ISTQB® Certified Tester Foundation Level Certification

CTFL 4.0

ISTQB® CTFL 4.0 Course Content

1. Fundamentals of Testing
2. Testing Throughout the Software Development Lifecycle
3. Static Testing
4. Test Analysis and Design
5. Managing the Test Activities
6. Test Tools

Static Testing

Md Rashed Karim

After completion of this chapter the student will –

1. The student learns about the static testing basics,
2. the feedback and review process

Chapter 3: Static Testing

Agenda

3.1 Static Testing Basics

3.2 Feedback and Review Process

3.1. Static Testing Basics

3.1. Static Testing Basics /1

- Static testing the software under test does not need to be executed.
- Code, specifications, architectures or other work products are evaluated through manual examination or with the help of a tool (static analysis)
- 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

3.1. Static Testing Basics /2

- Review techniques can be applied to ensure user stories are complete and understandable and include testable acceptance criteria
- Static analysis can identify problems prior to dynamic testing while often requiring less effort, since no test cases are required.
- Static analysis is often incorporated into CI frameworks. While largely used to detect specific code defects, static analysis is also used to evaluate maintainability and security.

3.1. Static Testing Basics /3

Static Testing Objectives:

- improving quality,
- detecting defects and assessing characteristics like readability,
- completeness,
- correctness,
- testability and
- consistency

3.1.1. Work Products Examinable by Static Testing

- Almost any work product can be examined using static testing. e.g. include SRS docs, code, test plans, test cases, product backlog items, test charters, project docs, etc.
- Any work product that can be read and understood can be the subject of a review.
- 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., 3rd party executable code due to legal reasons).

3.1.2. Value of Static Testing

- Can detect defects in the earliest of the SDLC. It can also identify defects which cannot be detected by dynamic testing.
- Provides the ability to evaluate the quality of, and to build confidence in work products.
- By verifying the documented SRS, the stakeholders can also make sure that these SRS describe their actual needs.
- Communication will also be improved between the involved stakeholders.
- Even though reviews can be costly to implement, the overall project costs are usually much lower than when no reviews are performed
- Code defects can be detected using static analysis more efficiently than in dynamic testing

3.1.3. Differences between Static Testing and Dynamic Testing

- 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 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)

3.1.3. Differences between Static Testing and Dynamic Testing

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

- Defects in requirements
- Design defects
- Certain types of coding defects (e.g., undefined, undeclared variables, unreachable or duplicated code, excessive code complexity)
- Deviations from standards
- Incorrect interface specifications
- Specific types of security vulnerabilities (e.g., buffer overflows)
- Gaps or inaccuracies in test basis coverage (e.g., missing tests for an acceptance criterion)

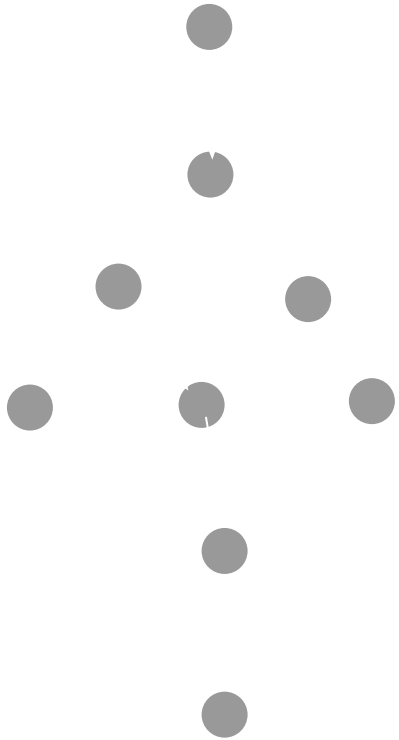
Control flow analysis /1

Aim

To find defects caused by wrong construction of the program code (dead branches, dead code etc.)

