Agile modeling

เสนอ

อาจารย์ ดร.จิตติมา ศังขมณี

จัดทำโดย

1. 63113013 เอกลักษณ์ จุลจงกล

2. 63120042 ธนภัทธ์ อิสระวัฒนา

เอกสารนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมซอฟต์แวร์ รายวิชา SWE62-261 กระบวนการซอฟต์แวร์และการปรับปรุงกระบวนการ

ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2564

มหาวิทยาลัยวลัยลักษณ์

สารบัญ

Agile Modeling

What is Agile Modeling: An Introduction

Agile Modeling (AM) เป็นวิธีการปฏิบัติสำหรับการสร้างแบบจำลองและการจัดทำเอกสารของระบบซอฟต์แวร์ได้อย่างมีประสิทธิภาพ ซึ่งมีวัตถุประสงค์เพื่อเป็นการรวบรวมผลลัพธ์ หลักการ และแนวทางปฏิบัติสำหรับการสร้างแบบจำลองซอฟต์แวร์ที่สามารถนำไปใช้กับโครงการพัฒนาซอฟต์แวร์ในลักษณะที่ยืดหยุ่นกว่าวิธีการสร้างแบบจำลองแบบดั้งเดิม

นักวิศวกรรมซอฟต์แวร์จะต้องสร้างระบบที่มีความสำคัญทางธุรกิจขนาดใหญ่ ขอบเขตและความซับซ้อนของระบบดังกล่าวจะต้องมีการสร้างแบบจำลอง เพื่อ

- เพื่อให้เข้าใจส่วนประกอบทั้งหมดว่ามีอะไรที่จำเป็นต้องทำ
- เพื่อแบ่งปัญหาออกเป็นส่วนๆ ให้เหมาะสมกับผู้ที่จะทำงานนี้
- เพื่อให้สามารถประเมินคุณภาพได้ทุกๆ ขั้นตอนเมื่อระบบกำลังถูกประกอบขึ้นมา

โมเดลของ Agile (AM : Agile Modeling)

- เลือกบางหลักการมาทำ
- เป็นวิธีหนึ่งที่จะเอาหลักการของ Agile มาจัดการกับเอกสารและระบบเดิมที่มีอยู่ได้
- ใน Agile ประกอบด้วย
    1. Value (ผลลัพธ์)
    2. Principle (หลักการ)
    3. Practices (แนวทางปฏิบัติ)
- ทั้งสามอย่างนี้เป็นส่วนหนึ่งใน Model agile ที่สามารถนำมาพัฒนาซอฟต์แวร์ให้มีประสิทธิภาพและเกิดค่าใช้จ่ายน้อย
- ให้มอง Agile เป็นส่วนขยายของกระบวนการพัฒนาซอฟต์แวร์ แบบเดิมได้
    1. ให้ Agile เข้าไปกำกับ ดูว่าอันไหนสำคัญก็ทำ ไม่สำคัญก็ละ
    2. นำ Agile มาจัดลำดับความสำคัญ ดูว่ากิจกรรมไหนควรทำ ไม่ควรทำ

Agile Modeling Embraces Five Values

ค่านิยมห้าประการของ Agile Modeling (AM) คือ

1. **Communication: เน้นการสื่อสาร**

   การสร้างแบบจำลอง Agile ส่งเสริมการสื่อสารระหว่างสมาชิกในทีม นักพัฒนา และผู้มีส่วนได้ส่วนเสีย และการสื่อสารเป็นกุญแจสู่ความสำเร็จในการสร้างแบบจำลองคือการมีการสื่อสารที่มีประสิทธิภาพระหว่างผู้มีส่วนได้ส่วนเสียทั้งหมด

2. **Simplicity: เน้นความเรียบง่าย**

   มุ่งมั่นที่จะพัฒนาโซลูชันที่ง่ายที่สุดที่ตอบสนองทุกความต้องการ โมเดลช่วยให้ทั้งซอฟต์แวร์และกระบวนการพัฒนาซอฟต์แวร์ง่ายขึ้น การวาดไดอะแกรมที่แสดงแนวคิดหรือแผนและกระบวนการพัฒนาที่เกี่ยวข้องสามารถขจัดชั่วโมงการทำงานที่ไม่จำเป็นและการเขียนโค้ดด้วยตนเองได้

