MGMT70043 Software Testing

Lesson 3: Static Testing

MGMT70043 LESSON 3

Agenda

1. Introduction and Overview
    - Quiz 2 based on Lesson 2 (and on textbook chapter 2)
    - Lesson 3: Static Testing

2. Lesson 3: Static Testing & Test Development
    - Static testing – definition, benefits, types
    - Reviews – definition, benefits, types of reviews, types of defects
    - Formal reviews – process, roles
    - Static analysis – definition, benefits, types of defects
    - Test development process

MGMT70043 LESSON 3

Static testing

- **Static testing** refers to the testing of software without executing it.
- Static testing is opposed to **dynamic testing**, which tests the software by executing it.
- The main **benefit** of static testing is that it is done earlier in the software development life cycle, when it is easier and cheaper to fix defects.
- There are two types of static testing: reviews and static analysis.

Static testing

- **Reviews** can be done on any artifact that is produced as part of the SDLC, including requirements documents, technical design documents and test case specifications.
- The goal of reviews is to uncover errors and ambiguities in the documents.
- Reviews are typically the first type of testing that is done in a project, as they can be done before any code is written.

MGMT70043 LESSON 3

Static testing

- **Static analysis** is done on the code itself.
- The goal of static analysis is to uncover structural defects in the code and deviations from development standards.
- Static analysis can be done after the code is written but before it is executed.

MGMT70043 LESSON 3

Reviews

https://www.youtube.com/watch?v=6ij9lSgAW7w

MGMT70043 LESSON 3

Reviews

- A **review** is a systematic examination of a document by someone other than the author with the goal of finding and removing defects.
- Note for BAs: the are many similarities between the review process described the ISTQB syllabus and the following business analysis tasks and techniques from BABOK v3 (Business Analysis Body of Knowledge) but the terminology is slightly different:
  - requirements verification
  - requirements validation
  - structured walkthrough

MGMT70043 LESSON 3

Reviews

The benefits of performing reviews include:

- Increased development productivity, since the requirements and design specifications will be clearer.
- Reduced testing costs, since testers will spend less time waiting for developers to provide fixes for avoidable defects.
- Reduced maintenance costs, since the final product will have fewer defects requiring ongoing support.
- Improved communication, since reviews produce a shared understanding of requirements and design.

MGMT70043 LESSON 3

Reviews

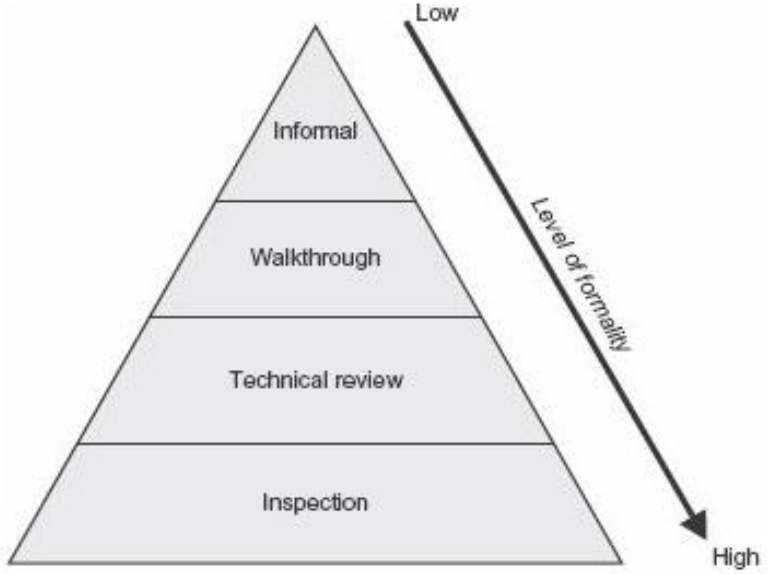
The following types of defects are typically found in reviews:

- Requirements defects. Requirements verification uncovers requirements that are unclear, incomplete or inconsistent; while requirements validation uncovers requirements that do not provide value to users.
- Design defects, such as a design that is not consistent with the requirements, or an interface specification that is not consistent with another interface specification.
- Deviations from standards, including external regulatory standards and internal organization standards.

