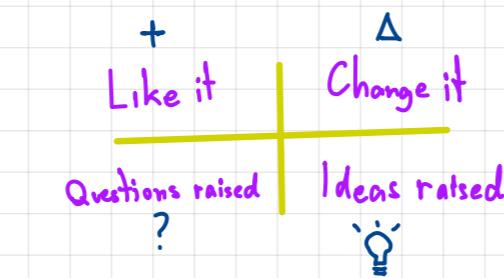


Design Thinking (Test)

- Test ให้แล้วกัน Feedback
- สร้าง Prototype อย่างมีประสิทธิภาพ Test ผ่าน mind set เทคนิคการเข้าร่วม (groupthink)
- Storytelling
 - เล่าเรื่อง
 - ติดปีกไว้
 - มองว่าอะไรหลังจากเปลี่ยนไป
- Storyboard
- พาบานไปที่ต้องนัดกับ User กำหนดหัวข้อ
- ให้ user บรรยายประสบการณ์ในเชิงลึก
- ฟังกุญแจไม่ที่เข้าสู่ประตู ให้เข้าไปช่วยไข้ดูดีด้วย
- ถาม >> Learn more About user, need, insight

► Feedback Capture Matrix



Software Development Goal

→ ทำน้องจาก DT

- ▶ เนื้อที่ต้อง จะเน้นต่อค่าผล หรือชั้นเดียวกัน ว่าจะต้องทำอะไร

ex. ลองดู เป้าหมายนี้

- เรียนให้ได้เก็บรักซึ่งมีลักษณะดังนี้
 - ข้อมูลนี้ต้องใช้ในการทำงาน

"Goal are about change"

ต้องเปลี่ยน → ไม่ถาวร

ก็ต้อง ตรวจสอบให้ได้ไปใหม่ใหม่



มีชื่อว่า Monitor and change

Software = app (siam say "ฟรานาณ")

- ▶ Computer Programming Vs IS (information System)

Computer Programming (app)

→ software ที่ทำงานบนอุปกรณ์ของ Computing device

ex. PC, mobile, Game Console

Software System or IS → คุณต่างคือ ไฟโน๊อกส์ที่ Business (func ทางธุรกิจ) → รวมรวมช่องทาง

- ก้าวแรก app
- ต่อไป Database System
and ครอบครุณถึงขั้นตอนที่คุณต้องทำ

ทำด้วย หัวใจ และหลักธรรมาภิบาล

ช่องทาง

- ลูกค้า

- ช่องทาง

- บินล่า

ผลลัพธ์ช่องทาง ฝรั่งเศส

- ▶ Software Artifacts → สิ่งที่สร้าง Software

- Codes
- Data structures โครงสร้างของ Data
- Documents เอกสารที่อธิบาย

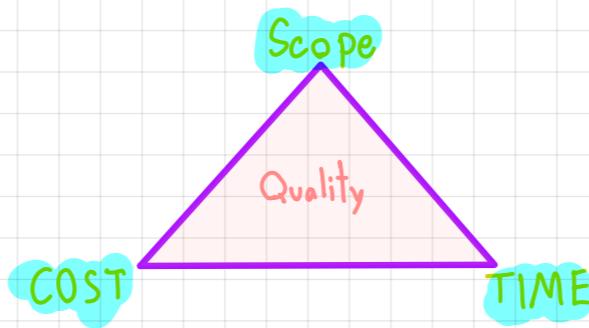
Recall → Goal of organization, e.g.

