

แนวทางการพัฒนาซอฟต์แวร์คุณภาพ

การฝึกอบรม “การประเมินราคางานพัฒนาซอฟต์แวร์ ประเภทโปรแกรม
ประยุกต์ (Application Software) ดิจิตอลคอนเทนต์ (Digital Content) และ
สื่อสร้างสรรค์ (Animation) ของประเทศไทย”

วันที่ 28 มีนาคม 2556

หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพและกระบวนการ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
- กระบวนการบริหารความเสี่ยง
- กระบวนการบริหารการเปลี่ยนแปลง
- การประกันคุณภาพซอฟต์แวร์

หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพและกระบวนการ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
- กระบวนการบริหารความเสี่ยง
- กระบวนการบริหารการเปลี่ยนแปลง
- การประกันคุณภาพซอฟต์แวร์

- ส่งงานช้า
- ไม่เคยเอาโปรแกรมมาให้ดูเลย
- ซอฟต์แวร์ใช้งานไม่ได้
- ซอฟต์แวร์ใช้งานยาก

- Bug เยอะมาก กดอะไรก็ error
- ทำเอกสารอะไรมาอ่านไม่รู้เรื่อง
- อื่นๆ อีกมากมาย

ซอฟต์แวร์ไม่
มีคุณภาพ !!!



- ลูกค้าไม่ confirm requirement
- โปรแกรมเมอร์ไม่อ่าน requirement
- ลูกค้าเปลี่ยน requirement
- มีเอกสารต้องทำเยอะแยะ แต่ไม่มีใครอ่าน
- ฯลฯ



- ผมไม่ได้นอนมา 3 วันแล้วครับ
- ลูกค้าเปลี่ยน requirement อีกแล้ว
- พรุ่งนี้จะส่งงานแล้ว ยังแก้ bug ไม่เสร็จเลย
- ต้องทำเอกสารด้วยเหรอ แค่เขียนโปรแกรมก็ไม่ทันแล้ว
- อื่น ๆ อีกมากมาย



Q: ลูกค้า/ผู้ใช้งาน/ผู้ว่าจ้าง และผู้ผลิตซอฟต์แวร์
อยากได้อะไร?

A: ซอฟต์แวร์ที่มีคุณภาพ

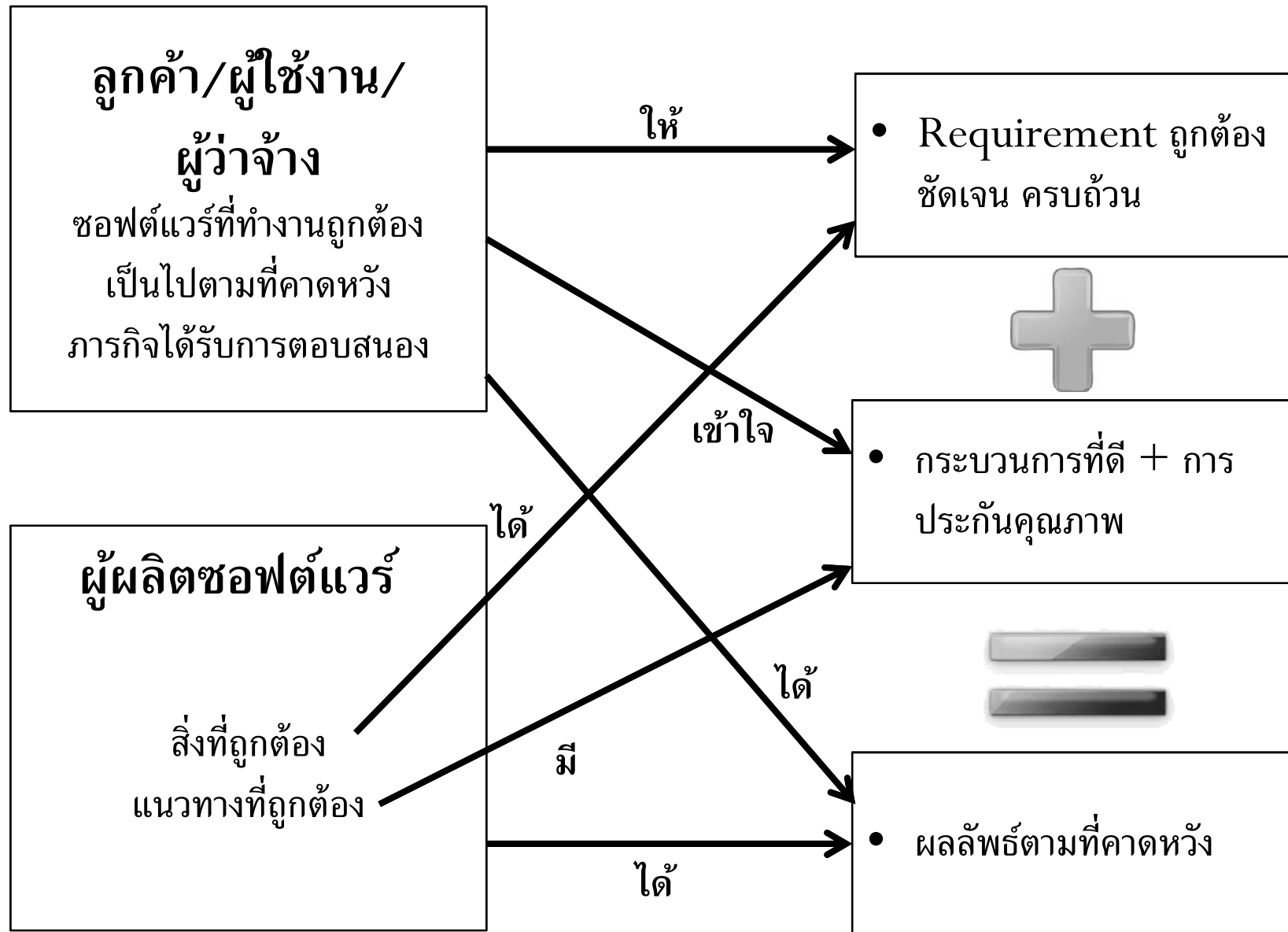
- ตรงตามความต้องการ (conformance to requirements
— นิยามโดย Philip Crosby)
- ใช้งานได้ดี (fit for use — นิยามโดย Dr. Joseph Juran
และ Dr. Edwards Deming)

คุณภาพในมุมมองของลูกค้า/ผู้ใช้งาน/ผู้ว่าจ้าง

- ได้รับผลิตภัณฑ์/บริการที่ถูกต้องสำหรับใช้งาน
- ผลิตภัณฑ์หรือบริการเป็นไปตามที่คาดหวัง
- มีความพึงพอใจที่ความต้องการของลูกค้าหรือผู้ใช้ได้รับการตอบสนอง
- ได้รับการปฏิบัติด้วยความเคารพ (respect) ให้เกียรติ สุภาพ (courtesy) และซื่อตรง (integrity)

คุณภาพในมุมมองของผู้ผลิตซอฟต์แวร์

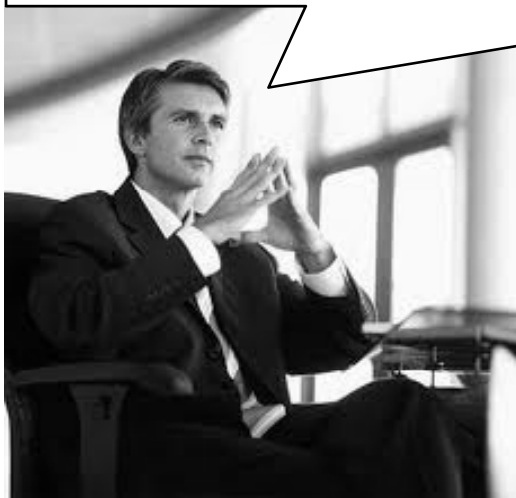
- ทำในสิ่งที่ถูกต้อง (Doing the right thing)
- ทำในแนวทางที่ถูกต้อง (Doing the right way)
- ทำให้ถูกต้องแต่แรก (Doing it right the first time)
- ทำให้ทันเวลาโดยไม่มีค่าใช้จ่ายส่วนเกิน (Doing it on time without exceeding cost)



กระบวนการคืออะไร

- กระบวนการ (Process) คือเครื่องมือที่ใช้ในการสื่อสารวิธีการในการผลิตสินค้าหรือบริการ
- กระบวนการ ประกอบด้วยกิจกรรมต่าง ๆ และวิธีการทำงานในแต่ละกิจกรรม

- ช่วยอธิบายวิธีการทำงานให้แก่ผู้ปฏิบัติงาน
- ใช้เป็นเครื่องมือในการถ่ายทอดความรู้ให้กับผู้ปฏิบัติงานที่เข้ามาใหม่
- ทำให้สามารถคาดการณ์คุณภาพของผลผลิตได้
- ช่วยในเรื่องของการปรับปรุงกระบวนการอย่างต่อเนื่อง
- ส่งเสริมให้ผู้ปฏิบัติงานมีส่วนร่วมในการปรับปรุงกระบวนการและคุณภาพเพื่อความพึงพอใจของลูกค้าด้วยการให้ผู้ปฏิบัติงานกำหนดและปรับปรุงกระบวนการทำงานที่ตนเองรับผิดชอบ
- ช่วยให้ฝ่ายบริหารสามารถทุ่มเทให้กับงานวางแผนและการติดต่อพบปะกับลูกค้าได้มากขึ้น



**ประโยชน์ของกระบวนการ
ในมุมมองของผู้บริหาร**

- ช่วยให้ผูปฏิบัติมั่นใจได้ว่าผลผลิตที่ตนเองผลิตออกนั้นจะได้รับการยอมรับจากลูกค้า
- ช่วยให้ผูปฏิบัติงานทำงานอย่างมีทิศทางที่ชัดเจน เพราะทราบว่าผลงานของตนเองจะถูกประเมินอย่างไร
- ส่งเสริมให้ผูปฏิบัติงานทุ่มเทและใช้ความคิดสร้างสรรค์ในการผลิตผลงาน โดยที่ไม่ต้องกังวลกับวิธีการทำงาน
- ช่วยให้ผูปฏิบัติงานสามารถวางแผนการทำงานประจำวันของตนเองได้ดียิ่งขึ้น

**ประโยชน์ของกระบวนการ
ในมุมมองของผูปฏิบัติงาน**



มีกระบวนการแล้วอย่างไร?

ในมุมมองของผู้ว่าจ้าง

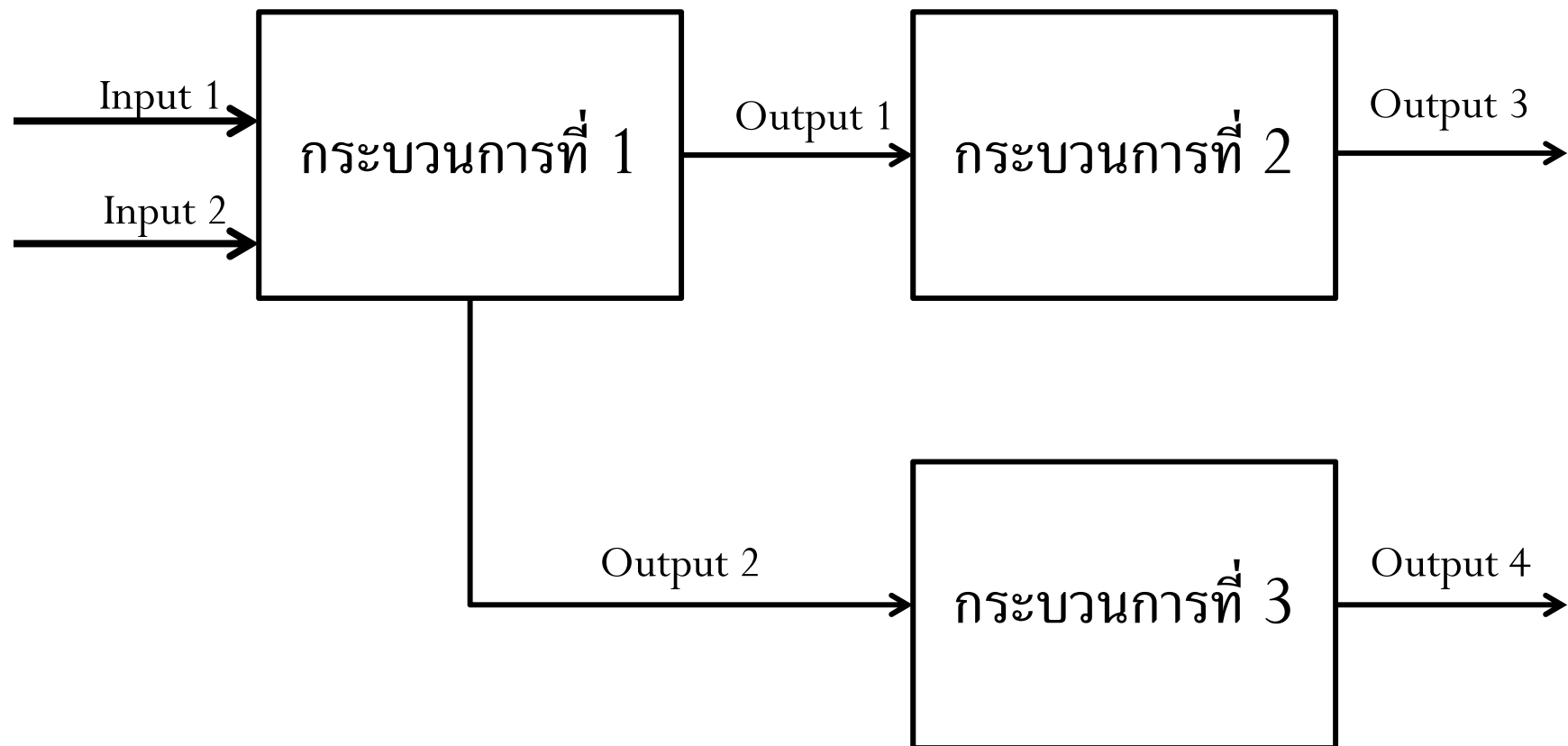
- ได้ทีมงานที่มีมาตรฐานในการดำเนินโครงการ เป็นผู้รับงาน
- สามารถติดตามความคืบหน้าของโครงการ และตรวจสอบความถูกต้องเหมาะสมของกระบวนการได้อย่างมีหลักการ เป็นขั้นเป็นตอน
- เมื่อจบโครงการแล้ว มีเอกสารอ้างอิงเพียงพอเพื่อดูแลรักษาระบบรวมถึงสามารถพัฒนาต่อยอดได้
- ทีมงานมีความรู้ความเข้าใจในกระบวนการพัฒนาระบบไปในทิศทางเดียวกัน เข้าใจการทำงานในแต่ละขั้นตอน ทำให้เกิดประสิทธิภาพในการดำเนินการ รวมถึงการทดแทนกันได้ในบางหน้าที่ ทำให้ลดความเสี่ยงในการส่งมอบงานล่าช้ากว่ากำหนด

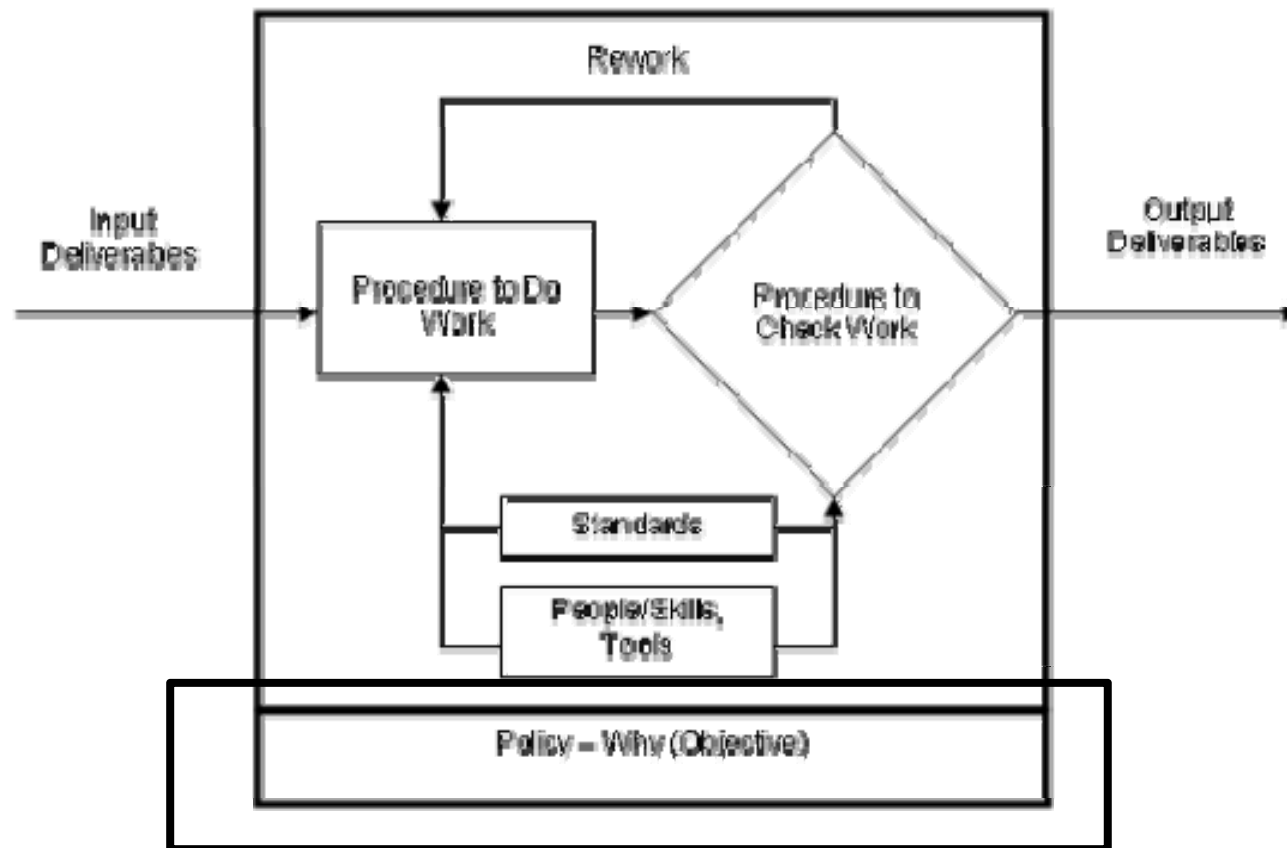
มีกระบวนการแล้วได้อย่างไร?