3. **Feedback: เน้นการตอบกลับ**

   สมาชิกในทีมที่ใช้ไดอะแกรมเพื่อสื่อสารความคิดของพวกเขาช่วยให้ผู้มีส่วนได้ส่วนเสียสามารถให้ข้อเสนอแนะได้อย่างรวดเร็ว ซึ่งจะช่วยลดเวลาการทำงานของโครงการ

4. **Courage: เน้นการกล้าตัดสินใจ**

   กล้าที่จะตัดสินใจและยึดมั่นในการตัดสินใจ ต้องการความกล้าหาญในการตัดสินใจที่ยากลำบากและการเปลี่ยนทางเลือกแม้ว่าทีมจะใช้เวลาและทรัพยากรไปมากกับงานแล้วก็ตาม

5. **Humility: เน้นความเคารพซึ่งกันและกัน**

   หมายถึงการเคารพในความคิดและข้อเสนอแนะของผู้อื่น และการยอมรับคุณค่าของการมีส่วนร่วมของผู้อื่น

What Are Agile Modeling's Core Principles?

การสร้างแบบจำลองประกอบด้วยหลักการสำคัญ 11 ประการ และ 2 หลักการเสริม

หลักการสำคัญ

1. **มีจุดประสงค์หรือเป้าหมายในการสร้างโมเดล**
   นักพัฒนาหลายคนกังวลเกี่ยวกับงานของพวกเขาเมื่อสร้างแบบจำลอง สิ่งที่พวกเขาไม่ได้ทำคือถอยออกมาและถามว่าทำไมพวกเขาถึงพัฒนาโมเดลและพวกเขากำลังพัฒนาโมเดลนี้เพื่อใคร
2. **ความเรียบง่าย**
   รักษาโมเดลให้เรียบง่ายที่สุดเท่าที่จะทำได้ และถือว่าวิธีแก้ปัญหาที่ง่ายที่สุดคือทางออกที่ดีที่สุด จำลองเฉพาะสิ่งที่คุณต้องการในวันนี้และไว้วางใจว่าคุณสามารถสร้างใหม่ได้ในภายหลังหากจำเป็น
3. **ยอมรับการเปลี่ยนแปลง**
   ยิ่งทีมเข้าใจโปรเจ็กต์มากขึ้นเท่าไร ก็ยิ่งมีโอกาสเปลี่ยนแปลงมากขึ้นเท่านั้น แทนที่จะต่อสู้กับการเปลี่ยนแปลง จงยอมรับและกล้าที่จะปรับตัวและสร้างใหม่
4. **เป้าหมายรองคือการทำให้เกิดความพยายามครั้งต่อไป**
   คนอื่นๆ อาจต้องขยายหรือปรับปรุงโครงการของคุณหลังจากที่คุณจากไป ปล่อยให้เอกสารและแบบจำลองเพียงพอเพื่อให้บุคคลหรือทีมต่อไปสามารถปรับปรุงหรือเปลี่ยนแปลงแบบจำลองได้
5. **การเปลี่ยนแปลงที่เพิ่มขึ้น**
   เป็นเรื่องยากมากที่แบบจำลองจะเสร็จสมบูรณ์ในครั้งแรกที่พัฒนา โมเดลมีการเปลี่ยนไปตามกาลเวลาเมื่อโครงการพัฒนาขึ้น ต้องทำการเปลี่ยนแปลงแบบจำลองแบบเล็กๆน้อยๆ ตามต้องการอย่างสม่ำเสมอ
6. **เพิ่มผลตอบแทนจากการลงทุนของผู้มีส่วนได้ส่วนเสียสูงสุด**
   ทีมงานต้องรับผิดชอบในการผลิตซอฟต์แวร์เพื่อเพิ่มผลตอบแทนสูงสุดให้กับลูกค้า
   ทีมงานต้องใช้ความพยายามอย่างเต็มที่ในการพัฒนาซอฟต์แวร์ที่ตรงกับความต้องการของผู้มีส่วนได้ส่วนเสีย
7. **พยายามใช้ multiple model มองหลายๆมุมมอง**
   มีหลายวิธีในการสร้างแบบจำลองเพื่อแก้ปัญหา โดยเลือกแบบจำลองที่เหมาะที่สุดตามสถานการณ์ปัจจุบัน

