

THE SCRUM PRIMER

ภาคภาษาไทย

Scrum Primer Translation Group
@Agile66

version 0.8

Table of Content

Traditional Software Development.....	4
Agile Development and Scrum	5
Scrum Summary	6
Scrum Roles.....	7
Starting Scrum.....	10
Sprint Planning	12
Daily Scrum.....	15
Updating Sprint Backlog & Sprint Burndown Chart	16
Product Backlog Refinement.....	17
Ending the Sprint	18
Sprint Review	18
Sprint Retrospective.....	19
Updating Release Backlog & Burndown Chart	20
Starting The Next Sprint.....	21
Release Sprint.....	21
Release Planning & Initial Product Backlog Refinement	22
Application or Product Focus	23
Common Challenges	24
Appendix: Terminology	26

Table of Figures

รูปภาพที่ 1: Scrum	7
รูปภาพที่ 2: Product Backlog	10
รูปภาพที่ 3: Estimating Available Hours	13
รูปภาพที่ 4: Sprint Backlog	14
รูปภาพที่ 5: Visual Management – Sprint Backlog tasks on the wall	14
รูปภาพที่ 6: Daily Updates of Work Remaining on the Sprint Backlog	16
รูปภาพที่ 7: Sprint Burndown Chart	17
รูปภาพที่ 8: Release Backlog (a subset of the Product Backlog)	20
รูปภาพที่ 9: Release Burndown Chart	21

Traditional Software Development

ตามขอบของการสรรค์สร้าง Software ไม่ว่าจะเป็นบริษัทเล็กหรือใหญ่ก็ตามจะเป็นไปตามลำดับกระบวนการที่ตายตัวที่เราคุ้นเคยกันดีเรียกว่า **Waterfall** ซึ่งมีได้หลากหลายรูปแบบ แล้วแต่การประยุกต์เช่น **V-Model** เป็นต้น โดยทั่วไปแล้วลำดับกระบวนการแบบนี้มักจะเริ่มต้นที่ขั้นตอนการวางแผนแบบละเอียดิบ ซึ่งตัวผลิตภัณฑ์ที่ได้จะผ่านกระบวนการคิดวิเคราะห์อย่างเอาใจใส่ ตามต่อด้วยการออกแบบ และการทำเอกสารอย่างละเอียดถี่ถ้วน หลังจากนั้นงานที่เกิดอย่างพอดิบพอดีและสอดคล้องกับตัวแบบที่วางเอาไว้ก็就会被เรียงเรียงผ่านเครื่องมืออย่าง **Gantt charts** และ โปรแกรมประยุกต์อย่าง **Microsoft Project** ที่มักจะร่วมกันประเมินระยะเวลาของการพัฒนาด้วยการประมาณงานในแต่ละขั้นตอนย่อย และนำมารวมเข้าด้วยกัน และเมื่อแผนงานทั้งหมดนี้ถูกพิจารณาและเห็นชอบจากลูกค้า ทีมจึงเริ่มต้นทำงานต่อไป เมื่อลูกค้าทีมเสร็จสิ้นงานในส่วนของตนเอง งานก็就会被ส่งต่อไปให้กับคนอื่นที่อยู่ในสายการผลิต และเมื่องานทั้งหมดเสร็จสมบูรณ์ ผลิตภัณฑ์ก็就会被ส่งไปสู่กระบวนการทดสอบ (บางบริษัทเรียกว่า **Quality Assurance**) และจะถูกส่งมอบให้กับลูกค้าเมื่อกระบวนการทดสอบสิ้นสุดลง ตลอดกระบวนการนี้จะถูกควบคุมอย่างเคร่งครัดเพื่อให้แน่ใจว่าสิ่งที่ผลิตได้นั้นเป็นไปตามสิ่งที่ออกแบบไว้

กระบวนการนี้มีทั้งข้อดีและข้อด้อย ข้อดีที่เยี่ยมยอดของ **Waterfall** ก็คือ ทุกอย่างที่ทำนั้นสมเหตุสมผลและเป็นไปตามตรรกะของการทำงานที่เราจะต้องเขียนระบุรายละเอียดทุกอย่าง และทำตามแผนงานที่วางไว้ และพยายามจัดสรรงานทุกอย่างให้มากที่สุดเท่าที่เป็นไปได้ ส่วนข้อด้อยที่ร้ายแรงที่สุดเพียงข้อเดียวของ **Waterfall** นั่นก็คือมนุษย์ ถูกรวมอยู่ในแผนการทั้งหมดนั้นด้วย

ตัวอย่างเช่น กระบวนการนี้จำเป็นต้องรวบรวมความคิดสร้างสรรค์ทั้งหมดตั้งแต่ตอนเริ่มต้นกระบวนการ แต่อย่างที่เราู้กันดีว่าความคิดที่นั้นเกิดขึ้นได้ตลอดการทำงาน อาจเป็นตอนเริ่มต้น ระหว่างการทำงาน หรือแม้กระทั่งก่อนวันเปิดตัวผลิตภัณฑ์ด้วยซ้ำ และกระบวนการที่ไม่รองรับการเปลี่ยนแปลงก็จะทำให้งานนวัตกรรมเหล่านั้นดับสูญไป ดังนั้นความคิดสร้างสรรค์ที่เยี่ยมยอดที่เกิดขึ้นระหว่างกระบวนการผลิตแบบ **Waterfall** กลับไม่ใช่ผลดีแต่เป็นผลเสีย

แนวคิดของ **Waterfall** นั้น ยังให้ความสำคัญกับการเขียนทุกอย่างลงบนกระดาษเพื่อใช้เป็นสื่อกลางในการสื่อสาร และสมมติฐานที่ค่อนข้างสมเหตุสมผลก็คือ ยิ่งเราสามารถเขียนทุกอย่างที่อยู่ในหัวลงในกระดาษได้มากเท่าไร เราจะยิ่งมั่นใจได้ว่าการถ่ายทอดข้อมูลไปสู่ผู้อื่นในทีมนั้นมีความถูกต้องมากขึ้นตามไปด้วย และมันยังเป็นหลักฐานได้อีกด้วยว่าเราได้เสร็จสิ้นงานของเราแล้ว แต่ในความเป็นจริงนั้นกลับไม่มีใครอ่านเอกสารที่จบความหนา 50 หน้า และถึงแม้มีคนอ่าน ความเข้าใจผิดก็มักจะเกิดตามมาเสมอๆ เอกสารที่ถูกเขียนขึ้นเป็นเหมือนภาพที่ไม่สมบูรณ์แทนความคิดของเรา และเมื่อผู้อื่นอ่านมัน เขาก็ได้สร้างภาพความเข้าใจของตัวเองขึ้นมาอีกชั้นหนึ่ง ซึ่งอาจจะแตกต่างไปจากสิ่งที่เราต้องการถ่ายทอดในตอนแรก ดังนั้นไม่ใช่เรื่องน่าแปลกใจเลยที่ความเข้าใจผิดอย่างร้ายแรงจะเกิดขึ้นได้

อีกอย่างหนึ่งที่เกิดขึ้นกับงานที่มีมนุษย์ร่วมอยู่คือ เมื่อเราเริ่มทดลองใช้งานผลิตภัณฑ์จริงๆ จะเกิดช่วงเวลา "อ่า-ฮ้า!" เพราะตั้งแต่วันที่แรกที่เรารเริ่มใช้งานผลิตภัณฑ์ เราก็มักจะคิดอีก 20 นทางที่สามารถทำให้มันดีกว่านี้ได้ แต่โชคร้ายที่ข้อมูลอันมี

ค่าเหล่านี้มักจะมาตอนใกล้ๆเปิดตัวผลิตภัณฑ์ ซึ่งจะนับเป็นช่วงเวลาแห่งความยากลำบากที่สุด ถ้าหากต้องการจะเปลี่ยนแปลงอะไรขึ้นมา หรือพูดอีกอย่างก็คือถ้าเราใช้ชนแบบเดิมๆ นี้ในการสร้างผลิตภัณฑ์ เราจะต้องจ่ายราคาแพงเพื่อทำสิ่งที่ถูกต้องเสมอ

อย่างที่เราเห็นว่ามนุษย์ไม่สามารถหยั่งรู้อนาคต เป็นต้นว่า คู่แข่งของเราประกาศข่าวที่เราไม่คาดคิด หรือเหตุขัดข้องทางเทคนิคที่เราไม่คาดคิดดันบังคับเราให้เปลี่ยนทิศทางการพัฒนา และมากกว่านั้นอีกคือมนุษย์เรานั้นช่วยแตกออกอย่างมากในการวางแผนอนาคตที่คาดเดาไม่ได้ง่ายๆ ตัวอย่างเช่น ให้เราลองจินตนาการถึงการใช้ชีวิตในอีกแปดเดือนข้างหน้า และนี่คือข้อผิดพลาดที่เกิดขึ้นบ่อยๆ แม้กระทั่งกับ Gantt chart ที่วางแผนมาอย่างดิบดี

อีกประการหนึ่งคือวัฏจักรแบบเป็นลำดับขั้นตอนมักส่งผลให้เกิดความขัดแย้งในการส่งต่องานระหว่างคนในทีม และประโยคตลกๆที่เรามักจะได้ยินก็คือ "เขาบอกให้ผมทำ สิ่งที่ไม่ได้ระบุไว้ในเอกสาร" "เออเปลี่ยนใจอีกแล้ว" "ผมไม่รับผิดชอบต่องานที่ผมไม่ได้ทำนะ" และเท่าที่มีการสำรวจมาพบว่าการพัฒนาแบบวัฏจักรที่เป็นลำดับขั้นตอนมักไม่สนุกเอาเสียเลย Waterfall กลายเป็นสาเหตุแห่งความทุกข์ยากของเหล่านักพัฒนา และทำให้งานที่ได้ ขาดซึ่งความคิดสร้างสรรค์ ขาดแคลนทักษะ และความมุ่งมั่นของทีมงาน มนุษย์ไม่ใช่หุ่นยนต์ และกระบวนการอะไรก็ตามที่ต้องการให้คนทำตัวเหมือนหุ่นยนต์ก็มักลงเอยด้วยความโศกสลด

กระบวนการที่เข้มงวดต่อการเปลี่ยนแปลงมักทำได้แค่ผลิตภัณฑ์ระดับปานกลาง จริงอยู่ที่ลูกค้าอาจจะได้ในสิ่งที่เค้าต้องการ (ถ้าไม่ต้องแปลเอกสารซ้ำซากอะนะ) แต่นั่นคือสิ่งที่เค้าอยากได้จริงๆหลังจากที่เห็นผลิตภัณฑ์รีเปล่า ผลจากการที่เราเก็บความต้องการของลูกค้าในตอนแรกแล้วก็แช่แข็งมันไว้ เราก็จะได้แค่ผลิตภัณฑ์ตามที่เราระบุไว้ในเอกสาร แทนที่เป็นผลิตภัณฑ์สุดเจ๋งที่เกิดจากความคิดสร้างสรรค์ของผู้คนและการค้นพบสิ่งใหม่

คนที่เคยมีประสบการณ์ในวัฏจักรแบบเป็นลำดับขั้นตอน มักจะเจอภาพเหล่านี้ซ้ำแล้วซ้ำเล่า แต่สามัญสำนึกของเค้ากลับกลายเป็นว่า "ถ้าเราทำให้มันดีขึ้นกว่านี้มันต้องไปได้แน่" ถ้าเพียงแต่เราวางแผนให้มากขึ้น ทำเอกสารให้มากขึ้น ต่อด้วยการเปลี่ยนแปลงให้มากขึ้น ทุกอย่างควรจะเป็นไปด้วยดี แต่โชคร้ายที่หลายทีมมักได้ผลที่ตรงกันข้าม ยิ่งเค้าพยายามมากขึ้นเท่าไร ทุกอย่างก็ยิ่งแย่ลง ในที่สุดทีมผู้จัดการก็มีชื่อเสียงที่เคยลงทุนลงแรงรวมถึงทรัพยากรอีกมากมายใน Waterfall Model ต่างก็กำลังเปลี่ยนไปสู่รากฐานที่แตกต่างที่สามารถแก้ไขข้อผิดพลาดที่ล้าๆมาได้ และ Scrum ก็คือ รากฐานที่แตกต่างนั้นนั่นเอง

Agile Development and Scrum

กระบวนการพัฒนาแบบ Agile มีที่มาจากความเชื่อแนวทางการทำงานที่ใกล้เคียงธรรมชาติของมนุษย์ และการพัฒนาผลิตภัณฑ์ที่เกิดจากการเรียนรู้วัฒนธรรมและความเปลี่ยนแปลงจะให้ผลที่ดีกว่า หลักการของ Agile จะเน้นในเรื่องของการสร้าง software ที่ผู้ใช้สามารถใช้งานได้จริงมากกว่าการเสียเวลาเขียน specification ให้เสร็จเรียบร้อยก่อนทั้งหมด การ

พัฒนาแบบ Agile จะเน้นให้ทุกคนในทีมที่ทำงานแบบ cross-functional มีอำนาจในการตัดสินใจด้วยตัวเอง ซึ่งต่างกับโครงสร้างการทำงานแบบแยกตามหน้าที่ นอกจากนี้ Agile ยังเน้นรอบการทำงานที่สั้นและให้ลูกค้าเข้ามามีส่วนร่วมในระหว่างขั้นตอนการพัฒนา คนทั่วไปมักจะรู้จัก Agile หรือ Scrum แบบคุ้นๆว่า มันเหมือนย้อนกลับไปในวันที่เพิ่งตั้งบริษัทตอนที่เรา “แค่ทำมัน” (just did it)

จนถึงบัดนี้รูปแบบการพัฒนาแบบ Agile ที่เป็นที่นิยมที่สุดคือ Scrum ซึ่งได้รับอิทธิพลมาจากบทความ Harvard Business Review ในปี 1986 ที่เกี่ยวกับแนวทางการพัฒนาผลิตภัณฑ์ที่ประสบความสำเร็จ ในบทความนี้พูดถึงคำว่า “รักบี้” ซึ่งต่อมากลายเป็นคำว่า “Scrum” ใน Wicked Problems, Righteous Solutions (1991, DeGrace and Stahl) โดยเปรียบเทียบแนวทางการพัฒนาทีมที่ประสบความสำเร็จในเกมรักบี้ ในเชิงของการจัดการทีมให้เคลื่อนที่ไปด้วยกันในสนามแข่งขัน และหลังจากนั้นก็ได้รับการปรับปรุงในปี 1993 โดย Ken Schwaber และ Dr. Jeff Sutherland ปัจจุบัน Scrum นิยมใช้กันอย่างแพร่หลายทั้งใน องค์กรขนาดใหญ่และขนาดเล็กเช่น Yahoo!, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne และ US Federal Reserve หลายๆทีมที่ได้นำเอา Scrum มาใช้ต่างบอกในทำนองเดียวกันว่านอกจากจะเพิ่มประสิทธิภาพการทำงานแล้ว มันยังเพิ่มความมุ่งมั่นในการทำงานอีกด้วย สำหรับ product developers หลายๆ คนที่มักจะโดนสั่งงานแบบตามใจฉัน จะมีประโยชน์มาก เพราะ Scrum นั้น เรียบง่ายและทรงประสิทธิภาพ