ในมุมมองของผู้รับจ้าง

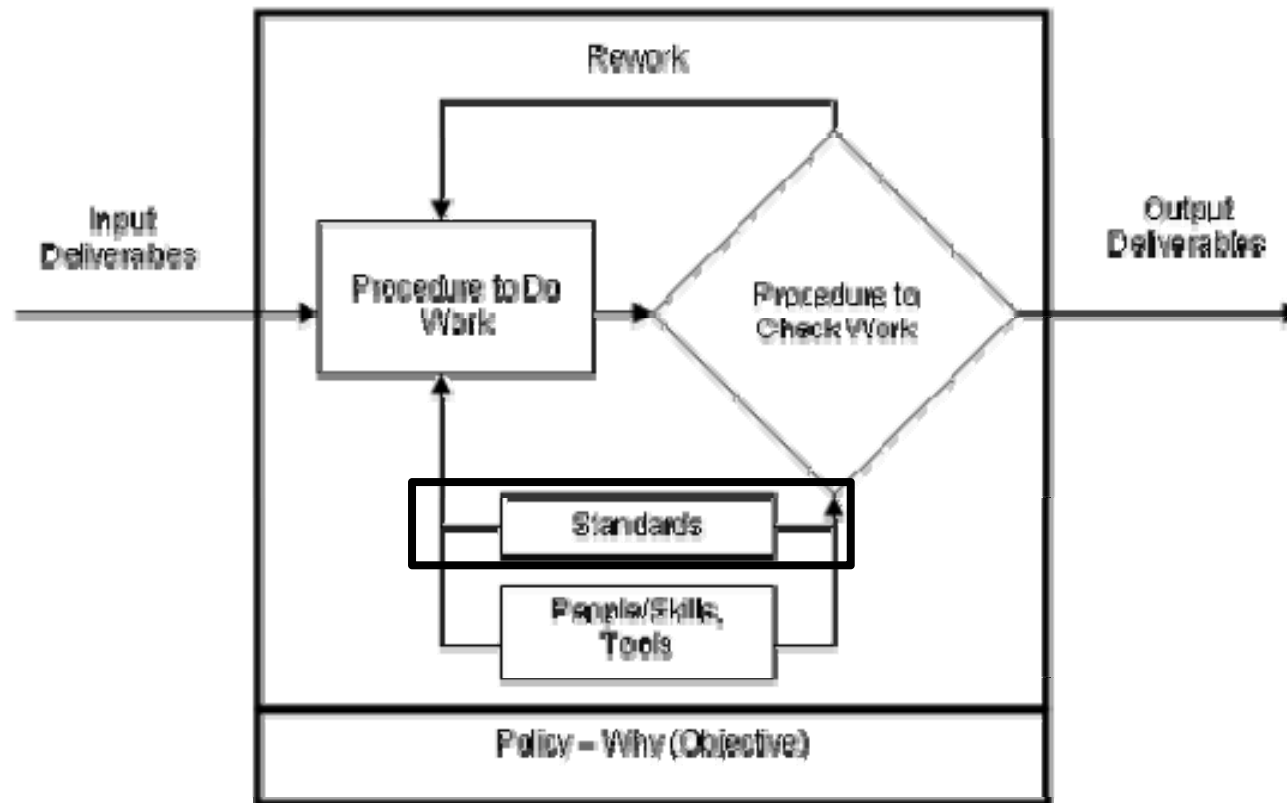
- สามารถดำเนินโครงการได้บรรลุเป้าหมาย
- มีกระบวนการในการพัฒนาระบบงานที่เป็นแนวทางเดียวกันทั้งองค์กร
- มีรูปแบบของการสรุปความต้องการที่ชัดเจน เพื่อลดข้อขัดแย้งและประเด็นปัญหา
- ทุกคนในองค์กร เข้าใจบทบาทและหน้าที่ความรับผิดชอบของตนเอง
- สามารถวิเคราะห์และประเมินระยะเวลาดำเนินโครงการได้แม่นยำขึ้น และลดความเสี่ยงที่อาจจะเกิดขึ้นกับโครงการได้
- สร้างความเชื่อมั่น ทั้งกับลูกค้าและทีมงานในองค์กร

องค์ประกอบของกระบวนการ

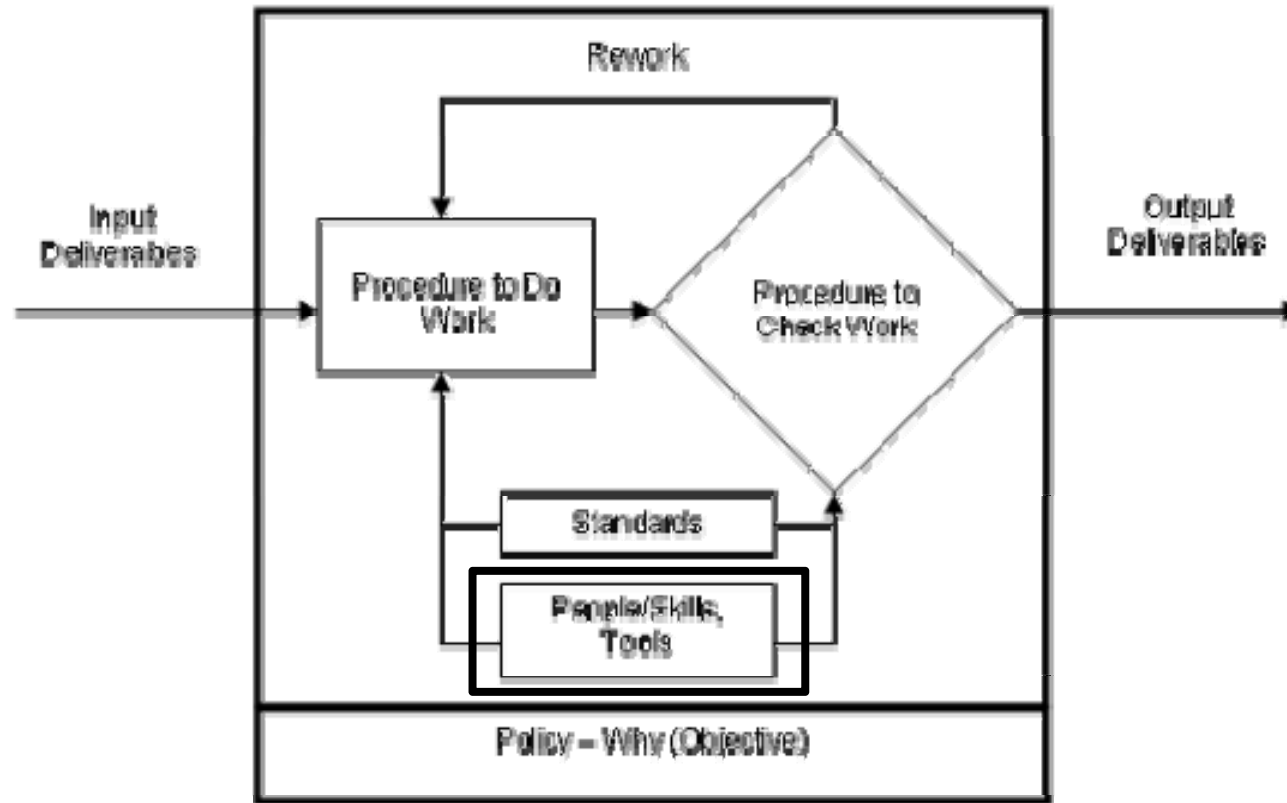




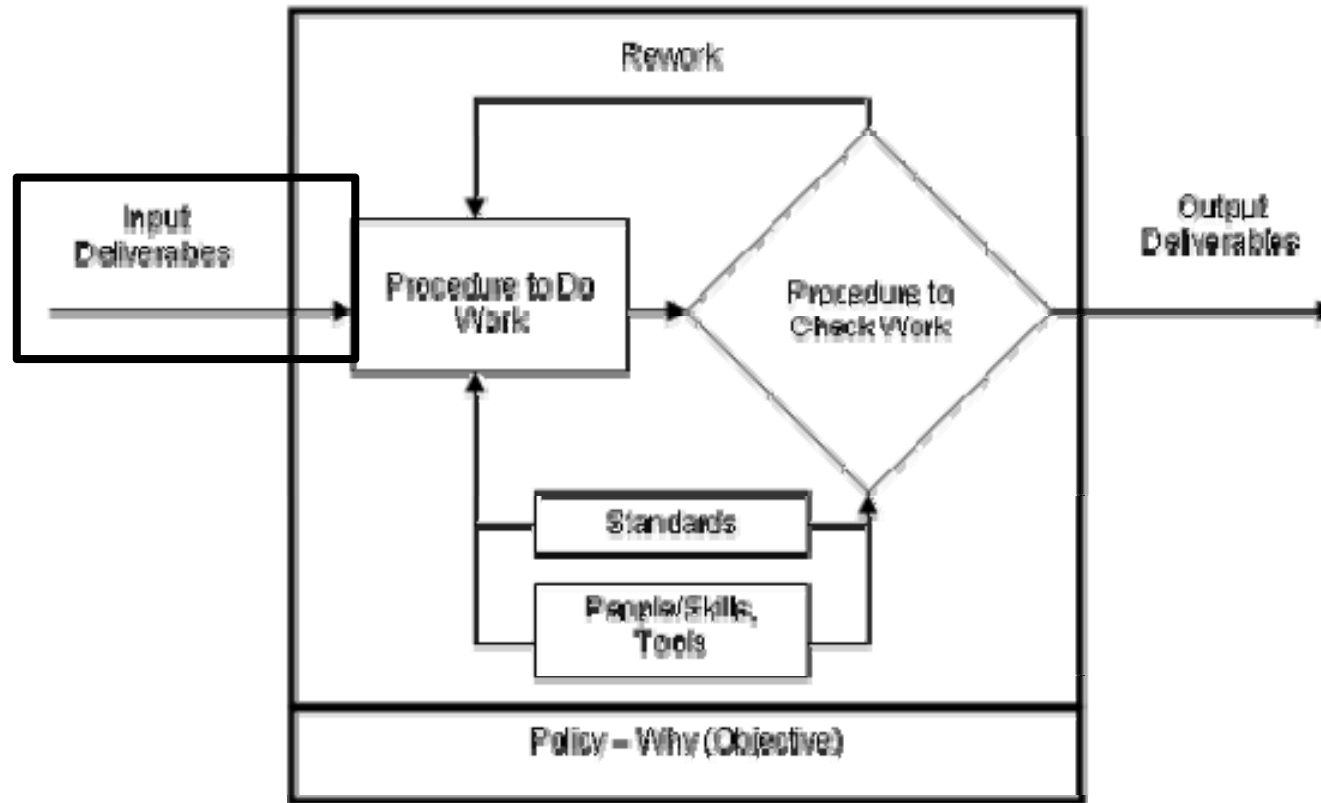
นโยบาย (Policy) – อธิบายวัตถุประสงค์ของกระบวนการ โดยมี
การเชื่อมโยงกับกลยุทธ์ขององค์กรและความต้องการในการตอบสนอง
ต่อความต้องการของลูกค้า



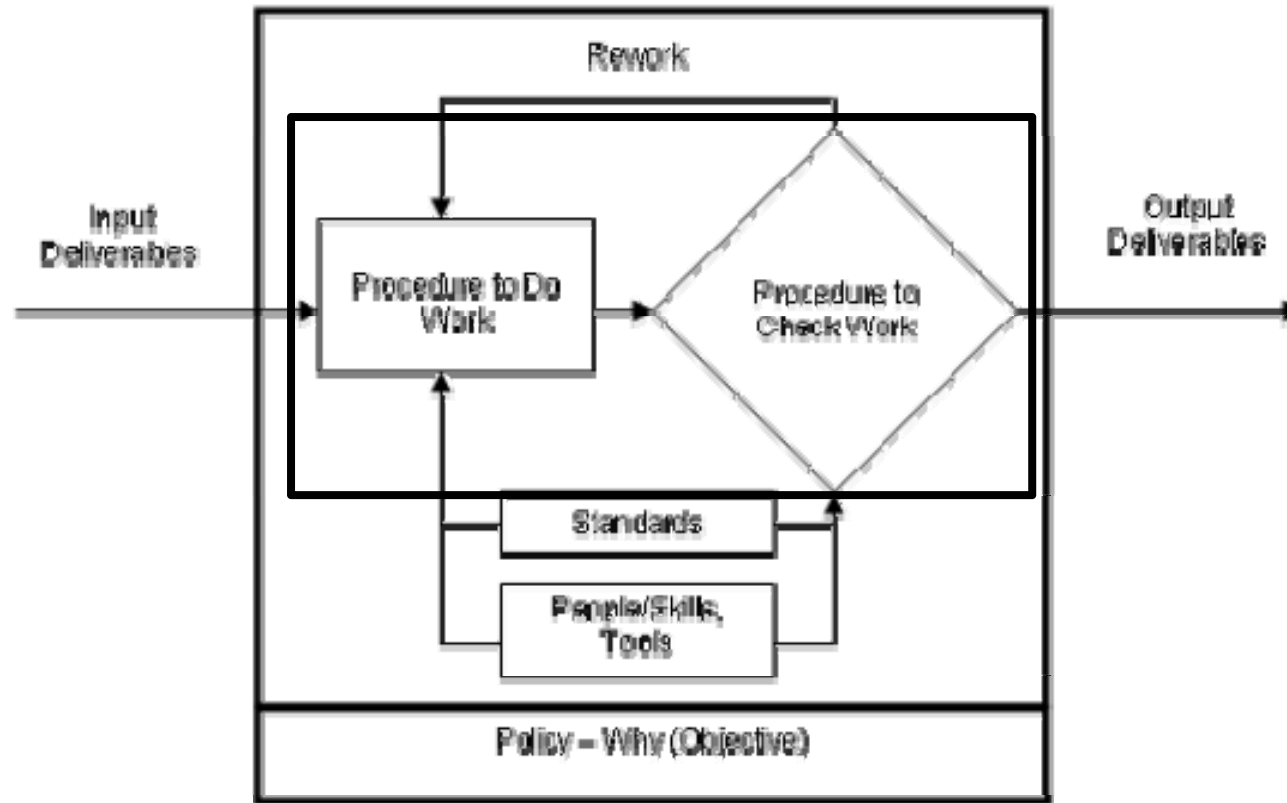
มาตรฐาน (Standard) — ระบุมาตรฐานของผลผลิตที่ได้จากกระบวนการ และตัวกระบวนการเองที่ต้องมีและทำได้ เพื่อให้บรรลุตามวัตถุประสงค์ของกระบวนการ มาตรฐานที่กำหนดขึ้น จะเป็นสิ่งบ่งชี้ว่าผลิตภัณฑ์ที่ได้ตรงตามความต้องการของลูกค้าหรือไม่ การกำหนดมาตรฐานต้องสามารถวัดได้ ส่งมอบได้ และเป็นสิ่งจำเป็นที่ต้องมี



