

ตัวอย่างรายการเพื่อใช้ตรวจสอบสำหรับ ScrumMaster

ไมเคิล เจมส์ (mj4scrum@gmail.com)

วันที่ 14 กันยายน 2007

(ปรับปรุงเมื่อ 2 กุมภาพันธ์ 2016)

พงษ์ศักดิ์ ฤทธิ์พิทักษ์พงศ์ (jane@odd-e.com)

วันที่ 21 ธันวาคม 2018

ผู้มีหน้าที่อำนวยความสะดวก เต็มเวลาหรือไม่?

สกรัมมาสเตอร์(ScrumMaster) ที่ดีจะสามารถดูแลทีมได้ครั้งละสองหรือสามทีม และหากคุณพอใจที่จะจำกัดบทบาทของคุณไว้เพียงแค่จัดระเบียบการประชุม บังคับให้ใช้เวลาตามที่กำหนดไว้ และตอบสนองต่อปัญหาต่าง ๆ ทันทีแล้วละก็ คุณสามารถทำบทบาทนี้ได้โดยไม่จำเป็นต้องเป็นงานเต็มเวลา ทีมอาจจะยังคงสามารถทำงานร่วมกันได้ตามความคาดหวังพื้นฐานขององค์กร และอาจจะไม่มีหายนะใด ๆ เกิดขึ้น

แต่ถ้าคุณกำลังจินตนาการถึงทีมที่ทำงานสำเร็จลุล่วงตามเป้าหมายร่วมกันได้อย่างมีความสุขในแบบที่ไม่เคยมีใครคิดฝันมาก่อน มันก็อาจจะถึงเวลาที่องค์กรควรพิจารณาถึงการมีสกรัมมาสเตอร์(ScrumMaster) ทำงานแบบเต็มเวลาได้แล้ว

“สกรัมมาสเตอร์(ScrumMaster) ที่ยอดเยี่ยม” จะดูแลทีมที่ละหนึ่งทีม เท่านั้น

คำแนะนำของพวกเราคือ สกรัมมาสเตอร์หนึ่งคนควรทำงานร่วมกับทีมที่มีสมาชิกประมาณ 7 คน ณ ช่วงก่อร่างสร้างทีม

เอกสารนี้ไม่ใช่เอกสารที่สมบูรณ์แบบสำหรับทุกคนในทุกบริบท โดยเอกสารประกอบด้วยส่วนต่าง ๆ สี่ส่วนโดยเน้นที่เรื่องที่คุณต้องคำนึงถึงสี่ด้านคือโปรดักส์โอเนอร์(Product Owner) ของคุณ ทีมของคุณ ความชำนาญทางวิศวกรรมของทีมของคุณ และหน่วยงานอื่น ๆ รอบ ๆ ทีมของคุณ คุณจะพบว่าเอกสารนี้ไม่ได้รวบรวมหน้าที่ทั้งหมดที่สกรัมมาสเตอร์(ScrumMaster) ควรทำได้ ผมเพียงแค่รวบรวมสิ่งที่ผมมักเห็นสกรัมมาสเตอร์(ScrumMaster) มองข้ามเอาไว้ในเอกสารนี้เท่านั้น

โปรดทำเครื่องหมายแต่ละช่อง ด้วยสัญลักษณ์ √, Δ,? หรือ N/A ตามคำอธิบายในหน้าสุดท้าย

ส่วนที่ 1 - โปรดักส์โอเนเนอร์(Product Owner) ของฉันเป็นอย่างไรบ้าง?

