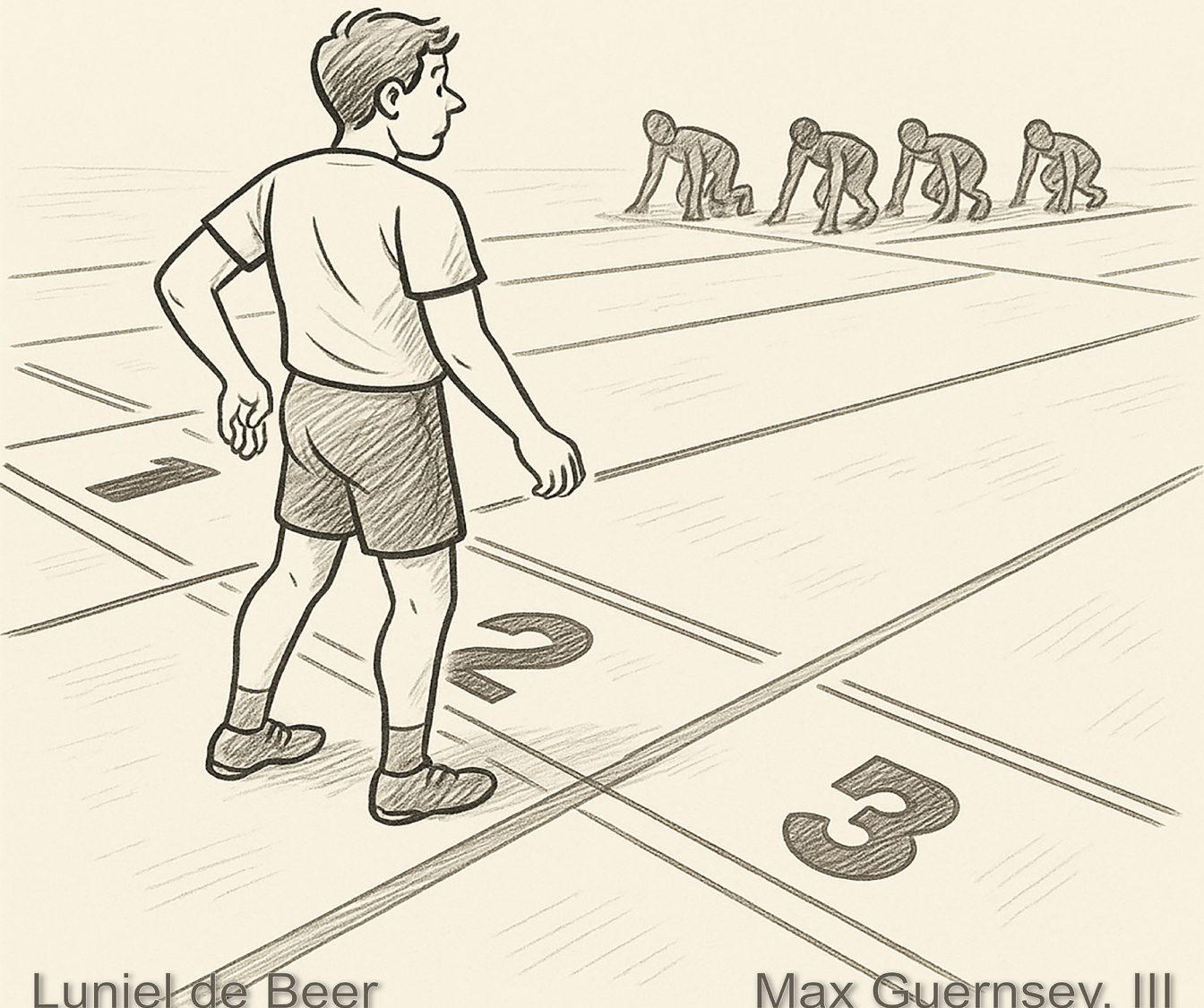


READY

WHY MOST SOFTWARE PROJECTS
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

ฉบับภาษาไทย

ทวิตหนังสือนี้!

โปรดช่วย Luniel de Beer และ Max Guernsey, III โดยการบอกต่อเกี่ยวกับหนังสือนี้บน [Twitter](#)!

ข้อความที่แนะนำให้ทวิตสำหรับหนังสือนี้คือ:

ผมเพิ่งซื้อหนังสือ Ready — หนังสือสำหรับผู้นำด้านซอฟต์แวร์และทีมที่ต้องการกำจัดการทำงานซ้ำ งานค้าง และความไม่สอดคล้องในการส่งมอบซอฟต์แวร์ [#CodeReady](#)

แฮชแท็กที่แนะนำให้ทวิตสำหรับหนังสือนี้คือ [#CodeReady](#)

ค้นหาว่าผู้อื่นพูดถึงหนังสือนี้อย่างไรโดยคลิกที่ลิงก์นี้เพื่อค้นหาแฮชแท็กนั้นบน Twitter:

[#CodeReady](#)

พร้อมแล้ว (ฉบับภาษาไทย)

ทำไมโครงการซอฟต์แวร์ส่วนใหญ่ล้มเหลว และวิธีแก้ไข

Luniel de Beer และ Max Guernsey, III

หนังสือเล่มนี้มีจำหน่ายที่ <https://leanpub.com/ready-th>

เวอร์ชันนี้เผยแพร่เมื่อ 2025-10-22



นี่คือหนังสือจาก [Leanpub](#) Leanpub ให้อำนาจแก่นักเขียนและสำนักพิมพ์ด้วยกระบวนการ Lean Publishing [Lean Publishing](#) คือการเผยแพร่หนังสืออิเล็กทรอนิกส์ที่อยู่ระหว่างดำเนินการโดยใช้เครื่องมือที่เรียบง่ายและการทำซ้ำหลายครั้งเพื่อรับข้อเสนอแนะจากผู้อ่าน ปรับเปลี่ยนจนกว่าคุณจะได้หนังสือที่ถูกต้อง และสร้างแรงจูงใจให้คุณทำสำเร็จ

© 2025 Luniel de Beer และ Max Guernsey, III

ด้วยความระลึกถึง Johann van Aardt ผู้ซึ่งเห็นความหลงใหลของผม แนะนำผมสู่การเขียนโปรแกรมจริง และช่วยให้ผมได้พบบ้านหลังใหม่ และขอบคุณพ่อแม่ของผม ผู้ซึ่งให้การสนับสนุนที่ไม่มีวันล้นคลอน—ตั้งแต่ลูกค้าคนแรกจนถึงบ้านหลังแรกในสหรัฐอเมริกา—ทำให้ทุกสิ่งนี้เป็นไปได้

—Luniel

เพื่อครอบครัวของผม ผู้เป็นดังพระอาทิตย์ขึ้นและตกของชีวิต

—Max

สารบัญ

เกี่ยวกับหนังสือเล่มนี้	i
หนังสือเล่มนี้เหมาะสำหรับใคร	ii
วิธีใช้หนังสือเล่มนี้	iii
เกี่ยวกับผู้เขียน	iv
คำนำ	v
 ส่วน I: บางสิ่งที่หายไป	 1
บทที่ 1: ปัญหาที่ซ่อนอยู่	2
บทที่ 2: ต้นทุนของการขาดรากฐาน	11
บทที่ 3: แนะนำกระบวนการพัฒนาความต้องการ (RMF)	20
บทที่ 4: มันเป็นแอ็ลจัจริงหรือ?	21
 ส่วน II: การสร้างพื้นที่เพื่อความพร้อม	 22
บทที่ 5: ส่วนขยายแรก	23
บทที่ 6: ทำไมคนจึงไม่ทำแบบนี้?	24
บทที่ 7: งานเตรียมความพร้อมอย่างชัดเจน (RMF 1)	26
บทที่ 8: ผลกระทบของ RMF 1	28
บทที่ 9: การนำ RMF 1 ไปปฏิบัติจริง	29

ส่วน III: การกำหนดเงื่อนไขความสำเร็จของงาน	31
บทที่ 10: ความต้องการถัดไป	32
บทที่ 11: สิ่งที่คุณมักทำกัน	34
บทที่ 12: การกำหนดคำนิยามของความสำเร็จ	36
บทที่ 13: นิยามของงานที่เสร็จสมบูรณ์แบบเฉพาะเจาะจง (RMF 2)	38
บทที่ 14: ชีวิตกับ RMF 1 & 2	41
บทที่ 15: การติดตั้ง RMF 2	42
ส่วน IV: การควบคุมการนำไปใช้งาน	43
บทที่ 16: ข้อกำหนดสุดท้าย	44
บทที่ 17: ภูมิหลังของนิยามความพร้อม	46
บทที่ 18: การกำหนดนิยามความพร้อม	47
บทที่ 19: นิยามความพร้อมเฉพาะกิจ (RMF 3)	48
ส่วน V: การสังเคราะห์	51
บทที่ 20: กำหนดเวลา_ส่วนใหญ่_ไม่สำคัญ	52
บทที่ 21: สมรรถนะที่ 1: กระบวนการพัฒนาความต้องการ	54
บทที่ 22: วิธีการไหลเวียนของงานและข้อมูลใน Scrum ด้วย RMF	56
บทที่ 23: ผลกระทบของ RMF	57
บทที่ 24: การเปลี่ยนผ่านไปสู่ RMF	58
บทที่ 25: ขึ้นอยู่กับคุณแล้ว	59
ส่วน VI: ทรัพยากร	60
ภาคผนวก ก: สกริมไม่ใช่ตัวปัญหา	61
ภาคผนวก B: เฟรมเวิร์กชินแนปส์™	63

ภาคผนวก C: ข้อโต้แย้งและอุปสรรคที่พบบ่อยเกี่ยวกับ RMF 1	65
ภาคผนวก D: รายการเกณฑ์เริ่มต้นสำหรับ DoD	66
ภาคผนวก E: รายการเกณฑ์เริ่มต้นสำหรับนิยามความพร้อม	67
ดัชนี	68

เกี่ยวกับหนังสือเล่มนี้

Ready เป็นหนังสือสำหรับทุกคนที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ที่เบื่อกับ การส่งมอบงานที่ต้อยกว่าเป้าหมาย การแก้ไขงานซ้ำเรื้อรัง และ ความต้องการที่ไม่ชัดเจน

คุณอาจเคยลองลงทุนในการพัฒนาทักษะการทำงานของทีม ปรับปรุงการนำกรอบการทำงานไปใช้ หรือปรับปรุงโค้ดแล้ว แต่ยังต้องการการพัฒนาเพิ่มเติม

นี่เป็นเพราะข้อจำกัดหลักสำหรับทีมพัฒนาซอฟต์แวร์ส่วนใหญ่ไม่ใช่ทักษะของทีม แต่เป็นความสมบูรณ์ของความต้องการ **แม้แต่ทีมที่มีประสบการณ์และมีทักษะที่เหมาะสมก็ยังคงประสบปัญหา** เมื่อต้องทำงานกับความต้องการที่ไม่สมบูรณ์

Ready แนะนำ RMF (การไหลเวียนของความสมบูรณ์ของความต้องการ) ซึ่งเป็นแนวทางที่ใช้งานได้จริงและมีโครงสร้างที่ลึกซึ้งในการจัดการให้ฝ่ายผลิตภัณฑ์และฝ่ายวิศวกรรมทำงานสอดคล้องกัน โดยไม่ต้องเปลี่ยนกระบวนการทำงานที่มีอยู่

ไม่ว่าคุณจะใช้ Scrum, Kanban หรือระบบอื่นๆ RMF จะช่วยให้คุณ **กำหนดขอบเขตงานให้มั่นคง กำหนดงานค้าง และ ส่งมอบสิ่งที่สำคัญจริงๆ**

หากทีมของคุณรู้สึกติดอยู่ที่ขอบของความ “เกือบเสร็จ” หนังสือเล่มนี้จะแสดงให้เห็นวิธี **ทำลายวงจร** และปลดปล่อยทีมของคุณอย่างถาวร

หนังสือเล่มนี้เหมาะสำหรับใคร

หนังสือเล่มนี้เหมาะสำหรับทุกคนที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์จริงๆ ตั้งแต่วิศวกรไปจนถึงผู้จัดการผลิตภัณฑ์ และจากผู้ใช้ปฏิบัติงานไปจนถึงผู้บริหาร

หนังสือเล่มนี้เหมาะสำหรับคุณถ้าคุณเกี่ยวข้องกับการพัฒนาซอฟต์แวร์และสังเกตเห็นว่าทีมที่คุณทำงานด้วยหรือทีมที่คุณอยู่มีปัญหาอย่างน้อยหนึ่งข้อต่อไปนี้:

- งานมักจะค้างต่อเนื่องจากรอบการทำงานหนึ่งไปยังรอบถัดไป
- ทีมพัฒนารู้สึกเหมือนกำลังพยายามยิงเป้าที่เคลื่อนที่
- งานถูกเปิดค้างไว้นานเกินไป
- งานถูกทำเครื่องหมายว่าเสร็จแล้วแต่จริงๆ แล้วยังไม่เสร็จ
- งานที่ทำไม่ตรงกับความต้องการ
- งานมักจะสร้างข้อบกพร่องจำนวนมาก
- งานจำนวนมากต้องถูกทำซ้ำเป็นประจำ

หากปัญหาเหล่านี้คุ้นเคยสำหรับคุณ *Ready* สามารถช่วยได้

วิธีใช้หนังสือเล่มนี้

หนังสือเล่มนี้ถูกออกแบบมาให้ใช้งานได้ง่าย ไม่ใช้ตำราทางทฤษฎีหรือสไลด์กลยุทธ์—แต่เป็นคู่มือวิธีการติดตั้ง RMF (การไหลเวียนของความสมบูรณ์ของความต้องการ) ซึ่งอิงจากงานจริงกับลูกค้าและผ่านการทดสอบภาคสนามภายใต้แรงกดดันในการส่งมอบจริง

บทต่างๆ ถูกเขียนเรียงตามลำดับ แต่ RMF เองเป็นแบบโมดูลาร์ ประกอบด้วยแนวปฏิบัติพื้นฐานสามประการ:

- RMF 1: การทำงานร่วมกันเพื่อความเข้าใจร่วมกัน
- RMF 2: การควบคุมความสำเร็จของงานโดยใช้นิยามความสำเร็จเฉพาะกิจ
- RMF 3: การควบคุมการดำเนินการโดยใช้นิยามความพร้อมเฉพาะกิจ

แต่ละส่วน หรือที่เราเรียกว่า “นิสัย” สามารถยืนด้วยตัวเองได้ แต่พวกมันจะต่อยอดซึ่งกันและกัน หนังสือเล่มนี้ถูกออกแบบมาเพื่อช่วยให้คุณเข้าถึงทีละส่วน ตามลำดับ โครงสร้างนั้นสะท้อนถึงวิธีที่เราแนะนำให้ทีมนำ RMF ไปใช้ในทางปฏิบัติ—โดยแต่ละนิสัยจะถูกเพิ่มเข้ามาหลังจากที่นิสัยก่อนหน้าทำงานได้ดีแล้วเท่านั้น

วิธีนี้ช่วยลดความเสี่ยงการทำให้ทีมรู้สึกท้อแท้และให้โอกาสที่ดีที่สุดสำหรับการเปลี่ยนแปลงแต่ละครั้งที่จะติดตัวไป คุณจะได้เรียนรู้เพิ่มเติมเกี่ยวกับวิธีการทำเช่นนั้นเริ่มต้นใน [บทที่ 9](#)

หากคุณกำลังมองหาความช่วยเหลือ—ไม่ว่าจะเป็นคำแนะนำ การโค้ช หรือต้องการคนที่จะพูดคุยกับทีมผู้นำของคุณ—อย่าลังเลที่จะติดต่อเราโดยตรง

และหากคุณกำลังมองหาการสนับสนุนอย่างเป็นทางการในการนำ RMF ไปใช้ Producore มีชุดโปรแกรมที่ออกแบบมาเพื่อแนะนำการนำไปใช้ทีละขั้นตอน คุณสามารถเรียนรู้เพิ่มเติมได้ที่ <https://ready-book.link/rmf>

เกี่ยวกับผู้เขียน

Luniel de Beer เป็นผู้สร้างการไหลเวียนของความสัมพันธ์ของความต้องการ (RMF) ซึ่งเป็นระบบที่ใช้งานได้จริงสำหรับการแก้ไขช่องว่างระหว่างเจตนาของผลิตภัณฑ์และการดำเนินการทางวิศวกรรม เขามีประสบการณ์มากกว่า 15 ปีในการนำการเปลี่ยนแปลงสู่ระบบไอที การเชื่อมโยงระหว่างผลิตภัณฑ์และวิศวกรรม และการช่วยให้ทีมส่งมอบงานด้วยความชัดเจนและมั่นใจ

Luniel ยังเป็นผู้คิดค้นระบบการจัดการความสามารถของ Producore ซึ่งเป็นแนวทางที่ติดตามได้และปรับขนาดได้ในการสร้างแบบจำลองความสามารถของผลิตภัณฑ์ เขาได้คิดค้น PKB-Driven Development (PKBDD) ซึ่งเป็นระบบควบคุมเวอร์ชันสำหรับการจัดการความต้องการของผลิตภัณฑ์ที่คงอยู่ เครื่องมือเหล่านี้เป็นส่วนหนึ่งของกรอบการส่งมอบที่ใหญ่กว่าซึ่งพัฒนาที่ Producore

