

08 Software Testing

INT210 Architecture, Integration and Deployment

Why do we want to test software?

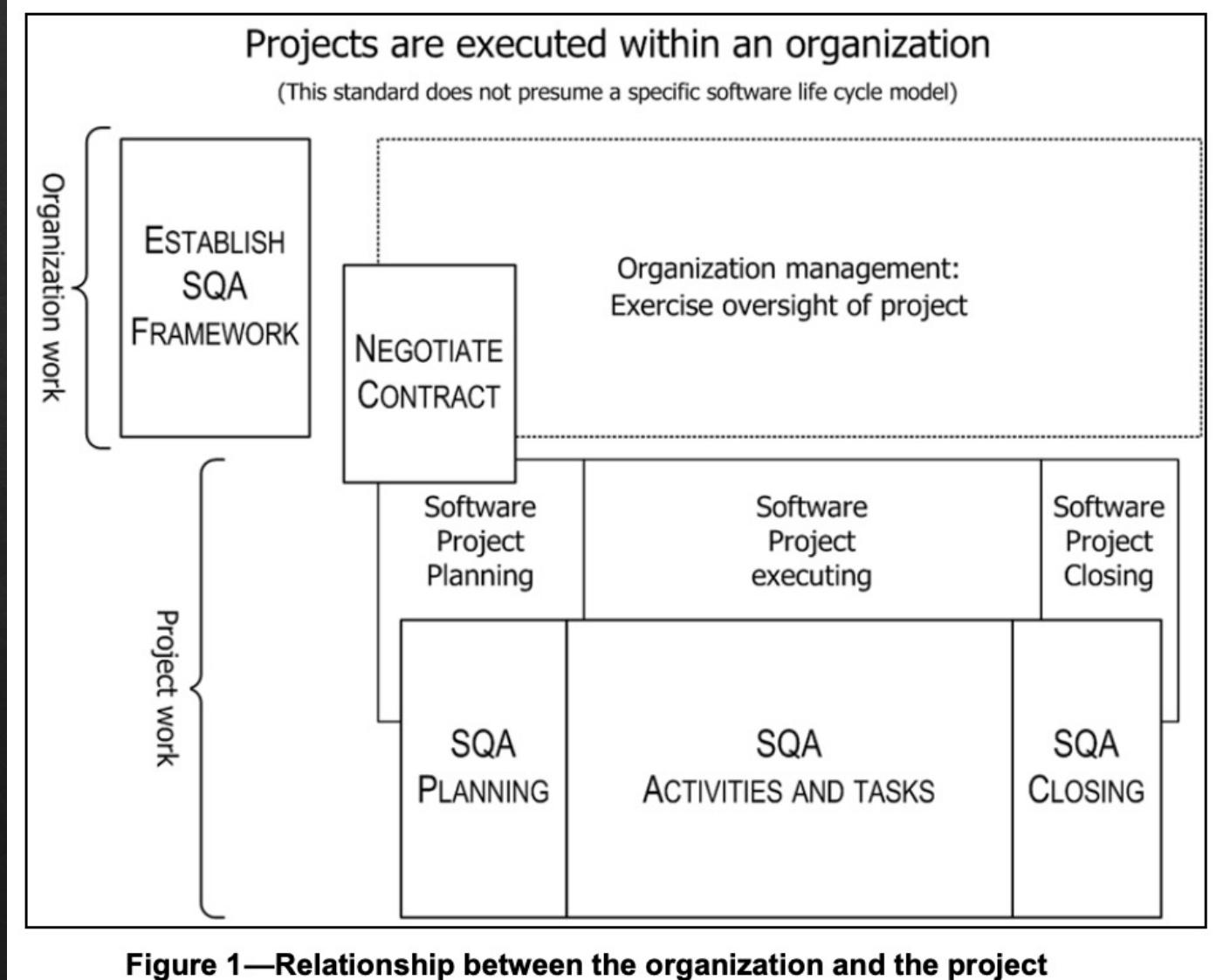
- ❖ What are we testing for?

