

รายงานผลการเข้ารับการฝึกอบรม

โครงการพัฒนานักประกันคุณภาพซอฟต์แวร์สู่ความเป็นเลิศ
(Quality Assurance Excellence's Project)

จัดโดย

เขตอุตสาหกรรมซอฟต์แวร์ประเทศไทย
(Software Park Thailand)
ถนนแจ้งวัฒนะ อำเภอปากเกร็ด จังหวัดนนทบุรี

วันที่ 9, 10, 23, 24 มิถุนายน พ.ศ.2557

ผู้จัดทำ

รองศาสตราจารย์ทัศนีย์วรรณ ศรีประดิษฐ์
สาขาวิชาวิทยาศาสตร์และเทคโนโลยี

ได้รับทุนสนับสนุนจากกองทุนพัฒนาบุคลากร
ประจำปีงบประมาณ 2557

รายงานการไปฝึกอบรม ฐาน ประชุม / สัมมนา
ตามระเบียบมหาวิทยาลัยสุโขทัยธรรมาราช ว่าด้วยการให้ทุนฝึกอบรม ฐาน
และประชุมทางวิชาการแก่ข้าราชการมหาวิทยาลัยสุโขทัยธรรมาราช

1. ผู้เข้ารับการฝึกอบรม

ชื่อ นางสาวทัศนีย์วรรณ ศรีประดิษฐ์ อายุ 50 ปี
ตำแหน่ง รองศาสตราจารย์ ระดับ 9
สังกัด สาขาวิชาวิทยาศาสตร์และเทคโนโลยี โทร. 8290
ไปฝึกอบรม โครงการพัฒนานักประกันคุณภาพซอฟต์แวร์สู่ความเป็นเลิศ (Quality Assurance Excellence's Project)

ณ เขตอุตสาหกรรมซอฟต์แวร์ประเทศไทย ถนนแจ้งวัฒนะ อำเภอปากเกร็ด จังหวัดนนทบุรี ประเทศไทย
ตั้งแต่วันที่ 9, 10, 23, 24 มิถุนายน 2557 รวมระยะเวลา 4 วัน

2. รายละเอียดเกี่ยวกับการฝึกอบรม

2.1 หัวข้อเรื่อง โครงการพัฒนานักประกันคุณภาพซอฟต์แวร์สู่ความเป็นเลิศ (Quality Assurance Excellence's Project)

2.2 ผู้เข้าร่วมฝึกอบรม จำนวน 23 คน ส่วนใหญ่เป็นนักทดสอบซอฟต์แวร์และนักประกันคุณภาพซอฟต์แวร์ของบริษัทเอกชน

2.3 วิธีการฝึกอบรม

บรรยายเป็นภาษาไทย โดยอิงตามหนังสือที่ใช้ประกอบการฝึกอบรม และมีกิจกรรมกลุ่มปฏิบัติการ (workshop) นอกจากนี้วิทยากรได้แบ่งปันประสบการณ์ในการเป็นนักทดสอบและนักประกันคุณภาพซอฟต์แวร์

2.4 วัตถุประสงค์ของการฝึกอบรม

- 1) เพื่อเพิ่มจำนวนนักประกันคุณภาพซอฟต์แวร์ (QA: Quality Assurance) ให้เหมาะสมและสอดคล้องกับการขยายตัวของอุตสาหกรรมซอฟต์แวร์
- 2) เพื่อพัฒนานักประกันคุณภาพซอฟต์แวร์ให้มีความรู้ความสามารถในระดับสากล
- 3) เพื่อผลักดันและส่งเสริมบริษัทซอฟต์แวร์ไทยให้มีการจ้างงานนักประกันคุณภาพซอฟต์แวร์ เพื่อเพิ่มคุณภาพในการพัฒนาซอฟต์แวร์

2.5 สรุปเนื้อหาสาระจากการฝึกอบรม วันที่ 9 มิถุนายน 2557



วิทยากร

ผู้ช่วยศาสตราจารย์ ดร.จิรพันธุ์ แดงเดช

คณบดีคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยอัสสัมชัญ

ประธานชมรม Thailand SPIN

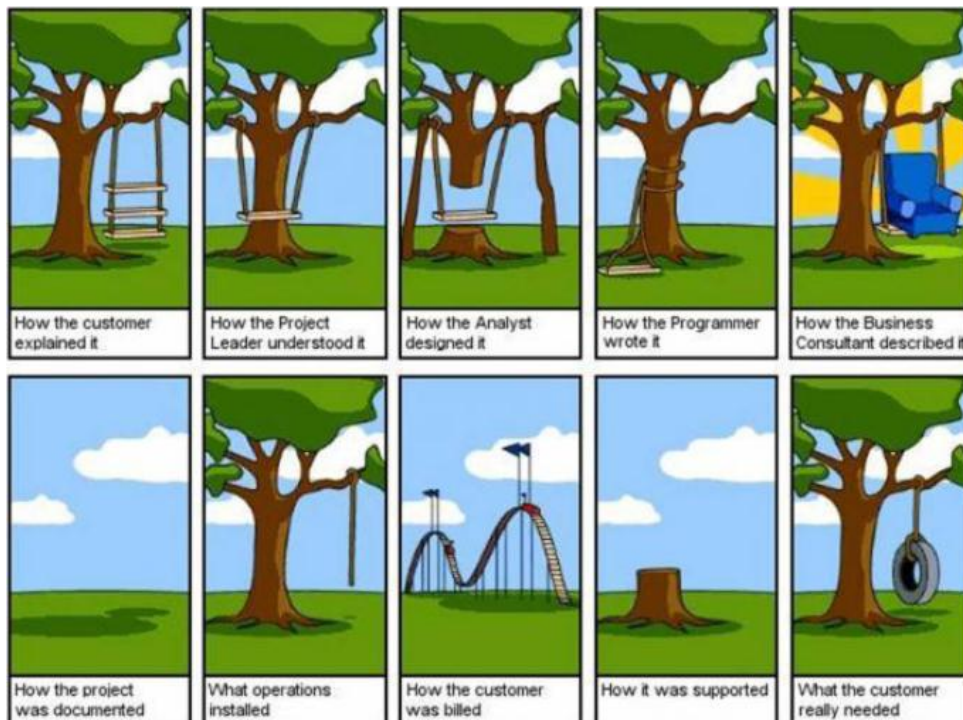
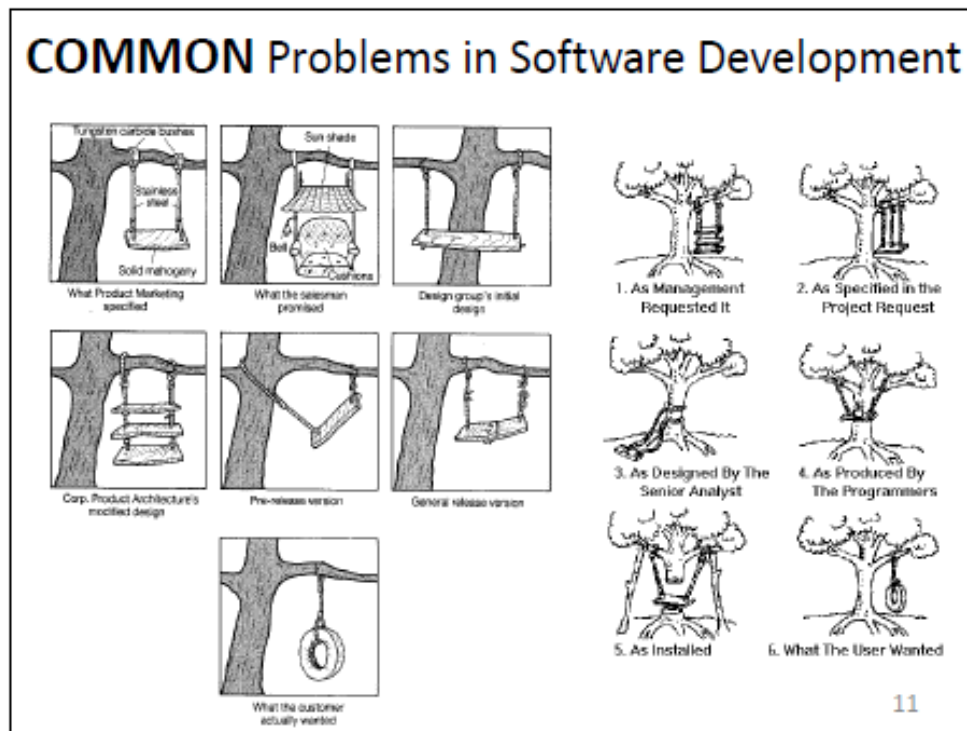
ที่ปรึกษาด้านวิศวกรรมซอฟต์แวร์ของหน่วยงานทั้งภาครัฐและเอกชน



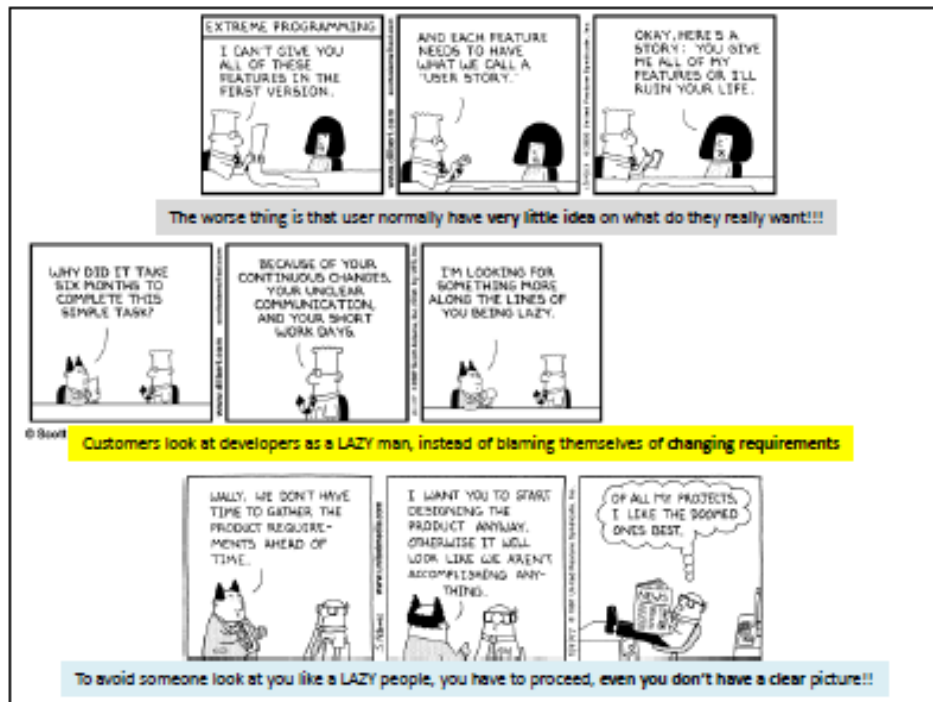
หัวข้อที่บรรยาย

- 1) ความรู้เบื้องต้นเกี่ยวกับการพัฒนาซอฟต์แวร์
- 2) ภาพรวมของการพัฒนาซอฟต์แวร์ในองค์กร
- 3) ปัญหาโดยทั่วไปที่พบในการพัฒนาซอฟต์แวร์ที่ทำให้โครงการส่วนใหญ่ไม่ประสบความสำเร็จคืออะไร
- 4) ปัญหาต่าง ๆ ที่เกิดขึ้น ถ้ามองจากขั้นตอนต่าง ๆ ในการพัฒนาซอฟต์แวร์ ปัญหาเหล่านั้นเกิดขึ้นที่ขั้นตอนไหนได้บ้าง และอย่างไร

