Software Engineering <Mid-Term>

Product

1. Software คือ set ของ item = configuration ซึ่งปกติคือ program, document, data
2. Software Characteristic
    a. ไม่ผ่านการผลิตแบบ Hardware สร้างคนละแบบ
    b. Quality ขึ้นอยู่กับ Design, People
    c. Failure Rate สูงในช่วงแรกทั้งคู่ แล้วค่อยๆลดลง แต่ Hardware จะสูงตอนใช้ไปนานๆ(พัง) ขณะที่ Software ลดลงไปเรื่อยๆจนล้าสมัย แต่จะสูงขึ้นใหม่เมื่อมีการปรับปรุง software
    d. Hardware สามารถเปลี่ยนส่วนภายในได้ถ้าพัง แต่ Software ทำไม่ได้
    e. Software ส่วนใหญ่ถูกสร้างขึ้นเองมากกว่าการรวมของ Component ที่มีอยู่แล้ว
3. Software Application มี 7 ชนิด
    a. System Software *บริการ Software อื่นๆ*
    b. Real Time Software *monitor/analysis/control Real world event*
    c. Business Software
    d. Engineering and Scientific Software
    e. Embedded Software *อยู่ใน RAM เพื่อควบคุมอุปกรณ์*
    f. PC Software
    g. AI Software *ใช้แก้ปัญหายากๆ (ใช้คอมแก้)*
4. Software Flexible ขึ้นอยู่กับ Requirements โดยผลกระทบจากการปรับปรุงจะสูงขึ้นถ้าเวลาพัฒนาผ่านไปนานขึ้น
5. Software Engineering คือการ Design และ Develop High- Quality Software ให้ทันเวลาและไม่เกินงบ
6. Computer Scientist มองที่คอมพิวเตอร์และภาษาคอม พิสูจน์Algo แต่ Software Engineer มองของเหล่านั้นเป็นอุปกรณ์ที่ใช้ในการออกแบบและสร้างเพื่อแก้ปัญหา
7. Participant in Developing a Project
    a. Customer — who is paying for software system to be developed
    b. Developer — who is building the software system
        i. ประกอบไปด้วย Requirements, Designer, Programmers, Tester, Trainer, Maintenance Team
    c. User — who actually use the system

Process

1. Layer Technology
    a. Quality focus — Software Engineer Focus
    b. Process — ทำให้ Delivery มีประสิทธิภาพ
    c. Methods — สร้าง Software อย่างไร
    d. Tools — support process and method
2. ความหมายของ Process : Sequence of step, Set of ordered Task
    a. ลำดับการทำงานในแต่ละ Task เหมือนกัน
    b. การสร้าง Software ➔ Software Lift Cycle
3. Process Characteristic
    a. ทุก step จะมี Activity, Constraints, Resource
    b. สามารถมี Tool, Technique
    c. อาจมี sub process เช่น Design ประกอบด้วย Interface DatabaseDesign
    d. ต้องมี entry และ exit Criteria (มีเริ่มและจบ)
4. Software Process
    a. Common Process Framework
        i. Framework Activities
            1. Task Sets
                a. Tasks
                b. Milestones (แต่ละงานเสร็จเมื่อไร), Deliverables (กำหนดส่ง)
                c. SQA Point (Software Quality Assurance point)
        ii. Umbrella Activity ต้องทำทุก process ปกติ SQA SCM
5. Capability Maturity Model Integration (CMMI)
    a. Model ที่วัดขีดความสามารถของผู้พัฒนา Software เพื่อตัดสินว่า Current State ถึงไหน
    b. สมัยก่อนคือ CMM มี 5 ระดับ แล้วพัฒนาเป็น CMM
        i. Continuous Model ดูหัวข้อ ระบุคะแนน 0-5
        ii. Staged Model ดูภาพรวมองค์กร โดยให้ 1-5
6. CMMI แบบ Continuous Model
    a. ระบุ Specific Goal & Specific Practices (SG & SP)
    b. ให้คะแนนเป็น Capability Level(CL)
    c. การให้คะแนนของ CL
        i. Level 0 : Incomplete : ไม่ได้บอก assess / ไม่สำเร็จตาม SG, Objective
        ii. Level 1 : Performed : พอไหว เสร็จแบบฟลุ๊ค ทำงานได้แต่เพราะคนไม่ใช่ระบบ

1. SG ใน Program Area เป็นที่น่าพอใจ

iii. Level 2 : Managed : โครงการสำเร็จตาม Goal 3-4 โครงการ ไม่จำเป็นต้องทั้งบริษัท มีกระบวนการในการพัฒนา Software ที่จัดการได้ repeat ได้

