

# Basic Software Testing

↳ ဘုရားအနေဖြင့် test

↳ မှတ်ပုံစံအနေဖြင့် test

dev → how can I make it

tester → how can I break it

↳ ဆိုလောက်ခွင့် ပို့ဆောင်ရေးလုပ်လောက်

↳ What is software testing?

↳ the process of verifying and validating that a software application or program

↳ ① meets business requirement and technical requirements that guided its design and development

② work as expected

↳ detect defect

↳ software process

requirement

①

design

②

implement

write code

maintenance and support  
deployment

④

testing

③

ရှိခိုးကိုမျှတိုက်စားစားမှုများ

software testing ဤပုံစံအတွက်

requirement

①

test စားစား

plan

② ပေါ်စေ test case

reporting

④

execution

③

မြန်မာစားစား

မြန်မာစားစား

↳ What, Why, and who

↳ cost of fixing a defect

↳ exposed requirements, design, code, test, operation

↳ Why does software have bugs

↳ no communication

↳ software complexity

↳ requirements change → စာမျက်နှာများ

↳ time pressure

↳ ego စာမျက်နှာ

↳ ယူလွှာ

↳ programming error

↳ library wrong

↳ automation script ပေါ်ပို့ → version

↳ tester ရရှိ

- ↳ Source of problems
- ↳ requirement Үзүүлэгч, Үзүүлэгч, Үзүүлэгч
- ↳ design Өнөөгүйгүйн (msflow)
- ↳ implementation → Үзүүлэгчийн code
- ↳ inadequate testing of software → incomplete testing, poor verification, mistake in debugging
- ↳ evolution
- ↳ adverse effects of faulty software

## Adverse Effects of Faulty Software

- **Communications:** Loss or corruption of communication media, non delivery of data.
- **Space Applications:** Lost lives, launch delays.
- **Defense and Warfare:** Misidentification of friend or foe.
- **Transportation:** Deaths, delays, sudden acceleration, inability to brake.
- **Safety-critical Applications:** Death, injuries.
- **Electric Power:** Death, injuries, power outages, long-term health hazards (radiation).
- **Money Management:** Fraud, violation of privacy, shutdown of stock exchanges and banks, negative interest rates.
- **Control of Elections:** Wrong results (intentional or non-intentional).
- **Control of Jails:** Technology-aided escape attempts and successes, accidental release of inmates, failures in software controlled locks.
- **Law Enforcement:** False arrests and imprisonments.

## Why do Software testing

## Why do Software Testing?

- software testing is focused on finding *defects in the final product.*
  - Gaining confidence about the level of quality
  - Providing information for decision-making
  - Preventing defects
- Software testing answers questions that development testing and code reviews can't.
  - Does it really work as expected?
  - Does it meet the users' requirements?
  - Is it what the users expect?
  - Do the users like it?
  - Is it compatible with our other systems?
  - How does it perform?
  - How does it scale when more users are added?
  - Which areas need more work?
  - Is it ready for release?



Key: Үзүүлэгчийн software Үзүүлэгчийн  
түүхийн замж

## ↳ What do we test

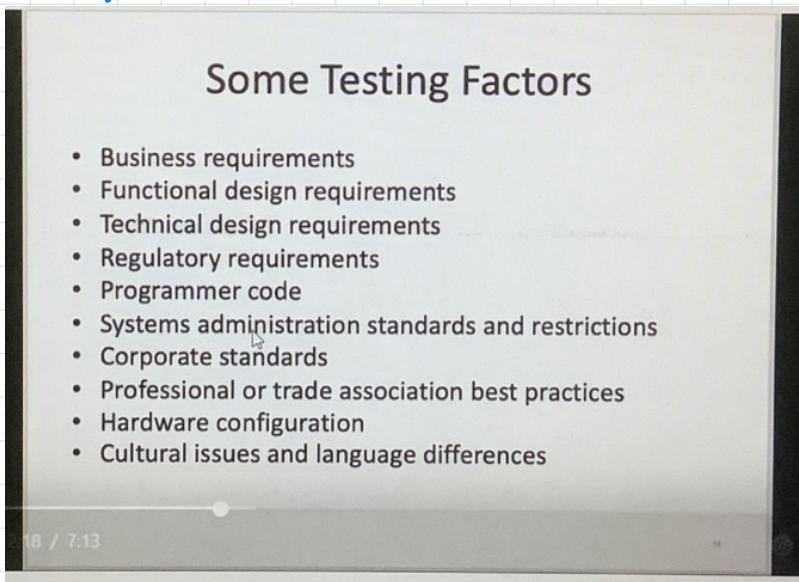
↳ core function → Үзүүлэгчийн орчиний нийтийн