- Revenues (รายได้)
- Profits (กำไร)
- Costs (ค่าใช้จ่าย)
- Sustainability (ค.นั่งบัง)
- Ranking (ติดอันดับ)
- Recognition (เป็นที่จดจำ)

Output (ผลผลิต) → Products ภาระ กิจกรรม

Outcome (ผลลัพธ์) → changes (เปลี่ยน)
benefits (ผลประโยชน์)
learning (เรียนรู้)
เกิดจาก Products, service or กิจกรรม

► Project Management Triangle



► Want vs Need

พัฒนากร้าวๆ ก็ได้ <- want → ทุกเช้าต้องดินกาแฟ เมื่อเริ่มต้นวันได้ฤทธิ์
Insight → สิ่งที่ทำให้เกิดความต้องการในเรื่องสักลายต่างๆ
ต้องหาเช้า <- need → ต้องหาเช้าต้อง เมื่อไม่พบคนรักคงไม่ได้เช้านั้นแต่เริ่มต้นวัน

ก้ากัน → ถูกต้อง
ก้าไม่กัน → ก้าไม่เข้าเป็นค่ายกัน

กัน
กัน จ่าเป็นต้องห้ามให้วันนี้
ความคุณต้องไว้ได้ เมื่อว่าง
ก้าไม่ห้ามความคุณต้องไว้ไม่ไว้

► Implicit vs Explicit

(ໂຄສະນິບ) Implicit → ນອກໄວ່ເຊື້ອ ແຕ່ກວາດຕົວ

(ชุดที่ 6) Explicit → น้องได้ชัดเจน กะบีมีรายละเอียด

▶ Sample Need and Expectations → រាយការណ៍នីមួយៗ

Whose Need?

- Customer?
 - Customer is king / god
 - Developers?
 - Stakeholders?

► Stakeholders

- ผู้ที่มีส่วนได้เสียในธุรกิจ → ผู้ใด ผู้ใด
- Customers ลูกค้า
- Users ผู้ใช้
- The workforce แรงงาน
- Partners
- Governing board คณะกรรมการบริหาร
- Stockholders ผู้ถือหุ้น
- Policy makers ผู้กำหนดนโยบาย
- Local and professional communities ชุมชนท้องถิ่น ทางอาชีพ

Quality

คุณภาพ
Satisfy customer needs and expectations
(ตามต้องการของลูกค้า)

มาตรฐานที่ต้องการ

Quality Software → ISO/IEC

Software Development Activities

► Building Analogy

- Buyer e.g. client, Customer

▷ ต้องการนักก่อสร้าง

- A vision → วิสัยทัศน์
- Need

▷ Options (ทางเลือก)

- ชื่อสีรูป
- สร้างเองแบ่งขาย
- ห้องคนอื่นสร้าง

- Construction workers

- Need detail of the house

- Blueprint แบบ

- ก่อสร้าง ออกแบบให้พิมพ์เขียว ก่อ

- คุณภาพสร้าง พิมพ์เขียว ดีอิ สูงมาก ร่วมกัน วิจัย

- Architect

- ผู้ออกแบบ ให้ความต้องการ ประยุกต์

- ใจลึกถูกตัว ที่ต้อง need → ผู้ออกแบบ ให้ความต้องการ ฝ่ายอื่น

- Define solution → กำหนดวิธีแก้ปัญหา

- ออกแบบ Detail ลง ข้อกำหนด

- ตรวจสอบว่าเป็นไปตามข้อกำหนดไม่

Buyer with a vision and needs (มีมองเห็น) -> ผู้ซื้อ

↑ ผู้ซื้อ

Architect ผู้ออกแบบ

→ ผู้ซื้อ

→ ผู้สร้าง

↓ ผู้ก่อสร้าง

Constructors with team and expertise

► Client/Customer with a vision and needs

- Software របៀបទូទៅ
 - Commercial off the shelf (COTS) មេដាយលើក
 - Cloud-based software-as-a-service such as O365 (អេឡិចត្រូនិក), ផ្លូវការលើក
 - Open Source → Software Libre
- In-house software → មេដាយខ្លួយ
- ផ្សាយ

► Process

- ខ្លួយ
 - Activities → សែវភាពរបៀបទូទៅ Empathize ឱ្យស្ថានភាព
- ការងារ
 - Actions → សំណើនាំ Activities របៀប Journey Map, What How Why, User Persona
- ការងារ
 - Task → ឱ្យការងារត្រួតពេលវេលា