Method

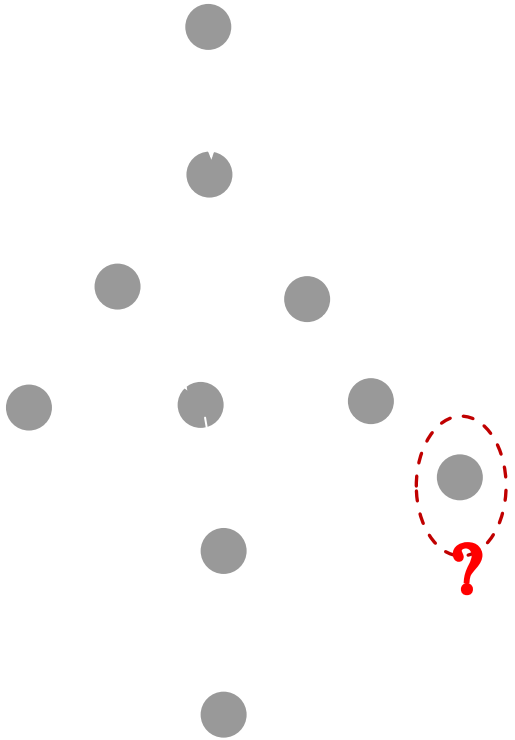
- Code structure is represented as a control flow graph
- **Nodes** represent statements or sequences of statements
- **Edges** represent control flow transfer



Control flow analysis /2

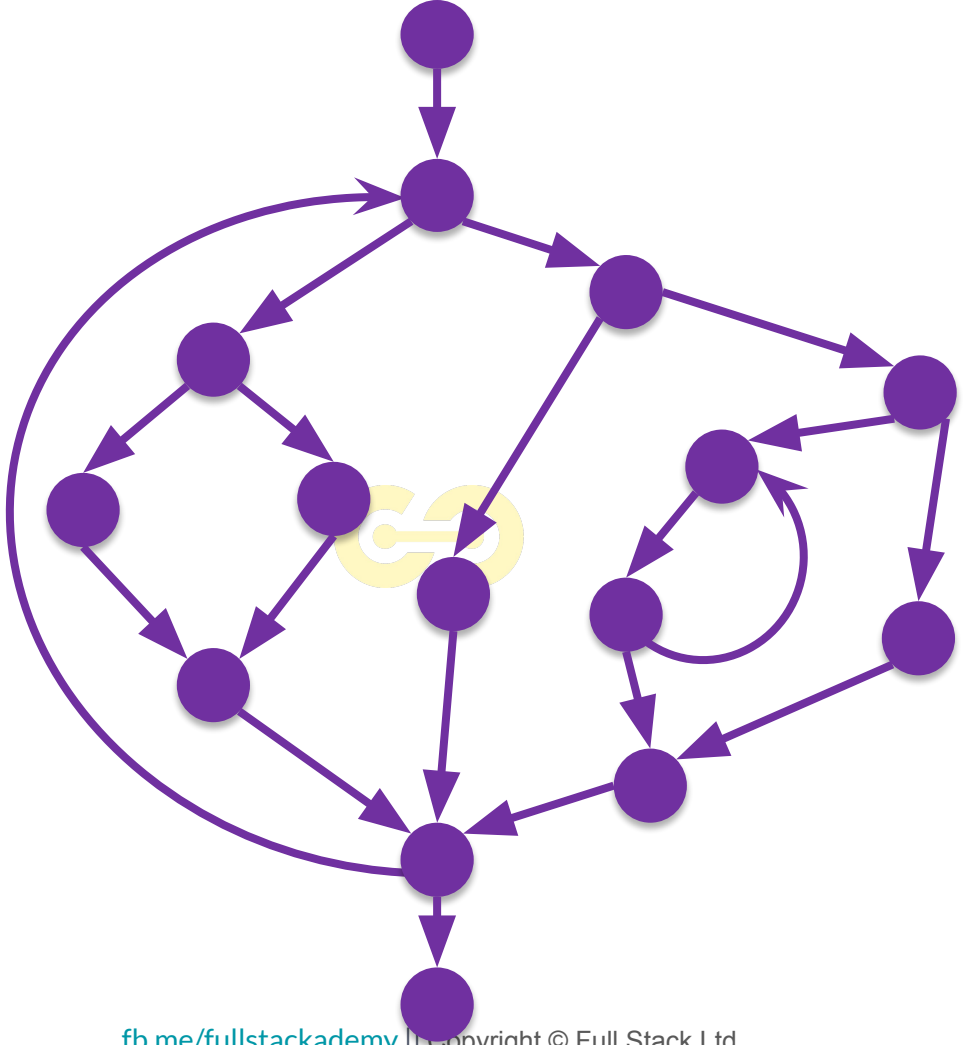
Target

- easy understandable overview of program code
- anomalies can easily be detected, defects stick out
  - loops exited by jumps
  - dead branches
  - multiple returns
- a control flow graph is just a simplified version of flow chart