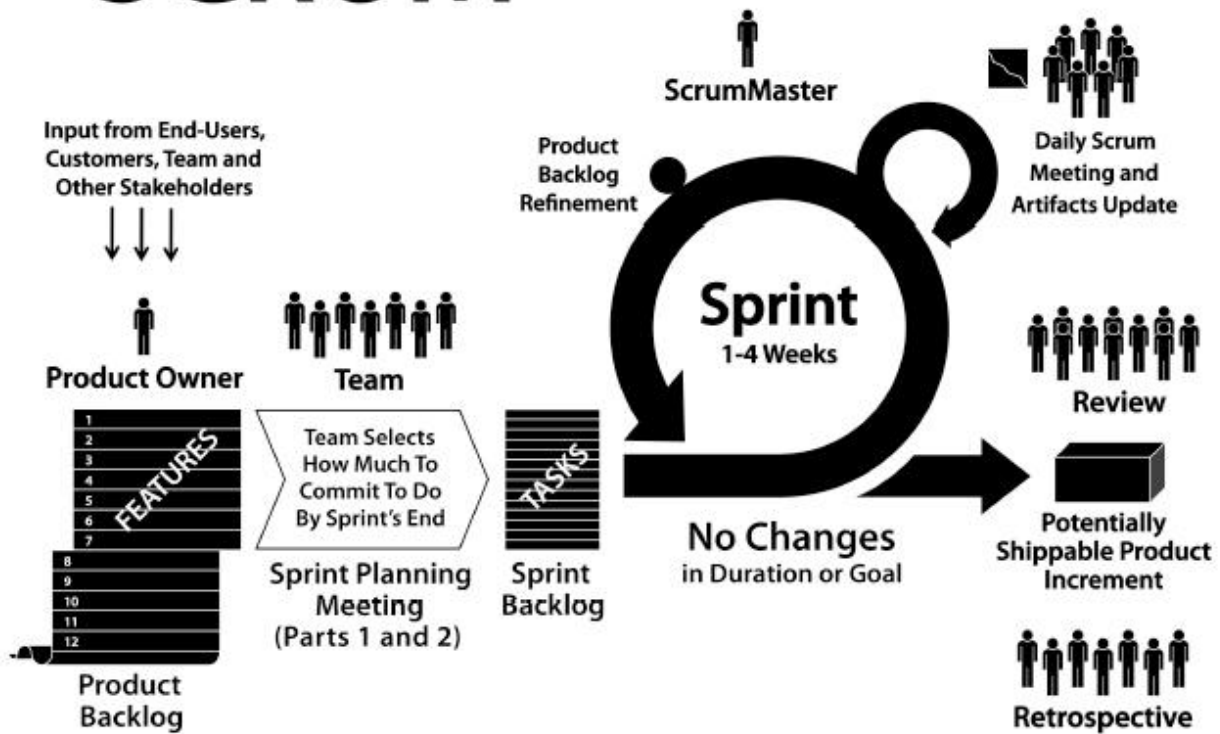
Scrum Summary

Scrum เป็น framework สำหรับการพัฒนาโครงการ software หรือ application ที่มีรูปแบบการพัฒนามีกรอบๆ และมีความสมบูรณ์มากขึ้นเรื่อยๆ ซึ่งรอบการพัฒนาที่เรียกว่า Sprint โดยมีระยะเวลาในแต่ละรอบไม่เกิน 1 เดือน และไม่มีวันหยุดพักระหว่างรอบ แต่ละรอบของ Sprint มีระยะเวลาที่แน่นอนไม่ว่าการทำงานจะเสร็จหรือไม่โดยไม่มีการต่อเวลา (Timeboxed) ในตอนเริ่มต้นแต่ละ Sprint cross-functional ทีมจะเลือกงาน (ความต้องการของลูกค้า) มาทำตามลำดับความสำคัญ และสัญญาว่าจะทำงานที่เลือกให้เสร็จก่อนจบ Sprint โดยไม่มีการเปลี่ยนงานที่เลือกมาทำ ในแต่ละวันทีมจะรายงานสรุปความคืบหน้าการทำงานและปรับเปลี่ยนการทำงานหากจำเป็นเพื่อให้งานส่วนที่เหลือเสร็จ หลังจากจบ Sprint ทีมจะทำการทบทวนและแสดงผลงานให้ผู้มีส่วนเกี่ยวข้องกับโครงการทราบเพื่อรวบรวมคำแนะนำและคำวิจารณ์ไปดำเนินการใน Sprint ถัดไป การทำงานแบบ Scrum จะเน้นที่ “ผลสำเร็จ” ของงานเมื่อสิ้นสุด Sprint เช่นในกรณีของการพัฒนา software ผลสำเร็จของงานหมายถึงชุดคำสั่งที่สมบูรณ์ ได้รับการทดสอบอย่างดี และสามารถส่งมอบงานได้ หน้าที่รับผิดชอบ ผลงาน และกระบวนการทำงานต่างๆ ได้ถูกรวบรวมไว้ในภาพที่ 1

หัวใจสำคัญของ Scrum คือ “การตรวจสอบและปรับปรุง” (inspect and adapt) เนื่องจากการพัฒนาจะต้องเกี่ยวข้องกับการเรียนรู้ นวัตกรรม และสิ่งแปลกใหม่อย่างหลีกเลี่ยงไม่ได้ Scrum จึงมุ่งเน้นที่การพัฒนาเป็นรอบสั้นๆ พร้อมทั้ง

ตรวจสอบผลงานและประสิทธิภาพการทำงานในแต่ละขั้นตอน เพื่อนำมาปรับปรุงเป้าหมายของผลงานและกระบวนการทำงานวนรอบซ้ำๆแบบนี้โดยไม่มีที่สิ้นสุด

SCRUM



รูปภาพที่ 1: Scrum

Scrum Roles

ใน Scrum เราแบ่งหน้าที่ออกเป็น 3 บทบาทด้วยกัน ได้แก่ Product Owner, ทีม และ ScrumMaster รวมทั้งหมดเรียกว่าทีม Scrum Product Owner คือคือผู้รับผิดชอบในการทำให้เกิด return on investment (ROI) สูงที่สุดด้วยการทำการขายการคุณสมบัติต่างๆของผลิตภัณฑ์ จัดเรียงรายการทั้งหมดตามลำดับความสำคัญ ตัดสินใจว่าสิ่งใดมีความสำคัญที่สุดเป็นลำดับแรกใน Sprint ถัดไป และสุดท้ายหมั่นจัดเรียงลำดับความสำคัญและปรับปรุงรายการนี้อย่างสม่ำเสมอ สำหรับผลงานที่สร้างรายได้ในท้องตลาดแล้ว Product Owner คือผู้รับผิดชอบในส่วนได้ส่วนเสียของผลงาน แต่สำหรับ application ที่ใช้ภายในองค์กร Product Owner ไม่ต้องรับผิดชอบในเรื่องการสร้างรายได้ แต่ยังต้องรับผิดชอบในการสร้างความคุ้มค่าของผลงานให้สูงที่สุด ด้วยการเลือกทำสิ่งที่ได้ประโยชน์ในเชิงธุรกิจสูงสุด ด้วยต้นทุนต่ำสุดในแต่ละรอบของ Sprint ในทางปฏิบัติ "คุณค่า" มีความหมายคลุมเครือ และลำดับความสำคัญบางที่ขึ้นอยู่กับปัจจัยหลายอย่าง เช่น ความพึงพอใจของลูกค้า ความเข้ากันได้กับเป้าหมายของแผนกลยุทธ์ กำจัดความเสี่ยง การปรับปรุงพัฒนา หรือปัจจัยอื่นๆที่เกี่ยวข้อง ในบางครั้ง Product Owner และลูกค้าจะเป็นคนเดียวกัน ซึ่งเป็นปกติสำหรับ application ที่ใช้

ภายในองค์กรหรือลูกค้าอาจจะเป็นคนนับล้านที่มีความต้องการหลากหลาย ในกรณีนี้บทบาทของ Product Owner จะคล้ายกับตำแหน่งผู้จัดการฝ่ายผลิตภัณฑ์ หรือผู้จัดการด้านการตลาดเหมือนในองค์กรอื่นๆ อย่างไรก็ตาม มีบางสิ่งที่ Product Owner ต่างจากตำแหน่งผู้จัดการฝ่ายผลิตภัณฑ์แบบเดิมๆ เพราะพวกเขาจะยืนที่จะติดต่อพูดคุยหรือมีปฏิสัมพันธ์กับทีมงานอย่างสม่ำเสมอ ให้ลำดับความสำคัญด้วยตัวเองและตรวจสอบผลลัพธ์ในแต่ละรอบทุกๆ สองหรือสี่สัปดาห์ แทนที่จะมอบหมายให้ผู้จัดการโครงการเป็นผู้ตัดสินใจในเรื่องต่างๆ เรื่องสำคัญมากมายที่ต้องจำไว้คือใน Scrum มีเพียงหนึ่งเดียว และผู้เดียวที่มีอำนาจในการตัดสินใจในขั้นสุดท้ายคือ Product Owner เขาหรือเธอผู้นั้นต้องเป็นผู้รับผิดชอบคุณค่าของผลงานที่ออกมา

Team (ทีม) เป็นผู้สร้างผลงานตามที่ Product Owner ชี้นำ ยกตัวอย่างเช่น application หรือ website ต่างๆ ทีม Scrum เป็นการรวมพลังของผู้เชี่ยวชาญในด้านต่างๆ (cross-functional) ที่จำเป็นต่อการส่งมอบผลงานที่พร้อมใช้งานออกไปในทุกๆ รอบ ของ sprint และสามารถจัดการเรื่องต่างๆ ได้ภายในตัวเอง (self-organizing) อย่างอิสระด้วยความรับผิดชอบที่สูงมาก ทีมจะให้สัญญาว่าจะทำอะไรด้วยวิธีการที่ดีที่สุดเพื่อให้ได้ผลลัพธ์ตามคำมั่นสัญญานั้นๆ ตามเรื่องเล่าของ Scrum ทีมจะเปรียบได้กับ “หมู” และคนอื่นๆ ทุกคนเปรียบได้กับ “ไก่” มาจากเรื่องตลกๆ ของหมูและไก่ที่ตกลงใจเข้าหูนเพื่อเปิดภัตตาคารร่วมกันชื่อว่า “แฮมและไข่” (Ham and Eggs) แต่เจ้าหมูก็เกิดกังวลขึ้นมาเพราะ “หมูจะถูกผูกมัด แต่ไก่แค่มีส่วนร่วมเฉยๆ”

ทีมใน Scrum จะมีประมาณ 7 คน บวกกับ 2 คน และสำหรับการพัฒนา software ทีมงานอาจจะรวมตัวจากคนที่มีความสามารถในด้านการวิเคราะห์, การพัฒนา software, การทดสอบ, การออกแบบส่วนติดต่อกับผู้ใช้, การออกแบบระบบฐานข้อมูล, สถาปัตยกรรมระบบ, การจัดทำเอกสาร และอื่นๆ เข้าด้วยกัน ทีมจะพัฒนาผลงานและเสนอความคิดใหม่ๆ ให้ Product Owner ทราบว่าจะทำอะไรให้ได้ผลงานชิ้นเทพออกมา ใน Scrum ทีมจะมีประสิทธิภาพและประสิทธิผลมากที่สุดถ้าลูกทีมทุกคนทุ่มเวลาทำงานทั้งหมด 100% ให้กับงานเพียงโครงการเดียวต่อหนึ่งรอบ Sprint หลักเล็งการทำงานหลายๆ โครงการหรือหลายผลิตภัณฑ์ในเวลาเดียวกัน รวมไปถึงหลักเล็งการเปลี่ยนแปลงสมาชิกในทีมด้วย เพื่อประสิทธิผลที่สูงกว่าเดิมกลุ่มของทีมพัฒนา application ที่มีสมาชิกจำนวนมาก จะถูกจัดออกเป็นทีม Scrum หลายๆ ทีม โดยแต่ละทีมจะสนใจในคุณสมบัติที่แตกต่างกันของผลิตภัณฑ์นั้นๆ และพยายามประสานงานร่วมกันอย่างใกล้ชิด เนื่องจากทีมต้องทำงานในทุกด้านตั้งแต่การวางแผนงาน การวิเคราะห์ การพัฒนาโปรแกรม และการทดสอบผลงาน เพื่อให้ผลงานเสร็จสิ้นในมุมมองของลูกค้า ทีมจึงถูกเรียกว่า feature team

ScrumMaster เป็นผู้ช่วยให้ทีมงานเรียนรู้และประยุกต์ใช้ Scrum เพื่อบรรลุเป้าหมายทางธุรกิจ เขาจะทำทุกอย่างที่อยู่ในอำนาจของตัวเองเพื่อให้ทีมงานและ Product Owner ประสบความสำเร็จ ScrumMaster ไม่ใช่ผู้จัดการของทีม หรือผู้จัดการโครงการ แต่เป็นช่วยเหลือทีมด้วยการป้องกันสิ่งรบกวนต่างๆ จากภายนอก ให้ความรู้และแนะนำ Product Owner และทีมงานให้เชี่ยวชาญในกระบวนการแบบ Scrum

ScrumMaster ต้องทำให้ทุกคนทั้ง Product Owner และเหล่าผู้บริหารเข้าใจและคล้อยตามแนวทางของ Scrum พวกเขาจะช่วยนำพาหน่วยงานผ่านอุปสรรคต่างๆ ที่มีผลต่อความสำเร็จด้วยการพัฒนาแบบ Agile ผลของ Scrum จะทำให้