► Software Process Activities

- Communication កុំពិនិត្យ
- Planning រាយរោង
- Modeling (រួមចារ៉ាស់) គំនិតរឿង
- ការការពីការងារដែលត្រូវ need/req
- ឬ Design ឧបករណ៍
- Construction (រៀបចំ)
 - code generation ផែនតែង
 - Testing >> user acceptance testing (UAT)!
- Deployment ដំឡើង user ទៅ server ជាបន្ទីរ
- ផ្តល់ព័ត៌មានពីការងារទៅ server ជាបន្ទីរ → Production Server
- Maintenance កំបាត់ការ កំបាត់រាយរោង ≠ កំហែង

Underlying Software Traits (ลักษณะของ Software)

ซับซ้อน - Software is Complex → ไม่สามารถคำนวณตัวตนของทุกสิ่งที่เกิดขึ้น เพราะมีมิติมากและมีความซับซ้อน

- ไม่ยอมแก้ แต่ยอมรับ และหาวิธีมาจัดการโดยการแบ่งเป็นส่วนๆ

ร่วมต้องไม่ได้ - Software is Intangible → ร่วมต้องไม่ได้ ไม่มีรูปธรรม ทำกัน

- ต้องการเข้าใจยาก และคุ้นเคยต้องการให้อ่านง่ายๆ

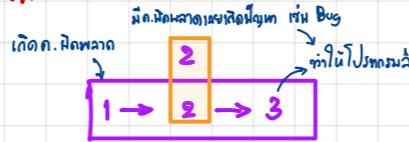
need → จำเป็น - From need to Artifacts

ต้องการให้มีรูปธรรม

- Mistakes always happen
↓ to error is
human
- Miscommunication → สื่อสารผิดพลาด
- Misunderstanding → เห็นใจไม่ตรงกัน

terminology

- Mistake 1
- Fault 2
- Failure 3
- Error container
- Defect ร่องรอย
- Bug 2
↳ เอกพจน์ “มีบีก”
ภาษาอังกฤษ “ค่าไม่ถูกต้อง”



Continuous Dev and Maintenance

→ มีรูปร่าง

- Software ต้องมีชีวิตอยู่ตลอด
- ผู้ใช้งานจะอุปถัมภ์
- OS support
- Maintenance
 - Traditional model : develop then maintain
(แบบเดิม) (พัฒนา แล้ว บำรุง)
 - Modern maintenance : ปรับปรุง เพื่อ improve (แบบใหม่)

Working Team

- 5C →
- Compassion เท็หอกันให้
 - Character มุตติก
 - Courage ด.กล้า
 - Critical Thinking คิดมากก่อน
 - Collaboration การทำงานร่วมกัน

Working Group

- Individual work products มุ่งส่วนตนบุคคล
- Individual accountability รับผิดชอบส่วนตัว
- Strong, clearly focused leader ผู้นำดุดันแต่ชัดเจน
- มีเป้าหมายร่วมเดิน (เป้าหมายของตัวเอง)
- พูดคุย ตัดสินใจ บนพื้นที่
- ไม่ชุมนุมอย่างมีประสิทธิภาพ

* accountability
ก่อสร้างผิดชอบ

Working Team

- Collective work product มุ่งทีมงานร่วม
- Individual and mutual accountability, commitment ร่วมผิดชอบทั้งร่วมเดิน และร่วมบุคคล, ภาระภารกิจร่วมกัน
- Shared leadership roles ร่วมกันเป็นผู้นำในเรื่องต่างๆ
- มีเป้าหมายเดียวกันทั้งทีม Common goal (มี goal เพื่อทีมกัน)
- พูดคุย ตัดสินใจ ทำงานร่วมกัน แห่งทีมงาน และต้องมีใจร่วมกัน
- ประชุมแบบภายใต้ lider กัน ใช้ภาษาบอร์ด ทางการสักพักใหญ่ แห่งทุกดูษ → ผู้คนหาอย่างต่อเนื่อง (ทีม, งาน, skill, social)