1. มี basis infrastructure
2. มีคนมีความสารมารถพียงพอ
3. ผู้ที่ได้รับผลกระทบ
4. Process ต้องมี monitored, controlled and review

iv. Level 3 : Define : ต้องทำได้ทั้งองค์กร

1. มี Tailored = การดัดแปลงให้ project มีประสิทธิภาพมากขึ้น

v. Level 4 : Quantitatively Managed : วัดได้ วิเคราะห์ได้ ปรับปรุงได้เรื่อยๆ

1. ประเมิณเพื่อวัดคุณภาพโดยใช้ Statistic & Technique อื่นๆ

vi. Level 5 : Optimizing : มีหลักฐานชัดเจน (lv4 ขึ้น lv5 ไม่ยาก แค่ทำ SPI)

1. SPI : Software Process Improvement
    a. PSP Personal Software Process
    b. TSP Team
    c. Six Sigma TQM

d. Project Planning : SG & SP

i. SG 1 การประมาณ

1. SP1.1-1 Project Scope
2. SP1.2-1 Work Product & Task Attributes
3. SP1.3-1 Project Life Cycle ลำดับการทำงาน
4. SP1.4-1 Effort & Cost

ii. SG 2 เขียน Flow Chart

iii. SG 3 Team member ทำตาม Plan

7. CMMI แบบ Stage Model
    a. Perform : ยังไม่ได้วัดก็ได้ในทุกองค์กร
    b. Managed : Basic Project Management
    c. Define : Process Standardization
    d. Quantitatively Managed : Quantitatively Management
    e. Optimizing : Continuous Process Improvement
8. Linear Sequential Model
    a. Classical Life Cycle or Water Fall Model
        i. System/ Information & Engineering Modeling (จริงๆคือ Analysis, Design)
        ii. Software Requirement Analysis Modeling

iii. Design
iv. Code generation (Implement)
v. Testing
vi. Support

b. Waterfall Model
i. ต้องเสร็จ state ปัจจุบันจึงไป state ถัดไปได้ เป็น Sequence ไม่สามารถข้ามได้
ii. นักพัฒนามองเห็น Lay Out เป็นอย่างไร บอกว่าสิ้นสุดที่ส่วนใด แต่ไม่ได้บอกว่าจะเปลี่ยน state ต้องทำอย่างไร
iii. No Prototype, No Guideline to Handle Change
iv. ช้าเกินไปที่จะให้ลูกค้าดู ➔ แบ่งงานเป็นส่วนๆแล้ว Deliver

9. V-Model (มีการ Testing ในทุกๆ Phase)
a. แบ่งการทำงานเป็น 2 ฝั่ง
i. ฝั่งซ้าย ➔ Analysis, Design
ii. ฝั่งขวา ➔ Testing, Maintenance
b. ถ้า Verification และ Validation พบปัญหา จะให้ฝั่งซ้ายทำการ Re-executed เพื่อ fix, improve ทันทีก่อนที่จะเปลี่ยน step ทางด้านขวา
c. การแก้ปัญหาเมื่อพบปัญหาที่ ขวา ไปซ้าย
i. Unit Test ➔ Method in Class ➔ Detail Design
ii. Integration Test ➔ Class Diagram ➔ High-Level Design
iii. System Test ➔ Req. Spec. Document ➔ Requirements Analysis

10. Prototyping Model : Repeat จนลูกค้าพอใจ
a. มี System บางส่วนหรือทั้งหมดที่สร้างอย่างเร็วให้ลูกค้าได้เห็น
b. มีขั้นตอนดังนี้ (วนไปเรื่อยๆ)
i. Listen to Customer — Requirement Gathering
ii. Build/Revise Mock-up — สร้าง Prototype
iii. Customer test driven Mock-up — user ประเมิณ Prototype
c. มี 3 แบบ
i. Throwaway Prototyping — สร้างอย่างเดียว ทิ้งไม่ได้ใช้
ii. Partial Prototyping — เห็นภาพทั้งหมดก่อนแล้วลงมือทำ
iii. Evolution Prototyping — พัฒนา เพิ่ม code ทุกครั้งที่ Listen

11. The RAD Model (Rapid Application Development)
a. ไม่ได้เริ่มจาก 0 เหมือน Waterfall แต่ Reuse Info/Component
b. ทำให้สร้าง Fully Functional System ในเวลา 2-3 เดือน
c. แบ่งออกเป็น 5 Phases

i. Business Modeling — understand current business process
ii. Data Modeling — attribute, relation between object
iii. Process Modeling — process description
✓ add, delete, modify, retrieve,
iv. Application Generation — ใช้ reuse component & reuse program
v. Testing & Turnover — new component, interface ต้อง TEST