Goals of testing

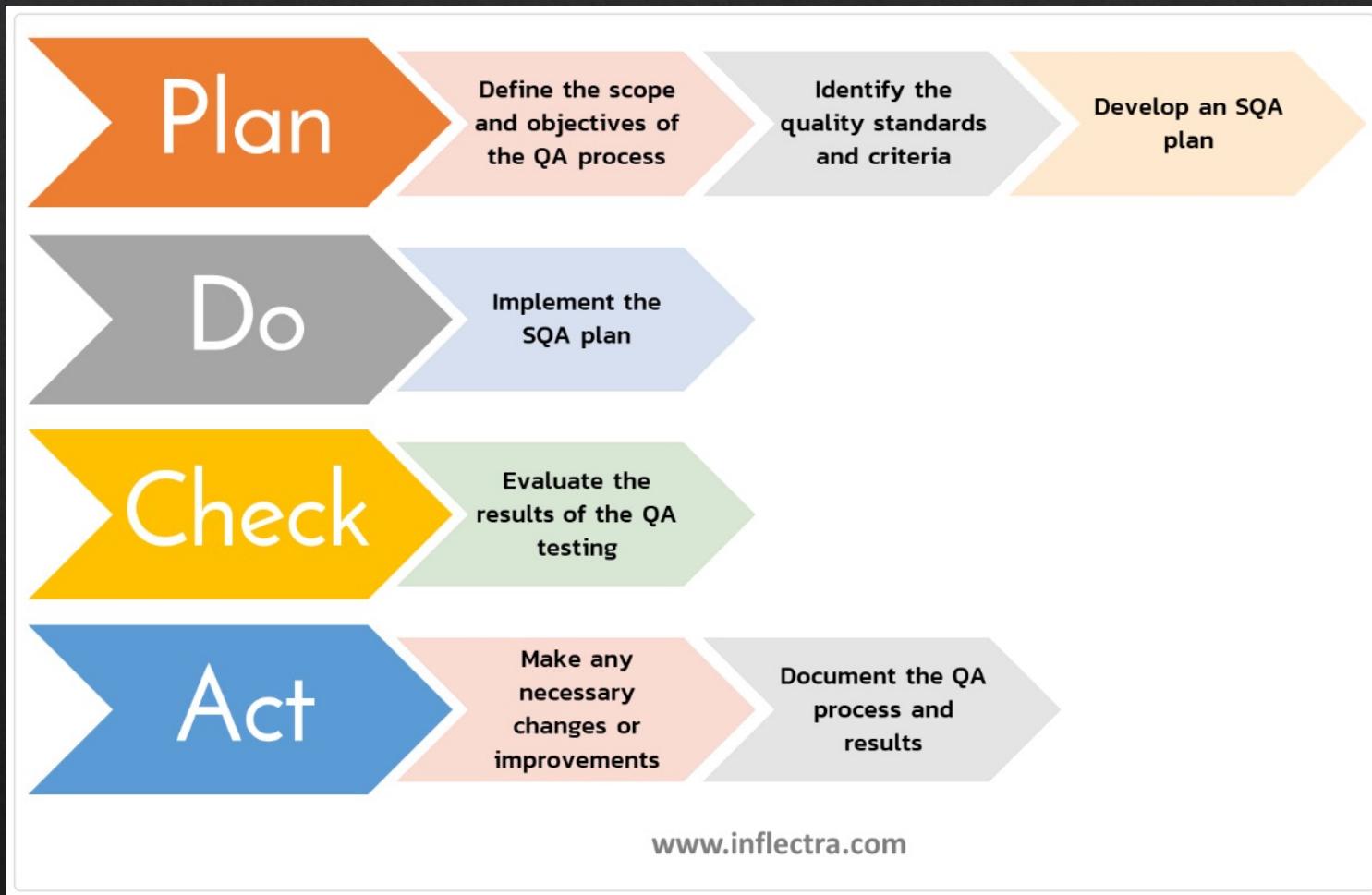
- ❖ Determine correctness, under certain assumptions/conditions
- ❖ Find faults and failures << to fix
- ❖ Make sure that SW works under all conditions?
 - ❖ All input combinations?
- ❖ Static test << review code
- ❖ Dynamic test << test running software
- ❖ Pass/fail criteria are determined by stakeholders << test results

Software Quality Assurance

- ❖ IEEE Std 730™-2014 (Revision of IEEE Std 730-2002)
- ❖ Quality activities throughout project



SQA Process



A definition (WikiPedia)

- ❖ **Software testing** is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. https://en.wikipedia.org/wiki/Software_testing
- ❖ Validation – are we building the right product? Does SW satisfy user requirements/needs?
- ❖ Verification – are we building the product right? Does SW conform to requirements (e.g. correctness, completeness, consistency, accuracy) [IEEE Standard for SQA Processes](#)

SW verification

- ❖ Are we building the product right?
- ❖ Review artifacts
 - ❖ Money deposit?
 - ❖ Class registration?
 - ❖ Sending a message?
 - ❖ Code review?