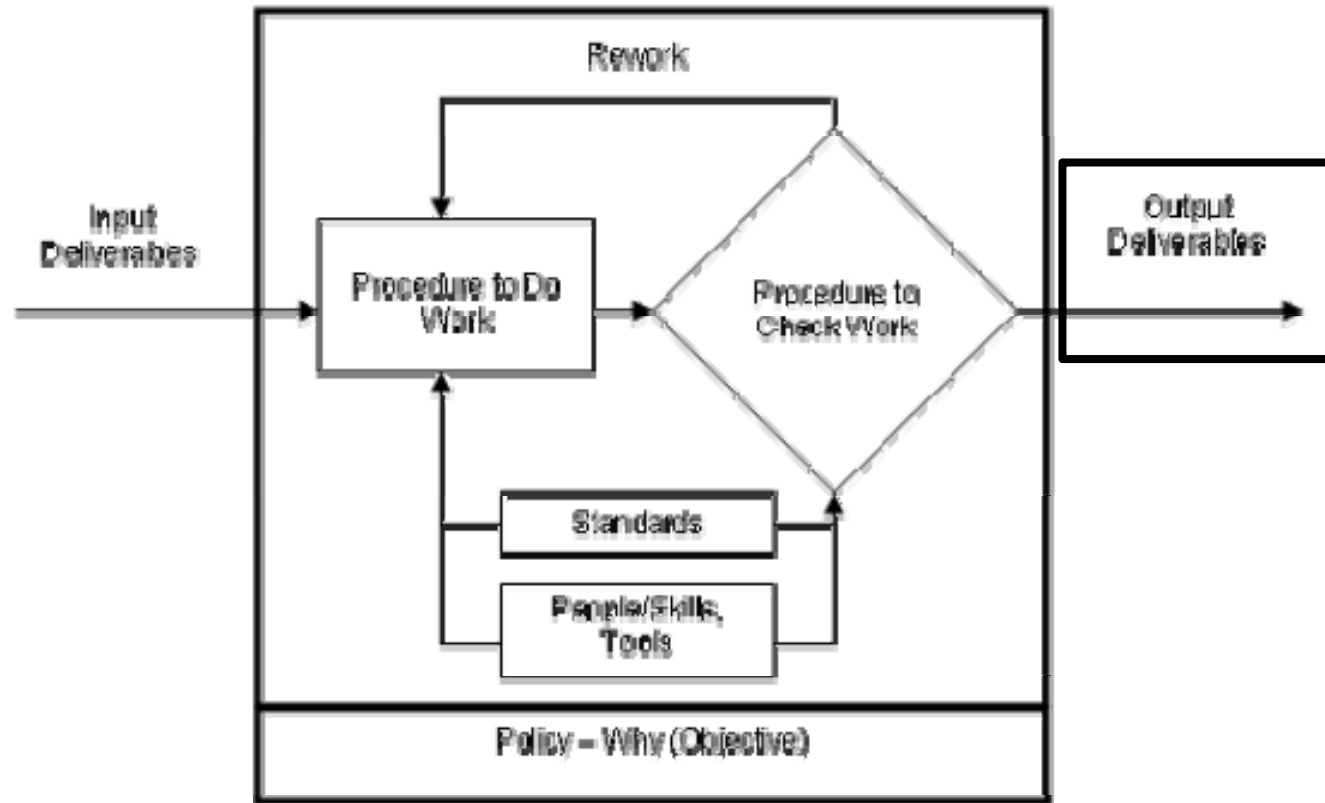
คน/เครื่องมือ (people/skills, tools) – อธิบายหน้าที่ความรับผิดชอบ ทักษะและความรู้ที่ควรต้องมีการทำงานตามกระบวนการ เช่น โปรแกรมเมอร์ต้องมีทักษะหรือความรู้ด้านโปรแกรมภาษา C++ รวมทั้ง เครื่องมือต่างๆ ที่ช่วยในการทำงานเช่น CASE Tool, เอกสารต้นแบบ ต่างๆ (template), test tool



สิ่งป้อนเข้า (Input) – หมายถึงเกณฑ์ในการเริ่มกระบวนการ (entry criteria) หรือ สิ่งที่ต้องป้อนเข้าเพื่อนำไปใช้ในกระบวนการก็ได้



วิธีการ (Procedure) – อธิบายวิธีการทำงาน เช่น จะใช้คน เครื่องมือ และเทคนิคในการทำงานอย่างไรให้ได้ตามมาตรฐาน หรือจะแปลง input เป็น output อย่างไร รวมทั้งการตรวจสอบวิธีการทำงานและผลผลิตที่ได้ หากไม่เป็นไปตามมาตรฐาน จะถูกนำไปแก้ไขและตรวจสอบจนกว่าจะได้ตาม มาตรฐานที่กำหนด



ผลลัพธ์ (Output/deliverable) – หมายถึงเกณฑ์ในการออกจากกระบวนการ (exit criteria) หรือผลลัพธ์ที่ได้จากการทำงานตามกระบวนการ ผลลัพธ์อาจเป็นสิ่งที่ใช้ภายในกระบวนการเอง (interim deliverable) ส่งต่อไปยังกระบวนการอื่นหรือส่งมอบให้กับลูกค้า (external deliverable)

ตัวอย่าง กิจกรรม : วิเคราะห์และจัดทำข้อกำหนดความต้องการระบบ

Policy	• เพื่อกำหนดความต้องการของลูกค้าให้ได้ตั้งแต่ช่วงต้นของโครงการ และเพื่อให้แน่ใจว่าทุกฝ่ายที่เกี่ยวข้องตีความความต้องการได้สอดคล้องและเป็นไปในแนวทางเดียวกัน • ข้อกำหนดความต้องการต้องสะท้อนความต้องการที่แท้จริงขององค์กร มีความสมบูรณ์ ถูกต้อง ทดสอบได้ และบำรุงรักษาง่าย เพื่อให้ข้อกำหนดความต้องการนี้สามารถใช้งานได้ไปตลอดจนจบโครงการ
Standard	• ความต้องการที่ได้จากการประชุมต้องได้รับการทบทวนและอนุมัติโดยผู้ที่เข้าร่วมประชุมภายในวันถัดไป • ข้อมูลหรือสารสนเทศที่มีการอ้างอิงถึงในข้อกำหนดความต้องการจะต้องถูกอธิบายไว้ในพจนานุกรมข้อมูล (Data dictionary)
People/ Skill/Tool	• นักวิเคราะห์ระบบ : มีความเข้าใจเกี่ยวกับ UML สามารถจัดทำเอกสาร SRS ได้โดยใช้ UML diagrams • เครื่องมือ : requirement management tool xxx , SRS template, SRS checklist

ตัวอย่าง กิจกรรม : วิเคราะห์และจัดทำข้อกำหนดความต้องการระบบ

Procedure	- นักวิเคราะห์ระบบ : ประชุมร่วมกับผู้ให้ requirement - นักวิเคราะห์ระบบ : ศึกษาคู่มือการปฏิบัติงานของลูกค้า - นักวิเคราะห์ระบบ : สรุปรายงานการประชุม - นักวิเคราะห์ระบบ : ใช้เครื่องมือ xxx ในการบันทึกความต้องการ - นักวิเคราะห์ระบบ : ใช้ template SRS สร้างรายการความต้องการ - นักวิเคราะห์ระบบ : จัดทำ requirement traceability matrix - นักวิเคราะห์ระบบ : ตรวจสอบ SRS ที่จัดทำขึ้นตาม SRS checklist - หัวหน้าทีมวิเคราะห์ระบบ : ทบทวนรายการความต้องการแต่ละรายการและชี้ประเด็นต่างๆที่ยังไม่ชัดเจน
Input	- ดำเนินการ Kick-off meeting กับลูกค้าแล้ว - รายชื่อผู้ที่เข้าร่วมประชุมในการกำหนด requirement - ศึกษาคู่มือการปฏิบัติงานของลูกค้า
Output	- เอกสารข้อกำหนดความต้องการ - Requirement traceability matrix

หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพ
- **มาตรฐานกระบวนการพัฒนาซอฟต์แวร์**
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
- กระบวนการบริหารความเสี่ยง
- กระบวนการบริหารการเปลี่ยนแปลง
- การประกันคุณภาพซอฟต์แวร์

การบริหารโครงการ (Project Management)

Project
Planning

Project Plan
Execution

Project
Assessment and
Control

Project
Closure

Risk Management

Change Management

**Software Quality
Assurance**

Software
Implementation
Initiation

Software Requirement
Analysis

Software
Architectural and
detailed design

Software Construction

Software Integration
and Tests

Product Delivery

การสร้างซอฟต์แวร์ (Software Implementation)

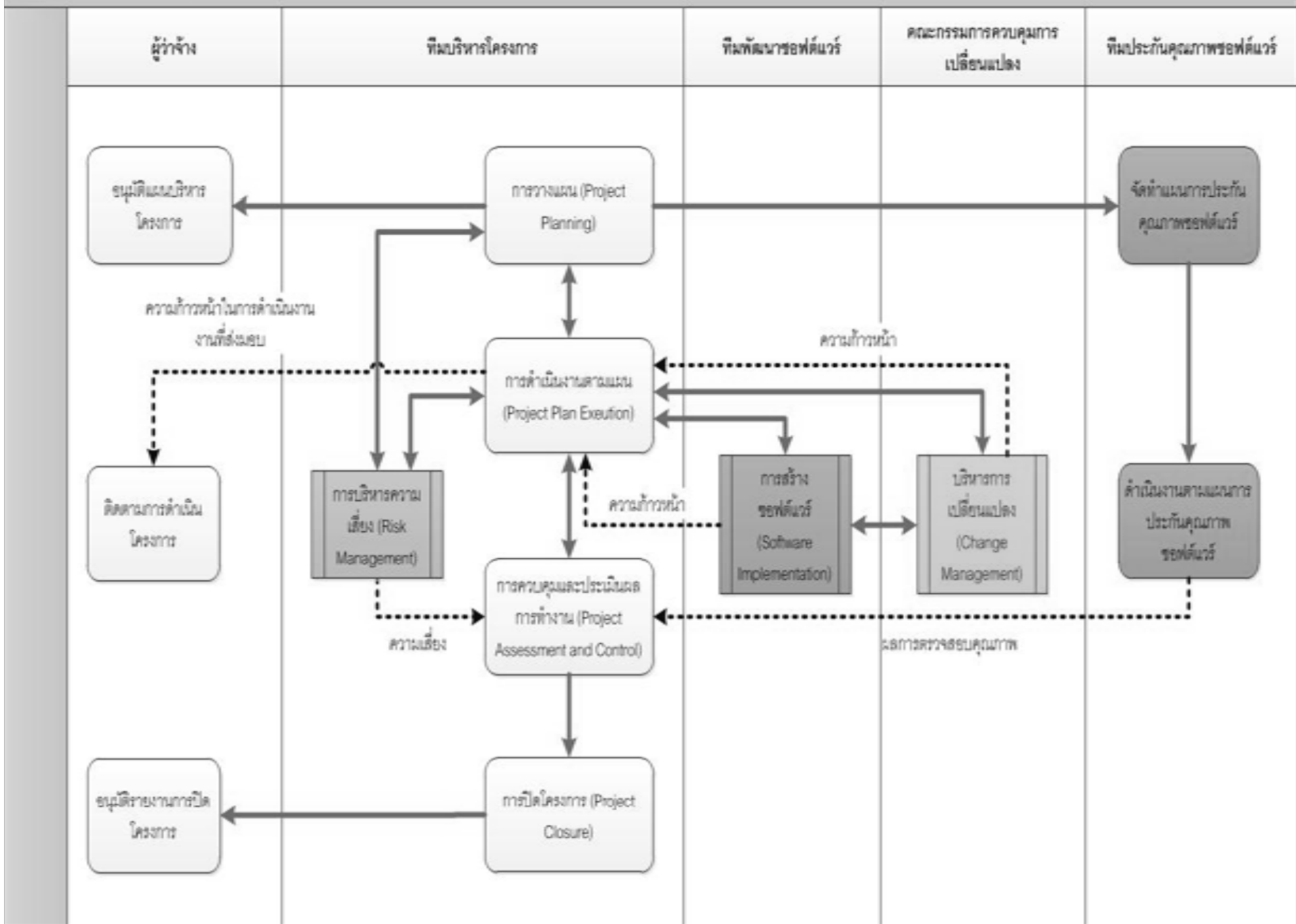
เนื้อหาตามกรอบมาตรฐานกระบวนการพัฒนาซอฟต์แวร์

- กระบวนการ
 - คำอธิบาย (description)
 - ข้อมูลนำเข้า (input)
 - ผลลัพธ์ (output)
 - เจ้าของงาน (owner)
 - ผู้ช่วยเหลือ (assistant)
- เอกสารต้นแบบ (template)
- ความรู้อื่นๆที่เกี่ยวข้อง

หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
 - กระบวนการสร้างซอฟต์แวร์
 - กระบวนการบริหารความเสี่ยง
 - กระบวนการบริหารการเปลี่ยนแปลง
 - การประกันคุณภาพซอฟต์แวร์

กระบวนการบริหารโครงการ (Project Management)



กิจกรรมในกระบวนการบริหารโครงการ

- การวางแผน (Project Planning)
- การดำเนินงานตามแผน (Project Plan Execution)
- การควบคุมและประเมินผลการทำงาน (Project Assessment and Control)
- การปิดโครงการ (Project Closure)

กิจกรรมย่อย : การวางแผน

- ทบทวนขอบเขตการดำเนินงาน
- จัดทำโครงสร้างจัดแบ่งงาน
- ประเมินระยะเวลาและจัดบุคลากรในการทำงาน
- ประเมินความเสี่ยงและจัดทำแผนการบริหารความเสี่ยง
- กำหนดองค์ประกอบซอฟต์แวร์และข้อมูลโครงการ
- จัดทำแผนการบริหารโครงการ
- จัดทำแผนการประกันคุณภาพซอฟต์แวร์
- มอบหมายงานและตกลงยอมรับแผนงานร่วมกัน

ตัวอย่างการอธิบายกิจกรรมย่อย : ทบทวนขอบเขตการดำเนินงาน

คำอธิบาย	ผู้จัดการโครงการและทีมงานบริหารโครงการทำการทบทวนขอบเขตงาน (Statement of work) โดยดำเนินการร่วมกับหัวหน้าทีมพัฒนาซอฟต์แวร์ และหัวหน้าทีมประกันคุณภาพ เพื่อให้เกิดความเข้าใจในขอบเขตงานตรงกันในทีมงานทั้งหมด รวมทั้งเพื่อให้แต่ละทีมงานสามารถไปดำเนินการจัดทำแผนงานย่อยของแต่ละทีมงานได้
ข้อมูลนำเข้า	ขอบเขตงานการพัฒนาซอฟต์แวร์ที่ระบุอยู่ในข้อกำหนดการจ้างหรือสัญญาว่าจ้าง
ผลลัพธ์	ความเข้าใจในขอบเขตงานโครงการ
เจ้าของงาน	ผู้จัดการโครงการ
ผู้ช่วยเหลือ	ทีมบริหารโครงการ

กิจกรรมย่อย : การดำเนินงานตามแผน

- ติดตามและบันทึกความก้าวหน้าในการดำเนินงานตามแผนการบริหารโครงการ
- ติดตามการดำเนินงานตามแผนตอบสนองความเสี่ยง
- ติดตามการดำเนินการตามคำร้องขอการเปลี่ยนแปลง
- ปรับแผนการดำเนินงาน
- สํารองและกู้กลับข้อมูลโครงการ