d. 1 System สามารถแบ่งเป็นหลายทีม ➔ แบ่ง Fn ให้ทำงานไปพร้อมกันได้

12. Incremental Model (Phase Development Model)
    a. System ถูกแบ่งออกเป็น Sub System โดย Function
    b. เริ่มจากการ Deliver fn เล็กๆให้ลูกค้าดูก่อน แล้วค่อยๆเพิ่ม fn ขึ้นเรื่อยๆ
    c. ประโยชน์
        i. Training ตั้งแต่ Release ครั้งแรกๆ
        ii. Business สามารถเริ่มได้เร็ว ทันตลาด
        iii. Fix ปัญหาได้รวดเร้ว
        iv. ขั้นตอน (วน 2,3 ไปเรื่อยๆจน 2 สมบูรณ์)
            1. Implement and Test First Build
            2. Implement, Integrate และ test จนกว่า product สมบูรณ์
            3. Operation เพื่อเพิ่ม Functional
13. The Spiral Model (คล้ายๆ Iterative)
    a. ส่งรอบแรก = ส่งทุก subsystem โดยแต่ละ subsystem มีแค่บาง Fn แต่ increment model ส่งทีละ subsystem ตรบทุก fn
    b. Focus ที่ Risk Management และ Control Risk
        i. Risk Analysis แล้ว develop prototype to verify feasibility or desirability
    c. ขั้นตอน (วนไปเรื่อยๆ)
        i. Requirement & Initial Plan
        ii. ประเมิณ Risk และสร้าง prototype
        iii. เขียน Concept of operation เพื่ออธิบาย high Lv. ว่า system ทำงานยังไง

The Unified Process (UML = Unified Modeling Language เกิดจากแนวคิด Object-Oriented)

1. Phases of the Unified Process
    a. Communication
        i. Inception : Business Requirement แบบคร่าวๆ
            1. เขียน Use-Case แบบ Fundamental Business Requirement
            2. บอกแค่มี feature, function อะไรบ้าง

3. อธิบาย sequence action ของ actor ต่างๆ
4. Use-Case ช่วยในการระบุ scope, project planning

b. Planning
   i. Elaboration คุยกับลูกค้าเรื่องรายละเอียด
      1. นำ Use-Case ใน Inception มาเขียน 5 Model
         a. Use-case Model, Analysis Model, Design Model, implement Model, Deployment Model
      2. Review Plan เพื่อดู Scope, Risk, Delivery date ที่เหมาะสม

c. Modeling
   i. Construction Phase
      1. Analysis & Design are Complete
      2. ทุก fn และ feature ถูกเขียนใน Source Code
      3. Unit Test ทุก Component ต้องถูก Execute, Integration ด้วย
      4. ใช้ Use-Case เพื่อสร้าง Acceptance Test ใช้ใน Phase ต่อไป

d. Construction
   i. Transition Phase
      1. ส่ง s/w ให้user เป็น Beta Testing เพื่อ บอกปัญหา,จุดที่ต้องเปลี่ยน
      2. s/w team เขียน Document เช่น user manual
      3. ใช้ s/w increment จะค่อยๆกลายเป็น usable software

e. Deployment
   i. Production คือ Deploy กับ Operation
      1. ขณะที่ใช้ s/w มีการ monitored
      2. Defect Report กับ Request for Change
      3. เริ่มสร้าง next software increment

2. Unified Process Work Product
   a. Inception (VRRU) = Project Plan, Business Model
      i. **V**ision Document, Business **R**equirement, **R**isk, Initial **U**se-Case
   b. Elaboration (RAD) = Use-case, Analysis Model
      i. **A**nalysis Model(Collection of class), **D**esign Model, **R**eview Risk
   c. Construction (TID) = Design Model, Test Plan, Test Case, Support Document
      i. **I**mplement Model, **D**eployment Model**, T**est Model
   d. Transition (DF) = Deliver Software Inclement, Feedback
      i. **D**eliver SW Inclement, **F**eedback from beta testing

Project Management Concept

1. Concept ของ Management (Management Spectrum) = 4P's
   a. People, Product, Process, Project
2. People
   a. คนใน software process
      i. Senior Manager, Project/Technique Manager, Practitioners, Customer, User
   b. Team Leader Characteristics
      i. Motivation, Organization, Ideas or Innovation, Problem Solving, Managerial Identity, Achievement, Influence and Team Building
   c. Software Team
      i. Democratic Decentralized (DD- Egoless Programmer Team)
         1. คนน้อย <10 หัวหน้าอาจเปลี่ยนหน้าที่ไปมา ทุกคนต้องเก่ง แก้ปัญหายากๆได้
         2. เหมาะกับคุณภาพและปริมาณที่ดี และเวลาไม่จำกัดมาก
         3. ช้ากว่า Centralized เยอะ มีช่องทางในการคุยเยอะไป
      ii. Control Decentralized (CD)
         1. มี Project Leader ดูแล Senior Programmer ดูแล Junior Programmer
         2. Group แบ่งตาม Role Play มากกว่า Module
         3. Project Manager กำหนด Goal และแบ่งงาน โดยสื่อสารทางแนวตั้ง
      iii. Control Centralized (CC Chief Programmer Team)
         1. เหมาะกับงานขนาดเล็ก ต้องการความเร็ว
         2. Chief Programmer ทำแทบทุกอย่าง planning, coordinate,review
         3. Programmer 2-5 คน
         4. Backup Engineer support และแทน CP ได้
         5. Specialist เช่น Telecom expert, Database Designer
         6. Librarian รักษาและควบคุม work product
   d. ปัจจัยในการเลือก Team Structure
      i. Difficulty, Size, Duration, Modularity(แบ่งงานได้ไหม),Reliable, Time, Sociability
   e. วิธีการติดต่อใน Team Project
      i. Formal impersonal approaches ใช้ Doc ex. Milestone, change report
      ii. Formal interpersonal procedures ประชุมทางการ
      iii. Informal interpersonal procedures ประชุมไม่ทางการ
      iv. Electronic communication Email, Video conference
      v. Interpersonal Networking คุยกันอย่างไม่เป็นทางการ