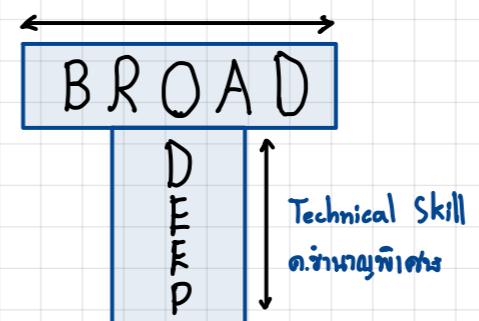
Musketeer Attitude

"All for One and One for All"

ทุกคนมุ่งสู่เป้าหมายเดียวกัน แต่ละคนทำภาระหนึ่งกัน

T-Shaped Skill

ต.สามารถใช้งานกว้าง ex. โทรสื่อสาร



Constantly learning
(อยู่ตลอดเวลา)

พัฒนาอย่างต่อเนื่อง

► Self-Organizing

team ผลลัพธ์ของทีมที่ตั้งใจทำด้วยตัวเอง

► Self-Reflection (การสะท้อนตนเอง)

- หลักการจัดการ

- 7/9 ภารกิจที่สำเร็จได้

- เคล็ดลับค่าแรงงานที่ดีที่สุด

► Time Management จัดการเวลาอย่างมีประสิทธิภาพ

► Procrastination → หลีกเลี่ยงการเรียนรู้

► ใช้เวลาทำงานไปก่อนงานที่ less important

► Take your time → ห้ามเร่งรีบ

► กำหนดห้าม deadline

► Over-estimation → ประเมินเวลาทำงานเกินไป

► Under-estimation → ประเมินเวลาทำงานน้อยเกินไป

► Unrealistic goal → ตั้งเป้าหมายที่เป็นไปไม่ได้

► ตั้ง end milestone ให้แน่น

► Trust Equation

$$T = \frac{C + R + I}{S}$$

C - ศักดิ์ศรี (ความเชื่อถือ)
 R - ความซื่อสัตย์ (ความจริงใจ)
 I - ความไว้วางใจ (ความน่าเชื่อ)
 S - think วิเคราะห์

Trusting + Trustworthiness = Trust

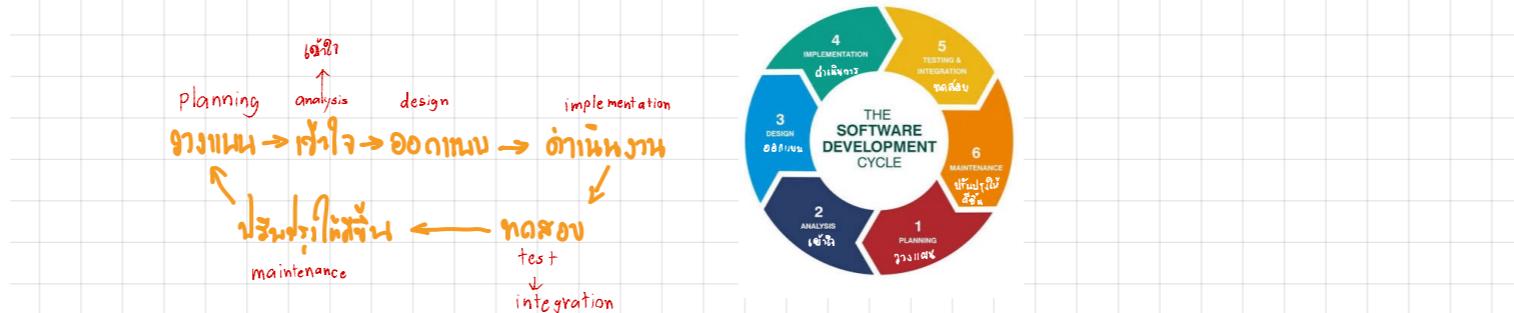
► Improving Engagement

- Purpose ค. ตั้งการ
- development พัฒนาตนเอง
- a caring manager ผู้ดูแลดี
- ongoing conversations จุติคุยกันป้องกัน
- ทำงานที่เป็นอุดหนู
- How to people more engaged? วิธีทำให้ผู้คนมีส่วนร่วม
 - ใจดี ล้ำลึก คิด ถอดรหัส ของทุกคน
 - ทำให้รู้สึกว่า งานที่ทำแล้วทำมีผลต่อหัวเราะ
 - ความตกลงที่จะมอบหมายให้กัน?
 - เท่าไหร่ใน aras และ ทำให้ปัญหาร่วมกัน