8. **ผลิตผลงานที่มีคุณภาพ**

   ไม่มีใครชอบงานที่ทำลวกๆ คนทำผลงานไม่ชอบเพราะเป็นสิ่งที่เขาภาคภูมิใจไม่ได้ คนที่เข้ามาในภายหลังเพื่อ refactor งาน (ด้วยเหตุผลใดก็ตาม) จะไม่ชอบมัน เพราะมันยากที่จะเข้าใจและปรับปรุง และผู้ใช้ปลายทางจะไม่ชอบผลงานเพราะมีแนวโน้มว่าจะมีปัญหา และ/หรือไม่เป็นไปตามความคาดหวัง

9. **ให้ข้อเสนอแนะอย่างรวดเร็ว**

   การทำงานอย่างใกล้ชิดกับลูกค้าของคุณ เพื่อทำความเข้าใจความต้องการ เพื่อวิเคราะห์ความต้องการเหล่านั้น หรือเพื่อพัฒนาส่วนต่อประสานกับผู้ใช้ที่ตรงกับความต้องการของพวกเขา ให้โอกาสในการได้ข้อเสนอแนะจากผู้มีส่วนได้ส่วนเสียอย่างรวดเร็ว

10. **ซอฟต์แวร์ที่ใช้งานได้คือเป้าหมายหลัก**

    เป้าหมายของการพัฒนาซอฟต์แวร์คือการผลิตซอฟต์แวร์คุณภาพสูง ซอฟต์แวร์สามารถทำงานโดยตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสียได้อย่างมีประสิทธิภาพ

11. **Travel Light**

    หมายความว่าคุณมีเอกสารเพียงพอเกี่ยวกับแบบจำลองที่คุณกำลังพัฒนา หากมีเอกสารน้อยเกินไป ทีมพัฒนาอาจหลงทาง ถ้ามีมากเกินไป ทีมพัฒนาอาจลืมไปว่าเป้าหมายหลักไม่ใช่การเขียนเอกสาร แต่เป็นการสร้างซอฟต์แวร์และแบบจำลองที่เหมาะสม

หลักการเสริม:

1. **เนื้อหามีความสำคัญมากกว่าการนำเสนอ**

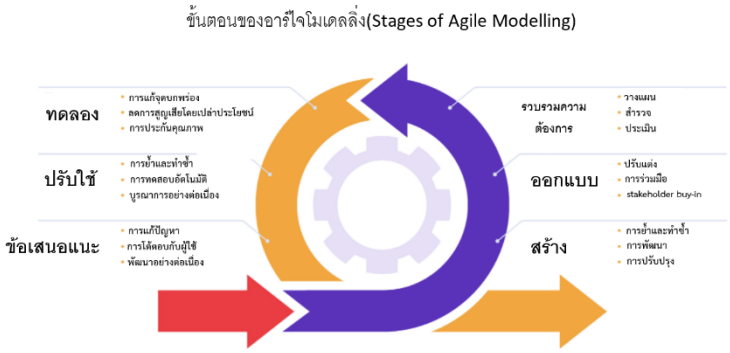
   ไม่เน้นเครื่องมือ เน้นที่เนื้อหาข้างใน

2. **การสื่อสารที่เปิดกว้างและตรงไปตรงมา**

   ผู้คนต้องเป็นอิสระและต้องรับรู้ว่าตนมีอิสระในการเสนอคำแนะนำ ซึ่งรวมถึงแนวคิดเกี่ยวกับแบบจำลองตั้งแต่หนึ่งแบบขึ้นไป อาจมีใครบางคนมีวิธีใหม่ในการเข้าถึงส่วนหนึ่งของการออกแบบ หรือมีความเข้าใจใหม่เกี่ยวกับข้อกำหนด การแจ้งข่าวร้าย เช่น ล่าช้า หรือเพียงสถานะปัจจุบันของงาน การสื่อสารอย่างเปิดเผยและตรงไปตรงมาช่วยให้ผู้คนสามารถตัดสินใจได้ดีขึ้น เนื่องจากคุณภาพของข้อมูลที่พวกเขาใช้อ้างอิงนั้นมีความแม่นยำมากขึ้น

Phases of the Agile Model

Agile แบ่งงานต่างๆ ออกเป็นส่วนย่อยๆ ซึ่งแต่ละงานมีระยะเวลาสั้นๆ (หนึ่งถึงสี่สัปดาห์) ในแบบจำลองกระบวนการโดยรวม ในกรณีของค่านิยมหลักของอาร์ไจมีหลายเวอร์ชั่นของโมเดลและเฟสของมัน นี่คือภาพประกอบ อาร์ไจโมเดลในหนึ่ง