3. Product
    a. กำหนด Objective & Scope ของ Product ➔ ดู input, output
    b. Objective : Goal ของ Product โดยไม่สนว่าทำได้ไหม
    c. Scope : บอกสิ่งที่จะทำ : Data(input/output), Function, Special Performance, Interface
4. Process
    a. เลือก Software Process Model ที่เหมาะ
    b. ระบุ Task, Activity, Milestone, Deliverables, SQA Point
5. Software Project
    a. ปัจจัยที่มีผลต่อผลลัพธ์
        i. Size, Deadline, Budget Cost, Application Domain, Technology, Constraint, Requirement, Resource
    b. Project ที่ดีต้อง
        i. เริ่มต้นดี ตีโจทย์แตก เลือกวิธีการเหมาสม / เตะเท้าที่ถูกต้อง, รักษาโมเมนตั้มความตั้งใจ, ติดตามงาน, ตัดสินใจดี, วิเคราะห์ข้อผิดพลาดจาก Project ที่จบไป

Software Process & Project Metric

1. คำศัพท์ : Measure = วัดแค่จุดๆเดียว, Measurement = วัดทั้งหมด, Metric = วัดเพื่อหาค่าเฉลี่ย
2. Metric มีวิธีการแบ่ง 2 แบบ
    a. แบบที่ 1
        i. Productivity Metric : วัด output เช่น function of effort & Time
        ii. Quality Metric : วัด Fitness of use เช่น วัด Reliability
    b. แบบที่ 2
        i. Process Indicators : วัด efficacy of Process เพื่อ improve
            1. วัด attribute และ improve set of attribute
            2. Process เป็นปัจจัยเดียวที่ควบคุมได้ในการ improve quality/performance
            3. 3 ปัจจัยสำคัญของ software quality/performance
                a. people(Skill/Motivation), Product(Complex), Technology
            4. วัดตรงๆไม่ได้(Indirectly) จึงวัดจาก outcome แล้วค่อยทำเป็น Metric
                a. จำนวน error ที่พบก่อน release
                b. จำนวน error ที่พบหลัง release

                } Testing Process

                c. Work product delivered
                d. Human Effort Expended
                e. Calendar Time Expanded
                f. Schedule Conformance

                } Overall Process

5. Private Metrics : Defect rates (by individual/module), Errorก่อนแจก
6. Public Metrics for the team Member : Error ตอน Review, LOC, FP
7. Public Metrics : Defect Rate, Effort, Calendar Time

ii. Project Indicators : ประเมิณ project, ค้นหาปัญหาก่อนเจอวิกฤติ, ปรับปรุงงาน

1. Project Manager เป็นผู้ใช้เพื่อปรับ Activity
2. ใช้ข้อมูลจาก Project เก่าๆ เป็น basis สำหรับ Current Project
    a. เช่นของเก่า LOC = 100 ➔ อันใหม่ก็น่าจะประมาณ 100
3. ใช้เพื่อ Monitor & Control Process