มองเห็นปัญหาและสิ่งคุกคามต่างๆ ที่มีผลต่อประสิทธิภาพของทีมและ Product Owner มันจึงสำคัญมากที่ ScrumMaster ต้องมุ่งมั่นทำงานด้วยพลังที่มีอยู่เพื่อช่วยคลี่คลายปัญหาต่างๆ ไมอย่างนั้นทีมและ Product Owner ก็ยากที่จะพบกับความสำเร็จ ดังนั้นเราควรแยก ScrumMaster ออกมาต่างหากเพื่อให้ทำหน้าที่ได้เต็มเวลา แม้แต่สำหรับทีมเล็กๆ จะต้องมีส่วนในทีมหนึ่งคนทำหน้าที่นี้ควบคู่ไปกับการทำงานในหน้าที่ปกติอื่นๆ ให้น้อยลง โดย ScrumMaster ขึ้นเทพสามารถมาได้จากพื้นเพต่างๆ กันเช่น วิศวกร นักออกแบบ ผู้ทดสอบผลงาน ผู้จัดการฝ่ายผลิตภัณฑ์ ผู้จัดการโครงการ หรือผู้ควบคุมคุณภาพของผลงานก็ได้

ScrumMaster และ Product Owner จะต้องไม่ใช่คนๆ เดียวกัน เพราะในบางครั้ง ScrumMaster จะต้องทักท้วงการทำงานของ Product Owner กลับไป ตัวอย่างเช่น ถ้าพวกเขาต้องการให้ทีมส่งมอบสิ่งใหม่ในช่วงกลางของ Sprint และ ScrumMaster ไม่เหมือนผู้จัดการโครงการ ตรงที่ ScrumMaster ไม่บอกว่าใครต้องทำอะไร หรือมอบหมายงานให้ผู้อื่น พวกเขาเพียงช่วยอำนวยความสะดวกในแง่กระบวนการทำงาน สนับสนุนทีมงานให้บริหารจัดการได้ด้วยตัวเอง ถ้า ScrumMaster เคยอยู่ในตำแหน่งของผู้จัดการของทีมมาก่อน พวกเขาจะต้องเปลี่ยนแปลงวิธีคิดและรูปแบบในการทำงานร่วมกับทีมให้ประสบความสำเร็จด้วย Scrum

ให้สังเกตว่าใน Scrum ไม่มีผู้จัดการโครงการเพราะว่าไม่จำเป็นต้องมี ความรับผิดชอบแบบเดิมๆ ของผู้จัดการโครงการถูกกระจายและเปลี่ยนมือให้กับ 3 บทบาทใน Scrum ในบางครั้งอดีตผู้จัดการโครงการสามารถก้าวเข้ามาทำหน้าที่เป็น ScrumMaster ได้ แต่เชื่อว่าจะไม่ประสบความสำเร็จ เพราะโดยพื้นฐานแล้วทั้งสองบทบาทมีความแตกต่างกัน ทั้งหน้าที่ความรับผิดชอบในแต่ละวันและวิธีคิดที่ต้องใช้เพื่อให้ประสบความสำเร็จ วิธีที่ดีในการทำความเข้าใจบทบาทของ ScrumMaster และการพัฒนาทักษะที่จำเป็นเพื่อความสำเร็จคือเข้ารับการอบรมในหลักสูตรประกาศนียบัตร ScrumMaster จากเครือ Scrum Alliance (the Scrum Alliance's Certified ScrumMaster)

นอกเหนือจากบทบาททั้งสามนี้ ยังมีผู้อื่นที่มีส่วนร่วมในการสร้างผลงานให้สำเร็จรวมไปถึงผู้จัดการในด้านต่างๆ ตัวอย่างเช่น ผู้จัดการด้านวิศวกรรม แม้ว่าบทบาทของพวกเขาจะเปลี่ยนไปใน Scrum แต่คุณค่าของพวกเขายังคงมีอยู่ ยกตัวอย่างเช่น

- พวกเขาช่วยสนับสนุนทีมงาน ด้วยความเคารพในกฎและจิตวิญญาณของ Scrum
- พวกเขาช่วยจัดสิ่งคุกคามต่างๆ ที่ทีมงานและ Product Owner ค้นพบ
- พวกเขาคอยให้คำปรึกษา ในด้านที่พวกเขาเชี่ยวชาญและจากประสบการณ์ที่มีอยู่

ใน Scrum เวลาของคนเหล่านี้ที่เคยหมดไปกับบทบาทของ “พี่เลี้ยงเด็ก” ด้วยการมอบหมายงาน ติดตามรายงานความคืบหน้าต่างๆ และรูปแบบอื่นๆ ที่เป็นการจัดการแบบกำกับทุกกระเบียดนิ้ว จะถูกแทนด้วยบทบาทของเหล่า “เทพ” และ “คนรับใช้” ของทีมงาน ด้วยการให้คำปรึกษา สอนงาน ช่วยจัดอุปสรรคต่างๆ นานา ช่วยแก้ไขปัญา ให้ความคิดเห็นเชิงสร้างสรรค์ และนำทางทีมงานให้พัฒนาทักษะที่จำเป็นในการสร้างสรรค์ผลงานชิ้นเยี่ยมออกมา

ในการเปลี่ยนแปลงนี้เหล่าผู้จัดการอาจจะต้องเปลี่ยนแปลงรูปแบบในการบริหารจัดการ ตัวอย่างเช่น ทำเนียบเป็นไม่รู้ที่มาที่ไปของปัญหา และถามทีมงานด้วยคำถามต่างๆ ให้ทีมงานได้ขบคิด ถกเถียง และถามคำถามซึ่งกันและกันภายในทีม (Socratic

Questioning) จนกระทั่งทีมงานค้นพบแนวทางในการแก้ปัญหาด้วยตัวเอง แทนที่จะตัดสินใจแนวทางแก้ปัญหา และมอบหมายให้ทีมดำเนินการต่อไป เจกเช่นทุกวันที่ผ่านมา

Starting Scrum

ก้าวแรกของ Scrum เริ่มจาก Product Owner วางแนวทางและเป้าหมายของผลิตภัณฑ์ ซึ่งจะกลายเป็นรายการของคุณสมบัติต่างๆ เรียงตามลำดับความสำคัญซึ่งเรียกว่า Product Backlog ซึ่ง Product Backlog นี้ที่จริงแล้วก็คือแผนงานการพัฒนาผลิตภัณฑ์ (ภาพที่ 2) ที่ใช้ตลอดช่วงการพัฒนาผลิตภัณฑ์นั่นเอง คำจำกัดความของ Product Backlog คือ “อะไรก็ตามที่ทีมสามารถทำให้สำเร็จได้ตามลำดับความสำคัญ” ตัว Product Backlog เองจะมีได้เพียงหนึ่งเดียวเท่านั้น โดย Product Owner ต้องจัดเรียงลำดับความสำคัญและตัดสินใจจากแง่มุมต่างๆ ให้ครบถ้วน ทั้งในมุมมองของผู้ที่มีส่วนได้ส่วนเสียและข้อเสนอแนะจากทีมงาน

Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	New Estimates of Effort Remaining as of Sprint...					
					1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5						
As a buyer, I want to remove a book in a shopping cart	...	2	6	2						
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13						
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20						
Upgrade all servers to Apache 2.2.3	...	5	5	13						
Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	6	2	3						
As a shopper, I want to create and save a wish list	...	7	7	40						
As a shopper, I want to add or delete items on my wish list	...	8	4	20						

Figure 2. The Product Backlog

รูปภาพที่ 2: Product Backlog

Product Backlog ประกอบด้วยสิ่งต่างๆที่หลากหลายเช่น ลูกเล่นสำหรับลูกค้าใหม่ (“ลูกค้าทุกคนเลือกหนังสือใส่ตะกร้าสินค้าได้”), การเพิ่มประสิทธิภาพเชิงระบบ (“ปรับปรุงระบบประมวลผล transaction ให้สามารถขยายเพิ่มเติมรองรับข้อมูลที่มากขึ้นได้”), งานทางด้านเทคนิคและวิจัย (“หาวิธีปรับปรุงความเร็วในการยืนยันบัตรเครดิต”), และข้อผิดพลาดต่างๆถ้าหากมีจำนวนไม่มากนัก (“หาสาเหตุและแก้ไขข้อผิดพลาดใน script ของระบบประมวลผลคำสั่งซื้อ”) (โดยปกติระบบที่มีข้อบกพร่องจำนวนมากมักมีระบบติดตามข้อผิดพลาดแยกต่างหาก) รายการใน Product Backlog จะต้องมีความชัดเจนและถูกต้องอยู่เสมอ ซึ่งโดยส่วนใหญ่จะนำมาจาก Use Case หรือ “user stories” เพื่อสื่อถึงคุณค่าที่ลูกค้าจะได้รับจากผลงาน

Product Backlog บางส่วนที่ตั้งเป้าไว้ว่าจะปล่อยออกมาให้ใช้งานจะเรียกว่า Release Backlog ซึ่ง Product Owner จะต้องให้ความสำคัญเป็นลำดับแรก

Product Owner จะต้องหมั่นปรับปรุง Product Backlog อยู่เสมอ ตามการเปลี่ยนแปลงที่เกิดจากความต้องการของลูกค้า, แนวความคิดใหม่ๆ, การเปลี่ยนแปลงของคู่แข่ง, อุปสรรคเชิงเทคนิค และอื่นๆ Product Owner จะรับผิดชอบต่อการประเมินคุณค่าทางธุรกิจของงานและให้ทีมเป็นคนประเมินความยากง่ายในการทำงานแต่ละงานใน Product Backlog สำหรับการประเมินคุณค่าทางธุรกิจอาจจะเป็นสิ่งที่ Product Owner ไม่ถนัด ScrumMaster สามารถเข้ามาช่วย Product Owner เรียนรู้การทำงานในส่วนนี้ได้ จากการประเมินทั้งสองอย่างนี้ (ความยากง่ายและคุณค่า) และอาจจะประเมินความเสี่ยงเพิ่มเติม Product Owner จะนำมาจัดลำดับความสำคัญของ Product Backlog (ซึ่งจริงๆแล้วตามปกติจะเป็นส่วนหนึ่งของ Release Backlog) เพื่อทำให้เกิด ROI สูงสุด (การเลือกงานที่มีคุณค่าสูงสุดและทำได้ง่ายที่สุด) หรืออีกนัยหนึ่งก็เพื่อลดความเสี่ยงต่าง ๆ นั้นเอง ในภายหลังเราจะพบว่า การประเมินเหล่านี้จะถูกปรับเปลี่ยนในแต่ละ Sprint เมื่อทีมมีประสบการณ์เพิ่มมากขึ้น และการปรับเปลี่ยนลำดับความสำคัญอย่างต่อเนื่องของ Product Backlog จะส่งผลให้เกิดการกระบวนการพัฒนาโดยตลอดนั่นเอง

Scrum ไม่ได้กำหนดเทคนิคที่ชัดเจนใน การบ่งบอกหรือจัดลำดับของงานรวมถึงการประเมินต่างๆใน Product Backlog แต่เทคนิคที่ใช้กันทั่วไปคือการประเมินในลักษณะของค่าเทียบเคียง (ดูจากความยากง่าย ความซับซ้อน และความเสี่ยง) โดยกำหนดหน่วยเป็น “จำนวนแต้มของเนื้อหา” (story points) หรือเรียกง่าย ๆ ว่า “แต้ม” (points)

ระหว่างการทำงาน ทีมจะคอยติดตามดูปริมาณงานที่สามารถทำได้ในแต่ละ Sprint ตัวอย่างเช่นเฉลี่ย 26 แต้มต่อ Sprint เป็นต้น ด้วยข้อมูลนี้ทำให้เราสามารถประเมินวันเปิดตัวผลงานที่มีคุณลักษณะครบถ้วนสมบูรณ์ได้ หรือสามารถบอกได้ว่าจะมีงานอะไรบ้างที่เราสามารถทำเสร็จได้ในวันที่กำหนด ถ้าหากว่าค่าเฉลี่ยการทำงานคงที่และไม่มีการเปลี่ยนแปลง ค่าเฉลี่ยนี้เราเรียกว่า “อัตราเร็ว” (velocity) ของทีม หน่วยของความเร็วของทีมจะเป็นหน่วยเดียวกับหน่วยประเมินของงานใน Product Backlog

งานใน Product Backlog สามารถปรับเปลี่ยนได้ตามขนาดหรือความยากง่ายของงาน งานชิ้นใหญ่ๆ อาจจะถูกแตกย่อยเป็นชิ้นเล็กๆ และงานชิ้นเล็กก็อาจจะถูกรวมเข้าด้วยกันในระหว่าง Product Backlog Refinement หรือ การประชุมวางแผน Sprint งานใน Product Backlog สำหรับ 2-3 Sprint ข้างหน้าควรมีขนาดเล็กและชัดเจนเพียงพอที่ทีมสามารถทำความเข้าใจได้ตรงกัน และให้คำมั่นว่าจะทำเสร็จได้ระหว่างการประชุมวางแผน Sprint ซึ่งขนาดของงานที่เหมาะสมนี้เรียกว่าขนาดที่ “สามารถทำสำเร็จได้”

ความเชื่ออย่างหนึ่งของ Scrum คือ ไม่มีความจำเป็นต้องเขียนรายละเอียด specification แต่ในความเป็นจริงแล้วมันขึ้นอยู่กับ Product Owner และทีมเป็นผู้ตัดสินใจว่ารายละเอียดแค่ไหนจึงจะเพียงพอ ซึ่งจะต่างกันไปแล้วแต่ความเข้าใจของทีมรวมถึงตัวแปรอื่นๆด้วยสำหรับแต่ละงานใน backlog สิ่งสำคัญคือให้ระบุว่าอะไรคือใจความสำคัญเท่าที่จำเป็น หรืออีกนัยหนึ่งคือระบุให้ชัดเจนว่าอะไรคือสิ่งสำคัญที่ต้องทำความเข้าใจโดยไม่ต้องอธิบายทุกรายละเอียดของงาน งานที่มีลำดับความสำคัญต่ำยังไม่ถูกทำในอนาคตรันไกล์ ซึ่งโดยปกติจะมีข้อมูลเพียง “หยาบๆ” มีขนาดใหญ่ และไม่มีรายละเอียดของความต้องการมากนัก และในทางกลับกันงานที่มีลำดับความสำคัญสูงและกำลังจะถูกทำในลำดับถัดไป ก็ควรมีรายละเอียดครบถ้วนสมบูรณ์กว่า

