

08 Software Testing

INT210 Architecture, Integration and Deployment

Why do we want to test software?

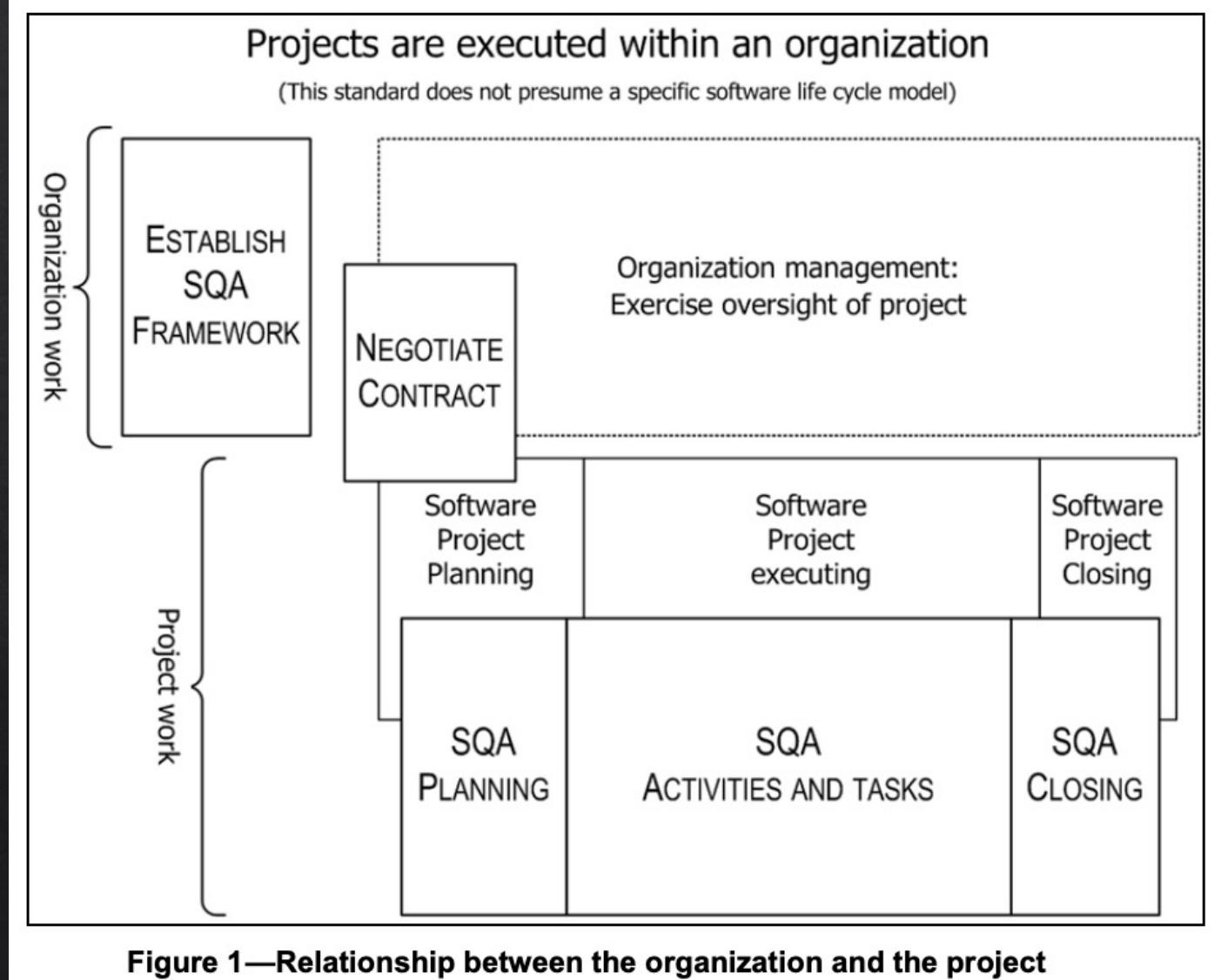
- ❖ What are we testing for?