สกรัมมาสเตอร์(ScrumMaster) สามารถช่วยปรับปรุงประสิทธิภาพของโปรดักส์โอเนเนอร์(Product Owner) ได้โดยการช่วยให้เค้าหาวิธีการที่เหมาะสมมาบำรุงรักษาโปรดักส์แบคล็อก(Product Backlog) รวมถึงการวางแผนของการส่งมอบได้ (อย่าลืมว่าโปรดักส์โอเนเนอร์(Product Owner) เป็นผู้รับผิดชอบและมีสิทธิ์ขาดในการเรียงลำดับความสำคัญของงาน)

- โปรดักส์แบคล็อก(Product Backlog) ได้ถูกปรับปรุงการเรียงลำดับความสำคัญตามความเห็นของโปรดักส์โอเนเนอร์(Product Owner) อยู่ตลอดเวลาหรือไม่?
- ความต้องการหรือความปรารถนาของผู้เกี่ยวข้องได้ถูกบันทึกไว้ในโปรดักส์แบคล็อก(Product Backlog) หรือไม่? (อย่าลืมว่าโปรดักส์แบคล็อก(Product Backlog) สามารถปรับเปลี่ยนได้ทุกวินาที)
- โปรดักส์แบคล็อก(Product Backlog) อยู่ในขนาดที่สามารถจัดการได้หรือไม่? จำนวนของชิ้นงานในโปรดักส์แบคล็อก(Product Backlog) อยู่ในจำนวนที่สามารถจัดการได้กล่าวคือ ให้งานที่มีมียาละเอียดชัดเจนอยู่ด้านบน และให้งานที่ยังไม่มีความชัดเจนหรือยังไม่มีรายละเอียดอยู่ด้านล่าง เพื่อป้องกันให้เราไม่ใช้เวลากับการวิเคราะห์รายละเอียดเชิงลึกเร็วเกินไป เพราะอย่างไรก็ตามรายละเอียดของชิ้นงานที่อาจจะมีการเปลี่ยนแปลงแก้ไขได้ตลอดในระหว่างที่เราพัฒนาผลิตภัณฑ์หรือพูดคุยกับลูกค้าและผู้ที่เกี่ยวข้องหรือแม้แต่ผลตอบรับจากการใช้งานจริง
- เป็นไปได้แค่ไหนที่ชิ้นงานแต่ละชิ้น โดยเฉพาะงานชิ้นบน ๆ ของโปรดักส์แบคล็อก(Product Backlog) สามารถจัดให้อยู่ในลักษณะของงานที่เป็นอิสระต่อกัน ต่อรองได้ มีคุณค่าทางธุรกิจ ประเมินความซับซ้อนได้ และทดสอบได้
- คุณเคยให้ความรู้โปรดักส์โอเนเนอร์(Product Owner) เกี่ยวกับ “หน้ทางเทคนิค” และวิธีการหลีกเลี่ยงหรือป้องกันหรือไม่? หนึ่งในวิธีการดังกล่าว เช่น การใช้เทคนิคการทดสอบแบบอัตโนมัติ(Automate Test) หรือการปรับเปลี่ยนรูปแบบของชุดคำสั่ง(Refactor) เพื่อให้ง่ายต่อการบำรุงรักษา โดยกำหนดให้เป็นหนึ่งในนิยามของคำว่าเสร็จ (DOD) ของงานทุกชิ้น
- โปรดักส์แบคล็อก(Product Backlog) เป็นจุดศูนย์กลางของข้อมูลทั้งหมดและเป็นທີ່ประจักษ์ต่อสายตาของทุกคนที่เกี่ยวข้องหรือไม่?
- หากคุณใช้เครื่องมืออัตโนมัติสำหรับการจัดการงานในโปรดักส์แบคล็อก(Product Backlog) ทุกคนสามารถเชื่อมั่นได้อย่างง่ายดายหรือไม่? (เครื่องมืออัตโนมัติมักมาพร้อมกับอันตรายที่ทำให้ข้อมูลของแต่ละชิ้นงานไม่ได้รับการปรับปรุงอยู่เสมอ)
- คุณสามารถช่วยทำให้ข้อมูลในโปรดักส์แบคล็อก(Product Backlog) ถูกเผยแพร่ให้ทุกคนได้โดยง่ายหรือไม่?
- คุณสามารถใช้เครื่องมือต่าง ๆ เพื่อช่วยทุกคนมองเห็นภาพรวมได้อย่างชัดเจนหรือไม่?
- คุณได้ช่วยโปรดักส์โอเนเนอร์(Product Owner) ในการจัดเรียงชิ้นงานให้อยู่ในรอบของการส่งมอบให้ลูกค้า หรือกลุ่มของลำดับความสำคัญที่เหมาะสมหรือไม่?