1. **การรวบรวมความต้องการ:**

   นี่คือที่ที่คุณกำหนดข้อกำหนดของโครงการ ระยะนี้รวมถึงการอธิบายโอกาสทางธุรกิจและการวางแผนเวลาและความพยายามที่จำเป็นสำหรับโครงการ เมื่อคุณหาจำนวนข้อมูลนี้แล้ว คุณสามารถประเมินความเป็นไปได้ทางเทคนิคและเศรษฐกิจของโครงการของคุณ

2. **ออกแบบข้อกำหนด:**

   เมื่อคุณระบุพารามิเตอร์ของโครงการแล้ว ให้ทำงานร่วมกับผู้มีส่วนได้ส่วนเสียเพื่อกำหนดข้อกำหนด

3. **การก่อสร้าง/การย้ำและทำซ้ำ:**

   หลังจากที่ทีมกำหนดและออกแบบข้อกำหนดแล้ว งานจริงก็เริ่มขึ้น ทีมผลิตภัณฑ์ การออกแบบ และนักพัฒนาเริ่มทำงานในโปรเจ็กต์ที่เกี่ยวข้อง ในที่สุดก็ปรับใช้ผลิตภัณฑ์หรือบริการที่ไม่คงที่

4. **การทดสอบ:**

   ทีมประกันคุณภาพ (QA) จะตรวจสอบและประเมินประสิทธิภาพของผลิตภัณฑ์ โดยมองหาจุดบกพร่องและข้อบกพร่องอื่นๆ

5. **การปรับใช้:**

   ทีมปรับใช้ผลิตภัณฑ์ในสภาพแวดล้อมการทำงาน

6. **ข้อเสนอแนะ:**

   เมื่อผลิตภัณฑ์ออกวางจำหน่ายแล้ว ทีมงานจะได้รับคำติชมเกี่ยวกับผลิตภัณฑ์และจัดการกับปัญหาใดๆ ที่อาจเกิดขึ้น

Agile Modeling Core Practices

Agile Modeling (AM) กำหนดชุดของหลักปฏิบัติหลักและเสริม ตามหลักการของ AM แนวทางปฏิบัติบางอย่างถูกนำมาใช้จาก Extreme Programming (XP) และได้รับการบันทึกไว้อย่างดีใน Extreme Programming Explained เช่นเดียวกับหลักการของ AM แนวทางปฏิบัติจะนำเสนอโดยเน้นที่ความพยายามในการสร้างแบบจำลอง ดังนั้นเนื้อหาที่นำมาใช้จาก XP อาจถูกนำเสนอในแง่มุมที่ต่างออกไป

แนวทางปฏิบัติของ AM แบ่งออกเป็นสองรายการ หลักปฏิบัติที่คุณต้องนำมาใช้เพื่อให้สามารถอ้างว่าคุณกำลังใช้แนวทาง Agile Model Driven Development (AMDD) อย่างแท้จริง และแนวทางปฏิบัติเพิ่มเติมที่คุณควรพิจารณาปรับให้เข้ากับกระบวนการซอฟต์แวร์ของคุณเพื่อให้ตรงตาม ความต้องการที่แท้จริงของสภาพแวดล้อมของคุณ