ปัญหาโดยทั่วไปในการพัฒนาซอฟต์แวร์ที่ทำให้โครงการส่วนใหญ่ไม่ประสบความสำเร็จ เกิดจากความเข้าใจในเรื่องของความต้องการ (requirement) ที่ไม่ตรงกันระหว่างผู้พัฒนาระบบ (developer) กับผู้ใช้ระบบ (user)



ปัญหาที่เกิดขึ้นในระหว่างขั้นตอนการเก็บรวบรวมความต้องการ (problem in requirement gathering)



Workshop 1 การวิเคราะห์ความต้องการของผู้ใช้ (user requirement)

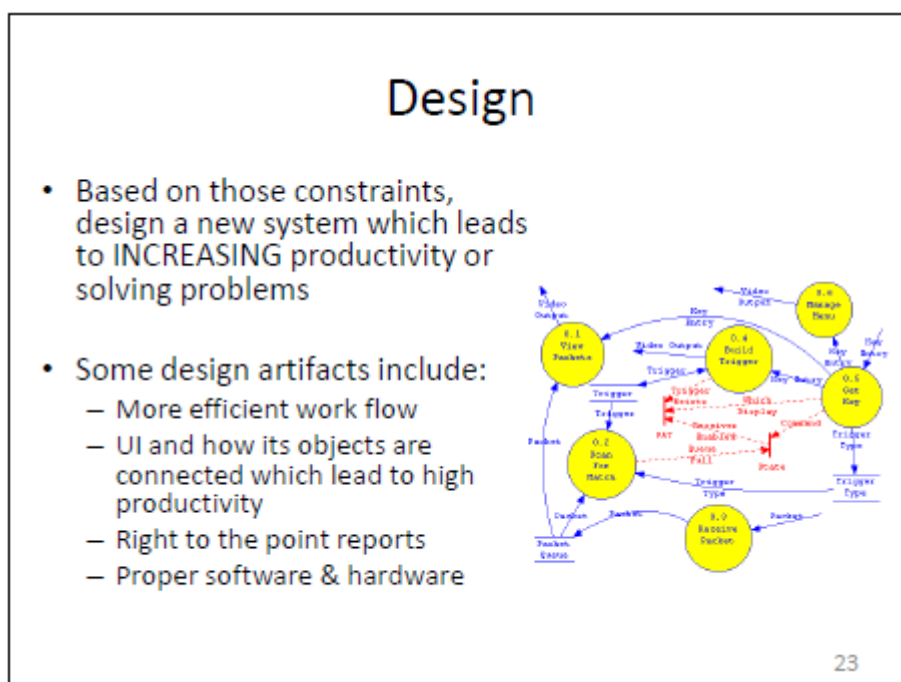
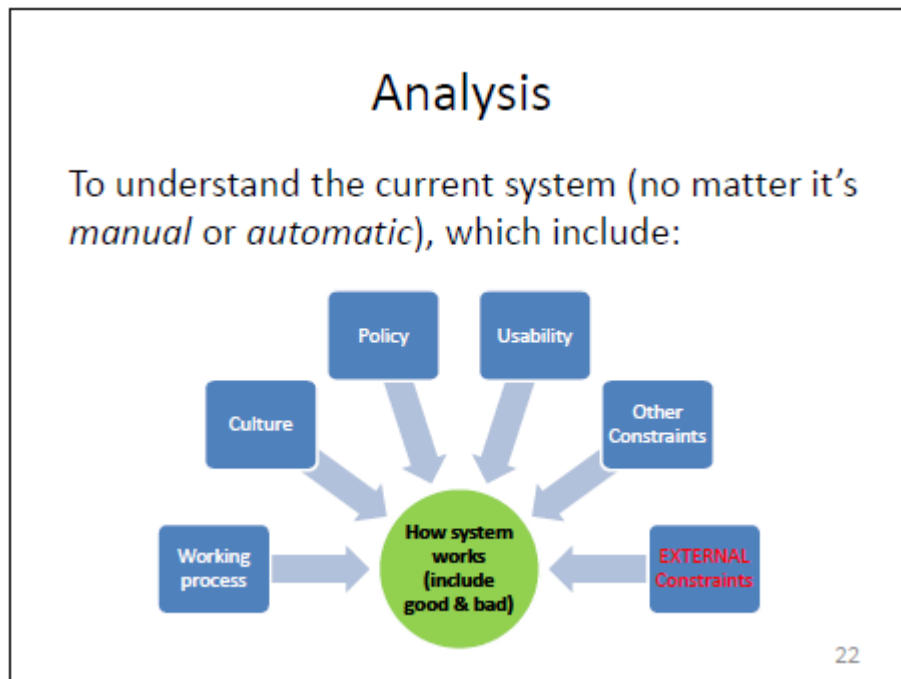
ให้ผู้เข้าอบรมจับคู่กัน และเลือกว่า ใครจะเป็น developer ใครจะเป็น user

จากนั้นให้ผู้เข้าอบรมที่เลือกเป็น user ใช้กระดาษ 1 แผ่น เขียนรูปทรงเรขาคณิตอะไรก็ได้ จำนวน 5 รูป เพื่อแสดงถึงความต้องการของตนเอง และให้ใช้กระดาษอีก 1 แผ่น เขียนบรรยายถึงรูปเรขาคณิตที่วาดไว้ตามความต้องการของตนเองแต่ละรูป และมอบคำบรรยายความต้องการนี้ให้แก่ผู้เข้าอบรมที่เป็น developer

เมื่อผู้เข้าอบรมที่เป็น developer อ่านคำบรรยายความต้องการของ user แล้ว ให้เขียนออกมาเป็นภาพเรขาคณิต โดยใช้เวลา 5 นาที เมื่อเสร็จแล้วให้นำภาพเรขาคณิตที่ user วาดไว้ มาเปรียบเทียบกับภาพเรขาคณิตที่ developer วาด ว่ามีภาพใดตรงกันบ้าง หรือใกล้เคียงกันบ้าง หรือไม่ตรงกันเลย

จาก Workshop นี้ ทำให้ทราบว่า การเก็บ requirement ต้องมีการสอบถามจาก user ให้แน่ชัด จะอ่านจากเอกสารเพียงอย่างเดียวไม่ได้

ปัญหาที่เกิดขึ้นในขั้นตอนของการวิเคราะห์และออกแบบ (problem in analysis and design)



Workshop 2 การวิเคราะห์และออกแบบซอฟต์แวร์

ให้ผู้เข้าอบรมแต่ละกลุ่มเขียนคำสั่ง เริ่มต้นจากเดินไปที่รถจอดอยู่และสิ้นสุดที่ขับรถเคลื่อนออกไปจากที่จอด แต่ละกลุ่มเขียนคำสั่งได้ละเอียดแตกต่างกันไป วิทยากรมาบอกตอนหลังว่า ถ้าคนที่เดินไปที่รถ ขับรถไม่เป็นเลย จะต้องเขียนคำสั่งอย่างไรให้สามารถขับรถออกไปได้

จาก Workshop นี้ ทำให้ทราบว่า การวิเคราะห์และออกแบบซอฟต์แวร์ จะต้องศึกษาและเข้าใจทักษะของผู้ใช้งานซอฟต์แวร์ด้วย

ปัญหาที่เกิดขึ้นในขั้นตอนของการทดสอบซอฟต์แวร์ (problem in software testing)

General Problems in Testing

Complete testing?
What do we mean by "complete testing"?



- Complete "coverage"? Tested every line / branch / basic path?
- Testers not finding new bugs?
- Test plan complete?

Complete testing ~~must~~ mean that, at the end of testing, you know there are no remaining unknown bugs.

- After all, if there are more bugs, you can't find them if you do more testing. So testing couldn't yet be "complete."

Black Box Software Testing Copyright © 2010-11 Dr. Rainer El-Zawie

Exhaustive Testing is Impossible!!

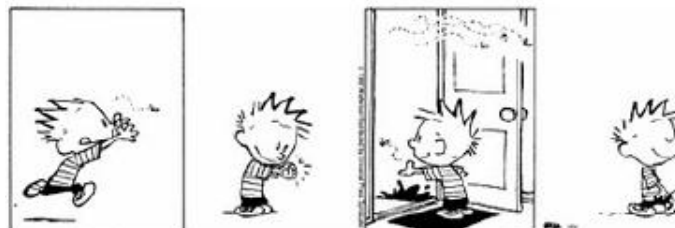
<http://www.qagenius.com/exhaustive-testing-is-impossible/>

Concept of Complete Testing | Exhaustive testing is impossible!!

<http://www.softwaretestingtimes.com/2010/04/concept-of-complete-testing-exhaustive.html>

28

Regression:
"when you fix one bug, you
introduce several newer bugs."



When to Stop Testing?