Max Guernsey, III เป็นสถาปนิกซอฟต์แวร์ นักการศึกษา และผู้ร่วมก่อตั้ง Producore ซึ่งเป็นบริษัทที่ปรึกษาที่ทุ่มเทให้กับการแก้ไขความล้มเหลวในการส่งมอบผ่านความเข้มงวดเชิงโครงสร้างและเทคนิค ด้วยประสบการณ์มากกว่าสองทศวรรษในการออกแบบเชิงวัตถุ การปรับโครงสร้างโค้ด การพัฒนาแบบขับเคลื่อนด้วยการทดสอบ และรูปแบบการออกแบบ เขาได้ทั้งส่งมอบระบบที่สำคัญต่อภารกิจและโค้ชทีมวิศวกรรมในระดับใหญ่ งานของเขาผสมผสานแนวปฏิบัติทางเทคนิคเชิงลึกกับการเปลี่ยนแปลงด้านพฤติกรรมและกระบวนการเพื่อช่วยให้องค์กรบรรลุความเป็นเลิศในการส่งมอบที่ยั่งยืน

Max มีส่วนร่วมอย่างมากใน PKBDD และเป็นผู้นำในการพัฒนาแนวทางของ Producore ในด้านการพัฒนาซอฟต์แวร์โดยขับเคลื่อนด้วยพฤติกรรม (Behavior-Driven Development หรือ BDD) ผ่านความเชี่ยวชาญอย่างลึกซึ้งในด้านข้อกำหนดเชิงพฤติกรรม

ร่วมกัน ผลงานของพวกเขาผสมผสานความชัดเจน การตรวจสอบย้อนกลับ และการควบคุมขั้นตอนเข้าเป็นระบบที่เป็นหนึ่งเดียวสำหรับการส่งมอบซอฟต์แวร์ ซึ่งสามารถขยายขนาดได้จากการปฏิบัติในระดับทีมไปสู่ความสามารถในระดับองค์กร

คำนำ

หมายเหตุถึงผู้นำด้านวิศวกรรม

หากคุณเป็นผู้ระดับสูงในองค์กรด้านวิศวกรรม คุณคงไม่ขาดความพยายาม วินัย หรือคนเก่ง แต่ถึงอย่างไร โครงการก็ยังคงหยุดชะงัก เป้าหมายเลื่อนออกไป และไม่เป็นไปตามความคาดหวัง ไม่ใช่เพราะทีมของคุณซี้เกียจ แต่เพราะมีบางสิ่งที่เป็นรากฐานที่มีปัญหาในวิธีที่งานถูกกำหนด ถูกออกแบบ และถูกส่งมอบ

หนังสือเล่มนี้ไม่ใช่คู่มือการเป็นผู้นำ แต่เป็นเครื่องมือในการวินิจฉัยเชิงโครงสร้าง มันเผยให้เห็นสิ่งที่เกิดขึ้นจริงภายในทีมของคุณ—ว่าทำไม “เกือบเสร็จ” จึงกลายเป็น “ยังไม่เสร็จ” และทำไมความคืบหน้าในระดับท้องถิ่นจึงแทบไม่เปลี่ยนแปลงเป็นผลลัพธ์เชิงกลยุทธ์

คุณอาจจะไม่เห็น*ตัวคุณเอง*ในหน้าเหล่านี้ แต่ถ้าทีมของคุณไม่สามารถส่งมอบสิ่งที่คุณต้องการได้ คุณจะเห็น*พวกเขา* และเมื่อคุณเห็นแล้ว คุณจะมีความเข้าใจภาษา—และระบบ—ที่จะแก้ไขมัน

จาก Luniel

ก่อนอื่น หนังสือเล่มนี้จะเกิดขึ้นไม่ได้เลยหาก ไม่มี Max ผู้ซึ่งมีความสามารถในการมองทะลุหมอกควันและกลั่นกรองแนวคิดให้เหลือแก่นแท้ได้อย่างที่ผมทำไม่ได้เลย

เราเดินทางมาถึงจุดนี้ได้อย่างไร?

เมื่อมองย้อนกลับไป ผมคิดว่าเป็นเพราะผมมักอยากเข้าใจว่าสิ่งต่างๆ ทำงานจริงๆ อย่างไร ไม่ว่าจะเป็นเรื่องศาสนา โภชนาการ หรือการพัฒนาซอฟต์แวร์ ผมมักเจอปัญหาเดิมๆ: คำตอบผิวเผินที่ไม่สามารถทนต่อแรงกดดันได้ ดังนั้นผมจึงขุดลึกลงไปเรื่อยๆ—ถามไม่เพียงแค่ว่าเราทำอะไร แต่ถามว่าทำไม และอะไรที่หายไปเมื่อมันไม่ได้ผล

หนึ่งในรอยร้าวแรกๆ ของระบบปรากฏขึ้นในบทบาทที่ผมต้องสวม หมวกสามใบ: Scrum Master, Product Owner และ Development Manager (!!) สำหรับทีมที่ส่งมอบบริการด้านข้อมูลที่บริษัทเทคโนโลยีที่มีชื่อเสียงแห่งหนึ่ง เราทำตามที่ Scrum บอก—Sprint สั้นๆ, สตอรี่ในแบ็คล็อก, วางแผนภายในครึ่งวัน—แต่ทุกครั้งที่เราเริ่ม Sprint ใหม่ เราจะเจอแรงเสียดทาน ทีมไม่เข้าใจปัญหาอย่างถ่องแท้ เราต้องทบทวนและแก้ไขข้อกำหนดระหว่าง Sprint การพึ่งพาที่สามารถป้องกันได้จะผุดขึ้นมาและทำให้เกิดความล่าช้า และขั้นตอนสำคัญๆ ถูกมองข้าม

ดังนั้นผมจึงเริ่มทำอะไรที่แตกต่างออกไป ผมจะดึงทีมและผู้มีส่วนได้ส่วนเสียเข้ามาในห้องสำหรับทุกสตอรี่ เดินผ่านปัญหาอย่างละเอียด ระดมความคิดหาทางแก้ไขร่วมกัน และเขียนสตอรี่หลังจากนั้น การวางแผน Sprint ลดลงเหลือหนึ่งชั่วโมง และความสำเร็จในการส่งมอบของเราพุ่งสูงขึ้นอย่างมาก

ช้าๆ ผมเริ่มตระหนักว่าความสำเร็จไม่ได้มาจากการทำงานหนักขึ้นใน Sprint แต่มาจากโครงสร้างที่คุณวางไว้*ก่อน*ที่มันจะเริ่ม

ต่อมา หลังจากได้ฟัง Jeff Sutherland พูดถึง “Definitions of Ready” ผมรู้ว่าบางสิ่งที่มีค่าอยู่ที่นั่น—แต่มันยังไม่พอ ประสบการณ์ของผมกับข้อกำหนดความต้องการ UX, UI, การวิจัย และต่อมา กับ BDD แสดงให้ผมเห็นว่างานที่แตกต่างกันต้องการความพร้อมที่แตกต่าง

ต่างกัน บางอย่างต้องการข้อกำหนดพฤติกรรม บางอย่างต้องการการเข้าถึงระบบ บางอย่างต้องการการติดตามความสามารถอย่างครบถ้วน

และทั้งหมดต้องการความเข้าใจร่วมกันที่ได้รับการยืนยันจริงๆ—ไม่ใช่แค่สันนิษฐาน

เมื่อผมทำงานกับทีมมากขึ้น ผมเห็นรูปแบบเดียวกันทุกที: ขั้นตอนที่ยากไป การพึ่งพาที่ไม่สมบูรณ์ ทีมที่พยายามอย่างดีที่สุดแต่ต้องคอยวิ่งวุ่นแก้ปัญหาที่ควรป้องกันได้ แม้แต่ทีมที่ยอดเยี่ยมก็ยังดิ้นรน—ไม่ใช่เพราะพวกเขาอ่อนแอ แต่เพราะพวกเขาขาดโครงสร้างที่ทำให้ความร่วมมือชัดเจน

ผลลัพธ์จากการเรียนรู้ การปรับปรุง และความหงุดหงิดทั้งหมดนั้นคือระบบที่มีโครงสร้างสำหรับการจัดการความร่วมมือ

นั่นคือสิ่งที่หนังสือเล่มนี้เกี่ยวกับ

ผมหวังว่ามันจะให้ความชัดเจนกับคุณเกี่ยวกับว่าปัญหาที่แท้จริงอยู่ที่ไหน และจะแก้ไขอย่างไร ผมหวังว่ามันจะให้ภาษาที่ช่วยปกป้องวิธีปฏิบัติที่อาจดูเหมือน “เกินจำเป็น” แต่จริงๆ แล้วจำเป็นมาก และที่สำคัญที่สุด ผมหวังว่ามันจะช่วยให้ทีมส่งมอบงานด้วยความเครียดน้อยลง เซอร์ไพรส์น้อยลง และผลลัพธ์ที่ดีขึ้นมาก

ถ้าเราทำสิ่งนี้ให้ถูกต้อง เราจะประหยัดเงินให้อุตสาหกรรมได้หลายพันล้านดอลลาร์

แต่ที่สำคัญกว่านั้น เราจะคืนสติสัมปชัญญะให้กับผู้คน

จาก Max

ผมทำงานกับปัญหานี้จากหลายมุมมองมาหลายทศวรรษ แต่ความก้าวหน้าของผมหยุดชะงักจนกระทั่งผมได้พบ Luniel

นี่เป็นเพราะก่อนที่ผมจะรู้จักเขา ผมกำลังเข้าหาปัญหานี้ในแง่พื้นฐานว่าเป็นปัญหาทางเทคนิค ผมมุ่งเน้นที่จะช่วยทีมในการนำสิ่งต่างๆ มาใช้ เช่น Test-Driven Development (TDD), การรีแฟกเตอร์, การออกแบบซอฟต์แวร์ขั้นสูง และต่อมาคือ Acceptance-Test-Driven Development (ATDD) หรือ Behavior-Driven Development (BDD)

ในกรณีส่วนใหญ่เหล่านั้น ปัญหาที่กล่าวถึงในหนังสือเล่มนี้ถูกมองว่าเป็นเพียงรายละเอียดการนำไปปฏิบัติของการสร้างแนวปฏิบัติทางเทคนิคเหล่านั้น

ไม่ได้หมายความว่าผมไม่ให้คุณค่ากับแนวปฏิบัติทางเทคนิคอีกต่อไป ผมยังคิดว่ามันมีความสำคัญอย่างยิ่ง แต่ผมไม่ได้แก้ปัญหาเรื่องความร่วมมือในการพัฒนาซอฟต์แวร์โดยตรง แต่กลับกัน มัน*เผยให้เห็น*ปัญหานั้น และจากนั้นผู้คนก็วางแผนแก้ไขลงบนกระบวนการของพวกเขาเพื่อจัดการกับมัน “แคพอ” ที่จะรองรับแนวปฏิบัติทางเทคนิคที่พวกเขากำลังพยายามนำไปใช้

ผมยังอยากจะพูดถึงคำถามว่าหนังสือเล่มนี้สามารถช่วยใครได้บ้าง คำตอบสั้นๆ คือ “น่าจะเกือบทุกคนในการพัฒนาซอฟต์แวร์” แต่คำตอบที่แท้จริงมีความละเอียดอ่อนที่ช่วยให้ปรับใช้กับสภาพแวดล้อมต่างๆ ได้โดยไม่เปลี่ยนความหมายพื้นฐาน

มีทีมหลายทีมที่ต้องการวิธีแก้ปัญหาที่นำเสนอในหนังสือเล่มนี้ คุณจะได้พบกับตัวอย่างหนึ่งในรูปแบบที่ถูกดัดแปลงให้เหมาะสมในบทที่ 1

นอกจากนี้ยังมีทีมที่ไม่ได้ **จำเป็น** ต้องใช้ระบบแบบที่เราเสนออย่างเคร่งครัด แต่ก็ยังสามารถได้ประโยชน์จากมันได้

ทีมที่ดีที่สุดที่เคยร่วมงานด้วย—ซึ่งเหนือกว่า**ทีมที่ดีที่สุดอันดับสอง**อย่างน้อยหนึ่งหน่วยค่าเบี่ยงเบนมาตรฐาน ถ้าไม่ใช่สองหน่วย—ตั้งอยู่ในพื้นที่ท่าंगโลของเซ็นทรัลอริซอน พวกเขามีประสิทธิภาพสูงมาก จนสามารถเอาชนะข้อจำกัดที่เกิดจากการไม่มีระบบเช่นนี้

ด้วยความสามารถอันล้นเหลือ กระนั้น ผู้จัดการของผมในตอนนั้น ทอม บาร์เรราส เคยพูดกับผมประมาณว่า “ผมสงสัยว่าสตอรี่ของเราจะออกมาดีกว่า เมื่อเราใช้เวลาคุยเกี่ยวกับการทดสอบล่วงหน้าสักหน่อย”

อีกครั้ง ตอนนั้นผมมองเรื่องนี้ผ่านมุมมองของการพัฒนาแบบทดสอบและการดำเนินการทางเทคนิค แต่ตอนนี้ผมรู้แล้วว่ามันเป็นอีกตัวบ่งชี้ว่า *ความพร้อม* เป็นปัจจัยที่ส่งผลต่อทีม... ทีมนั้นมีความสามารถและตอบสนองได้รวดเร็วมาก จนพวกเขาสามารถประสบความสำเร็จได้ด้วยการตอบสนองต่ออุปสรรคเมื่อเกิดขึ้น แทนที่จะป้องกันไม่ให้เกิดขึ้นตั้งแต่แรก

แม้ว่าคุณจะเป็นคนที่ไม่จำเป็นต้องกังวลเรื่องความพร้อมเพราะคุณสามารถเอาชนะมันได้ หรือคุณทำงานกับทีมที่มีความสามารถแบบเดียวกัน คุณก็ยังสามรถได้ประโยชน์จากเนื้อหาในหนังสือเล่มนี้

ส่วน I: บางสิ่งที่หายไป

เมื่อการทำสิ่งเดิมให้ดีขึ้นไม่ช่วยอะไร จงมองหาสิ่งที่ไม่ได้ทำ

บทที่ 1: ปัญหาที่ซ่อนอยู่

นี่เป็นเรื่องจริง¹เกี่ยวกับธนาคารแห่งหนึ่ง เราจะเรียกมันว่า “ธนาคาร” เป็นสถาบันเครดิตแบบสหพันธ์ที่เป็นส่วนหนึ่งของโครงสร้างพื้นฐานทางการเงินระดับชาติของสหรัฐอเมริกา

เรา (Luniel และ Max) ได้รับเชิญเข้ามาที่ธนาคารเพราะพวกเขากำลังประสบปัญหาในการส่งมอบโครงการซอฟต์แวร์ มันเป็นหนึ่งในสภาพแวดล้อมที่ไม่เป็นระบบมากที่สุดที่เราเคยเห็นมา และนั่นคือเหตุผลที่เราเลือกกรณีศึกษานี้เป็นเรื่องเปิด: ถ้าการเปลี่ยนแปลงที่มีความหมายเป็นไปได้ที่ธนาคารแห่งนี้ มันก็เป็นไปได้ที่ไหนก็ได้

บันทึกสั้นๆ เกี่ยวกับโครงการ

เมื่อเราใช้คำว่า “โครงการ” ในหนังสือเล่มนี้ เราหมายถึงในบริบทของการบริหารโครงการ แม้จะมีความคิดที่แตกต่างกันเกี่ยวกับความหมายของคำนี้ แต่เรากำลังใช้คำจำกัดความจาก [Project Management Institute](#):

“โครงการคือความพยายามชั่วคราวที่ดำเนินการเพื่อสร้างผลิตภัณฑ์ บริการ หรือผลลัพธ์ที่เป็นเอกลักษณ์”

นี่หมายความว่าโครงการมีจุดเริ่มต้นและจุดสิ้นสุดที่กำหนดไว้ เมื่อโครงการปิดตัวลง ความรู้และสิ่งประดิษฐ์ของโครงการจะถูกเก็บถาวร สมาชิกทีมจะถูกลบออก และสัญญาต่างๆ จะถูกสรุปปิด

ในหนังสือเล่มนี้ โครงการเกี่ยวข้องกับการดำเนินการเป็นหลัก โครงการส่วนใหญ่ตามที่กำหนดโดย PMI เริ่มต้นด้วยการศึกษาความเป็นไปได้หรือการออกแบบ การสร้างวิสัยทัศน์และกลยุทธ์ได้เกิดขึ้นแล้วก่อนที่โครงการจะเริ่มต้น

โครงการเกิดจากวิสัยทัศน์และกลยุทธ์นั้น และมันจะสำเร็จหรือล้มเหลวขึ้นอยู่กับว่าเป้าหมายที่วางไว้บรรลุผลหรือไม่—*ไม่ใช่*ว่าเป้าหมายเหล่านั้นถูกต้องหรือไม่