- ทุกคนรู้หรือไม่ว่าแผนของการส่งมอบ ยังคงสะท้อนกับความเป็นจริง? เช่น มีการปรับปรุงแผนภูมิรีริสเบิร์นดาวน์(Release Burndown Chart) หลังจากงานแต่ละชิ้นได้รับการยอมรับว่า “เสร็จ” ในการประชุมสปริ้นท์รีวิว(Sprint Review) แผนภูมิดังกล่าวจะแสดงทั้งชิ้นงานที่เสร็จจริง ๆ และชิ้นงานที่เพิ่มเข้ามาใหม่ซึ่งนั่นหมายถึงการอนุญาตมีการปรับแผนของการส่งมอบไปจากเดิม
- โพรดักส์โอเนอร์(Product Owner) ได้ปรับปรุงแผนของการส่งมอบหลังจากจบการประชุมสปริ้นท์รีวิว(Sprint Review) หรือไม่? โพรดักส์โอเนอร์(Product Owner) ที่ดีจะปรับปรุงแผนของการส่งมอบทุกรอบการทำงาน และนั่นอาจจะหมายถึงอาจจะมียางบางชิ้นต้องเลื่อนออกไปหากค้นพบว่ามียางชิ้นที่สำคัญกว่าต้องทำก่อน

ส่วนที่ 2 - ทีมของคุณเป็นอย่างไรบ้าง?

ในขณะที่คุณได้รับการสนับสนุนให้เป็นผู้นำด้วยวิธีการทำให้คุณเป็นตัวอย่างโดยรวมมือกับสมาชิกในทีมฝึกฝนผ่านการทำงานของพวกเขา จึงมีความเสี่ยงที่คุณจะจมลงไปในงานและรายละเอียดทางเทคนิคได้ง่าย สิ่งเหล่านี้คือสิ่งที่ต้องพิจารณาและเป็นความรับผิดชอบหลักของคุณต่อทีม

- ทีมของคุณอยู่ในสถานะที่เรียกว่า “ไหลลื่น”(Flow) หรือไม่? โดยทั่วไปจะหมายถึงทีมมีคุณลักษณะดังต่อไปนี้
 - เข้าใจเป้าหมายชัดเจน (ความคาดหวังที่จะบรรลุเป้าหมาย สอดคล้องกับทักษะและความสามารถ)
 - มีสมาธิจดจ่ออยู่กับงานที่ทำ
 - สมาชิกทุกคนมั่นใจและ รู้ตัวดี ว่ากำลังทำอะไรอยู่
 - ได้ฟีดแบ็ค(Feedback) จากกิจกรรมที่ทำอย่างชัดเจน และทันที (ทั้งในส่วนของ ความสำเร็จและความผิดพลาด ทำให้สามารถปรับปรุงพฤติกรรมได้ทันทีตามความจำเป็น)
 - มีความสมดุลระหว่างระดับความสามารถ และความท้าทาย (งานที่ทำไม่ยากหรือว่าง่ายเกินไป)
 - มีความรู้สึกมั่นใจ ว่าสามารถควบคุมสถานะการหรือกิจกรรมที่ทำได้
 - งานที่ทำ ทำให้เขารู้สึกมีคุณค่าไม่รู้สึกเสียแรงเปล่า
- สมาชิกในทีมมีเรื่องที่มีปัญหากันเข้าใจผิดกันหรือโอกาสในการปรับปรุงที่ทีม ไม่ได้พูดคุยกันเพราะความเกรงใจหรืออึดอัดหรือไม่?
- คุณมีความหลากหลายของวิธีการหรือสถานที่ในการทำสปริ้นท์เรโทรสเปคทีฟ(Sprint Retrospective) หรือไม่?
- ทีมจะทำงานอย่างมุ่งมั่นเพื่อให้ทำตามเป้าหมายของรอบการทำงานนั้น ๆ หรือไม่ คุณอาจจะลองหากิจกรรมเพื่อตรวจสอบ กลางรอบการทำงานเพื่อตรวจสอบว่าทีมยังคงการทำงานเพื่อให้บรรลุเป้าหมายตามรายละเอียดของแต่ละชิ้นงานที่รับเข้ามาในรอบการทำงานนี้หรือไม่?
- สปริ้นท์แบ็คล็อก(Sprint Backlog) สะท้อนงานจริงที่ทีมกำลังทำอยู่หรือไม่? ะวังงานที่ซ่อนอยู่ (ทีมทำอยู่แต่ไม่อยู่บนสปริ้นท์แบ็คล็อก(Sprint Backlog) และงานที่ต้องใช้เวลามากกว่าหนึ่งวัน รวมถึงงานที่ไม่เกี่ยวข้องกับข้อตกลงของงานในรอบการทำงานนั้น ๆ