↳ common usage situation → Үзүүлэгчийн орчиний нийтийн → мэдээллийн төслийн төмөрчийн

→ Үзүүлэгч

- ↳ ແບບປົກຕິວໜັດວຽດ
- ↳ ດີນສົ່ງລູກຄົມ, ອາຫວັດທຸກຄົມ

## ↳ Testing Factors



## ↳ Who does Testing

- ↳ programmer → ຂໍເນື້ອໃຈ້ນ unit testing → that function work 100%
- ↳ tester → ອົບລົມທຳກຳຈິດລົງ

## ↳ The Role of Testing

- ↳ ມີຄະດີ 100% → ດັດການເຊື່ອ, ທີ່ໄດ້ຮັບ ພົມກົດສຸດ
- ↳ ຖັນດາການສ່ວຍງົບສັນ

↳ ສົດຍາດນາງ software

↳ ອົບລົມ standard

## ↳ Who involve in testing

↳ requirement analysts

↳ dev → unit function

↳ tester → ອົບລົງ

↳ trainer → ກວດສອບຕົວຢ່າງເຕັມ

↳ user → ເພີ້ມຕະຫຼາດໃຈ້ນທີ່ໄດ້ product

↳ project manager → ພົມກົດ, riske ເລື່ອງເກີດ

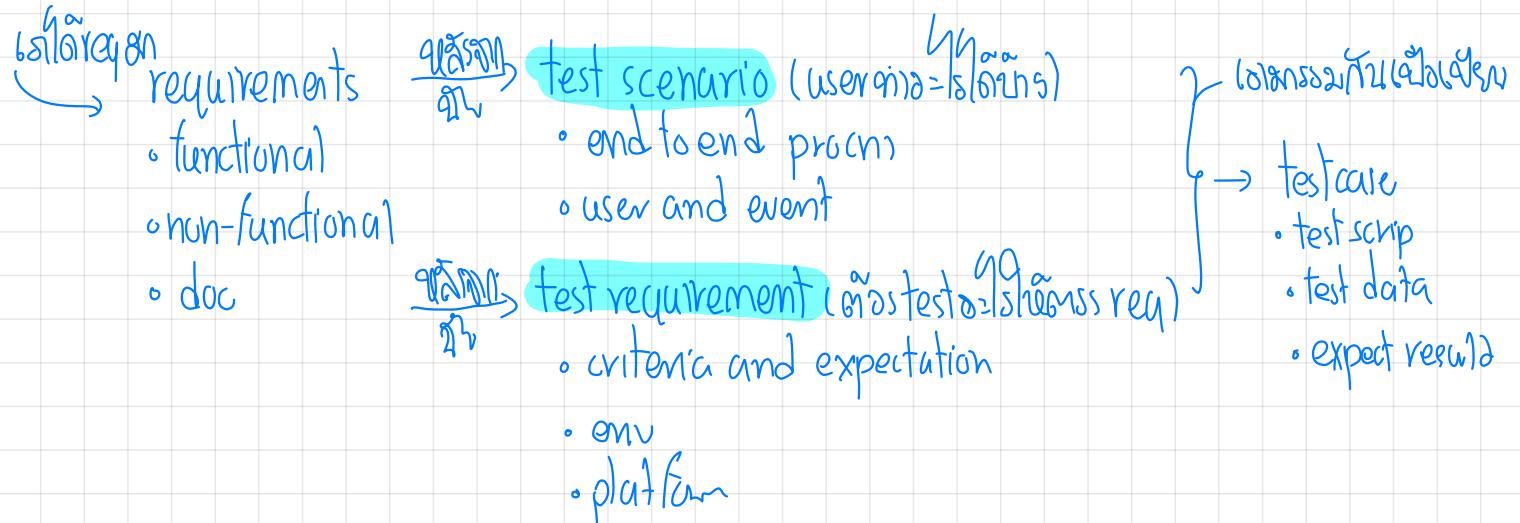
## Fundamental

- ↳ សេចក្តីថាគារពិនិត្យនេះបង្កើតឡើងនៅក្នុងការរាយការណ៍នៃកម្មវិធីនៃការអនុវត្តកូដ
- ↳ ភាពខ្សោយនៃ test នេះនៅលើ → *white box* (readable code)
- ↳ type of testing → unit testing → ត្រូវឲ្យ source code អ្នកបង្កើតតាំង execute ឲ្យខ្សោយ → dev  
 integration testing → ធនធាននៃ module → white or black box → tester, dev  
 (ឬសង្គម) functional and system testing → tester, black box → input & output  
 acceptance testing → user, customer → ធនធាននៃការងារ
- ↳ regression testing → dev, test → ជាដីវិញ្ញាណនៃកូដ → test → version  
 beta testing → ផ្ទើរតាមការងារនៃកូដទៅឱ្យអ្នកប្រើប្រាស់  
 smoke test → ត្រូវឲ្យឱ្យ server និងការងារត្រូវឲ្យនាំងឱ្យនូវការងារដែលត្រូវឲ្យនូវការងារ