Goals of testing

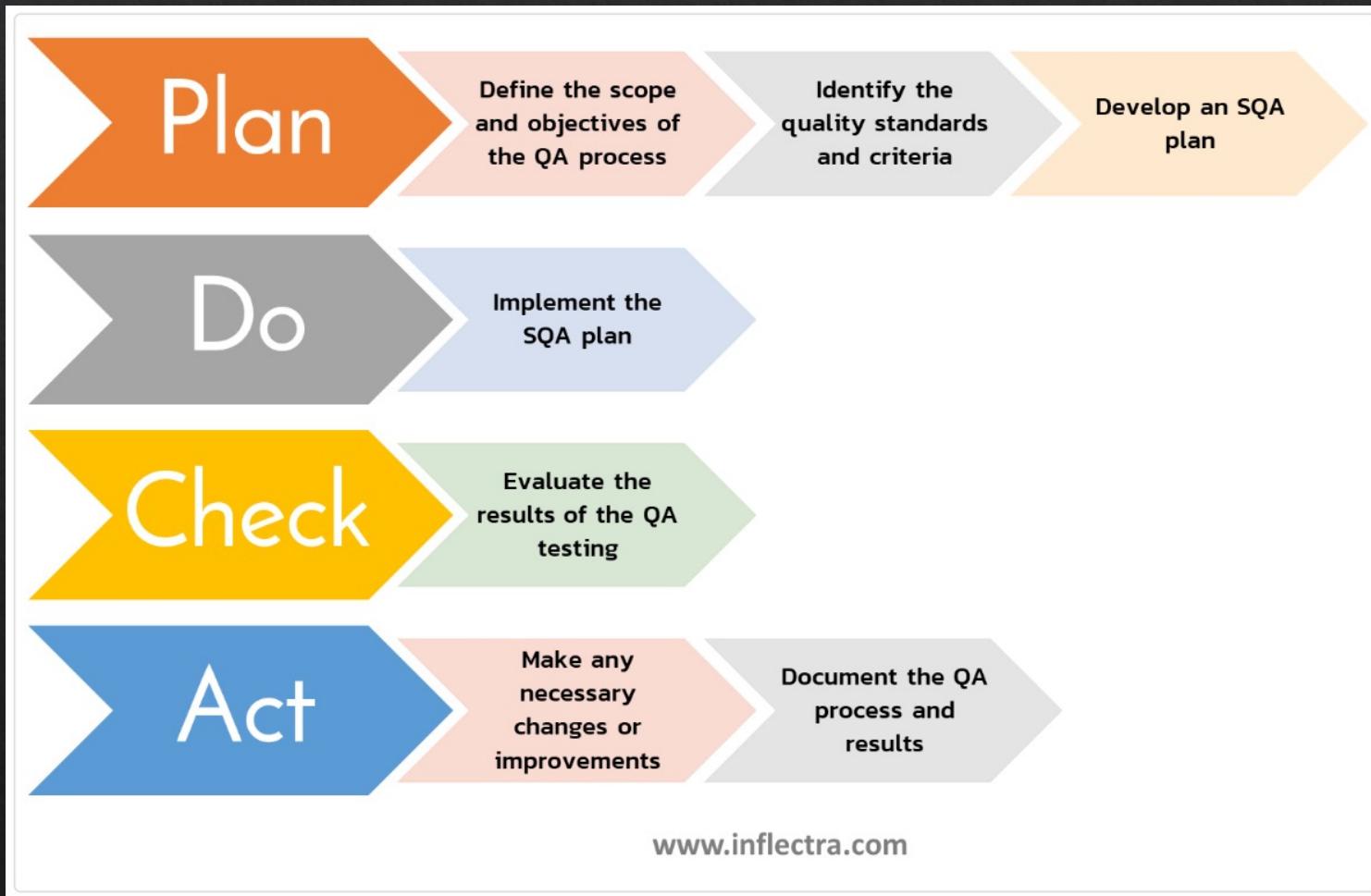
- ❖ Determine correctness, under certain assumptions/conditions
- ❖ Find faults and failures << to fix
- ❖ Make sure that SW works under all conditions?
 - ❖ All input combinations?
- ❖ Static test << review code
- ❖ Dynamic test << test running software
- ❖ Pass/fail criteria are determined by stakeholders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