MGMT70043 LESSON 3

Reviews

- There are four types of reviews, each with a different level of formality.



MGMT70043 LESSON 3

Reviews

- **Informal.** An informal review might consist of a document author asking a colleague to do a "sanity check" of a document before it is circulated more widely. Informal reviews are often not documented.
- **Walkthrough**. Walkthroughs are typically led by the author and have a primary goal of increasing understanding. Participants may be asked to review the document prior to the meeting, and the results may be documented.
- **Technical review**. Technical reviews have the goal of solving technical problems and making decisions regarding the solution. Technical reviews are often checklist-based.
- **Formal inspection**. Formal inspections, which are sometimes called Fagan inspections after their inventor, are always checklist-based and have formal entry and exit criteria. Issues are logged but discussion is not allowed.

Reviews

- The level of **formality of the review** will depend on various factors, including the maturity of the development process and the regulatory framework.
- For example, a safety-critical application would require a more formal review, such as a technical inspection or a formal inspection.
- The type of review performed will also depend on the **review objectives**.
- Common review objectives include identifying defects, creating a common understanding, and making decisions.
- For example, if the objective is to increase understanding, then a walkthrough would be appropriate.

Reviews

Regardless of the type of review, all reviews follow the same **basic review process**:

1. The document is studied by one or more reviewers.
2. The reviewers identify issues and communicate them to the author.
3. The author decides on corrective action and updates the document.

MGMT70043 LESSON 3

Reviews

In order to ensure that reviews are successful, the following success factors apply:

1. The objectives of the review should be clear.
2. The type of review should be appropriate for the review objectives.
3. The right people should be invited & should represent a variety of backgrounds.
4. Participants should be provided with checklists and/or assigned to roles.
5. Issues should be raised in an objective way & should be welcomed by the author.
6. The review should not be used to evaluate the author of the document.
7. The emphasis should be on increasing learning and on process improvement.
8. The review process should have management support.

Reviews

In order to measure the effective of reviews, the following KPIs (key performance indicators) can be used:

- Number of defects found during the review
- Time required to complete the review
- Percentage of budget used by the review vs. percentage of budget saved

MGMT70043 LESSON 3

Group activity

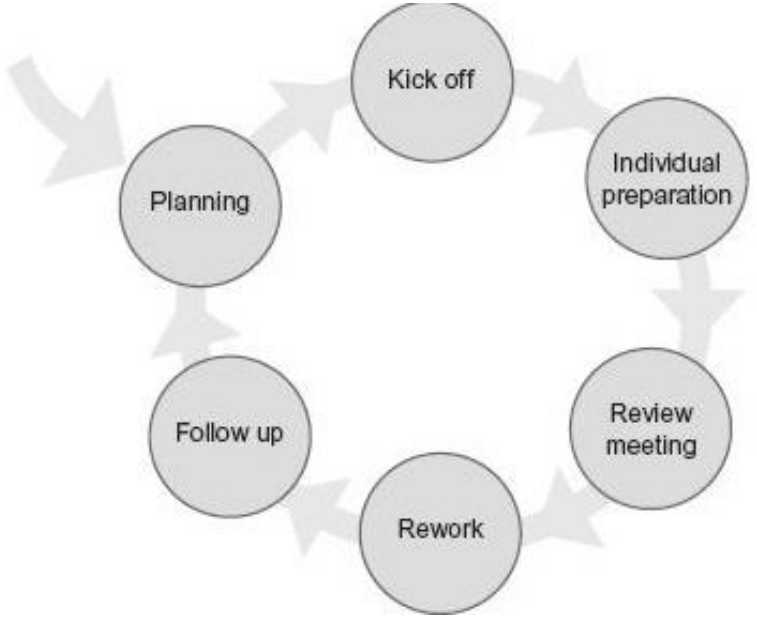
You are planning a formal review of the BRD and you have asked the following individuals to attend:

- PM
- Tester
- Development manager
- BA manager
- Operations
- End user

Prepare a checklist for each of the reviewers, indicating items that each individual should look for.