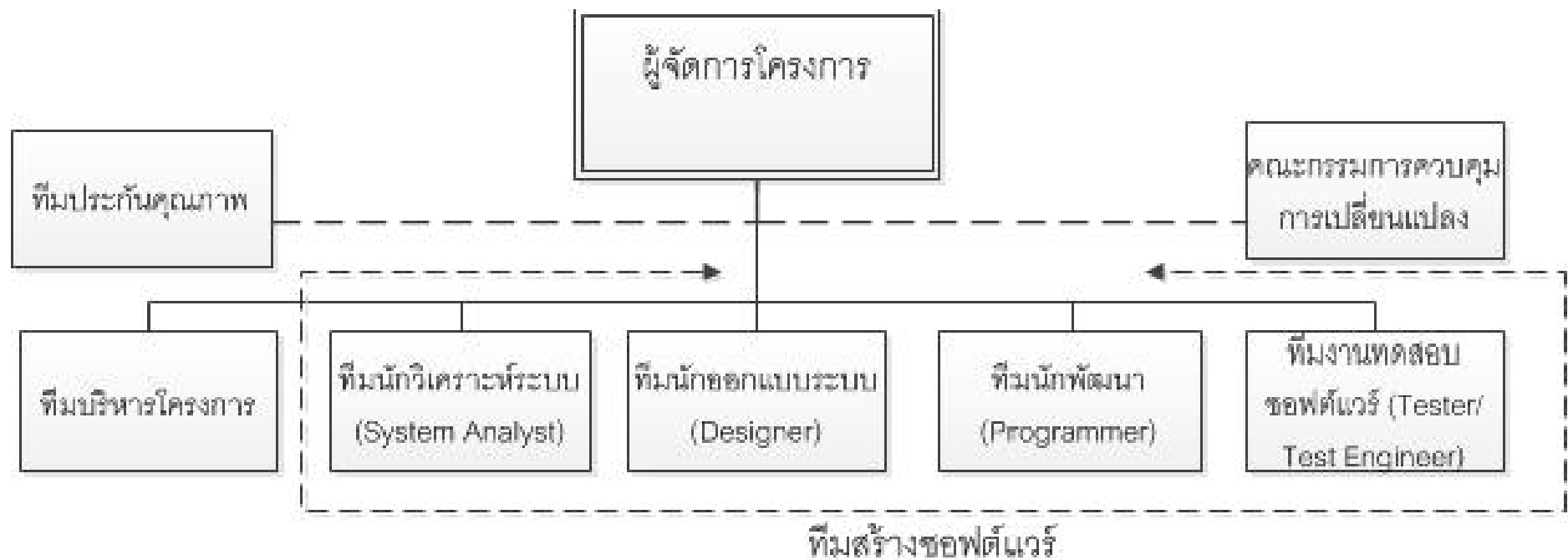
กิจกรรมย่อย : การตอบดูและประเมินผลการทำงาน

- การประเมินความก้าวหน้าในการดำเนินงาน
- การประเมินความเบี่ยงเบนจากแผนการดำเนินงาน

กิจกรรมย่อย : การปิดโครงการ

- ดำเนินการส่งมอบซอฟต์แวร์
- ประสานการตรวจรับซอฟต์แวร์
- ปรับปรุงข้อมูลโครงการให้ตรงกับความเป็นจริง
- จัดทำรายงานการปิดโครงการ

การจัดโครงสร้างทีมงาน



ผู้จัดการโครงการ

หน้าที่หลัก ๆ ของผู้จัดการโครงการ

- กำหนดขอบเขตโครงการ
- กำหนดผู้มีส่วนได้ส่วนเสีย ผู้ตัดสินใจ และวิธีดำเนินการ
- พัฒนารายละเอียดของงาน
- ประเมินเวลาที่ต้องการ
- พัฒนาผังการบริหารโครงการเริ่มแรก
- กำหนดทรัพยากรและงบประมาณที่ต้องการ
- ประเมินความต้องการ
- ระบุและประเมินความเสี่ยง
- เตรียมแผนฉุกเฉิน
- กำหนดความพึงพาระหว่างกิจกรรม
- กำหนดและตามรอยหลักไมล์ที่วิกฤต
- มีส่วนร่วมในขั้นตอนการทบทวน
- ปกป้องรักษาทรัพยากรที่จำเป็น
- บริหารกระบวนการควบคุมการเปลี่ยนแปลง
- รายงานสถานภาพโครงการ

ผู้จัดการโครงการ

- คุณสมบัติของผู้จัดการโครงการ
 - ความรู้การบริหารโครงการ
 - ประยุกต์ความรู้ มาตรฐาน และกฎระเบียบ
 - ความรู้สภาวะแวดล้อมโครงการ ผู้จัดการโครงการต้องเข้าใจการเปลี่ยนแปลง และการทำงานในองค์กร สภาวะแวดล้อมทางด้านการเมือง และกายภาพ
 - ความรู้ในพื้นฐานด้านเทคนิค เช่น ภาษา หรือ framework ที่ใช้ในการพัฒนาซอฟต์แวร์ อาจไม่จำเป็นต้องรู้ในรายละเอียด แต่ควรมีความเข้าใจในภาพรวมของเทคนิคที่เลือกใช้ในแต่ละโครงการ
 - ความรู้ความเข้าใจด้านกระบวนการพัฒนาซอฟต์แวร์ที่เลือกใช้ในโครงการ เช่น waterfall, iterative หรือ agile

ผู้จัดการโครงการ

- คุณสมบัติของผู้จัดการโครงการ
 - มีทักษะในการเจรจาต่อรอง
 - มีทักษะด้านการปฏิสัมพันธ์กับหน่วยงานภายในและภายนอก
 - ความรู้และทักษะทั่วไปด้านการบริหาร ผู้จัดการโครงการควรเข้าใจประเด็นที่สำคัญที่เกี่ยวข้องกับการบริหารด้านการเงิน บัญชี การจัดซื้อจัดจ้าง การตลาด สัญญา การผลิต การกระจายสินค้า การส่งกำลังบำรุง (logistics) ห่วงโซ่อุปทาน การวางแผนเชิงยุทธศาสตร์ การวางแผนเชิงกลยุทธ์ การบริหารการปฏิบัติงาน โครงสร้างองค์การ การบริหารพฤติกรรมบุคคล

ทีมบริหารโครงการ

- มีหน้าที่ในการ วางแผนการดำเนินงานในโครงการ ควบคุมกำกับดูแลการดำเนินการให้เป็นไปตามแผนงาน ประสานงานโครงการ ควบคุมคุณภาพให้ เป็นไปตามที่มาตรฐานกำหนด และการดำเนินการปิดโครงการ บทบาทที่ เกี่ยวข้องในทีมงานบริหารโครงการ
 - ผู้จัดการโครงการ
 - ทีมสร้างซอฟต์แวร์
 - ทีมงานประกันคุณภาพซอฟต์แวร์

ทีมงาน/คณะทำงานควบคุมการเปลี่ยนแปลง

- มีหน้าที่ในการวิเคราะห์คำร้องขอการเปลี่ยนแปลง โดยวิเคราะห์ถึงผลกระทบที่มีต่อโครงการ จัดลำดับความสำคัญของการร้องขอ รวมถึงติดตามการเปลี่ยนแปลงที่เกิดขึ้น
- การวิเคราะห์ผลกระทบ ควรประกอบด้วย ผู้จัดการโครงการ, ตัวแทนจากทีมนักวิเคราะห์ระบบ, ตัวแทนจากทีมนักออกแบบระบบ, ตัวแทนทีมนักพัฒนาซอฟต์แวร์, ตัวแทนจากนักทดสอบระบบ, ตัวแทนจากผู้ใช้งานระบบในฝั่งผู้ว่าจ้าง
- การจัดการองค์ประกอบซอฟต์แวร์

เครื่องมือในการบริหารโครงการ

ความรู้	เทคนิคและเครื่องมือ
การบริหารการบูรณาการ	วิธีการเลือกโครงการ ระเบียบวิธีการบริหารโครงการ การวิเคราะห์ผู้มีส่วนได้เสีย เอกสารสิทธิ์โครงการ (project charters) แผนการบริหารโครงการ ซอฟต์แวร์การบริหารโครงการ คณะกรรมการควบคุมการเปลี่ยนแปลง การบริหารองค์ประกอบซอฟต์แวร์ การประชุมทบทวนโครงการ ระบบการอนุมัติงาน
การบริหารขอบเขต	ข้อกำหนดขอบเขตโครงการ โครงสร้างจำแนกงาน ข้อกำหนดของงาน แผนการบริหารขอบเขต การวิเคราะห์ความต้องการ การควบคุมการเปลี่ยนขอบเขต
การบริหารเวลา	แผนภูมิแกนต์ ผังเครือข่ายโครงการ การวิเคราะห์เส้นทางวิกฤต เทคนิคการทบทวนและประเมินผลการทำงาน (PERT) ตารางเวลาช่วงวิกฤต การเร่งรัดเวลา (crashing) เส้นทางลัด (fast track) การทบทวนหลักไมล์ (milestones)
การบริหารค่าใช้จ่าย	มูลค่าปัจจุบัน อัตราผลตอบแทนจากการลงทุน การวิเคราะห์การจ่ายคืนทุน แฟ้มธุรกิจ (business case) การบริหารมูลค่าที่ได้รับ (earned value management) การบริหารกลุ่มโครงการ (project portfolio management) ประมาณการค่าใช้จ่าย แผนการบริหารค่าใช้จ่าย ซอฟต์แวร์ด้านการเงิน

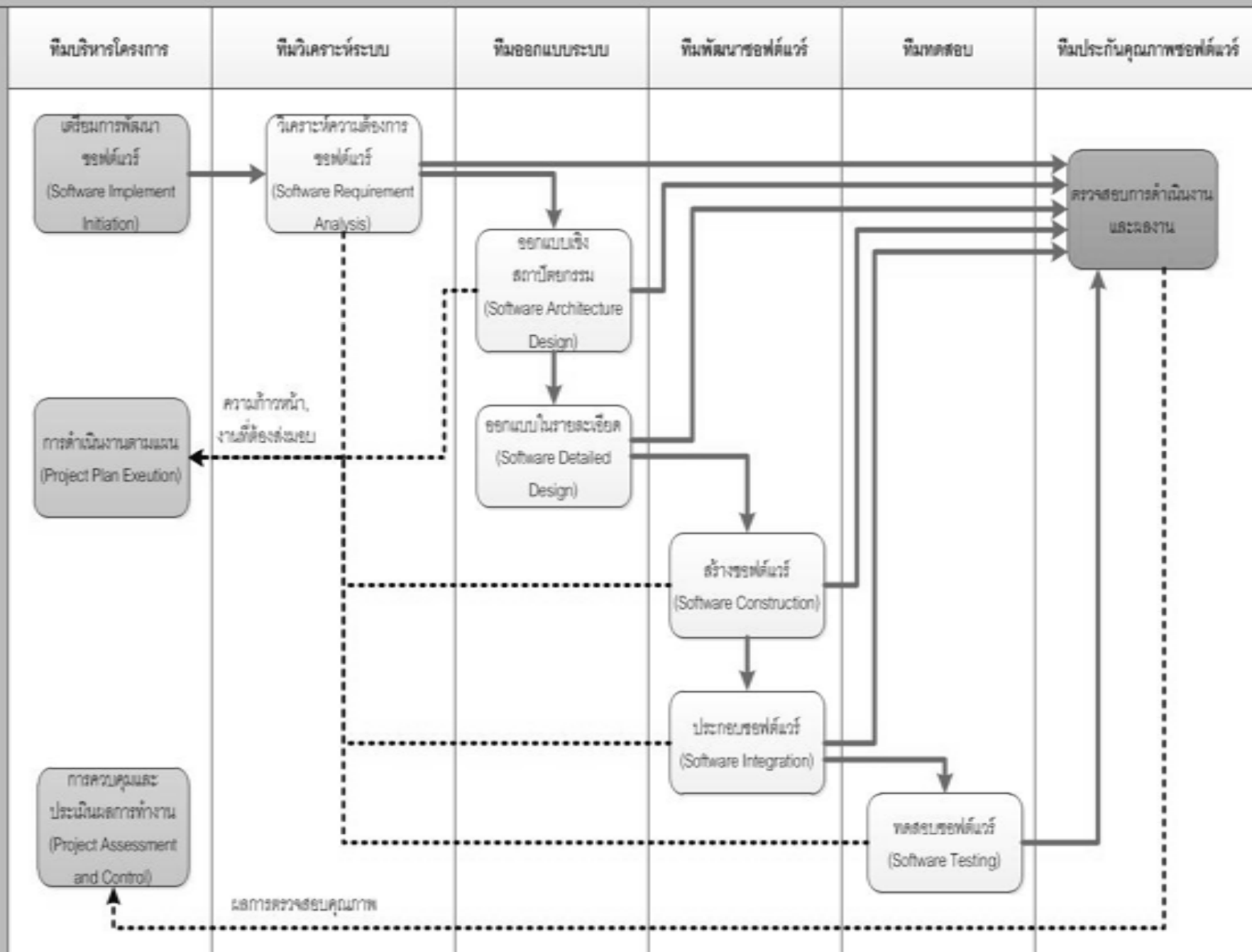
เครื่องมือในการบริหารโครงการ

ความรู้	เทคนิคและเครื่องมือ
การบริหารคุณภาพ	ซิกส์ซิกมา (six sigma) ฝังควบคุมคุณภาพ ฝังพาเรโต ฝังก้างปลา หรือ ฝังอิชิกาวา การตรวจสอบคุณภาพ (quality audit) ตัวแบบวุฒิภาวะ (maturity models) วิธีการเชิงสถิติ
การบริหารทรัพยากรมนุษย์	เทคนิคการจูงใจ การฟังอย่างเห็นอกเห็นใจ (empathic listening) สัญญาทีมงาน ฝังการจัดการองค์กร มอบหมายความรับผิดชอบ แผนภูมิแบบแท่งทรัพยากร การจัดระดับทรัพยากร การสร้างทีม
การบริหารการสื่อสาร	แผนการบริหารการสื่อสาร การบริหารความขัดแย้ง การเลือกสื่อการสื่อสาร โครงสร้างพื้นฐานการสื่อสาร รายงานสถานภาพ แม่แบบ เว็บไซต์โครงการ
การบริหารการจัดซื้อจัดจ้าง	การวิเคราะห์การทำการหรือการซื้อ สัญญา คำร้องขอข้อเสนอโครงการ หรือข้อเสนอราคา การเลือกแหล่งสินค้าหรือบริการ การต่อรอง การจัดซื้อจัดจ้างแบบอิเล็กทรอนิกส์
การบริหารความเสี่ยง	แผนการบริหารความเสี่ยง ฝังผลกระทบ/ความเป็นไปได้ การจัดลำดับความเสี่ยง การจำลองแบบมอนติ คาร์โล (Monte Carlo simulation) การติดตามความเสี่ยงสิบอันดับแรก

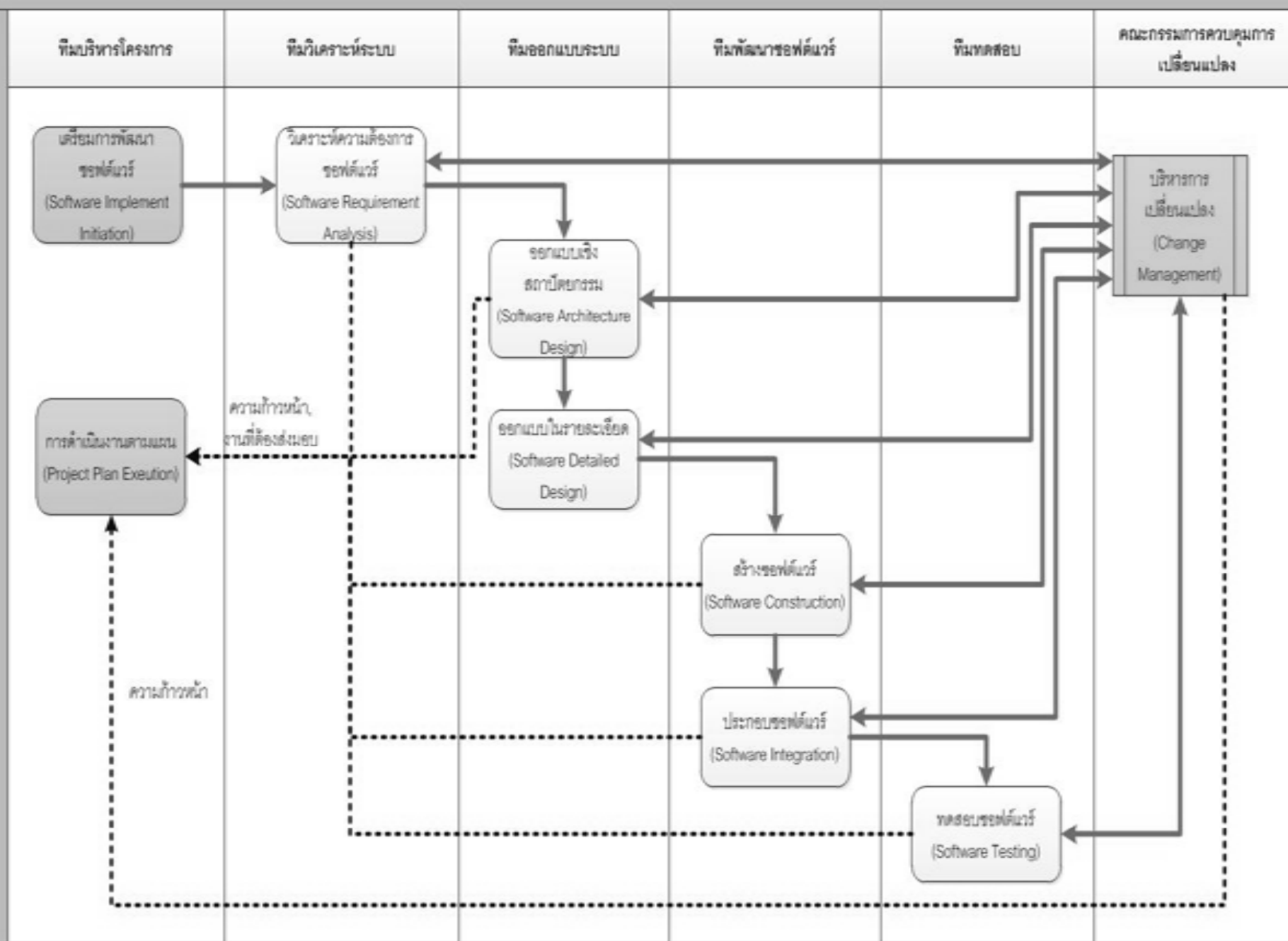
หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
 - กระบวนการบริหารความเสี่ยง
 - กระบวนการบริหารการเปลี่ยนแปลง
 - การประกันคุณภาพซอฟต์แวร์