Summary				
Testing Type	Specification	General Scope	Approach	Who generally does it?
Unit	Low-Level Design	Actual Code Structure Small unit of code no larger than a class	White Box	Programmer who wrote code
Integration	Low-Level Design High-Level Design	Multiple classes	White Box Black Box	Programmers who wrote code
Functional	High Level Design	Whole product	Black Box	Independent tester
System	Requirements Analysis	Whole product In representative Environments	Black Box	Independent tester
Acceptance	Requirements Analysis	Whole product In customer's Environment	Black Box	Customer
Beta	Ad hoc	Whole product In customer's Environment	Black box	Customer
Regression	Changed Documentation High-Level Design	Any of the above	Black Box White Box	Programmer(s) or independent testers

## ↳ Test requirement

### ↳ សម្រេចតួនាទីថាមព័ត៌មាន requirement



↳ សម្រេចតួនាទីថាមព័ត៌មាន function និង function

### ↳ ធ្វើពីរុច្សាស្ត្រ

#### ↳ business function

↳ find task in functions → សម្រេចតួនាទីថាមព័ត៌មាន

↳ find the transactions for identified task → ត្រូវពិនិត្យពី database ណា

↳ find the datatype of each transaction → កំណត់ប្រភពឈាន

↳ identify field validation → ត្រូវពិនិត្យ

↳ ex. ឯកសារធនធាន ចាន់ខ្លួន ០០ ឱ្យឈើៗ០០០

ចំណាំខ្លួន ០០ ឱ្យឈើៗ០០០

ឯកសារធនធាន

ឯកសារធនធាន

ឯកសារធនធាន

ឯកសារធនធាន

ឯកសារធនធាន save / title ឬជាបាន Dr., Mr., Mrs. ✕  
" " ឯកសារធនធាន list ក្នុងការងារ  
ឯកសារធនធាន ឯកសារធនធាន ឯកសារធនធាន

វគ្គិស្ស  
test case

↳ រាយការណ៍លក្ខណៈ និងការត្រួតពិនិត្យ និងការរៀបចំ (ត្រូវ)

Name Card Data Entry

Name *:	<input type="text"/>
Surname *:	<input type="text"/>
Title *:	<input type="text"/>
Address :	<input type="text"/>
Contact Phone *:	<input type="text"/>
Contact Email *:	<input type="text"/>
<input type="button" value="Clear"/> <input type="button" value="Save"/>	

ឯកសារធនធាន ឯកសារធនធាន ឯកសារធនធាន  
ឯកសារធនធាន save ឬជាបាន name ឬជាបាន

||--> save ✓ || --> រាយការណ៍លក្ខណៈ  
||--> save ✕ || --> ឯកសារធនធាន  
||--> save ✓ || --> ឯកសារធនធាន  
save ✕ || --> ឯកសារធនធាន

⋮  
⋮  
⋮

ការរៀបចំនៅក្នុង → រាយការណ៍

ឯកសារធនធាន name ឬជាបាន

## Example : Add Name Card Information

Task	Transaction	Data Type	Field
<ul style="list-style-type: none"> <li>Select Add function</li> <li>Fill in correct Name card information</li> <li>Fill in duplicated information</li> <li>Fill in incorrect name card information</li> <li>Click Save</li> <li>Click Clear</li> </ul>	<ul style="list-style-type: none"> <li>Name card record</li> </ul>	<ul style="list-style-type: none"> <li>Correct Data Type is allowed to insert.</li> <li>Incorrect Data Type is Fail to insert.</li> <li>Telephone must be numeric number.</li> </ul>	<p>Email is in correct format.</p> <p>Name length is at least 50 characters long.</p> <p>Surname is at least 100 characters long.</p> <p>Address is at least 1000 characters long.</p> <p>Name and Surname must be Thai and English Characters.</p>

↳ only regression test requirement

↳ Oracle Workbench aids requirement elicitation based on function-based testing

↳ ពាណិជ្ជកម្មរបស់ខ្លួន reg ត្រូវតាំង→ ដែលក្នុងរឿងពាណិជ្ជកម្ម

↳ 95 man day worths, 95 million worths

↳ զարդարական համակարգեր աշխատանքի ժամանակաշրջանում

↳ ჰელიკონი automation გრანს, გაუტინი manual ჭავჭავაძეების

↳ Nutzen Sie den z-Test einzuführen

ເບີ່ງຂອງເມືອງ ປາກ (ເຊິ້ນວັນສະປາໂລເຈົ້າ)

→ report bug

b) Where does  $T\theta$  come from?