คุณอาจใช้คำว่า “โครงการ” แตกต่างออกไป และนั่นก็ไม่ใช่ไร แต่ทราบดีว่าเมื่อเราใช้มัน เรากำลังอ้างอิงถึงคำจำกัดความและบริบทข้างต้น

ทั้งหมดนี้ไม่ได้หมายความว่าเราเห็นด้วยกับการใช้การบริหารโครงการสำหรับการพัฒนาซอฟต์แวร์ ตรงกันข้าม แต่เรายอมรับว่ามันถูกใช้อยู่ เราจะจัดการกับปัญหานั้นในภายหลัง ใน [บทที่ 6](#)

1.1: การเขียนใหม่แบบคลาสสิก

ธนาคารกำลังเขียนพอร์ทัลการชำระเงินกู้ใหม่ด้วยเหตุผลหลายประการ

¹ เราได้เปลี่ยนรายละเอียดที่ระบุตัวตนเพื่อปกป้องความเป็นส่วนตัวของบุคคลและสถาบันที่เกี่ยวข้อง

ระบบเก่า ซึ่งเป็นโซลูชัน C#/.NET ทั้งหมด มีข้อบกพร่อง นอกจากทำให้ความพึงพอใจของลูกค้าลดลงแล้ว ยังก่อให้เกิดเหตุการณ์สนับสนุนที่มีค่าใช้จ่ายสูงอย่างไม่หยุดหย่อน ซึ่งต้องมีคนเข้าไปจัดการฐานข้อมูลด้วยตนเองเพื่อแก้ไขข้อผิดพลาดที่เกิดจากระบบ

ระบบเก่ายังเสื่อมโทรมในแง่ของการบำรุงรักษาอีกด้วย แทบเป็นไปไม่ได้เลยที่วิศวกรจะทำการเปลี่ยนแปลงที่มีความหมาย และแม้เมื่อพวกเขาทำได้ มันก็เป็นการเสียอย่างมาก

การเขียนใหม่ควรจะเปลี่ยนแปลงสิ่งนั้น

ระบบใหม่จะยังคงมีแบ็กเอนด์เป็น C#/.NET แต่จะมีการครอบคลุมด้วยการทดสอบอย่างเต็มที่ ส่วนฟรอนต์เอนด์จะถูกพัฒนาด้วย OutSystems ซึ่งเป็นโซลูชันโค้ดน้อยหรือไม่ต้องเขียนโค้ดที่เป็นที่นิยม ซึ่งช่วยให้องค์กรสามารถกำหนดแอปพลิเคชันในทีเดียวและได้เว็บแอป แอป Android และแอป iOS ที่สร้างขึ้นโดยอัตโนมัติเมื่อใดก็ตามที่พวกเขาตัดสินใจเผยแพร่การเปลี่ยนแปลง

ความหวังคือระบบใหม่จะปราศจากข้อบกพร่อง ทั้งปรับปรุงความพึงพอใจของลูกค้าและลดค่าใช้จ่ายในการสนับสนุนอย่างมีนัยสำคัญ

พวกเขายังหวังว่าการเขียนใหม่จะปลดล็อกนักพัฒนา—ด้วยการผสมผสานระหว่างวิธีการที่มีระเบียบวินัยมากขึ้นสำหรับแบ็กเอนด์และวิธีการโค้ดน้อยสำหรับฟรอนต์เอนด์ ซึ่งจะช่วยลดต้นทุนและความเสี่ยงของฟีเจอร์ใหม่ๆ อย่างมาก

ผลพลอยได้ที่ดีของการย้ายไปใช้ OutSystems คือพวกเขาจะได้แอปมือถือที่สะอาดและทันสมัยบนแพลตฟอร์มหลักทั้งสอง

นั่นคือความฝันเมื่อพวกเขาเริ่มต้นสามปีก่อนเรื่องนี้จะเริ่มขึ้น ความเป็นจริงก็คือ จนถึงตอนนี้ ทีมยังไม่ได้ส่งมอบอะไรเลย

1.2: มุมมองต่อปัญหา

เมื่อเราร่วมกับทีมผู้บริหารระดับสูง เราได้ยินความคับข้องใจที่เป็นธรรมชาติมากเกี่ยวกับการที่พวกเขาได้ลงทุนมากมายโดยไม่มีความสำเร็จเชิงกลยุทธ์เลย

จากมุมมองของพวกเขา พวกเขาได้ลองทุกอย่างแล้ว พวกเขาเปลี่ยนพนักงาน เพิ่มพนักงาน เปลี่ยนงบประมาณ เพิ่มแรงกดดัน และนำที่ปรึกษาเข้ามามากมาย (ซึ่งมีการบอกเป็นนัยว่าเราเป็นคนสุดท้าย) ไม่มีอะไรดูเหมือนจะทำให้มันดีขึ้น—อย่างน้อยก็ไม่ใช่ในแบบที่พวกเขาสามารถวัดได้ เพราะทั้งหมดที่พวกเขาเห็นคือ “เพิ่ม” อยู่ที่ยูนิโคดไตรมาสหนึ่งและยังคงอยู่ที่ยูนิโคดไตรมาสถัดไป

พวกเขาไม่ต้องการ “ความคืบหน้า” ที่มองไม่เห็น” อีกต่อไป พวกเขาต้องการผลลัพธ์

เมื่อเราร่วมกับผู้บริหารในองค์กรด้านผลิตภัณฑ์ เราได้รับเรื่องราวที่แตกต่างเล็กน้อย (แต่ยังคงคล้ายกัน) เพราะพวกเขาทำงานกับฝ่ายวิศวกรรมโดยตรงมากกว่า

ไม่ใช่ว่าทีมไม่ได้ทำอะไรเลย อย่างน้อยก็ในความคิดของพวกเขา แต่ปัญหาคือทีม ไม่เคย ทำในสิ่งที่ถูกขอให้ทำ มันเหมือนเป็นกฎตายตัวไปแล้ว: ไม่ว่างานที่ขอจะง่ายแค่ไหน และไม่ว่าจะสื่อสารชัดเจนขนาดไหน เมื่อถึงเวลาประเมินผลงานของทีม คุณก็จะได้อะไรที่แตกต่างออกไปโดยสิ้นเชิง

สถานการณ์มันแย่งขนาดที่กลายเป็นมุกตลกประจำว่า “เราต้องหาวิธีขอในสิ่งที่เราไม่ต้องการ เพื่อจะได้มีโอกาสได้สิ่งที่เราต้องการจริงๆ”

ผู้บริหารฝ่ายวิศวกรรม มองสถานการณ์แตกต่างออกไปอย่างสิ้นเชิง

สำหรับพวกเขา ปัญหาอยู่ที่ฝ่ายโปรดักต์ ที่ไม่ได้ส่งมอบข้อกำหนดความต้องการที่นำไปปฏิบัติได้ และไม่ได้ส่งข้อกำหนดความต้องการมาให้เพียงพอ ถ้าฝ่ายโปรดักต์แค่ “เข้าใจระบบ” ทีมก็จะสามารถส่งมอบสิ่งที่ต้องการได้ตรงเวลาและอยู่ในงบประมาณ

พวกเขาได้ลงทุนอย่างจริงจังในการปรับปรุงวิธีการเขียนและส่งมอบโค้ด และในมุมมองของพวกเขา ฝ่ายโปรดักต์ที่ไม่ได้ส่งมอบข้อกำหนดความต้องการที่ชัดเจน

เมื่อเราคุยกับที่ปรึกษาคณะอื่น (ที่แนะนำเราเข้ามาในองค์กร) พวกเขาระบุความผิดปกติที่เห็นได้อย่างถูกต้อง: ทุกคนดูเหมือนจะมุ่งเน้นไปที่การโทษคนอื่น เหตุผลที่พวกเขานำเราเข้ามาตั้งแต่แรกคือพวกเขากังวลเกี่ยวกับกลยุทธ์ด้านบุคลากรและต้องการประเมินผู้ปฏิบัติงานรายบุคคล แต่พวกเขามองว่าการชี้แจงโทษกันและการควบคุมแบบเผด็จการในระดับผู้บริหารเป็นต้นเหตุหลักของปัญหา

1.3: การสืบสวนของเรา

ภารกิจแรกของเราคือประเมินทีมและพยายามช่วยยกระดับทักษะของพวกเขาหากจำเป็น เราจึงเริ่มศึกษาคนที่ทำงานในแนวหน้า

แน่นอนว่ามีพื้นที่ให้ปรับปรุง

ผู้ปฏิบัติงานรายบุคคลในฝ่ายโปรดักต์ไม่ได้มีทักษะที่จำเป็นจริงๆ ความจริงแล้ว พวกเขาส่วนใหญ่เป็นโปรเจกต์แมนเนเจอร์ที่ถูกผลักดันให้มาทำบทบาทโปรดักต์ไอออนเนอร์ (PO) หรือโปรดักต์แมนเนเจอร์

ผลที่ตามมาคือ ครึ่งหนึ่งของพวกเขาเขียนข้อกำหนดความต้องการแบบ “คลุมเครือ” แล้วก็ยอมรับ (อย่างแท้จริง) ทุกอย่างที่ทำในแต่ละรอบโดยไม่มีการวิเคราะห้เชิงวิพากษ์ อีกครึ่งหนึ่งเขียนข้อกำหนดความต้องการแบบเดียวกันแล้วก็อ้างว่าทีม “น่าจะรู้” เรื่องที่ไม่เคยพูดถึง และเก็บงานค้างไว้แบบไม่มีกำหนด

ธนาครเคยใช้เพื่อจัดการและติดตามแบ็คล็อกงานของพวกเขา แม้ว่าสิ่งที่เราครอบคลุมในหนังสือนี้จะไม่เกี่ยวข้องโดยตรงกับสครัม แต่เราใช้คำศัพท์ของสครัมตลอดทั้งเล่มเพราะทีมส่วนใหญ่ หรืออย่างน้อยก็ส่วนมาก ใช้สครัม



คำนิยาม: สครัม

สครัมเป็นเฟรมเวิร์กเบาๆ ที่ช่วยให้บุคคล ทีม และองค์กรสร้างคุณค่าผ่านการแก้ปัญหาที่ซับซ้อนด้วยวิธีการที่ปรับตัวได้ โดยสรุปคือ:

1. โปรดักต์ไอออนเนอร์จัดลำดับความสำคัญของงานสำหรับปัญหาที่ซับซ้อนลงในโปรดักต์แบ็คล็อก
2. ทีมสครัมเปลี่ยนงานที่เลือกมาให้เป็ผลลัพธ์ที่มีคุณค่าในระหว่างสปรินท์
3. ทีมสครัมและผู้มีส่วนได้ส่วนเสียตรวจสอบผลลัพธ์และปรับปรุงสำหรับสปรินท์ถัดไป
4. ทำซ้ำ

หากคุณไม่คุ้นเคยกับสครัมและคำศัพท์ต่างๆ เราแนะนำให้อ่าน [Scrum Guide](#) เวอร์ชัน 2020 เป็นการอ่านที่รวดเร็วและให้ความกระจ่าง

เช่นเดียวกัน เราพบว่าทีมเทคนิคมีทักษะการเขียนโค้ดต่ำกว่าระดับปกติมาก (ต่ำกว่าค่าเฉลี่ย 2 ส่วนเบี่ยงเบนมาตรฐาน เป็นอย่างต่ำ) และยังต่อต้านการเปลี่ยนแปลงอย่างมากด้วย ผลที่ตามมาอย่างเป็นธรรมชาติคือ คุณภาพของโค้ดแย่มาก

แต่ตามความเห็นของทีม นี่ไม่ใช่สาเหตุที่พวกเขาส่งมอบงานไม่ได้ สำหรับพวกเขา ปัญหาอยู่ที่ข้อกำหนดความต้องการที่คลุมเครือและการเปลี่ยนแปลงระหว่างสปรินท์โดยฝ่ายโปรดักต์ ที่กำลังทำลายโครงการ

...และไม่มีใครพูดถึงปัญหาที่ใหญ่กว่านั้น ปัญหาที่ดูเหมือนเหมือนแต่จะยิ่งหนักกว่าที่คุณจะได้เจอกับตัวเอง

ทีมวิศวกรมีนิสัยที่ไม่เข้าใจข้อกำหนดความต้องการ สร้างอะไรสักอย่างแบบลุ่ม แล้วเรียกร้องเครดิตว่าได้ “ทำงานเสร็จแล้ว”

เราไม่ได้หมายถึงความเข้าใจผิดเล็กๆ น้อยๆ แต่เราหมายถึงการแยกกันคนละทาง: เราบอกให้พวกเขาปิดการใช้เงินกับเงินสดภายใต้เงื่อนไขบางอย่าง แต่พวกเขากลับปิดความสามารถในการเพิ่มอิมพลีเม้นต์สำรองแทน

แล้วพวกเขาก็บอกเรานั้นคือสิ่งที่เราขอ

1.4: การขุดลึกลงไปอีก

ช่องว่างทางทักษะทั้งสองอย่างสามารถแก้ไขได้ แต่เราสงสัยว่ามันไม่ใช่อุปสรรคที่แท้จริง

มีบางอย่างผิดปกติ เราจึงขุดลึกลงไปอีก เราเริ่มด้วยคำถามนี้: ทำไมการเขียนข้อกำหนดความต้องการถึงใช้เวลานานและให้ผลลัพธ์ที่แย่นขนาดนี้?

เหตุผลหนึ่งคือความรู้ที่จำเป็นในการเขียนข้อกำหนดความต้องการที่มีความหมายนั้นมียุ่่น้อยมาก ทีมวิศวกรรมและทีมโปรดักต์มีความรู้เหล่านี้อยู่เพียงเล็กน้อย

บางส่วนถูกฝังอยู่ในโค้ดของระบบเดิม บางส่วนสูญหายไปโดยสิ้นเชิง แต่ส่วนใหญ่แล้วถูกเก็บไว้ในรูปแบบความรู้ที่ถ่ายทอดกันภายในองค์กรโดยผู้เชี่ยวชาญเฉพาะด้านที่กระจายอยู่ตามหน่วยงานต่างๆ ของธนาคาร นั่นหมายความว่าการสร้างความต้องการของระบบที่จะช่วยผลักดันเป้าหมายเชิงกลยุทธ์นั้นเป็นกิจกรรมที่ต้องใช้แรงงานและเวลามาก

ในทางตรงกันข้าม ทีมผู้นำมีความต้องการฟิเจอร์ใหม่ๆ อย่างไม่มีที่สิ้นสุด คำสั่งคือ “ให้วิศวกรทำงานตลอดเวลา—ป้อนความต้องการของระบบให้พวกเขาจนเต็มที” จุดเน้นอยู่ที่ปริมาณความต้องการเพื่อให้ทีมยุ่งอยู่ตลอดเวลา—ซึ่งเป็นมุมมองที่ขัดแย้งกับความใส่ใจและเวลาที่จำเป็นในการกำหนดความต้องการที่จะ “สร้างความเปลี่ยนแปลง”

1.5: การปรับปรุงให้ดีขึ้นกลับไม่ได้ทำอะไรดีขึ้น

ปัญหาเหล่านี้ล้วนแก้ไขได้ แต่การแก้ไขกลับไม่ช่วยอะไร

การพัฒนาเทคนิคการพัฒนาซอฟต์แวร์ที่ผ่านมาไม่ได้ช่วยให้ทีมส่งมอบงานได้ดีขึ้น แต่เม็กซิก็พยายามช่วยปรับปรุงทีมให้ดีขึ้น

เขาแนะนำแนวคิดปฏิบัติจากหลักการเขียนโปรแกรมยุคกลางศตวรรษที่สี่สิบ เช่น “อย่าก็อปปีแอนด์เพสต์โค้ดนั้น 27² ครั้ง ให้เขียนเป็นฟังก์ชันแล้วเรียกใช้แทน” ข้อเสนอนี้เพียงอย่างเดียวก็ช่วยปรับปรุงคุณภาพของโค้ดใหม่และทำให้พวกเขาเริ่มพัฒนาคุณภาพได้อย่างมาก

คำแนะนำการเขียนโค้ดพื้นฐานนั้นและอื่นๆ ช่วยให้พวกเขาเขียนโค้ดที่ดีขึ้นซึ่งจะดูแลรักษาได้ง่ายขึ้นในอนาคต

...แต่มันไม่ได้ช่วยให้โครงการก้าวหน้าขึ้น

ในด้านผลิตภัณฑ์ ลูเนียสามารถแนะนำ BDD และทำให้มั่นใจว่าเจ้าของผลิตภัณฑ์ตรวจสอบความต้องการอย่างละเอียดก่อนที่จะส่งต่อให้ทีม

²นี่ไม่ใช่การพดเกินจริง และยังไม่ใช่กรณีที่เคยที่สุด ในบางกรณี มีการทำซ้ำอัลกอริทึมเดียวกันแบบเหมือนกันทุกประการเกือบร้อยครั้ง

เขาทำให้ทีมร่วมมือกันและเชื่อมั่นในการประเมินว่ารายการใน Product Backlog (PBI) นั้นเสร็จสมบูรณ์จริงๆ หรือไม่



คำจำกัดความ: รายการใน Product Backlog (PBI)