Sprint Planning

ในช่วงเริ่มต้นของแต่ละ Sprint จะมีการประชุม Sprint Planning ซึ่งจะถูกแบ่งออกเป็น 2 ช่วง โดยช่วงแรกจะเรียกว่า Sprint Planning ส่วนที่ 1

ใน Sprint Planning ส่วนแรก, Product Owner และทีม (โดยมี ScrumMaster คอยแนะนำ) ช่วยกันเลือกงานที่มีลำดับความสำคัญสูงจาก Product Backlog ซึ่งคืองานที่ Product Owner ให้ความสนใจมาทำใน Sprint นี้ พวกเขาจะประชุมหารือเพื่อกำหนดเป้าหมายและบริบทของงานที่มีความสำคัญใน Product Backlog และทำความเข้าใจในสิ่งที่ Product Owner ต้องการอย่างลึกซึ้ง Product Owner และทีมยังช่วยกันทบทวน “Definition of Done” (ที่ได้ถูกตั้งไว้ก่อนหน้านี้) ที่งานทั้งหมดจะต้องทำให้ได้ตามนั้น ตัวอย่างเช่น “สำเร็จหมายถึงการเขียนคำสั่งโปรแกรมให้ได้ตามมาตรฐาน, การตรวจทาน, การนำ test-driven development (TDD) มาใช้ และทดสอบงานทั้งหมดโดยใช้ระบบทดสอบอัตโนมัติ, การรวม code เข้าด้วยกัน และจัดทำเอกสาร” ในขั้นตอนส่วนที่ 1 นี้ จะเน้นที่ทำความเข้าใจสิ่งที่ Product Owner ต้องการ จากกฎเกณฑ์ของ Scrum เมื่อสิ้นสุดขั้นตอนส่วนที่ 1 แล้ว Product Owner (ที่มักจะอยู่อยู่เสมอ) อาจจะออกจากการประชุมไปก่อน แต่ก็ควรจะยังสามารถติดต่อได้ (เช่นทางโทรศัพท์) ในระหว่างการประชุมถัดไป อย่างไรก็ตามการประชุมในส่วนที่ 2 ก็ยินดีให้ Product Owner เข้าร่วมประชุมด้วยเสมอ

Sprint Planning ส่วนที่ 2 จะมุ่งเน้นในเรื่องการวางแผนงานอย่างลงรายละเอียดถึงวิธีการทำชิ้นงานที่ทีมตัดสินใจเลือกขึ้นมาทำก่อน ทีมจะเลือกงานจาก Product Backlog ที่ให้คำมั่นว่าจะทำให้เสร็จภายในรอบ Sprint นี้ โดยเริ่มจากงานลำดับบนสุดใน Product Backlog (หรืออีกนัยหนึ่งคือเริ่มจากงานในลำดับที่มีความสำคัญสูงสุดสำหรับ Product Owner) แล้วค่อยเลือกงานตามลำดับถัดไปในรายการ หัวใจของการทำงานแบบ Scrum คือ ทีมเป็นผู้ตัดสินใจว่าจะสามารถทำงานให้สำเร็จได้เท่าไร แทนที่จะถูกมอบหมายงานให้โดย Product Owner ซึ่งจะทำให้ทีมสามารถให้คำมั่นที่เชื่อถือได้ เนื่องจากทีมตัดสินใจด้วยพื้นฐานจากการวิเคราะห์และวางแผนด้วยตนเองมากกว่ามีคนอื่นมาตัดสินใจให้ ในขณะที่ Product Owner ไม่ใช่ผู้ที่กำหนดปริมาณงานที่ทีมสามารถทำให้สำเร็จได้ แต่เขาก็จะมั่นใจได้ว่างานที่ทีมจะต้องทำจะถูกเลือกจากงานลำดับบนสุดใน Product Backlog หรืออีกนัยหนึ่ง งานที่เขาหรือเธอให้ความสำคัญมากที่สุดนั่นเอง ทีมสามารถที่จะชักจูงได้ว่าเลือกทำงานใดบ้างจากรายการทั้งหมด ซึ่งเป็นเรื่องปกติเมื่อทีมและ Product Owner เข้าใจตรงกันว่าม้งานบางอย่างที่สำคญน้อยกว่าแต่ทำได้ง่ายและเหมาะสมกว่า เมื่อเทียบกับงานที่มีความสำคัญมากกว่า

การประชุม Sprint Planning ส่วนมากจะเสร็จสิ้นภายในเวลาไม่กี่ชั่วโมงแต่ไม่ควรเกิน 8 ชั่วโมงสำหรับ Sprint ระยะเวลา 4 สัปดาห์ ทีมต้องตัดสินใจด้วยความระมัดระวังอย่างยิ่งเพื่อที่จะสามารถให้คำมั่นว่าจะทำงานให้สำเร็จได้ ทีมอาจจะเริ่ม Sprint Planning ส่วนที่ 2 ด้วยการประเมินเวลาที่สมาชิกแต่ละคนในทีมมีให้กับงานใน Sprint นี้ หรือคิดง่ายๆโดยเอาค่าเฉลี่ยจำนวนวันทำงานลบด้วยเวลาที่ใช้ในการประชุม, อ่าน email, พักกลางวัน และอื่นๆ ซึ่งสำหรับคนส่วน

ใหญ่จะมีเวลาสำหรับทำงานประมาณ 4-6 ชั่วโมงต่อวัน และนี่จะถูกกำหนดเป็นขีดความสามารถของทีมสำหรับ Sprint ที่กำลังจะทำดูภาพที่ 3 ประกอบ

Sprint Length		2 weeks	
Workdays during Sprint		8 days	

Team Member	Available Days During Sprint*	Available Hours per Day	Total Available Hours
Tracy	8	4	32
Sanjay	7	5	35
Phillip	8	4	32
Jing	6	5	30

* Net of vacation and other days out of office

รูปภาพที่ 3: Estimating Available Hours

เมื่อเราเข้าใจขีดความสามารถของทีมแล้ว ทีมจะคำนวณว่าพวกเขาจะทำงานใน Product Backlog ให้เสร็จทันเวลาได้กี่ชิ้น และทำด้วยวิธีการใด โดยทั่วไปการออกแบบจะเริ่มจากกระดานไวท์บอร์ด จนเมื่อทีมมีความเข้าใจตรงกันตามที่ออกแบบไว้แล้ว ทีมจะแตกรายละเอียดของงานใน Product Backlog กระจายออกเป็นงานชิ้นเล็กๆ ทีมจะทำงานจากงานชิ้นแรกที่อยู่ด้านบนสุดของ Product Backlog หรืออีกนัยหนึ่ง งานที่ Product Owner มองว่าสำคัญที่สุดนั่นเอง และจะเริ่มระดมสมองในการแตกรายละเอียดลงมาเป็นงานเล็กๆ ของแต่ละคน และรวบรวมไว้ในเอกสารที่เรียกว่า Sprint Backlog (ภาพที่ 4) ตามที่เคยบอกไว้ Product Owner จะต้องพร้อมเสมอในตอนที่ทำ Sprint Planning ส่วนที่ 2 เช่น พร้อมให้โทรหาได้ทันทีเพื่อจะที่ตอบข้อสงสัยทุกอย่างที่เป็นไปได้ ทีมจะแตกรายละเอียดของงานไล่ไปตามลำดับที่วางไว้ใน Product Backlog จนกระทั่งครบถ้วนตามที่ได้ประเมินเอาไว้ เมื่อเสร็จการประชุม ทีมจะสร้างรายการของงานย่อยๆ ทั้งหมดพร้อมขนาดที่ประเมินเอาไว้ ซึ่งโดยปกติจะประเมินกันด้วยจำนวนชั่วโมงหรือส่วนย่อยๆ ของวันแต่ละวัน

Scrum สนับสนุนให้เราทำงานได้หลายๆหน้าที่ มากกว่าที่จะทำตาม “ตำแหน่ง” เช่น “นักทดสอบระบบ” ก็ทำแต่เรื่องของการทดสอบระบบ หรือพูดอีกอย่างคือ สมาชิกในทีมจะต้อง “หางานอะไรสักอย่างทำ” และให้ความช่วยเหลือทีมเท่าที่เป็นไปได้ เช่นถ้ามีงานทดสอบระบบมากเกินไป สมาชิกทีมทุกคนจะต้องช่วยกันทดสอบระบบ แต่ก็ไม่ได้แปลว่าคนในทีมทุกคนจะต้องเป็นเทพไปหมดทุกอย่าง เพราะบางคนก็เด่นมากในด้านการทดสอบ (หรือทักษะอื่นๆ) แต่สมาชิกในทีมจะทำงานร่วมกันและเรียนรู้ทักษะใหม่ๆ จากผู้อื่นไปพร้อมๆกัน ด้วยเหตุนี้ ตอนที่ทีมประเมินและแตกงานย่อยๆ ใน Sprint Planning ก็ไม่จำเป็นที่จะมีคนๆหนึ่งจะอาสาทำทุกอย่าง “ที่พวกเขาทำได้ดีที่สุด” สิ่งที่ดีกว่าการทำงานอยู่เรื่องเดียวนั่นคือ เมื่อเวลาที่จะหยิบงานใหม่ ให้เลือกจากความคิดว่าเราอยากเรียนรู้งานด้านไหนเพิ่มเติม ด้วยการทำงานเป็นคู่หูกับผู้เชี่ยวชาญด้านนั้นๆ

ที่ว่ามาทั้งหมด มีกรณีที่เกิดขึ้นได้ยากเหมือนกันเช่น จอห์น นูโว มีความเชี่ยวชาญเฉพาะด้านที่คนอื่นในทีมทำตามได้ยากหรือใช้เวลานานมากในการเรียนรู้ เช่น จอห์น เป็นคนเดียวที่มีฝีมือในการวาดภาพ ในขณะที่คนอื่นๆวาดอะไรไม่ได้เลย แม้กระทั่ง “มนุษย์ไม่ขีด” หากเป็นเช่นนั้น ถ้ามันไม่ใช่เรื่องยากและทีมสามารถที่จะเรียนรู้ได้แสดงว่ามีบางอย่างผิดปกติ บางทีก็จำเป็นต้องถามว่า ถ้าทีมวางแผนให้จอห์นเป็นคนวาดภาพทั้งหมด จะทำได้หมดใน Sprint นั้นๆ หรือไม่

ทีมงานหลายๆทีม จะมีเครื่องมือบางอย่างในการติดตามงาน ด้วยบอร์ดขนาดกำลังดีติดอยู่บนกำแพง และแปะงานย่อยๆต่างๆ ที่เขียนบนกระดาษ Post-It ที่ย้ายที่ไปตามแถวต่างๆ ที่มีป้ายแปะว่า “ยังไม่เริ่ม (Not Yet Started)” , “กำลังทำ (In Progress)” และ “เสร็จแล้ว (Completed)” เช่นในภาพที่ 5

Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort	New Estimates of Effort Remaining as of Day...					
				1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart	modify database		5						
	create webpage (UI)		8						
	create webpage (Javascript logic)		13						
	write automated acceptance tests		13						
	update buyer help webpage		3						
	...								
Improve transaction processing performance	merge DCP code and complete layer-level tests		5						
	complete machine order for pRank		8						
	change DCP and reader to use pRank http API		13						

รูปภาพที่ 4: Sprint Backlog



รูปภาพที่ 5: Visual Management – Sprint Backlog tasks on the wall

จุดสำคัญอย่างหนึ่งของ Scrum ก็คือ เมื่อทีมตกลงว่าจะทำอะไรแล้ว ถ้ามีอะไรเพิ่มเติมหรือเปลี่ยนแปลงก็ต้องถูกยกไปทำใน Sprint ถัดไป นี่หมายความว่าถ้าอยู่กลาง Sprint แล้ว Project Owner อาจจะเพิ่มงานเข้ามาให้ทีมทำ ก็ยังทำไม่ได้จนกว่าจะถึง Sprint หน้า ถ้าเกิดสถานการณ์เปลี่ยนแปลงจนทำให้ลำดับความสำคัญของงานต้องเปลี่ยนไป จนถึงขั้นว่าถ้าทีมยังทำงานเดิมต่อไปก็เป็นการเสียเวลาเปล่าๆ ทั้งตัวทีมและ Product Owner ก็สามารถจะยกเลิก Sprint นั้นได้ ซึ่งหลังจากทีมหยุดทำงานแล้ว ก็ประชุม Sprint Planning กันเพื่อเริ่มต้น Sprint ใหม่ การทำให้มันดูเป็นเรื่องใหญ่แบบนี้ก็ดีแล้วเพราะจะได้ทำให้ไม่อยากทำกัน

การที่ทีมได้รับการปกป้องจากการเปลี่ยนแปลงของเป้าหมายในระหว่าง Sprint จะส่งผลกระทบที่ดีอย่างมากต่อทีม อย่างแรกก็คือ ทีมสามารถทำงานได้อย่างมั่นใจว่า สิ่งที่ตกลงกันไว้จะไม่มีการมาเปลี่ยนกัน ทำให้ทีมมุ่งหน้าทำงานเพื่อให้เสร็จตามที่ตกลงกันไว้ อีกอย่างหนึ่งคือมันทำให้ Product Owner คิดมาอย่างดีก่อนว่าอะไรต้องทำเป็นอันดับต้นๆใน Product Backlog เพื่อที่จะส่งต่อไปกับทีม

การทำงานตามแบบของ Scrum จะส่งผลดีกับ Product Owner สองอย่าง อย่างแรกคือ Product Owner จะมั่นใจได้ว่า ทีมได้ตกลงทำงานมากหรือน้อย ตามที่ตัวทีมเองเป็นคนเลือกกว่าทำได้จริงซึ่งก็จะช่วยให้ทีมสามารถกำหนดจำนวนของงานที่จะทำได้แม่นยำมากขึ้นเรื่อยๆ อย่างที่สอง คือ Product Owner สามารถจะเปลี่ยนแปลงอะไรก็ได้ใน Product Backlog ก่อนที่จะถึง Sprint หน้า ในช่วงเวลานี้จะเพิ่ม ลบ แก้ไข หรือ สับเปลี่ยนตำแหน่งอย่างไรก็ได้ แต่ไม่สามารถไปแก้งานที่ทีมเลือกไปทำแล้วในช่วง Sprint ที่กำลังทำอยู่ อย่างไรก็ตาม Product Owner ก็ต้องรอเป็นระยะเวลายาวแค่ Sprint เดียวก่อนที่จะมารับงานใหม่ที่ยากให้ทำ การทำแบบนี้ทำให้ความน่ากลัวของการเปลี่ยนแปลงน้อยลงไป ไม่ว่าจะเป็นการเปลี่ยนแปลง เปลี่ยน requirement หรือ เกิดอยากจะเปลี่ยนใจขึ้นมาเฉยๆ นี่อาจจะเป็นเหตุผลสำคัญที่ Product Owner มักจะตื่นเต้นกับ Scrum มากพอๆ กับทุกคน