"Quality Is in the Eye of the Beholder"

Debanjan Mitra and Peter N. Golder
Harvard Business Review, April 2007

"Quality is in the eye of the beholder:
meeting users' requirements for Internet
quality of service"

Anna Bouch, Allan Kuchinsky, Nina Bhatti

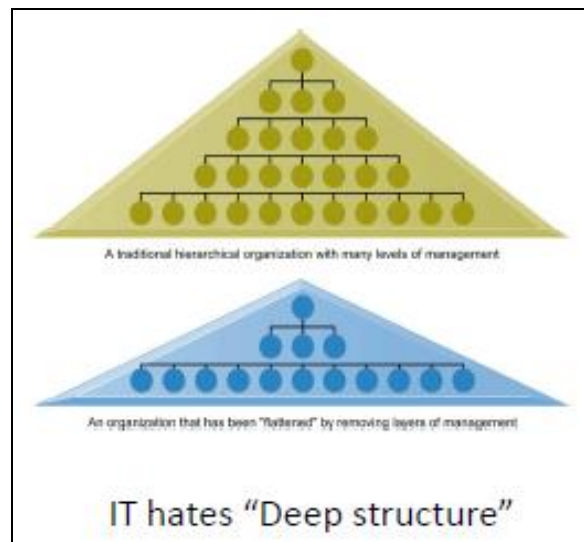
Proceedings of the ACM Conference on Human Factors in Computing
Systems(2000) Volume: 2, Issue: 1, Publisher: ACM, Pages: 297-304

การทดสอบที่นอกเหนือข้อกำหนดของความต้องการ

ปัญหาอื่น ๆ ที่ไม่เอื้ออำนวยต่อการพัฒนาซอฟต์แวร์ เช่น



กฎหมาย กฎเกณฑ์ ระเบียบ ข้อบังคับ



โครงสร้างองค์กรที่มีระดับชั้นมากเกินไป



เทคโนโลยีที่ทันสมัย แต่ใช้งานยากและไม่คงทน

What is **quality**?

- Quality, simplistically, means that a product should **meet its specification**.
- This is problematical for software systems
 - There is a **tension** between **customer** quality requirements (efficiency, reliability, etc.) and **developer** quality requirements (maintainability, reusability, etc.);
 - Software specifications are **usually incomplete** and often **inconsistent**.

อาจเกิดความต้องการที่ขัดกันเอง ระหว่าง ความสวยงาม (graphic) กับ ความเร็วในการแสดงผล

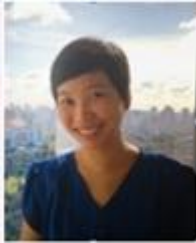
Quality assurance and standards

- **Product standards** define **characteristics** that all components should exhibit e.g. a common programming style.
- **Process standards** define **how** the software process should be enacted.

Software quality attributes

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

วันที่ 10 มิถุนายน 2557

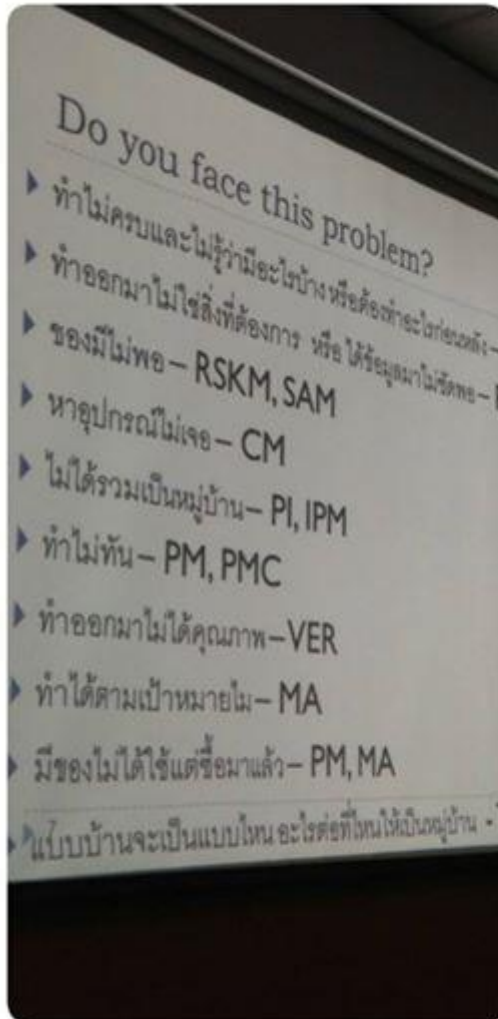


วิทยากร

นางสาวนพรัตน์ ศลิษฐ์อรธรกร

Technical Specialist – Software Process

บริษัท ทอมสัน รอยเตอร์ (Thomson Reuters)



หัวข้อที่บรรยาย

- 1) CMMI Overview
- 2) Examples of Process Areas
- 3) กระบวนการที่ช่วยในการพัฒนาซอฟต์แวร์ให้มีคุณภาพ นอกเหนือจากการทดสอบ
- 4) 5 Whys and Fishbone

Introduction to CMMI



CMMI:

Capability Maturity Model Integration by SEI (Software Engineering Institute)

Capability

the ability to do something

Maturity

the condition of growing

Model

framework or direction to do something repeatedly

Integration

combining things together

"A framework that aid an organization in improving processes and its abilities to develop, acquire, and maintain its products and services"



Carnegie Mellon
Software Engineering Institute

CMMI Representatives

Stage representation

(Maturity Levels)

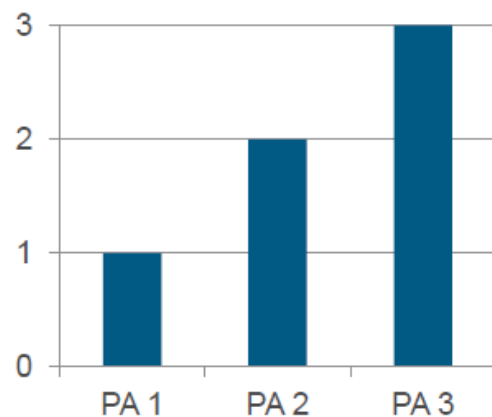
Characteristics of the Maturity levels



Continuous representation

(Capability Levels)

Continuous (Target Profile)



Workshop 3 สร้างความเข้าใจในกระบวนการ CMMI

Instruction

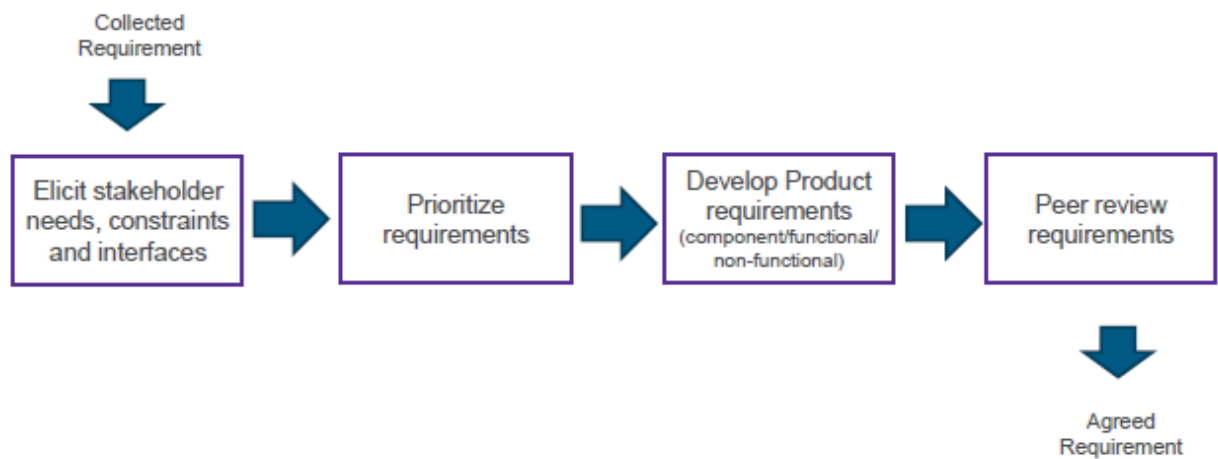
- Village – Theme colorful
- At least 3 houses completed in phase
- Customer will inform more requirements
- Sales price per quality house: \$100
- Late deliver charge per house \$30
- Incomplete village cannot be sold.
- Materials that you can buy
 - Magazine paper
 - A4 paper
 - Scissor
 - Cutter
 - Pen color
 - Ruler
 - Post its
 - Pencil



ผลงานที่ได้ไม่ตรงตามความต้องการเพียงแค่ 50% เท่านั้น

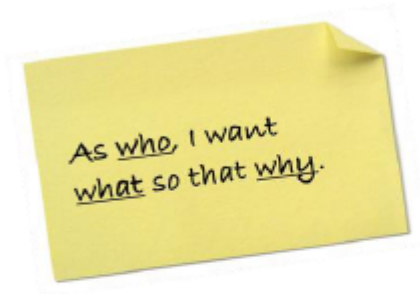


Requirement Development Process



การเขียนผลการวิเคราะห์ความต้องการ โดยใช้ User Story

User Story



As a Line user,
I want to create group chat
so that I can chat with many friends at the same time.



INVEST

Independent	Negotiable	Valuable	Estimatable	Small	Testable
เป็นอิสระจากกัน	ปรับเปลี่ยนได้	มีมูลค่าเพิ่ม	ประมาณการได้	เล็ก	ทดสอบได้

Acceptance Criteria

- After click creating group, the group name will be active within 1 seconds.
- The created group chat will be shown on Friends page under group category.
- The number of participants in group will be shown behind the group name.
- The maximum number of participants in each group is 50 persons.
- If the name of group hasn't defined, the group name will be shown as "no name".



SMART

Specific Measurable Achievable Relevant Time-bound

5 Whys

- 5 Why's can be used to investigate a specific failure to find a problem's real root cause.



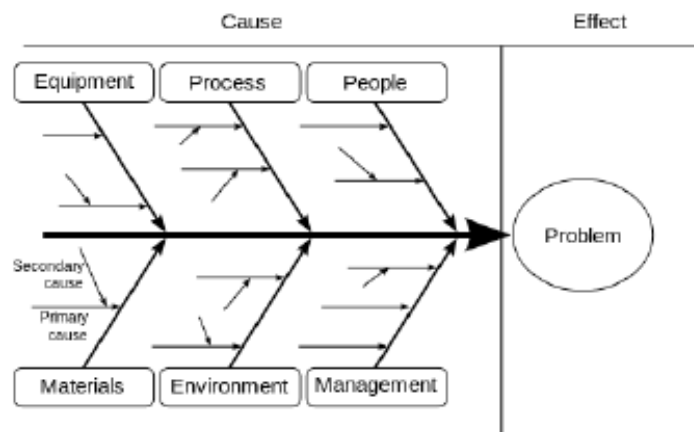
Tips:

- Keep asking until get the root cause
- If that didn't happen, would the problem go away?