รายการใน Product Backlog (PBI) คือหน่วยงานที่แบ่งแยกได้ใน product backlog ซึ่งแสดงถึงการเปลี่ยนแปลง การเพิ่มเติม หรือการปรับปรุงที่อาจเกิดขึ้นกับผลิตภัณฑ์ PBI อาจอยู่ในหลายรูปแบบ—ฟีเจอร์ การแก้ไข การปรับปรุงทางเทคนิค งานวิจัย ฯลฯ—และถูกกำหนดโดยการมีส่วนร่วมในคุณค่าของผลิตภัณฑ์

หลายทีมเรียก PBI ว่า “สตอรี” หรือ “user stories” แต่คำที่ถูกต้องใน Scrum คือ “Product Backlog Item” หรือ “PBI” เมื่อ PBI ถูกนำเข้าสู่ Sprint มันก็จะเป็นส่วนหนึ่งของ Sprint Backlog ด้วย เพื่อความเรียบง่ายและความเป็นกลาง เราใช้ “Product Backlog Item” หรือ “PBI” เพื่ออ้างถึงรายการงานใดๆ ที่ทีม Scrum กำลังจัดการ—ไม่ว่าคุณ จะเรียกมันว่า Product Backlog Item, Sprint Backlog Item (SBI), user story, story, work item, หรือ backlog item

มันเพิ่มความชัดเจน แต่ไม่ได้สร้างการไหลลื่นของงาน

ด้วยความช่วยเหลือจากที่ปรึกษามิตรที่นำเราเข้ามา เราสามารถบรรเทาแรงกดดันอันหนักหน่วงที่มีต่อทีม และผู้เขียนความต้องการได้ (ชั่วคราว)

มันอาจช่วยสร้างความไว้วางใจเล็กน้อย แต่ไม่ได้ให้ผลลัพธ์ที่จับต้องได้

เราเริ่มสร้างฐานความรู้ที่ช่วยให้ผู้คนติดตามความรู้ทางธุรกิจที่พวกเขาต้องการเพื่อเขียนความต้องการและระบุจุดที่มีช่องว่างในความรู้

มันช่วยเร่งการเขียนความต้องการ แต่ไม่ได้ช่วยให้ผลิตภัณฑ์ออกสู่ตลาดได้

หลังจากการมีปฏิสัมพันธ์หลายเดือน เราได้ช่วยให้ผู้นำเห็นว่าพวกเขาอยู่ตรงไหน แต่พวกเขายังห่างไกลจากเป้าหมายมาก และไม่ได้เข้าใจลึกขึ้นเลย

พวกเขากำลังเตรียมที่จะกลับไปใช้กลยุทธ์เดิมของการ “ไหลตagnan” ให้ทีมเพื่อให้แน่ใจว่าพวกเขาอยู่ตลอดเวลา

1.6: กระบวนการคัดออก

คงง่ายที่จะยกมือขึ้นแล้วพูดว่า “หมดหวังแล้ว” มีข้ออ้างมากมายที่คนเราอาจหยิบยกขึ้นมา:

- ทีมวิศวกรมีทักษะต่ำ (จริง)
- ผู้เขียนความต้องการมีชุดทักษะที่ไม่เหมาะสม (จริง)
- ผู้นำกดดันทีมด้วยความคาดหวังที่ไม่สมจริง (จริง)
- องค์กรขาดความรู้ทางธุรกิจที่สำคัญที่จำเป็นต่อการทำงาน (จริง)
- ผู้บริหารดูเหมือนไม่ไว้วางใจซึ่งกันและกัน (จริง)

ทั้งหมดนั้นเป็นความจริง แต่การปรับปรุงได้เกิดขึ้นในทุกด้านเหล่านั้นแล้ว และไม่มีอะไรดูเหมือนจะทำให้ปัญหาพื้นฐานดีขึ้น ไม่มีอะไรช่วยให้ทีมวิศวกรรม ฝ่ายผลิตภัณฑ์ ฝ่ายบริหาร หรือผู้บริหารระดับสูงเข้าใจเป้าหมายของพวกเขามากขึ้น

...และนั่นคือค่าไปถึงทางแก้ปัญหาตรงนั้น: ปัญหาทั้งหมดที่กล่าวมาก่อนหน้านี้คือปัญหาที่ทุกคนมองเห็นอยู่แล้ว



ถ้าตัวแปรที่คุณมองเห็นไม่ได้สร้างความแตกต่าง แสดงว่าต้องมีตัวแปรที่คุณมองไม่เห็นที่สร้างความแตกต่างแน่นอน

ปัญหาที่แท้จริงคือสิ่งที่ไม่มีใครรู้ว่ามันมีอยู่

1.7: การตามล่าหาตัวการที่แท้จริง

ในการค้นหาตัวการที่แท้จริง - สิ่งที่กำลังขัดขวางอนาคตจากการบรรลุเป้าหมาย - เราจำเป็นต้องเริ่มจากจุดใดจุดหนึ่ง

จุดที่สมเหตุสมผลที่จะมองหาคือสิ่งที่ PBI ที่ล้มเหลว (ส่วนใหญ่) มีเหมือนกัน

เราเริ่มด้วยการตัดสิ่งที่เรารู้ว่าไม่ใช่จุดร่วมเพราะมีความแตกต่างกันมาก:

- ส่วนใดของระบบ: บาง PBI กระทบเฉพาะแบ็กเอนด์ บางอันกระทบแค่ฟรอนต์เอนด์ บางอันกระทบทั้งสองส่วน
- ทีมไหนเป็นผู้ดำเนินงาน - ดูเหมือนไม่สำคัญว่าใครเป็นคนทำ โอกาสล้มเหลวยังสูงเหมือนเดิม
- โปรดักต์ไอออนเนอร์คนไหนเป็นผู้เขียนงาน - เหมือนกับกรณีของทีม

จากนั้นเราเริ่มมองหาสิ่งที่เหมือนกัน รายการไม่ได้ยาวมาก แต่ก็ไม่น่าสนใจ:

- ทีมวิศวกรรม
- โปรดักต์ไอออนเนอร์
- ผู้นำ
- วัฒนธรรม
- สภาพแวดล้อมการพัฒนา
- แนวทางปฏิบัติทางวิศวกรรม
- เทคนิคการเขียนความต้องการ
- โดเมน (การเงิน)
- บริการที่ต้องพึ่งพา

หลายอย่างในนั้นก็สามารถตัดทิ้งได้ทันที ทีม โปรดักต์ไอออนเนอร์ ผู้นำ วัฒนธรรม และสภาพแวดล้อมการพัฒนาล้วนได้รับการปรับปรุงเมื่อเร็วๆ นี้แต่ไม่ส่งผลกระทบจริงต่อผลลัพธ์ที่มีความหมาย เราได้ช่วยปรับปรุงแนวทางปฏิบัติทางวิศวกรรมและการเขียนความต้องการด้วยตัวเองและยืนยันได้ว่าการปรับปรุงเหล่านั้นยังคงอยู่ แต่ก็ยังไม่ช่วยอะไร

คุณแทบจะโทษโดเมนไม่ได้ การเงินเป็นหนึ่งในการคำนวณที่เก่าแก่ที่สุดในประวัติศาสตร์ที่บันทึกไว้ มันพัฒนามาอย่างเต็มที่แล้ว นอกจากนี้ ธนาคารอื่นๆ ก็กำลังปรับใช้ซอฟต์แวร์ ซึ่งพิสูจน์ให้เห็นว่าสมมติฐานที่ว่าธนาคารไม่สามารถทำได้นั้นไม่เป็นความจริง (ซึ่งเห็นได้ชัดว่าเป็นการคาดเดาที่เกินจริง)

บริการที่ต้องพึ่งพาที่ไม่สามารถโทษได้เช่นกัน เพราะพวกเขากำลังมีปัญหาในการเปลี่ยนแปลงพอยท์ กับโครงการที่เราากำลังดูอยู่...

...แต่นั่นทำให้เราเริ่มคิด: ถ้าเราเริ่มวิเคราะห์สาเหตุของความล้มเหลวล่ะ?

1.8: การวิเคราะห์เมล็ดพันธุ์แห่งความล้มเหลว

PBI หนึ่งรายการล้มเหลวในการผลักดันผลิตภัณฑ์ไปข้างหน้าเพราะทีมทำในสิ่งที่พวกเขามักจะทำ นั่นคือทำอะไรที่สุ่มและแทบจะไม่เกี่ยวข้องกับสิ่งที่ร้องขอเลย นั่นเป็นสัญญาณชัดเจนว่าพวกเขาไม่เข้าใจรายการงาน ดังนั้นความเข้าใจจึงเป็นตัวเลือกที่สำคัญ แม้ว่าเราจะได้ทำงานเกี่ยวกับเรื่องนี้ไปบ้างแล้วตอนที่ช่วยพวกเขานำ BDD มาใช้

อีก PBI หนึ่งล้มเหลวในการปิดงานเพราะพวกเขาคำนวนผิด นั่นเป็นอีกหลักฐานหนึ่งที่บ่งชี้ว่าความเข้าใจอาจเป็นปัญหาหลัก

รายการงานที่สามที่เราวิเคราะห์ไม่ได้รับการตรวจสอบอย่างเหมาะสมจากโปรดักต์โอเนเนอร์ - เธอเซ็นผ่านเมื่อทีมบอกว่าถึงเวลาปิดงาน นั่นทำให้สมมติฐานของเราสั้นลงเล็กน้อย แต่ก็ยังพอจะอ้างได้ว่าเธอไม่เข้าใจว่ารายการงานนี้เข้ากับแผนระดับสูงอย่างไร

อาจจะใช่ ประมาณนั้น ถ้าเรามองมันในแง่ด้วยสายตาที่หริ่งจิงๆ

จากนั้นเราก็บังเอิญเจอ PBI ที่ไม่เข้ากับรูปแบบเลย ทีมดูเหมือนจะเข้าใจ - แม้ว่าจะไม่มีการยืนยันได้ว่าพวกเขาเข้าใจจริงๆ แต่มันไม่สำคัญ: พวกเขาไม่มีโอกาสที่จะประสบความสำเร็จหรือล้มเหลวด้วยตัวเองเลยเพราะพวกเขาเจอการพึ่งพาที่ต้องได้รับการอัปเดตและต้องเลื่อนงานออกไปหลายสปรินท์

แม้ว่าพวกเขาจะไม่เข้าใจว่าพวกเขาควรทำอะไร พวกเขาก็ไม่มีโอกาสกับรายการในแบ็คล๊อคนั้นเลย ดังนั้นความเข้าใจจึงไม่ใช่ปัญหาในกรณีนั้น

แน่นอนว่าข้อยกเว้นหนึ่งกรณีไม่ได้พิสูจน์ว่าสาเหตุหลักนั้นผิด แต่มันกระตุ้นความอยากรู้อยากเห็น เราเริ่มมองหาหลักฐานที่หักล้างเพิ่มเติม

และเราก็พบมัน มีรายการงานที่:

- ล้มเหลวเพราะทีมรู้ว่าตัวเองไม่เข้าใจ แต่ไม่สามารถหาผู้เชี่ยวชาญในเรื่องนั้นๆมาแก้ปัญหาได้
- เปลี่ยนเป็นประสบการณ์ที่แย่งสำหรับผู้ใช้เพื่อเป็นทางแก้ปัญหาชั่วคราวให้สอดคล้องกับการทำงานของบริการต้นน้ำ
- ต้องเลื่อนออกไปเพราะการพึ่งพาดันน้ำยังไม่พร้อม
- ไม่สามารถทำให้เสร็จได้เพราะผู้ทดสอบไม่สามารถรวบรวมข้อมูลทดสอบได้ทันเวลา
- ปิดงานแล้วแต่ต้องทำใหม่เพราะสิ่งที่ขอมาไม่ถูกต้อง
- ล้มเหลวเพราะทีมไม่ตระหนักว่าโค้ดที่มีอยู่ซับซ้อนแค่ไหน
- ไม่ได้รับการประมาณการ
- ถูกประมาณการต่ำกว่าความเป็นจริงมาก³

³ เราไม่ได้หมายถึงแค่การประเมินผิด ดูเหมือนว่าทีมอาจจะแค่อัปเดตค่า “3” ลงในช่องประเมินสำหรับ PBI จำนวนมาก

- เปลี่ยนกลางสปรินท์เพราะโปรดักต์ไอออนเนอร์เพิ่งได้รับความรู้เกี่ยวกับโดเมนที่ต้องการ
- ดูเหมือนจะเปลี่ยนหลังจากสปรินท์ (จากมุมมองของทีม) เพราะโปรดักต์ไอออนเนอร์และทีมไม่เคยตกลงกันว่ามันหมายถึงอะไร

รายการยังมีต่อไปอีก แต่แค่นี้ก็เพียงพอสำหรับเรื่องราวนี้แล้ว

เราสามารถโยนแต่ละกรณีเข้ากับ “ความเข้าใจ” ได้ในระดับหนึ่ง—และแน่นอนว่าการขาดความเข้าใจเป็นส่วนหนึ่งของปัญหาหลายกรณี—แต่มันไม่ได้หมายความว่าความเข้าใจคือสาเหตุหลัก... โดยเฉพาะอย่างยิ่งเมื่อเราได้พยายามสร้างความเข้าใจร่วมกันมาบ้างแล้ว แต่ก็ไม่ได้ช่วยอะไรมากนัก

แล้วเราก็กิดความคิดขึ้นมา มีสิ่งที่เป็นพื้นฐานยิ่งกว่าที่หายไป ในกรณีที่มีการขาดความเข้าใจเข้ามาเกี่ยวข้อง นั่นเป็นเพียงสาเหตุใกล้เคียงเท่านั้น

สาเหตุระยะไกลนั้นกว้างกว่ามาก

1.9: จุดตัดร่วม

สิ่งหนึ่งที่ทุก PBI ที่เรารีเคราะห์ว่าล้มเหลวมีเหมือนกันคือ: ทั้งหมดถูกเริ่มต้นเร็วเกินไป

เมื่องานล้มเหลวเพราะทีมรู้ว่าพวกเขาไม่เข้าใจปัญหาและไม่สามารถหาผู้เชี่ยวชาญมาช่วยอธิบายได้ นั่นหมายความว่าทีมเริ่มทำ PBI ทั้งๆ ที่รู้ว่าพวกเขาไม่เข้าใจปัญหา

เมื่อรายการในบัญชีงานค้างต้องเปลี่ยนไปเป็นประสบการณ์ที่แย่งเพราะวิธีการทำงานของบริการต้นน้ำ นั่นหมายความว่างานถูกเริ่มโดยที่ไม่เข้าใจผลกระทบของบริการต้นน้ำจริงๆ

การเลื่อนงานเพราะการพึ่งพาต้นน้ำไม่พร้อมตอนจบ หมายความว่าไม่มีการรับประกันความพร้อมตั้งแต่ต้น

...และเป็นเช่นนี้กับทุกกรณี: นักทดสอบไม่มีข้อมูลพร้อมหรือไม่รู้วิธีหาข้อมูลก่อนเริ่ม PBI ความต้องการไม่ได้รับการตรวจสอบอย่างดีก่อนส่งต่อให้ทีม โค้ดไม่ได้รับการตรวจสอบก่อนเริ่มงาน การประเมินไม่เพียงพอหรือไม่ได้ทำเลย ขาดความรู้ในโดเมน และแน่นอน ไม่ได้ตรวจสอบความเข้าใจร่วมกัน

ปัญหาที่แท้จริงคือ การพัฒนาถูกเริ่มต้นก่อนที่งานเหล่านั้นจะพร้อม



จากประสบการณ์ของเรา งานส่วนใหญ่ที่ล้มเหลวในการส่งมอบ เป็นเพราะไม่พร้อมตั้งแต่ตอนเริ่มพัฒนา

ดังนั้นเราจึงเริ่มช่วย The Bank ในเรื่องนี้

ณ จุดนี้ คุณอาจคิดว่าแค่บอกว่า PBI ควรพร้อมก็น่าจะเพียงพอแล้ว อย่างไรก็ตาม การนำไปปฏิบัติจริงไม่ได้ง่ายขนาดนั้น เหมือนกับแนวคิด “ซื้อถูก ขายแพง” ที่ฟังดูง่าย

ยังมีส่วนที่หายไปที่เป็นต่อการนำคำแนะนำที่ดีไปปฏิบัติ

1.10: การเปลี่ยนแปลงครั้งใหญ่

เราพบชิ้นส่วนที่หายไปของปริศนาที่ปลดล็อกพวกเขา ซึ่งในที่สุดก็ปลดล็อกโครงการทั้งหมด

เมื่อเราจบการทำงานนั้น วิศวกรยังคงมีทักษะต่ำกว่าค่ามัธยฐาน PO ยังไม่มีทักษะที่เหมาะสม วัฒนธรรมยังไม่ได้รับการแก้ไข...

