Test Case Design

IT6206-Software Quality Assurance

Level III - Semester 6

Overview

- This chapter talks about the static testing techniques.

- In static testing we look software product and the source code to identify issues without running them.

- We will discuss four types of static techniques and their characteristics in this module.

Intended Learning Outcomes

At the end of this lesson, you will be able to:

- Identify different types of static testing techniques are how and why they’re used.
- Identify the benefits of static testing.

List of sub topics

4.1 Static Methods

4.2 Benefits of Static Testing

- 4.2.1 Informal Reviews
- 4.2.2 Walkthroughs
- 4.2.3 Technical Reviews
- 4.2.4 Inspection

Static Methods

4.1 Static Methods

- During static testing, software work products are examined manually, or with a set of tools, but not executed.

- Can be used to ‘test’ any form of document including source code, design documents and models, requirement specifications, test plan, and user manual.

- The primary objective of static testing is to improve the quality of software products by assisting engineers to recognize and fix their own defects early in the software development process.

- It starts early in the Life cycle.

4.1 Static Methods

- Types of the defects that are easier to find during the static testing are:
  - Deviation from standards
  - Missing requirements
  - Design defects
  - Non-maintainable code
  - Inconsistent interface specifications

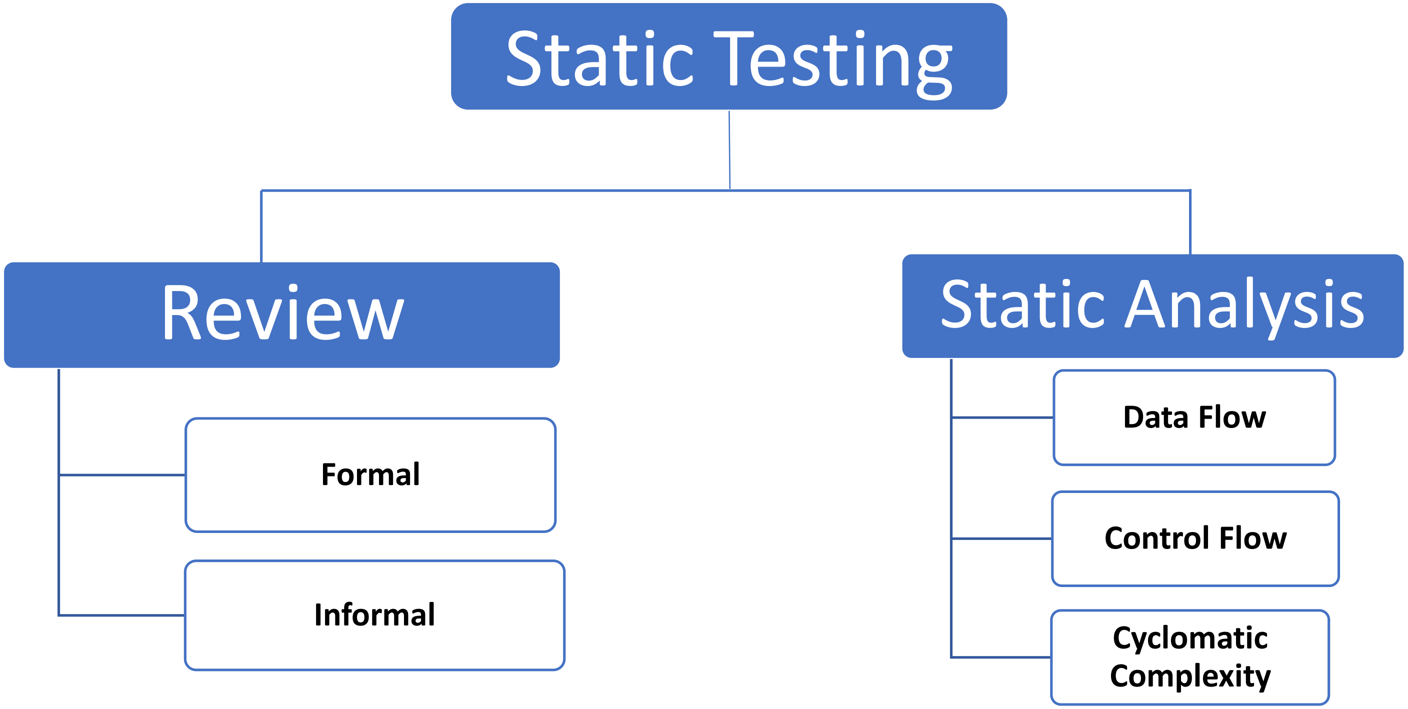
4.1 Static Methods

Features that can be tested in Static Testing

- BRD [Business Requirements Document]
- Functional or system Requirements
- Unit Use Cases
- Prototype
- Prototype Specification Document
- Test Data
- DB Fields Dictionary Spreadsheet
- Documentation/Training Guides/ User Manual
- Test Cases/Test Plan Strategy Document
- Traceability Matrix Document
- Performance Test Scripts/Automation

4.1 Static Methods

Types of Static Methods



4.1 Static Methods

Review

- Review methods aim to identify possible flaws in the software design.
- They involve a thorough examination of supporting documents, such as software requirements specifications, to identify and correct errors and defects.
- During the review, individuals analyze the documents to identify errors, redundancies, and ambiguities.
- A review process can be informal or formal.

4.1 Static Methods

Informal reviews

- A two person team conduct an informal review.
- The goal is improve the quality of the work. Informal reviews are not documented.

Formal review

- Formal reviews follow a formal process
- During project planning, project manager should allow time for formal reviews

4.1 Static Methods

- There for types of Reviews,
  - Informal (buddy check, pairing, pair review)
  - Walkthrough
  - Technical review
  - Inspection

Formal Reviews

