

รายงานผลการเข้ารับการอบรมหลักสูตร
Change Management in Software Project Workshop

จัดโดย

เขตอุตสาหกรรมซอฟต์แวร์ประเทศไทย
(Software Park Thailand)
ถนนแจ้งวัฒนะ อำเภอปากเกร็ด จังหวัดนนทบุรี

วันที่ 20-21 ธันวาคม พ.ศ.2555

ผู้จัดทำ

รองศาสตราจารย์ทัศนีย์วรรณ ศรีประดิษฐ์

ได้รับทุนสนับสนุนจากกองทุนพัฒนาบุคลากร
ประจำปีงบประมาณ 2556

รายงานการไปฝึกอบรม ศึกษาดูงาน ประชุม / สัมมนา
ตามระเบียบมหาวิทยาลัยสุโขทัยธรรมาธิราช ว่าด้วยการให้ทุนฝึกอบรม ศึกษาดูงาน
และประชุมทางวิชาการแก่ข้าราชการมหาวิทยาลัยสุโขทัยธรรมาธิราช

1. ผู้เข้ารับการฝึกอบรม

ชื่อ	นางสาวทัศนีย์วรรณ ศรีประดิษฐ์	อายุ	49 ปี
ตำแหน่ง	รองศาสตราจารย์	ระดับ	9
สังกัด	สาขาวิชาวิทยาศาสตร์และเทคโนโลยี	โทร.	8290

ไปฝึกอบรมหลักสูตร Change Management in Software Project Workshop

ณ เขตอุตสาหกรรมซอฟต์แวร์ประเทศไทย ถนนแจ้งวัฒนะ อำเภอปากเกร็ด จังหวัดนนทบุรี ประเทศไทย
ตั้งแต่วันที่ 20 ธันวาคม 2555 ถึงวันที่ 21 ธันวาคม 2555 รวมระยะเวลา 2 วัน

2. รายละเอียดเกี่ยวกับการฝึกอบรม

2.1 วิธีการฝึกอบรม

เป็นการบรรยายแนวคิด/ทฤษฎี โดยอิงตามหนังสือที่ใช้ประกอบการฝึกอบรม และทำ Workshop ท้ายบทเรียนทุกบท นอกจากนี้วิทยากรได้แบ่งปันประสบการณ์และเทคนิคเพื่อลดการเปลี่ยนแปลงความต้องการในทุกกระบวนการของโครงการ ซึ่งเป็นเทคนิคที่ใช้ได้ผล และไม่สามารถหาได้ในตำราใด

2.2 สาระสำคัญของการฝึกอบรม

วิทยากร :



คุณวิสุทธิ ลือชัยเฉลิมสุข

Organization :	VL Business Consultant Co., Ltd.
Qualifications :	2002-ปัจจุบัน : Managing Director บจก.วีแอลบิสซิเนสคอนซัลแตนท์, บจก.ซอฟต์แวร์คอนสตรัคชั่น 2000-2002 : Strategic Business Unit Manager บจก.ซอฟต์แวร์อินเทอร์เน็ตเซ็นแนล 1997-1999 : Application Manager บจก.ซอฟต์แวร์อินเทอร์เน็ตเซ็นแนล 1995-2002 : Board of Director กลุ่มบริษัทซอฟต์แวร์ 1994-1996 : System Analyst บจก.ซอฟต์แวร์1999 1992-1993 : Programmer Analyst บจก.ซอฟต์แวร์1999 1989-1991 : Programmer บจก.ซอฟต์แวร์1999
Education :	MSc (Information Technology), KMUTT BBA (Operation Research), Chulalongkorn University
Experiences :	Work Experience : CMM Standard Consultant CMM Project Leader IT Project Consultant System Analysis and Design Consultant Marketing and Selling Software Process and Productivity Management IT Project and Team Management Set up standard of software (process, project, product) System Analysis and Design Software Development System Implementation CMM Consultative Experiences : Professional Computer Co.,Ltd. Soft Venture Co.,Ltd.

วัตถุประสงค์ :

- 1) เพื่อทราบและตระหนักถึงสาเหตุที่ทำให้เกิดการเปลี่ยนแปลงความต้องการในโครงการพัฒนาซอฟต์แวร์
- 2) เพื่อให้สามารถรับมือกับการเปลี่ยนแปลงความต้องการได้อย่างมีประสิทธิภาพ
- 3) เพื่อลดความล่าช้าของโครงการที่มีสาเหตุมาจากการเปลี่ยนแปลงความต้องการ
- 4) เพื่อส่งมอบงานที่มีคุณภาพให้แก่ลูกค้า
- 5) เพื่อสร้างความพึงพอใจให้แก่ลูกค้าและทีมงานโครงการ

รายละเอียดของหลักสูตร :

ในหลักสูตรนี้ผู้เข้ารับการอบรมจะได้เรียนรู้ทั้งศาสตร์และศิลป์ ในการจัดการการเปลี่ยนแปลงความต้องการในโครงการ โดยตลอดหลักสูตร วิทยากรได้มุ่งเน้นให้ผู้เข้ารับการอบรมเข้าใจในทุกขั้นตอนและตัวอย่างเอกสารตั้งแต่เริ่มเกิดการเปลี่ยนแปลง ไปจนถึงการส่งมอบงานที่เกิดจากการเปลี่ยนแปลง พร้อมกับการทำ Workshop เพื่อให้เกิดความเข้าใจอย่างแท้จริง

