

## 3.1. Static Testing Basics

In contrast to dynamic testing, in static testing the software under test does not need to be executed. Code, process specification, system architecture specification or other work products are evaluated through manual examination (e.g., reviews) or with the help of a tool (e.g., static analysis). Test objectives include improving quality, detecting defects and assessing characteristics like readability, completeness, correctness, testability and consistency. Static testing can be applied for both verification and validation.

Testers, business representatives and developers work together during example mappings, collaborative user story writing and backlog refinement sessions to ensure that user stories and related work products meet defined criteria, e.g., the Definition of Ready (see section 5.1.3). Review techniques can be applied to ensure user stories are complete and understandable and include testable acceptance criteria. By asking the right questions, testers explore, challenge and help improve the proposed user stories.

Static analysis can identify problems prior to dynamic testing while often requiring less effort, since no test cases are required, and tools (see chapter 6) are typically used. Static analysis is often incorporated into CI frameworks (see section 2.1.4). While largely used to detect specific code defects, static analysis is also used to evaluate maintainability and security. Spelling checkers and readability tools are other examples of static analysis tools.

### 3.1.1. Work Products Examinable by Static Testing

Almost any work product can be examined using static testing. Examples include requirement specification documents, source code, test plans, test cases, product backlog items, test charters, project documentation, contracts and models.

Any work product that can be read and understood can be the subject of a review. However, for static analysis, work products need a structure against which they can be checked (e.g., models, code or text with a formal syntax).

Work products that are not appropriate for static testing include those that are difficult to interpret by human beings and that should not be analyzed by tools (e.g., 3<sup>rd</sup> party executable code due to legal reasons).

### 3.1.2. Value of Static Testing

Static testing can detect defects in the earliest phases of the SDLC, fulfilling the principle of early testing (see section 1.3). It can also identify defects which cannot be detected by dynamic testing (e.g., unreachable code, design patterns not implemented as desired, defects in non-executable work products).

Static testing provides the ability to evaluate the quality of, and to build confidence in work products. By verifying the documented requirements, the stakeholders can also make sure that these requirements describe their actual needs. Since static testing can be performed early in the SDLC, a shared understanding can be created among the involved stakeholders. Communication will also be improved between the involved stakeholders. For this reason, it is recommended to involve a wide variety of stakeholders in static testing.

Even though reviews can be costly to implement, the overall project costs are usually much lower than when no reviews are performed because less time and effort needs to be spent on fixing defects later in the project.

Code defects can be detected using static analysis more efficiently than in dynamic testing, usually resulting in both fewer code defects and a lower overall development effort.

### 3.1.3. Differences between Static Testing and Dynamic Testing

Static testing and dynamic testing practices complement each other. They have similar objectives, such as supporting the detection of defects in work products (see section 1.1.1), but there are also some differences, such as:

- Static and dynamic testing (with analysis of failures) can both lead to the detection of defects, however there are some defect types that can only be found by either static or dynamic testing.
- Static testing finds defects directly, while dynamic testing causes failures from which the associated defects are determined through subsequent analysis
- Static testing may more easily detect defects that lay on paths through the code that are rarely executed or hard to reach using dynamic testing
- Static testing can be applied to non-executable work products, while dynamic testing can only be applied to executable work products
- Static testing can be used to measure quality characteristics that are not dependent on executing code (e.g., maintainability), while dynamic testing can be used to measure quality characteristics that are dependent on executing code (e.g., performance efficiency)

Typical defects that are easier and/or cheaper to find through static testing include:

- Defects in requirements (e.g., inconsistencies, ambiguities, contradictions, omissions, inaccuracies, duplications)
- Design defects (e.g., inefficient database structures, poor modularization)
- Certain types of coding defects (e.g., variables with undefined values, undeclared variables, unreachable or duplicated code, excessive code complexity)
- Deviations from standards (e.g., lack of adherence to naming conventions in coding standards)
- Incorrect interface specifications (e.g., mismatched number, type or order of parameters)
- Specific types of security vulnerabilities (e.g., buffer overflows)
- Gaps or inaccuracies in test basis coverage (e.g., missing tests for an acceptance criterion)

## 3.2. Feedback and Review Process

### 3.2.1. Benefits of Early and Frequent Stakeholder Feedback

Early and frequent feedback allows for the early communication of potential quality problems. If there is little stakeholder involvement during the SDLC, the product being developed might not meet the stakeholder's original or current vision. A failure to deliver what the stakeholder wants can result in costly rework, missed deadlines, blame games, and might even lead to complete project failure.

Frequent stakeholder feedback throughout the SDLC can prevent misunderstandings about requirements and ensure that changes to requirements are understood and implemented earlier. This helps the development team to improve their understanding of what they are building. It allows them to focus on those features that deliver the most value to the stakeholders and that have the most positive impact on identified risks.