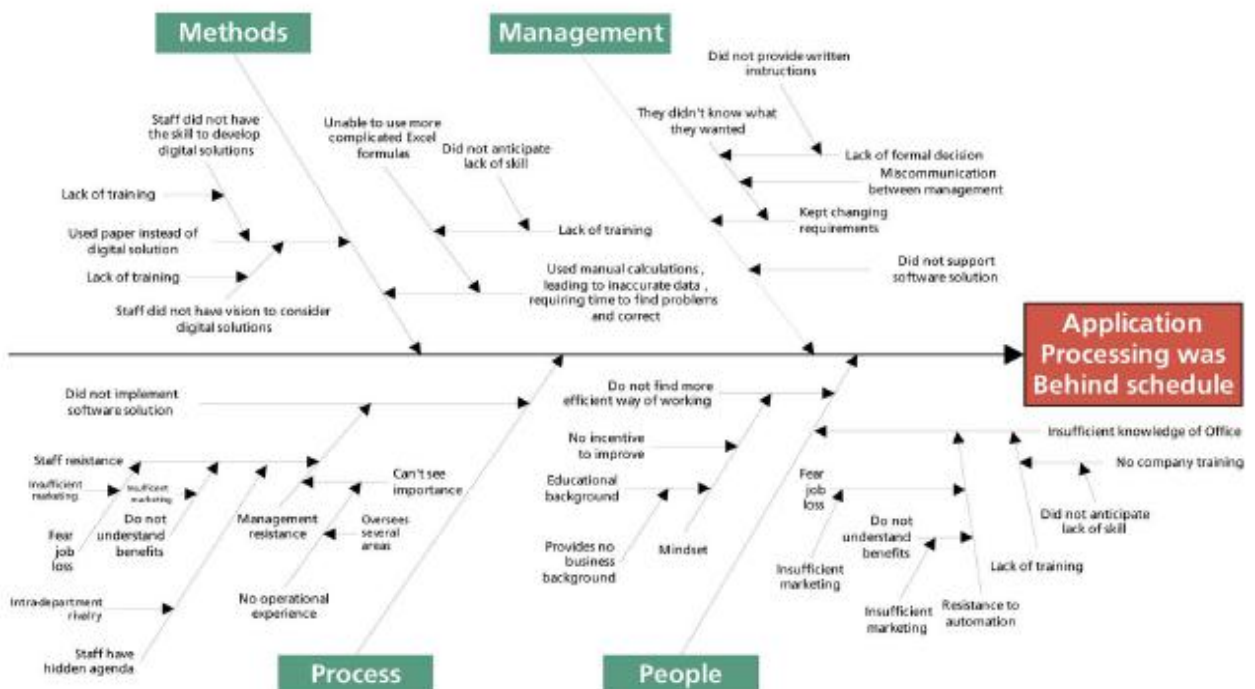
<http://www.educational-business-articles.com/5-whys.html>

Fishbone diagram

- "Cause-and-Effect Diagram" or "Ishikawa Diagram"
- Kaoru Ishikawa – 1960s
- A graphical method
- Finding the root causes of an effect (-/+).



Example



ศิริราชก้าวไกล ส่งยาทั่วไทยทางไปรษณีย์



<http://www.thairath.co.th/content/424212>



THOMSON REUTERS

ผศ.นพ.วิเชียร วามวาทินชัย ผู้อำนวยการ รพ.ศิริราช เปิดเผยว่า จากจำนวนผู้ป่วยที่รับการรักษาใน โรงพยาบาลศิริราชมีจำนวนเพิ่มขึ้นอย่างต่อเนื่องในแต่ละวัน ส่งผลถึงความแออัด จนทำให้เกิดการรอคอยในการรับยา ชำระค่าบริการ กว่าจะได้กลับบ้านก็หมดเวลาไปเกือบครึ่งวัน ตนจึงมอบนโยบายให้ รศ.นพ.เชิดชัย นพณัฐจารีสุเลศ รองผู้อำนวยการรพ.ศิริราช ดำเนินการผู้ป่วยนอกและพัฒนาคุณภาพ คิดหาวิธีที่จะช่วยให้ผู้ป่วยกลับบ้านไวขึ้น จากผลการวิเคราะห์ระบบงานเพื่อลดความสูญเสียตามแนวคิด Lean จึงเกิดโครงการ "ศิริราชก้าวไกล ส่งยาทั่วไทย ทางไปรษณีย์" ซึ่งเรามองว่า "ไปรษณีย์ไทย" เป็นหน่วยงานที่มีศักยภาพ ที่จะช่วยให้ยาของเราไปถึงมือผู้ป่วยโดยเร็ว มีคุณภาพ และประสิทธิภาพ ด้วยไปรษณีย์ด่วนพิเศษในประเทศ หรือ EMS ส่งด่วนทั่วไทย นับเป็นทางเลือกใหม่ที่ฝ่ายเภสัชกรรม รพ.ศิริราช ได้ร่วมกับ บริษัท ไปรษณีย์ไทย จำกัด อำนวยความสะดวกให้ผู้ป่วย

โดยที่ ผศ.วิเชิด ตั้งจิตติพร หัวหน้าฝ่ายเภสัชกรรม รพ.ศิริราช ประธานคณะทำงานการจัดส่งยาทางไปรษณีย์ พร้อมทีมงานเป็นกำลังสำคัญในการทำงาน ด้วยขั้นตอนการให้บริการแบบครบวงจร "นั่งรับการตรวจรักษาและได้รับใบสั่งยาจากแพทย์แล้วผู้ป่วยสามารถขอเข้าร่วมรับยาทางไปรษณีย์ตามขั้นตอนดังนี้

1. ติดต่อที่เคาน์เตอร์ให้บริการส่งยาทางไปรษณีย์ (ตรงข้ามเคาน์เตอร์ออกใบนัดผู้ป่วย) หน่วยตรวจอายุรศาสตร์ชั้น 2 ห้อง 210 ดึกผู้ป่วยนอก
 2. ผู้ป่วยกรอกใบแจ้งความจำนงค์เข้าร่วมการจัดส่งยาทางไปรษณีย์พร้อมเขียนชื่อ ที่อยู่ที่ต้องการจัดส่ง
 3. เภสัชกรประเมินการใช้ยาของผู้ป่วย อธิบายยาที่ผู้ป่วยได้รับ และให้ตารางการรับประทานยาแก่ผู้ป่วย
- พร้อมให้เอกสารใบยืนยันการรับยาทางไปรษณีย์ที่มีหมายเลขพัสดุ 13 หลัก
4. ผู้ป่วยชำระเงินค่ายา และค่าจัดส่งทางไปรษณีย์ ราคา 150 บาท
 5. รับยาตามที่อยู่ระบุ โดยจัดส่งยาทางด่วนพิเศษในประเทศ หรือ EMS ส่งด่วนทั่วไทย ภายใน 1-3 วัน หากการ นับจากวันที่ตรวจเสร็จ"
- อย่างไรก็ตาม ผู้ที่จะเข้าร่วมบริการจัดส่งยาทางไปรษณีย์ ต้องได้รับการประเมินจากเภสัชกรในการใช้ยาว่าสามารถใช้ยาได้อย่างถูกต้องและปลอดภัย และเป็นผู้ป่วยที่สามารถควบคุมอาการได้คงที่แล้ว

http://www.si.mahidol.ac.th/th/hotnews_detail.asp?hn_id=1562

42

PDCA

ITIL

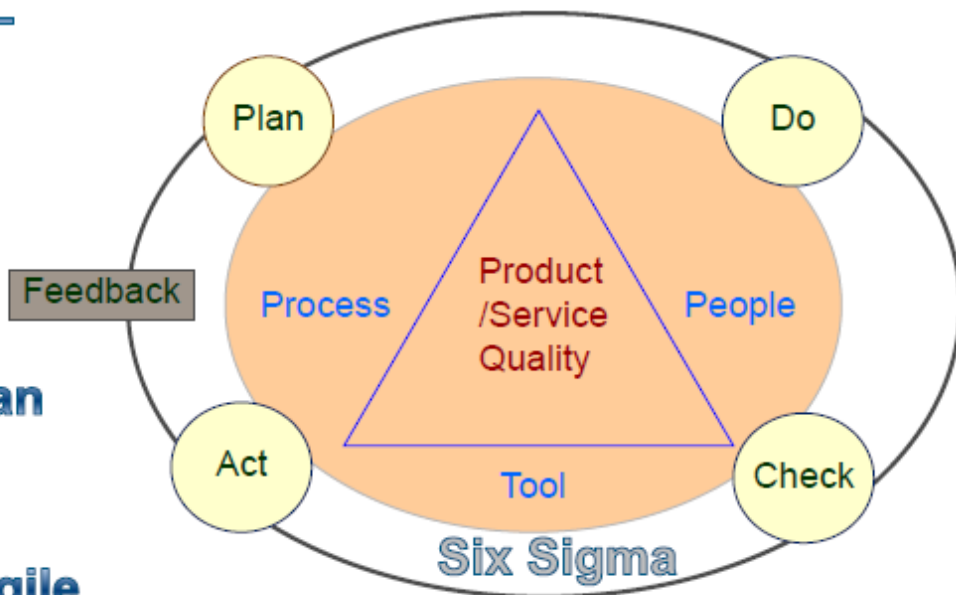
ISO

Lean

CMMI

Agile

Six Sigma



PDCA: Dr. W. Edwards Deming

วันที่ 23 มิถุนายน 2557



วิทยากร

ดร.นล รัตนันท์

ผู้อำนวยการทางคุณภาพ (Quality Director)

บริษัท ทอมสัน รอยเตอร์ (Thomson Reuters)

รองประธานชมรม Thailand SPIN



หัวข้อที่บรรยาย

- 1) Software Testing คืออะไร
- 2) Software Testing แบ่งออกได้เป็นอะไรบ้าง
- 3) เทคนิคการทดสอบซอฟต์แวร์
- 4) เทคนิคการออกแบบ Test Cases แบบ Black Box
- 5) เทคนิคการออกแบบ Test Cases แบบ White Box

True Purpose of software testing

- TO IMPROVE SOFTWARE DEVELOPMENT
- Just find bugs and help developers remove will not create lasting improvement
- By analyzing bugs thoroughly, reporting them "skillfully", working with developers to understand factors that cause them
 - Fewer bugs in future

"The real value of tests is not that they detect bugs in the code, but that they detect inadequacies in the methods, concentration and skill of those who design and produce the code."