Daily Scrum

เมื่อเริ่ม Sprint แล้ว ทีมต้องมีหลักปฏิบัติร่วมกันที่เรียกว่า Daily Scrum ซึ่งก็คือการประชุมกันสั้นๆไม่เกิน 15 นาทีในทุกวันตามเวลาที่นัดหมายกันไว้โดยเน้นว่าทุกคนต้องเข้าร่วม และเพื่อให้การประชุมกระชับขึ้นเราขอแนะนำว่าควรให้ทุกคนยึดถือการประชุม Daily Scrum เป็นช่วงเวลาที่มีจะพูดคุยแลกเปลี่ยนการทำงานของตัวเองและปัญหาที่พบเจอให้กับทีมได้รับทราบร่วมกัน ในการประชุมนี้ทุกคนจะต้องรายงาน 3 เรื่อง (ย้ำว่า 3 เรื่องเท่านั้น) ได้แก่ (1) มีอะไรที่ทำเสร็จไปแล้วบ้างหลังจากการประชุมกันครั้งที่แล้ว (2) อะไรที่กำลังจะทำให้เสร็จก่อนที่จะประชุมกันอีกในครั้งต่อไป (3) ติดขัดหรือมีอุปสรรคหรือไม่ พึงจำไว้ว่า Daily Scrum ไม่ใช้การประชุมกันเพื่อรายงานความคืบหน้าในการทำงานต่อผู้จัดการ แต่เป็นช่วงเวลาที่มีจะจัดการระบบระเบียบของตัวเอง และเข้ามาบอกคนอื่นในทีมว่ามีอะไรเกิดขึ้นบ้าง เพื่อที่จะประสานงานกันได้ อย่างถูกต้อง บางคนอาจจะจดรายการปัญหาที่เจอมา แล้ว ScrumMaster จะเป็นผู้ผลักดันให้ทีมแก้ไขปัญหาเหล่านั้น แต่จะไม่มีการประชุมหรือกันในช่วงเวลาของการ Daily Scrum มีเพียงแต่การรายงาน 3 เรื่องเท่านั้น หากจำเป็นต้องมีการปรึกษากันก็ควรจะปรึกษากันในการประชุมติดตามผล ซึ่งจะจัดขึ้นหลังจากเสร็จสิ้น Daily Scrum ทันที และไม่จำเป็นต้องให้ทุกคนเข้าร่วมในการประชุมนี้ การประชุมติดตามผลนั้นเกิดขึ้นเป็นปกติเพื่อให้ทีมสามารถปรับตัวกับข้อมูลที่ได้ใน Daily Scrum หรือพูดอีกอย่างหนึ่งก็คือเป็นอีกกระบวนการหนึ่งที่เกิดขึ้นเพื่อตรวจสอบและปรับแต่งการทำงานใน Daily Scrum นั้น มีคำแนะนำที่ถือเป็นเรื่องปกติก็คือไม่ควรให้ผู้บริหารหรือผู้มีอำนาจเข้าร่วมในการประชุม เพื่อไม่ให้ทีมรู้สึกเหมือนถูกตรวจสอบและถูกกดดัน จนต้องรายงานแต่ความก้าวหน้าที่ดีสวยหรือทุกวัน (ซึ่งเป็นความคาดหวังจอมปลอม) แต่กลับละเลยการรายงานถึงปัญหาที่เกิดขึ้นด้วย ซึ่งมีแนวโน้มจะบ่อนทำลายทีมที่สามารถบริหารตัวเองได้อยู่แล้ว ให้กลายเป็นทีมหุ่น

เจ็ดของผู้บริหารไป ยังมีประโยชน์เสียกว่าหากให้ผู้ที่มีส่วนได้ส่วนเสียเข้ามาร่วมประชุมเพื่อเข้าถึงคนในทีม และอาจช่วยแก้ปัญหา หรืออุปสรรคที่ทีมกำลังเผชิญอยู่เพื่อให้ทีมสามารถเดินหน้าต่อไปได้

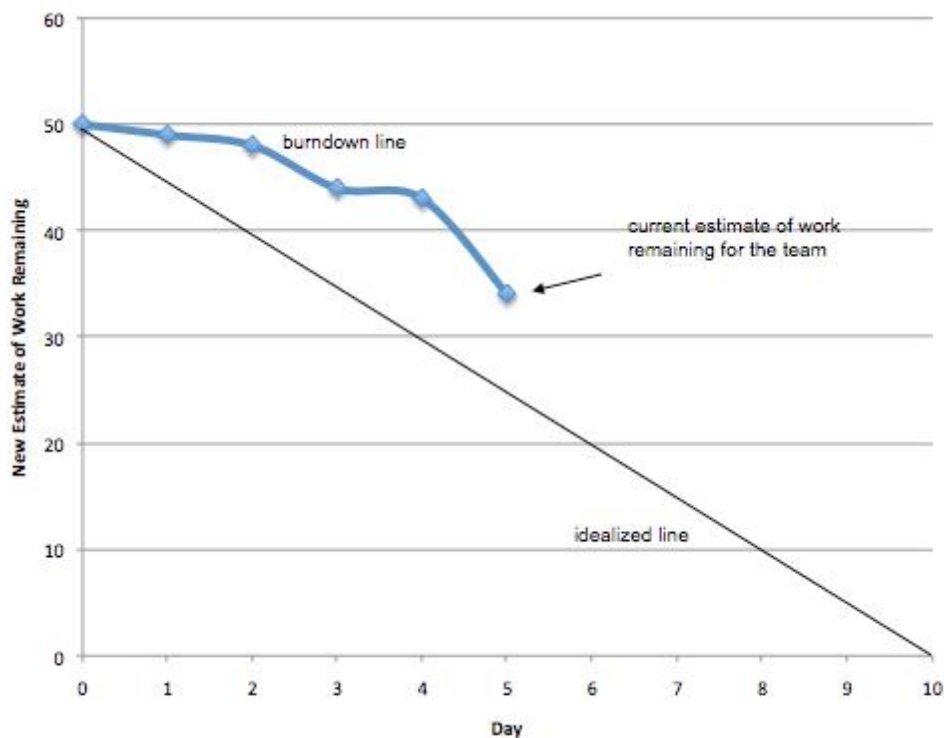
Updating Sprint Backlog & Sprint Burndown Chart

ทีมใน Scrum คือทีมที่ทำงานได้ด้วยตัวเองไม่ได้มีใครมาคอยเจ้ก้เจ้าการสั่ง และรู้จักวิธีการทำงานให้ประสบผลสำเร็จในทุกๆวัน ทีมจะแก้ไขปรับปรุงจำนวนเวลาโดยประมาณของงานที่ยังไม่แล้วเสร็จลงใน Sprint Backlog (ภาพที่ 6) หลังจากนั้นก็ให้ใครสักคนจัดการรวบรวมเวลาที่เหลืออยู่ทั้งหมดของทีมแล้วก็วาดจุดใหม่เพิ่มเข้าไปใน Sprint Burndown Chart (ภาพ ที่ 7) กราฟจะแสดงให้เห็นว่าในแต่ละวันยังมีงานเหลืออยู่อีกเท่าไรจนกว่างานทั้งหมดจะเสร็จสิ้น โดยหน่วยเป็นจำนวนชั่วโมงต่อคน ตามอุดมคติแล้วกราฟนี้จะถึงหัวปักกลางสูงสุดศูนย์ในวันสุดท้ายของ Sprint พอดี ซึ่งหมายความว่าไม่มีงานใดๆเหลืออยู่อีกมันถึงถูกเรียกว่า Burndown ยังไงเล่า ในความเป็นจริงสิ่งที่เกิดขึ้นไม่ได้สวยงามอย่างนั้น สิ่งสำคัญคือกราฟนี้แสดงให้เห็นความคืบหน้าของงานเทียบกับเป้าหมาย ไม่ใช่ว่าทีมใช้เวลาทำงานไปแล้วเท่าไร (ซึ่งไม่ใช่ความคืบหน้าของงานจริงๆ) แต่ชี้ให้เห็นชัดๆว่ายังมีงานเหลืออีกเท่าไรในอนาคต ถ้ากราฟไม่ได้ดังลงไปตามที่ควรจะเป็นเมื่อสิ้นสุด Sprint นั่นแปลว่ากำลังมีอะไรผิดปกติ ทีมจะต้องปรับเปลี่ยนวิธีการทำงาน เช่นลดขอบเขตของงาน หรือจะทำการอะไรให้การทำงานมีประสิทธิภาพมากขึ้นโดยที่ทุกอย่างยังดำเนินต่อไปได้

ตามปกติกราฟ Sprint Burndown สามารถสร้างได้จากโปรแกรม Spread Sheet ทั่วไป แต่หลายๆทีมพบว่าวิธีที่ดีที่สุดคือการวาดลงบนกระดานแผ่นใหญ่ติดไว้บนผนังที่ทำงาน แก้ไขปรับปรุงข้อมูลต่างๆด้วยปากก้าบ้านๆ วิธีนี้ง่ายแต่ได้ผลแบบนี้ เร็วดี!! เรียบง่าย และชัดเจนกว่าข้อมูลบนคอมพิวเตอร์

Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort	New Estimates of Effort Remaining at end of Day...					
				1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart	modify database	Sanjay	5	4	3	0	0	0	
	create webpage (UI)	Jing	3	3	3	2	0	0	
	create webpage (Javascript logic)	Tracy & Sam	2	2	2	2	1	0	
	write automated acceptance tests	Sarah	5	5	5	5	5	0	
	update buyer help webpage	Sanjay & Jing	3	3	3	3	3	0	
Improve transaction processing performance	merge DCP code and complete layer-level tests		5	5	5	5	5	5	
	complete machine order for pRank		3	3	8	8	8	8	
	change DCP and reader to use pRank http API		5	5	5	5	5	5	
						
		Total (person hours)	50	49	48	44	43	34	

รูปภาพที่ 6: Daily Updates of Work Remaining on the Sprint Backlog



รูปภาพที่ 7: Sprint Burndown Chart

Product Backlog Refinement

สิ่งดีๆ เรื่องหนึ่งที่เราไม่ค่อยรู้จักกัน นั่นคือใน Scrum เราควรแบ่งเวลา 5-10% ของแต่ละ Sprint ให้ทีมได้เรียบเรียง หรือปรับปรุงรายการต่างๆ ใน Product Backlog ให้ทีมได้ใช้เวลาวิเคราะห์ในรายละเอียดว่าลูกค้าต้องการอะไร แบ่งรายการที่มีขนาดใหญ่ออกเป็นส่วนเล็กๆ ประเมินขนาดให้กับรายการใหม่ๆ และประเมินขนาดของรายการเดิมๆ ให้มีความถูกต้องยิ่งขึ้น

เนื่องจากใน Scrum ไม่มีการระบุว่าจะสำเร็จได้อย่างไร แต่เราจะใช้วิธีนี้ที่ทีมงานมาประชุมหารือกันในช่วงใกล้ๆจบ Sprint ให้ทีมและ Product Owner มาทำงานร่วมกันโดยไม่มีอะไรมารบกวน สำหรับ Sprint ที่ใช้เวลา 2 สัปดาห์ต่อรอบ 5% ของ Sprint จะใช้เวลาประมาณครึ่งวัน ในการประชุมเพื่อทำสิ่งที่เรียกว่า Product Backlog Refinement

การเรียบเรียง Product Backlog ไม่ใช่การเลือกรายการใหม่ๆ เพิ่มเข้าไปในระหว่าง Sprint แต่เป็นการเตรียมงานให้กับ Sprint รอบถัดๆ ไปอีกหนึ่งหรือสองรอบ Sprint ในอนาคต ถ้าเราทำตามนี้ การวางแผน Sprint จะเป็นเรื่องง่ายๆ เพราะ Product Owner และทีมจะวิเคราะห์งานมาแล้วอย่างดี มีการประเมินงานอย่างรัดกุม และมีความชัดเจนในแผนงาน โดยจะมีสัญญาณบางอย่างที่เตือนเราว่าการประชุมเพื่อเรียบเรียง Product Backlog สลอปไปในทางที่ไม่ดี นั่นคือในช่วงวางแผนของ Sprint ทีมจะมีคำถามสำคัญๆ แต่ไม่มีคำตอบชัดเจน สับสน และไม่เข้าใจเนื้องานอย่างทะลุปรุโปร่ง ซึ่งส่งผลกระทบไปในระหว่าง Sprint เองด้วย และนี่คือสิ่งที่เราไม่พึงปรารถนา

Ending the Sprint

หนึ่งในแก่นแท้ของ Scrum คือช่วงเวลาของ Sprint จะไม่มีการขยายออกไป จุดสิ้นสุดจะถูกกำหนดโดยไม่สนใจว่าทีมจะทำงานเสร็จตามที่ตั้งใจเอาไว้หรือไม่ โดยทั่วไปทีมจะประมาณการเกินไป ในรอบแรกของ Sprint และทำตามที่ตั้งใจเอาไว้ไม่ได้ แต่ในบางครั้งทีมก็เผื่อเวลามากเกินไป ให้คำสัญญาไว้น้อยเกินไป และสุดท้ายก็ทำงานเสร็จก่อนเวลา (ในกรณีนี้ทีมควรจะถาม Product Owner ให้เลือกงานจาก Product Backlog มาทำเพิ่มเติม) แต่เมื่อเวลาผ่านไป Sprint รอบที่ 3-4 ทีมจะพบขีดความสามารถในการส่งงาน (โดยส่วนใหญ่จะเป็นแบบนี้) และทำงานได้ใกล้เคียงกับเป้าหมายที่วางไว้มากขึ้นเรื่อยๆ หลังจากนั้นทีมจะถูกสนับสนุนให้กำหนดระยะเวลาของ Sprint เช่น 2 สัปดาห์และห้ามเปลี่ยนแปลงในภายหลัง นี่เป็นสิ่งที่ช่วยให้ทีมเรียนรู้ว่าพวกเขาสามารถทำงานให้เสร็จได้ขนาดไหน และยังช่วยให้การประเมิน และการวางแผนการส่งมอบงานในระยะยาวทำได้ดีขึ้นอีกด้วย มันช่วยให้ทีมรู้จักภาระการทำงานของตัวเอง เราเรียกสิ่งนี้ว่า “จังหวะชีวิต” ของทีมงานใน Scrum