SW validation

- ❖ Are we building the right product?
- ❖ Does SW meet user requirements?
 - ❖ Which users?
 - ❖ Who else?
- ❖ Who should perform the validation?
- ❖ When should validation be done?
- ❖ Does validating the Requirement Specification count?

Software Testing Process

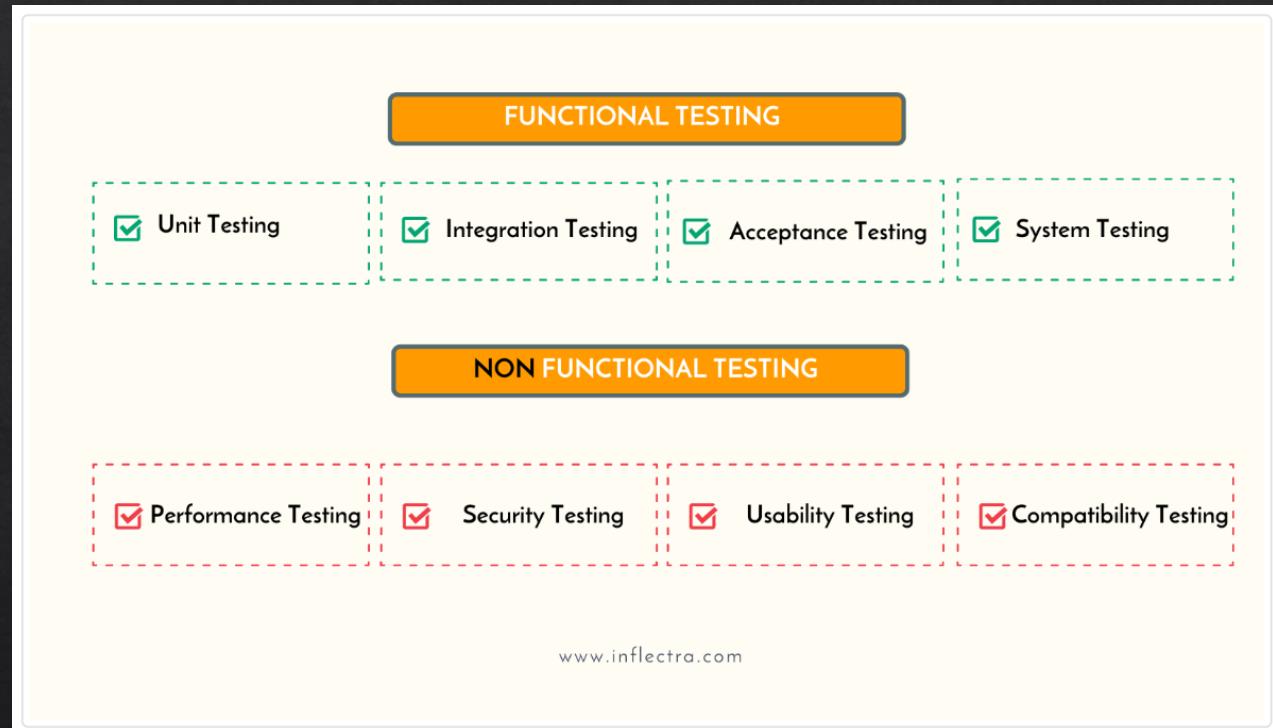
- ❖ Manual - team or individual will manually **operate the software product** to ensure it **behaves as expected**
- ❖ Automated – many different tools, test code, test logic, performance, etc., and simulate human testing, e.g. JUnit, Cypress, Selenium



<https://www.atlassian.com/continuous-delivery/software-testing>

Types of SW Testing

- ❖ Unit tests
- ❖ Integration tests
- ❖ Functional tests
- ❖ End-to-end tests
- ❖ Acceptance testing
- ❖ Performance testing
- ❖ Smoke testing
- ❖ Exploratory testing



[Atlassian](#)

[inflectra](#)

[WiKi - UnitTestTools](#)

Quick description (inflectra)

- **Unit testing** - performed on individual units or components of the software to verify that they function as intended.
- **Integration testing** - performed on multiple units or components of the software to make sure that they work together properly.
- **System testing** - performed on a complete application or system to confirm that it meets the specified requirements and functions expected of it.
- **Acceptance testing** - determines whether an application or system is ready for release to the public based on its criteria.
- **Regression testing** - ensures that changes or updates to an application or system do not introduce new defects or issues.
- **Performance testing** - assesses the speed, scalability, and reliability of an application or system.
- **Security testing** - verifies that an application or system is secure and protects sensitive data from unauthorized access.