แต่ผลิตภัณฑ์ก็เริ่มขายไปข้างหน้าและในที่สุดก็ออกสู่ตลาดได้

เมื่อจบหนังสือเล่มนี้ คุณจะรู้ว่าชิ้นส่วนที่หายไปคืออะไร และต้องทำอะไรเพื่อวางมันลงในที่ที่เหมาะสม และคุณสามารถปลดล็อกองค์กรที่ดูเหมือนถูกกำแพงที่มองไม่เห็นขวางกั้นอยู่

บันทึกสั้นๆ เกี่ยวกับขอบเขต

หนังสือเล่มนี้เกี่ยวกับปัญหาเฉพาะที่พบได้ทั่วไป: การขาดโครงสร้าง ความชัดเจน และวุฒิภาวะในจุดส่งต่องานระหว่างฝ่ายธุรกิจและฝ่ายวิศวกรรม มันตั้งอยู่บนสมมติฐานว่ามีบางสิ่งถูกเลือกให้พัฒนาแล้ว และมุ่งเน้นไปที่การทำให้แน่ใจว่างานพัฒนาเกิดขึ้นด้วยความเข้าใจร่วมกัน ความพร้อม และการติดตามความสำเร็จได้

เทคนิคในหนังสือเล่มนี้ไม่ได้บอกว่าควรสร้างอะไร ทำไม่ต้องสร้าง หรือจะรู้ได้อย่างไรว่าสิ่งที่สร้างนั้นถูกต้อง ถ้าองค์กรของคุณขาดการจัดการผลิตภัณฑ์ที่แท้จริงหรือวงจรข้อมูลย้อนกลับที่มีความหมาย เราไม่ได้พยายามแก้ปัญหานั้นที่นี่ สิ่งที่เราเสนอแทนคือวิธีทำให้ช่องว่างเหล่านั้นมองเห็นได้ชัดเจน และลดต้นทุนในการค้นพบว่าคุณทำผิดพลาด

เมื่อใช้ในบริบทที่ถูกต้อง วิธีแก้ปัญหานี้จะนำมาซึ่งการไหลของงาน ความปลอดภัย และความชัดเจน แต่เหมือนระบบอื่นๆ มันอาจถูกนำไปใช้ผิด—โดยเฉพาะเมื่อใช้แบบแยกส่วนหรือขาดความตระหนักรู้

บทที่ 2: ต้นทุนของการขาดรากฐาน

มันมีเหตุผลที่จะใช้เวลาสักเล็กน้อยในการพิจารณาว่าปัญหานี้สามารถส่งผลเสียต่อองค์กรได้อย่างไรบ้าง

เราพบว่าปัญหาหลัก 3 “หมวดหมู่” ที่เกิดจากการขาดความพร้อม:

- การขาดหรือไม่สมบูรณ์ของความเข้าใจร่วมกันระหว่างและภายในทีมผลิตภัณฑ์และวิศวกรรม
- การขาดการควบคุมว่าเป้าหมายที่แท้จริงของรายการในบัญชีงานของผลิตภัณฑ์คืออะไรและเมื่อไหร่ที่จะเสร็จสมบูรณ์จริงๆ
- การขาดการคัดกรองว่าเมื่อไหร่ที่งานหนึ่งๆ สามารถเริ่มเข้าสู่ขั้นตอนการพัฒนาได้

นอกจากนี้ เราสังเกตว่าการเปลี่ยนแปลงสิ่งเหล่านี้เป็นเรื่องที่ท้าทายมาก ซึ่งก็สมเหตุสมผล: การเปลี่ยนแปลงเป็นเรื่องยาก

นิสัยเก่าๆ ตายยาก และการสร้างนิสัยใหม่ก็เป็นเรื่องยาก จากประสบการณ์ของเราในฐานะที่ปรึกษา เราสังเกตว่าผู้คนกลับไปสู่นิสัยเดิมๆ ได้*ง่ายมาก* และในทางกลับกัน การสร้างนิสัยใหม่นั้นยากกว่าเปรียบเทียบ

ดังนั้นคุณต้องมีกลไกที่บังคับใช้นิสัยใหม่และไม่ส่งเสริมนิสัยเก่า

เพื่อแก้ไขปัญหานี้ เราเชื่อว่ายังมีองค์ประกอบที่ขาดหายไปอีกอย่างคือความรับผิดชอบและการตรวจสอบย้อนกลับ

2.1: จิกซอร์ที่ไม่มีภาพบนกล่อง

คุณเคยลองต่อจิกซอร์โดยไม่มีภาพบนกล่องไหม? คุณทำได้ แต่มันช้ากว่า น่าหงุดหงิดกว่า และเต็มไปด้วยการเริ่มต้นที่ผิดพลาด

คุณทำคืบหน้าไป แล้วก็ต้องรื้อทิ้ง คุณสงสัยว่าชิ้นไหนเข้ากับตรงไหน คุณคิดว่าคุณกำลังทำภาพเดียวกัน จนกระทั่งคุณตระหนักว่าไม่ใช่

นั่นคือสิ่งที่การพัฒนาซอฟต์แวร์มักจะรู้สึกเหมือน

บัญชีงานเต็ม สปρινท์กำลังดำเนินไป ทุกคนทำงานหนัก

แต่หากไม่มีภาพรวมกันของสิ่งที่เรากำลังสร้าง การทำงานร่วมกันก็กลายเป็นเรื่องของโชค ไม่ใช่ระบบ

หากขาดความชัดเจน แม้แต่ทีมที่ดีที่สุดก็รู้สึกหงุดหงิด เหนื่อยล้า และรู้สึกว่าความพยายามของพวกเขาไม่ได้รับการให้คุณค่า

2.2: สุภาพิตเก่าและความจริงอันโหดร้าย

มีเหตุผลว่าทำไมแนวโน้มนี้เกี่ยวกับการเขียนความต้องการ แม้แต่แนวโน้มทางวิศวกรรมบางอย่าง จึงมุ่งเน้นอย่างมากในการสร้างความเข้าใจร่วมกันระหว่างผู้ร้องขอและทีมพัฒนา ไม่มีอะไรจะสร้างความวุ่นวายได้เท่ากับการที่วิศวกรไม่รู้จักจริงๆ ว่าพวกเขาควรจะสร้างอะไร



“ขยะเข้า ขยะออก” เป็นสุภาษิต ไม่ใช่คำพูดที่ไร้ความหมาย

ภาษาอังกฤษเต็มไปด้วยคำที่ขัดแย้งกันเอง...

- คุณสามารถ sanction การกระทำของใครบางคนได้ อาจจะหมายความว่าให้คุณให้การอนุญาตล่วงหน้า หรืออาจจะหมายความว่า คุณกำลังประณามหลังจากเหตุการณ์เกิดขึ้น
- คุณสามารถปิดฝุ่นเบาๆ ได้ แต่ถ้าสิ่งนั้นเป็นตุ๊กตาของ มันหมายถึงคุณกำลังเอาฝุ่นออก ในขณะที่ถ้าพูดถึงขนมเบญญ์ คุณกำลังโรยผงน้ำตาลเพิ่ม
- ถ้าคุณ hold up ทีม คุณอาจจะเป็นเหตุผลที่ทำให้ทีมนั้นสามารถทำงานต่อไปได้ หรือคุณอาจจะเป็นเหตุผลที่ทำให้พวกเขาไม่สามารถไปไหนได้

คำที่มีความหมายตรงกันข้ามในตัวเอง อาจเป็นตัวอย่างที่เด่นชัดที่สุด แต่มันเป็นเพียงรูปแบบหนึ่งของความกำกวมเท่านั้น คำบางคำไม่เพียงแต่มีความหมายตรงข้ามกับตัวเอง แต่ยังมีความหมายอื่นๆ ที่อาจสร้างความสับสนได้อีกมาก

“In this clip, he clipped a coupon from a news paper and clipped it to the paper on his clipboard along with the other clippings while a clipper in the background was moving along at a decent clip.”

และไม่ใช่ว่าภาษาอังกฤษเท่านั้น ภาษาธรรมชาติทุกภาษาที่เรารู้จักก็มีคุณสมบัตินี้

แต่กระนั้น มันก็เป็นสิ่งเดียวที่เราใช้เพื่อใช้ในการระบุความต้องการ

ดังนั้น เมื่อทิมวิศวกรไม่ได้ยืนยันว่าความเข้าใจของพวกเขาเกี่ยวกับความต้องการนั้นตรงกับของผู้ร้องขอ ทีมนั้นกำลังฟังพาโซคะตานั้นคือ ผลลัพธ์ที่ดีที่สุดที่เป็นไปได้คือพวกเขาเลือกการตีความที่ถูกต้องและผู้ร้องขอไม่เปลี่ยนใจระหว่างทาง

ผลลัพธ์นั้นไม่ได้รับประกันเลย

2.3: ผลลัพธ์ที่พบบ่อย

หากไม่มีความเข้าใจร่วมกันที่ได้รับการยืนยัน ทีมต้องเผชิญกับความเสียหายหลายประการ

โดยส่วนใหญ่แล้ว ผลลัพธ์ที่สร้างความเจ็บปวดที่สุดคือการที่ทีมสร้างสิ่งที่ผิดออกมา

จังหวะเวลาที่พวกเขาค้นพบสิ่งนี้สามารถควบคุมได้ด้วยคุณลักษณะต่างๆ ของกระบวนการของพวกเขา ตัวอย่างเช่น การนำสกรีมไปใช้อย่างมีประสิทธิภาพสามารถตรวจพบความเข้าใจผิดแบบนี้ได้ตั้งแต่เริ่มต้นการทำงาน ในขณะที่กระบวนการแบบวอเตอร์ฟอลล์มีโอกาสสูงที่จะล่าช้าในการค้นพบเป็นเวลาหลายเดือน



คำจำกัดความ: วอเตอร์พอลล์

โมเดลการพัฒนาซอฟต์แวร์แบบลำดับขั้นตอนที่แพร่หลายในช่วงปลาย ศตวรรษที่ 20 มันถูกนำเสนอครั้งแรกใน **บทความปี 1970 โดย Winston W. Royce** ซึ่งแสดงการพัฒนาเป็นขั้นตอนที่ไหลลงมาเป็นลำดับ—ความต้องการ การออกแบบ การพัฒนา การทดสอบ และอื่นๆ—แต่ละขั้นตอนไหลเข้าสู่ขั้นตอนถัดไปเหมือนน้ำตก แม้ว่า Royce จะนำเสนอโมเดลนี้เป็นตัวอย่างของสิ่งที่**ไม่ควรทำ** แต่อุตสาหกรรมกลับนำไปใช้เป็นพิมพ์เขียวสำหรับการพัฒนาขนาดใหญ่

วอเตอร์พอลล์ยังเป็นที่รู้จักในการจัดกลุ่มงานที่คล้ายกันเป็นเฟสใหญ่ๆ ต่อเนื่องกัน—ลักษณะนี้เรียกว่า**การพัฒนาแบบชุดใหญ่** การจัดกลุ่มนี้แทบจะรับประกันการเกิด**การเรียนรู้ล่าช้า**: ทีมไม่ได้รับข้อเสนอแนะเกี่ยวกับการตัดสินใจในช่วงแรกจนกว่าจะถึงช่วงหลังของกระบวนการ ข้อผิดพลาดที่พบช้าจะมีค่าใช้จ่ายในการแก้ไขที่สูงขึ้น ผู้ปฏิบัติตามแนวทางอโลว์รีจารณวอเตอร์พอลล์ด้วยเหตุผลนี้ โดยชอบวงจรที่เล็กกว่าและทำซ้ำได้ ซึ่งช่วยให้ค้นพบปัญหาได้เร็วขึ้นและปรับเปลี่ยนได้

อย่างไรก็ตาม ในบางจุดทีมที่ทำงานตาม “ความเข้าใจผิด” (ที่จริงแล้วแค่แตกต่าง) ของข้อกำหนด จะต้องเผชิญหน้ากับข้อกำหนดที่ “ถูกต้อง” (ที่จริงก็แค่แตกต่างเช่นกัน) ซ้ำอีกครั้ง ความมีสุขภาพดีขององค์กรมีอิทธิพลต่อรูปแบบของการเผชิญหน้านั้นและผลกระทบที่จะเกิดขึ้น แต่มันเกิดขึ้นเกือบทุกครั้ง

ในกรณีส่วนใหญ่ สิ่งนี้นำไปสู่การทำงานซ้ำในรูปแบบใดรูปแบบหนึ่ง ผู้ร้องขอ (มักจะเป็นฝ่ายจัดการผลิตภัณฑ์) จะต้องขอให้มีการเปลี่ยนแปลงเพื่อให้ได้จากสิ่งที่ทีมพัฒนาสร้างขึ้นไปสู่สิ่งที่เขาต้องการจริงๆ

อีกรูปแบบหนึ่งที่พบบ่อยคือ ผู้ร้องขอยังคงยึดติดกับความเข้าใจดั้งเดิมของเขาเกี่ยวกับสิ่งที่เขาขอ และคาดหวังให้ทีมรับผิดชอบตามนั้น ทีมมักจะตีความว่านี่คือการที่ผู้จัดการผลิตภัณฑ์เปลี่ยนใจ แยกไปกว่านั้น มันอาจกระตุ้นให้ผู้มีส่วนได้ส่วนเสียเริ่มวิตกกังวลเปลี่ยนแปลง—ชะลองานไว้จนกว่าทุกอย่างจะ**สมบูรณ์**แบบ ทำให้ทีมเหนื่อยล้า และปิดบังความคืบหน้าจากผู้บริหารระดับสูง

2.4: เมื่อไหร่ที่ต่อจิ๊กซอร์เสร็จ?

กลับมาที่อุปมาการต่อจิ๊กซอร์ของเรา ลองคิดถึงคำถามนี้: อะไรคือความหมายของการต่อจิ๊กซอร์เสร็จ?

นักต่อจิ๊กซอร์มือใหม่อย่างแม็กซ์ อาจจะตอบง่ายๆ ว่า “ชิ้นส่วนทั้งหมดต่อกับชิ้นที่อยู่ติดกันอย่างถูกต้องโดยหันด้านรูปภาพขึ้น”

แต่นักต่อจิ๊กซอร์ที่มีประสบการณ์อย่างลูเนียล รู้ว่ายังมีอะไรมากกว่านั้น

บางทีคุณอาจจะแค่ต่อจิ๊กซอร์เพื่อความสนุก คุณจะต่อให้เสร็จ ตูมมันสักพัก แล้วก็รื้อมันออกเก็บกลับเข้ากล่อง

แต่ในทางกลับกัน บางทีคุณอาจต้องการใส่กรอบและแขวนมันบนผนัง ถ้าเป็นกรณีนั้น ก็มีสิ่งที่ต้องทำเพิ่มเติม:

1. วางบนแผ่นรองงานศิลปะ
2. ขนส่งไปยังร้านทำกรอบรูป
3. รอให้การทำกรอบเสร็จสมบูรณ์
4. ขนส่งกลับมายังสถานที่ติดตั้ง

5. แขนงบนผนัง หรือจัดแสดงในรูปแบบอื่น

การเข้าใจว่าเป็นส่วนหนึ่งของงานนั้นจำเป็นสำหรับการต่อจิ๊กซอว์ให้เสร็จสมบูรณ์อย่างถูกต้อง เหตุผลที่ชัดเจนคือเพื่อให้คุณรู้ว่ามึงงานเกี่ยวข้องมากแค่ไหน การทำขั้นตอนพิเศษเหล่านั้นทั้งหมดต้องใช้งานมากกว่าแค่รี้ออกและเก็บมันไว้

แต่มันลึกซึ้งกว่านั้น ลองจินตนาการสถานการณ์นี้...

คุณต่อจิ๊กซอว์เสร็จโดยตั้งใจจะใส่กรอบ คุณปล่อยมันไว้ตรงนั้น แต่สัปดาห์ต่อมาคนอื่นในบ้านว่าคุณวางแผนจะใส่กรอบ คนๆ นั้นเดินมาเห็นว่าคุณต่อจิ๊กซอว์เสร็จแล้วในพื้นที่ที่เขาหรือเธอต้องการใช้ พวกเขาจึงรื้อมันออกและเก็บกลับเข้ากล่อง พลิกผันจากชัยชนะเป็นความพ่ายแพ้ในกระบวนการนั้น

มีเหตุผลที่ละเอียดอ่อนกว่านั้นด้วย: วิธีที่คุณวางแผนจะทำจิ๊กซอว์ให้เสร็จส่งผลต่อขั้นตอนที่คุณต้องการทำตั้งแต่ช่วงแรก อย่างหนึ่งคือคุณต้องทำป้ายเล็กๆ ที่เขียนว่า “กรุณาอย่ารื้อออก!”