4.1 Static Methods

Static Analysis

- Static analysis involves assessing the quality of the code created by developers.

- Various tools are utilized to analyze the code and compare it to a set of standards. This process assists in identifying several defects such as:
  - Unused variables
  - Dead code
  - Infinite loops
  - Variable with undefined value
  - Wrong syntax

4.1 Static Methods

Static Testing Tools

- Static testing tools can be used to automate the static testing process. Some examples include the following:
  - SourceMeter
    - SourceMeter is a static testing tool that can analyze code written in various programming languages including C, C++, Java, C#, and Python. It has the capability to integrate with other static testing tools such as PMD or FindBugs.
  - PyCharm
    - PyCharm is an integrated development environment (IDE) for Python that includes static code testing capabilities. This tool supports Python as well as web technologies such as HTML, CSS, and JavaScript. In addition to its static testing features, PyCharm also offers useful tools like a debugging feature.

Benefits of Static Testing

© 2022 e-Learning Centre, UCSC

4.2 Benefits of Static Testing

- Lowered costs during the initial development phase due to a decrease in the amount of rework required to fix errors.

- Reduced development time.

- Feedback taken in this stage can enhance the overall functionality of the software. As other types of testing, such as dynamic testing, begin, there will be fewer errors to detect.

- Software code becomes more maintainable.

- This process will also help give developers a better idea of the quality issues found in the software.

4.2 Benefits of Static Testing

- Automated tools can expedite the code and document review process.

- Static testing can also boost the amount of communication between teams.

Informal Review

4.2.1 Informal review (buddy check, pairing, pair review)

- An informal review is characterized by the following attributes:
  - Main purpose/objective: detecting potential defects.
  - Possible additional purposes: generating new ideas or solutions
  - Not based on a formal (documented) review process.
  - May not involve a review meeting.
  - May be performed by a colleague of the author (buddy check) or by more people.
  - Results may be documented (but often are not).
  - Varies in usefulness depending on the reviewer(s).
  - Use of checklists is optional.
  - Very commonly used in Agile development.

4.2.1 Informal review (buddy check, pairing, pair review)

- An informal review may simply be one person saying to a colleague, 'Could you have a quick look at what I've just done?' The colleague may spend less than an hour looking through and giving any comments back to the author, such as typos, something missing, or a 'But have you thought of this?' comment.