กระบวนการสร้างซอฟต์แวร์ (Software Implementation)



กระบวนการสร้างซอฟต์แวร์ (Software Implementation)



กิจกรรมในกระบวนการสร้างซอฟต์แวร์

- การเตรียมการพัฒนาซอฟต์แวร์
- การวิเคราะห์ความต้องการซอฟต์แวร์
- การออกแบบเชิงสถาปัตยกรรม
- การออกแบบในรายละเอียด
- สร้างซอฟต์แวร์
- ประกอบซอฟต์แวร์
- ทดสอบซอฟต์แวร์

กิจกรรมย่อย : การเตรียมการพัฒนาซอฟต์แวร์

- ทบทวนและยอมรับแผนงาน
- จัดเตรียมสภาพแวดล้อมในการทำงาน

ตัวอย่างคำอธิบายกิจกรรมย่อย : ทบทวนและยอมรับแผนงาน

คำอธิบาย	หัวหน้าทีมสร้างซอฟต์แวร์ และผู้จัดการโครงการประชุมร่วมกับทีมสร้างซอฟต์แวร์เพื่อทบทวนความเข้าใจในแผนการสร้างซอฟต์แวร์ มอบหมายงานให้กับทีมงานพัฒนาซอฟต์แวร์ และแจ้งให้ทีมงานได้เข้าใจถึงนโยบายการจัดการองค์ประกอบซอฟต์แวร์ มาตรฐานต่างๆที่กำหนดให้ใช้ในโครงการ และการประกันคุณภาพซอฟต์แวร์ ผู้จัดการโครงการควรทำความเข้าใจกับทีมสร้างซอฟต์แวร์ถึงเป้าหมายและประโยชน์ของการสร้างซอฟต์แวร์ รวมถึงความคาดหวังจากผู้ว่าจ้าง การที่ผู้จัดการโครงการแบ่งปันข้อมูลในส่วนนี้ จะช่วยให้ทีมงานรู้สึกเป็นส่วนหนึ่งและมีความสำคัญกับโครงการ อีกทั้งยังเสริมสร้างขวัญและกำลังใจให้กับทีมงานอีกด้วย ในการประชุมร่วมกัน ควรจัดทำรายงานการประชุมอย่างเป็นทางการ เพื่อให้มั่นใจว่าทีมสร้างซอฟต์แวร์ มีความเข้าใจ ยอมรับและพร้อมที่จะดำเนินงานตามแผนการบริหารโครงการ
ข้อมูลนำเข้า	แผนการบริหารโครงการ
ผลลัพธ์	รายงานการประชุม
เจ้าของงาน	หัวหน้าทีมสร้างซอฟต์แวร์
ผู้ช่วยเหลือ	ผู้จัดการโครงการ, ทีมสร้างซอฟต์แวร์

กิจกรรมย่อย : การวิเคราะห์ความต้องการซอฟต์แวร์

- วิเคราะห์และจัดทำข้อกำหนดความต้องการระบบ
- ตรวจสอบความครบถ้วนและถูกต้องของเอกสารความต้องการซอฟต์แวร์และจัดทำรายงานผลการตรวจสอบ
- อนุมัติเอกสารความต้องการซอฟต์แวร์
- เบสไลน์ข้อกำหนดความต้องการซอฟต์แวร์

กิจกรรมย่อย : การออกแบบเชิงสถาปัตยกรรมและการออกแบบในรายละเอียด

- ทบทวนข้อกำหนดความต้องการซอฟต์แวร์
- ออกแบบระบบซอฟต์แวร์ตามมาตรฐานที่กำหนด
- จัดทำเมทริกซ์การติดตาม (Traceability Matrix)
- ตรวจสอบความครบถ้วนและถูกต้องของข้อกำหนดการออกแบบซอฟต์แวร์
- จัดทำกรณีทดสอบ
- ตรวจสอบความครบถ้วนและถูกต้องของกรณีทดสอบ
- อนุมัติข้อกำหนดการออกแบบระบบ

กิจกรรมย่อย : การสร้างซอฟต์แวร์

- การเขียนโปรแกรมขึ้นส่วนซอฟต์แวร์
- จัดทำกรณีทดสอบและทดสอบขึ้นส่วนซอฟต์แวร์
- ปรับปรุงเมตริกซ์การติดตาม
- ตรวจสอบการสร้างซอฟต์แวร์และทดสอบขึ้นส่วนซอฟต์แวร์

กิจกรรมย่อย : การประกอบชิ้นส่วนซอฟต์แวร์

- ประกอบชิ้นส่วนซอฟต์แวร์
- ตรวจสอบการประกอบชิ้นส่วนซอฟต์แวร์

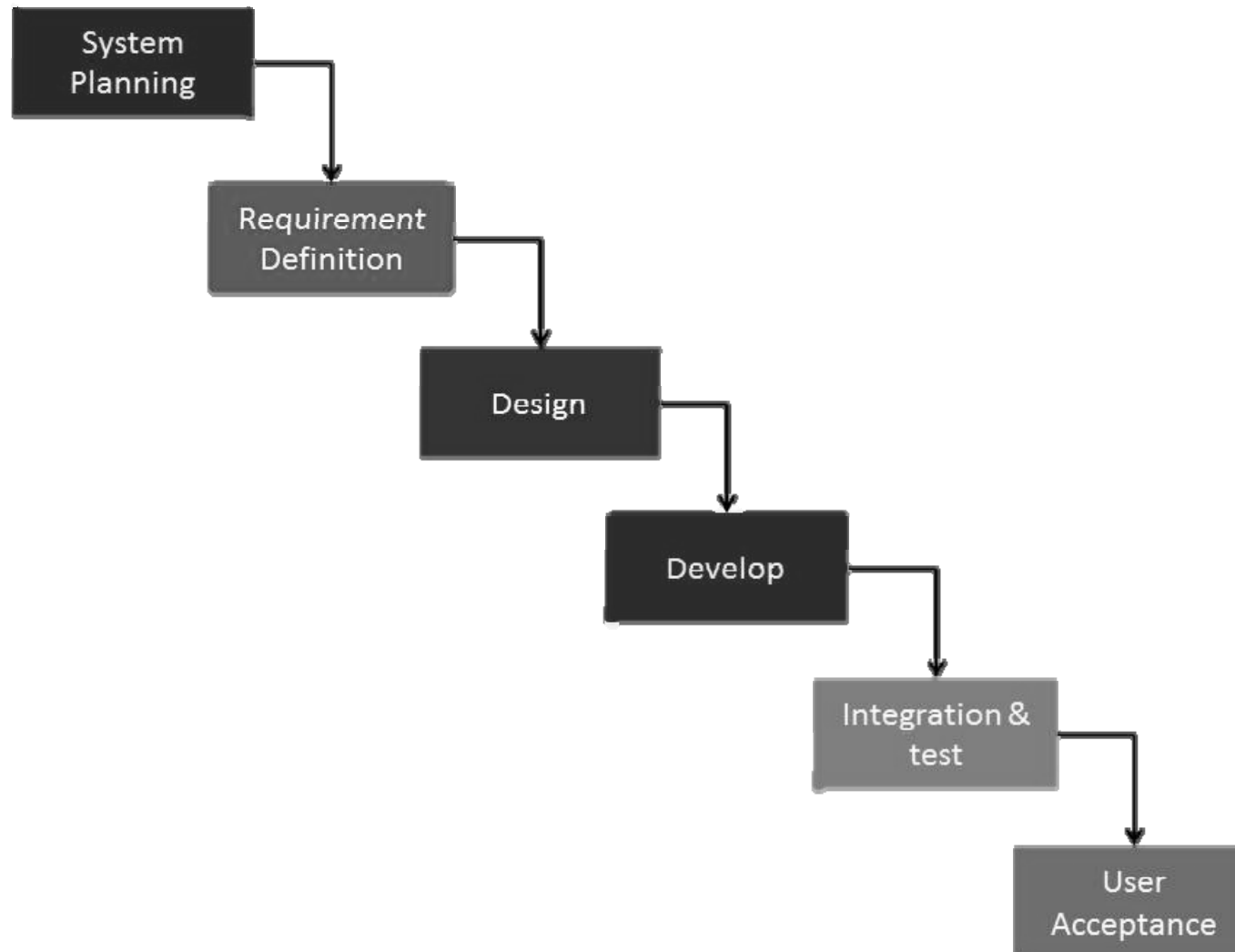
กิจกรรมย่อย : การทดสอบซอฟต์แวร์

- จัดเตรียมสภาพแวดล้อมสำหรับทดสอบและติดตั้งซอฟต์แวร์
- ทดสอบซอฟต์แวร์
- ปรับปรุงแก้ไขซอฟต์แวร์
- ทำรีเกรสชันเทสต์ (regression test) และรายงานผลการทดสอบ
- ตรวจสอบผลการทดสอบ
- จัดทำเอกสารคู่มือการใช้งาน/คู่มือการพัฒนาระบบ
- เบสไลน์ซอฟต์แวร์และองค์ประกอบซอฟต์แวร์
- ตรวจสอบความครบถ้วน/ถูกต้องของเอกสาร
- อนุมัติเอกสาร

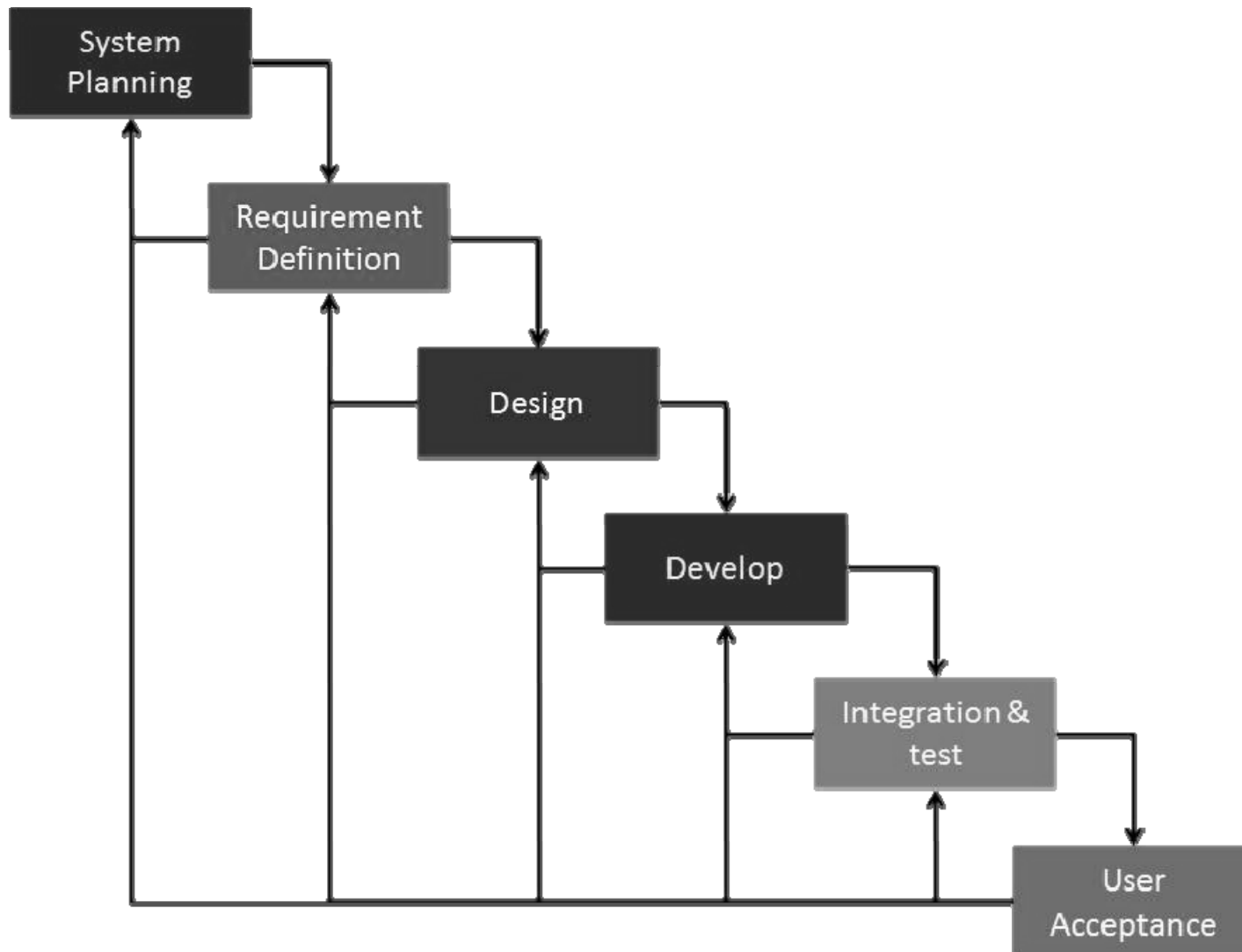
กิจกรรมย่อย : ดำเนินการส่งมอบ

- ประสานการทดสอบการยอมรับระบบ
- จัดทำแผนการนำระบบขึ้นใช้งานจริงและแผนการบำรุงรักษาระบบ
- ตรวจสอบความครบถ้วน/ถูกต้องของเอกสาร
- อนุมัติแผน
- ฝึกอบรมและถ่ายทอดเทคโนโลยี
- จัดเตรียมและติดตั้งซอฟต์แวร์บนสภาพแวดล้อมจริง
- ตรวจสอบการดำเนินการตามแผนการนำระบบขึ้นใช้งานจริง

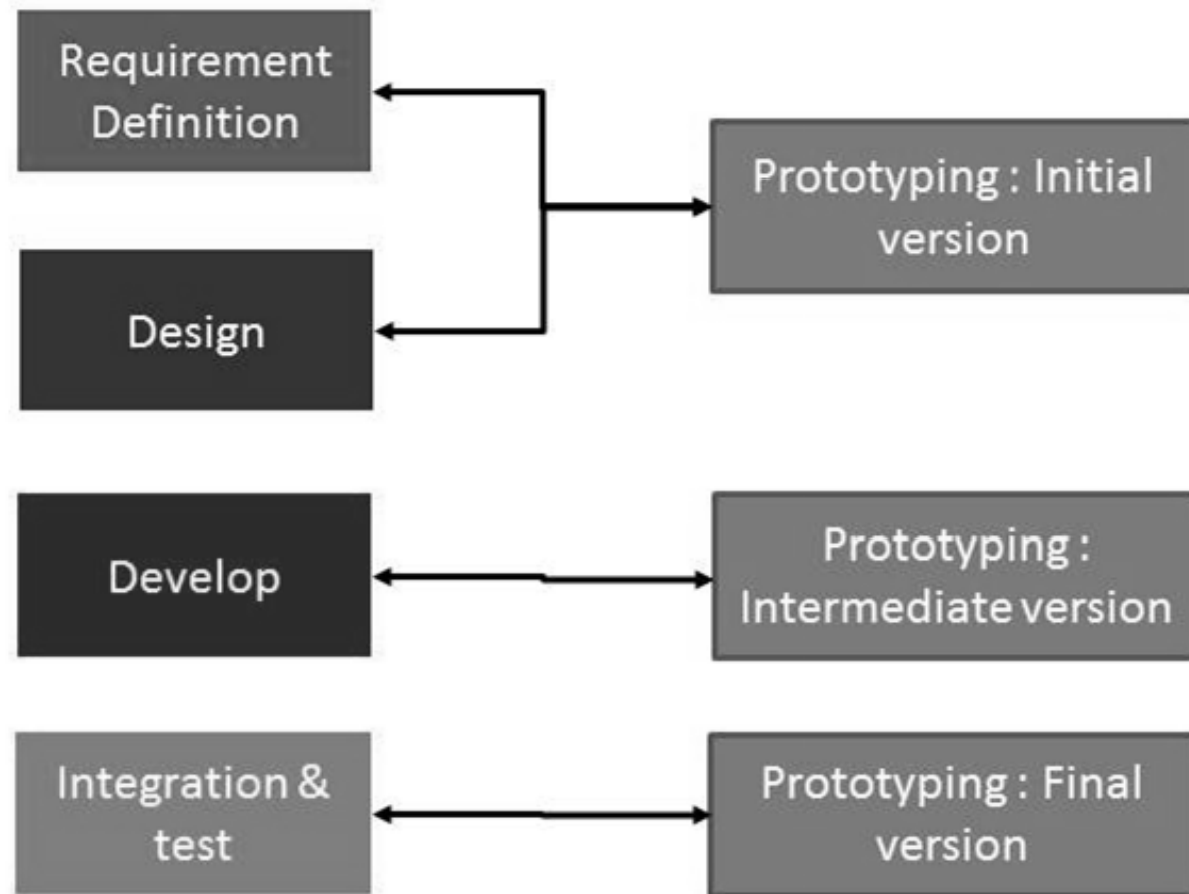
วงจรชีวิตการพัฒนาซอฟต์แวร์ : Classis Waterfall Lifecycle model



