ISTQB Foundation: Static Testing

Overview:

* Static vs. Dynamic Testing
* Roles and responsibilities
* Review Types
* Review techniques

Both static and dynamic testing seek to assess and improve quality, identify defects early. They are complimentary as they find different defects.

Intro to static testing:

* What this works on?
* Why it matters
* How does static testing differ from dynamic

Static Testing

* Unlike dynamic testing the code is never run
* Used early in the lifecycles
* A tool-driven evaluation of work products with structure
* Examines things before they go wrong, includes reviews and tool driven analysis w/o running the code
* Pros:
  + Identifies issues before they make it to production and become costly/dangerous, cheaper to remove
  + Defects should be thought of as a ‘product recall’- a great deal of work from updating the user help files to confirmation and regression testing
  + Static testing is an upfront investment
  + Improves productivity
  + Prevents defects
  + Reduces costs
  + Code does not have to be executed (can save time)
  + Defects are found in work products with less effort and cost
  + Dormant defects can be identified
    - Defects which are on a rarely exercised path and is hard to reach which may be impossible to simulate
  + Allows the team to improve the consistency of internal work products and documentation (internal focus compared to dynamic’s external focus)
  + Testers are unlikely to think of everything – eg. The banking app in which a customer incurred a penalty charge for being overdraw on an old account when they transferred their money out of it – this could be found with a user story
  + Finds defects in design such as inefficient algorithms or database structures, good practice?
  + Finds incorrect interface specifications eg. puts something mi when it should be km
  + Finds security vulnerabilities – buffer overflows, for instance
  + May find gaps in test coverage – tests should cover all acceptance criteria
  + Maintainability defects can **only** be found with static testing
  + Allows for better team communication with a clear structure
* Static Method:
  + Includes reviews – examines design documents, requirement specifications
  + Small modules analysed manually, large ones with tools
  + Manual Techniques (NTK for Exam):
    - Informal reviews
    - Walkthroughs
    - Formal Technical reviews
    - Software inspection
      * Although useful code inspection is not as effective as early interventions in the lifecycle
  + Tool Based Techniques:
    - Used for important safety critical systems, security testing
    - Static analysis is built into the automated build and delivery systems, especially for those systems which require regular testing
* Work products are things which can be examined through static testing – if a document can be read and understood it can be examined statically
* Examples of Software Work Products:
  + Business requirements, architecture and design specifications, epics, user stories, acceptance criteria, code, webpages, user guides

Example:

What kind of issues might be detected in static testing?

* For instance in reviewing requirements:
* Text

  Description automatically generated with medium confidence
* Qs.
  + If it is a new account – what do the old ones pay?
  + Does the new account need to share characteristics with the old one?
  + Is the interest tax free? Monthly or annual interest?
  + A bonus *may* be paid, but under what criteria? Does it apply to new customers?
  + Between? Does that include the values 5000 and 10000?

In static testing you should review the design before writing any code. Also has the advantage of allowing devs to know what is expected of them, clear easy to follow rules. Can reduce unit testing time.

Example: A bank tried to launch an app which enabled ‘joint accounts’, but failed to realise this could mean more than two accounts, it would have been cheaper had the requirements been reviewed earlier and the term ‘joint account’ clarified.

Table

Description automatically generated

Above: Column one is for benefits that improve productivity, column 2 is for benefits which prevent defects and column 3 is for cost savings

Dynamic Testing:

* When carried out:
  + Test conditions must be identified
  + Test cases written
  + Test data must be sourced
  + A test environment needs to be configured
  + Tests must be run against the code and observed
  + Results must be reported
* Focuses on the external observable behaviours of the software, not the quality of the code

Summary:

* You should be able to list work products, including specifications and code that can be statically reviewed
* Value of early testing
* List the values of static testing
* Static does not require the code be run
* Know the advantages of Static over Dynamic and that each identifies different defects

The Review process

* Used in many lifecycle types
* The activities of a formal review process

Why they matter:

* Examine work products early in the lifecycle and raise issues – which can occur anywhere and can be potential defects, recommendations or questions