Software Development Life-Cycle Models

จุดนูนารายการ Software : สิ่งที่คุณต้องทำ กันต.ต้อง กดชั่ง ใจจะเป็นคนชนะเรื่องพื้นที่ลง
ด.เข้าใจของท่านต่อไปได้ตรงกันและกัน แบบเป็นผู้เชี่ยวชาญ

สิ่งที่ต้องแก้ไขให้ดี : Software Dev Life-cycle or SDLC



1.1 Code and fix model

- ไม่มีการออกแบบ วิเคราะห์ ตามที่มีการ ข้อกำหนด
- เป็นไปได้ ผิดพลาด รุนแรง

1.2 Waterfall mode

สรุป : ขั้นตอนเดือน SDLC แนวตั้ง

- 1 requirement ความต้องการ
- 2 Analysis วิเคราะห์
- 3 Design ออกแบบ
- 4 implementation ดำเนินงาน
- 5 Post-delivery Maintenance บำรุงรักษา

► Problem of waterfall

- ต้องรองรับการเปลี่ยนแปลงที่ต้องดึงด้วย Software ใหม่ๆ
- หา Test ยากที่สุด พัฒนาไปตรงสีบานต้องการซ้อมคูกัดๆ
 - High risk ภัยคุกคาม ต้องเรียนรู้ใหม่ ทำให้เสียเวลา
- ต้องเขียน doc ทันทันใดในส่วนต่อไป ให้เขียนอีก
 - ทำให้เกิดความซ้ำซ้อน human error
- ขาดการเปลี่ยนแปลงที่สำคัญที่สุด
 - เช่น OS ใหม่ๆ ที่ต้องไปซื้อต้องใหม่
- ถ้ายากลิกกากลาก็ต้อง Product ที่มีงานได้ดีมาก

วันนี้

1.3 Iterative and Incremental Development [IID]

- เท็มชีหกนล่าตนคานสำเร็จ, งานที่ต้องทำ เช่น ฝึกอบรม → ถอนพัน

- ทำในช่วงเวลา

ข้อดี

- จัดการกับ Unknown & Changes ได้ดีกว่า Waterfall
方

- Feedback, learning, improvement

เสริมทักษะที่สำคัญ ให้กับผู้เชี่ยวชาญ Business and technical และทำให้มีพัฒนาการเรื่อยๆ

- Working Software ได้ช่วงต้นๆ แต่ๆ ได้เริ่มต้น เพราะ หากตอบแล้วเป็นรูป

- User can validate early

- Return On Investment (ROI)

ผลตอบแทนของทุน
เริ่มต้นทันเพื่อเข้าตัวเร็วๆ

Scrum Overview

▶ Definition of Scrum

→ เป็น framework เพื่อจัดการกับปัญหาที่มีด. ชั้นชั้น และ ปรับเปลี่ยนได้เรื่อยๆ

→ ใช้เวลาชั้นชั้นเป็นไป ก่อตากำหนดเป้าหมาย แล้วต้องร่วมมือกัน ร่วมมองดูผลลัพธ์ให้ดูกล้า

Scrum Is :
- Lightweight ไม่ซับซ้อน
- Simple to understand ทำให้เข้าใจง่าย
- Difficult to master ยากที่จะเข้าใจแต่ เมื่อเข้าใจแล้วจะสามารถนำไปใช้ได้

▶ “Unknown unknowns”
เราไม่รู้ว่า เราไม่รู้ว่ามันคือ

▶ การที่เราคาดหวัง effect จะเป็นไปได้ Maj retrospect (มองข้างหลัง)

▶ Contemporary business, markets
(ร่วมสมัย)