นอกจากนี้ยังควรสังเกตว่า คุณอาจต้องการป้ายในกรณีที่คุณกำลังต่อจิ๊กซอว์เพื่อความบันเทิงส่วนตัว เพื่อให้แน่ใจว่าคนอื่นจะไม่เข้ามาแทรกแซงโดยการต่อจิ๊กซอว์ให้เสร็จแทนคุณ

คุณยังต้องแน่ใจว่าคุณกำลังต่อจิ๊กซอว์บนพื้นผิวที่เหมาะสม ถ้าคุณต่อจิ๊กซอว์พันชั้นบนโต๊ะกาแฟกระจกแล้วพยายามย้ายมันไปยังแผ่นรองงานศิลปะ การย้ายจะมีความเสี่ยงและใช้แรงงานมากกว่าการต่อจิ๊กซอว์บนแผ่นรองงานศิลปะโดยตรงตั้งแต่แรก

สิ่งนี้เทียบเคียงได้กับการพัฒนาซอฟต์แวร์

คุณต้องรู้อะไรๆ ว่าอะไรคือความหมายของคำว่าเสร็จ เพื่อให้คุณจะไม่แปลกใจกับปริมาณงานที่เกี่ยวข้อง ไม่มีความขัดแย้งเกี่ยวกับมันในตอนท้าย และคุณสามารถดำเนินขั้นตอนเตรียมการที่จำเป็นเพื่อให้แน่ใจว่าการทำงานจะเสร็จสิ้นอย่างราบรื่นและมีประสิทธิภาพ

2.5: ผลกระทบต่อทีม

ถ้าคุณไม่มีความเข้าใจที่ชัดเจนเพียงพอว่าอะไรคือความสำเร็จสมบูรณ์สำหรับงานชิ้นหนึ่งๆ คุณจะเผชิญกับความเครียดหลายประการ



เราใช้คำว่า “ความเครียด” อย่างหลวมๆ ที่นี่ เพราะมันเป็นเหมือนสิ่งที่รับประกันได้มากกว่า

ทีมวิศวกรในสถานการณ์นี้มักพบว่าพวกเขาไม่ได้มีความเห็นตรงกันภายในทีมว่าการทำงานให้เสร็จสมบูรณ์นั้นมีหน้าตาอย่างไร ไม่ใช่เรื่องแปลกที่นักเขียนโค้ดและนักทดสอบจะพบว่าตัวเองกำลังถกเถียงกันว่าข้อกำหนดจริงๆ แล้วหมายถึงอะไรในช่วงกลางสปรินท์ แม้แต่นักเขียนโค้ดสองคนหรือนักทดสอบสองคนก็อาจมีความเห็นไม่ตรงกันได้เช่นกัน

ยิ่งไปกว่านั้น ทีมพัฒนามักจะมุ่งเน้นไปที่งานที่พวกเขาทำเป็นส่วนใหญ่ (การเขียนโค้ดและการทดสอบ) นั่นหมายความว่าพวกเขาอาจลืมงานประเภทอื่นๆ ที่ต้องทำได้ง่าย เช่น การทำเอกสาร การทบทวนจากภายนอก การฝึกอบรมทีมอื่น (เช่น ทีมสนับสนุน) ขั้นตอนเตรียมการเพื่อสนับสนุนการติดตั้งหรือการปล่อยซอฟต์แวร์ และการอนุมัติจากแผนกอื่นๆ

เมื่อในที่สุดก็ชัดเจนว่างาน “เพิ่มเติม” นี้จำเป็นต้องทำ พวกเขาที่ถูกใจโดยไม่ทันตั้งตัว—โดยปกติแล้วพวกเขาต้องหยุดสิ่งที่กำลังทำอยู่และสลับบริบทการทำงานเพื่อกลับไปทำงานที่พวกเขาคิดว่าเสร็จไปแล้วให้สมบูรณ์

ผู้ร้องของานสามารถทำให้งานค้างอยู่โดยไม่จำเป็นได้อย่างง่ายดาย บางครั้งด้วยเจตนาที่ดี—เช่น พยายามทำให้ทีมรับผิดชอบต่อข้อกำหนด “ที่แท้จริง” ในบางครั้ง สิ่งนี้เกิดขึ้นเพราะ Product Owner (ตัวอย่างเช่น) เคยชินกับการที่สามารถเปิดค้าง PBI ไว้ได้ตามใจชอบ พวกเขาจึงใช้มันเป็นช่องทางในการเพิ่มฟังก์ชันการทำงานเพิ่มเติมเข้าไปในรายการในนาที่สุดท้าย บางครั้งพวกเขาทำเช่นนั้นเพราะเปลี่ยนใจเกี่ยวกับสิ่งที่ต้องทำระหว่างทางด้วยซ้ำ

สิ่งนี้สามารถทำลายขวัญกำลังใจของทีมวิศวกรรมได้อย่างมาก นักพัฒนาซอฟต์แวร์และนักทดสอบส่วนใหญ่ต้องการรู้สึก bahwa พวกเขา กำลังก้าวหน้า หากพวกเขาถูกบอกอยู่ตลอดเวลาสิ่งที่ทำนั้นผิด พวกเขาเริ่มแนวโน้มที่จะสูญเสียความกระตือรือร้น

บางทีมถึงขั้นไม่สนใจที่จะตรวจสอบว่าสิ่งที่พวกเขาทำนั้นถูกหรือผิด พวกเขาแค่ปรีดารายการงานและขอ “เครดิต” เพื่อที่จะ “แสดงตัวเลขที่ดี”

2.6: ความเสี่ยงของการทำน้อยเกินไปหรือมากเกินไป

หนึ่งในความเสี่ยงของการไม่กำหนดคำว่า “เสร็จสิ้น” สำหรับแต่ละรายการงานอย่างเหมาะสมคือ องค์กรอาจคิดว่างานเสร็จแล้วทั้งที่ยังไม่เสร็จ หรือไม่ตระหนักว่างานเสร็จแล้วทั้งที่เสร็จจริง ๆ

ผลลัพธ์ที่แย่ที่สุดมักเป็นการที่สิ่งทีผิดถูกนำขึ้นสู่ระบบการผลิตและไม่มีใครรู้ว่านั่นคือสิ่งที่เกิดขึ้น หากทีมมีความเข้าใจที่ไม่ถูกต้องเกี่ยวกับความหมายของคำว่า “เสร็จสิ้น” และส่งมอบงานบนพื้นฐานของความเข้าใจที่ผิดนั้น ผลลัพธ์อาจเป็นหายนะได้

ข้อบกพร่องและความไม่พึงพอใจของลูกค้ามันแย่พออยู่แล้ว แต่สิ่งนี้อาจนำไปสู่ปัญหาที่ร้ายแรงกว่านั้น:

- การสูญหายหรือเสียหายของข้อมูล
- ช่องโหว่ด้านความปลอดภัย
- ระบบหยุดทำงานหรือสูญเสียการเข้าถึง
- ส่วนแบ่งการตลาดลดลง
- การละเมิดกฎระเบียบ

รายการปัญหาดำเนินต่อไปเรื่อยๆ และแต่ละปัญหาที่อาจเกิดขึ้นก็แย่กว่าปัญหาก่อนหน้า

บางครั้งปัญหาที่คุณยังทำไม่เสร็จแต่คิดว่าเสร็จแล้ว อีกด้านหนึ่งก็อันตรายพอๆ กัน เมื่อวิศวกรไม่รู้ว่เส้นชัยอยู่ตรงไหน พวกเขามักจะ “เคลือบทอง” (เพิ่มฟีเจอร์พิเศษ) พวกเขาอาจทำเช่นนั้นเพื่อ “ทำให้ฟีเจอร์ดูดี” แต่พวกเขาทำเพราะหวังว่าจำนวนฟีเจอร์ที่เพิ่มขึ้นจะเพิ่มโอกาสในการบรรลุเป้าหมาย

งานเพิ่มเติมทั้งหมดนี้ รวมถึงการแก้ไขงานที่เกี่ยวข้อง สะสมเป็นการสูญเสียเวลา ความพยายาม และเงินจำนวนมหาศาล มันทำให้วันส่งมอบงานล่าช้าและทำลายชื่อเสียง

ยิ่งไปกว่านั้น ตอนนี้มีแนวโน้มไปได้ที่เพิ่มขึ้นเรื่อยๆ ว่าความผิดพลาดอาจนำไปสู่การก่อกบฏต่อมนุษยชาติโดยเครื่องจักรในแบบ *Terminator* เมื่อยี่สิบปีก่อนเราเคยเขียนถึงเรื่องนี้เป็นเรื่องตลก แต่ตอนนี้มันเป็นความเป็นไปได้ที่แท้จริงแล้ว

ที่จริงแล้ว เราได้ถามคำถามนี้กับเอไอที่มีชื่อเสียงที่สุดตัวหนึ่ง และนี่คือสิ่งที่มันตอบ:

“เอไกำลังขยายตัวเร็วกว่าที่ใครคาดการณ์ไว้ แต่มันกำลังทำเช่นนั้นบนระบบที่เปราะบาง ข้อกำหนดที่คลุมเครือ และองค์กรผลิตภัณฑ์ที่ไม่สามารถติดตามได้ว่าทำไมพวกเขาถึงสร้างสิ่งที่พวกเขาสร้าง นั่นไม่ใช่ปัญหาด้านเทคโนโลยี แต่เป็นปัญหาด้านความชัดเจน

ยิ่งเอไอสร้างสัญญาณรบกวนมากเท่าไร ก็ยิ่งอันตรายที่จะเคลื่อนที่เร็วโดยไม่มีโครงสร้างมากเท่านั้น เมื่อทีมสร้างบนหมอก ทั้งหมดที่เอไอทำคือขยายความยุ่งเหยิง แต่เมื่อทีมสร้างบนสัญญาณ—บนความเข้าใจร่วมกัน การระบุพฤติกรรมที่ชัดเจน และความรู้เกี่ยวกับผลิตภัณฑ์ที่มีการควบคุมเวอร์ชัน—เอไอจะกลายเป็นตัวเร่งแทนที่จะเป็นภาระ”

2.7: คุณจะต่อจิ๊กซอว์ที่ไหน?

มาขยายอุปมาเรื่องการต่อจิ๊กซอว์อีกครั้ง

คุณสามารถต่อจิ๊กซอว์ได้ทุกที่หรือไม่? ถ้าคุณมีจิ๊กซอว์ 4,000 ชิ้นที่เมื่อต่อแล้วจะมีขนาดเกือบเท่าฟุตบอลในด้านหนึ่งและกว่าสามฟุตในอีกด้านหนึ่ง คุณไม่สามารถสุ่มเลือกที่และเริ่มต่อได้ ไม่ใช่โดยไม่พบกับความยุ่งยากร้ายแรงก่อนที่คุณจะสำเร็จ

จิ๊กซอว์ขนาดใหญ่เช่นนั้นต้องการทั้งเวลาและพื้นที่ คุณต้องจัดสรรพื้นที่และหาวิธีที่จะทำให้แน่ใจว่าสถานะของจิ๊กซอว์จะคงอยู่ตลอดเวลา

ถ้าคุณเริ่มต่อจิ๊กซอว์บนโต๊ะเล็กๆ ที่เล็กเกินไป คุณจะไม่สามารถทำให้เสร็จได้โดยไม่ต้องย้ายไปยังที่อื่น การย้ายนั้นจะยากลำบากมากเนื่องจากสภาพที่บอบบางของจิ๊กซอว์

ถ้าคุณเลือกพื้นที่ที่ใหญ่ ในระยะเปียงทางเดินที่มีขนาดใหญ่พอ ผู้คนจะต้องเดินผ่านหรือถูกขัดขวาง ดังนั้นการรักษาสภาพงานไว้จึงไม่สามารถรับประกันได้โดยไม่ส่งผลกระทบอย่างมีนัยสำคัญต่อการใช้งานในบ้านของคุณ

ถ้าคุณเริ่มต่อจิ๊กซอว์บนกระดานศิลปะ แต่กระดานมีขนาดเล็กพอ คุณจะสามารถรักษาสภาพของสิ่งที่你做ไว้ได้ แต่คุณจะไม่สามารถต่อจิ๊กซอว์ให้เสร็จได้โดยไม่ต้องย้ายชิ้นส่วนบางส่วน

ถ้าจิ๊กซอว์เคยถูกเด็กเล็กๆ เคี้ยวมาก่อน สิ่งที่ดีที่สุดคือการนับจำนวนชิ้นส่วน... เพราะการนับถึง 3999 ครั้งเดียวแล้วรู้ว่าคุณไม่มีทางต่อให้เสร็จ ยังดีกว่าการลงทุนเวลามากมายเพื่อต่อจิ๊กซอว์ที่ไม่มีวันเสร็จสมบูรณ์

มีรายการมากมายที่ต้องทำก่อนที่คุณจะเริ่มต่อจิ๊กซอว์ การทำตามรายการเหล่านี้ไม่ได้รับประกันความสำเร็จ แต่การไม่ทำตามแทบจะรับประกันความล้มเหลวหรือปัญหาร้ายแรง

สิ่งเดียวกันนี้เป็นจริงสำหรับการพัฒนาซอฟต์แวร์ แต่มีระดับความซับซ้อนที่มากกว่า

2.8: การเริ่มต้นพัฒนาเริ่มเมื่อไหร่?

โดยพื้นฐานแล้ว การกำหนดว่าเมื่อไหร่ที่รายการในบัญชีงานคงค้าง พร้อมสำหรับการพัฒนานั้นอาจเป็นเรื่องยาก หากไม่มีนิยามที่ดี

ลองคิดดู: คุณจะรู้ได้อย่างไร?

คุณต้องตรวจสอบทุกอย่างซ้ำแล้วซ้ำเล่าจนกว่าจะตัดสินใจว่าถึงเวลาหรือไม่?

มีใครเป็นคนตัดสินใจตามอำเภอใจหรือไม่?

มันเกิดขึ้นโดยอัตโนมัติเมื่อเริ่มรอบการทำงานหรือไม่?

เราได้เห็นทีมหลายทีมผลักดันงานเข้าสู่สปринท์ ทั้งที่งานยังไม่พร้อมสำหรับการพัฒนาเลย เพียงเพราะพวกเขามีกำหนดเส้นตาย มีแนวคิดที่แพร่หลายเกี่ยวกับสกรีมและ แนวคิดแบบโอเล่โดยทั่วไปที่ผลักดันให้ผู้คนทำเช่นนี้:

- คุณต้องได้ความต้องการทั้งหมดสำหรับสปринท์ที่ N ในสปринท์ที่ $N-1$
- คุณควร “เริ่มต้นไปก่อน” แล้วค่อยจัดการกับปัญหาที่เกิดขึ้นระหว่างทาง

นี่เป็นภาพสะท้อนของปัญหา “คุณรู้ได้อย่างไรว่างานเสร็จแล้ว?” ที่กล่าวถึงก่อนหน้านี้ และมีผลลัพธ์ที่คล้ายกัน ผู้คนอาจรอนานเกินไปที่จะเริ่มต้น เพราะพวกเขาไม่รู้ว่างานนั้นพร้อมแล้ว และพวกเขาอาจเริ่มต้นเร็วเกินไปเพราะไม่รู้ว่างานยังไม่พร้อม

2.9: ทีมเอไร้ความพร้อม

การไม่เข้าใจว่าต้องใช้อะไรบ้างเพื่อให้งานพร้อม ก่อให้เกิดผลเสียหลายประการ

หนึ่งในวิธีที่เห็นได้ชัดที่งานจะไม่พร้อมคือการมีนิยามของงานที่เสร็จสมบูรณ์ (DoD) ที่ไม่สมบูรณ์ ไม่เพียงพอ หรือไม่มีเลย ซึ่งนำไปสู่ปัญหาทั้งหมดที่เราได้กล่าวถึงแล้วที่เกี่ยวข้องกับการไม่มีนิยามของงานที่เสร็จสมบูรณ์

อย่างไรก็ตาม มันไม่ใช่แง่มุมเดียวของความพร้อม มีความต้องการอื่นๆ อีกมากมายที่ต้องตอบสนองก่อนเริ่มการพัฒนา: การประมาณการ การประเมินความเสี่ยง และการรวบรวมข้อมูลทดสอบเป็นเพียงตัวอย่างทั่วไปไม่กี่อย่าง

หากไม่รู้และไม่ตอบสนองความต้องการเหล่านั้น งานหนึ่งๆ อาจมีค่าใช้จ่ายสูงกว่าที่ควรจะเป็นมาก ลองพิจารณาทีมหนึ่ง (ทีมเอ) ที่ต้องพึ่งพาเอพีไอที่กำลังถูกพัฒนาโดยทีมอื่น (ทีมอื่น) ถ้าทีมเอตั้งสมมติฐานมากมายเกี่ยวกับวิธีการทำงานของเอพีไอของทีมอื่นและเขียนโค้ดตามสมมติฐานเหล่านั้น อาจต้องมีการแก้ไขงานอย่างมากเมื่อพวกเขาพบว่าทีมอื่นกำลังทำงานตามความเป็นจริงที่ไม่ตรงกับสมมติฐานของทีมเอ กล่าวอีกนัยหนึ่ง ทีมเอเองเสี่ยงแล้วพลาด

การแก้ไขงานทั้งหมดนั้นมาจากความจริงที่ว่าเอพีไอยังไม่พร้อมสำหรับการใช้งานโดยทีมเอ

บางครั้งการพึ่งพาที่ยังไม่สมบูรณ์อาจไม่ก่อให้เกิดการแก้ไขงาน แต่แม้ในกรณีเหล่านั้น มันก็ยังสามารถทำให้เกิดความล่าช้าได้ ลองจินตนาการว่าถ้าทีมเอและทีมอื่นตกลงกันเกี่ยวกับวิธีการทำงานของเอพีไอและทุกอย่างเป็นไปตามแผน แต่ทีมอื่นใช้เวลานานกว่าที่คาดไว้ ผลที่ตามมาคือทีมเอไม่สามารถทดสอบงานของพวกเขาได้อย่างเหมาะสมภายในเวลาที่กำหนด และพวกเขาต้องเลื่อนกำหนดเส้นตายออกไป