หลักสูตรนี้มีทั้งหมด 11 เรื่อง ดังนี้

- 1) การวัดคุณภาพของความต้องการด้านซอฟต์แวร์
(quality measures of software requirement)
- 2) สาเหตุที่ทำให้เกิดการเปลี่ยนแปลงความต้องการ
(the cause of requirement change)
- 3) ทำความเข้าใจกับ work product, product, configuration item
(understanding the work product, product, configuration item)
- 4) การเลือกรายการให้เป็น configuration item
(selection item to be configuration item)
- 5) ตารางการติดตามความต้องการ
(requirement traceability matrix)
- 6) กระบวนการควบคุมการเปลี่ยนแปลง
(change control process)
- 7) การประเมินขอบเขตของผลกระทบที่เกิดจากการเปลี่ยนแปลง
(assessing the scope of impact from change)
- 8) กระบวนการควบคุมโครงร่างภายนอก
(configuration control process)
- 9) การจัดการโครงร่างภายนอกของซอฟต์แวร์และการควบคุมรุ่นของชิ้นงาน
(managing software configuration and control version for work product)
- 10) เทคนิคในการลดการเปลี่ยนแปลงในโครงการพัฒนาซอฟต์แวร์
(technique to decrease change in software project)
- 11) เทคนิคการเจรจาต่อรองเมื่อมีการเปลี่ยนแปลงเกิดขึ้น
(negotiation techniques when change occurred)

เรื่องที่ 1 การวัดคุณภาพของความต้องการด้านซอฟต์แวร์

มีตัววัด 9 ตัว ได้แก่

1) *ความถูกต้อง (correct)* ไม่มีเครื่องมือใดๆ ที่จะช่วยรับประกันได้ว่าความต้องการนั้นถูกต้อง นอกเสียจากจะได้รับการตรวจสอบด้วยการทบทวนและยอมรับโดยลูกค้าหรือผู้ใช้นั้น

2) *ไม่กำกวม (unambiguous)* ความต้องการต้องมีขอบเขตที่ชัดเจน และตีความได้ความหมายเดียวเท่านั้น

3) *ความสมบูรณ์ (completeness)* ต้องมีเอกสารอ้างอิงที่ชัดเจน และสามารถระบุเงื่อนไขที่ใช้ในการตรวจรับงานได้

4) *ความสอดคล้อง (consistency)* ต้องได้รับการทบทวนจากผู้พัฒนาระบบว่าไม่ขัดแย้งกับงานในส่วนอื่นๆ เช่น ความต้องการของฝ่ายคลังสินค้าอาจขัดแย้งกับความต้องการของฝ่ายบัญชี เป็นต้น

5) *สามารถจัดลำดับความสำคัญและ/หรือความเสถียรได้ (ranked for important and/or stability)* เมื่อจัดลำดับความสำคัญได้แล้ว จะได้เลือกทำความต้องการที่มีความสำคัญและเสถียรมากกว่าก่อน

6) *สามารถทดสอบได้ (verifiable)* โดยพิจารณาว่า คุ่มค่าหรือมีผลกระทบกับต้นทุนที่ใช้ในการทดสอบหรือไม่ เช่น ความต้องการที่ระบุว่า ผู้ใช้ต้องสามารถใช้งานได้พร้อมๆ กัน 500 คน จะทดสอบได้อย่างไร หากต้องใช้เครื่องมือช่วยทดสอบจะคุ้มค่าหรือไม่ เป็นต้น

7) *เอื้อให้เกิดการแก้ไขได้ง่าย (modifiable)* ถ้าจำเป็นต้องปรับแก้ไขก็สามารถทำได้ง่าย

8) *สอบทานไประหว่างกันได้ (traceable)* เช่น ระบุว่า หน้าจอที่พัฒนาเพิ่มขึ้นเพื่อตอบสนองความต้องการใด เป็นต้น มีการระบุหมายเลขเพื่อใช้อ้างอิงได้สะดวก

9) *เข้าใจได้ (understandable)* ทั้งผู้ใช้และผู้พัฒนาระบบมีความเข้าใจในความต้องการที่ตรงกัน

เรื่องที่ 2 สาเหตุที่ทำให้เกิดการเปลี่ยนแปลงความต้องการ

ปัจจัยที่ทำให้เกิดการเปลี่ยนแปลง มาจาก 2 แหล่ง ได้แก่

1) *ปัจจัยภายใน (internal factors)* เช่น ส่งพนักงานที่ไม่เหมาะสมไปเก็บข้อมูล ไม่มีการจัดเตรียมคำถามก่อนที่จะไปสอบถามความต้องการจากผู้ใช้ สอบถามข้อมูลผิดคน ไม่ออกแบบระบบด้วยปาก (ข้อมูลที่เก็บรวบรวมมาได้ต้องมีการจดบันทึกเป็นลายลักษณ์อักษร) เป็นต้น