↳ මැණ්ඩුලතාවය, උග්‍ර, dev

## b) Criteria

↳ assessing business requirement

images (n)

ATM : Practice Writing Test Requirements

## Business Requirements:

- "ATM must do withdrawals"
  - "Withdrawals are between 100-20,000"
  - "Withdrawals are in 100 multiples"

## Group Exercise!

1. Limit the scope to these 3 requirements.
  2. What will you validate (test for)?
  3. Are there any implied requirements that may not be written out?

” මෙයින් සැම්බුද්ධීය ප්‍රතිඵලිය නිරූපිත කළ යුතුවයි

" "            " " *giggle*

၁၁ အုပ္ပန်းကျင့်ပြတ်ရ ဂုဏ်ဆောင်ရွက်ခြင်း

೧೨ ಪ್ರಾಚೀನಕೆಡಿ ೩ ರಾಜ್ಯಗಳಿಗೆ ಸಂಪರ್ಕ

## ၁၁ ဂနားပို့ချေခွဲမြို့၏အကြောင်း

занято



## Test Requirements

- ตรวจสอบว่ามีปุ่มให้เลือกห้ามตัวบัตรเดบิตได้เลือก ✓
- ตรวจสอบว่าบัตรเดบิตในรายการห้ามเลือกให้ใช้ห้ามรายการได้ ✓
- ตรวจสอบว่าต้องใส่เลขบัตร 16 หลักที่มี Check Digit ถูกต้องเพื่อจะให้ทำการรายการ ✓
- ตรวจสอบว่าต้องใส่เลขบัตร 16 หลักที่มี Check Digit ไม่ถูกต้องเพื่อจะให้ทำการรายการ ✓
- ตรวจสอบว่าต้องใส่เลขบัตร 16 หลักไม่ให้ทำการรายการ ✓
- ตรวจสอบว่าไม่ได้เลือกบัตรเดบิตมาและแสดงข้อความเตือนและให้ไว้ในหน้าจอ ✓
- ตรวจสอบว่า CVV เป็นตัวเลขที่นี้ ✓ → 2 ตัว, 3 ตัว, 4 ตัว ตามช่อง
- ตรวจสอบว่าบันทึกจากได้รับอนุมัติและแสดงบนหน้าจอว่าได้รับอนุมัติผ่าน Third Party Software ✓
- ตรวจสอบว่าได้รับ Approval Code จาก Third Party Software จึงจะได้ยืนยันการทำการรายการ ✓
- ตรวจสอบว่าได้รับ Approval Code จาก Third Party Software ในช่วงเวลาที่กำหนด ✓
- ต้องแสดงข้อความที่อนุมัติเมื่อติดต่อผู้ดูแล ✓

• ทดสอบใน browser ต้องใช้เบราว์เซอร์... → In test case ที่ต้อง

## Test Scenario

- การทดสอบฟังctional ของ software test, process ตัวต่อตัวเชิงลึก, event กับ user
- โดยจะรวมไป test requirement และจัดทำ成 test case
- เขียนการทดสอบเชิงลึก requirement → test ยังซึ่งจะต้องรับรองได้
- step : ① น้ำทึบข้อทดสอบ → จุดติดตามความต้องการ
  - ② ติดต่อผู้ดูแล → จัดตั้งรายละเอียด
  - ③ ติดต่อ scenario จัดตั้งรายละเอียด
  - ④ จัดตั้งรายละเอียด
  - ⑤ เรียกเชิงที่มีผู้ใช้ stakeholder ร่วมตรวจสอบต่อไป

ex. email scenario 1. แจ้งรหัสผ่าน login ใหม่

scenario 2. แจ้งรหัสผ่านตั้งครั้งใหม่ ใหม่

scenario 3. แจ้งยอดคงเหลือเงินเดือนของเดือนนี้

scenario 4. แจ้งยอดเงินเดือนที่ต้องชำระ

scenario 5. แจ้งยอดคงเหลือเงินเดือนที่ต้องชำระ

Pratic online shopping : Lazada

เข้าสู่เว็บไซต์ lazada.com ของลาซาด้า

กด menu หน้าแรก

ซื้อของออนไลน์

ซื้อ บ้าน อุปกรณ์ (ห้อง)