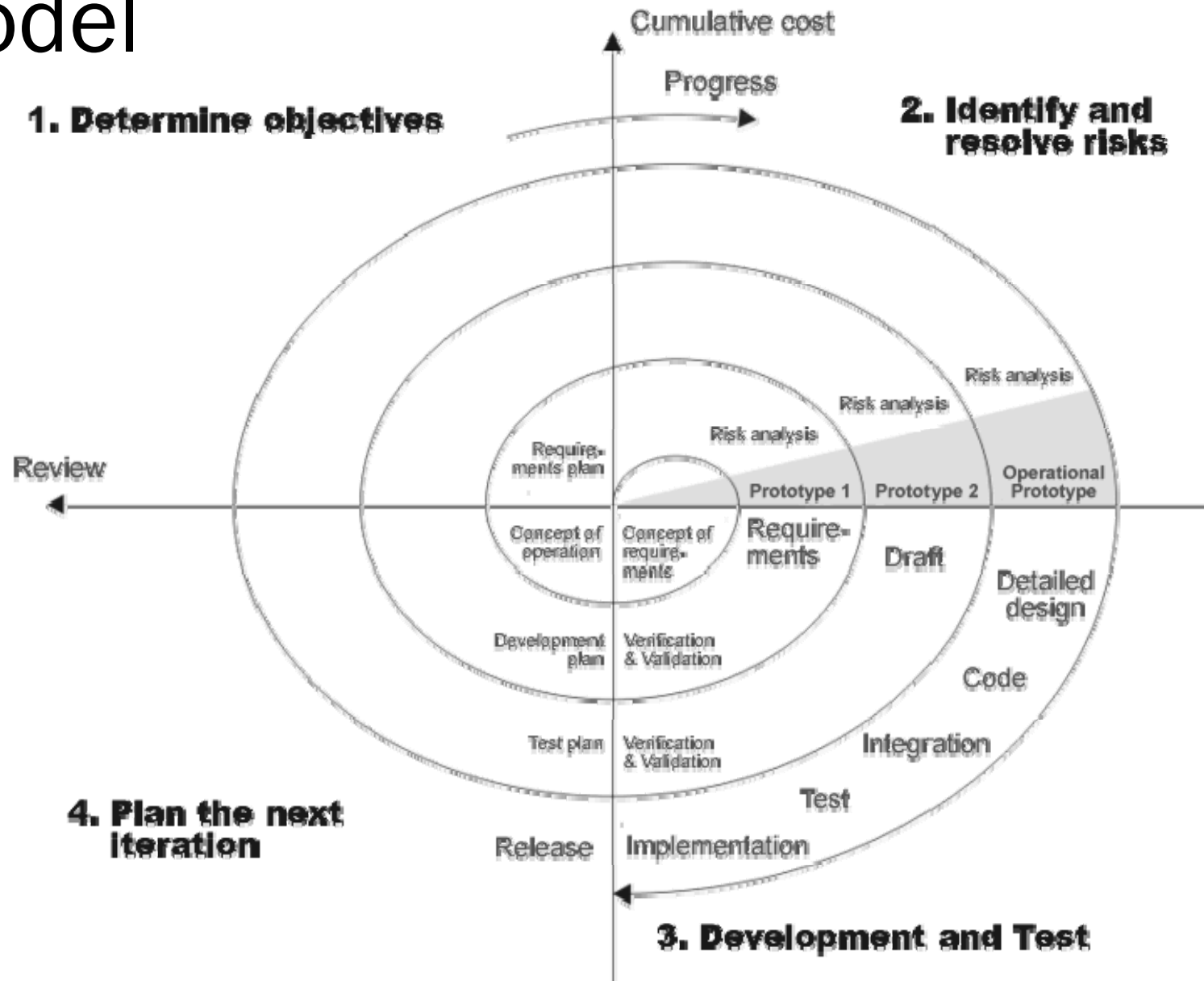
วงจรชีวิตการพัฒนาซอฟต์แวร์ : Applied Waterfall life cycle model



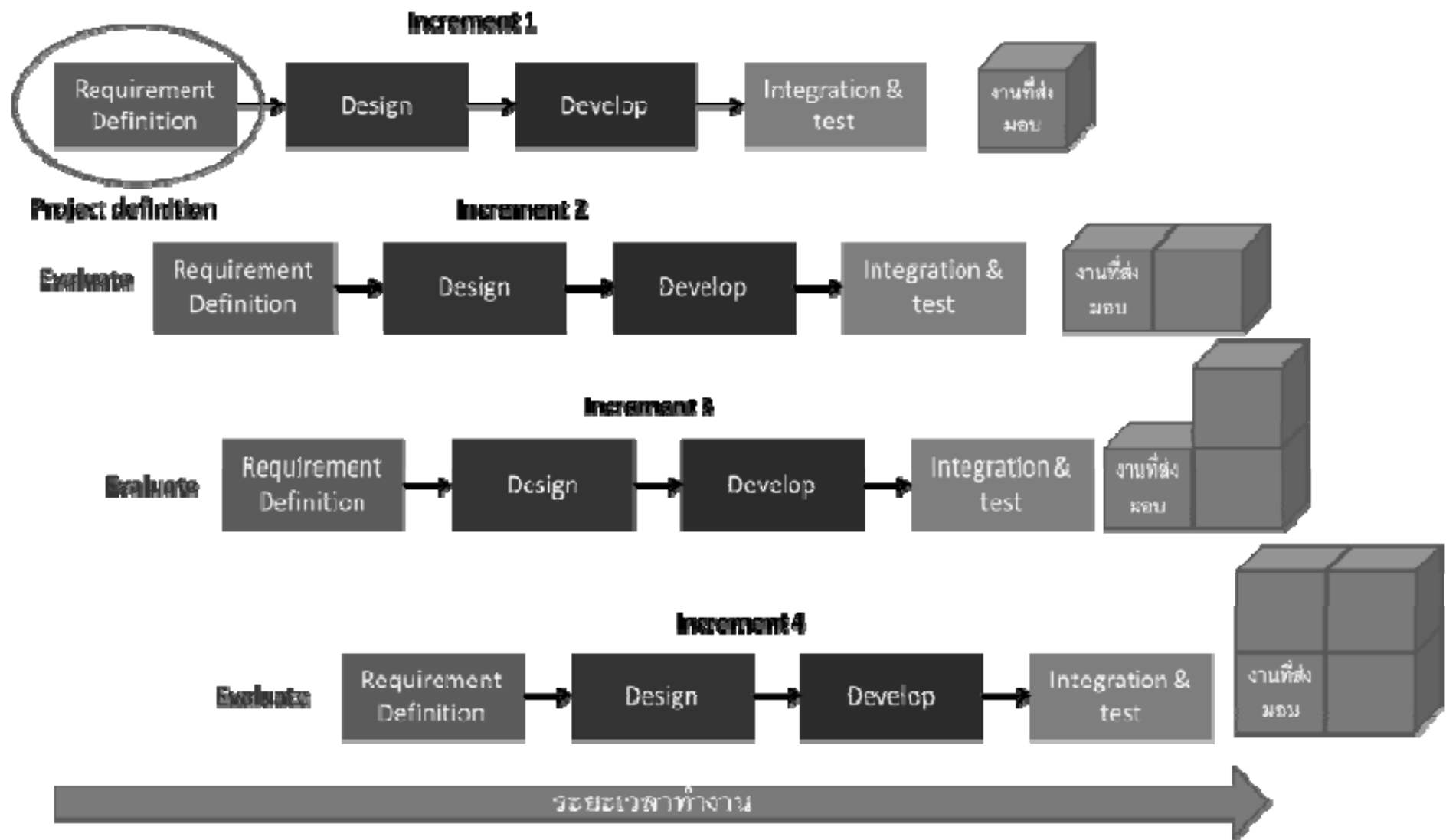
วงจรชีวิตการพัฒนาซอฟต์แวร์ : Applied Prototyping life cycle model



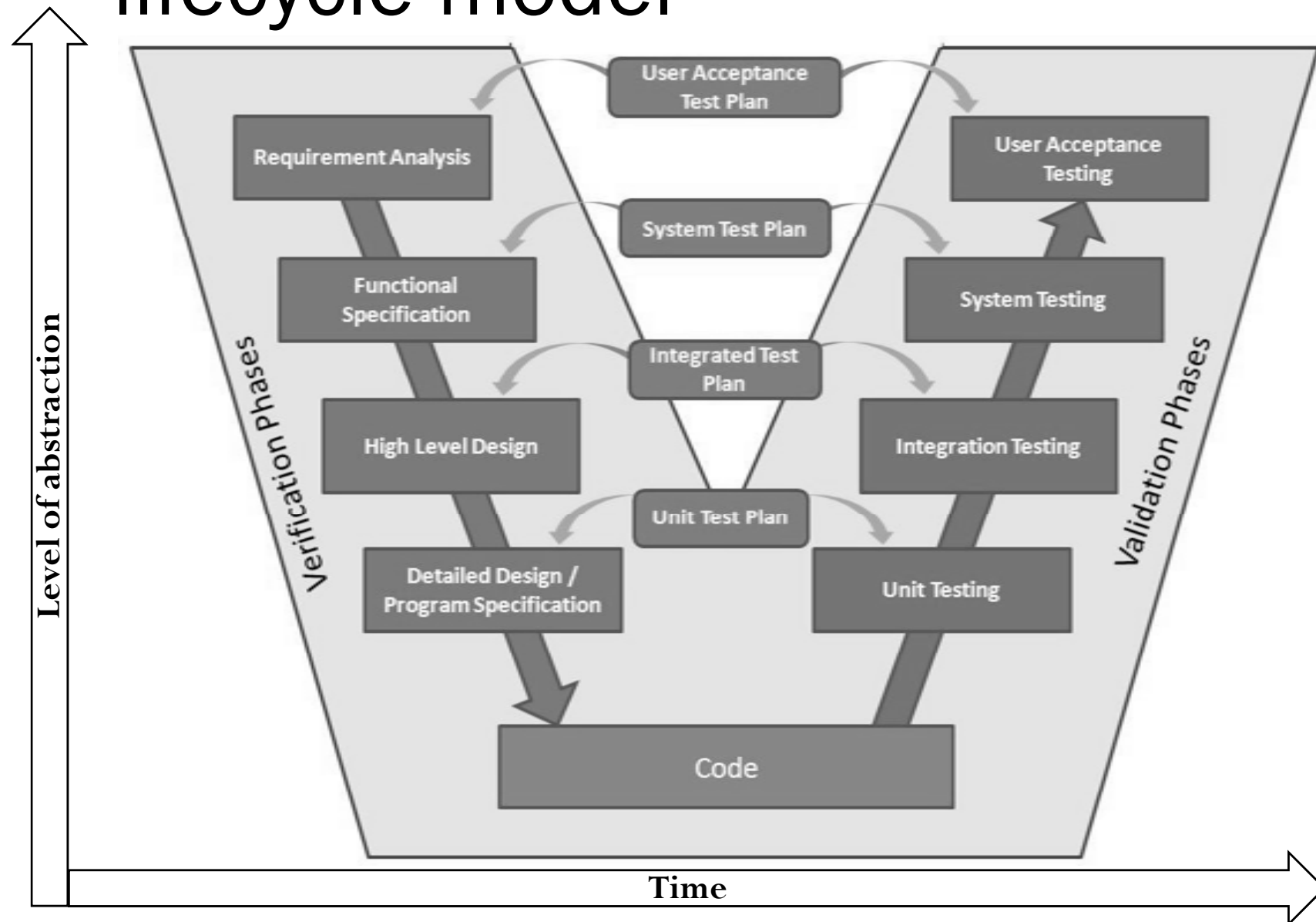
วงจรชีวิตการพัฒนาซอฟต์แวร์ : Spiral lifecycle model



วงจรชีวิตการพัฒนาซอฟต์แวร์ : Incremental lifecycle model



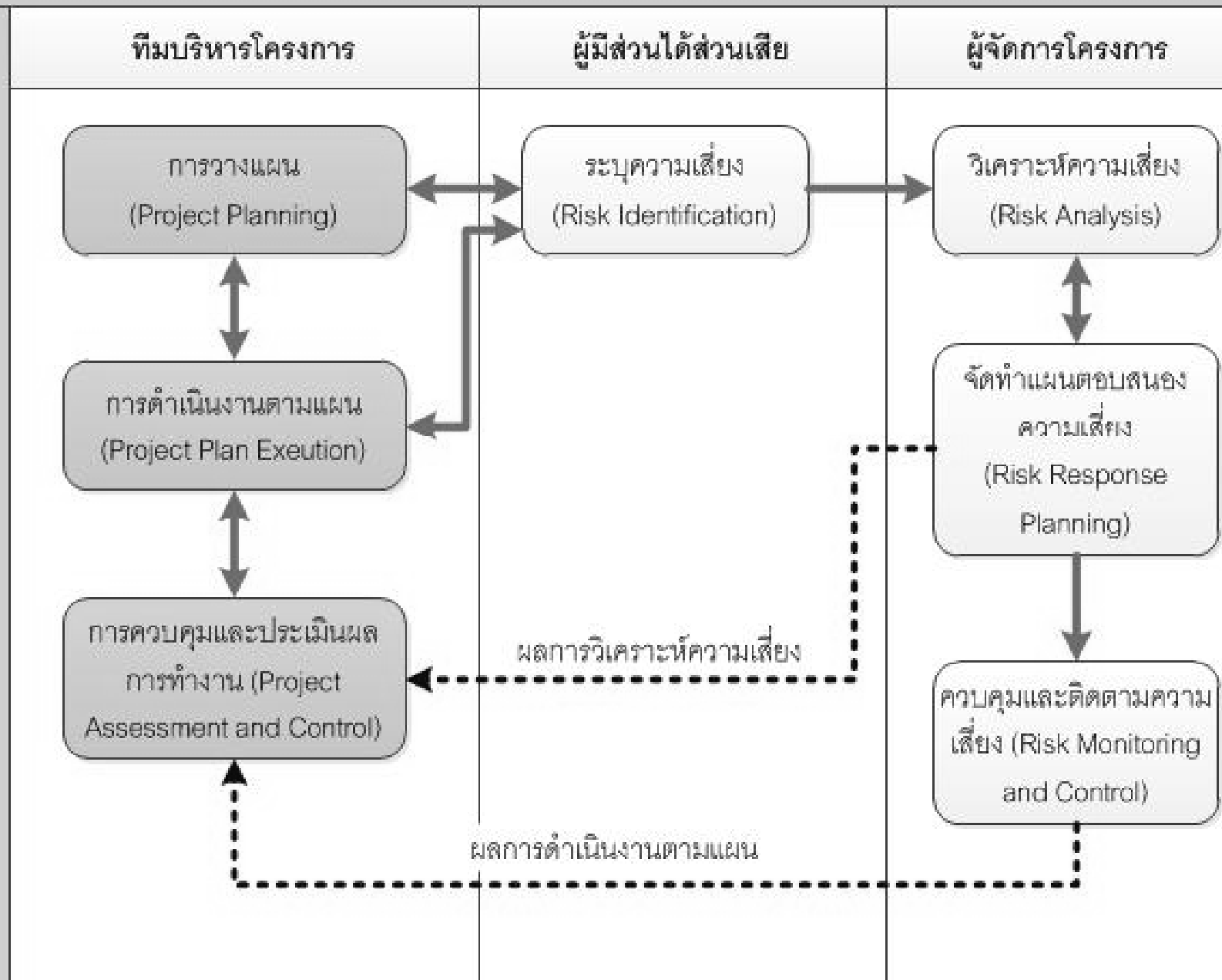
วงจรชีวิตการพัฒนาซอฟต์แวร์ : V-Shaped lifecycle model



หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
- กระบวนการบริหารความเสี่ยง
- กระบวนการบริหารการเปลี่ยนแปลง
- การประกันคุณภาพซอฟต์แวร์

การบริหารความเสี่ยง (Risk Management)



