. The total count of lines of code.
b. Walkthrough of a requirements document.
c. Large documents reviewed as a whole.
d. Each review has clear objectives.

Answers to questions in the chapter

SA1. The correct answer is a.
SA2. The correct answer is c.
SA3. The correct answer is d.

Answers to example examination questions

E1. The correct answer is a.
The other answers could be examined in a static review; only a is identified in the syllabus.

E2. The correct answer is c.
The correct sequence is: planning, initiate, individual review, issue communications and analysis, fixing and reporting. All of the other options have either the activities in the wrong order or activities missing from the strict flow.

E3. The correct answer is d.
The other answers are incorrect because:

ii. Walkthroughs do not require code and static analysis does require code.

iv. Ad hoc reviews need little preparation.

E4. The correct answer is b.
All other options have a dynamic testing defect amongst the options.

E5. The correct answer is b.
The Author is the only role that is typically used in a review. A Champion might sponsor the review process but is not a defined role within an actual review; a Project sponsor, if technically competent, might be asked to play a defined role within the review process, but while in that role they will not be a Project sponsor; finally, a Custodian might ensure that the results are stored safely but is not involved in the actual review itself.

E6. The correct answer is d.
Each review has clear objectives. The remaining answers are not success factors for reviews.