ต่อไปนี้เป็นแนวทางปฏิบัติที่ดีที่สุด 13 ประการ

1. **การมีส่วนร่วมของผู้มีส่วนได้ส่วนเสียอย่างแข็งขัน**

   Stakeholders ต้องให้ข้อมูล ตัดสินใจเกี่ยวกับผลิตภัณฑ์ในเวลาที่เหมาะสม และมีส่วนร่วมอย่างแข็งขันในกระบวนการพัฒนาให้มากที่สุดโดยใช้เครื่องมือและเทคนิคที่ครอบคลุม การเข้าถึงผู้ใช้ที่มีอำนาจและความสามารถในการให้ข้อมูลที่เกี่ยวข้องกับระบบที่ถูกสร้างขึ้นและเพื่อการตัดสินใจที่เกี่ยวข้องและทันเวลาเกี่ยวกับข้อกำหนดและการจัดลำดับความสำคัญของข้อกำหนดดังกล่าว เพื่อให้มีผู้มีส่วนได้ส่วนเสียในโครงการ รวมถึงผู้ใช้โดยตรง ฝ่ายบริหาร ผู้บริหารระดับสูง เจ้าหน้าที่ฝ่ายปฏิบัติการ และพนักงานสนับสนุน (ฝ่ายช่วยเหลือ) – มีส่วนร่วมอย่างแข็งขันในโครงการ ซึ่งรวมถึงการตัดสินใจในการจัดหาทรัพยากรอย่างทันท่วงทีโดยผู้บริหารระดับสูง การสนับสนุนจากภาครัฐและเอกชนสำหรับโครงการโดยผู้บริหารระดับสูง การมีส่วนร่วมอย่างแข็งขันในการดำเนินงาน และเจ้าหน้าที่สนับสนุนในการพัฒนาข้อกำหนดและแบบจำลองที่เกี่ยวข้องกับพื้นที่ของตน ทีมสามารถส่งเสริมการมีส่วนร่วมของผู้มีส่วนได้ส่วนเสียอย่างแข็งขันในโครงการ หากทีมนำเทคนิคการสร้างแบบจำลองที่ครอบคลุมมาใช้

2. **โมเดลร่วมกับผู้อื่น**

   เมื่อคุณสร้างแบบจำลองอย่างมีจุดมุ่งหมาย คุณมักจะพบว่าคุณกำลังสร้างแบบจำลองเพื่อทำความเข้าใจบางสิ่ง คุณกำลังสร้างแบบจำลองเพื่อสื่อสารความคิดของคุณกับผู้อื่น หรือคุณกำลังพยายามพัฒนาวิสัยทัศน์ร่วมกันในโครงการของคุณ นี่คือกิจกรรมกลุ่ม ซึ่งคุณต้องการให้คนหลายๆ คนทำงานร่วมกันอย่างมีประสิทธิภาพ คุณมักจะพบว่าทีมพัฒนาของคุณต้องทำงานร่วมกันเพื่อสร้างชุดแบบจำลองหลักที่สำคัญต่อโครงการของคุณ ตัวอย่างเช่น ในการอุปมาการพัฒนาหรือสถาปัตยกรรมสำหรับระบบของคุณ

คุณมักจะต้องสร้างแบบจำลองร่วมกับกลุ่มคนเพื่อพัฒนาโซลูชันที่ทุกคนเห็นด้วย เช่นเดียวกับรูปแบบที่เรียบง่ายที่สุด ส่วนใหญ่วิธีที่ดีที่สุดในการทำเช่นนี้คือการพูดคุยกับบุคคลๆหนึ่งหรือหลายๆคน การสร้างแบบจำลองร่วมกับผู้อื่นคือตัวอย่างของ "Non-Solo Development" เช่นเดียวกับ pair programming

3. **ใช้ Artifact ที่เหมาะสม**

   Artifact แต่ละรายการมีการประยุกต์ใช้เฉพาะของตัวเอง สำหรับการสร้างโซลูชัน ความหมายก็คือ คุณจำเป็นต้องรู้จุดแข็งและจุดอ่อนของ Artifact แต่ละประเภท เพื่อให้คุณรู้ว่าเมื่อใดควรใช้และเมื่อใดที่จะไม่ใช้งาน โปรดทราบว่าการดำเนินการนี้อาจเป็นเรื่องยากมากเพราะคุณมีแบบจำลองหลายแบบที่สามารถใช้ได้ อันที่จริงแล้ว แบบจำลองแบบ Agile แสดงรายการแบบจำลองมากกว่า 35 ประเภท และไม่มีทางสรุปได้ชัดเจน