Tony Hoare, 1996

Some English first

- **Effectiveness** = To do **right** things
- **Efficiency** = To do things **right**

Therefore,

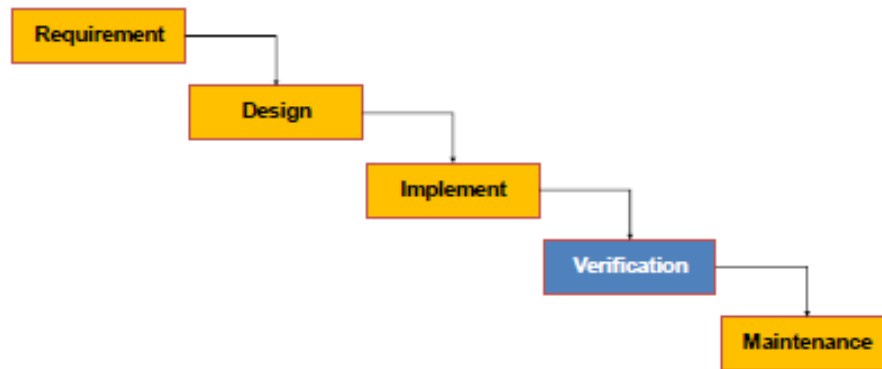
- Find bugs more **effectively** = Find **right** bugs
- Find bugs more **efficiently** = **Right** way to find bugs

Topics

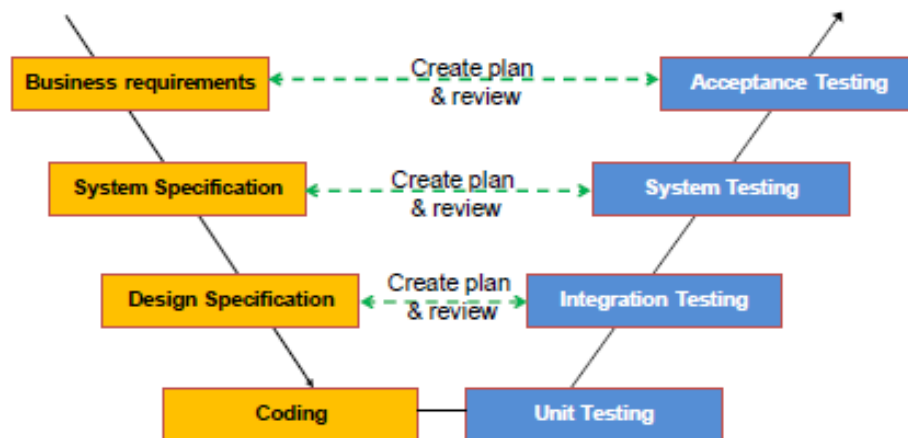
- Level of testing
 - When in SDLC tests happen
 - Type of testing
 - Different test approaches/focuses
 - Test techniques
 - Equivalence Partitioning, Boundary Value testing, Pairwise
 - Manage testing
 - E.g., Test strategy, Test report, Bug report
- Effectively*
- Efficiently*

ขั้นตอนในการทดสอบของแต่ละโมเดล จะอยู่ในช่วงที่แตกต่างกัน ดังนี้

Water Fall Model



V - Model



ระดับของการทดสอบ แบ่งออกเป็นระดับต่าง ๆ ดังนี้

Test level (1)

■ Unit testing

- Testing of individual program units or components
- Usually the responsibility of the component developer (except sometimes for critical systems)
- Tests are based on experience, specifications and code
- A principal goal is to detect functional and structural defects in the unit
- Developer uses a debugger or self-developed tools to find out whether the code meets his expectations and the technical specification.

Test level (2)

■ Integration testing

- Testing of groups of components integrated to create a sub-system
- Responsibility could be given to developers or testers
- Tests are based on a system specification (technical specifications, designs)
- A principal goal is to detect defects that occur on the interfaces of units and their common behavior
- Often use Gray Box testing

Test level (3)

■ System testing

- Testing the system as a whole
- Usually the responsibility of an independent testing team
- Tests are usually based on a requirements document (functional requirements/specifications and quality requirements)
- A principal goal is to evaluate attributes such as usability, reliability and performance (assuming unit and integration testing have been performed)

Test level (4)

■ Acceptance testing

- Testing the system as a whole
- Usually the responsibility of the customer
- Tests are based on a requirements specification or a user manual
- A principal goal is to check if the product meets customer requirements and expectations

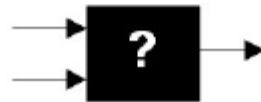
ประเภทของการทดสอบ จำแนกได้หลายวิธี ได้แก่

- “Black Box” / “White Box”
- Functional / Non-Functional
- Requirement-based / Risk-based

They are not mutually exclusive; many can be applied at particular test level

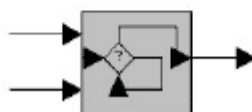
Black Box Testing

- Synonyms : closed-box and input / output – driven
- View the software as a black box with inputs and outputs
- Knowledge of inside the box is not required
- Concentrate on **what** the software does, **not how** it does



White Box Testing

- Synonyms : code-based or structural testing
- Knowledge of how the software is design & implemented, how it works, is required
- Requires programming skills to identify all paths through the software



Functional Testing

- Function: “What” a system does
 - Focus on testing functionality that attracted users to the software in the first place
 - Often use Black or Gray box testing
- Described by
 - Requirements specification, use cases
 - Functional specification
 - Undocumented / Common knowledge / Understood

Non-Functional Testing

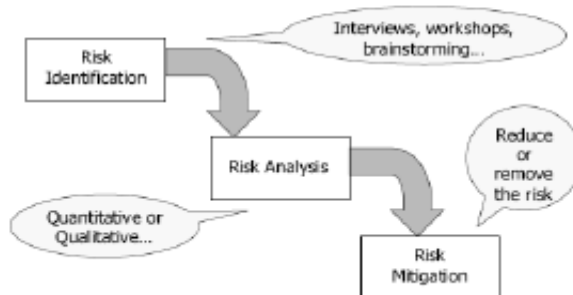
- Testing of software product characteristics
 - Often not captured in product requirement document
 - “How Well” does system operates
 - Does system behave about the same as similar systems?
- Examples
 - Performance
 - Load
 - Stress
 - Security
 - *What else?*
 - **Resiliency** : ถ้ามี Server ตัวหนึ่ง down จะมีตัวอื่น support ได้หรือไม่
 - **Operability** : ถ้ามี Error เกิดขึ้น จะถูกส่งไปแจ้งยังจุดที่เหมาะสมหรือไม่

Requirement-based testing

- Requirements = Set of ideas that collectively define quality of a product
- Requirement-based testing attempts to
 - prove that product satisfy its *stated* requirements
 - Design test cases so they are traceable to one or more stated requirements and vice versa

Risk-based testing

- Risk Management Process



เทคนิคการทดสอบซอฟต์แวร์ แบ่งออกเป็น

1) **เทคนิคการทดสอบฟังก์ชัน** ประกอบด้วย

1.1 **เทคนิคการแบ่งกลุ่มให้เท่ากัน (Equivalence Partitioning: EP)**

- Equivalence Partition rule state that each value in a partition can be representative for the whole partition
 - If one value works, all will work
- We don't need to test all value in partition
 - One from each partition better than all from one
- Two style of equivalence partition
 - One to one : a separate test case is designed for each identified partition
 - Minimized : a minimal set of test cases is created by designing each test case to cover as many new partitions as possible (input must be independent)

ตัวอย่าง ฟังก์ชันโปรแกรมเพื่อแปลงวันที่ (date) ในปี ค.ศ. 2007 ให้เป็นวัน (day)

เช่น Input = 14/02 (dd/mm) Output = Wednesday

วิธีทำ

- First Step
 - Define the input condition
- Second Step
 - Define Valid and Invalid partition for each input
- Third Step
 - Draw Equivalence Partition Table

Step#1: Define the input condition

- | | |
|---|---|
| <ul style="list-style-type: none">• Input 1 = Month• Input 2 = Date<ul style="list-style-type: none">– 1-31 (Month 1,3,5,7,8,10,12)– 1-30 (Month 4,6,9,11)– 1-28 (Month 2) | <p>These are all input conditions:</p> <ul style="list-style-type: none">• Input 1 = Month• Input 2 = Date of Month 1,3,5,7,8,10,12• Input 3 = Date of Month 4,6,9,11• Input 4 = Date of Month 2 |
|---|---|

Step#2 & 3: Define Valid and Invalid + Draw table

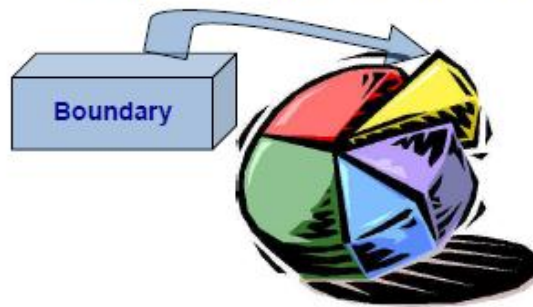
Condition	Valid Partition	Tag	Invalid Partition	Tag
Month	1-12	V1	<1 >12	X1 X2
Day (Month = 1,3,5,7,8,10,12)	1-31	V2	<1 >31	X3 X4
Day (Month = 4,6,9,11)	1-30	V3	<1 >30	X5 X6
Day (Month = 2)	1-28	V4	<1 >28	X7 X8

Identifying the test cases

Test case	Condition & Input	Expected Output	Cover Tag
1	03/02	Saturday	V1,V4
2	35/01	Error – Invalid Date	X4
3	31/02	Error – Invalid Date	X8
4	-1/12	Error – Invalid Date	X3
5	1/15	Error – Invalid Month	X2
6			

1.2 การทดสอบที่ค่าของขอบเขต (Boundary Value testing: BV)

- Why Boundary Value Analysis?
 - Faults tend to live near boundaries
 - Good place to look for faults
 - Test values on and both sides of boundaries



ตัวอย่าง

In a system designed to work out the tax to be paid:
An employee has \$4000 of salary tax free.
The next \$1500 is taxed at 10%
The next \$28000 after that is taxed at 22%
Any further amount is taxed at 40%