Cyclomatic Complexity

- Metric to measure the static complexity of a program based on its control flow graph
- Measure linear independent program paths, as an indication for testability and maintainability
- The cyclomatic number is made up of
  - Number of edges (E)
  - Number of nodes (N)
  - Number of inspected independent program parts (P) (mostly 1)
- Values up to 10 are acceptable. Beyond this, code should be reworked/improved (best practice, McCabe)



Cyclomatic Complexity

1 program part: P = 1

15 nodes : N = 15

20 edges : E = 20

The number to a cyclomatic complexity

CC = E - N + 2 * P

CC = 7

Cyclomatic Complexity - implication

The cyclomatic complexity can be used as a target value for code reviews

The cyclomatic complexity can also be calculated as the number of independent decisions plus one. If both ways of calculation give different results, it may be due to

- A superfluous branch
- A missing branch

The cyclomatic complexity also gives as indication for the number of necessary test cases (to achieve decision coverage)

3.2. Feedback and Review Process

3.2.1. Benefits of Early and Frequent Stakeholder Feedback

- Allows for the early communication of potential quality problems. If there is less involvement of stakeholder during the SDLC, the product might not meet the stakeholder's requirements.
- 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.
- Can prevent misunderstandings about requirements and ensure that changes to requirements are understood and implemented earlier.
- 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

- **Planning**
- **Review initiation**
- **Individual review**
- **Communication and analysis**
- **Fixing and reporting**

3.2.2. Review Process Activities

Planning:

During the planning phase the followings will be defined –

- the scope,
- the purpose,
- the test basis to be reviewed,
- quality characteristics to be evaluated,
- areas to focus on
- exit criteria,
- effort and the timeframes for the review.

3.2.2. Review Process Activities

Review initiation:

During review initiation, the goal is to -

- make sure that everyone and everything involved is prepared to start the review.
- every participant has access to the work product under review, understands their role and responsibilities and
- receives everything needed to perform the review.

3.2.2. Review Process Activities

Individual review:

Every reviewer performs–

- 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.
- Reviewers log all identified anomalies, recommendations, and questions.

3.2.2. Review Process Activities

Communication and analysis:

- analyze all these anomalies 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.

3.2.2. Review Process Activities

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.
- review results are reported.

3.2.3. Roles and Responsibilities in Reviews

Reviews involve various 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** (facilitator) – effective running of review meetings, including mediation, time management, and a safe review environment in which everyone can speak freely
- **Scribe (recorder)** – records anomalies, 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

3.2.4. Review Types

- Review types ranging from informal to formal.
- Review formality depends on factors such as –
  - SDLC being followed
  - maturity of the development process
  - criticality and complexity of the work product.
- 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.
- The selection is not only based on the objectives, but also on factors such as –
  - project needs
  - available resources
  - work product type and risks
  - business domain
  - company culture

3.2.4. Review Types

Some commonly used review types are:

- Informal review
- Walkthrough
- Technical Review
- Inspection

3.2.4. Review Types

Informal review:

- does not follow a defined process
- does not require a formal documented output
- the main objective is detecting anomalies

3.2.4. Review Types

Walkthrough:

- 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.

3.2.4. Review Types

Technical Review:

- is performed by technically qualified reviewers and led by a moderator.
- The objectives of a technical review are
  - to gain consensus
  - make decisions regarding a technical problem
  - to detect anomalies
  - evaluate quality and build confidence in the work product
  - generate new ideas
  - to motivate and enable authors to improve

3.2.4. Review Types

Inspection:

- As inspections are the most formal type of review,
- they follow the complete generic process.
- 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.
- 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 and participants to achieve the given objectives
- Reviews on small chunks, so that reviewers do not lose concentration during an individual review and/or the review meeting
- Providing feedback from reviews to stakeholders and authors so they can improve the product and their activities
- Providing adequate time to prepare for the review
- 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

Thank You