Functional Testing

Functional testing typically involves six steps

1. The identification of **functions** that the software is expected to perform
2. The **creation** of **input data** based on the function's specifications
3. The **determination** of **output** based on the function's specifications
4. The execution of the test case
5. The comparison of actual and expected outputs

Goal: check whether the application works as per the customer need.

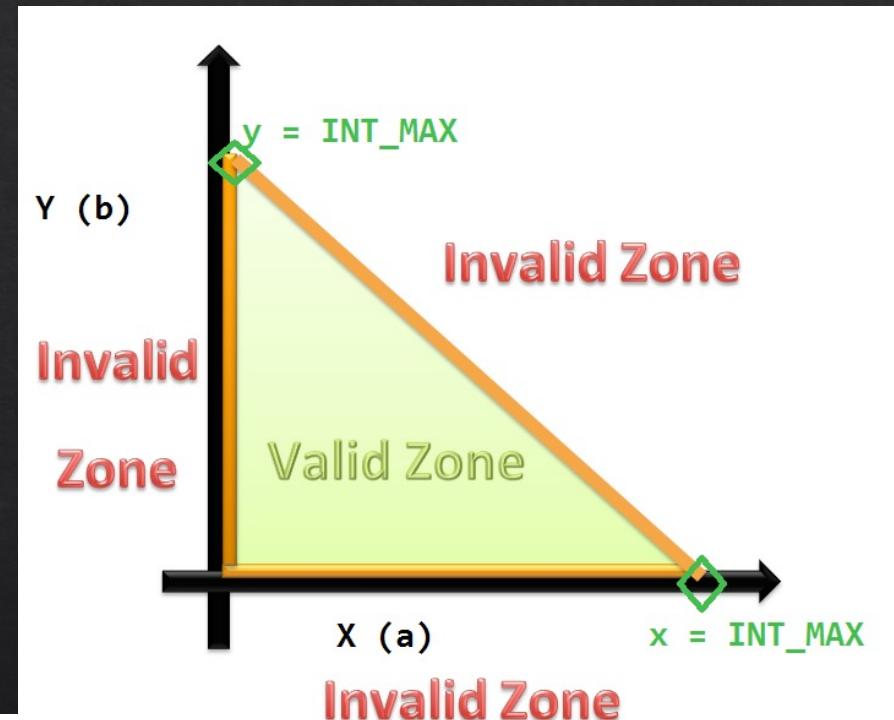
https://en.wikipedia.org/wiki/Functional_testing

—-Box Testing

- ❖ Black-box – test against requirements, do not check code
 - ❖ E.g. Equivalence partitioning, boundary values, decision table
- ❖ White-box – verify internal structure
 - ❖ Used for unit/integration/system testing
 - ❖ Test logic paths, API, code coverage, fault injection
 - ❖ Reviews, walkthroughs, inspections

Equivalence partitioning

- ❖ Test whether $a + b$ has resulted in an overflow
- ❖ What are the classes of test inputs?
- ❖ What are the test values?