2) *ปัจจัยภายนอก (external factors)* เช่น มีการเปลี่ยนแปลงปัญหาทำให้ต้องคิดในเรื่องใหม่ ผู้ใช้ระบบเปลี่ยนใจ สภาพแวดล้อมภายนอกเปลี่ยนแปลง (เช่น คู่แข่งขัน) ทำให้ต้องปรับเปลี่ยนความต้องการใหม่ หรืออาจเป็นระบบที่ไม่เคยมีมาก่อนทำให้การคิดพัฒนาไม่ครอบคลุม เป็นต้น

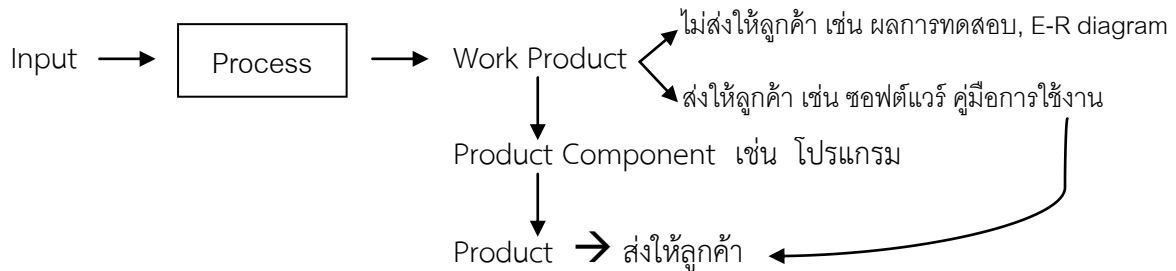
แหล่งที่มาของการเปลี่ยนแปลงความต้องการ เกิดได้จาก 2 แหล่ง ได้แก่

1) *เกิดจากลูกค้า (official source)* เช่น ลูกค้าขอเสนอความต้องการใหม่ ซึ่งเปลี่ยนแปลงไปจากเดิมโดยสิ้นเชิง เป็นต้น

2) *เกิดจากผู้พัฒนา (unofficial source)* เช่น มีทีมงานบางคนไปตกลงรับปากกับลูกค้าไว้ว่าจะดำเนินการให้ โดยที่ผู้บริหารโครงการไม่เคยรับรู้มาก่อน และไม่ได้แจ้งให้บริษัททราบ เป็นต้น กรณีเช่นนี้หากมีจำนวนมากอาจส่งผลกระทบต่อขอบเขตของโครงการได้

เรื่องที่ 3 ทำความเข้าใจกับ work product, product, configuration item

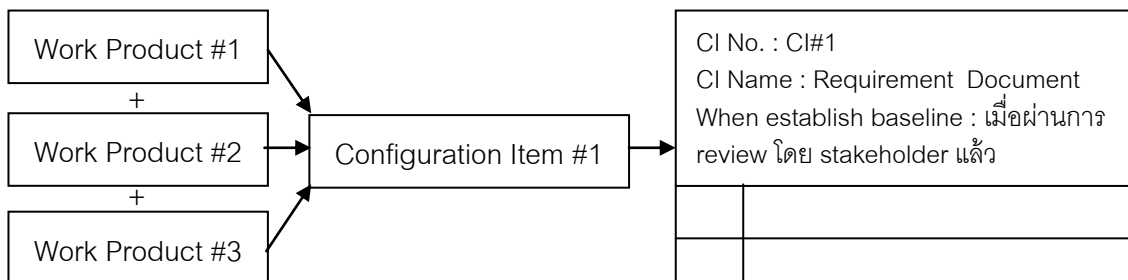
work product คือ ชิ้นงานที่เกิดจากกระบวนการหนึ่ง ซึ่งสามารถนำไปใช้ประโยชน์ได้ เช่น ไฟล์ เอกสาร สินค้า ชิ้นส่วนผลิตภัณฑ์ บริการ หรือใบแจ้งหนี้ เป็นต้น



product component เป็น work product ที่เป็น component ระดับล่างสุดของ product เป็นส่วนหนึ่งที่ใช้ผลิต product

product เป็น work product ที่มีจุดมุ่งหมายเพื่อจัดส่งให้กับลูกค้าหรือผู้ใช้อย่างปลายทาง

configuration item (CI) เป็นการรวบรวม work product ที่ได้ถูกกำหนดไว้แล้ว เพื่อใช้เป็น baseline ณ เวลาใดเวลาหนึ่ง สามารถนำมาใช้เพื่อการเปรียบเทียบหรือต่อยอดได้ เช่น ถ้าระบบ 1 ระบบ ประกอบด้วยหลายๆ โปรแกรม ดังนั้น CI อาจประกอบด้วย โปรแกรมแต่ละตัว + เอกสารที่เกี่ยวข้อง + ข้อมูล เป็นต้น



เรื่องที่ 4 การเลือกรายการให้เป็น configuration item

ลักษณะของ work product ที่จะถูกเลือกเป็น configuration item

- 1) work product ที่คาดว่าจะมีการเปลี่ยนแปลง
- 2) work product ที่อาจมีผู้ใช้งานมากกว่า 2 กลุ่มขึ้นไป
- 3) work product ที่ขึ้นอยู่กับ work product อื่น
- 4) work product ที่มีความสำคัญมาก
- 5) work product ที่อาจส่งผลกระทบต่อความมั่นคงปลอดภัย
- 6) work product ที่มีความซับซ้อนสูง หรือมีความต้องการด้านประสิทธิภาพสูง
- 7) เป็นรายการที่ได้รับการพัฒนาขึ้นมาใหม่หรือไม่
- 8) เกี่ยวข้องกับเทคโนโลยีใหม่หรือไม่
- 9) เชื่อมต่อกับซอฟต์แวร์อื่นหรือไม่
- 10) ติดตั้งอยู่ในแพลตฟอร์มที่แตกต่างจากส่วนอื่นๆ ของระบบหรือไม่
- 11) ต้องเชื่อมต่อกับ Configuration Item อื่นหรือไม่
- 12) มีแนวโน้มที่จะใช้ซ้ำ (reused) หรือไม่
- 13) จำเป็นต้องแจ้งสถานะของรุ่นให้ลูกค้าทราบหรือไม่