วิธีทำ

0 – 4,000	No tax
4,001 – 5,500	10%
5,501 – 33,500	22%
> 33,501	40%

1. To the nearest whole dollar, which of these is a valid Boundary Value Analysis test cases?

- A. \$32001
- B. \$28000
- C. \$1500
- ☒ D. \$33501

2. To nearest whole dollar, which of these groups of numbers fall into three DIFFERENT equivalence classes?

- A. \$28000; \$28001; \$32001
- ☒ B. \$4000; \$5000; \$5500
- C. \$4000; \$4200; \$5600
- D. \$32001; \$34000; \$36500

ตัวอย่าง

A hotel telephone system can perform 3 functions

- Call another hotel room by entering a room number (201 to 500)
- Call an external line by entering a 9, followed by the number
- Call various hotel services
 - 0 = operator
 - 7 = room service
 - 8 = reception

Write set of test cases for this telephone system.
(10 mins)



สรุปกรณีทดสอบทั้งหมด ดังนี้

T/C	Special character (1 click)						Hotel rm			Ext number			Expected results	
	V			I			V	I		V	I			
1	x	7	8	9	*	#	1-6	201-500	1-200	501-999	<Thai province #>	<International #>	Invalid phone #	V: Call operator
2		x												
3			x											
4					x			x						I:
5						x		x						
6							x	x						
7					x						x			
8					x							x		
9						x						x		
10														
11				x				x						
12														
13														
<more>														

2) เทคนิคการทดสอบด้วยกรณีทดสอบที่เหมาะสม (test case optimization technique)

- A method used to derive or select test cases.
- Test case - inputs to test the system and the expected outputs (or predicted results) from these inputs if the system operates correctly under specified execution conditions
 - Inputs may include an initial state of the system
 - Outputs may include a final state of the system
 - When test cases are executed, the system is provided with the specified inputs and the actual outputs are compared with the outputs expected
- Test case = test rule + test data
 - Test rule = a determined operation that leads to a define result
 - Test data = data needs with the rule to execute and evaluate a test

วันที่ 24 มิถุนายน 2557

วิทยากร



นายประธาน ด้านสกุลเจริญกิจ

ScrumMaster and Agile Practitioner

บริษัท สยามชำนานุกิจ จำกัด

ผู้ก่อตั้งเว็บไซต์ WeLoveBug.com

รองประธานชมรม Thailand SPIN

ประสบการณ์ 10 ปี ในฐานะผู้ลงมือปฏิบัติ เป็นที่ปรึกษา เป็นผู้บริหาร
ทั้งส่วนของทีมทดสอบ และทีม IT



หัวข้อที่บรรยาย

- 1) การพัฒนาซอฟต์แวร์ ด้วยแนวคิด Agile
- 2) ผู้ทดสอบซอฟต์แวร์จะต้องมี awareness อะไรบ้าง
- 3) Testing with efficiency : the cost of quality
- 4) Non technical side of QA : soft skill หรือทักษะเกี่ยวกับพฤติกรรมเชิงมนุษย์ที่มีประโยชน์และจำเป็น
- 5) เครื่องมือที่ใช้ทดสอบซอฟต์แวร์
- 6) เปรียบเทียบระหว่าง Automation Testing และ Manual Testing



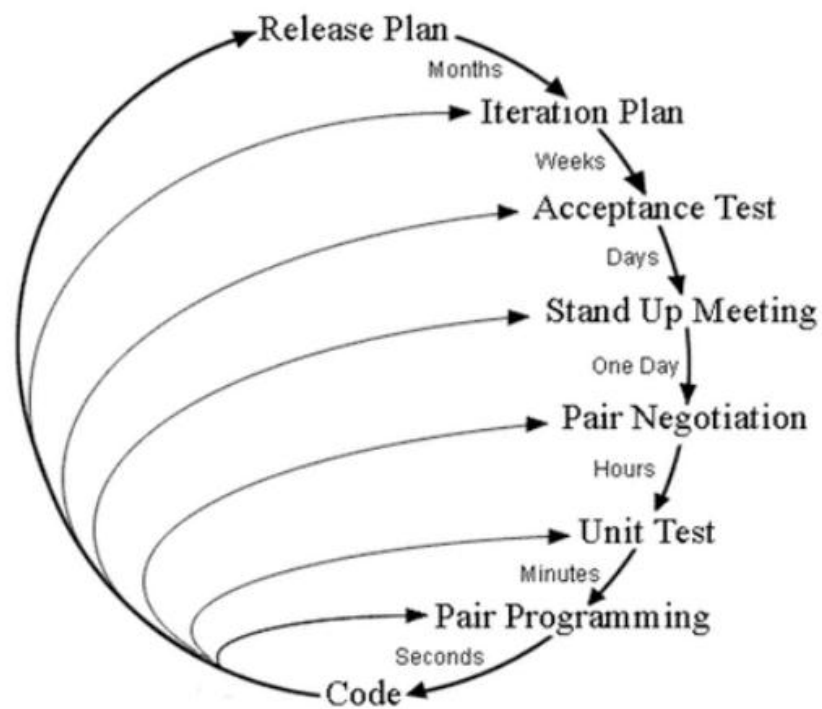
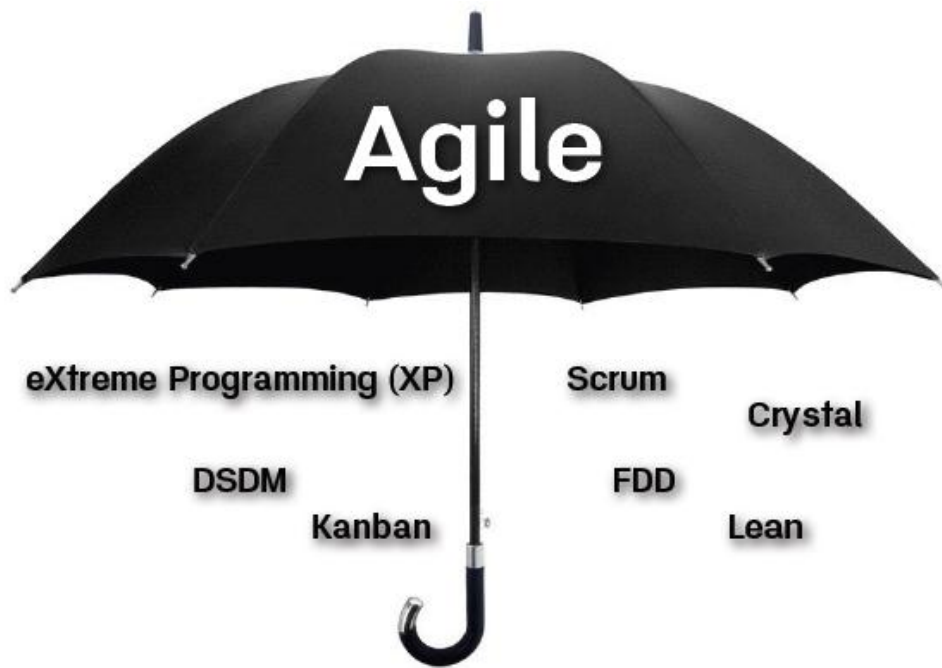
4 core values
+
12 principles