Reviews

- The figure on the right shows the six steps in the **formal review process**



MGMT70043 LESSON 3

Reviews

1. **Planning**. During planning, the document (or document part) to be reviewed will be specified, the participants will be identified, and the entrance and exit criteria will be defined.
2. **Kickoff**. During kickoff, the document is distributed to participants, and the review objective is communicated to them.
3. **Individual preparation**. During the individual preparation phase, participants review the document and note any issues.
4. **Meeting**. During the meeting, issues are raised and logged, and recommendations are made on how to resolve them.
5. **Rework**. During the rework phase, the author revises the document based on recommendations.
6. **Follow-up**. During the follow-up phase, the moderator checks that issues have been addressed and that exit criteria have been met.

Reviews

There are five **roles involved in the formal review process**:

- The **manager**, who allocates time in the project plan for the review.
- The **moderator**, who plans the review, runs the meeting, and follows up on issues.
- The **author**, who writes the document and makes fixes to it.
- The **reviewers**, who provide different perspectives on the document.
- The **scribe**, who logs issues raised in the meeting.

MGMT70043 LESSON 3

In-class exercise

1. Follow-up
2. Individual preparation
3. Kickoff
4. Meeting
5. Planning
6. Rework

a) the moderator checks that issues have been addressed
b) the document is distributed to participants
c) the author revises the document
d) participants review the document and note any issues
e) issues are raised and logged
f) entrance and exit criteria will be defined

MGMT70043 LESSON 3

In-class exercise

1. The manager
2. The moderator
3. The author
4. The reviewers
5. The scribe

a) Provide different perspectives on the document.
b) Logs issues raised in the meeting.
c) Plans the review, runs the meeting, and follows up on issues.
d) Allocates time in the project plan for the review.
e) Writes the document and makes fixes to it.

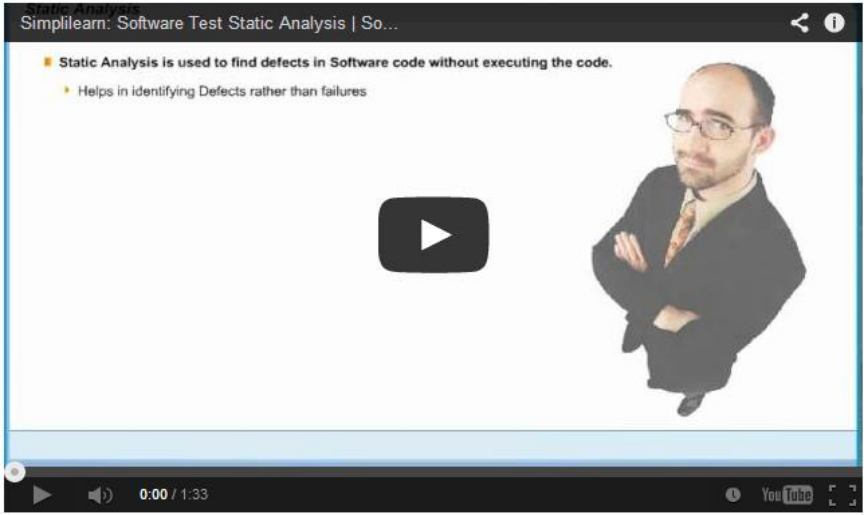
MGMT70043 LESSON 3

Static analysis

- **Static analysis** refer to the testing of code without executing it.
- It is performed almost exclusively by developers, and it is typically done using an automated tool.
- The benefits of performing static analysis include:
  - The early detection and prevention of defects.
  - Early warning about factors that can make the code difficult to maintain, such as excessive complexity.
  - Identification of defects that cannot be found using dynamic techniques, such as a lack of adherence to standards.