เรื่องที่ 5 ตารางการติดตามความต้องการ (traceability matrix)

คือ เอกสารที่ใช้แสดงความสัมพันธ์ระหว่าง 2 baseline documents โดยเป็นความสัมพันธ์แบบหลายต่อหลาย (many to many) เป็นเครื่องมือช่วยทบทวนว่าได้คิดหรือพิจารณาครบถ้วนทุกประเด็นแล้ว

ตัวอย่างของ Requirement Traceability Matrix เป็นดังนี้

Customer Request	Product Request	Product Component Request	Design No.	Program No.	Test Script No.
1. Aaaa 2. Bbbb	1.1 1.2 2.1 2.2	1.1.1 1.1.2 1.2.1			

Horizontal Traceability

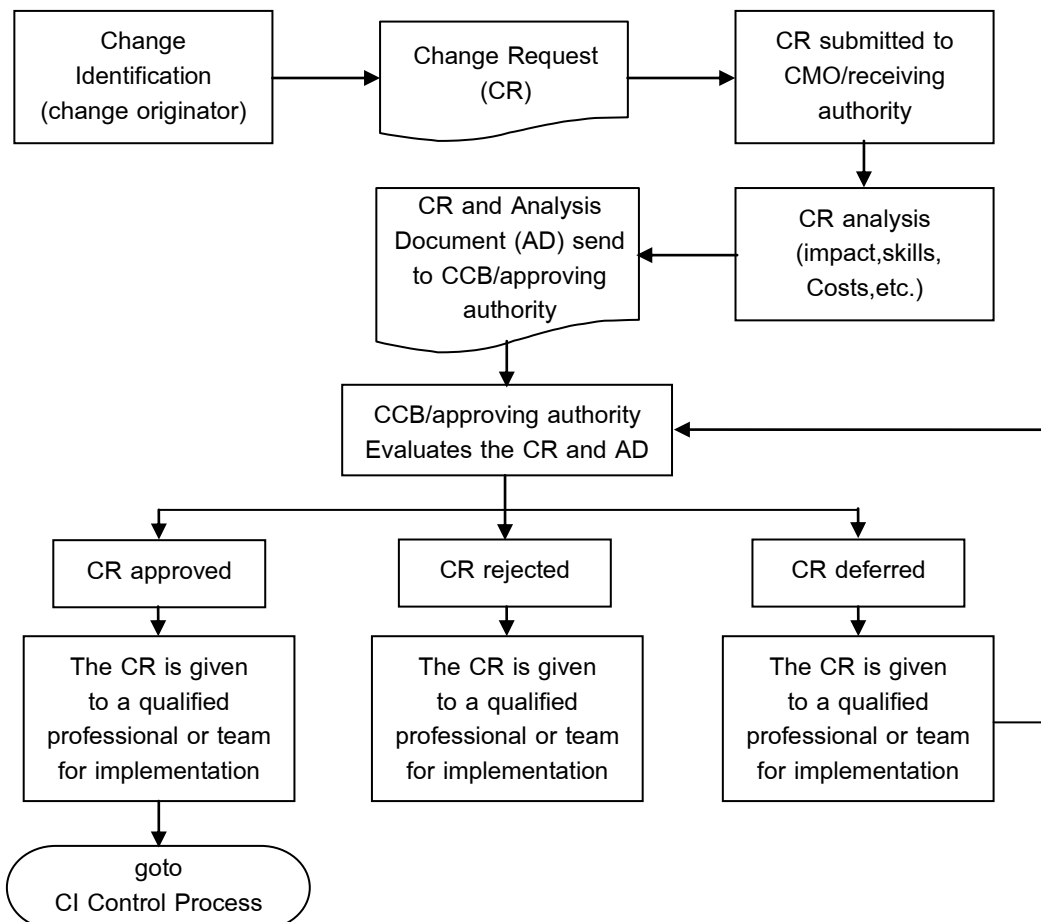
→ Forward traceability

← Backward traceability

↔ Vertical Traceability

เรื่องที่ 6 กระบวนการควบคุมการเปลี่ยนแปลง

แผนภาพแสดงกระบวนการจัดการและควบคุมการเปลี่ยนแปลง



กระบวนการควบคุมการเปลี่ยนแปลง มี 4 ขั้นตอน ได้แก่

- 1) *Change initiation* เมื่อเริ่มมีการเปลี่ยนแปลง ต้องระบุผู้ที่มีหน้าที่รับ CR อาจเป็นผู้บริหารโครงการ หรือ CMO หรือบุคคลใดในทีมงานที่ระบุไว้ จากนั้นต้องกำหนดหมายเลขงาน ระบุว่าเป็นการเปลี่ยนแปลงประเภทใด ได้แก่ ความต้องการใหม่ เปลี่ยนแปลงความต้องการ หรือแก้ปัญหาข้อบกพร่อง
- 2) *Change classification* จำแนกระดับของผลกระทบที่มีต่อการเปลี่ยนแปลง เช่น ระดับรุนแรงมาก ระดับสำคัญ ระดับส่งผลกระทบ ระดับที่เกี่ยวข้องกับต้นทุน เป็นต้น