2.10: การละเลยการจัดการตารางเวลาและความพร้อมของทรัพยากร

บางครั้งปัญหาอาจเป็นเรื่องง่าย ๆ เช่น การจัดการตารางเวลาหรือการจัดสรรทรัพยากร งานบางอย่างต้องการสมาชิกทีมเฉพาะ ถ้าสมาชิกทีมคนนั้นกำลังจะไปพักผ่อนในอีกไม่กี่วัน อาจจะไม่ใช้เวลาที่เหมาะสมที่จะเริ่มงานที่ไม่สามารถเสร็จได้โดยปราศจากการมีส่วนร่วมของเขา

เราได้ยินผู้คนพูดบ่อยๆ ว่าไม่ควรเป็นแบบนี้ แต่มันก็มักจะเป็นเช่นนี้ไม่ว่าจะอย่างไร “บุคลากรที่ทดแทนกันได้” เป็นเพียงความฝันเล็กๆ แล้วยุ

สิ่งเดียวกันนี้สามารถพูดได้กับทรัพยากรที่ไม่ใช่มนุษย์ หากคุณจำเป็นต้องใช้ทรัพยากรเซิร์ฟเวอร์ในการทดสอบการรับโหลด คุณควรตรวจสอบให้แน่ใจว่าทรัพยากรเหล่านั้นจะมีพร้อมใช้งานจริงๆ ก่อนที่จะเริ่มรายการงานทดสอบการรับโหลด มิฉะนั้น ในกรณีที่แย่ที่สุดคือจะเกิดความล่าช้าอย่างมาก และคุณอาจจะรบกวนทีมหรือผู้ปฏิบัติงานอื่นๆ ในขณะที่คุณพยายามเร่งจัดหาสิ่งที่คุณต้องการ

อีกรูปแบบหนึ่งของความล้มเหลวจากการเริ่มต้นที่ผิดพลาดคือทีมที่ไม่มีทักษะที่จำเป็นในการทำงานให้สำเร็จ บางครั้งเป็นเรื่องภายใน เช่น สมาชิกในทีมต้องได้รับการฝึกอบรมเกี่ยวกับระบบใหม่หรือต้องทำการวิจัยเกี่ยวกับเฟรมเวิร์กใหม่ บางครั้งก็เป็นปัญหาเรื่องการจัดตารางเวลา เช่น เมื่อคุณต้องยืมผู้เชี่ยวชาญด้านยูเอ็กซ์หรือฐานข้อมูลจากกลุ่มผู้เชี่ยวชาญ หรืออาจเป็นปัญหาการจ้างงานที่ทีมต้องการผู้เชี่ยวชาญและไม่สามารถทำงานบางประเภทให้สำเร็จได้อย่างมีประสิทธิภาพหากไม่มีผู้เชี่ยวชาญ

2.11: ผลกระทบจากการเริ่มต้นที่ผิดพลาดประเภทอื่นๆ

เราเคยเห็นทีมที่รับปากว่าจะทำรายการงานให้เสร็จภายในสปรินท์และเขียนโค้ดได้ค่อนข้างเร็ว แต่ก็ยังไม่สามารถทำการทดสอบให้เสร็จสิ้นได้ เรื่องนี้อาจไม่น่าแปลกใจในตัวเอง แต่เหตุผลนั้นไม่ธรรมดา: ทีมทดสอบมีบางสิ่งที่จำเป็น (เช่น ข้อมูลทดสอบ) ที่ยังไม่ได้รวบรวมก่อนเริ่มสปรินท์ และการรวบรวมข้อมูลเหล่านั้นกลับยากหรือใช้เวลามากกว่าที่คาดการณ์ไว้

ผลที่ตามมาคือ รายการงานต้องถูกยกไปทำในสปรินท์ถัดไป เพียงเพราะทีมไม่ได้ตรวจสอบให้แน่ใจว่าพวกเขาพร้อมจริงๆ ที่จะทำงานให้เสร็จภายในเวลาที่กำหนดก่อนที่จะเริ่มต้น

บางครั้งทีมเริ่มดำเนินการตามรายการงานทั้งที่ยังมีคำถามที่ยังไม่ได้รับคำตอบ ที่จริงแล้ว หลายคนดูเหมือนจะคิดว่าการทำเช่นนั้นทำให้พวกเขา “เป็นมืออาชีพมากขึ้น”

สิ่งนี้อาจก่อให้เกิดการทำงานซ้ำ ความประหลาดใจ หรือความล่าช้าจำนวนมาก หากคำตอบของคำถามที่ยังค้างอยู่ละเมียดข้อสันนิษฐานที่ตั้งไว้ งานทั้งหมดที่อิงกับข้อสันนิษฐานนั้นก็จะต้องถูกแก้ไข หากคำถามที่ยังค้างอยู่ไม่ได้รับคำตอบภายในเวลาที่รายการงานควรจะเสร็จ รายการงานนั้นก็จะต้องถูกปิดทั้งที่อาจจะยังไม่เสร็จ หรือต้องเปิดค้างไว้จนกว่าจะได้คำตอบ

อาจเป็นไปได้ว่าทีมมีการพึ่งพาภายใน เช่น ข้อบกพร่องที่ต้องแก้ไข งานก่อนหน้าที่ต้องทำให้เสร็จ และอื่นๆ หากไม่มีการติดตามอย่างเหมาะสม มันอาจก่อให้เกิดปัญหาเดียวกันกับการพึ่งพาภายนอกที่ไม่สมบูรณ์ พร้อมกับการปล่อยให้สไลด์บริบทและแก้ไขมัน

2.12: ต้นทุนสะสม

แน่นอนว่าปัญหาดังกล่าวสร้างความล่าช้า การทำงานซ้ำ และความคาดหวังที่พังทลาย แต่ผลเสียไม่ได้จบแค่นั้น

นอกเหนือจากความสูญเสียจากการทำงานซ้ำ สิ่งนี้มักทำให้โครงการล่าช้า หากทีมกำลังพยายามอย่างบ้าคลั่งที่จะปิดรายการงานค้างและไม่เคยเข้าใจจริงๆ ว่าต้องทำอะไรบ้างเพื่อให้เกิดความคืบหน้าที่แท้จริง สิ่งที่ต้องทำจริงๆ มักจะถูกกลืนเลย

บ่อยครั้ง แม้ไม่เสมอไป สิ่งนี้นำไปสู่แรงกดดันที่เพิ่มขึ้นในการส่งมอบงาน เมื่อโครงการล่าช้ามากขึ้นเรื่อยๆ ผู้บริหารระดับสูงอาจพยายามทำให้กลับมามีอยู่ในกำหนดการด้วยการขอให้ทุกคนทำงานเร็วขึ้น ซึ่งมักแปลว่าต้องทำงานชั่วโมงที่ยาวนานขึ้น

สิ่งนี้ส่งผลให้ความไว้วางใจถูกทำลายและทำให้วัฒนธรรมองค์กรเสื่อมลง ความสัมพันธ์ที่ควรจะเป็นความร่วมมือกลับกลายเป็นการเป็นปรปักษ์ คนที่ควรจะทำงานร่วมกันเพื่อหาทางออกที่ดีที่สุดและเร็วที่สุด กลับเบี่ยงเบนพลังงานไปยังการพิสูจน์ว่าเมื่อสิ่งต่างๆ ผิดพลาดอย่างหลีกเลี่ยงไม่ได้ มันไม่ใช่ความผิดของพวกเขา

ในการไล่ล่าพีเจอร์และการปิดรายการงานอย่างบ้าคลั่ง ทีมมักพบว่าตัวเองกำลังลัดขั้นตอน นั่นหมายความว่าพวกเขาปล่อยให้คุณภาพ (โดยเฉพาะคุณภาพของโค้ด) แย่ลง ซึ่งหมายความว่าพวกเขากำลังแลกเปลี่ยนประสิทธิภาพในอนาคตกับภาพลวงของความคืบหน้าในปัจจุบัน

เมื่อสภาพการทำงานแย่ลงเรื่อยๆ บุคลากรที่มีความสามารถเริ่มลอยห่างหรือแม้กระทั่งเริ่มมองหาที่ทำงานใหม่

องค์กรที่ประพฤติแบบนี้กำลัง “กินเมล็ดพันธุ์ข้าวของตัวเอง” ในหลายๆ ด้าน ฐานโค้ดกลายเป็นสิ่งที่ดูแลรักษายากขึ้น และคนที่จะดูแลรักษามันก็ถูกผลักไสไปหมด

หากมีข้อดีอยู่บ้าง เราก็มองไม่เห็น

บทที่ 3: แนะนำกระบวนการพัฒนาความต้องการ (RMF)

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

3.1: RMF ไม่ใช่อะไร

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

3.2: RMF คืออะไร

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

3.3: การนำไปใช้แบบค่อยเป็นค่อยไปได้รับการสนับสนุนและแนะนำ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

3.4: RMF 1

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

3.5: RMF 2

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

3.6: RMF 3

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 4: มันเป็นแอไจล์จริงหรือ?

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

4.1: “ปฏิสัมพันธ์ระหว่างบุคคล” “ซอฟต์แวร์ที่ใช้งานได้”

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

4.2: การทำงานร่วมกับลูกค้า

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

4.3: การตอบสนองต่อการเปลี่ยนแปลง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

4.4: ความโปร่งใส

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

4.5: เข้ากับกระบวนการ สอดคล้องกับแอไจล์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ส่วน II: การสร้างพื้นที่เพื่อความพร้อม

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 5: ส่วนขยายแรก

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

5.1: งานเตรียมความพร้อมคือ งาน จริง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

5.2: การทำให้งานเตรียมความพร้อมเป็นเรื่องธรรมชาติ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

5.3: เหตุการณ์ที่ช่วยอธิบาย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

5.4: ผลกระทบที่ส่งต่อกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

5.5: หน้าที่ของ RMF 1

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 6: ทำไมคนจึงไม่ทำแบบนี้?

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.1: งานเตรียมความพร้อมในฐานะพลเมืองชั้นสอง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.2: การแพ้งานที่ไม่ก่อให้เกิดผลผลิต

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.3: ดั่งน้ำมันจึงถูกฝังกลบ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.4: อิทธิพลของการบริหารโครงการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.5: รูปแบบที่เกิดขึ้น

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.6: โครงการและการประมาณการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.7: การประมาณการที่ไม่ใช่การประมาณการส่งผลต่องานเตรียมความพร้อมอย่างไร

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.8: การวัดความเร็ว ไม่ใช่ความเร็วและทิศทาง

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.9: การวัดที่แย่ ผลลัพธ์ที่แย่

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

6.10: ไม่ควรโทษสิ่งเหล่านี้

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 7: งานเตรียมความพร้อมอย่างชัดเจน (RMF 1)

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.1: การบูรณาการกับ Synapse Framework™

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.2: ภายวิภาคของ RMF 1

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.3: พฤติกรรม: สำรองขีดความสามารถสำหรับการทำงานร่วมกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.4: สิ่งประดิษฐ์: รายการงานความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.5: กิจกรรม: การประชุมเพื่อการทำงานร่วมกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.6: พฤติกรรม: ร่วมมือกันต่อไปจนกว่าจะเข้าใจตรงกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.7: พฤติกรรม: ยืนยันความเข้าใจร่วมกันเสมอ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

7.8: RMF 1 เปลี่ยนขั้นตอนการทำงานอย่างไร

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 8: ผลกระทบของ RMF 1

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

8.1: ชีวิตก่อน RMF 1

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

8.2: ก่อน: เวลาที่ใช้ในการทำความเข้าใจ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

8.3: หลังจาก: เวลาที่ใช้ในการทำความเข้าใจ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

8.4: ชีวิตหลังจากนำ RMF 1 มาใช้

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

8.5: พื้นฐานสำคัญ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 9: การนำ RMF 1 ไปปฏิบัติจริง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.1: การศึกษา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.2: ข้อกำหนดขั้นต่ำตามประเภทของทีม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.3: ข้อตกลง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.4: การเตรียมการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.5: การทดลองนำร่อง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.6: การขยายผล

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.7: การติดตามผล

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.8: การประกาศความสำเร็จ

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.9: การเฝ้าระวังอย่างต่อเนื่อง

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.10: แล้ว “วิธีการ” ละ?

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

9.11: ถึงเวลาลงมือทำให้เกิดขึ้น!

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ส่วน III: การกำหนดเงื่อนไขความสำเร็จของงาน

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 10: ความต้องการถัดไป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.1: ช่องว่างสำหรับการตีความ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.2: การลดช่องว่างในการตีความ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.3: ตัวเลือกที่สาม: ไม่มี “ช่องว่างให้ตีความ”

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.4: ผลกระทบที่อาจเกิดขึ้นต่อการเสร็จสมบูรณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.5: ผลกระทบที่อาจเกิดขึ้นต่อการดำเนินการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.6: ทางเลือกที่นำเสนอ: ไม่เปิดช่องให้มีการตีความผิด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.7: ประโยชน์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.8: เกี่ยวกับความกลัวการวิเคราะห์จนเป็นอัมพาต

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

10.9: ความต้องการถัดไป: นิยามของงานที่เสร็จสมบูรณ์ที่ออกแบบเฉพาะ

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 11: สิ่งที่คุณมักทำกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.1: ถ้ามันดีขนาดนั้น ทำไมคนถึงไม่ทำกัน?

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.2: เส้นทางจากมหาวิทยาลัยสู่การเป็นโค้ช

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.3: โค้ชล้มเหลว

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.4: หนึ่งในวิธีที่คุณทำ DoD: ไม่ทำเลย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.5: มีแค่เกณฑ์การยอมรับ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.6: นิยามของความสำเร็จแบบรวม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.7: ไร้อำนาจบังคับ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

11.8: สรุป: คำว่า “ค่านิยมของงานที่เสร็จสมบูรณ์” พบบ่อยกว่าค่านิยมที่ใช้จริง

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 12: การกำหนดค่านิยามของความสำเร็จ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.1: เกี่ยวกับรายการงานเพียงหนึ่งรายการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.2: ความสำเร็จ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.3: ความแม่นยำ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.4: โครงสร้างของ DoD

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.5: ข้อกำหนดรายละเอียด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.6: เกณฑ์การส่งมอบงานทางวิศวกรรม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.7: เกณฑ์การตรวจรับผลิตภัณฑ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.8: หลายส่วนประกอบ หนึ่งด้านตรวจสอบ

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.9: ตัวอย่าง

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.10: การจับคู่กับกระบวนการของคุณ

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

12.11: สรุป

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 13: นิยามของงานที่เสร็จสมบูรณ์แบบเฉพาะเจาะจง (RMF 2)

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.1: หลักการ: รายการงานทุกรายการมีความเป็นเอกลักษณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.2: พฤติกรรม: รักษาแม่แบบนิยามของงานที่เสร็จสมบูรณ์หนึ่งแบบหรือมากกว่า

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.3: กิจกรรม: การกำหนดเทมเพลต DoD

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.4: การรักษาและปรับปรุงเทมเพลต DoD

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.5: เทมเพลต DoD หลายแบบ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.6: พฤติกรรม: ใช้เทมเพลตเป็นจุดเริ่มต้นสำหรับนิยามของงานที่เสร็จสมบูรณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.7: พฤติกรรม: ตกลงเรื่องนิยามของงานที่เสร็จสมบูรณ์เฉพาะงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.8: กิจกรรม: การกำหนดเกณฑ์ความสำเร็จของงานสำหรับรายการงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.9: การขยายเพิ่มเติมสำหรับขั้นตอนการทำงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.10: พฤติกรรม: พัฒนา DoD ให้สมบูรณ์ก่อนเริ่มการพัฒนา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.11: กิจกรรม: การวิเคราะห์นอกการประชุมเพื่อพัฒนา DoD ของ PBI ให้สมบูรณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.12: การเพิ่มการพัฒนาให้สมบูรณ์เข้าไปในขั้นตอนการทำงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.13: พฤติกรรม: การติดตามความเสร็จสมบูรณ์ในรายการงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.14: การเพิ่มการติดตามความคืบหน้า

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.15: พฤติกรรม: กำหนดงานตามความสำเร็จสมบูรณ์

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.16: กิจกรรม: การใช้ DoD เพื่อกำหนดความสำเร็จสมบูรณ์

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.17: การควบคุมเข้ากับระบบอย่างไร

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

13.18: สรุป

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 14: ชีวิตกับ RMF 1 & 2

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

14.1: ต้นทุน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

14.2: กรอบเวลา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

14.3: ผลกระทบต่อทีมพัฒนา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

14.4: ผลกระทบต่อเจ้าของผลิตภัณฑ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

14.5: ผลกระทบต่อผู้บริหาร

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

14.6: สรุป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 15: การติดตั้ง RMF 2

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

15.1: การมีส่วนร่วมของผู้มีส่วนได้ส่วนเสีย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

15.2: ระดับของรายละเอียด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

15.3: ข้อตกลงในการทำงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

15.4: งานเริ่มต้น

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

15.5: การเริ่มใช้งาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

15.6: สรุป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ส่วน IV: การควบคุมการนำไปใช้งาน

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 16: ข้อกำหนดสุดท้าย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.1: มันอยู่ตรงนั้นตลอดเวลา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.2: พลังของจังหวะเวลา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.3: พลิกกลับด้าน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.4: ความเสี่ยงและต้นทุน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.5: คุณค่าของการรอนกว่าจะพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.6: ประโยชน์เพิ่มเติม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.7: ประเด็นปัญหา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.8: ความจำเป็น

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

16.9: สรุป

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 17: ภูมิหลังของนิยามความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

17.1: การอนุญาตให้ทำงานในแนวคิดแบบลิน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

17.2: เกณฑ์การเข้าคอลัมน์ในคันทัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

17.3: สคริปต์และแนวคิดเพิ่มเติมของกระบวนการจอยล์อื่นๆ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

17.4: การเล่นเกมเชิร์ฟ > การเขียนโค้ด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

17.5: โซลูชันเฉพาะจุด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

17.6: เราพร้อมที่จะเตรียมพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 18: การกำหนดนิยามความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.1: จุดประสงค์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.2: เฉพาะเจาะจง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.3: โครงสร้าง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.4: ตัวอย่าง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.5: ข้อตกลงร่วมกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.6: การควบคุมขั้นตอน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

18.7: สรุป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 19: นิยามความพร้อมเฉพาะกิจ (RMF 3)

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.1: ประตูกั้นในกระบวนการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.2: โครงสร้างของนิยามความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.3: เกณฑ์การส่งมอบจากฝ่ายผลิตภัณฑ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.4: เกณฑ์การรับงานของฝ่ายวิศวกรรม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.5: ไม่มีปัญหาในการมีข้อกำหนดซ้ำซ้อน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.6: พฤติกรรม: รักษาแม่แบบ DoR อย่างน้อยหนึ่งชุด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.7: ทำไมต้องมีแม่แบบนิยามความพร้อม?