- With the right person reviewing, this buddy check can be very effective (and at little cost in time).

- Other forms of informal review include pair working, when one person works with another to produce a work product, the second person continually evaluating (that is, reviewing) what the first person is typing.

Walkthroughs

© 2022 e-Learning Centre, UCSC

4.2.2 Walkthroughs

- A walkthrough is characterized by the following attributes:
  - Main purposes: find defects, improve the software product, consider alternative implementations, evaluate conformance to standards and specifications.
  - Possible additional purposes: exchanging ideas about techniques or style variations, training of participants. achieving consensus.
  - Individual preparation before the review meeting is optional.
  - Review meeting is typically led by the author of the work product.
  - Use of a scribe is mandatory.
  - Use of checklists is optional.
  - May take the form of scenarios, dry runs or simulations.
  - Potential defect logs and review reports may be produced.

4.2.2 Walkthroughs

- Within a walkthrough, the author does most of the preparation.
- The participants who are selected from different departments and backgrounds, are not generally required to do a detailed study of the work products in advance (but it is an option).
- Because of the way the meeting is structured, a large number of people can participate and this larger audience can bring a great number of diverse viewpoints regarding the contents of the work product being reviewed.
- A walkthrough is especially useful for higher-level work products, such as requirement specifications and architectural documents.
- A walkthrough is often used to transfer knowledge and educate a wider audience about a particular work product.

Technical Reviews

4.2.3 Technical Reviews

- A technical review is characterized by the following attributes:
  - Main purposes: gaining consensus, detecting potential defects.
  - Possible further purposes: evaluating quality and building confidence in the work product, generating new ideas, motivating and enabling authors to improve future work products, considering alternative implementations.
  - Reviewers should be technical peers of the author, and technical experts in relevant disciplines.
  - Individual preparation before the review meeting is required.
  - Review meeting is optional, ideally led by a trained moderator (typically not the author).
  - Scribe is mandatory, ideally not the author.
  - Use of checklists is optional.
  - Potential defect logs and review reports are typically produced.

4.2.3 Technical Reviews

- A technical review is often a discussion meeting that focuses on achieving consensus about the technical content of a work product that all the participants have studied before the meeting.

- During technical reviews, defects are found by experts, who focus on the content of the work product.

- The experts who participate in a technical review may include, for example, architects, chief designers and key users.

- It is useful to have an independent facilitator, especially if there are a number of strong opinions about technical issues.

- Technical reviews are typically less formal than inspections, but generally more formal than walkthroughs. In practice, technical reviews may vary from quite informal to very formal.

Inspection

4.2.4 Inspection

- An inspection is characterized by the following attributes,

  - Main purposes: detecting potential defects, evaluating quality and building confidence in the work product. preventing future similar defects through author learning and root cause analysis.
  - Possible further purposes: motivating and enabling authors to improve future work products and the software development process, achieving consensus.
  - A defined process is followed, with formal documented outputs, based on rules and checklists.
  - There are clearly defined roles, which are mandatory and may include a dedicated reader who reads/paraphrases the work product aloud during the review meeting.
  - Individual preparation before the review meeting is required.

4.2.4 Inspection

- Inspection is the most formal review type.

- The work product under inspection is prepared and checked thoroughly by the reviewers before the meeting and using rules and checklists.

- In the inspection meeting, the defects found are logged and any discussion is postponed until the discussion part of the meeting.

- Depending on the organization and the objectives of a project, inspections can be balanced to serve a number of goals. For example. if the time to market is extremely important, the emphasis in inspections will be on efficiency.

4.2.4 Inspection

- When inspections are done well, they not only help to identify defects in the work products being reviewed, but the emphasis on process improvement and learning leads to better ways of producing work products, both for the author and for other reviewers.

Summary

- In static testing software product and the source code are examined without running them to identify issues.
- static testing it starts early in the Life cycle.
- Static testing techniques discussed in this module are informal review, walkthrough, technical review and inspection.