### 3.2.2. Review Process Activities

The ISO/IEC 20246 standard defines a generic review process that provides a structured but flexible framework from which a specific review process may be tailored to a particular situation. If the required review is more formal, then more of the tasks described for the different activities will be needed.

The size of many work products makes them too large to be covered by a single review. The review process may be invoked a couple of times to complete the review for the entire work product.

The activities in the review process are:

- **Planning.** During the planning phase, the scope of the review, which comprises the purpose, the work product to be reviewed, quality characteristics to be evaluated, areas to focus on, exit criteria, supporting information such as standards, effort and the timeframes for the review, shall be defined.
- **Review initiation.** During review initiation, the goal is to make sure that everyone and everything involved is prepared to start the review. This includes making sure that every participant has access to the work product under review, understands their role and responsibilities and receives everything needed to perform the review.
- **Individual review.** Every reviewer performs an individual review to assess the quality of the work product under review, and to identify anomalies, recommendations, and questions by applying one or more review techniques (e.g., checklist-based reviewing, scenario-based reviewing). The ISO/IEC 20246 standard provides more depth on different review techniques. The reviewers log all their identified anomalies, recommendations, and questions.
- **Communication and analysis.** Since the anomalies identified during a review are not necessarily defects, all these anomalies need to be analyzed and discussed. For every anomaly, the decision should be made on its status, ownership and required actions. This is typically done in a review meeting, during which the participants also decide what the quality level of reviewed work product is and what follow-up actions are required. A follow-up review may be required to complete actions.
- **Fixing and reporting.** For every defect, a defect report should be created so that corrective actions can be followed-up. Once the exit criteria are reached, the work product can be accepted. The review results are reported.

### 3.2.3. Roles and Responsibilities in Reviews

Reviews involve various stakeholders, who may take on several roles. The principal roles and their responsibilities are:

- **Manager** – decides what is to be reviewed and provides resources, such as staff and time for the review
- **Author** – creates and fixes the work product under review

- Moderator (also known as the facilitator) – ensures the effective running of review meetings, including mediation, time management, and a safe review environment in which everyone can speak freely
- Scribe (also known as recorder) – collates anomalies from reviewers and records review information, such as decisions and new anomalies found during the review meeting
- Reviewer – performs reviews. A reviewer may be someone working on the project, a subject matter expert, or any other stakeholder
- Review leader – takes overall responsibility for the review such as deciding who will be involved, and organizing when and where the review will take place

Other, more detailed roles are possible, as described in the ISO/IEC 20246 standard.

### 3.2.4. Review Types

There exist many review types ranging from informal reviews to formal reviews. The required level of formality depends on factors such as the SDLC being followed, the maturity of the development process, the criticality and complexity of the work product being reviewed, legal or regulatory requirements, and the need for an audit trail. The same work product can be reviewed with different review types, e.g., first an informal one and later a more formal one.

Selecting the right review type is key to achieving the required review objectives (see section 3.2.5). The selection is not only based on the objectives, but also on factors such as the project needs, available resources, work product type and risks, business domain, and company culture.

Some commonly used review types are:

- **Informal review.** Informal reviews do not follow a defined process and do not require a formal documented output. The main objective is detecting anomalies.
- **Walkthrough.** A walkthrough, which is led by the author, can serve many objectives, such as evaluating quality and building confidence in the work product, educating reviewers, gaining consensus, generating new ideas, motivating and enabling authors to improve and detecting anomalies. Reviewers might perform an individual review before the walkthrough, but this is not required.
- **Technical Review.** A technical review is performed by technically qualified reviewers and led by a moderator. The objectives of a technical review are to gain consensus and make decisions regarding a technical problem, but also to detect anomalies, evaluate quality and build confidence in the work product, generate new ideas, and to motivate and enable authors to improve.
- **Inspection.** As inspections are the most formal type of review, they follow the complete generic process (see section 3.2.2). The main objective is to find the maximum number of anomalies. Other objectives are to evaluate quality, build confidence in the work product, and to motivate and enable authors to improve. Metrics are collected and used to improve the SDLC, including the inspection process. In inspections, the author cannot act as the review leader or scribe.

### 3.2.5. Success Factors for Reviews

There are several factors that determine the success of reviews, which include:

- Defining clear objectives and measurable exit criteria. Evaluation of participants should never be an objective
- Choosing the appropriate review type to achieve the given objectives, and to suit the type of work product, the review participants, the project needs and context
- Conducting reviews on small chunks, so that reviewers do not lose concentration during an individual review and/or the review meeting (when held)
- Providing feedback from reviews to stakeholders and authors so they can improve the product and their activities (see section 3.2.1)
- Providing adequate time to participants to prepare for the review
- Support from management for the review process
- Making reviews part of the organization's culture, to promote learning and process improvement
- Providing adequate training for all participants so they know how to fulfil their role
- Facilitating meetings