ห้องน้ำ

จัดการสัมภาระ

จัดการจ่ายเงิน

ตรวจสอบรายการสั่งซื้อ

จัดการข้อมูล

จัดการห้องน้ำ จัดการห้องน้ำ

จัดการ

→ ตรวจสอบว่าบันทึกได้ตามที่ต้อง

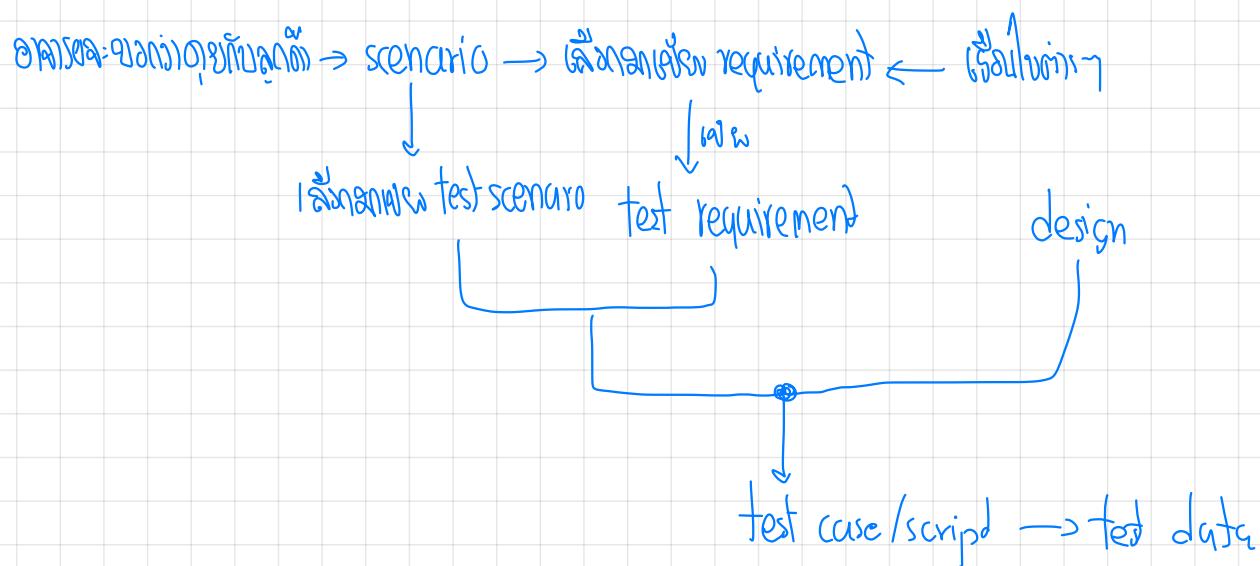
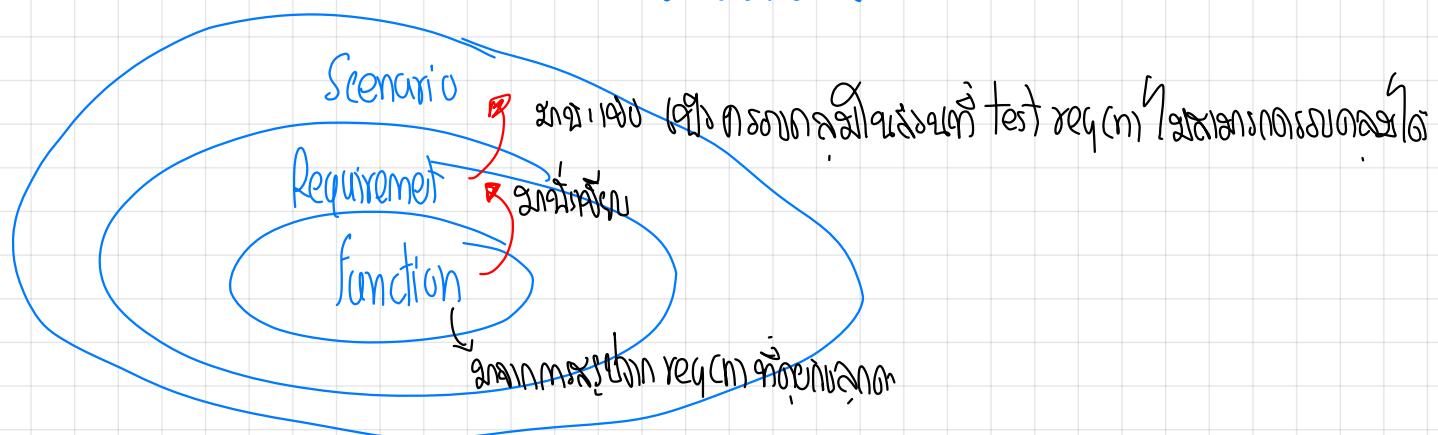
แบบ