กิจกรรมในกระบวนการบริหารความเสี่ยง

- การระบุความเสี่ยง
- การวิเคราะห์ความเสี่ยง
- การวางแผนตอบสนองความเสี่ยง
- การควบคุมและติดตามความเสี่ยง

การระบุความเสี่ยง

- เป็นกระบวนการทำความเข้าใจว่าเหตุการณ์ใดที่มีศักยภาพที่จะทำร้ายโครงการ หรือส่งเสริมให้โครงการดีขึ้น การระบุความเสี่ยงที่มีศักยภาพเป็นเรื่องที่จำเป็น ผู้จัดการโครงการต้องระบุความเสี่ยงอย่างต่อเนื่องตามการเปลี่ยนแปลงของสถานะแวดล้อม
- เทคนิคที่ใช้ในการระบุความเสี่ยง
 - การประชุม/ระดมสมอง
 - การสัมภาษณ์
 - การวิเคราะห์ SWOT
 - ฯลฯ

การจำแนกประเภทความเสี่ยง

- ความเสี่ยงด้านตลาด (market risk)
 - ถ้าโครงการเทคโนโลยีสารสนเทศเป็นโครงการที่สร้างผลิตภัณฑ์หรือบริการใหม่ โครงการนี้จะมีประโยชน์ต่อองค์กรหรือไม่ หรือสามารถทำตลาดให้กับผลิตภัณฑ์อื่นหรือไม่
 - ผู้ใช้จะยอมรับและใช้ผลิตภัณฑ์หรือบริการหรือไม่
 - มีใครอื่นที่สร้างผลิตภัณฑ์หรือบริการที่ดีกว่าหรือเร็วกว่าหรือไม่ ซึ่งจะทำให้โครงการเสียเวลาและเงิน
- ความเสี่ยงด้านการเงิน (financial risk)
 - องค์กรสามารถสนับสนุนโครงการให้ดำเนินการได้หรือไม่
 - ผู้มีส่วนได้เสียจะมั่นใจในการคาดการณ์ทางการเงินได้อย่างไร
 - โครงการจะทำได้ตามค่ามูลค่างปัจจุบันสุทธิ อัตราผลตอบแทนจากการลงทุน และระยะเวลาการคืนทุนที่ได้ประมาณการหรือไม่ ถ้าทำไม่ได้ องค์กรสามารถสนับสนุนโครงการให้ดำเนินการต่อไปได้หรือไม่
 - โครงการนี้เป็นทางเลือกที่ดีที่สุดที่จะใช้ทรัพยากรการเงินขององค์กรหรือไม่

การจำแนกประเภทความเสี่ยง (ต่อ)

- ความเสี่ยงด้านเทคโนโลยี (technology risk)
 - โครงการนี้มีความเป็นไปได้เชิงเทคนิคหรือไม่
 - โครงการจะใช้เทคโนโลยีที่มีวุฒิภาวะ หรือใช้เทคโนโลยีที่ล้าหน้าหรือไม่
 - เมื่อไรจึงตัดสินใจว่าจะใช้เทคโนโลยีอะไร
 - ฮาร์ดแวร์ ซอฟต์แวร์และเครือข่ายทำงานได้อย่างเหมาะสมหรือไม่
 - เทคโนโลยีนี้จะมีให้ใช้ทันเวลาหรือไม่
 - เทคโนโลยีนี้อาจล้าสมัยก่อนผลิตภัณฑ์จะผลิตออกมาหรือไม่
 - เทคโนโลยีที่เลือกใช้มีข้อจำกัด หรือไม่สอดคล้องกับความต้องการหรือไม่

การจำแนกประเภทความเสี่ยง (ต่อ)

- ความเสี่ยงด้านคน (people risk)
 - องค์กรมีหรือสามารถหาบุคลากรที่มีทักษะที่เหมาะสมเพื่อให้โครงการเสร็จสมบูรณ์หรือไม่
 - บุคลากรมีทักษะเชิงเทคนิคและเชิงบริหารหรือไม่
 - บุคลากรมีประสบการณ์เพียงพอหรือได้รับการฝึกอบรมที่เพียงพอหรือไม่
 - ผู้บริหารอาวุโสสนับสนุนโครงการหรือไม่
 - มีผู้สนับสนุนโครงการหรือไม่
 - องค์กรมีความคุ้นเคยกับลูกค้าหรือผู้สนับสนุนหรือไม่
 - ความสัมพันธ์กับลูกค้าหรือผู้สนับสนุนโครงการดีแค่ไหน

การจำแนกประเภทความเสี่ยง (ต่อ)

- ความเสี่ยงด้านโครงสร้าง/ กระบวนการ (structure/process risk)
 - โครงการจะเสนอการเปลี่ยนแปลงระดับใดกับขั้นตอนทางธุรกิจและส่วนที่เกี่ยวข้องกับผู้ใช้
 - มีกลุ่มผู้ใช้ที่แตกต่างจำนวนมากแค่ไหนที่โครงการจำเป็นต้องตอบสนองความต้องการ
 - ระบบอื่นที่โครงการต้องทำงานร่วมมีมากแค่ไหน
 - องค์กรมีกระบวนการอยู่แล้วที่จะทำให้โครงการเสร็จสมบูรณ์หรือไม่

การวิเคราะห์ความเสี่ยง

- เป็นการวิเคราะห์ผลกระทบและจัดลำดับความสำคัญของความเสี่ยง
- เทคนิคในการวิเคราะห์ความเสี่ยง
 - การใช้ผังความน่าจะเป็น/ผลกระทบ
 - การใช้ตารางผลกระทบความเสี่ยง
 - Countermeasure value analysis

ตัวอย่างฟังก์ชันความเป็น/พละทบ

ความน่าจะเป็น สูง	1 ความเสี่ยง 6	4 ความเสี่ยง 9	7 ความเสี่ยง 1 ความเสี่ยง 4
ปานกลาง	2 ความเสี่ยง 3	5 ความเสี่ยง 2 ความเสี่ยง 5 ความเสี่ยง 11	8
ต่ำ	3 ความเสี่ยง 7	6 ความเสี่ยง 8 ความเสี่ยง 11	9 ความเสี่ยง 12
	ต่ำ	กลาง	สูง wans:nu

ตัวอย่างตารางผลกระทบความเสี่ยง

ความเสี่ยง (ภัยคุกคาม)	ความน่าจะเป็น (P) 0-1	ผลกระทบ (I) 0-10	คะแนน $P \times I$
สมาชิกหลักของทีมงานออกจากโครงการ	0.40	4	1.6
ลูกค้าไม่สามารถกำหนดขอบเขตและความต้องการ	0.50	6	3.0
ลูกค้าประสบปัญหาการเงิน	0.10	9	0.9
ส่งมอบงานล่าช้า	0.80	6	4.8
เทคโนโลยีไม่บูรณาการกับระบบงานที่มี	0.60	7	4.2
ผู้จัดการด้านฟังก์ชันนำทรัพยากรของโครงการออกไป	0.20	3	0.6
ลูกค้าไม่สามารถได้รับใบอนุญาต	0.05	7	0.4

การวางแผนตอบสนองความเสี่ยง

- การหลีกเลี่ยงความเสี่ยง (risk avoiding) โดยการขจัดสาเหตุของความเสี่ยง เช่น ทีมงานไม่คุ้นเคยกับฮาร์ดแวร์หรือซอฟต์แวร์ ซึ่งทำให้เกิดความเสี่ยงที่มีนัยสำคัญ ดังนั้น การใช้ฮาร์ดแวร์หรือซอฟต์แวร์ที่คุ้นเคยเป็นการหลีกเลี่ยงความเสี่ยง
- การยอมรับความเสี่ยง (risk acceptance) เนื่องจากโอกาสหรือความน่าจะเป็นที่จะเกิดเหตุการณ์เสี่ยงนั้นมีน้อยมาก ในทางกลับกัน ในกรณีที่ความน่าจะเป็นของการเกิดเหตุการณ์เสี่ยงอาจจะค่อนข้างสูง แต่ส่งผลกระทบน้อยมากต่อเวลาและงบประมาณโครงการ ซึ่งไม่คุ้มกับค่าใช้จ่ายที่ต้องเสียไปในการลดความเสี่ยง จึงเป็นเหตุผลที่ผู้จัดการโครงการตัดสินใจยอมรับความเสี่ยงนั้นไว้โดยไม่ทำอะไร ดังนั้น เพื่อจัดการกับความเสี่ยงในกรณีนี้ ผู้จัดการโครงการควรตั้งเงินทุนสำรอง และจัดทำแผนสำรอง
- การโอนความเสี่ยง (risk transfer) หรือย้ายผลของความเสี่ยงและความรับผิดชอบไปยังบุคคลที่สาม เช่น การซื้อประกันฮาร์ดแวร์ ถ้าเครื่องเสียหาย บริษัทรับประกันต้องหาเครื่องมาทดแทนตามที่กำหนดในสัญญา
- การบรรเทาความเสี่ยง (risk mitigation) หรือลดผลกระทบของความเสี่ยงโดยการลดความน่าจะเป็นของเหตุการณ์ เช่น ใช้พนักงานที่มีความชำนาญ การใช้เทคนิคการวิเคราะห์และทดสอบที่หลากหลาย จัดระบบควบคุมและตรวจสอบความก้าวหน้าของงานเป็นระยะ

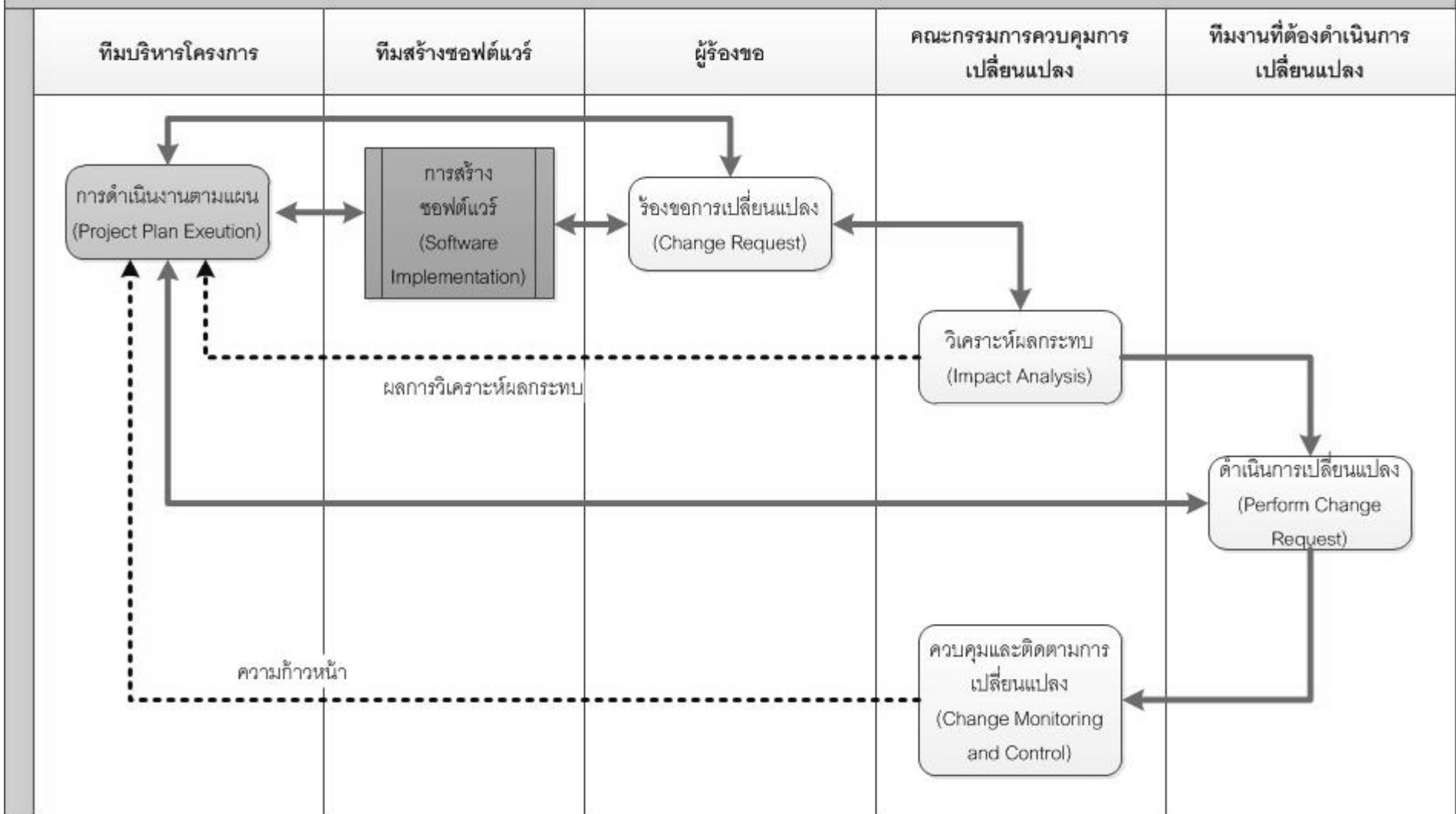
การตอบดูและติดตามความเสี่ยง

หัวข้อความเสี่ยง	เดือน นี้	เดือนที่ แล้ว	จำนวนเดือน	ความก้าวหน้าในการแก้ความเสี่ยง
การวางแผนไม่เพียงพอ	1	2	4	กำลังทำการทบทวนแผนทั้งโครงการ
การนิยามขอบเขตของ โครงการไม่ดี	2	3	3	จัดประชุมกับลูกค้าของโครงการและ ผู้สนับสนุนเพื่อให้ขอบเขตของ โครงการชัดเจน
ขาดผู้นำ	3	1	2	เพิ่งกำหนดผู้จัดการโครงการคน ใหม่หลังจากผู้จัดการคนก่อนลาออก
การประมาณการค่าใช้จ่าย ไม่ดี	4	4	3	กำลังทบทวนการประมาณการ ค่าใช้จ่าย
การประมาณการเวลาไม่ดี	5	5	3	กำลังทบทวนการประมาณระยะเวลา

หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
- กระบวนการบริหารความเสี่ยง
- กระบวนการบริหารการเปลี่ยนแปลง
- การประกันคุณภาพซอฟต์แวร์

การบริหารการเปลี่ยนแปลง (Change Management)



กิจกรรมในกระบวนการบริหารการเปลี่ยนแปลง

- การร้องขอการเปลี่ยนแปลง
- การวิเคราะห์ผลกระทบ
 - ผลกระทบต่อขอบเขตงาน
 - ผลกระทบต่อคุณภาพของงานหรือสิ่งที่ส่งมอบ
 - ผลกระทบต่อระยะเวลาและการส่งมอบงาน
- ดำเนินการเปลี่ยนแปลง
 - Software configuration management
- ควบคุมและติดตามความก้าวหน้า

หัวข้อที่น่าสนใจ

- ความหมายและนิยามของซอฟต์แวร์ที่มีคุณภาพ
- มาตรฐานกระบวนการพัฒนาซอฟต์แวร์
- กระบวนการบริหารโครงการ
- กระบวนการสร้างซอฟต์แวร์
- กระบวนการบริหารความเสี่ยง
- กระบวนการบริหารการเปลี่ยนแปลง
- การประกันคุณภาพซอฟต์แวร์