Test cases

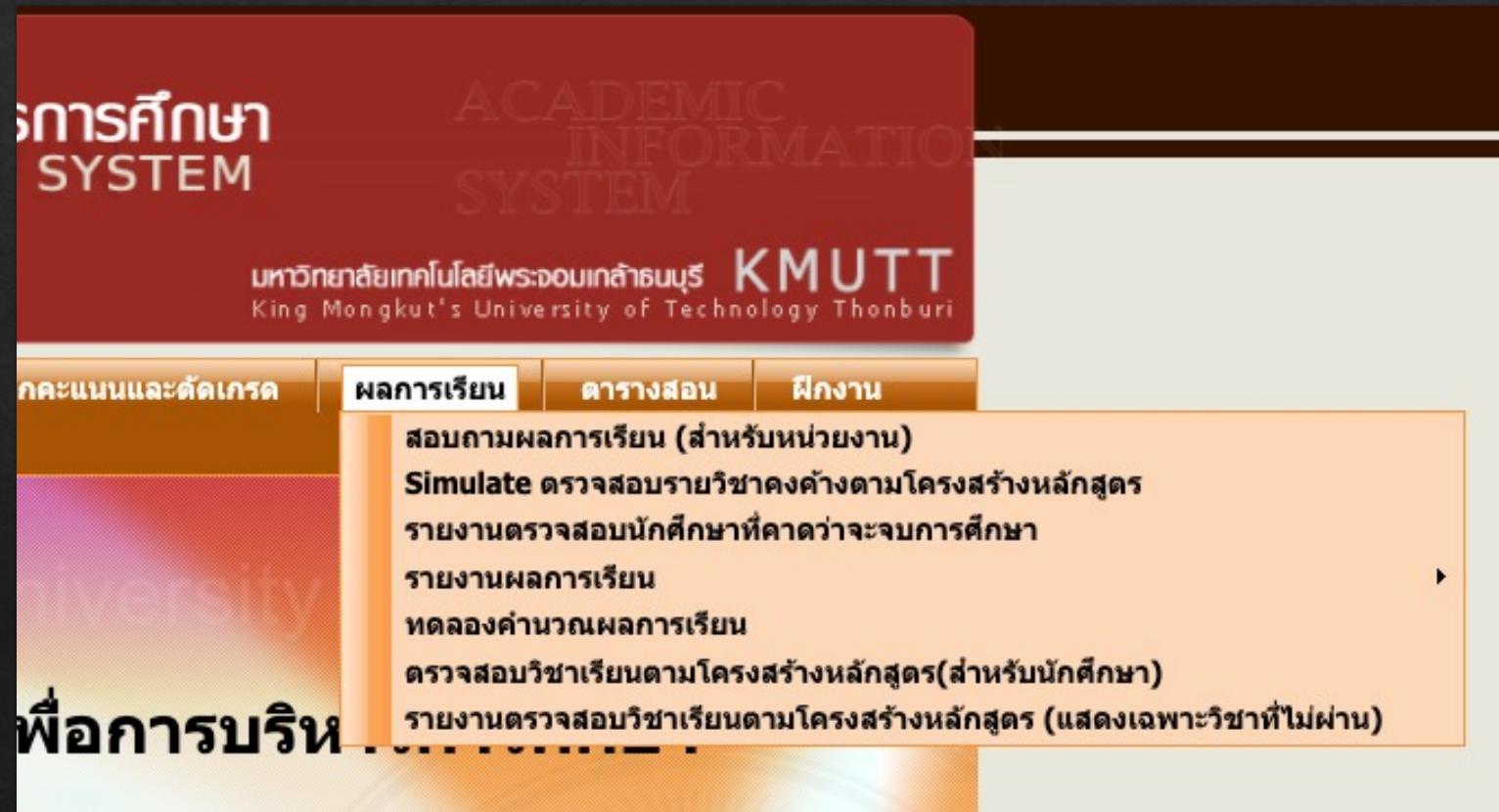
- ❖ Test case No.
- ❖ Description
- ❖ Test condition
- ❖ Output

Assignment: design tests for two functions

Equivalence partitioning

Boundary values

Decision table



9 User Acceptance Testing

INT210 Architecture, Integration and Deployment

Definition

- ❖ Formal testing with respect to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria^[3] and to enable the user, customers or other authorized entity to determine whether to accept the system.
- ❖ — *Standard Glossary of Terms used in Software Testing*^{[4]:2}
- ❖ Validate end-to-end business flow

UAT

- ❖ Business Analyst (BA) prepare a UAT test plan << approved by PM, Business Owner
- ❖ Test cases prepared by BA and approved by business owner
- ❖ May re-run the test multiple times
- ❖ Results: completed tests/scope, overall pass/fail percentages, planned upcoming test cases (if not finished), issues, concerns, blockers, high-level synopsis

<https://uit.stanford.edu/pmo/UAT>

UAT Test Plan

- 1. Introduction
- 2. Testing Scope
- 3. Testing Approach
- 4. Testing Deliverables
- 5. High-Level Schedule
- 6. Error Management & Reporting
- 7. Environmental Requirements
- 8. Roles & Responsibilities
- 9. Risks & Assumptions
- 10. UAT Exit Criteria

<https://uit.stanford.edu/pmo/UAT>

UAT Plan

- ❖ Scope: business requirements will be tested
- ❖ Approach: blackbox, operational, manual, automated
- ❖ Deliverables: code, test data, design, requirement documents, reports, results, etc.
- ❖ Environment: client, server, network, devices, etc.
- ❖ Roles and responsibilities: qualified clients, end users, helpers, interpretation/analysis, approvals

Resources

- ❖ <https://www.altexsoft.com/blog/engineering/user-acceptance-testing/>
- ❖ <https://www.panaya.com/blog/testing/what-is-uat-testing/>
- ❖ <https://drive.google.com/file/d/1cVFI5To4MHO9mowu-kiPXPzvdVdOsQ9s/view> (from Stanford UIT)