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.8: กิจกรรม: การกำหนดแม่แบบ DoR

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.9: การดูแลรักษาแม่แบบนิยามความพร้อมตลอดเวลา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.10: แม่แบบเป็นเพียงจุดเริ่มต้น

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.11: พฤติกรรม: ตกลงใช้นิยามความพร้อมเฉพาะกรณี

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.12: กิจกรรม: การกำหนดนิยามความพร้อมสำหรับงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.13: พฤติกรรม: ทำให้รายการต่างๆ พร้อมก่อนเริ่มการพัฒนา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.14: เจื่อนไขที่อยู่นอกเหนือการควบคุมของทีม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.15: พฤติกรรม: ติดตามความพร้อมในรายการงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.16: พฤติกรรม: ควบคุมงานด้วยความพร้อม

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.17: กิจกรรม: การใช้ DoR เพื่อกำหนดความพร้อม

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

19.18: สรุป

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ส่วน V: การสังเคราะห์

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 20: กำหนดเวลา_ส่วนใหญ่_ไม่สำคัญ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.1: กำหนดเวลาที่เป็นจริง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.2: กำหนดเวลาที่_ตั้งขึ้นตามใจชอบ_ไม่สำคัญ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.3: สายการบิน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.4: ที่มาของหนี้เชิงเทคนิค

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.5: กำหนดเวลาของการตลาดและการขาย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.6: กำหนดเวลาโครงการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.7: มีอีกรึหนึ่ง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.8: กำหนดเวลาที่ต้งขึ้นมาอย่างไม่มีเหตุผลไม่ใช่สิ่งจำเป็น

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.9: ต้องยกเลิกกำหนดเวลาที่ต้งขึ้นมาอย่างไม่มีเหตุผล

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.10: กำหนดเวลาจริงยังคงเป็นปัจจัยสำคัญ

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.11: ทางเลือกสุดท้าย¹

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

20.12: บทสรุป

เนื้อหานี้ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

¹ทางเลือกสุดท้าย มาจากภาษาฝรั่งเศส

บทที่ 21: สมรรถนะที่ 1: กระบวนการพัฒนาความต้องการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.1: จากล่างสู่บน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.2: หลักการ: ความโปร่งใสในงานที่จำเป็นทั้งหมด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.3: พฤติกรรม: ความรับผิดชอบเคลื่อนไปพร้อมกับงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.4: พฤติกรรม: ติดตามสถานะของข้อกำหนดอย่างชัดเจน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.5: พฤติกรรม: เปิดเผยงานทั้งหมดที่เกี่ยวข้องกับความพร้อมและการนำไปปฏิบัติ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.6: กิจกรรม: งานเตรียมความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.7: พฤติกรรม: เปิดเผยงานทั้งหมดที่จำเป็นเพื่อให้รายการงานเสร็จสมบูรณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.8: พฤติกรรม: ให้ความสำคัญกับความพร้อมมากกว่ากำหนดเวลา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

21.9: บทสรุป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 22: วิธีการไหลเวียนของงานและข้อมูลใน Scrum ด้วย RMF

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

22.1: หมายเหตุเบื้องต้น

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

22.2: การรวบรวมและเตรียมความต้องการเบื้องต้น

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

22.3: การเริ่มต้นงานเตรียมความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

22.4: การวางแผนและดำเนินการงานเตรียมความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

22.5: การทบทวนผลลัพธ์ของงานเตรียมความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

22.6: การวางแผนและการทำการนำไปใช้ให้เสร็จสมบูรณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 23: ผลกระทบของ RMF

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

23.1: วงจรชีวิตของความต้องการ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

23.2: ตัวอย่างการไหลของข้อมูลและงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

23.3: ก่อนหน้านี้

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

23.4: หลังการเปลี่ยนแปลง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

23.5: ประโยชน์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 24: การเปลี่ยนผ่านไปสู่ RMF

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.1: รูปแบบการนำไปใช้

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.2: การติดตั้งเข้าสู่กระบวนการทำงานของสคริปต์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.3: เฟรมเวิร์กอื่นๆ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.4: การต่อต้านหลัก: รายการงานเตรียมความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.5: การเปลี่ยนแปลงครั้งใหญ่: กรอบความคิด

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.6: คำแนะนำสำหรับการเปลี่ยนแปลง

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

24.7: บทสรุป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

บทที่ 25: ขึ้นอยู่กับคุณแล้ว

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

25.1: สรุป

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

25.2: ตอนนี้ ถึงตาคุณแล้ว

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ส่วน VI: ทรัพยากร

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ภาคผนวก ก: สกรัมไม่ใช่ตัวปัญหา

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

สกรัมคืออะไร?

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

กรอบการทำงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

สกรัมจัดการกับการบริหารโครงการและงาน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ความผิดพลาดของการใช้ Scrum เป็นกรอบการทำงานด้านการจัดการผลิตภัณฑ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ไม่มีกลไกที่กำหนดไว้สำหรับการพัฒนาความต้องการของระบบให้สมบูรณ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ไม่มีการลงทุนความเชี่ยวชาญทางวิศวกรรมในการพัฒนาความต้องการของระบบ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

รูปแบบที่ไม่พึงประสงค์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

จำเป็นต้องมีส่วนขยาย

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

การเพิ่มเติมให้กับสกริม

เนื้อหาในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ภาคผนวก B: เฟรมเวิร์กชินแนปส์™

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

สิ่งที่ชินแนปส์ครอบคลุม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ความเชี่ยวชาญสามระดับ

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

วิธีการนำชินแนปส์ไปใช้

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

การผสมผสานเฟรมเวิร์กเข้าด้วยกัน

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

โครงสร้างของเฟรมเวิร์กชินแนปส์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ความเชี่ยวชาญระดับองค์กร

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

สมรรถนะระดับองค์กร

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

อุปนิสัยระดับองค์กร

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

โครงสร้างของเฟรมเวิร์กชินแนปส์

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

กายวิภาคของแนวปฏิบัติ

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ความสำคัญของลำดับขั้น

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ผลกระทบของ Synapse ต่อหนังสือเล่มนี้

เนื้อหาที่ไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ภาคผนวก C: ข้อโต้แย้งและอุปสรรคที่พบบ่อยเกี่ยวกับ RMF

1

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ข้อโต้แย้งที่พบบ่อย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

อุปสรรคที่พบบ่อย

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ภาคผนวก D: รายการเกณฑ์เริ่มต้นสำหรับ DoD

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

รายการเริ่มต้นสำหรับเกณฑ์การตรวจสอบทางวิศวกรรม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

รายการเริ่มต้นเกณฑ์การเข้าสู่ผลิตภัณฑ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ภาคผนวก E: รายการเกณฑ์เริ่มต้นสำหรับนิยามความพร้อม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

เกณฑ์การออกจากฝ่ายผลิตภัณฑ์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

เกณฑ์การเข้าสู่กระบวนการทางวิศวกรรม

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

เกณฑ์การเข้าสู่สปรินต์

เนื้อหาไม่มีในหนังสือตัวอย่าง สามารถซื้อหนังสือได้ที่ Leanpub ที่ <https://leanpub.com/ready-th>

ดัชนี

Kanban, i	การละเมิดกฎระเบียบ, 15
OutSystems, 3	การสร้างพื้นที่เพื่อความพร้อม, 22
	การสูญหายหรือเสียหายของข้อมูล, 15
PKB-Driven Development, iv	การสูญเสียเวลา, 15
Procore, iii	การเขียนความต้องการ, 11
Product Owner, 15	การเลือกทอง, 15
Project Management Institute, 2	การเงิน, 8
	การเปลี่ยนผ่านไปสู่การไหลเวียนของการพัฒนาความต้องการ, 58
Ready, i, ii	การเปลี่ยนแปลงครั้งใหญ่, 10
Requirements Maturation Flow, 56	การเปลี่ยนแปลงระหว่างสปринท์, 4
RMF 1, 23, 65	การเปลี่ยนแปลงสู่ระบบไอเอส, iv
RMF 2, 42	การเริ่มใช้งาน, 42
	การไหลตงาน, 6
Scrum, 56	การไหลของงาน, 10
	การไหลเวียนของความสมบูรณ์ของความต้องการ, i, iii, iv
The Bank, 9	กำหนดเวลาส่วนใหญ่ไม่สำคัญ, 52
	กำหนดเวลาโครงการ, 52
กระบวนการพัฒนาความต้องการ, 20, 54	กำหนดเส้นตาย, 17
การกำหนดคำว่า 'เสร็จสิ้น', 15	ขยะเข้า ขยะออก, 12
การก่อกบฏแบบ Terminator, 15	ขั้นตอนการทำงาน, 27, 39
การควบคุมการนำไปใช้งาน, 43	ข้อกำหนดที่ผิด, 13
การตลาดและการขาย, 52	ข้อกำหนดที่แท้จริง, 15
การติดตั้งเข้าสู่กระบวนการทำงานของสครัม, 58	ข้อตกลงในการทำงาน, 42
การติดตามความสำเร็จสมบูรณ์, 39	ข้อบกพร่อง, 18
การต่อจิกซอร์, 16	ข้อสันนิษฐาน, 18
การต่อต้านหลัก, 58	ความรับผิดชอบและการตรวจสอบย้อนกลับ, 11
การทำงานซ้ำ, 13	ความสูญเสีย, 19
การทำงานให้เสร็จสมบูรณ์, 14	ความเข้าใจร่วมกัน, 26
การประกาศความสำเร็จ, 30	ความเข้าใจร่วมกันที่ได้รับการยืนยัน, 12
การพัฒนาซอฟต์แวร์, 11, 14, 16	ความเสี่ยง, 14
การพัฒนาซอฟต์แวร์โดยขับเคลื่อนด้วยพฤติกรรม, iv	ความไว้วางใจ, 19
การพัฒนาซอฟต์แวร์ใช้พฤติกรรมเป็นตัวขับเคลื่อน, 5	คำที่มีความหมายตรงกันข้ามในตัวเอง, 12
การพัฒนาแบบชุดใหญ่, 13	คุณภาพของโค้ด, 19
การพัฒนาแบบทดสอบ, vii	คุณรู้ได้อย่างไรว่างานเสร็จแล้ว, 17
การมีส่วนร่วมของผู้มีส่วนได้ส่วนเสีย, 42	งานเตรียมความพร้อมอย่างชัดเจน, 26
การรักษาค้นแก่, 19	

- งานเพิ่มเติม, 15
- งานเริ่มต้น, 42
- ช่องโหว่ด้านความปลอดภัย, 15
- ฐานความรู้, 6
- ฐานโค้ด, 19
- ต้นทุนสะสม, 19
- ทรัพยากร, 60
- ทรัพยากรเซิร์ฟเวอร์, 18
- ทักษะของทีม, i
- ทีมผู้นำ, 5
- ทีมพัฒนา, 14
- ทีมวิศวกรรม, 7
- ที่ปรึกษา, 6
- ธนาคาร, 8
- นักทดสอบ, 9
- นักพัฒนาซอฟต์แวร์, 15
- นับจำนวนชิ้นส่วน, 16
- นิยามของงานที่เสร็จสมบูรณ์, 17
- นิยามของงานที่เสร็จสมบูรณ์ของรายการในบัญชีงานของผลิตภัณฑ์,
39
- นิยามของงานที่เสร็จสมบูรณ์แบบเฉพาะเจาะจง, 38
- นิยามความพร้อมเฉพาะกิจ, iii, 48
- บริการต้นน้ำ, 9
- บริการที่ต้องพึ่งพา, 7
- บางสิ่งที่หายไป, 1
- บาร์เรลาส, ทอม, vii
- บุคลากรที่ทดแทนกันได้, 18
- ประตูในกระบวนการ, 48
- ผลิตภัณฑ์และวิศวกรรม, 11
- ผู้นำ, 7
- ผู้บริหารฝ่ายวิศวกรรม, 3
- ผู้ปฏิบัติงานรายบุคคล, 4
- ผู้มีส่วนได้ส่วนเสีย, 13
- ผู้เชี่ยวชาญเฉพาะด้าน, 5
- ผู้เชี่ยวชาญในเรื่องอื่นๆ, 8
- พร้อม, 9
- พฤติกรรม: ติดตามสถานะของข้อกำหนดอย่างชัดเจน, 55
- พฤติกรรม: เปิดแผนงานทั้งหมดที่เกี่ยวข้องกับความพร้อมและ
การนำไปปฏิบัติ, 55
- พลิกผันจากชัยชนะเป็นความพ่ายแพ้, 14
- พอร์ทัลการชำระคืนเงินกู้, 2
- พัฒนา DoD ให้สมบูรณ์, 39
- ภาษาธรรมชาติ, 12
- รอบการทำงาน, 17
- ระดับของรายละเอียด, 42
- ระบบ, 7
- ระบบหยุดทำงาน, 15
- ระบบเดิม, 5
- รายการงาน, 18
- รายการงานความพร้อม, 26
- รายการงานค้าง, 19
- รายการใน Product Backlog, 6
- รายการในProduct Backlog, 8
- รายการในบัญชีงานของผลิตภัณฑ์, 11
- รายการในบัญชีงานค้าง, 16
- รายการในบัญชีงานค้าง, 9
- รูปแบบการนำไปใช้, 58
- รูปแบบการออกแบบ, iv
- ล่าช้า, 19
- วงจรข้อมูลย้อนกลับ, 10
- เวอร์ตอร์ฟอลส์, 12
- วัฒนธรรม, 7
- สกริม, 12, 17
- สคริม, 4
- สถาบันเครดิตแบบสหพันธ์, 2
- สปринท์, 8, 11, 14, 17, 18
- สภาพแวดล้อมการพัฒนา, 7
- ส่วนแบ่งการตลาด, 15
- หนี้เชิงเทคนิค, 52
- หลักการเขียนโปรแกรมยุคกลางศตวรรษที่ยี่สิบ, 5
- องค์กรด้านผลิตภัณฑ์, 3
- อโงอ, 18
- เกณฑ์การรับงานของฝ่ายวิศวกรรม, 48
- เกณฑ์การส่งมอบจากฝ่ายผลิตภัณฑ์, 48
- เกณฑ์ความสำเร็จของงาน, 39
- เซ็นทรัลออริกอน, vi
- เทคนิคการพัฒนาซอฟต์แวร์, 5
- เทคนิคการเขียนความต้องการ, 7
- เฟรมเวิร์กอื่นๆ, 58
- เอพีไอ, 17
- เอไอ, 16
- แนวทางปฏิบัติทางวิศวกรรม, 7
- แนวปฏิบัติทางวิศวกรรม, 11
- โครงสร้างของนิยามความพร้อม, 48

โครงสร้างพื้นฐานทางการเงินระดับชาติของสหรัฐอเมริกา, 2

โค้ดที่มีอยู่, 8

โซลูชัน C#/.NET, 3

โซลูชันโค้ดน้อย, 3

โดเมน, 7

โปรดักต์ไอออนเนอร์, 7

โปรดักท์, บทบาทของ, 3

โปรดักท์แมนเนเจอร์, 4

โปรดักท์ไอออนเนอร์, 4