- 3) *Change evaluation or Change analysis* ประเมิน/วิเคราะห์ผลกระทบ เช่น กระทบกับฟังก์ชันงาน ส่วนต่อประสานผู้ใช้ ต้นทุน ระยะเวลา ความเสถียรของระบบ หรือกระทบกับสัญญา
- 4) *Change disposition* ผู้มีอำนาจต้องพิจารณาตัดสินใจว่า จะอนุมัติการเปลี่ยนแปลงหรือไม่อนุมัติ หรือจะเก็บไว้ก่อนเอาไว้พิจารณาภายหลัง หากอนุมัติรายการเปลี่ยนแปลงดังกล่าวจะถูกส่งต่อไปยังผู้รับผิดชอบเพื่อดำเนินการเปลี่ยนแปลงงานให้ต่อไป

เรื่องที่ 7 การประเมินขอบเขตของผลกระทบที่เกิดจากการเปลี่ยนแปลง

จัดทำรายละเอียดการวิเคราะห์ผลกระทบที่เกิดจากการเปลี่ยนแปลง ดังนี้

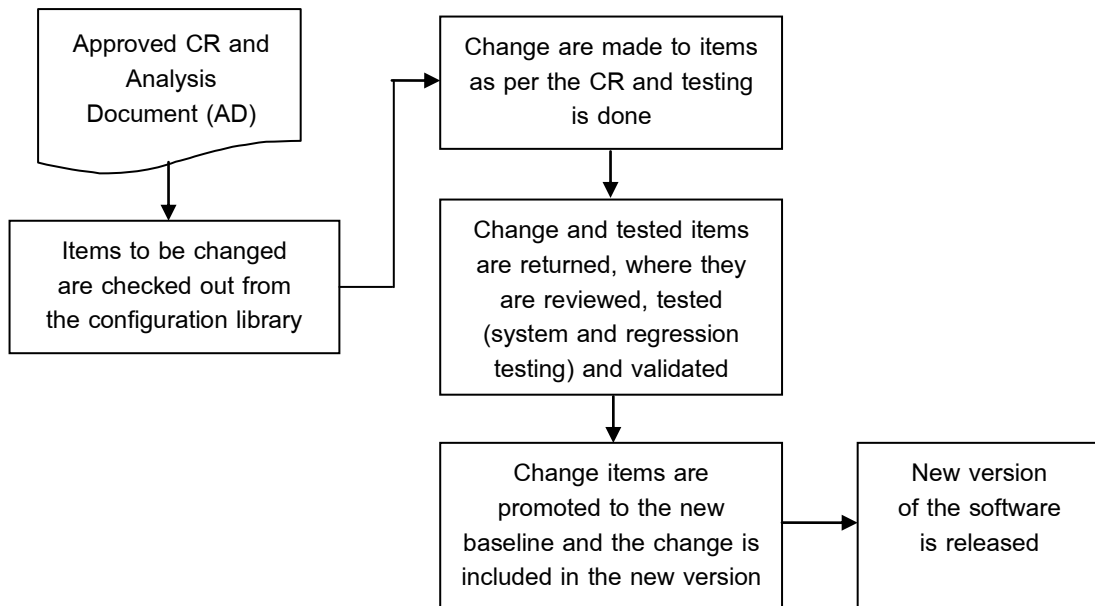
- 1) ระบุว่าการเปลี่ยนแปลงนี้ขัดแย้งกับความต้องการที่มีอยู่เดิมหรือไม่
- 2) ระบุว่าการต้องการใดต้องได้รับการปรับเปลี่ยนตามการเปลี่ยนแปลง
- 3) มีอะไรจะเกิดขึ้นหรือไม่ ถ้าไม่ดำเนินการเปลี่ยนแปลงตามความต้องการ
- 4) เมื่อปรับเปลี่ยนไปแล้ว ผู้ใช้จะรู้สึกว่าการใช้งานไม่เหมือนเดิมหรือยากขึ้นหรือไม่
- 5) การเปลี่ยนแปลงกระทบกับความมั่นคงปลอดภัยของระบบหรือไม่
- 6) ผู้พัฒนาระบบมีทักษะหรือความถนัดเพียงพอหรือไม่ เพราะอาจทำให้งานล่าช้าได้
- 7) ต้องใช้ทรัพยากรมากขึ้นหรือไม่
- 8) ต้องใช้เครื่องมือพิเศษในการทดสอบหรือใช้งานหรือไม่
- 9) การปรับเปลี่ยนมีผลกระทบกับลำดับการทำงานเดิมหรือไม่
- 10) ควรทำต้นแบบ (prototype) เสนอให้ลูกค้าพิจารณายืนยันก่อนลงมือเขียนโปรแกรม
- 11) เมื่อลูกค้าต้องการให้เพิ่มฟิลด์ ย่อมมีผลกระทบกับหน้าจอสำหรับเพิ่มและลบข้อมูล รวมทั้ง