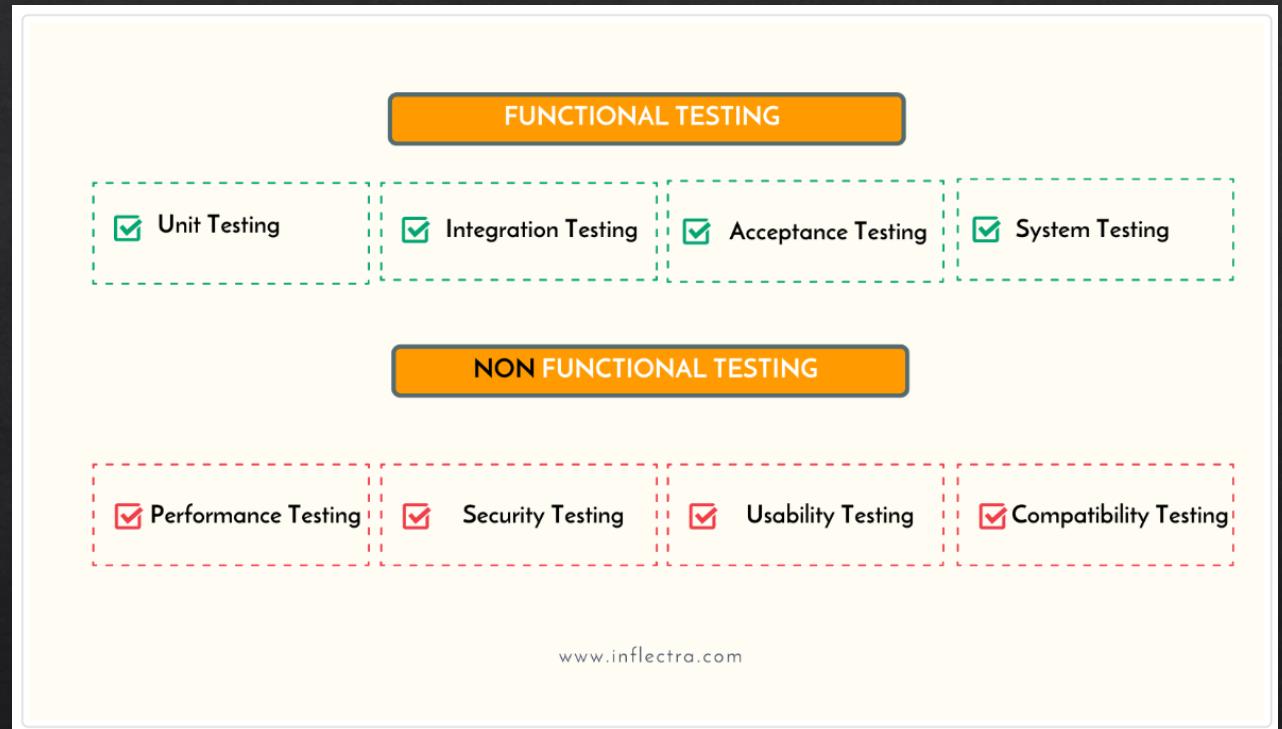
<https://browserbench.org/Speedometer3.0/>



<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
- ❖ 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.