4. **ทำซ้ำกับ Artifact อื่น**

   เมื่อคุณกำลังทำงานเกี่ยวกับ Artifact ในการพัฒนา เช่น กรณีใช้งาน การ์ด CRC แผนภาพลำดับ หรือแม้แต่ซอร์สโค้ด และพบว่าคุณติดขัด คุณควรพิจารณาทำงานกับ Artifact อื่นในขณะนี้ Artifact แต่ละชิ้นมีจุดแข็งและจุดอ่อน Artifact แต่ละชิ้นเหมาะสำหรับงานบางประเภท เมื่อใดก็ตามที่คุณพบว่าคุณกำลังประสบปัญหาในการทำงานกับ Artifact ชิ้นหนึ่ง บางทีคุณอาจกำลังทำงานเกี่ยวกับกรณีใช้งานและพบว่าคุณกำลังดิ้นรนเพื่ออธิบายตรรกะทางธุรกิจ นั่นเป็นสัญญาณว่าคุณควรทำซ้ำกับ Artifact อื่น ตัวอย่างเช่น หากคุณกำลังทำงานกับกรณีการใช้งานที่จำเป็น คุณอาจต้องการพิจารณาเปลี่ยนโฟกัสเพื่อเริ่มทำงานกับต้นแบบ UI ที่จำเป็น โมเดล CRC , กฎธุรกิจ , system use case หรือ change case กา รวนซ้ำไปยัง Artifact อื่น คุณจะ "ไม่ติดขัด" ทันที เนื่องจากคุณกำลังดำเนินการกับ Artifact อื่นนั้นแล้วคืบหน้า นอกจากนี้ การเปลี่ยนมุมมองของคุณมักจะพบว่าคุณจัดการกับสิ่งที่ทำให้คุณติดอยู่ตั้งแต่แรกได้

5. **พิสูจน์ด้วยโค้ด**

   โมเดลเป็นสิ่งที่เป็นนามธรรม ซึ่งควรสะท้อนถึงแง่มุมของสิ่งที่คุณกำลังสร้างอย่างแม่นยำ แต่จะได้ผลไหมในการพิจารณานั้น คุณควรพิสูจน์โมเดลของคุณด้วยโค้ด เขียนโค้ดและแสดงส่วนต่อประสานผู้ใช้ที่เป็นผลลัพธ์แก่ผู้ใช้ของคุณเพื่อขอความคิดเห็น คุณได้พัฒนาไดอะแกรมลำดับ UML ที่แสดงถึงตรรกะในการใช้กฎธุรกิจที่ซับซ้อนใช่หรือไม่ เขียนรหัสการทดสอบ รหัสธุรกิจ และเรียกใช้การทดสอบเพื่อให้แน่ใจว่าคุณทำถูกต้อง อย่าลืมว่าด้วยแนวทางแบบวนซ้ำเพื่อการพัฒนาซอฟต์แวร์ ซึ่งเป็นบรรทัดฐานสำหรับโครงการส่วนใหญ่ การสร้างแบบจำลองเป็นเพียงหนึ่งในหลาย ๆ งานที่คุณจะทำ ทำแบบจำลอง เขียนโค้ด ทำการทดสอบ (เหนือสิ่งอื่นใด)

6. **ใช้เครื่องมือที่ง่ายที่สุด**

   โมเดลส่วนใหญ่สามารถวาดบนกระดานไวท์บอร์ด บนกระดาษ หรือแม้แต่หลังผ้าเช็ดปาก เมื่อใดก็ตามที่คุณต้องการบันทึกหนึ่งในไดอะแกรมเหล่านี้ คุณสามารถถ่ายภาพไดอะแกรมนั้นด้วยกล้องดิจิตอล หรือแม้แต่คัดลอกลงบนกระดาษ วิธีนี้ได้ผลเพราะไดอะแกรมส่วนใหญ่เป็นแบบใช้แล้วทิ้ง คุณค่าที่แท้จริงของพวกเขามาจากการดึงพวกเขาให้คิดผ่านปัญหา และเมื่อปัญหาได้รับการแก้ไขแล้ว แผนภาพก็ไม่ได้ให้คุณค่ามากนัก ด้วยเหตุนี้ ไวท์บอร์ดและมาร์กเกอร์จึงเป็นทางเลือกเครื่องมือสร้างแบบจำลองที่ดีที่สุดของคุณ: ใช้เครื่องมือวาดภาพเพื่อสร้างไดอะแกรมเพื่อนำเสนอต่อผู้มีส่วนได้ส่วนเสียที่สำคัญ และบางครั้งใช้เครื่องมือสร้างแบบจำลองหากพวกเขาให้คุณค่ากับความพยายามในการเขียนโปรแกรม เช่น การสร้างโค้ด คิดแบบนี้: หากคุณกำลังสร้างแบบจำลองที่เรียบง่าย ซึ่งมักจะเป็นแบบจำลองที่ใช้แล้วทิ้ง เพราะหากคุณกำลังสร้างแบบจำลองเพื่อทำความเข้าใจ คุณก็ไม่จำเป็นต้องเก็บแบบจำลองไว้อีกต่อไปเมื่อคุณเข้าใจปัญหาแล้ว อาจไม่จำเป็นต้องใช้เครื่องมือสร้างแบบจำลองที่ซับซ้อน