Sprint Review

หลังจากจบ Sprint ทีมและ Product Owner จะทบทวนการทำงานของ Sprint ที่ผ่านมาในการประชุม Sprint Review ซึ่งหลายๆครั้งเราจะเข้าใจผิดว่ามันคือการ “แสดงผลงาน” ซึ่งไม่ตรงกับจุดประสงค์ที่แท้จริงของการประชุมนี้ จุดประสงค์หลักของ Scrum คือการตรวจสอบและปรับปรุง เพื่อเรียนรู้สิ่งที่เกิดขึ้นแล้วนำมาปรับปรุงพัฒนาจากความเห็นต่างๆที่ได้รับอย่างต่อเนื่อง และ Sprint Review ก็คือการกระบวนกรในการตรวจสอบและปรับปรุงผลงานนั่นเอง ในระหว่าง Sprint Review Product Owner จะได้เรียนรู้การทำงานของทีมและตัวผลงาน ซึ่งทีมก็จะได้เรียนรู้แง่มุมของ Product Owner และตลาดด้วย ดังนั้นสิ่งที่สำคัญที่สุดในการทำ Sprint Review ก็คือการพูดคุยทำความเข้าใจอย่างลึกซึ้งระหว่าง Product Owner กับทีม เพื่อที่จะเรียนรู้สถานการณ์และแลกเปลี่ยนคำแนะนำระหว่างกัน ในระหว่าง Sprint Review ทีมจะแสดงตัวอย่างผลงานที่ทำเสร็จระหว่าง Sprint ด้วย แต่ถ้ามองไปที่การแสดงผลงานเพียงอย่างเดียวแทนที่จะเป็นการพูดคุยแลกเปลี่ยน ความไม่สมดุลก็เกิดขึ้นได้

สิ่งดีๆที่คนมักจะละเลยในแนวทางการทำงานของ Scrum ก็คือความรับผิดชอบของ ScrumMaster ที่จะต้องทำให้ทุกคนในทีมเข้าใจ “นิยามของสิ่งที่ทำเสร็จ” (“Definition of Done”) ของผลิตภัณฑ์หรือชิ้นงานที่ปล่อยออกสู่ตลาด ScrumMaster จะต้องควบคุมทีมไม่ให้แสดงผลงานหรือส่งงานใน Product Backlog ที่ยังไม่ “เสร็จ” ตามนิยามของ “Definition of Done” งานไหนที่ยังไม่ “เสร็จ” จะต้องถูกนำกลับไปใส่ไว้ใน Product Backlog ซึ่งจะถูกระบุลำดับตามความสำคัญใหม่โดย Product Owner หนึ่งในแง่ความโปร่งใสของคุณภาพงาน ทีมจะต้องไม่หลอกลวงโดยการนำเสนองานที่ดูเหมือนว่ามีคุณภาพ ทำงานได้ดี แต่แอบซ่อนความยุ่งเหยิง ด้วยคุณภาพ และมีบางส่วนยังไม่ถูกทดสอบ

ผู้เข้าร่วมประชุม Sprint Review ประกอบด้วย Product Owner, สมาชิกในทีม, และ ScrumMaster รวมถึงลูกค้า, ผู้ที่มีส่วนได้ส่วนเสียกับผลงาน, ผู้เชี่ยวชาญ, ผู้บริหารระดับสูง และผู้สนใจทั่วไป การแสดงความคืบหน้าของงานในระหว่าง

Sprint Review ไม่ใช่ “การนำเสนองาน” (“presentation”) ของทีม และไม่เกี่ยวกับเอกสารนำเสนอ (slideware) แนวทางปฏิบัติของ Scrum คือไม่ควรใช้เวลาเกินกว่า 30 นาทีในการเตรียมการประชุม ถ้าใช้มากกว่านั้น อาจเป็นสัญญาณบ่งบอกว่ามีบางอย่างผิดพลาด ซึ่งความจริงแล้วเป็นเพียงการนำเสนอว่าทีมทำอะไรลงไปบ้างนั่นเอง ผู้เข้าร่วมประชุมสามารถถามคำถามและออกความเห็นได้อย่างอิสระ

Sprint Retrospective

Sprint Review คือการตรวจสอบและปรับปรุงผลิตภัณฑ์ แต่สำหรับ Sprint Retrospective ที่ทำต่อจากการ Review จะเป็นการตรวจสอบและปรับปรุงกระบวนการทำงาน บางทีมได้ข้ามการปฏิบัติขั้นนี้ไปซึ่งเป็นเรื่องที่น่าเสียดายมาก เนื่องจากมันเป็นกระบวนการหลักของ Scrum ที่จะทำให้เห็นจุดที่ควรปรับปรุงแล้วทำให้มันเกิดผล มันเป็นโอกาสดีที่ทีมจะมาคุยกันว่าอะไรใช้แล้วได้ผลดีหรือไม่ดี และตกลงกันว่าจะปรับปรุงเปลี่ยนแปลงอะไรบ้าง

ในการประชุมนี้ประกอบไปด้วยทีมและ ScrumMaster รวมไปถึง Product Owner มาร่วมได้แต่ไม่จำเป็น ในบางครั้ง ScrumMaster ก็สวมบทบาทเป็นผู้จัด Retrospective เองแต่จะให้ดีน่าจะหาคนนอกที่เป็นกลางมาเป็นคนจัด วิธีหนึ่งที่ดีคือให้ ScrumMaster ทั้งหลายผลัดกันจัดการประชุม Retrospective ข้ามทีม เพื่อให้เกิดการแลกเปลี่ยนความรู้ระหว่างทีม

สำหรับเทคนิคการทำ Sprint Retrospective ก็มีอยู่หลายวิธี หนังสือ “Agile Retrospective (Derby, Larson, 2006)” ก็มีรายการของเทคนิคต่างๆที่มีประโยชน์ วิธีง่ายๆ ที่ทำได้คือใช้กระดานที่แบ่งออกเป็นสองช่องใหญ่โดยด้านหนึ่งให้เขียนว่า “What’s Working Well” และอีกด้านหนึ่งเขียนว่า “What Could Work Better” จากนั้นทีมแต่ละคนจะเริ่มเขียนกระดาษแปะลงไปทั้งสองช่อง จากนั้นทีมจะอธิบายสิ่งที่เขียนลงในกระดานทีละข้อ ถ้ามีข้อไหนถูกพูดซ้ำก็ตักเพิ่มลงไปช่องเดิม จากนั้นทีมจะร่วมกันหาสาเหตุและช่วยกันคิดวิธีการทำงานเพื่อแก้ไขข้อผิดพลาดเหล่านั้นร่วมกัน และตกลงร่วมกันว่าผลของการทดลองใช้ครั้งนี้จะถูกตรวจทานอีกครั้งใน Sprint Retrospective ถัดไป

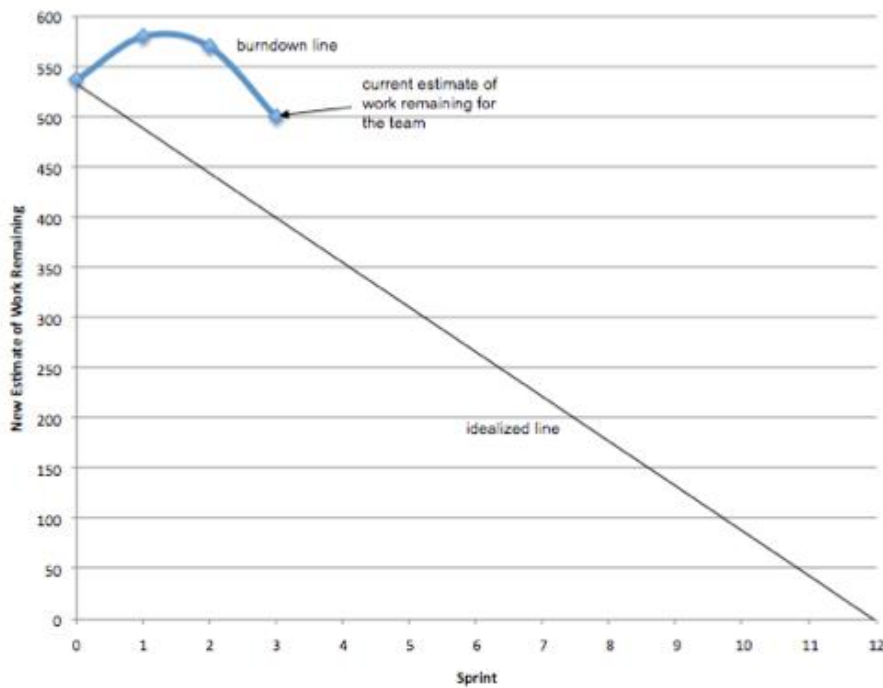
เทคนิคที่สามารถใช้ได้ตอนจบ Sprint Retrospective คือการใส่ป้ายลงไปบนหัวข้อที่มีในแต่ละช่องด้วยรหัส “C” (Caused) ถ้าเกิดจาก Scrum (ถ้าไม่ใช่ Scrum ปัญหานี้จะไม่เกิด) “E” (Exposed) ถ้ามันถูกแสดงให้เห็นโดย Scrum (ปัญหานี้จะเกิดไม่ว่าเราจะใช้หรือไม่ใช้ Scrum แต่ Scrum ทำให้เราเห็นปัญหานี้) “U” (Unrelated) ไม่เกี่ยวข้องกับ Scrum (ไม่เกี่ยวเลยเช่น อากาศไม่ดี) ซึ่งสุดท้ายตอนจบเราอาจจะเห็น “C” ในช่อง “What’s Working Well” และ E ในช่อง “What Could Work Better” ซึ่งก็ถือว่าเป็นเรื่องที่ดีถึงแม้ว่าจะมีอะไรต้องแก้ไข เพราะไม่ว่าการแก้ปัญหาใดๆก็ตาม คือทำให้มันปรากฏออกมาก่อน และ Scrum ก็เป็นตัวเร่งปฏิกิริยาที่ทรงพลังมากในด้านนี้

Updating Release Backlog & Burndown Chart

ถึงตรงนี้นางานบางอย่างอาจจะเสร็จแล้ว บางอย่างถูกเพิ่มเข้ามา บางอย่างมีการประเมินใหม่และบางอย่างถูกถอดออกจากแผนการเปิดตัวผลิตภัณฑ์ ดังนั้น Product Owner จะต้องคอยตรวจสอบการเปลี่ยนแปลงเหล่านี้ ซึ่งมีผลต่อ Release Backlog (รวมถึงมุมมองต่างๆอย่าง Product Backlog ด้วย) และยิ่งไปกว่านั้น Scrum ได้เพิ่มกราฟ Release Burndown ซึ่งแสดงให้เห็นถึงความคืบหน้าของงานไปจนถึงวันเปิดตัวผลิตภัณฑ์ด้วย ซึ่งจะคล้ายๆกับกราฟ Sprint Burndown แต่เป็นในมุมมองจากการต้องการของลูกค้ามากกว่าจะเป็นรายละเอียดของงานแต่ละชิ้น เนื่องจากคนที่ฟังเป็น Product Owner ใหม่ๆ มักไม่ค่อยรู้จักวิธีการสร้างกราฟนี้ หรือไม่รู้ว่าจะทำกราฟนี้ทำไม นี่จึงเป็นโอกาสสำหรับ ScrumMaster ที่จะเข้ามาช่วย Product Owner ในการทำสิ่งเหล่านี้ (ดูภาพที่ 8 และ 9 แสดงถึงตัวอย่างของกราฟ Release Backlog และ Release Burndown)

New Estimates of Effort Remaining at end of Sprint...										
Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5	0	0	0			
As a buyer, I want to remove a book in a shopping cart	...	2	6	2	0	0	0			
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13	13	0	0			
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20	20	20	0			
Upgrade all servers to Apache 2.2.3	...	5	5	13	13	13	13			
Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	6	2	3	3	3	3			
As a shopper, I want to create and save a wish list	...	7	7	40	40	40	40			
As a shopper, I want to add or delete items on my wish list	...	8	4	20	20	20	20			
...							
Total				537	580	570	500			

รูปภาพที่ 8: Release Backlog (a subset of the Product Backlog)



รูปภาพที่ 9: Release Burndown Chart

Starting The Next Sprint

หลังจากจบ Sprint Review แล้ว Product Owner อาจจะมีการปรับเปลี่ยน Product Backlog ตามความรู้ความเข้าใจใหม่ๆที่เกิดขึ้น หลังจากนั้นทั้ง Product Owner และทีมก็พร้อมที่จะเริ่มต้น Sprint รอบใหม่โดยไม่มีการเว้นช่วงระหว่าง Sprint ตามปกติแล้วทีมสามารถทำ Sprint Retrospective ในช่วงท้าย และเริ่มวางแผน Sprint ถัดไปในช่วงเช้าของวันรุ่งขึ้น (หรือหลังจากช่วงวันหยุด) ได้เลย

หนึ่งในหลักการพื้นฐานของการพัฒนาระบบแบบ agile คือ “การก้าวหน้าอย่างยั่งยืน” และด้วยการทำงานเฉพาะช่วงเวลาทำงานปกติในระดับที่สมเหตุสมผล ทีมก็สามารถทำงานวนรอบต่อเนื่องไปได้อย่างไม่มีที่สิ้นสุด

Release Sprint

ความสมบูรณ์แบบของ Scrum คือ เราจะได้ผลงานที่สามารถส่งมอบได้เลยในแต่ละ Sprint ซึ่งหมายความว่าไม่มีงานจำพวกการทดสอบหรืองานเอกสารที่ต้องทำอีก ทุกๆอย่างจะเสร็จสิ้นสมบูรณ์ในแต่ละ Sprint เลย ทำให้สามารถส่งมอบหรือปล่อยออกสู่ตลาดได้ทันทีหลังจบ Sprint Review นั่นหมายความว่าทุกๆรอบการทำงาน เราจะได้ชิ้นส่วนของงานที่เสร็จ

สมบูรณ์และเห็นภาพตรงกันระหว่าง Product Owner กับผู้เกี่ยวข้อง โดยสามารถรู้ความคืบหน้าได้อย่างแน่ชัดเมื่อจบการทำงานในแต่ละ Sprint