scenario 5  
→ ต้องบันทึกเป็น 1 รอบ 1 รอบ

ก้าว

ក្នុង line tu → តម្រូវខ្លួនរបស់ពី → scenario និងការអនុវត្តសម្រាប់  
 :  
 ↓  
 ផ្សេងៗនៃការងាររវាង  
 គឺចង្វារណាទាមរយៈ  
 គឺជាបញ្ជីត្រួតពិនិត្យ  
 ទៅពេលវេលាបាន

សម្រាប់  
 ការងារដែលបានបង្កើត  
 ដែលមាននូវការងារ



## Preparation of test data

→ ទូទៅ → white box → ឱ្យ source code រាយការណ៍នៃការងារ

→ black box → ឱ្យ input × ឱ្យ output

### Compare two types of Testing

Black Box Testing
Testing that UUT satisfies requirements
Focus: I/O behavior
No knowledge of the internal of the UUT
Goal: Cover all the requirements

White-box testing
Testing control structures
Focus: Thoroughness (coverage)
Knowledge of the internal structure of the UUT
Goal: Cover all the code

> UUT = "Unit under test"

- ↳ Black box's aim
  - ↳ function ဆောင်ရည်၊ ဆွဲ-ခွဲများ
  - ↳ interface errors → UI, SI  
↳ service → API
  - ↳ error in data structure
  - ↳ behavior or performance error → အိပ်ခိုင်မှုများ
  - ↳ initialization and termination error → စတင်ခံမှုများ
  - ↳ requirements boundary, equivalence classes, decision tables, state transitional diagram
  - ↳ စုစုပေါင် test case များ
  - ↳ ဒုလ္လာများ  
↳ စုစုပေါင် 18-56
  - ↳ check 18 ? → 79 ? → 17 ? X  
56 ? → 55 ? → 57 ? X

ပြုပေးသော်လည်းကောင်း၊ ပြုပေးသော်လည်းကောင်း၊ ပြုပေးသော်လည်းကောင်း၊ ပြုပေးသော်လည်းကောင်း
- | State | authentication | Compl. Wr.n. |
|-------|----------------|--------------|
| s1    | world for pm   |              |
| s2    | first call     | s3           |
| s3    | sec call       | s4           |
| s4    | th call        | s5           |
| s5    | Access Acv     | s6           |
| s6    | exit acv       |              |
- 
- ပြုပေးသော်လည်းကောင်း၊ ပြုပေးသော်လည်းကောင်း

- ↳ White box
- ↳ ဘဏ်သာမ်း execute & စာမျက်နှာ
- ↳ အောက် input အောက်တော်းဆွဲမှုများ ဖြစ်ပေါ်မှုများ
- ↳ Requirements traceability matrix (မီးတွေ)
- ↳ အဆောက်အအုပ် doc များ → test case test မှုများ
- ↳ reqn မှုများ test မှုများ
- ↳ forward traceability, backward traceability

↳ លទ្ធផលរបាយអង្គភាពនៃការពិនិត្យ និងសម្រាប់ការពិនិត្យ និងសម្រាប់ការពិនិត្យ

Traceability matrix (example)			
Requirement Section	Function Section	Test case	Notes
1.1. Validation of user login credentials.	4.1. User login validation.	6.1.4. User login with proper credentials.	
		6.1.5. User login with invalid username.	
		6.1.6. User login with invalid password.	
		10.1.1. Valid credit card information input.	
1.2. Validation of credit card information.	7.2.4. Credit card information verification.	10.1.2. Invalid credit card number.	
		10.1.3. Invalid credit card name.	
		...	