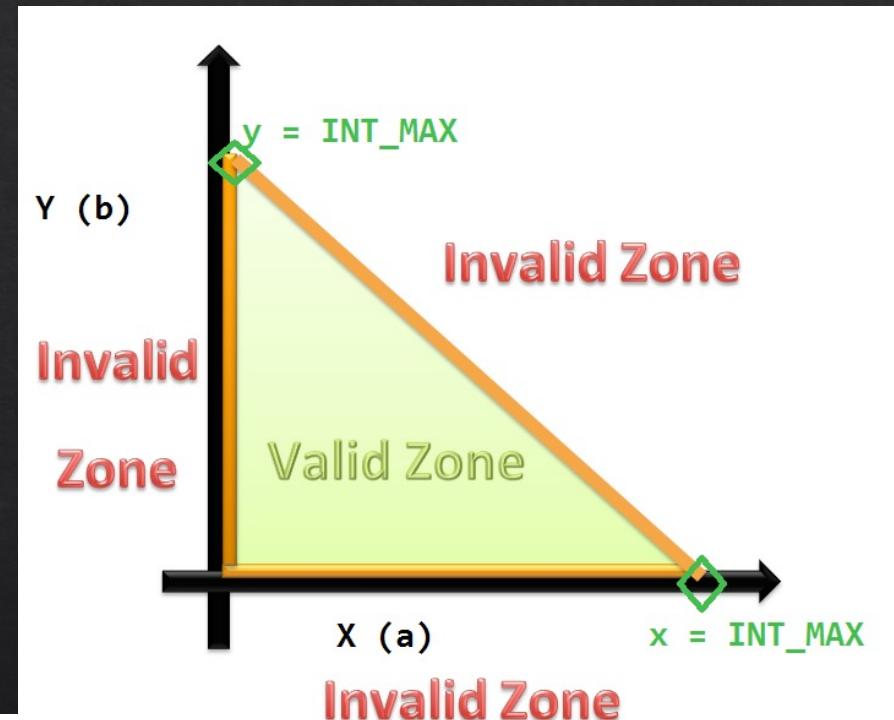
https://en.wikipedia.org/wiki/Functional_testing

—-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?



Equivalence partitioning and boundary values

- ❖ Passwords condition
 - ❖ Valid password class
 - ❖ Invalid password classes
 - ❖ Empty, min length, complexity
- ❖ Student ID number
 - ❖ Valid format and range
 - ❖ Invalid value: not-number, range

Equivalence Partitioning and BVA Benefits

- ◊ Equivalence Benefits
 - ◊ Improved test coverage
 - ◊ save time and resource
 - ◊ Increased probability of detecting defects
- ◊ Boundary value benefits
 - ◊ Efficiency (no redundancy)
 - ◊ Early detection of boundary conditions defects
 - ◊ Correct handling of edge cases