หน้าจอแสดงผลและรายงานด้วย

- 12) ต้องดำเนินการแปลง (convert) ข้อมูลใหม่อีกหรือไม่
- 13) ต้องย้อนกลับไปปรับแก้ในส่วนของการออกแบบด้วยเสมอ
- 14) ต้องปรับกรณีที่ใช้ทดสอบ (test case) ด้วย
- 15) ระบุผลกระทบทั้งหมด โดยแสดงเป็นตัวเลข

เรื่องที่ 8 กระบวนการควบคุมโครงสร้างภายนอก

แผนภาพแสดงกระบวนการควบคุมโครงสร้างภายนอก (configuration control process)



กระบวนการควบคุมโครงสร้างภายนอก มีขั้นตอนดังนี้

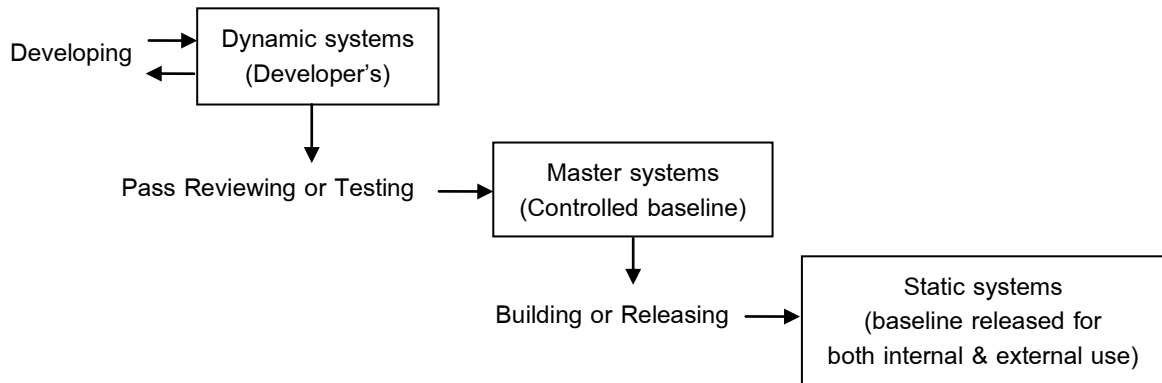
- 1) *ขั้นเริ่มต้นทำการเปลี่ยนแปลง (change implementation)* โดยกำหนดวันที่ต้องทำการเปลี่ยนแปลงให้เสร็จ ทำการคัดลอกไฟล์ที่เกี่ยวข้องออกมาจาก configuration library แล้วจึงทำการปรับแก้ไข
- 2) *ขั้นตรวจทาน (change verification)* เมื่อปรับแก้ไขเรียบร้อยแล้ว ต้องทำการทดสอบใหม่ ซึ่งผลการทดสอบต้องไม่ไปกระทบกับระบบงานเดิมที่เคยทำงานได้ถูกต้องอยู่แล้ว เมื่อดำเนินการเรียบร้อยแล้วให้ทำการบันทึกลงในแฟ้มประวัติของรายการเปลี่ยนแปลง (history change file)
- 3) *ขั้นกำหนดให้เป็น baseline (baseline change control)* เพื่อลดจำนวนเวอร์ชันของรายการเปลี่ยนแปลง จึงควรรวมรายการให้เป็นกลุ่มแล้วจึงสร้างกำหนดเป็น release ใหม่

เรื่องที่ 9 การจัดการโครงสร้างภายนอกของซอฟต์แวร์และการควบคุมรุ่นของชิ้นงาน

ดำเนินการดังนี้

- 1) ระบุว่ารายการไหนบ้างที่เป็น configuration item
- 2) กำหนดกลไกการจัดการควบคุมแบบหลายระดับ โดยที่
 - ระดับการควบคุมขึ้นอยู่กับ วัตถุประสงค์ของโครงการ ความเสี่ยง และทรัพยากรโครงการ
 - ระดับการควบคุมอาจแตกต่างกันขึ้นอยู่กับ ประเภทของระบบที่พัฒนา ความต้องการเฉพาะ
 - ตัวอย่างระดับการควบคุม เช่น ใครก็สามารถแก้ไขได้ (uncontrolled) มีการควบคุมการเปลี่ยนแปลง (work-in-progress) ดำเนินการเสร็จแล้ว (released)
- 3) ควบคุมการเข้าถึงเพื่อให้มั่นใจว่าผู้มีสิทธิที่แท้จริงเท่านั้นจึงจะเข้าถึงระบบได้ เช่น กำหนดว่าใครที่อ่านได้ ใครที่แก้ไขได้ ใครที่ลบทิ้งได้ ใครบ้างที่ไม่อนุญาตให้เห็น

4) การจัดเก็บและการเรียกใช้ configuration item ในระบบ configuration management มี 3 ระยะ ได้แก่ ระยะที่ยังเปลี่ยนแปลงตลอดเวลา (dynamic system) ระยะที่เป็นปัจจุบัน (master system) และระยะที่นิ่งแล้ว (static system) ทั้ง 3 ระยะเกี่ยวข้องกันดังแสดงในภาพ