อย่างไรก็ตามหลายๆองค์กรอาจจะมีวิธีการพัฒนาระบบ เครื่องมือ และโครงสร้างพื้นฐานที่ไม่แข็งแกร่งนัก หรือเกิดสถานการณ์ที่ไม่คาดคิดมาก่อน (เช่น “เครื่องมีปัญหา”) จนทำให้ไม่สามารถทำงานได้อย่างสมบูรณ์ในอุดมคติได้ ในกรณีเช่นนี้อาจทำให้มีงานค้างที่ยังทำไม่เสร็จ เช่นการทดสอบงานที่ถูกรวมเข้าด้วยกันในขั้นตอนสุดท้าย เป็นต้น จึงมีความจำเป็นที่จะต้องมีการมีขั้นตอนดูแลงานที่ยังเหลืออยู่ ซึ่งก็คือ “Release Sprint”

หนึ่งความจำเป็นในการทำ Release Sprint เป็นสัญญาณบ่งบอกถึงความอ่อนด้อยในการทำงาน ซึ่งตามอุดมคติแล้ว ไม่มีความจำเป็นเลย Sprint จะดำเนินไปอย่างต่อเนื่องจนกระทั่งถึงจุดที่ Product Owner ตัดสินใจได้ว่าผลงานพร้อมแล้ว สำหรับการปล่อยออกสู่ตลาด ก็จะทำขั้นตอน Release Sprint เพื่อเตรียมการปล่อย ถ้าหากทีมทำตามขั้นตอนการทำงานที่ดี เช่นทำ refactoring และ continuous integration และทำการทดสอบที่มีประสิทธิภาพในระหว่างแต่ละ Sprint แล้ว ความจำเป็นในการทำผลิตภัณฑ์ให้สมบูรณ์แบบก่อนปล่อยออกสู่ตลาด (pre-released stabilization) หรืองานเก็บตกอื่นๆ ก็ควรจะเกิดขึ้นไม่มาก

Release Planning & Initial Product Backlog Refinement

คำถามที่บางครั้งมักจะถูกถามคือเราจะวางแผน Release ในระยะยาวด้วยการทำงานแบบเป็นรอบๆได้อย่างไร มีอยู่ 2 กรณีที่เราต้องพิจารณาคือ (1) ผลิตภัณฑ์ใหม่ในการเปิดตัวสู่ตลาดครั้งแรก (2) ผลิตภัณฑ์เดิมในการเปิดตัวครั้งถัดไป

ในกรณีที่ผลิตภัณฑ์ใหม่ หรือผลิตภัณฑ์เดิมแต่เพิ่งนำ Scrum มาปรับใช้ มันมีความจำเป็นที่จะต้องเริ่มทำ Product Backlog Refinement ก่อนที่จะเริ่มต้น Sprint แรก ซึ่ง Product Owner และทีมจะร่วมกันทำ Product Backlog ให้เป็นรูปเป็นร่างอย่างเหมาะสม ขบวนการนี้อาจจะใช้เวลาไม่กี่วันจนถึง 1 สัปดาห์ และจะเกี่ยวข้องกับการวางวิสัยทัศน์เชิงปฏิบัติการ, การวิเคราะห์รายละเอียดของ requirements และการประมาณการของทุกงานที่ระบุว่าจะทำในการเปิดตัวครั้งแรก

เรื่องที่น่าสนใจเกี่ยวกับ Scrum ก็คือ ในกรณีของผลิตภัณฑ์ที่เข้าที่และมี Product Backlog ที่อยู่ตัวแล้ว ก็ไม่น่าจะมีความจำเป็นพิเศษใดๆที่จะต้องวางแผนสำหรับการปล่อยสู่ตลาดครั้งต่อไป ทำไมนะหรือ? ก็เพราะว่า Product Owner และทีมควรจะต้องทำ Product Backlog Refinement ทุกๆ Sprint อยู่แล้ว (ประมาณ 5 – 10% ของแต่ละ Sprint) วิธีการพัฒนาผลิตภัณฑ์อย่างต่อเนื่องเช่นนี้จะเป็นการป้องกันไม่ให้เกิดการขัดจังหวะของขั้นตอนการทำงานต่างๆ (การเตรียมงาน-การดำเนินการ-การตัดสินใจ) ที่เรามักจะพบในกระบวนการพัฒนาตามลำดับขั้น (sequential life cycle) แบบเดิมๆ

ในการทำ Product Backlog Refinement เชิงปฏิบัติการตอนเริ่มต้นและระหว่างการทำ Product Backlog Refinement อย่างต่อเนื่องในแต่ละ Sprint ทีมและ Product Owner จะทำการวางแผนการเปิดตัว (release planning), ปรับปรุงการประเมินงาน, ลำดับความสำคัญ และเนื้อหาเมื่อพวกเขาได้เรียนรู้เพิ่มขึ้น

ในบางครั้งแผนการเปิดตัวสินค้าจะถูกกำหนดวันเอาไว้แล้ว ตัวอย่างเช่น “เราจะปล่อยผลิตภัณฑ์รุ่น 2.0 ออกสู่ตลาดที่งานแสดงสินค้าในวันที่ 10 พ.ย.” ในกรณีนี้ทีมจะทำการ Sprint ให้ได้มากที่สุด (และพัฒนาคุณลักษณะให้ได้มากที่สุด) เท่าที่เวลาเอื้ออำนวย แต่ผลิตภัณฑ์บางอย่างอาจต้องการการพัฒนาคุณลักษณะที่จะจบลงบางอย่างให้เรียบร้อยก่อนที่จะเรียกได้ว่าเป็นผลิตภัณฑ์ที่สมบูรณ์ ซึ่งจะไม่ปล่อยออกสู่ตลาดจนกว่าจะพัฒนาคุณลักษณะตามที่ต้องการเสร็จไม่ว่าจะใช้เวลาแค่ไหนก็ตาม เนื่องจาก Scrum จะมุ่งเน้นการสร้างผลงานที่สามารถนำไปใช้ได้ในแต่ละ Sprint ซึ่ง Product Owner อาจจะตัดสินใจปล่อยผลิตภัณฑ์รุ่นทดลองออกมาก่อน เพื่อให้ลูกค้าได้ลองใช้งานและคาดการณ์ถึงประโยชน์ที่จะได้รับจากผลงานที่สมบูรณ์ได้ก่อนล่วงหน้า

จากการที่เราไม่สามารถรู้ล่วงหน้าว่าอะไรจะเกิดขึ้นบ้างตั้งแต่แรก เป้าหมายจึงมุ่งไปที่การสร้างและปรับแต่งแผนงานให้สามารถกำหนดทิศทางของการเปิดตัวผลงานแบบกว้างๆ และอธิบายได้ว่าการตัดสินใจในการเลือกระหว่างข้อดีข้อเสียมีที่มาที่ไปอย่างไร (ตัวอย่างเช่น ขอบเขตงานกับตารางงาน) ลองนึกถึงสิ่งเหล่านี้ว่าเป็นแผนกลยุทธ์ที่จะนำไปสู่เป้าหมาย ซึ่งเส้นทางที่คุณเดินและการตัดสินใจที่คุณเลือกตลอดการเดินทางอาจจะถูกตัดสินใจในระหว่างเดินทาง

Product Owner ส่วนใหญ่จะเลือกแนวทางการส่งมอบผลงานแบบครั้งเดียว ตัวอย่างเช่นพวกเขาจะกำหนดวันส่งมอบและประชุมกับทีมเพื่อประเมินงานใน Release Backlog ที่จะสามารถทำให้เสร็จได้ภายในวันที่กำหนด ในสถานการณ์ที่มีความต้องการในการกำหนด "ราคา / วันส่งมอบ / ผลงานที่ต้องส่งมอบ" อย่างแน่นอนแล้ว (เช่นโครงการพัฒนาที่มีการเซ็นสัญญา) จะต้องมีการวางแผนเพื่อความไม่แน่นอนและความเปลี่ยนแปลงของตัวแปรต่างๆเหล่านี้ด้วย ในกรณีเช่นนี้ Scrum ก็ไม่มี ความแตกต่างจากการพัฒนาแบบอื่นๆ

Application or Product Focus

ไม่ว่าจะเป็นระบบหรือผลิตภัณฑ์ใดๆ ทั้งที่เปิดตัวสู่ท้องตลาดหรือใช้งานภายในองค์กร Scrum เปลี่ยนทีมงานให้ฉีกออกจากการทำงานแบบเดิมๆ ที่เน้นไปที่การวางแผนโครงการ ไปเป็นการพัฒนาและปรับปรุงผลงานอย่างต่อเนื่อง

ไม่มีอีกแล้วกับโครงการที่มีจุดเริ่มต้น ช่วงกลางของโครงการ จุดสิ้นสุดที่ชัดเจน รวมไปถึงผู้จัดการโครงการในแบบเดิมๆอีกด้วย มีเพียงแค่ Product Owner ที่ไม่เปลี่ยนคนบ่อยๆ และทีมที่สามารถจัดการตัวเองได้ในระยะยาวที่ร่วมมือกันพัฒนาผลงานเป็นรอบๆอย่างไม่มีที่สิ้นสุดจนกระทั่งผลิตภัณฑ์ หรือระบบนั้นๆ จะไม่ถูกใช้งานอีกต่อไป ทุกสิ่งทุกอย่างที่จำเป็นในการบริหารจัดการโครงการจะทำโดยทีมและ Product Owner ที่เป็นเหมือนตัวแทนลูกค้า ภายในทีมงาน หรือมาจากทีมงานที่บริหารจัดการผลิตภัณฑ์โดยตรง ทีมงานจะไม่ถูกจ้ำจี้จ้ำไชจากผู้จัดการฝ่าย IT หรือใครซักคนจากทีมบริหารจัดการโครงการ

Scrum ยังสามารถใช้ได้กับโครงการที่เกิดขึ้นเพียงครั้งเดียวแล้วสิ้นสุดกันไป (ซึ่งเกิดขึ้นบ่อยกว่าเมื่อเทียบกับการสร้างหรือการพัฒนาแบบเพื่อใช้งานในระยะยาว) ในกรณีนี้ทีมบริหารจัดการโครงการยังคงจัดการโดยทีมงาน และ Product Owner

จะเกิดอะไรขึ้นถ้างานใหม่ๆจากระบบที่มีอยู่เพียงระบบเดียวหรือมากกว่า มีไม่เพียงพอให้กับทีมงานที่รับประกันว่าจะทำงานให้กับระบบใดระบบหนึ่งเพียงอย่างเดียวในระยะยาว? ในกรณีนี้ทีมดังกล่าวจะเลือกงานจากระบบใดระบบหนึ่งเพียงระบบเดียวมาทำต่อหนึ่งรอบ Sprint จากนั้นค่อยเลือกงานจากระบบอื่นๆ มาทำใน Sprint ถัดไปสลับกันไปมา ในสถานการณ์แบบนี้ Sprint แต่ละรอบจะใช้เวลานานๆและถี่ขึ้น ตัวอย่างเช่นรอบละ 1 สัปดาห์

หรือในบางครั้งหากงานใหม่ๆมีไม่เพียงพอ ทำให้ทีมอาจต้องเลือกงานจากหลายๆระบบมาทำใน Sprint เดียวกัน อย่างไรก็ตาม อย่างไรก็ตามให้ระวังไว้ว่าการทำแบบนี้ อาจลดประสิทธิผลของทีม เนื่องจากการทำงานหลายอย่างจากระบบที่แตกต่างกันพร้อมๆกัน เพราะโดยพื้นฐานแล้ว ประสิทธิภาพตามแนวทางของ Scrum เกิดจากการที่ทีมจะต้องทุ่มเทให้กับผลิตภัณฑ์หรือระบบใด ๆ ก็ตามเพียงอย่างเดียวต่อหนึ่ง Sprint

Common Challenges

Scrum ไม่ได้เป็นเพียงหลักปฏิบัติที่นำไปใช้ได้อย่างเป็นรูปธรรม สิ่งที่สำคัญกว่านั้นคือ มันคือกรอบการทำงาน (Framework) ที่โปร่งใส และมีกลไกที่ทำให้เกิดการตรวจสอบและปรับปรุง (“inspect and adapt”) เมื่อใช้ Scrum เราจะเห็นความผิดปกติและอุปสรรคต่างๆที่ส่งผลกระทบต่อประสิทธิภาพของ Product Owner และทีม เมื่อนั้นพวกเขาจะหาทางจัดการกับปัญหานั้นๆ เช่น Product Owner ไม่รู้จักตลาด ไม่รู้ว่าสินค้าต้องทำอะไรได้บ้างและประเมินคุณค่าทางธุรกิจไม่เป็น หรือปัญหาของทีม เช่น ทีมมีความสามารถไม่พอทั้งในการประเมินขนาดของงาน และความสามารถในการพัฒนาระบบ

กรอบการทำงาน (Framework) ของ Scrum จะเผยจุดอ่อนต่างๆออกมาได้อย่างรวดเร็ว Scrum ไม่ได้แก้ปัญหาในขั้นตอนการพัฒนาสินค้า แต่มันช่วยให้เห็นความบกพร่องต่างๆ และให้กรอบการทำงานที่ช่วยส่งเสริมการค้นหาวិธีแก้ปัญหาภายในรอบสั้นๆ และปรับปรุงให้ดีขึ้นไปที่ละน้อยๆ

สมมติว่าทีมส่งงานตามที่สัญญาเอาไว้ไม่ทันใน Sprint แรก เนื่องจากยวิเคราะห์และประเมินงานไม่เก่งพอ ซึ่งทำให้ทีมรู้สึกแย่ แต่จริงๆแล้วประสบการณ์ที่ได้จะเป็นก้าวแรกที่สำคัญของทีมงาน ในการวิเคราะห์และประเมินงานให้ได้ใกล้เคียงความจริง และรอบคอบมากขึ้นตามคำสัญญาที่ให้ไว้ ด้วยรูปแบบของ Scrum ที่ช่วยเปิดเผยข้อบกพร่องต่างๆ ทำให้ทีมสามารถจัดการกับปัญหานั้นๆ ซึ่งรูปแบบที่เป็นกลไกพื้นฐานนี้จะก่อให้เกิดคุณค่าอย่างที่สุดกับทีมที่ใช้ Scrum ในการทำงาน