▶ Probe - Sence - Respond
ทดลอง ดูแล 修正

▶ คณะกรรมการไปได้ ต้อง reflect

▶ ท้องตรงด. ติดสร้างสรรค์

▶ Explore เพื่อเรียนรู้เกี่ยวกับโลก
(สำรวจ)
inspect เพื่อตรวจสอบ

adapt 修正
fail-Safe → ต้องมีการตัดสินใจไม่ทำจนเกิดภัยไว้

▶ Safe-fail → ต้องมีสิ่งที่ suces หรือรุ่งฟื้นตัว ลักษณะนี้

▶ ดูยังไงมากกว่า ทำได้ หรือไม่ได้

some ulige

► Key Concepts

- ▶ งานที่ทำ คาดคะเนว่าได้ และ complex
เห็นรูปแบบที่คาดคะเนว่าต้องทำ
- ▶ Based On Observ and Experimentation
 - (การสังเกต)
 - (การทดลอง)
 - ▶ ทำไปต่อไปจนกว่าจะถูก

Product

↑
Process

↑
Backlog

(รายการที่ต้องทำ ไม่แน่นอน)

▶ Transparency → ทุกคนใน team จะต้องรู้ว่ากับเราต้องนั่งเขียนบันทึก ไม่ใช่แค่รอตัวเอง

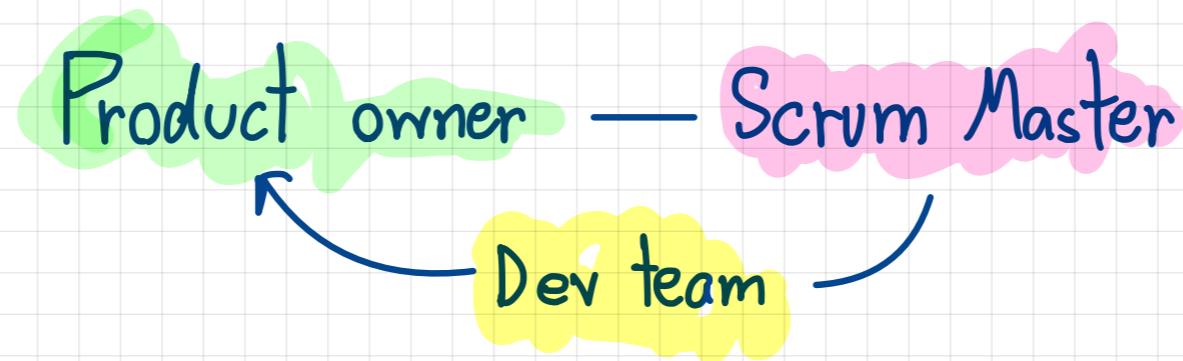
▶ Inspection → ตรวจวินิจฉัยทุกสิ่ง about Software
ความดีเด่นที่ให้มาอย่างน่าภูมิใจ แต่ก็มีข้อเสียที่ต้องปรับปรุง
เพื่อหาว่า สิ่งที่ทำอยู่ เมื่อเวลา 1 วัน ไม่ตรงหรือไม่

▶ Adaptation → เมื่อ Inspect แล้วเจอ บางสิ่ง Must be adjusted ให้ทัน
as soon as possible
(ทำให้เร็วที่สุด ทำต่อไปได้)

เช่น 199 ใน Sprint_1 ควรรีบแก้ทันทีใน sprint_1

► Scrum Team

- ทีมที่ • flexibility
 - Creativity
 - Productivity
- Self Organizing and Cross Functional
 - ↓
จัดการกีโน่อในทีม
ทุกคนรู้หน้าที่ของ
 - ↓
team have all competencies
(ท.ภายนอก)
ไม่ต้องหานักเชี่ยวชาญ



PO → - Manage backlog
- เช้าใจ user need & Requirement
- รับผิดชอบการสร้างศูนย์ค่าของ Product (คุณภาพของสิ่งที่ต้องการ)

SM → • Promote and Support Team
• Being Coach ให้แก่ทีม

Dev Team → รับผิดชอบในการพัฒนา ออกแบบ
building, testing the product