3. Measurement มีทั้งหมด 2 แบบ
    a. Direct Measure
        i. Process Metrics : Effort & Cost ที่ใช้
        ii. Product Metrics : LOC, Memory Size, Execute Speed
    b. Indirect Measure
        i. Process : วัดจาก outcome ➔ CMMI
        ii. Product : Functionality, Complexity, ... ability ความง่ายในการ...
            1. เช่น Uses ability ความง่ายในการใช้งาน = วัดชม.การเรียนรู้
4. Normalization for Metric : วัดให้มี Standard เดียวกัน
    a. Size-Oriented : Line of Code ➔ LOC
        i. เป็น Direct Measure แต่มักไม่เป็นที่ยอมรับเพราะขึ้นกับภาษา และอื่นๆ
        ii. เช่น KLOC/MM, Defects/KLOC, Cost = $/LOC, pages of document/LOC
    b. Function-Oriented : Function Point ➔ FP
        i. เป็น Indirect Measure โดย focus ที่ functional
        ii. เช่น FP, FP/MM, Cost = $/FP, Error/FP, Defects/FP, Page of Document/FP
        iii. 5 System's Basic Functions
            1. External Input : จำนวน user input 3,4,5
            2. External Output : จำนวน user output 4,5,7
            3. External Inquiries : จำนวนคำร้องขอที่ได้คำตอบ 3,4,6
            4. Internal Logical Files : จำนวนไฟล์ที่เกี่ยวข้อง 7,10,15
            5. External Interface File : จำนวนโครงสร้างไฟล์นอกระบบ 5,7,10
        iv. ปรับค่า $FP = total \times [0.65 + 0.01 \sum F_i]$ โดย F_i มีค่า 0-5
5. Metric for Software Quality : 3 viewpoint of McCall's Quality Factors
    a. Product Operation (Using it) : Correctness, Reliability, Usability, Integrity, Efficiency
        i. Correctness = defects/KLOC
        ii. Integrity = [(1-threat)*(1-security)]

b. Product Revision (Changing it) : Maintainability, Flexibility, Testability
    i. Maintainability : Indirect ➔ MTTC (Mean Time to Change) เวลาก่อนเปลี่ยน
c. Product Transition(ปรับให้ทำงานใน Env. ใหม่) : Portability, Reusability, Interoperability

6. Defect Removal Efficiency (DRE)
    a. DRE = E / (E+D) E = จำนวน error ก่อน Deliver, D = จำนวน error หลัง Deliver
7. Measure process/product ทำให้รู้ Current State ➔ ทราบ base line และทราบ estimation
    a. Software Process/Project/Product รวมได้ Data Collection ➔ ทำ Measure
    b. หลังจาก Measure แล้วคำนวณ Metric เพื่อสร้างตัววัด process/product

Software Project Planning เตรียม framework เพื่อประเมิณ resource, cost, schedule

1. ขั้นตอน
    a. Scoping : เข้าใจปัญหาว่าต้องทำอะไรให้เสร็จ
    b. Estimation : effort, time, resource เท่าไร
    c. Risk : จะทำทางไหน หลบปัญหาอย่างไร ต้องทำและแก้อย่างไร
    d. Schedule : จะใช้ resource ตอนไหน, Milestone ตรงไหน
    e. Control Strategy : จะควบคุมแผน, ควบคุมการเปลี่ยนแปลงอย่างไร
2. Observations on Estimating
    a. การ Estimate ต้องใช้ประสบการณ์ ข้อมูล historical ➔ risk ทำให้ Estimate ไม่ถูกต้อง
    b. ปัจจัยที่ทำให้ estimate ไม่ถูกต้อง : Complexity และ Size
    c. Estimate ต้องเสร็จเร็วและ update ตลอด , มีทั้ง best/bad case
3. Software Scope ทำอะไรบ้าง
    a. Functions problem statement และรายละเอียดการ estimate
    b. Performance Response Time Requirement
    c. Constraints Limit ต่างๆ เช่น h/w, s/w, memory, exist system
    d. Reliability อาจดูจาก Mean time before Failure (MTBF) = เวลาก่อนเจ๊งโดยเฉลี่ย
4. Understand Scope :
    a. Customer Need, Business Context, Project Boundary, Customer Motivation, Likely Path to Change
5. Feasibility ความเป็นไปได้ พิจารณาหลังกำหนด scope
    a. Technology, Finance, Time, Resource
6. Resource : ต้อง Estimate Resource Requirement ด้วย
    a. Development Environments : Hardware/Software Tool
    b. Reusable Software Components
    c. People
    d. แต่ละอันมี 4 ลักษณะ : Description, State of Available, Time when require, Duration