Test cases

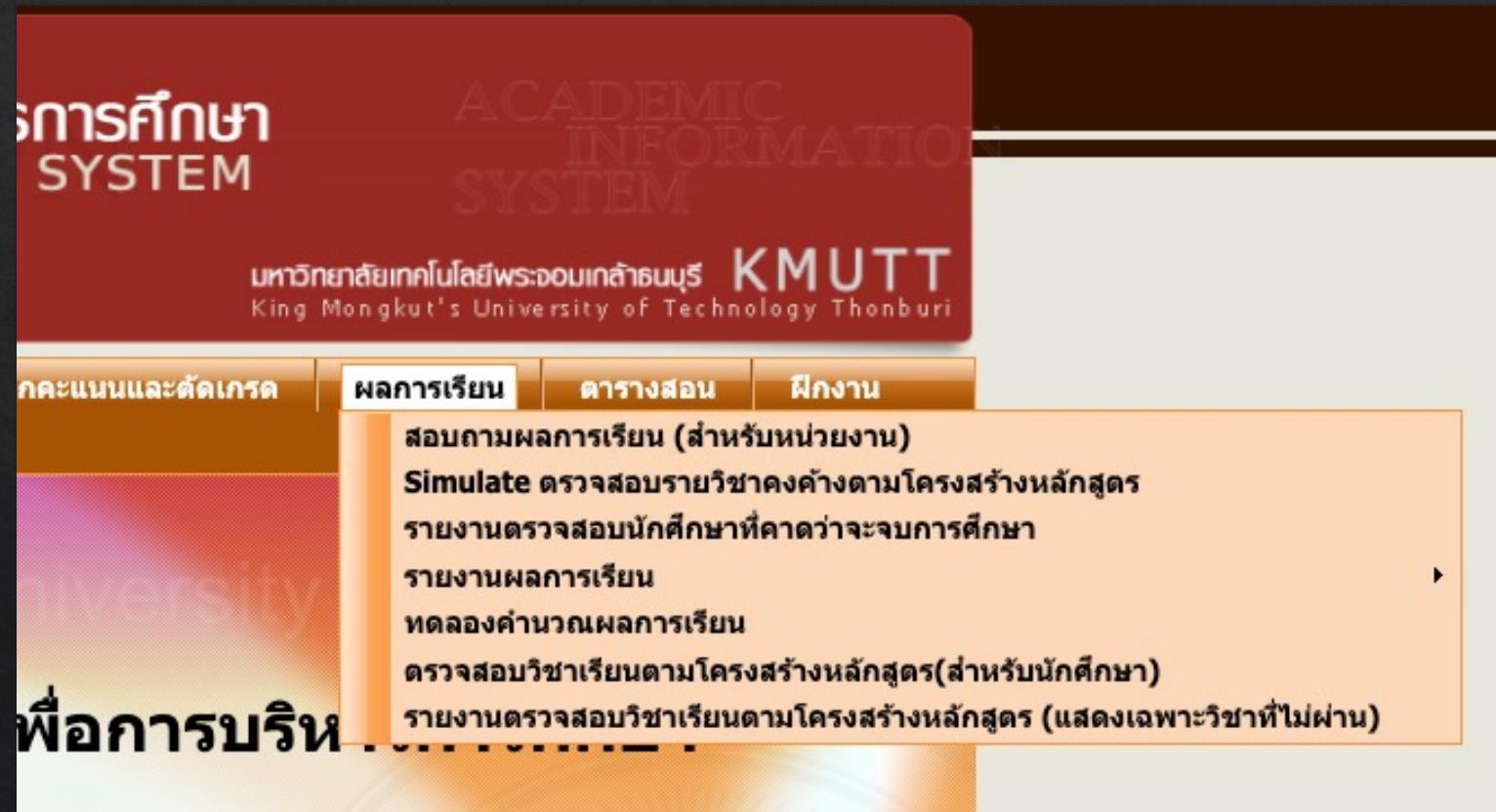
- ❖ Test case No.
- ❖ Description
- ❖ Test condition, procedures, input
- ❖ Expected Output

Assignment: design tests for two functions

Equivalence partitioning

Boundary values

Decision table



Decision table testing

1. Identify all possible Conditions.
2. Identify the corresponding Actions that may occur in the system.
3. Generate all possible combinations of conditions (Business Rules).
4. Identify actions based on the combination of conditions.
5. Use this decision table to generate checklist or test cases for each of these combinations.

Step 2: Identify the corresponding Actions that may

	RULE 1	RULE 2
CONDITIONS		
Customer without loyalty card		
Customer with loyalty card		
Extra discount selected		
\$100 spent		
ACTIONS		
No discount		
Extra discount		
Loyalty points		

<https://www.linkedin.com/pulse/decision-table-testing-benefits-steps-create-liudmila-akulovich>

Requirements

◊ A supermarket has a loyalty scheme that is offered to all customers. Loyalty card holders enjoy the benefits of either additional discounts on all purchases or the acquisition of loyalty points, which can be converted into vouchers for the supermarket or to equivalent points in schemes run by partners. Customer without a loyalty card receive an additional discount only if they spend more than \$100 on any one visit to the supermarket, otherwise only the special offers offered to all customers apply.

Step 3: Generate all possible combinations of conditions (*Business Rules*).

	RULE 1	RULE 2	RULE 3	RULE 4
CONDITIONS				
Customer without loyalty card	T	T	F	F
Customer with loyalty card	F	F	T	T
Extra discount selected	-	-	T	F
\$100 spent	F	T	-	-
ACTIONS				
No discount				
Extra discount				
Loyalty points				

Example 2: Online Store Pricing

Consider an online store that calculates the price of a product based on its category, customer type, and current promotions. The decision table could be as follows:

Condition	Condition	Condition	Action
Product Category	Customer Type	Promotion	Price
Electronics	Regular	None	\$100
Electronics	Regular	Discount (10%)	\$90
Electronics	Premium	None	\$95
Electronics	Premium	Discount (20%)	\$80
Clothing	Regular	None	\$50
Clothing	Regular	Discount (5%)	\$47.5
Clothing	Premium	None	\$45
Clothing	Premium	Discount (10%)	\$40.5