MGMT70043 LESSON 3

Static analysis



https://www.youtube.com/watch?v=FaojePE4oE0

MGMT70043 LESSON 3

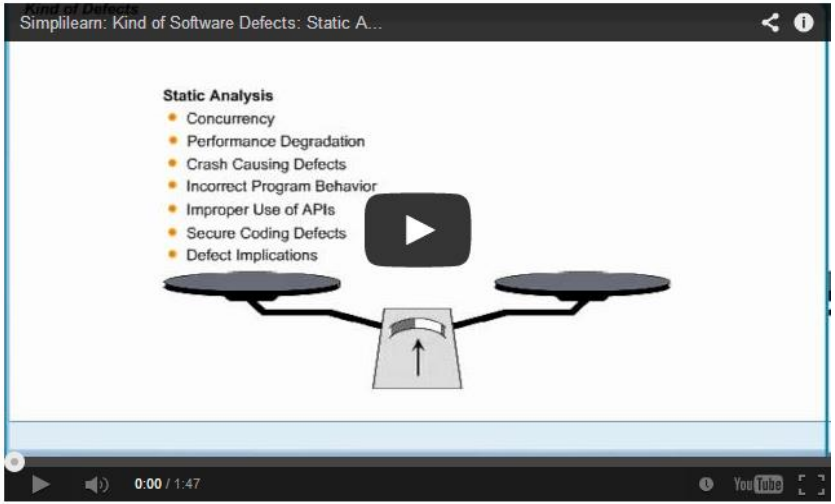
Static analysis

The following types of defects are typically found using static analysis:

- standards violations
- undefined variables
- unused variables
- dead code
- syntax violations
- security vulnerabilities
- cyclomatic complexity

MGMT70043 LESSON 3

Static analysis



https://www.youtube.com/watch?v=izsp5yjN_lQ

MGMT70043 LESSON 3

In-class activity

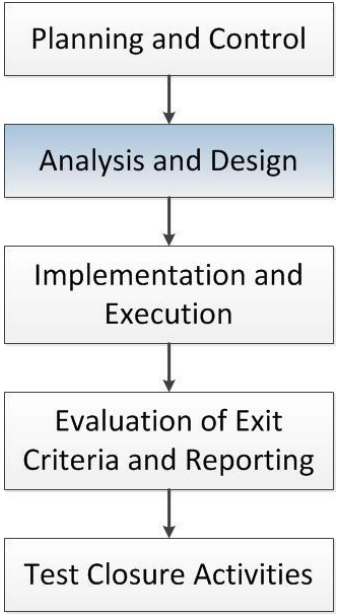
You are planning a large event that will include a multicourse meal and that will be attended by hundreds of people. To save money, you will be performing "static analysis" of the evening programme and recipes prior to execution. Find examples of the following types of defects:

a) Coding standards violations
b) Undefined or unused variables
c) Interface errors
d) Dead code
e) Syntax violations
f) Security vulnerabilities
g) Cyclomatic complexity

MGMT70043 LESSON 3

Test development process

- The test case development process takes place during the Analysis and Design phase of the Fundamental Test Process (FTP).
- It's during the Analysis and Design phase that you review documentation, identify items to be tested, and design the tests, among other things.



MGMT70043 LESSON 3

Test development process

The test development process consists of three steps:

1. Identify the **test condition**. A test condition is an item (such as a feature, a function, a business rule, a non-functional requirement, or a structural element) to be verified with test cases.
2. Specify the **test case**. A test case contains **input values** and **expected results**, **preconditions** that must be in place prior to execution, as well as **postconditions** that will exist after execution. Test cases should also show **traceability** between the requirements and the expected results.
3. Specify the **test procedure**. A test procedure, also known as a **test script**, describes the steps that must be taken in order to execute a test. For example, if a system must be in a certain state in order to execute a test, the test procedure would describe getting the system into the starting state, inputting the values, and then checking the results.