7. Human Resource : แบ่งงาให้เหมาะสม, จะทราบจำนวนคนได้หลัง Estimation of Development
    a. ถ้า project size น้อยกว่า 6 person-month ➔ 1 คนทำทุกอย่าง (มีที่ปรึกษาได้)
8. Reusable Software Resource
    a. Off-the-shelf(ใช้ได้เลย)/ full-experienced/ partial-experience/ new Component
9. Software Project Estimate
    a. Cost estimation error เป็นได้ทั้ง กำไร และ ขาดทุน
    b. Option ในการประมาณ Cost and Effort
        i. Delay Estimation until late
        ii. Base Estimation on similar Project
        iii. Use Decomposition Techniques (Conventional Method)
            1. Problem-Base LOC/FP Base Estimation
            2. Process-Based มองที่งานจริง มี cost มากกว่า เช่น ฝึกสอน สังสรรค์
            3. Example ดูในชีท **
        iv. Empirical Model (สูตรทางสถิติมาคำนวณ)
            1. LOC-Oriented Estimation Model
                a. $E = 5.2 \times (KLOC)^{0.91}$ Walston-Felix Model
                b. $E = 5.5 + 0.73 \times (KLOC)^{1.16}$ Bailey-Basili Model
                c. $E = 3.2 \times (KLOC)^{1.05}$ Boehm Simple Model
                d. $E = 5.288 \times (KLOC)^{1.047}$ Doty Mod for KLOC>9
            2. FP-Oriented Estimation Models
                a. $E = -13.9 + 0.0545\ FP$ Albrecht and Gaffney Model
                b. $E = 60.62 \times 7.728 \times 10^{-8}\ FP^3$ Kemerer Model
                c. $E = 5.875 + 15.12\ FP$ Matson, Barnett, and Mellichamp
10. COCOMO Model (COnstruct COst MOdel)
    a. มี 3 form
        i. Basic Cocomo : Estimate แบบหยาบๆ แทนค่าออก
            1. $E = a(LOC)^b$: Effort person-month
            2. $D = c*E^d$: Development Time (Months)
        ii. Intermediate Cocomo : พิจารณา 15 ปัจจัยที่เรียกว่า Cost Driven Attribute
            1. $E = a(LOC)^b$ * EAF : EAF คือ Effort Adjustment Factor 15 ตัว
        iii. The Advanced Cocomo : อาจารย์ไม่พูดถึง น่าจะไม่ออกสอบ
11. Class of Software Project
    a. Organic Mode (App ทั่วๆไป)
        1. ไม่มี communication overhead

2. Development Team มีขนาดเล็ก ประสบการณ์กว้างและคุ้นเคยกัน
3. คุ้นเคยกับลูกค้า
4. In-House Environment

b. Semi-Detached Mode (Utility)
   i. ทีมมีประสบการณ์กว้าง lv.กลางๆกับงานที่ทำ (มีคนที่มีและไม่มีประสบการณ์)

c. Embedded Mode (system)
   i. Tight Constraint ทุกอย่างต้องเป๊ะ
   ii. Strong Coupled Complex of hw, sw, regulation, operational procedures
   iii. ต่อรองเปลี่ยนแปลงได้ยาก ใช้ความพยายามอย่างมากในการทำให้ตรง spec

12. COCOMO II :
    a. Application Composition : ใช้ระหว่าง stage ต้นๆ
    b. Early Design Stage Model : ใช้เมื่อ Requirement พร้อมไม่เปลี่ยนแล้ว
    c. Post-Architecture-Stage Model : ใช้ระหว่าง construct Software

SQA : Software Quality Assurance

1. SQA ประกอบด้วย
   a. Quality Management, Effective Methodology, formal review, multitier testing, doc ...
2. Quality concepts
   a. Hardware : Control Differences between variation
   b. Software :
      i. ลดความแตกต่างระหว่างที่ทำนายกับ Resource จริง
      ii. test ครอบคลุมทุกปัญหา
      iii. ลดจำนวน bug
   c. 2 kind of Qualities base on measurable characteristics
      i. Quality of Design :
         1. ตรงกับ Design Spec
         2. Grade ของ Material, Tolerance, performance
      ii. Quality of Conformance : Degree ของสิ่งที่ Design สามารถผลิตได
   d. Quality Control คือ Series of inspections (ตรวจอย่างละเอียด), reviews, and tests
   e. Quality Control รวมถึงการ Feedback ซ้ำๆไปที่ process เพื่อสร้าง work product
   f. Quality Assurance ➔ Analysis, Auditing, Reporting Activity
   g. Cost of Quality : (รวมในค่าพัฒนาด้วย)
      i. Prevention Costs : ช่วง Planning, Training
      ii. Appraisal Costs : ช่วงที่ประเมิณ ใช้ Tool