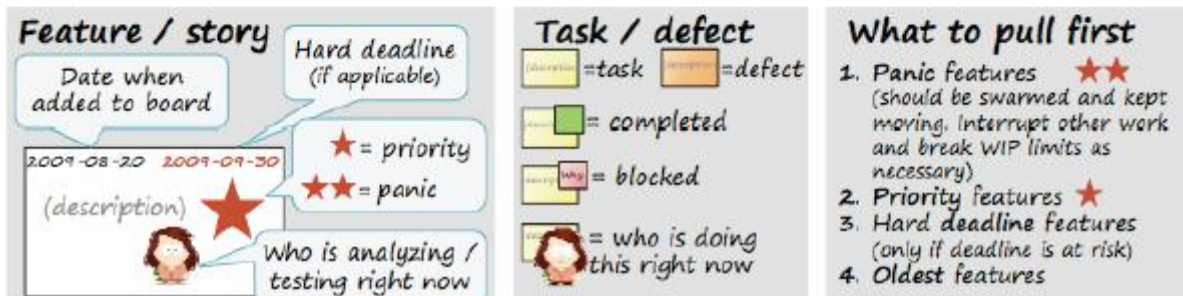
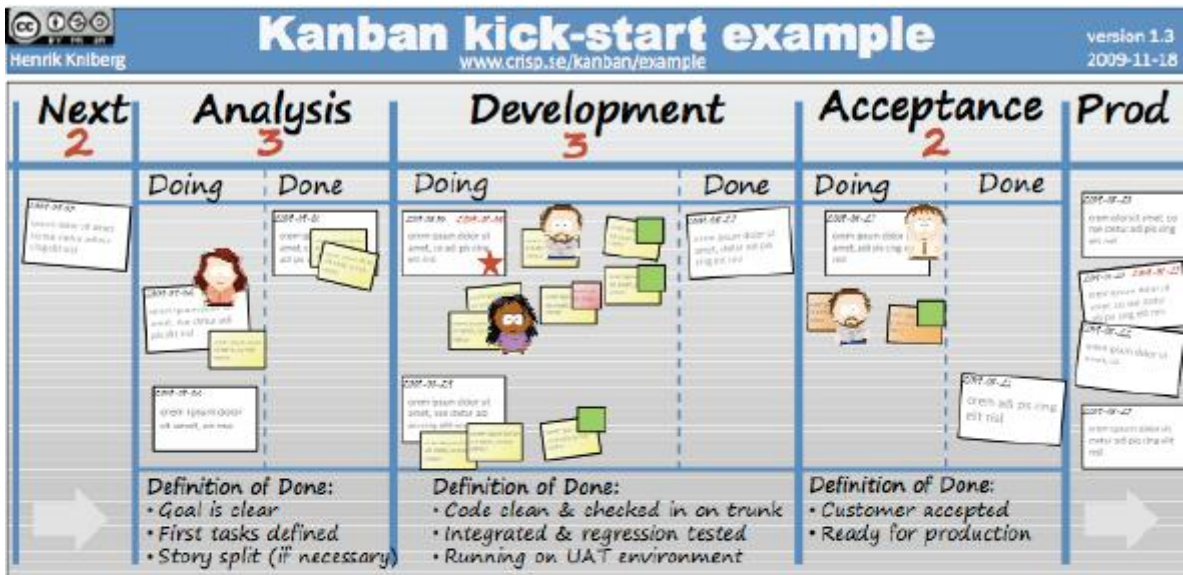
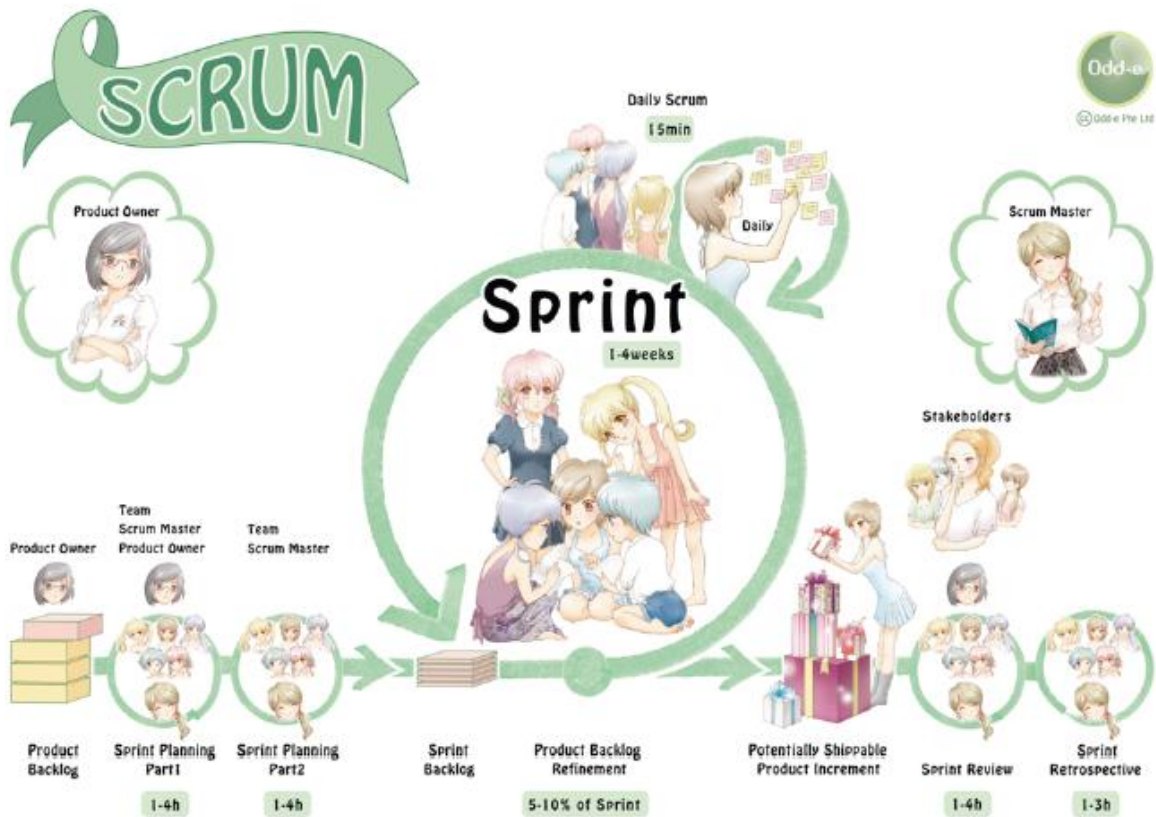
หลักการสำคัญ (core values) ของการพัฒนาซอฟต์แวร์แบบบอจิล์ (Agile) 4 ข้อ ได้แก่

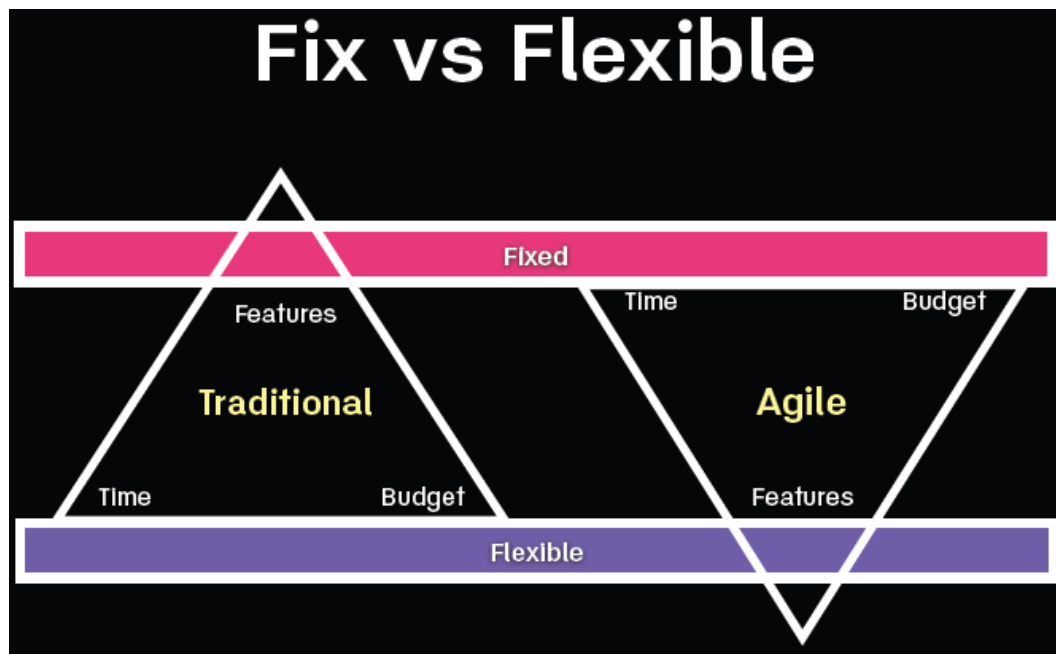
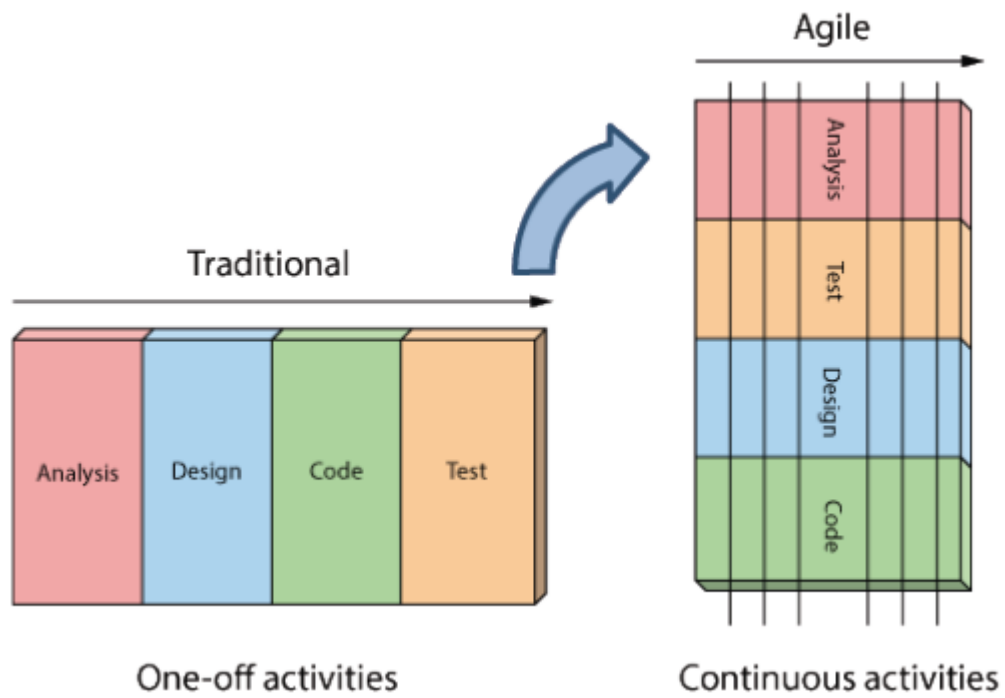
1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

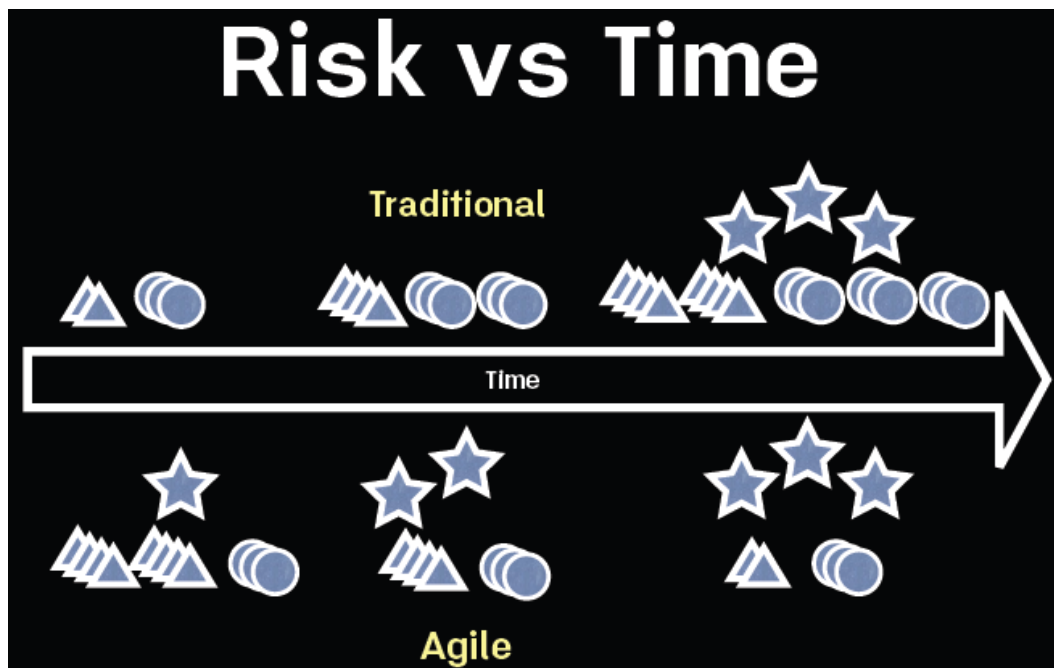
อีก 12 หลักการ ได้แก่

1. Satisfy the customer
2. Welcome change
3. Deliver frequently
4. Working as a team
5. Motivate people
6. Talk, face-to-face
7. Measure working software
8. Maintain pace
9. Excellent at quality
10. Keep it simple
11. Self-organize team
12. Reflect and Adjust regularly