Test cases: Number of registered credits

“ข้อ 15 การกำหนดจำนวนหน่วยกิต ต่อภาคการศึกษาในการลงทะเบียนเรียน

- 15.1 นักศึกษาจะต้องลงทะเบียนในแต่ละภาคการศึกษาปกติไม่น้อยกว่า 12 หน่วยกิต และไม่เกิน 19 หน่วยกิต ยกเว้นกรณีรายวิชาที่ยังเหลือตามหลักสูตรและเปิดสอนในภาคการศึกษานั้นมีหน่วยกิตรวมกันต่ำกว่า 12 หน่วยกิต หรือในกรณีที่หลักสูตร หรือโครงการกำหนดไว้เป็นอย่างอื่น ส่วนในภาคการศึกษาพิเศษจะลงทะเบียนได้ไม่เกิน 9 หน่วยกิต
- 15.2 กรณีที่นักศึกษาลงทะเบียนเรียนในจำนวนหน่วยกิตที่น้อยกว่าเกณฑ์ขั้นต่ำ หรือมากกว่าเกณฑ์ขั้นสูงที่กำหนดไว้ จะต้องได้รับการอนุมัติจากอาจารย์ที่ปรึกษา ทั้งนี้ต้องไม่เกิน 3 หน่วยกิต และจำนวนหน่วยกิตรวมขั้นสูงต้องไม่เกิน 22 หน่วยกิต ต่อภาคการศึกษา กรณีที่มีเหตุจำเป็นที่ต้องลงทะเบียนเรียนต่ำ หรือมากกว่าในวรรคแรก ต้องได้รับความเห็นชอบจากหัวหน้าภาควิชา และได้รับการอนุมัติจากคณะกรรมการประจำคณะ
- 15.3 การนับจำนวนหน่วยกิตในข้อ 15.1 นี้ไม่นับหน่วยกิตของวิชาฝึกงาน หรือวิชาที่ได้รับผลการศึกษาไป

Test cases: 1st class and 2nd class honors

“ข้อ 32 นักศึกษาผู้ที่ได้รับปริญญาเกียรตินิยมต้องเรียนครบจำนวนหน่วยกิต ตามหลักสูตร และต้องอยู่ในเกณฑ์ดังต่อไปนี้

- 32.1 นักศึกษาที่สำเร็จการศึกษา และผลการศึกษามีแต้มระดับคะแนนเฉลี่ยสะสมไม่ต่ำกว่า 3.60 จะได้เกียรตินิยมอันดับ 1
- 32.2 นักศึกษาที่สำเร็จการศึกษา และผลการศึกษามีแต้มระดับคะแนนเฉลี่ยสะสมไม่ต่ำกว่า 3.25 จะได้เกียรตินิยมอันดับ 2
- 32.3 มีระยะเวลาในการศึกษาไม่เกินระยะเวลาที่กำหนดไว้ในหลักสูตรทั้งนี้ไม่นับระยะเวลาที่ลาพักการศึกษา ตามข้อ 51.1.1 – 51.1.3 แห่งระเบียบนี้
การศึกษาในภาคการศึกษาพิเศษทุกภาคการศึกษา จนถึงภาคการศึกษาพิเศษหลังภาคการศึกษาปกติ ภาคการศึกษาสุดท้าย ไม่เป็นการเรียนเกินระยะเวลาที่กำหนด
- 32.4 ไม่เคยได้รับผลการศึกษาตก (F, Fa, Fe) หรือได้รับผลการศึกษามิ่งพอใจ (U) ในรายวิชาใด
- 32.5 ไม่เคยถูกพิจารณาโทษจากการทุจริตในการสอบ หรือโทษทางวินัยใดๆ
- 32.6 ไม่เป็นผู้ที่ขอเทียบโอนรายวิชามากกว่าหนึ่งในสี่ของจำนวนหน่วยกิตตามหลักสูตร ยกเว้นการย้ายสาขาวิชา ตามข้อ 33 แห่งระเบียบนี้”

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 that 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)