iii. Failure Costs : ค่า Maintenance ระบบหลังเสร็จ

1. Internal : Rework, repair
2. External : Complaint Sol, Product Return and Replacement ...

3. SQA มี 5 Phases
   a. Process Definition & Standards
   b. Formal Technical Reviews
   c. Test Planning & Review
   d. Measurement
   e. Analysis & Reporting
4. SQA Activity แบ่งออกเป็น 2 Group คือ
   a. Software Engineers : ดูด้าน Technical, Review, Perform
   b. SQA Group : ช่วย SE ให้ได้ High Quality Product (เป็น Third-Party มา test โปรแกรม)
5. SQA Activity
   a. Prepare a SQA Plan for a Project
   b. Participate in Development of the Project Software Process มาดูงานตั้งแต่ต้น
   c. Review activity to verify
   d. Audit Software work Product
   e. Ensure that Deviations (ออกนอกลู่นอกทาง) มี Document ในการ Handle
   f. Record any noncompliance แล้วรายงาน Senior Manager โดยไม่ต้องผ่านคนในทีมได้เลย
6. Software Review (อ.มองว่าสำคัญที่สุดใน SQA)
   a. เกิดแต่ละจุดขณะ develop (=milestone)เพื่อกำจัด uncovered error/บอกให้ SE แก้ที่ไหนบ้าง
   b. ให้ Developerจูงมือคอยเดินอธิบาย ถ้า SQA สงสัยก็ถามได้ แต่ไม่ใช่ให้ SQA เดินเองอ่านเอง
   c. Effectiveness Scale (บน ทางการ ทำยาก , ล่าง ไม่ทางการ ทำง่าย)
      i. Inspection (FTR)
      ii. Walkthrough (FTR)
      iii. Formal Presentation
      iv. Informal Presentation
      v. Peer Group Review
      vi. Casual Conversation
7. Formal Technical Review (FTR) มีประสิทธิภาพที่สุด
   a. Objective : หา error ต่างๆ, ตรงกับ requirement, uniform manner, more manageable
   b. Review Meeting
      i. Constraints
         1. 3-5 คน ไม่ควรเยอะ, แต่ละคนเตรียมไม่เกิน 2 ชั่วโมง, คุยกันน้อยกว่า 2 ชั่วโมง

ii. Review Team : Review Leader, Producer, Reviewer, Recorder

8. Review's Preparation
    a. **เข้าใจ context, skim อ่านคร่าวๆ, อ่านแล้วเขียนสิ่งที่สงสัย, ไม่ถามเรื่องสำนวน**
9. Conducting The Review **สิ่งที่ต้องทำเมื่อเป็นประธานในที่ประชุม**
    a. **เตรียม ประเมิณ** product **ก่อนประชุม**
    b. Review not producer
    c. Keep your tone mild, ask question no make accusations
    d. Stick to the agenda
    e. Raise issue, don't resolve them
    f. Avoid discussion of style
    g. Schedule review
    h. Record and report all review reuse. (Summary andser 3 question)
        i. What was review, Who reviewed it, What were the conclusion
10. Review Guideline
    a. Review Product, not the producer
    b. **สร้างกฎและรักษา**
    c. Limit **การโต้แย้ง**
    d. Don't attempt to solve every problem note
    e. **จดลงโน้ตด้วยเสมอ**
    f. Limit **จำนวนของผู้เข้าร่วม** review
    g. **สร้าง** checklist
    h. **จอง** Resource **และ** Time Schedule **สำหรับ** FTR
    i. Review your early review
11. Metrics Derived for Reviews
    a. Inspection Time per page of documentation
    b. Inspection Time per KLOC or FP
    c. Inspection effort per KLOC or FP
    d. Error uncover per reviewer hour
    e. Error uncover per preparation hour
    f. Error uncover per software engineering task
    g. Number of minor errors
    h. Number of major errors
    i. Number of errors found during preparation
12. The SQA Plan [IEEE std 730-1984]

a. Initial section — Overall ของ Project
b. Reference section — ร่างเอกสารของ Project
c. Management section — ขอบเขต SQA
d. Document section — อธิบาย Work Product
e. Standards, Practices, Conventions section — บอกทุก standard
f. Review and Audit section — บอกว่า review อะไร
g. Test section — อ้างจาก Test Plan วิธีการ บันทึกผล
h. Problem report and correct section — รายงานปัญหา ใครรับผิดชอบส่วนใด

13. ISO 9000 Quality Standard มี 20 Requirement, for all engineering discipline

Requirement Elicitation

1. Requirement = Feature that must have or satisfy to be accept by the client
    a. ส่วนช่องว่างระหว่าง Developer กับ Client ใช้ Scenario และ use-case ช่วย
2. Requirement Engineering มี 2 activities
    a. **Requirement Elicitation** > requirement specification
        i. **Functional**
        ii. **Non-functional** (ability to...)
    b. **Analysis** > an analysis model
        i. Static/analysis object model เช่น class diagram
        ii. Dynamic เช่น activity, sequence diagram

โดย System specification เขียนเป็น natural language (THAI/ENG) เป็นสื่อกลางระหว่าง Developer กับ Client ส่วน analysis model ใช้ Formal & Informal เช่น UML ช่วยลดความกำกวม & ใช้ตอน design