หนึ่งในปัญหายอดนิยมที่เจอกันเมื่อเราเริ่มใช้ Scrum คือ การไปเปลี่ยนวิธีการทำงานแบบเดิมๆ มาเป็น Scrum ยกตัวอย่าง เช่น เมื่อทีมพบปัญหาตอนส่งงานตามที่ให้สัญญาเอาไว้ใน Sprint และทีมตัดสินใจที่จะขยายเวลาของ Sprint ให้นานขึ้น

ทำให้ใน Sprint นี้ ไม่หมดเวลาลงซักที ซึ่งแน่นอนว่าทีมจะไม่ได้เรียนรู้ว่าจะทำงานให้ดีขึ้นได้อย่างไร ทั้งในด้านการประเมินขนาดของงานและการจัดการเวลา ด้วยวิธีการแบบนี้ถ้าหากไม่มีการสอนงานและสนับสนุนจาก ScrumMaster ที่มีประสบการณ์ จะทำให้ Scrum ขององค์กรนั้นๆ กลายร่างเป็นเพียงกระจกเงาที่สะท้อนจุดอ่อน และความผิดพลาดต่างๆ ของตัวเอง โดยเข้าไม่ถึงประโยชน์ที่แท้จริงของ Scrum นั่นคือ การตีแผ่ข้อดีข้อเสียของทีม ให้ทีมได้มีทางเลือกในการพัฒนาตัวเองขึ้นสู่ระดับที่สูงขึ้นไป

อีกหนึ่งความเข้าใจผิดๆ ที่พบบ่อยๆ คือการเข้าใจไปเองว่าวิธีการบางอย่างไม่มีความจำเป็นหรือถูกห้ามไว้ เพียงเพราะว่า Scrum ไม่ได้กำหนดชัดเจนว่าต้องทำมัน ยกตัวอย่างเช่น Scrum ไม่ได้กำหนดให้ Product Owner ต้องวางแผนกลยุทธ์ในระยะยาวให้กับผลิตภัณฑ์ หรือ ไม่ได้กำหนดให้วิศวกร ต้องขอคำแนะนำจากวิศวกรที่มากประสบการณ์คนอื่นๆ เพื่อแก้ปัญหาทางเทคนิคที่ซับซ้อน Scrum ปลอมเรื่องนี้ให้เป็นสิ่งที่ผู้เกี่ยวข้องแต่ละคน เป็นผู้ตัดสินใจว่าจะใครควรไม่ควรด้วยตัวเอง และโดยส่วนใหญ่แล้วตัวอย่างทั้งสองกรณี (และอีกหลายๆ อย่าง) ก็เป็นสิ่งที่ควรทำอยู่แล้ว

ในบางครั้งเราต้องระวังว่า ผู้จัดการจะเข้ามาจัดแจงวิธี Scrum ในทีมของเขาเอง ซึ่งจริงๆ แล้ว Scrum เปิดช่องและมีเครื่องมือให้ทีมบริหารตัวเองได้อยู่แล้ว ทั้งนี้การทำตามคำสั่งย่อมไม่ใช่สูตรสำเร็จใน Scrum วิธีที่ดีกว่าคือเริ่มจากปรึกษาผู้ร่วมงาน หรือหัวหน้างาน และรับการอบรมจากมืออาชีพ จากนั้นให้ทีมร่วมตัดสินใจตามสิ่งที่เรียนรู้มาภายในเวลาที่กำหนดไว้ และเมื่อหมดเวลา ให้ทีมได้ประเมินในสิ่งที่พวกเขาทำ และตัดสินใจว่าก้าวต่อไปจะเป็นอย่างไร

ข้อดี คือว่า ใน Sprint แรกซึ่งเป็นสิ่งท้าทายมากสำหรับทีม มักจะเห็นประโยชน์ของ Scrum เมื่อสิ้นสุด Sprint แล้ว ทำให้ทีม Scrum หน้าใหม่พากันบอกต่อกันว่า “ทำ Scrum มันยาก แต่เชื่อสิ มันดีกว่าสิ่งที่เราเคยทำกันมาอย่างแน่นอน!”

Appendix: Terminology

Burn Down

แนวโน้มของงานที่ยังเหลืออยู่เมื่อเทียบกับเวลาใน Sprint, Release หรือ Product จากข้อมูลดิบใน Sprint Backlog และ Product Backlog มาแสดงเป็นกราฟ โดยแสดงจำนวนงานที่เหลืออยู่ในแกนตั้งและเวลาที่มีหน่วยเป็นวันใน Sprint หรือหลายๆ Sprint ในแกนนอน

Chicken

ใครก็ตามที่สนใจในโครงการแต่ไม่ได้มีหน้าที่และความรับผิดชอบตามบทบาทใน Scrum อย่างเป็นทางการ (ทีม, Product Owner, ScrumMaster)

Daily Scrum

การประชุมสั้นๆประจำวันของแต่ละทีม ในระหว่างการประชุมสมาชิกทีมจะตรวจสอบงาน แลกเปลี่ยนข้อมูลกับสมาชิกคนอื่นๆ รายงานความคืบหน้า สถานะ และอุปสรรคต่างๆ ให้กับ ScrumMaster เพื่อช่วยขจัดสิ่งกีดขวางออกไป ทั้งนี้การประชุมเพื่อปรับเปลี่ยนงานให้เหมาะสมกับ Sprint อาจเกิดขึ้นได้หลังการประชุม Daily Scrum

Done

เสร็จตามที่ทุกฝ่ายได้ตกลงร่วมกัน สอดคล้องกับมาตรฐาน ข้อตกลง และข้อเสนอแนะต่างๆ ของหน่วยงาน เมื่องานบางอย่างถูกบันทึกว่า “done” ในการประชุม Sprint Review มันจะต้องเป็นไปตามที่นิยามไว้ในข้อนี้

Estimated Work Remaining (Sprint Backlog items)

จำนวนชั่วโมงที่ทีมประเมินว่ายังเหลืออยู่สำหรับงานนั้นๆ การประเมินจะถูกปรับปรุงในทุกๆ สิ้นวัน เมื่องานนั้นถูกหยิบมาทำจาก Sprint Backlog จำนวนชั่วโมงที่ประเมินนี้ มาจากผลรวมของจำนวนชั่วโมงทั้งหมดที่เหลืออยู่ โดยไม่สนใจจำนวนคนทำงาน

Increment

คุณสมบัติของผลิตภัณฑ์ที่พัฒนาขึ้นในระหว่าง Sprint เป็นคุณสมบัติที่มีแนวโน้มว่าใช้งานได้จริง หรือผู้มีส่วนได้ส่วนเสียของ Product Owner นำไปใช้งานได้

Increment of Potentially Shippable Product Functionality

ส่วนของงานที่เสร็จสมบูรณ์ของผลิตภัณฑ์หรือระบบโดยรวม ซึ่ง Product Owner หรือผู้มีส่วนได้ส่วนเสียอื่นๆสามารถนำไปใช้งานได้หากพวกเขาเลือกที่จะพัฒนามันจริงๆ

Sprint

การวนซ้ำหรือหนึ่งรอบการทำงานที่คล้ายกันเพื่อสร้างผลงานหรือระบบที่สมบูรณ์ขึ้นเรื่อยๆ โดยแต่ละรอบจะไม่เกินหนึ่งเดือน และโดยทั่วไปแล้วจะใช้เวลามากกว่าหนึ่งสัปดาห์ต่อรอบ ระยะเวลาจะถูกจำกัดให้คงที่ตลอดช่วงเวลาทำงานทั้งหมด และทีมงานทุกทีมที่ทำงานบนระบบหรือผลิตภัณฑ์เดียวกันจะมีระยะเวลาต่อรอบเท่ากัน

Pig

ใครบางคนที่สวมบทบาทอย่างใดอย่างหนึ่งในสามของ Scrum (ทีม, Product Owner, ScrumMaster) ผู้ที่ให้สัญญาและอำนาจที่จะเติมเต็มมัน

Product Backlog

Requirement ต่างๆที่เรียงลำดับความสำคัญไว้แล้ว พร้อมเวลาโดยประมาณที่ต้องใช้ในการทำงานให้แล้วเสร็จ การประเมินจะมีความแม่นยำมากขึ้นตามความสำคัญของรายการใน Product Backlog ซึ่งเปลี่ยนแปลงไปตามเงื่อนไขทางธุรกิจหรือเทคโนโลยีต่างๆที่เปลี่ยนไปในตอนนั้นๆได้

Product Backlog Item

Requirement หลักๆของระบบ และ requirement อื่นๆที่ไม่ใช่คุณสมบัติหลักๆ และปัญหาต่างๆ ที่ถูกเรียงลำดับตามความสำคัญทางธุรกิจ เป็นอิสระต่อกัน และถูกประเมินขนาดของความซับซ้อนแล้ว โดยความแม่นยำของค่าประเมินขึ้นกับความสำคัญและความชัดเจนของรายการนั้นๆใน Product Backlog ซึ่งรายการที่มีความสำคัญที่สุดที่ควรต้องทำใน Sprint ถัดไป ยิ่งต้องเป็นขึ้นเป็นอันและมีความแม่นยำสูง

Product Owner

ผู้รับผิดชอบในการจัดการ Product Backlog ให้มีคุณค่ากับโครงการสูงสุด Product Owner รับผิดชอบในการเป็นตัวแทนของทุกคนที่มีส่วนได้ส่วนเสียในโครงการและผลลัพธ์ที่เกิดขึ้น

Scrum

ไม่ใช่คำย่อ แต่หมายถึงกลไกในเกมรักบี้ เพื่อเริ่มเล่นบอลอีกครั้ง จากที่บอลออกจากการเล่นในสนามไป

ScrumMaster

ผู้รับผิดชอบการทำ Scrum ให้มีการดำเนินการอย่างถูกต้องและเกิดประโยชน์มากที่สุด

Sprint Backlog

รายการของงานต่างๆที่กำหนดโดยทีมงานใน Sprint เป็นรายการงานที่ต้องทำในระหว่าง Sprint โดยทุกงานจะต้องระบุผู้รับผิดชอบและประเมินขนาดของงานที่ยังเหลืออยู่ในแต่ละวันระหว่าง Sprint

Sprint Backlog Task

งานแต่ละชิ้น ซึ่งทีมหรือสมาชิกในทีมให้สัญญาว่าจะหยิบจาก Product Backlog มาทำเป็นคุณสมบัติของระบบ

Sprint Planning meeting

การประชุมภายใน 1 วัน ด้วยเวลาที่จำกัดเอาไว้ที่ 8 ชั่วโมง (สำหรับ Sprint ที่ใช้เวลา 4 สัปดาห์) ที่ต้องเกิดขึ้นในตอนต้นของทุกๆ Sprint การประชุมจะแบ่งออกเป็น 2 ส่วน และใช้เวลาไม่เกิน 4 ชั่วโมงต่อส่วน ภายใน 4 ชั่วโมงแรก Product Owner จะนำเสนอ Product Backlog ที่มีความสำคัญสูงสุดให้กับทีม ทีมและ Product Owner ต้องร่วมมือกันวิเคราะห์ว่า Product Backlog ใดบ้างที่จะหยิบมาทำจริงๆ ใน Sprint ถัดไป และทีมจะต้องให้สัญญาตามที่ตกลงกับ Product Owner เอาไว้เมื่อจบการประชุมในส่วนแรก ในส่วนที่สองอีก 4 ชั่วโมงถัดมา ทีมจะวางแผนการทำงานว่าจะออกแบบอย่างไรให้ได้ตามที่สัญญาเอาไว้ และลงรายละเอียดต่างๆ ตามแผนที่วางไว้ใน Sprint Backlog

Sprint Retrospective meeting

การประชุมภายในเวลา 3 ชั่วโมงที่จัดแจงโดย ScrumMaster ทุกคนในทีมจะคุยกันเพื่อสรุปสิ่งต่างๆที่เกิดขึ้นใน Sprint ที่เพิ่งผ่านไป และตัดสินใจว่าจะเปลี่ยนแปลงอะไรบ้างใน Sprint หน้า ให้ทีมพอใจกับงานหรือสร้างผลงานได้มากขึ้น

การประชุม Sprint Review (Sprint Review meeting)

การประชุมเมื่อจบ Sprint ภายในเวลา 4 ชั่วโมง การประชุมนี้มีขึ้นเพื่อให้ทีม Product Owner และผู้ที่มีส่วนเกี่ยวข้องได้ร่วมมือและพูดคุยกันว่ามีอะไรเกิดขึ้นบ้างใน Sprint ที่ผ่านมา โดยมากมักเริ่มจากการแสดงผลงานที่ทำเสร็จจาก Product Backlog, สนทนาเกี่ยวกับโอกาส ข้อจำกัด และประสบการณ์ที่ได้มา รวมถึงสิ่งที่ควรทำในขั้นตอนต่อไป (อาจทำให้เกิดการเปลี่ยนแปลงใน Product Backlog) เฉพาะงานที่ทำเสร็จเรียบร้อยแล้วเท่านั้นที่สามารถนำมาแสดงได้

ผู้ที่มีส่วนเกี่ยวข้อง (Stakeholder)

ผู้ที่สนใจว่าจะได้ประโยชน์อะไรบ้างจากโครงการ โดยอาจจะเป็นผู้ลงทุน ผู้ที่ต้องการใช้งาน หรือผู้ที่ได้รับผลกระทบจากผลงานนี้

ทีม (Team)

กลุ่มคนที่สามารถทำงานแทนกันได้ (cross-functional) ที่รับผิดชอบดูแลและบริหารจัดการกันเองในการพัฒนาผลงานที่ค่อยๆ สมบูรณ์ขึ้นเรื่อยๆ ในแต่ละ Sprint

การจำกัดเวลา (Time box)

ช่วงระยะเวลาสำหรับเหตุการณ์หรือการประชุมที่ไม่สามารถขยายได้ ตัวอย่างเช่นการประชุม Daily Scrum ที่จำกัดเวลาประมาณ 15 นาที และหยุดทันทีหลังจากครบ 15 นาทีโดยไม่มีการต่อรอง สำหรับการประชุมเวลาที่ใช้อาจสั้นกว่าที่กำหนด แต่สำหรับ Sprint แล้วระยะเวลาที่ใช้จะต้องเท่ากับที่กำหนดไว้เสมอ

ทีมงาน

รอบแรก

รอบที่สอง

User	Kamon Treetampinij
ScrumMaster	Varokas Panusuwan
Product Owner	Piya Pakdeechaturun
Team	Johny Sparrow, Peerawat Poombua, Sunai Sukanake