1:22 / 4:14

Traceability matrix (example)		TC 001	TC 002	TC 003
SR-1.1	User should be able to do this	x		
SR-1.2	User should be able to do that	x		
SR-1.3	On clicking this, following message should appear		x	
SR-1.4			x	
SR-1.5		x	x	
SR-1.6			x	
SR-1.7			x	

ref to test case id

↳ Test Documentation

software test plan

planning test

↳ begin tr, ts, cencan

↑ ↑

software design doc

software reqm specification

↑ តាមរបាយការណ៍

perform tests

↳ run test script

Document Test Results

→ erams test → ចំណាំ

test Logs

↓ រាយការណ៍

Software Test Documentation

## Software Testing Plan

- ព័ត៌មាន: Testing Mobile Business Applications

Approach	Type of Testing	Manual Testing		Automated Testing on Device
		Using Device	Using Emulators	
Standard Testing	Unit Testing	No	Yes	No
	Integration Testing	No	Yes	No
	System Testing	Yes	No	No
	Regression testing	Yes	No	Yes
	Acceptance testing	Yes	No	No
Special type of testing to address specific challenges	Compatibility Testing	Yes	No	Yes
	GUI Testing	Yes	No	No
Type of testing more relevant for enterprise mobile business application	Performance Testing	Yes	No	Yes
	Security Testing	Yes	No	Yes
	Synchronization Testing	Yes	No	No

Ref: • (image) <http://narayan-singh.blogspot.com/2013/03/test-plan-for-mobile-application-testing.html>

29

## ↳ Test Case Description

↳ ມາແລກນັ້ນທີ່ໄດ້ຮັບອະນຸມາດຕະຫຼາດ

↳ ອົງຈະມີລົດຕະບູນ test ID, test description, step , input data , expected output ກີ່ເຄີຍ

ຕິດຕະຫຼາດ

Test Case Description						
<ul style="list-style-type: none"> <li>Verify the login of Gmail</li> </ul>						
Project Name:	Google Email					
Module:	If any					
Created By:	Raj Kumar					
Date of Creation:	DD-MMM-YY					
Date of review:						
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT
TC_LOGIN_001	Verify the login of Gmail	Enter valid User Name and valid Password	1. Need a valid Gmail Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" Button	1. Enter User Name 2. Enter Password 3. Click "Login" Button	Successful login
TC_LOGIN_002	Verify the login of Gmail	Enter valid User Name and invalid Password	1. Need a valid Gmail Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" Button	1. Enter User Name 2. Enter Password 3. Click "Login" Button	A message "The email and password you entered don't match" is shown
TC_LOGIN_003	Verify the login of Gmail	Enter invalid User Name and valid Password	1. Need a valid Gmail Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" Button	1. Enter User Name 2. Enter Password 3. Click "Login" Button	A message "The email and password you entered don't match" is shown
TC_LOGIN_004	Verify the login of Gmail	Enter invalid User Name and invalid Password	1. Need a valid Gmail Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" Button	1. Enter User Name 2. Enter Password 3. Click "Login" Button	A message "The email and password you entered don't match" is shown

Ref: \* (content) <http://www.softwaretestingmaterial.com/test-case-template-with-explanation/>

31

## ↳ Detect bugs

↳ ມາດຕະຖານາລົບສາມາດຂຶ້ນ

↳ detect severity → critical, high, medium, low

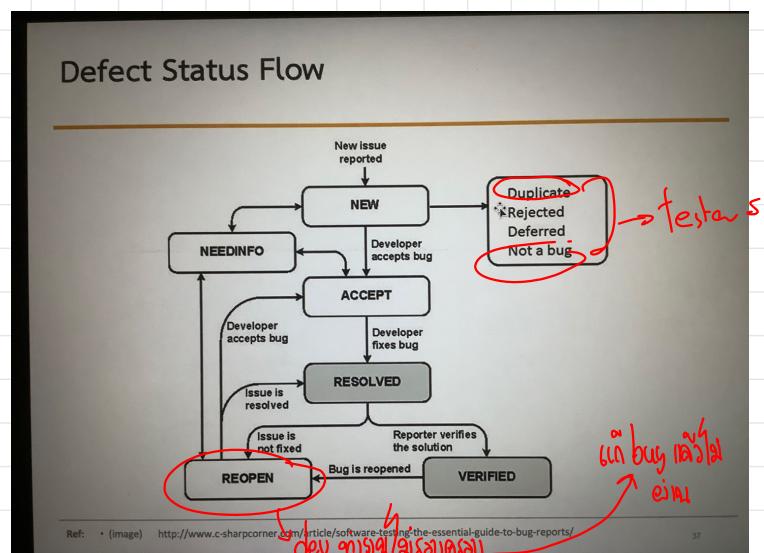
↳ ໃນນັ້ນ

↳ ຢື່ຢັ້ງເຊື່ອຕະຫຼາດໃຫຍ່ຈົບດັບ, ດີເລີ້ມຕົ້ນ

↳ ຄວາມຄວາມລົບລຳກໍານົດສົດ

↳ ອົງຈະມີລົດຕະບູນ

## ↳ Detect status flow



Ref: \* (Image) <http://www.c-sharpcorner.com/article/software-testing-the-essential-guide-to-bug-reports/>