5) แบ่งปันและโอนใช้งาน configuration item ระหว่างระดับการควบคุมภายในระบบ configuration management

- 6) จัดเก็บและกู้คืน configuration item ให้ถูกรุ่น (version)
- 7) จัดเก็บ ปรับปรุง และเรียกใช้งาน configuration item ให้ถูกรุ่น
- 8) จัดทำรายงานการเปลี่ยนแปลงที่เกิดขึ้นใน configuration management
- 9) บำรุงรักษารายการข้อมูลในระบบ configuration management เช่น ทำการสำรองข้อมูลในระบบ configuration management และกู้คืนข้อมูลหากมีข้อผิดพลาดเกิดขึ้น เป็นต้น
- 10) ปรับโครงสร้างของ configuration management เป็นระยะอย่างสม่ำเสมอ
- 11) มีกระบวนการตรวจสอบ (audit) ความครบถ้วนถูกต้องในกระบวนการปฏิบัติงาน

เรื่องที่ 10 เทคนิคในการลดการเปลี่ยนแปลงในโครงการพัฒนาซอฟต์แวร์

กลวิธีลดการเปลี่ยนแปลงในโครงการ มีดังนี้

- 1) ให้ทีมพัฒนาซอฟต์แวร์ร่วมสำรวจปัญหาและความต้องการในเบื้องต้น ตั้งแต่ก่อนเริ่มโครงการหรือปิดการขาย
- 2) สอบถามความต้องการให้ใหญ่ แต่เลือกกำหนดขอบเขตโครงการให้เล็ก
- 3) คิดค่าบริการตาม effort-based ไม่ใช่ project-based
- 4) ให้ลูกค้าเป็นผู้เขียนความต้องการของตนเอง (ไม่ใช่ความคาดหวัง)
- 5) ต้องมีการประชุมหารือร่วมกัน เพื่อทำความเข้าใจในความต้องการร่วมกัน
- 6) เพิ่ม commitment ของลูกค้าที่มีต่อความต้องการ โดยให้ลูกค้าเซ็นรับรองความต้องการ
- 7) ถ่ายทอดงานอย่างเป็นลายลักษณ์อักษรระหว่าง Project phase เช่น
 - ระหว่าง requirement phase กับ design phase
 - ระหว่าง design phase กับ development phase
 - ระหว่าง development phase กับ testing phase

8) ทำการออกแบบและพัฒนาโปรแกรมสำหรับความต้องการที่มีความเสถียรและมีความสำคัญสูงก่อน แล้วค่อยขยายไปยังความต้องการอื่น

9) ทำการออกแบบร่วมกันกับลูกค้า

10) ใช้การออกแบบที่ดี เพื่อช่วยลดผลกระทบที่จะเกิดขึ้นจากการเปลี่ยนแปลงความต้องการ เช่น (Object-Oriented Design: OOD)

11) สร้าง change control procedure ทั้งจากบุคคลภายนอกและภายในโครงการ ทำให้สามารถมองเห็นการเปลี่ยนแปลงที่เกิดขึ้นในโครงการ สามารถควบคุมและติดตามการเปลี่ยนแปลงที่เกิดขึ้นได้ง่าย

12) อย่าแก้ไขให้ทันทีที่ลูกค้าขอ เพราะหากต้องแก้ไขกลับจะทำได้ยาก

13) เพิ่มคุณภาพของการเปลี่ยนแปลงด้วยการให้ลูกค้าเป็นผู้เขียน change request

14) การประเมินผลกระทบจากการเปลี่ยนแปลง (impact analysis) มีความสำคัญอย่างมากต่อการลดจำนวนการเปลี่ยนแปลงในโครงการ

15) สะสมการเปลี่ยนแปลงไว้ระดับหนึ่ง แล้วค่อยทำรอบเดียว

16) แยก phase ของการเปลี่ยนแปลงออกจาก project phase หลังจากพัฒนางานหลักเรียบร้อยแล้ว

17) ทำให้ phase ต่างๆ ในโครงการมีระยะห่างกันน้อยที่สุด เพราะหากทอดเวลาห่างกันมากเท่าใด ยิ่งเพิ่มโอกาสของการเกิดการเปลี่ยนแปลงมากขึ้นเท่านั้น

18) ขอบิตโครงการที่กำลังดำเนินการก่อน แล้วจะทำงานที่ขอเพิ่มให้ภายหลัง หรือนับส่วนเพิ่มหรือแก้ไขเป็นโครงการใหม่

เรื่องที่ 11 เทคนิคการเจรจาต่อรองเมื่อมีการเปลี่ยนแปลงเกิดขึ้น

3 คำถาม ที่ควรถามตัวเองก่อนเริ่มการเจรจาต่อรอง

- 1) เราต้องการอะไรจากการเจรจาต่อรอง
- 2) ทำไมเขาจึงต้องเจรจาต่อรองกับเรา
- 3) เรามีทางเลือกอื่นหรือไม่

ลักษณะ 7 ประการ ของการเป็นนักเจรจาต่อรองที่ประสบความสำเร็จ

- 1) ต้องรู้ว่าเราต้องการอะไรและสื่อสารได้ดี
- 2) ต้องมั่นใจว่าเราควรได้รับสิ่งนั้น
- 3) ต้องมั่นใจว่าสามารถทำสิ่งนั้นได้
- 4) มีความกระตือรือร้นเกี่ยวกับสิ่งที่ต่อรอง
- 5) กล้าเผชิญหน้ากับความกลัว
- 6) เรียนรู้จากความล้มเหลว
- 7) ไม่ท้อถอยในการต่อรอง