- สมาชิกในทีมแต่ละคนชื่นชอบพอซึ่งกันและกัน ออกไปสนุกด้วยกัน ฉลองให้กับความสำเร็จของกันและกันหรือไม่?
- สมาชิกในทีมมีความรับผิดชอบต่อกันและกันสูง และท้าทายกันและกันเพื่อเติบโตอยู่เสมอหรือไม่?
- ทีมงานมีจำนวนสมาชิกระหว่าง 3-9 คนโดยผสมผสานทักษะต่าง ๆ ที่จำเป็นต้องใช้ในการพัฒนาผลิตภัณฑ์หรือไม่?
- สปรินท์แบ็กล็อก(Sprint Backlog) ได้รับการปรับปรุงอยู่ตลอดหรือไม่?
- สิ่งของหรืออุปกรณ์ต่าง ๆ ที่ทีมต้องใช้งานเพื่อให้ทีมสามารถบริหารจัดการตัวเองได้ ถูกจัดไว้ให้เห็นได้อย่างชัดเจน และทีมสามารถใช้งานได้อย่างสะดวกหรือไม่?
- สปรินท์แบ็กล็อก(Sprint Backlog) ถือเป็นทรัพย์สินของทีม ทั้งนี้เพื่อช่วยให้ทีมใช้ในการบริหารจัดการตัวเอง คุณได้ทำให้มั่นใจว่าสปรินท์แบ็กล็อก(Sprint Backlog) ของทีมไม่ถูกคนที่ไม่ใช่ทีม เช่น โปรดักต์โอเนอร์(Product Owner) ผู้จัดการ หรือคนที่ไม่ได้อยู่ในทีม นำไปใช้เป็นข้อมูลในการตัดสินใจให้คุณหรือโทษใด ๆ ทั้งสิ้น หรือไม่?
- สมาชิกในทีมอาสาที่จะหยิบชิ้นงานไปทำเองหรือไม่?
- มีความต้องการที่จะลดหรือกำจัดหนี้ทางเทคนิค และได้ถูกทำเป็นข้อตกลงในคำจำกัดความของคำว่า เสร็จ(Definition of Done) อย่างชัดเจน เพื่อค่อย ๆ ทำให้ซอร์ซโค้ด(Source Code) เป็นของที่นำทำงานขึ้นเรื่อย ๆ อยู่เสมอหรือไม่?
- สมาชิกในทีมไม่ได้ให้ความสำคัญกับชื่อตำแหน่ง และร่วมด้วยช่วยกันทำงานทุกอย่างเป็นทีมเพื่อให้งานสำเร็จ (การทดสอบ, จัดทำเอกสาร, ...) หรือไม่?

ส่วนที่ 3 -- ความชำนาญทางวิศวกรรมของทีมของฉันทันเป็นอย่างไรบ้าง?