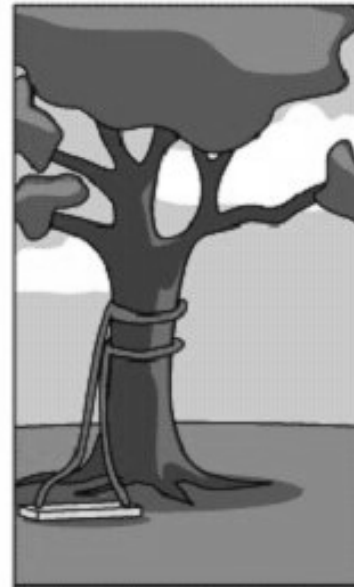
How the customer explained it



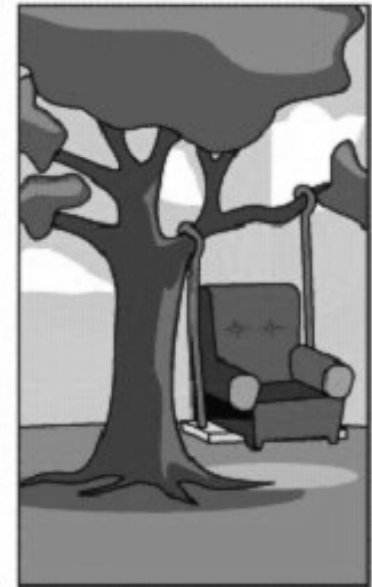
How the Project Leader understood it



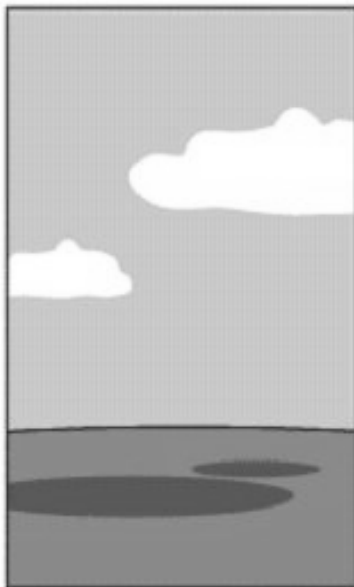
How the Analyst designed it



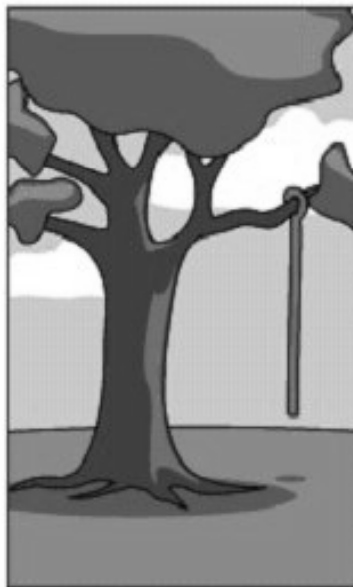
How the Programmer wrote it



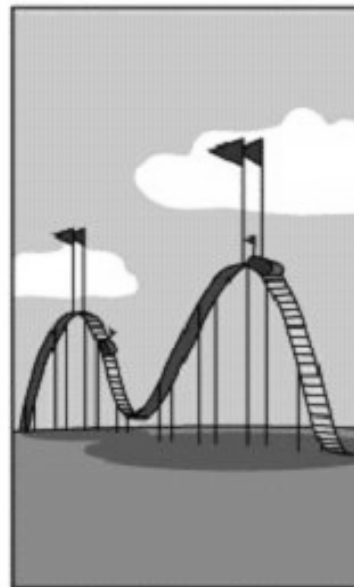
How the Business Consultant described it



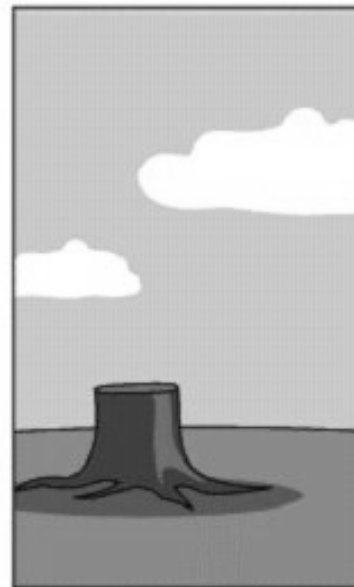
How the project was documented



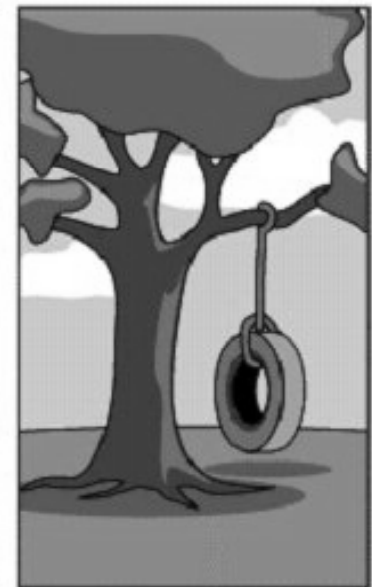
What operations installed



How the customer was billed



How it was supported



What the customer really needed

ทำไมต้องประกันคุณภาพ?

- เพื่อให้มีกระบวนการที่ดี ที่สามารถผลิตซอฟต์แวร์ออกมาได้อย่างมีคุณภาพสม่ำเสมอ
- เพื่อให้มีกระบวนการที่ดี ที่สามารถผลิตซอฟต์แวร์ที่มีคุณภาพที่ดียิ่งขึ้น
- เพื่อรับประกันว่าผลิตภัณฑ์ซอฟต์แวร์และกระบวนการผลิตตลอดช่วงชีวิตของโครงการเป็นไปตามความต้องการ และตามมาตรฐานต่าง ๆ ที่ได้กำหนดไว้

QA VS QC

- Quality Control (QC) หมายถึงกระบวนการ หรือวิธีการที่ใช้สำหรับ
 - การเปรียบเทียบคุณภาพของผลิตภัณฑ์ที่ผลิตออกมากับข้อกำหนดความต้องการ (requirements) และมาตรฐานต่างๆที่ใช้ในการผลิตซอฟต์แวร์
 - การดำเนินการอย่างใดอย่างหนึ่งหากพบสิ่งที่ไม่เป็นไปตามข้อกำหนดหรือมาตรฐาน
- การควบคุมคุณภาพจะใช้วิธีการทบทวน (review) และทดสอบ (testing) เพื่อตรวจสอบ ค้นหาและแก้ไขข้อผิดพลาดก่อนที่จะส่งมอบผลิตภัณฑ์ การควบคุมคุณภาพ ควรเป็นหน้าที่รับผิดชอบของหน่วยผลิตซอฟต์แวร์และควรรวมเข้าเป็นหนึ่งเดียวกับกิจกรรมในการผลิตซอฟต์แวร์ ซึ่งอาจดำเนินการโดยทีมงานที่ผลิตซอฟต์แวร์เอง หรือจัดตั้งทีมงานเฉพาะที่ทำหน้าที่ในการควบคุมคุณภาพของผลิตภัณฑ์

QA VS QC

- Quality Assurance (QA)

- เป็นการดำเนินกิจกรรมต่าง ๆ เพื่อให้มีกระบวนการพัฒนาซอฟต์แวร์ที่ดีและมีการปรับปรุงกระบวนการอย่างต่อเนื่อง เพื่อให้องค์กรสามารถใช้กระบวนการดังกล่าวในการผลิตและส่งมอบผลิตภัณฑ์ (ซอฟต์แวร์และบริการ)
- การประกันคุณภาพเป็นการป้องกันการเกิดปัญหา
- ให้คำแนะนำและแนวทางที่ถูกต้องในเวลาที่เหมาะสม
- สร้างความเข้าใจเกี่ยวกับแนวคิดด้านคุณภาพภายในองค์กรและสร้างทัศนคติที่ดีด้านคุณภาพ
- ทำให้บุคลากรทั้งในระดับบริหารและระดับปฏิบัติเข้าใจและเห็นถึงประโยชน์ของคุณภาพ
- ทำให้แนวคิดด้านคุณภาพนั้นกลายเป็นวินัยและเป็นส่วนหนึ่งของการทำงาน

โดยสรุปหน้าที่ของ QA คือ

- ทำให้เกิดกระบวนการพัฒนาซอฟต์แวร์ กระบวนการควบคุมคุณภาพ กระบวนการบำรุงรักษาและกระบวนการอื่น ๆ ที่เกี่ยวข้องกับการผลิตซอฟต์แวร์ที่ดีและเหมาะสมสำหรับองค์กร
- ประเมินผลกระบวนการที่กำหนดขึ้น เพื่อค้นหาจุดอ่อนและปรับปรุงกระบวนการอย่างต่อเนื่อง

QC

- | | |
|--|---|
| <ul style="list-style-type: none">• ทำงานเฉพาะเจาะจงลงไปแต่ละผลิตภัณฑ์ (ซอฟต์แวร์และบริการ)• ตรวจสอบว่าคุณสมบัติต่าง ๆ ของซอฟต์แวร์และบริการที่ควรจะมีนั้น มีครบหรือไม่ | <ul style="list-style-type: none">• การค้นหาข้อบกพร่องของซอฟต์แวร์และบริการ มีวัตถุประสงค์หลักคือเพื่อแก้ไขข้อบกพร่องเหล่านั้น• การควบคุมคุณภาพเป็นหน้าที่รับผิดชอบของผู้ปฏิบัติ |
|--|---|

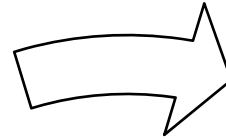
QA

- | | |
|--|---|
| <ul style="list-style-type: none">• ทำหน้าที่ในการจัดตั้งหรือทำให้มีกระบวนการ• จัดทำวิธีการประเมินผลกระบวนการ• ทำการระบุจุดอ่อนของกระบวนการเพื่อปรับปรุง• เป็นความรับผิดชอบในระดับบริหาร แต่ส่วนใหญ่ดำเนินการโดยเจ้าหน้าที่ที่ทำหน้าที่เฉพาะด้านประกันคุณภาพ) | <ul style="list-style-type: none">• ประเมินผลกระบวนการควบคุมคุณภาพ โดยมีวัตถุประสงค์เพื่อค้นหาจุดอ่อนของกระบวนการเพื่อการปรับปรุง• ให้ความสำคัญกับทุกผลิตภัณฑ์ที่ผลิตโดยกระบวนการ• ผู้ที่ทำหน้าที่ด้านการประกันคุณภาพไม่ควรทำหน้าที่ควบคุมคุณภาพ นอกจากการทำเพื่อตรวจสอบและประเมินกระบวนการควบคุมคุณภาพ |
|--|---|

Enable Process Improvement

- Process Improvement

Act



Plan

Enable Process Definition

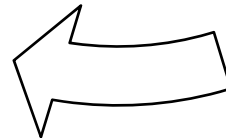
- Process Inventory
- Process Mapping
- Process Planning



Do

Enable Process Execution

- Process Definition
- Process Control



Check

Enable Process Assessment

- Process Measurement



การประกันคุณภาพในการจ้างพัฒนาซอฟต์แวร์



Copyright © 2004 Stephen E. Giden

เราสามารถซื้อหรือจ้างพัฒนาระบบอะไรก็ได้ แต่สิ่งที่เราไม่สามารถซื้อหรือจ้างได้คือ

“ความรับผิดชอบต่อคุณภาพซอฟต์แวร์”

ปัญหาที่พบในการซื้อหรือจ้างพัฒนาซอฟต์แวร์

- ไม่มีการกำหนดปัจจัยด้านคุณภาพที่สามารถวัดได้ เช่น ปัจจัยด้านความน่าเชื่อถือของระบบ หรือปัจจัยด้านความง่ายในการใช้งาน ทำให้ซอฟต์แวร์ที่รับมอบมา ไม่สามารถทำงานได้ตามความต้องการของหน่วยใช้
- ไม่มีหลักเกณฑ์ที่ชัดเจนที่สามารถใช้ในการทดสอบซอฟต์แวร์ตามข้อกำหนดความต้องการ (requirement) ทำให้ซอฟต์แวร์ที่รับมอบนั้น ทำงานไม่ได้ตามที่หน่วยใช้คาดหวัง
- ไม่มีการกำหนดมาตรฐานด้านการปฏิบัติงานและด้านเอกสาร อาจทำให้ผู้พัฒนาซอฟต์แวร์ออกแบบซอฟต์แวร์ให้มีการใช้งานที่ซับซ้อนและยากเกินกว่าที่หน่วยใช้จะสามารถใช้งานได้
- จัดทำข้อกำหนดความต้องการ (requirement) ไม่ครบ เนื่องจากขาดการวิเคราะห์ความต้องการในรายละเอียด ในระหว่างที่ผู้พัฒนาทำการพัฒนาซอฟต์แวร์ ผู้ว่าจ้างอาจจะพบว่าความต้องการไม่ครบถ้วน แต่การจะทำให้ซอฟต์แวร์ทำงานได้ตามความต้องการที่ครบถ้วนนั้น อาจต้องมีค่าใช้จ่ายเพิ่มเติมในจำนวนที่สูงมาก

ปัญหาที่พบในการซื้อหรือจ้างพัฒนาซอฟต์แวร์

- ประเด็นด้านเทคโนโลยี ในกรณีที่เทคโนโลยีที่ระบุให้ใช้ในสัญญามีการเปลี่ยนแปลงไป ผู้รับจ้างอาจมีค่าใช้จ่ายเพิ่มเติมที่สูงมากในการปรับปรุงซอฟต์แวร์เพื่อให้รองรับเทคโนโลยีใหม่นั้น
- การฝึกอบรมและการนำซอฟต์แวร์ไปใช้งานจริงมีความยากและซับซ้อน เนื่องจากจากผู้รับจ้างพัฒนาซอฟต์แวร์เป็นผู้ที่มาจากหน่วยงานภายนอก อาจไม่มีความเข้าใจภูมิหลังของผู้เข้าฝึกอบรม ทำให้ไม่สามารถให้การฝึกอบรมที่เหมาะสมแก่ผู้ใช้ได้
- ความแตกต่างทางด้านวัฒนธรรม และกฎระเบียบของแต่ละประเทศ เช่น การใช้สิทธิวันลา วันหยุด
- ปัญหาด้านการสื่อสาร อาจทำให้การตีความหรือการทำความเข้าใจข้อกำหนดความต้องการ ไม่ตรงหรือไม่สอดคล้องกัน
- ผู้ว่าจ้างและผู้รับจ้างอยู่ในเขตเส้นแบ่งเวลา (time zone) ที่แตกต่างกันมาก ทำให้การสื่อสารปัญหาที่พบในการใช้ซอฟต์แวร์ไม่สามารถทำได้อย่างทันท่วงที

แนวทางการประกันคุณภาพในการจ้างพัฒนาซอฟต์แวร์



การคัดเลือกผู้รับจ้าง

- บุคลากรที่เกี่ยวข้องในการคัดเลือกผู้รับจ้าง — ผู้ควบคุมคุณภาพ, นักวิเคราะห์ระบบ, ผู้ใช้ระบบ, หน่วยจัดหา , ที่ปรึกษากฎหมาย
- การศึกษาความเป็นไปได้ — ต้องระบุคุณลักษณะของซอฟต์แวร์ให้ชัดเจน
- การกำหนดเกณฑ์ในการจัดจ้าง
 - กำหนดหลักเกณฑ์ในการตรวจรับที่สามารถทำการทดสอบได้
 - ผู้รับจ้างมีกระบวนการผลิตซอฟต์แวร์ที่ดี
 - ผู้รับจ้างมีกระบวนการทดสอบที่มีประสิทธิภาพ
 - ผู้รับจ้างมีการส่งรายงานความก้าวหน้าสม่ำเสมอ
 - มีการถ่ายทอดความรู้และเทคโนโลยี

การใช้งาน

- การตรวจรับ
 - ตรวจสอบให้แน่ใจว่าสิ่งที่ส่งมอบตรงตามข้อกำหนดความต้องการที่กำหนดในสัญญา
 - ส่งมอบตรงเวลา
 - มีข้อมูลสำหรับทดสอบที่เพียงพอ เพื่อให้แน่ใจว่าการทดสอบเพื่อตรวจรับสามารถดำเนินการได้อย่างครบถ้วนเพียงพอ
 - มีการเตรียมการสำหรับการตรวจรับ
 - มีการฝึกอบรมก่อนดำเนินการตรวจรับ
 - ดำเนินการตรวจรับตามวันที่กำหนดในสัญญา
 - มีการทดสอบขั้นตอนในการนำระบบขึ้นใช้งานจริง ในระหว่างการตรวจรับ
 - ฯลฯ

การใช้งาน

- การใช้งานซอฟต์แวร์
 - มีการควบคุมที่ดีเพียงพอ
 - มีเอกสารที่ดีเพียงพอ
 - ความรวดเร็วในการให้บริการ
 - ศักยภาพในการให้บริการ
 - ความพร้อมของอุปกรณ์หรือฮาร์ดแวร์
 - บุคลากรของผู้ว่าจ้าง ควรได้รับการอบรมให้มีทักษะเพียงพอในการใช้งานซอฟต์แวร์
 - การจัดทำเอกสารเพื่อบันทึกข้อผิดพลาดของซอฟต์แวร์
 - ฯลฯ

เอกสารต้นแบบ (Template)

- แผนการบริหารโครงการ (Software Project Management Plan)
- แผนการประกันคุณภาพซอฟต์แวร์ (Software Quality Assurance Plan)
- การจัดการประกอบของโครงการ (Software Configuration Management)
- ข้อกำหนดความต้องการระบบ (Software Requirement Specification)
- ข้อกำหนดการออกแบบระบบ (Software Design Specification)
- เมทริกซ์การติดตาม (Requirement Traceability Matrix)
- แผนการติดตั้งและทดสอบระบบ (Installation and Test Plan)
- แบบฟอร์มรายงานข้อผิดพลาด (Defect Report Form)
- แบบฟอร์มร้องขอการเปลี่ยนแปลง (Change request Form)
- รายงานการปิดโครงการ (Project Closure Report)
- แผนการนำระบบขึ้นใช้งานจริง (Production Plan)
- แผนการบำรุงรักษาระบบ (Maintenance Plan)

Q&A

สอบถามเพิ่มเติมได้ที่ ระพีพรรณ ถาวรวันชัย

E-Mail : rapeepan@qualitynow.co.th