MGMT70043 LESSON 3

Test development process

- Once you've written the test procedures, you can create a **test execution schedule** to put them into the right sequence, taking priorities and dependencies into account.
- Based on **priorities** identified in the test plan, you should schedule test procedures for the most important features first.
- Once priorities have been accounted for, you can use **dependencies** to achieve efficiencies.
- Ideally, you should sequence tests so that the postcondition for one test becomes the precondition for the next test.

MGMT70043 LESSON 3

Test development process

Although test condition, test case and test procedure are conceptually distinct, in practice they are usually documented in the same place, either in the same document (typically an Excel spreadsheet) or even on the same screen in a test management tool.

The amount of documentation produced, as well as the formality of the test development process, will depend on various factors, including:

- The maturity of the development and test processes.
- The software development model. For example, an Agile model would require less test documentation, although it would not eliminate the need for it.
- The nature of the system. For example, a safety-critical system would require a more formal test development process and a greater amount of test documentation.

MGMT70043 LESSON 3

Test development process

For example, let's say the requirements document (the **test basis**) states the following:

1.1 The system shall validate ATM users with a PIN

- 1.1.1 If the user enters a valid PIN, the system shall display the transaction selection page
- 1.1.2 The first four times that the user enters a invalid PIN, the system shall display an error message and prompt the user to try again
- 1.1.3 The fifth time that the user enters an invalid PIN, the system shall keep the card and suspend the account

Step 1: Identify the test condition, for example requirement 1.1.3.

Test development process

Step 2: Create a **test case**

- The **input values** would be a valid ATM card and an incorrect PIN: 9999. Note: because test cases must include specific values, we could theoretically write one test case for each incorrect PIN, which would result in an almost infinite number of test cases. So if the correct PIN is 1234, then we could write a test case for each of the following input values: 9999, 9998, 9997, etc.
- The **expected result** would be that the system keeps the card and suspends the account.
- The **precondition** would be that the ATM user has an active account and has entered an incorrect PIN exactly four times.
- The **postcondition** would be that the ATM user has a suspended account.
- For **traceability**, the test case should contain a cross-reference to requirement 1.1.3.

MGMT70043 LESSON 3

Test development process

Step 3: Create the test procedure, also known as a **test script**, which lists the steps needed to execute the test case. The test script might look like this:

1. Setup: create an ATM user with an active account who has entered an incorrect PIN exactly four times
2. Insert the card into the ATM machine
3. Verify that the system prompts the user for the PIN
4. Enter the invalid PIN 9999
5. Verify that the system keeps the card and suspends the account

After we've written all the test scripts, we would sequence them in the **test execution schedule** based on priorities and dependencies. For example, from a dependency perspective, it would make sense to schedule tests for requirement 1.1.2 ahead of tests for requirement 1.1.3.

MGMT70043 LESSON 3

Group activity

Create a test case, test procedure (test script) and test execution schedule for test conditions 1.1.1 and 1.1.2.

1.1 The system shall validate ATM users with a PIN

- 1.1.1 If the user enters a valid PIN, the system shall display the transaction selection page
- 1.1.2 The first four times that the user enters a invalid PIN, the system shall display an error message and prompt the user to try again
- 1.1.3 The fifth time that the user enters an invalid PIN, the system shall keep the card and suspend the account

References

- Thompson, G. (2010). Chapter 3: Static Testing. In B. Hambling (Ed.), *Software Testing: An ISTQB-ISEB Foundation Guide* (pp. 57-73). Swindon: British Informatics Society Limited.
- Wahab, S. (2008, December 15). Can I Have My Requirements and Test Them Too?. *BA Times*. Retrieved from batimes.com.
- Static Techniques. ISTQB Foundation Level Syllabi. Retrieved from astqb.org.