- ในระบบที่ใช้ในการพัฒนาของคุณมีปุ่มที่ชื่อว่า “Push to Test” และอนุญาตให้สมาชิกทุกคน (ไม่ว่าจะในทีมเดียวกัน หรือต่างทีม) สามารถกดเพื่อตรวจสอบว่าสิ่งที่แก้ไขทำให้ฟังก์ชันงานเดิมที่เคยทำงานได้บกพร่องหรือไม่ ปกติแล้วสามารถใช้ xUnit Framework ต่าง ๆ เช่น JUnit, NUnit เป็นต้น
- คุณให้สมดุลระหว่างการทดสอบการทดสอบระบบอัตโนมัติ(End-To-End Test) และการทดสอบหน่วยย่อยอัตโนมัติ(Unit Test) อย่างเหมาะสมหรือไม่?
- ทีมใช้ภาษาในการเขียนการทดสอบระบบและ การทดสอบหน่วยย่อย(Unit Test) เป็นภาษาเดียวกันกับระบบที่พวกเขา กำลังพัฒนาหรือไม่? เราจะไม่สามารถพัฒนาการทำงานร่วมกันได้หากใช้ภาษาพิเศษหรือระบบในการทดสอบที่มีแค่ส่วนหนึ่งของทีมเท่านั้นที่รู้วิธีการดูแลรักษา
- ทีมทราบถึงความคล้ายคลึงระหว่าง การทดสอบระบบ(System test) และการทดสอบหน่วยย่อย(Unit test) หรือไม่?

- การควรวมต่อเนื่อง(Continuous Integration) มีการส่งเสียงหรือแจ้งเตือนอัตโนมัติเมื่อมีสมาชิกทำให้ระบบที่เคยทำงานได้บกพร่องหรือไม่? และพยายามลดเวลาของการได้รับฟีดแบ็ค(Feedback) นี้ลงได้เรื่อยๆ ตั้งแต่ระดับชั่วโมง จนถึงไม่กี่นาที
- ผลการทดสอบทั้งหมดได้ถูกรวบรวมเข้ากับ ระบบการควรวมต่อเนื่อง(Continuous Integration) หรือไม่?
- ทีมของคุณพบความสนุกของการออกแบบอย่างต่อเนื่องและ การปรับปรุงอย่างสม่ำเสมอ(Refactor) ในเวลาที่เหมาะสมหรือไม่? การออกแบบและการปรับปรุงอย่างต่อเนื่องถือเป็นทางเลือกแทนการทำการออกแบบเพื่อเพื่อความสมบูรณ์แบบ(Up Front Design) การปรับปรุงอย่างสม่ำเสมอ(Refactoring) มีข้อกำหนดที่เคร่งครัด หมายถึง การทำการเปลี่ยนแปลงโครงสร้างภายในโดยไม่เปลี่ยนแปลงพฤติกรรมภายนอก โดยทั่วไปการปรับปรุงสม่ำเสมอ(Refactor) ควรเกิดขึ้นอย่างน้อย 7-9 ครั้งต่อชั่วโมง หรือเมื่อไหร่ก็ตามที่เกิดความซ้ำซ้อน ตรรกที่ซับซ้อน (เห็นได้ชัดจาก Method ที่ยาวๆ) การตั้งชื่อที่ไม่สื่อความหมาย หรือการที่ Object ผูกติดกันแน่น เป็นต้น อย่างไรก็ตามการปรับปรุงสม่ำเสมอ(Refactor) จะเกิดขึ้นได้เมื่อมีการทดสอบอัตโนมัติที่ครอบคลุม การละเลยการปรับปรุงสม่ำเสมอ ทำให้ระบบยากต่อการเปลี่ยนแปลงในอนาคตโดยเฉพาะการทำ นักพัฒนาเก่ง ๆ ที่ยินดีมาทำงานต่อบนระบบที่แย่
- คำจำกัดความของคำว่าเสร็จ(Definition of Done) สำหรับแต่ละชิ้นงานได้รวมถึงความครอบคลุมของการทำการทดสอบอัตโนมัติ และการปรับปรุงสม่ำเสมอ(Refactor) แล้วหรือไม่? คุณสามารถเพิ่มความเป็นไปได้ด้วยการเรียนรู้เทคนิค การขับเคลื่อนการพัฒนาด้วยการทดสอบ(Test Driven Development)
- ทีมใช้เวลาส่วนใหญ่ในการใช้เทคนิคการทำงานเป็นคู่หู(Pair Programming) หรือไม่? การทำงานเป็นคู่หู(Pair Programming) สามารถเพิ่มความสามารถของ การบำรุงรักษาระบบ รวมถึงการลดจำนวนข้อผิดพลาด ได้อย่างมาก มันอาจจะดูเชื่อได้ยากมาก โดยเฉพาะเมื่อเราวัดผลลัพธ์ จากจำนวนบรรทัดของซอร์สโค้ด(Source Code) ที่เขียนได้ มากกว่าการวัดที่ ฟังก์ชันงานที่สามารถส่งมอบได้จริง คุณสามารถใช้วิธีการทำให้ดูเป็นตัวอย่ง ได้โดยการเริ่มเป็นคู่หูของสมาชิกในทีมและทำงานด้วยกันสักวันหนึ่ง หรือบางคนอาจชอบกับการทำงานเป็นคู่หู(Pair Programming) ตั้งแต่ต้นอยู่แล้ว