คุณสมบัติของนักเจรจาต่อรองที่ดี

- เป็นผู้ฟังที่ดี
- สามารถเข้าถึงความต้องการของผู้อื่นได้เป็นอย่างดี
- สามารถประนีประนอม หรือหาทางออกเพื่อแก้ไขปัญหาที่เกิดขึ้นได้
- เชื่อมมั่นในแนวคิดของการชนะร่วมกัน (Win-Win)
- มีความเชี่ยวชาญในการจัดการกับความขัดแย้ง
- พร้อมทั้งจะหาข้อมูลและวิเคราะห์ประเด็นต่างๆ ที่เกี่ยวข้องอย่างครบถ้วน
- มีความอดทนต่อการยั่วและความก้าวร้าว
- มีความอดกลั้นสูงในการเผชิญกับความเครียด
- มีความหนักแน่น ไม่ตื่นเต้นตกใจง่ายกับการใช้กลยุทธ์ของฝ่ายตรงข้าม
- สามารถสรุปประเด็นหลักที่เกี่ยวข้องกับการเจรจาต่อรองได้อย่างรวดเร็ว

การวางแผนในการเจรจาต่อรอง

- 1) กำหนดประเด็นในการเจรจาต่อรอง เช่น ระยะเวลาโครงการ วันที่ส่งมอบ การชำระเงิน
- 2) จัดหมวดหมู่และกำหนดความสำคัญของแต่ละประเด็น
- 3) กำหนดสิ่งที่ต้องการ และเหตุผลของความต้องการ
- 4) ประเมินความต้องการของคู่เจรจา
- 5) ปรึกษากับผู้ที่เกี่ยวข้องก่อนเริ่มการเจรจา ทั้งฝ่ายเดียวกันและคู่เจรจา
- 6) ระบุข้อจำกัดของฝ่ายตน เช่น ค่าใช้จ่าย ระยะเวลา
- 7) กำหนดเป้าหมายในการเจรจาต่อรอง โดยกำหนดเป็น 4 จุด ได้แก่ จุดนำเสนอ

จุดเป้าหมายเฉพาะ จุดทางเลือก และจุดต่อต้าน

- 8) หาข้อมูลสนับสนุนการเจรจาต่อรอง เช่น ตัวเลข ภาพวาด กราฟ
- 9) สร้างแรงกดดันในการเจรจาต่อรอง ได้แก่ การยื่นกราน การแข่งขัน ความเชี่ยวชาญ

หลักกฎหมาย การมีส่วนร่วม และทัศนคติ

- 10) วิเคราะห์คู่เจรจา ว่าเป็นผู้มีอำนาจในการตัดสินใจ หรือเป็นเพียงตัวแทนมารับข้อเสนอข้อมูลทางการเงินต่างๆ

2.3 ผู้เข้าอบรม

จำนวน 8 คน ประกอบด้วย

ผู้ทำงานในบริษัทเอกชน

จำนวน 7 คน

ผู้ทำงานในหน่วยงานราชการ (มสธ.)

จำนวน 1 คน

2.4 สถานที่ฝึกอบรม

เขตอุตสาหกรรมซอฟต์แวร์ประเทศไทย ถนนแจ้งวัฒนะ อำเภอปากเกร็ด จังหวัดนนทบุรี ประเทศไทย



3. ประโยชน์ที่ได้รับ

3.1 ประโยชน์ที่ผู้รับทุนได้รับ

- 1) ได้พบปะแลกเปลี่ยนความคิดเห็นในหัวข้อการจัดการการเปลี่ยนแปลงในกระบวนการพัฒนาซอฟต์แวร์กับวิทยากรและผู้เข้าร่วมอบรม
- 2) ได้รับความรู้เพื่อนำมาเขียนหน่วยการสอนในเอกสารการสอนชุดวิชาการพัฒนาระบบสารสนเทศ
- 3) ได้ทราบความก้าวหน้า แนวโน้ม และทิศทางของการพัฒนาซอฟต์แวร์

3.2 ประโยชน์ที่มหาวิทยาลัยได้รับ

- 1) บุคลากรของมหาวิทยาลัยมีความรู้เกี่ยวกับการจัดการการเปลี่ยนแปลงความต้องการในกระบวนการพัฒนาซอฟต์แวร์
- 2) เอกสารการสอนชุดวิชาการพัฒนาระบบสารสนเทศมีเนื้อหาที่ทันสมัยและสามารถให้ความรู้กับนักศึกษาได้สอดคล้องกับสถานการณ์ที่แท้จริงในกระบวนการพัฒนาซอฟต์แวร์

4. ข้อเสนอแนะ

บุคลากรที่ทำงานเกี่ยวข้องกับเทคโนโลยีสารสนเทศและการสื่อสาร ควรเข้ารับการฝึกอบรมในหลักสูตรที่เกี่ยวข้องกับการปฏิบัติงานอย่างสม่ำเสมอ เนื่องจากเทคโนโลยีมีการเปลี่ยนแปลงไปอย่างรวดเร็ว