Common Review types:

* Informal review
  + Identifies potential defects, generates ideas without a defined process or generating outputs/paperwork
* Walkthrough
  + Usually led by the author of the work product, aims to find defects, to consider alternative solutions
* Technical review
  + Team effort, technical peers and other technical experts look to gain consensus and identify issues
* Inspection
  + The team must perform a detailed examination of a work product with clearly defined roles to identify defects, build confidence and uses root cause analysis

Table

Description automatically generated with medium confidence

* Informal reviews tend to happen on agile projects
* Complex products in developed companies tend to result in formal reviews – may require an ordit trail and legal obligations

The Review Process:

* Planning
* Initiation
* Preparation (or individual review)
* Issue communication and analysis
* Fixing and reporting

Planning a Review:

* First: Define the scope of a review:
  + What is it’s purpose?
    - Ie. ‘To prevent defects in system A going into production and to get the new developers learning’
  + What is to be reviewed?
    - Ie. The specification
  + Identify quality characteristics
    - Ie. Functionality, usability
* Second: Review Characteristics:
  + The review leader decides who takes part in the review and allocated roles – they should choose a mix of skills and experiences, as in the expression ‘something old (experienced), something new (learning), something borrowed (independent, maybe from another department), something blue (a critic)’
  + Decide what type of review is appropriate eg. Formal technical review
  + What activities? Eg. Individual reviews, review meeting
  + Assign roles for each participant
  + Create checklists for the new products
* Third: Estimating effort and timeframe
  + You do this looking at the size of the product,
  + The rate at which it is examined
  + The time spent – ther more time given the more indepth the analysis should be
* Fourth: Exit and Entry criteria for formal reviews:
  + Entry Criteria: Should we begin the review?
  + The entry criteria should be checked prior to the review so you don’t waste people’s time on unnecessary reviews for an unfinished product
    - Should answer questions like: Is the author ready? Does a quick scan reveal issues? Are the pages numbered and formatted?
  + Exit criteria:
    - Until the exit criteria are met the work product can not be passed onto the next stage

Initiating the Review

* Everyone on the review team must be formally notified that it is about to happen
  + Often done with a workshop or virtual meeting brief
* In this stage a review leader must:
  + 1. Distribute the work product and supporting material
  + 2. Explain the process of the review, i.e. scope, objectives of the review, roles, etc.
  + 3. Question and answer session

Individual Review/preparation

* Notes potential defects
* Makes recommendations
* Notes your questions

Issue Communication and analysis

* Begins with a meeting in more formal reviews to communicate issues
* Everyone on the team should contribute their perspectives
* Notes potential defects
* Provides recommendations to the author
* The author answers questions from the reviewers
* The debate often yields more issues, BUT potential solutions should NOT be discussed – the focus should be on owning defects and assigning ownership and status to them
* Quality characteristics – essential to the success of the finished product, performance and usability are high on this list for customers, ease of maintenance, portability, etc.
  + For QCs one should:
    - Evaluate the issues
    - Document the findings
* In tech reviews consensus is important, but in inspection one must check the exit criteria
* Three potential outcomes:
  + The team accept the work product as it is
  + Accept it with minor changes
  + Reject – major changes and further review necessary
* 4 aims:
  + Make everyone aware of issues
  + Assign ownership
  + Evaluate QCs
  + Make review decisions

Fixing and Reporting

* Addresses the issues found during review
* Consists of:
  + Creating defect reports for anywhere change is needed
  + The author will fix defects
  + The leader will communicate defects and update defect status
  + Metrics may be gathered for the review process if formal
  + The exit criteria must be met
  + Accepting the work product

The five stages of the review process:

1. Plan the review
2. Initiate the review
3. Individual review/preparation
4. Issue communication
5. Fixing the bug

Reviews are static testing techniques use early in the lifecycle to identify potential defects in work products

Roles and responsibilities in a review:

* Author
  + Creates the work product and addressed the issues
* Facilitator
  + Runs effective meetings and mediates opposing views, aim is to identify issues not provide solutions
* Scribe
  + Records issues found in the review
* Review leader
  + Take overall responsibility for the review, selects participants, organises the where and when
* Reviewers
  + Bring their own perspective to the project, depending on their specialty. They may be technical, project staff, business experts, etc.
* Management
  + Responsible for review planning – determines the type of review strategy, allocates resources, monitors cost effectiveness of reviews
* You can remember these as L.FARMS

Module Summary:

* Early reviews save money
* There are 4 review types: Informal, walkthrough, technical review, inspection
* The review process: Plan, initiate, individual review, issue communication and analysis, fixing
* The six roles in a format review: Leader, facilitator, author, reviewers, management, scribe

Review Types Compared:

Overview:

* Review types: Informal, walkthrough, technical review, inspection
* Compare review types in terms of:
  + Purpose
  + People involved
  + Formality of process
  + Documentation

Informal Review:

* Undocumented or very little, fewer people involved, clarity or improvement or generating ideas, no process

Walkthrough

* A step by step process to walk colleagues through the work products
* Purpose – to improve the product – perhaps considering changing the design
* People – the author and colleagues
* Formality – medium
* Documents -
* The author prepares presentation for colleagues
* Pros:
  + The author alone does the work
  + Allows for a large group with diverse viewpoints
  + More likely to identify major issues with large group
* Cons:
  + Boring if group is too large – too slow for experienced, too fast for inexperienced
  + Questions should seek only clarification – note taking could be a distraction
  + Author may get defensive if the review leader attempts to make changes
* May involve scenarios, dry runs or simulations
* Graphical user interface, text, application

  Description automatically generated

Technical Reviews:

* Purpose – To gain consensus, find defects, build confidence, generate new ideas and improvements
* People - The Goldilocks principal – you want a team not too big not too small, the reviewers will be technical peers of the author
* Process
  + Formal, meeting is usuall
  + Panning, initialtion, individual review, issue communication (will not have a meeting if too many or too few issues), fixing bugs – the easy ones can be immediately seen too
* Post-review:
  + Creates a defect report
  + Only major changes require further review processes

Inspection:

* Formal defined process
* Diagram

  Description automatically generated
* More likely to be used on a traditional life cycle with safety critical systmes
* Inspection teams are smaller and some people may have multiple roles, peers of the author or tech experts
* Team Roles:
  + Graphical user interface, text, application

    Description automatically generated
  + Meetings should be run by a facilitator not the auther
* Entry requirements must be met before a review to ensure it is justified and the product is complete and worth spending time on
  + Examples of entry criteria:
    - A black and white sign

      Description automatically generated with low confidence
* **Inspection Rules** - Ensures best practice – inspection rules are met, the inspection rules are kept up to date and enable reviewers to judge the work product – there are general rules for the organisation and ones specific to the product. Eg. All documents must be unambiguous
* **Inspection Checklists** – derived from the rules, have a question format, are used by reviewers, eg.
* Graphical user interface, text

  Description automatically generated
* **Inspection Optimum Checking Rate -**  Are we reviewing efficiently? Issues should be logged at a constant rate, a logging rate. Time should not be spent on solutions. This depends on the number of documents, checking rate (pages per hour) and the hours spent. The leader should set the optimum rate for the team and thus the time for individual review
* **Inspection Metrics** – Used to help improve the inspection process. The leader reports to management including:
  + Checking rate per reviewer
  + Loggin Rate
  + Remaining defects per page
* **Inspection Preparation/individual review** – uses checklist and assigned checking role, works at the required optimum checking rate so they cover each page in the right level of detail
* **Inspection Exit Criteria** – must be met before approval, example exit criteria:
* **Graphical user interface, text

  Description automatically generated**
* Purpose: To find defects
* People – small teams
* Process – formal, defined
* Post-Review – metrics reported

Review Type Selection:

* How to select them?
* Selection Criteria
  + Company culture
  + Business domain
  + Project needs
  + Available resources
* Type of work product and Risks are moxt important determining factors
* Formal reviews for safety critical
* Informal – no meetings or prep
* Walkthroughs – allows understanding
* Tech review- author attends but does not moderate meeting
* Table

  Description automatically generated
* In tech and inspection reviews moderator led

Review Module 4:

* All review types identify potential defects
* The team learns and improves through review
* Compare review types: Purpose, process, people, post review
* Many review types may be used
* Agile tend to be informal review
* Selection criteria

Module 5.