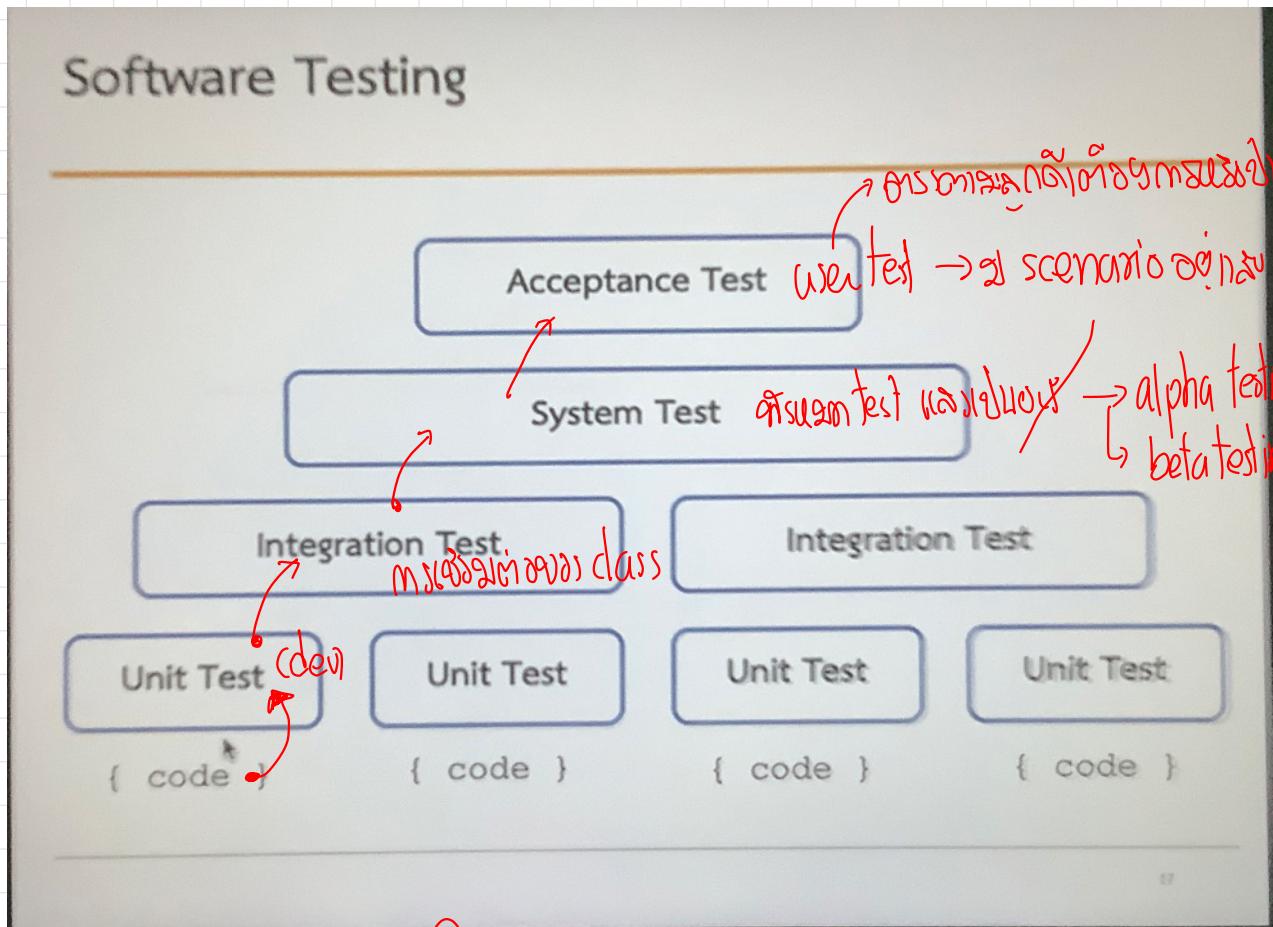
37

in bug  
is not  
fix

dev on site/  
is not done

ໂຄງການທີ່ໄດ້ນຳໃຊ້ດ້ວຍ

# Software Testing



Performing test in pdf " "