3. **Requirement Elicitation** เน้น
    a. System functionality หน้าที่หลักของระบบที่ควรทำได้
    b. Interaction between user and system
    c. Error handling
    d. Environment condition
4. **Functional Requirement**
    a. อธิบายการทำงานของแต่ละ function ว่ามี interact ระหว่าง system กับ environment (user or external system)
    b. เน้นเฉพาะ interact ที่เป็นไปได้
5. **Nonfunctional Requirement** [RUPS - LIPSI]
    a. **U**sability > user ใช้งานได้ง่าย

b. **R**eliability > MTBF (mean tune between failure)

c. **P**erformance > response time = ตอบสนองเร็วมั้ย

> throughput = ระบบสามารถทำงานมากแค่ไหนในเวลาที่กำหนด

> availability = เวลาที่ระบบพร้อมให้ใช้งาน

d. **S**upportability (หลัง deployment แล้ว)

i. Adaptability > การเพิ่ม function ใหม่

ii. Maintainability > การแก้ไข defects

iii. Portability > การเปลี่ยน HW/SW

e. **I**mplementation Requirement การบังคับให้ใช้ เช่น ภาษา java, window

f. **I**nterface requirement การบังคับ interface เช่น ประสานข้อมูลที่รับมา

g. **O**perational requirement หรือ space requirement ใครนั่งตรงไหน

h. **P**ackaging requirement การส่งมอบระบบ เช่น CD1 Manual

i. **L**egal Requirement ~ licensing, certification

6. **Validation** หลังเขียน requirement specification ต้องตรวจสอบดังนี้

a. Correct ตรงความต้องการของลูกค้า

b. Complete เก็บทุก scenarios

c. Consistent ทุกเอกสารไม่จัด....เอง ไม่มีข้อมูลขัดแย้งกัน

d. Unambiguous ทุกคนอ่าน แปลตรงกัน

e. Realistic Requirementนี้ต้องสามารถ implement ได้

7. **Verifiability & Traceability**

a. **Verification** = work products properly reflects requirement specification

b. **Validation** = แสดงให้เห็นว่า product จะ fulfill และใช้งาน

c. **Requirement specification จะ verifiable** เมื่อสามารถ design/implement ได้

d. ตัวอย่างของ non-verifiable requirement

i. Product shall have <u>good</u> UI > good ยังไง ทุกคนมองต่างกัน

ii. Product shall be <u>error free</u> > .....

e. **Non-specification จะ traceability** เมื่อมี last case มาตรวจแล้วผ่าน

8. **วิธีการสร้าง product ให้ตรง specification requirement**

a. **Greenfield Engineering** พัฒนาอะไรที่ยังไม่มีบนโลก เช่น iPhone

b. **Reengineering** พัฒนา/ปรับปรุง จาก exist system

c. **Interface Engineering** ปรับ UI ให้ใช้งานง่าย แต่ core เดิม (= Lagency system)

9. **Actor**

a. External entity ที่ interact กับ system เช่น คน, ธนาคาร, ระบบ

b. 1 คนอาจะเป็นหลาย Actor ที่ role ต่างกันได้

10. **Scenarios**
    a. คือ Narrative description ของเหตุการณ์ที่คนพยายามใช้ system หรือ application
    b. Scenario : ค่อนข้าง concrete & informal
    c. Scenarios มีหลายอย่างได้ แต่ ใช้การดำเนินการเหมือนกัน เช่น
       ซื้อของที่เซเว่น กับซื้อที่ Tops Supermarket
    d. Scenarios มี 4 ชนิด
        i. -is = อธิบาย current situation
        ii. Visionary = อธิบาย Future situation
        iii. Evaluation = user.....ช่วยคิดว่าระบบควรมี scenarios.....
        iv. Training = ใช้ในการสอน new user
    e. Scenario = Instance ของ Use case
11. **Use Case**
    a. Specifies all possible scenarios
    b. Use case ถูก initial by actor
    c. บอก complete flow of events > ทุกเหตุการณ์ที่เป็นไปได้
       เช่น การซื้อของที่เซเว่นด้วย เงินสด, credit, smart purse, debit
    d. Use case เน้น completeness & correctness
    e. Use case ต้องอธิบาย case แปลกๆ ด้วย (exception handling)
    f. EXTEND = แยก exceptional จาก normal event
    g. INCLUDE = ลด redundancy use case
12. **Analysis Objects** หลังเขียน use case แล้ว ดูว่า มี class ไรมั่ง?
    a. เอาไปใช้ในการสร้าง analysis model
    b. คำที่ใช้ควรให้ developer & user เข้าใจง่าย
    c. มักใช้ noun จาก use case มาเป็น analysis object
13. **Documenting requirement elicitation**
    Requirement analysis document (RAD) = ผลลัพธ์การทำ Elicitation ซึ่ง RAD เป็นเอกสาร
    a. functional and nonfunctional requirement
    b. ใช้สื่อสารกับ customer และ developer
    c. ควรเขียนเสร็จหลังจาก use case stable หรือเข้า dev. Phase
    d. ผู้ฟัง RAD = client, user, project

Chapter 5 : Analysis

ไม่รู้เรื่อง + Sheet พี่ T ไม่ได้เขียนสรุป ไปอ่านชีทแทนดีกว่า แหะๆ