Applying Review Techniques

* Reading techniques allow you to find errors better
* Reviewing the PRODUCT *not* the PERSON
* Techniques for each review type:
  + Ad hoc
  + Checklist based
  + Scenarios and dry runs
  + Role based
  + Perspective based reading
* Success
* factors for a review:
  + Organisational:
    - Clear objectives for a review (done during planning, clear measurable exit criteria)
    - Select appropriate review types for the product and experience involved
    - Suitable review techniques (checklists should be uptodate and include major risk factors)
    - Have large documents reviewed in small chuncks for quality control – early and frequen feedback
    - Adequate notice to allow everyone to prepare – optimum checking rate
    - Management must allow adequate time for review
  + People related
    - A leader must insure they include the right people in the review – different experiences and skills
    - Testers should be valued as reviewers
    - Reviewers should not rush the process
    - Reviews should be done in small chunks to prevent loss of concentration
    - Should not be made an emotional issue – a product not person
    - Well managed meetings to ensure well managed meetings are not a waste of time – time should be managed, unnecessary criticism avoided, etc.
    - Atnosphere of trust
    - Adequate training – for formal review types especially
    - A culture of learning
  + Review techniques
    - Used by a reviewer in preparation to review the issues of a product
    - How effective a technique is may depend on the type of review used
    - See below for more details

Ad hoc reviews – rely on the reviewer’s experience

Checklist-based – A list provided of frequent errors

Scenarios and Dry Runs – reading the document based on expected usage

Role based – Encourage different stakeholder views of the software

Perspective -based reading– extends role based

Ad-hoc

* a one off, for this specific issue
* Preparation is minimal
* Relies on reviewer expertise
* No guidance for how to review, so read sequentially start to finish – invites boredom and missing things
* Duplicate issues reported by diff reviewers

Checklist

* List of frequent errors – updated from past project
* Systematic technique
* Specific to the type of work product, must be maintained
* Common defects covered
* Reviewer should ensure reviewers do not *just* rely on the checklist, it is for guidance only
* Derived from rules under Inspection
* One checklist per work product
* Updated when new defects are found, when they are outdated (when devs are so used to them not relevant checks any more)
* A checklist should be one page only

Simulations and dry runs

* Often done in walk throughs
* Reviewer provides structured guidelines on how to read through the work product
* Allows for dry runs based on the expected returns, more specific to this project
* More indepth than checklists
* Less constrained to documented scenarios – but what is the customer did this?

Role Based Techniques

* Evaluate the work product from the perspective of stakeholder roles
* These should include use by:
  + An experienced user who want shortcuts and new functionality
  + A senior citizen
  + A child
  + A notice or infrequent user – otherwise they will never be a frequent user
  + Developers, testers, performance and system admin

Perspective based Reading

* The technical aspects of a review
* Need to understand past defects and target accordingly
* Devs reading the work of devs
* Widely applicable and customisable – non-specific process
* Taylor the perspective for each reading and write questions designed to find defects
* Reviewers each have stakeholder roles – how someone will use a work product determines how important the info is to that user
* You use the work product with that person’s perspective –
* more indepth, less duplication of issues across reviewers because all have different roles, makes a team less reliant on one reviewers expertise
* Aim is to create a high level version of the work product typical for what the user normally uses – so a reviewer taking on the role of designer would seek a high level system design, a tester role requires a system test plan, customer role – enumerating the described functionality
* Perspective based reading provides concrete instruction to help you uncover defects

Exercise:

Text

Description automatically generated

Summary for the Module:

* Success factors for reviews
  + Organisational and people-related
* The review techniques
  + Ad-hoc, checklist based, scenarios and dry runs, role-based, perspective based