ส่วนที่ 4 -- องค์กรของคุณเป็นอย่างไรบ้าง

- มีการประสานงานระหว่างทีมอย่างเหมาะสมหรือไม่? “Scrum of Scrums” อาจจะไม่ใช่วิธีที่ดีที่สุด ยังคงมีวิธีอื่นที่น่าจะส่งเสริมให้เกิดการประสานงานกัน
- ทีมแต่ละทีมสามารถสร้างส่งมอบชิ้นงาน(Feature) ที่สามารถทำงานได้อย่างอิสระต่อกันหรือไม่? ถึงแม้ว่าการแก้ไขต้องข้ามเส้นแบ่งของสถาปัตยกรรมของระบบก็ตาม
- สกอร์มาสเตอร์(ScrumMaster) ได้มีการนัดพบกันเองในกลุ่มสกอร์มาสเตอร์(ScrumMaster) เพื่อช่วยกันแก้ไขอุปสรรคขององค์กรหรือไม่?

- อุปสรรคต่าง ๆ ขององค์กรได้ถูกทำให้ผู้มีอำนาจแก้ไขเปลี่ยนแปลงรับทราบในทุกเวลาที่เป็นไปได้ หรือไม่? อุปสรรคดังกล่าวอาจจะส่งผลให้ต้นทุนสูงขึ้น ส่งมอบของสู่ตลาดได้ช้าลง คุณภาพลดลง หรือทำให้ลูกค้าเสียโอกาส อย่างไรก็ตาม สิ่งที่ได้เรียนรู้จากความผิดพลาดของ เคน ชเวบเบอร์(Ken Schwaber) คือ “สกรัมมาสเตอร์(ScrumMaster) ที่ตายแล้วคือสกรัมมาสเตอร์(ScrumMaster) ที่ไม่มีประโยชน์”
- คุณได้ทำการช่วยเหลือและทำความเข้าใจถึง นโยบายของฝ่ายทรัพยากรบุคคลเพื่อลดความเข้าใจผิด เรื่องของความก้าวหน้าในสายอาชีพ เช่น การเป็นนักพัฒนา และนักทดสอบระบบ ได้รับคำตอบแทนที่ ต่างกันมากหรือ เขียนโปรแกรมให้เก่งเพื่อให้เป็นผู้จัดการที่ไม่ต้องเขียนโปรแกรมอีก
- องค์กรของคุณได้รับการยอมรับจากสื่อต่าง ๆ ว่าเป็นที่ที่นำทำงานมากที่สุดหรือ เป็นผู้นำในธุรกิจหรือไม่?
- คุณได้พยายามสร้างองค์กรแห่งการเรียนรู้หรือไม่?