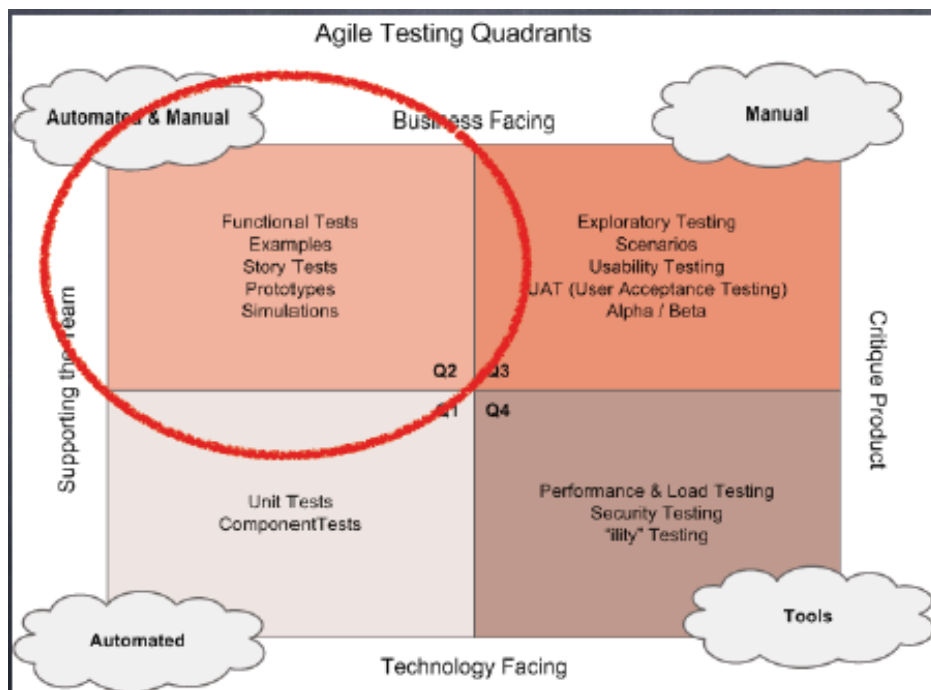






Agile vs Traditional Waterfall

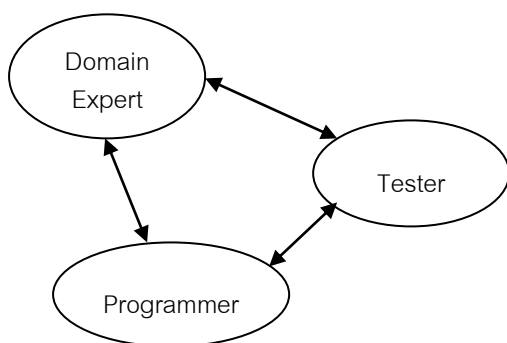
Metric	Waterfall	Agile
Planning Scale	Long-Term	Short-Term
Distance between customer and developer	Long	Short
Time between specification and implementation	Long	Short
Time to discover problems	Long	Short
Project schedule risk	High	Low
Ability to respond quickly to change	Low	High



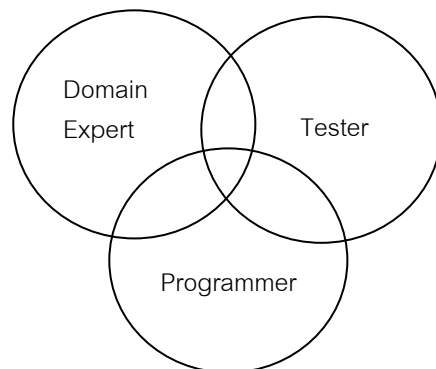
Some Facts of Testing

- ALWAYS be a chance that software will have bugs
- CANNOT prove software works by testing it
- DOES NOT prove that software is bug-free
- Testing after software has been created **DOES NOT** improve its quality
- Cost of removing bugs rises **DRAMATICALLY** in the later phases of the software development process
- **QUALITY** depends on 3 factors: PEOPLE, PROCESS and TOOL

ใครต้องรับผิดชอบเกี่ยวกับคุณภาพ เมื่อซอฟต์แวร์ทำงานไม่ตรงตามที่ต้องการ



ถ้าเป็นองค์กรแบบเดิม ต้องมีใครสักคนรับผิดชอบ



แต่ทางเลือกใหม่ คือ รับผิดชอบร่วมกันทั้งหมด

Testing Level

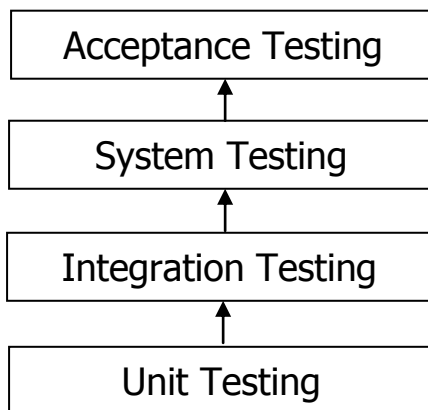
Functional Testing

Unit testing
Integration testing
System testing
Acceptance testing

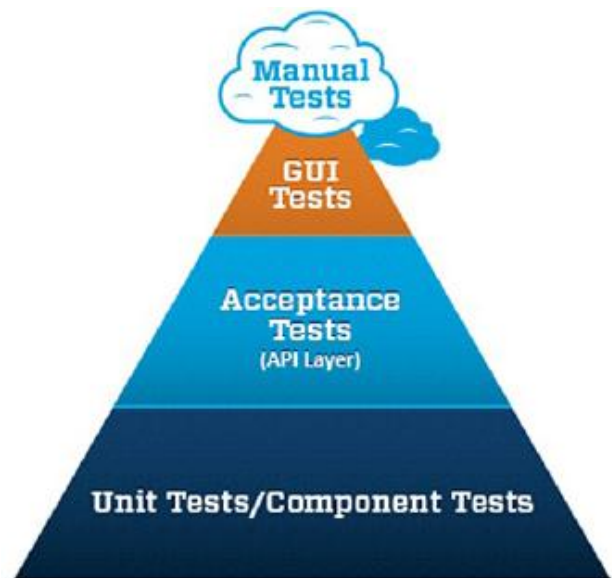
non Function Testing

Performance testing
Security testing
Usability testing
Compatibility testing

Simple Testing Level



Test Automation Pyramid



Mike Cohn's test automation pyramid.

$$\text{Acceptance Test} = \begin{matrix} \text{Acceptance Criteria} \\ + \\ \text{Examples (Data + Scenarios)} \end{matrix}$$

โครงสร้าง Test ที่ดี

“ควรกำหนดโครงสร้างให้ชัดเจน”

```
@Test
public void inputOneShouldBeSayOne() {
    //Arrange
    FizzBuzz fizzBuzz = new FizzBuzz();

    //Act
    String actualResult = fizzBuzz.say(1);

    //Assert
    assertEquals("1", actualResult);
}
```

คุณสมบัติของ Test ที่ดี

1. ทำงานรวดเร็ว (ในระดับวินาที)
2. อ่านง่าย และ เข้าใจ
3. เล็ก และ เรียบง่าย
4. Focus หมายถึง ในแต่ละ test case ควรมี 1 assertion
5. มีชื่อ test case ที่ดี
6. ควรมีการ Clean up หลังจากทำงานเสร็จ

Test ที่ไม่น่าจะดี

1. ใน Test case มีรายละเอียดมากเกินไป
2. มีการ Setup ต่างๆ ก่อนการทดสอบมาก



2.6 สถานที่ฝึกอบรม

เขตอุตสาหกรรมซอฟต์แวร์ประเทศไทย ถนนแจ้งวัฒนะ อำเภอปากเกร็ด จังหวัดนนทบุรี ประเทศไทย



3. ประโยชน์ที่ได้รับ

3.1 ประโยชน์ที่ผู้รับทุนได้รับ

- 1) ได้รับความรู้และทักษะในเรื่องของการประกันคุณภาพซอฟต์แวร์ และมีโอกาสพัฒนาเป็นนักประกันคุณภาพซอฟต์แวร์ (QA: Quality Assurance) เพื่อให้สอดคล้องกับการขยายตัวของอุตสาหกรรมซอฟต์แวร์ในอนาคต
- 2) ได้รับความรู้เพื่อนำมาเขียนเอกสารการสอนในเอกสารการสอนชุดวิชาการพัฒนาระบบสารสนเทศและชุดวิชาวิศวกรรมซอฟต์แวร์ ในหลักสูตรวิทยาศาสตรบัณฑิต แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร และถ่ายทอดให้กับนักศึกษาในกิจกรรมการฝึกปฏิบัติเสริมทักษะชุดวิชาการโปรแกรมคอมพิวเตอร์
- 3) ได้ทราบความก้าวหน้า แนวโน้ม และทิศทางของอุตสาหกรรมซอฟต์แวร์
- 4) ได้พบปะแลกเปลี่ยนความคิดเห็นและประสบการณ์ในเรื่องของการทดสอบและการประกันคุณภาพซอฟต์แวร์กับวิทยากรและผู้เข้าร่วมอบรม

3.2 ประโยชน์ที่มหาวิทยาลัยได้รับ

- 1) บุคลากรของมหาวิทยาลัยมีความรู้เกี่ยวกับการประกันคุณภาพซอฟต์แวร์ และทราบความก้าวหน้า แนวโน้ม และทิศทางของอุตสาหกรรมซอฟต์แวร์
- 2) เอกสารการสอนชุดวิชาการพัฒนาระบบสารสนเทศมีเนื้อหาที่ทันสมัยและสามารถให้ความรู้กับนักศึกษาได้สอดคล้องกับสถานการณ์ที่แท้จริงในกระบวนการทดสอบและประกันคุณภาพซอฟต์แวร์

4. ข้อเสนอแนะ

บุคลากรที่ทำงานเกี่ยวข้องกับเทคโนโลยีสารสนเทศและการสื่อสาร ไม่ว่าจะสาขาวิชาการหรือสายสนับสนุน ควรได้รับการฝึกอบรมในหลักสูตรที่เกี่ยวข้องกับการปฏิบัติงานอย่างสม่ำเสมอ เนื่องจากเทคโนโลยีสารสนเทศและการสื่อสารเปลี่ยนแปลงไปอย่างรวดเร็วตลอดเวลา