บทสรุป

ถ้าคุณทำเครื่องหมายถูกในหัวข้อส่วนใหญ่ที่กล่าวมาและยังพอมีเวลาเหลือ ผมยินดีรับฟังความเห็นของคุณ

มันไม่มีสูตรสำเร็จที่สามารถทำให้มนุษย์ฉลาดขึ้นในทุก ๆ บริษัท เอกสารฉบับนี้ได้แสดงถึงรายการที่อาจจะช่วยหรือไม่ช่วยก็ได้ในบริษัทที่คุณกำลังเจอ

เมื่อคุณตระหนักได้ว่าคุณจะสร้างความแตกต่างให้เกิดขึ้นได้อย่างไร หรือคุณอาจจะพบว่าคุณกลัวที่จะทำมัน นี่ถือเป็นสัญญาณว่าคุณมาถูกทางแล้วของการเป็นสกรัมมาสเตอร์(ScrumMaster) ที่ยอดเยี่ยม

แบบฟอร์มอุปสรรคขององค์กร

ปัญหาที่พบ :

ต้นเหตุของปัญหา (ใช้เทคนิคการถามทำไม ห้าครั้ง) :

ผลกระทบต่อธุรกิจ :

ผลกระทบต่อความรู้สึก :

ความเข้าใจ หรือ รายการที่จะดำเนินการ :

แบบฟอร์มอุปสรรคขององค์กร

ปัญหาที่พบ :

ต้นเหตุของปัญหา (ใช้เทคนิคการถามทำไม ห้าครั้ง) :

ผลกระทบต่อธุรกิจ :

ผลกระทบต่อความรู้สึก :

ความเข้าใจ หรือ รายการที่จะดำเนินการ :

คำแนะนำ

หากคุณได้รับเอกสารฉบับนี้เพื่อเป็นแบบสอบถามในการอบรมและสถานการณ์ที่คุณกำลังเผชิญอยู่คือการพยายามทำบางอย่างที่ดูคล้าย Scrum ขอให้คุณนึกถึงภาพที่คุณเคยเห็นมา และทำเครื่องหมายในข้อต่าง ๆ ดังนี้

✓ สำหรับ “ทำได้ดี”

Δ สำหรับ “สามารถทำให้ดีกว่านี้ได้ และคุณรู้ว่าจะเริ่มทำอะไร”

? สำหรับ “สามารถทำให้ดีกว่านี้ได้ แต่คุณไม่รู้ว่าจะทำอย่างไร”

N/A สำหรับ “ไม่เหมาะสม หรือ อาจจะไม่มีความหมาย”

หรือ สถานการณ์ที่คุณเผชิญอยู่คือการพยายามใช้ Scrum กรุณานึกถึงภาพที่คุณเคยเห็นมา และทำเครื่องหมายในข้อต่าง ๆ ดังนี้

✓ สำหรับ “ทำได้ดี” หรือ “น่าจะทำได้ดีได้ไม่ยาก”

Δ สำหรับ “เป็นเรื่องที่ท้าทาย และคุณรู้ว่าจะเริ่มทำอะไร”

? สำหรับ “เป็นเรื่องที่ท้าทาย แต่คุณไม่รู้ว่าจะทำอย่างไร”

N/A สำหรับ “ไม่เหมาะสม หรือ อาจจะไม่มีความหมาย”

เมื่อคุณทำเครื่องหมายครบทุกข้อแล้ว ให้ลองนึกถึงอุปสรรคที่องค์กรของคุณกำลังเจอ 2-6 ข้อโดยใช้แบบฟอร์มอุปสรรคขององค์กรตามที่แนบมา ไม่ว่าจะพบบ้างหรือไม่พบบ้างจากการทำรายการนี้หรือไม่ก็ตาม จากนั้นเลือกอุปสรรคที่คุณรู้สึกมีความหวังอย่างน้อย 1% ที่คุณสามารถที่จะเปลี่ยนแปลงมันได้ แล้วทำมัน