7. **แบบจำลองเพิ่มทีละน้อย**

   การพัฒนาที่เพิ่มขึ้นเรื่อย ๆ โดยที่คุณจัดระเบียบความพยายามที่ใหญ่ขึ้นเป็นส่วนย่อย ๆ ที่คุณปล่อยเมื่อเวลาผ่านไป โดยหวังว่าจะเพิ่มขึ้นทีละหลายสัปดาห์หรือหนึ่งหรือสองเดือน จะเพิ่มความคล่องตัวของคุณ โดยช่วยให้คุณสามารถส่งมอบซอฟต์แวร์ให้ถึงมือผู้ใช้ได้เร็วยิ่งขึ้น

8. **ข้อมูลแหล่งเดียว**

   ข้อมูลควรเก็บไว้ในที่เดียวเท่านั้น กล่าวอีกนัยหนึ่ง ไม่เพียงแต่คุณควรใช้ Artifact ที่ถูกต้องเท่านั้น คุณควรจำลองแนวคิดเพียงครั้งเดียวและครั้งเดียวเท่านั้น โดยจัดเก็บข้อมูลในที่ที่ดีที่สุด เมื่อคุณสร้างแบบจำลอง คุณควรถามคำถามเสมอว่า "ฉันจำเป็นต้องเก็บข้อมูลนี้อย่างถาวรหรือไม่", "ถ้าเป็นเช่นนั้น ที่ที่ดีที่สุดที่จะเก็บข้อมูลนี้อยู่ที่ไหน" และ "ข้อมูลนี้ถูกบันทึกไว้ในที่อื่นแล้วซึ่งฉันสามารถอ้างอิงได้หรือไม่"

9. **กรรมสิทธิ์ร่วมกัน**

   ทุกคนสามารถทำงานกับโมเดลใดก็ได้ และอันที่จริงแล้ว Artifact ใดๆ ก็ตามในโครงการหากต้องการ

10. **สร้างแบบจำลองหลายแบบในแบบคู่ขนาน**

    เนื่องจากแบบจำลองแต่ละประเภทมีจุดแข็งและจุดอ่อน จึงไม่มีแบบจำลองเดียวเพียงพอสำหรับความต้องการในการสร้างแบบจำลอง ตัวอย่างเช่น เมื่อคุณกำลังสำรวจข้อกำหนด คุณอาจต้องพัฒนากรณีการใช้งานหรือเรื่องราวของผู้ใช้ที่จำเป็น ต้นแบบ UI ที่จำเป็น และกฎเกณฑ์ทางธุรกิจบางอย่าง ร่วมกับการฝึกทำซ้ำกับ Artifact อื่น ผู้สร้างแบบจำลอง agile มักจะค้นพบว่าพวกเขาทำงานได้อย่างมีประสิทธิผลมากกว่าในหลายโมเดลพร้อมๆ กัน มากกว่าการมุ่งเน้นที่ตัวใดตัวหนึ่งในช่วงเวลาหนึ่งๆ

11. สร้างเนื้อหาที่เรียบง่าย

คุณควรรักษาเนื้อหาที่แท้จริงของแบบจำลองของคุณ -- ความต้องการของคุณ การวิเคราะห์ สถาปัตยกรรม หรือการออกแบบของคุณ ให้เรียบง่ายที่สุดเท่าที่จะทำได้ในขณะที่ยังคงตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสียในโครงการของคุณ ความหมายคือ คุณไม่ควรเพิ่มลักษณะอื่นๆ ให้กับโมเดลของคุณ เว้นแต่จะสมเหตุสมผล -- หากคุณไม่มีข้อกำหนดในการเพิ่มคุณลักษณะการตรวจสอบระบบ ก็อย่าเพิ่มคุณลักษณะเหล่านั้นลงในโมเดลของคุณ

12. แสดงให้เห็นโมเดลอย่างง่ายๆ

เมื่อคุณพิจารณาไดอะแกรมที่เป็นไปได้ที่คุณสามารถใช้ได้ (ไดอะแกรม UML ไดอะแกรมส่วนต่อประสานผู้ใช้ โมเดลข้อมูล และอื่นๆ) คุณจะตระหนักได้อย่างรวดเร็วว่าโดยส่วนใหญ่คุณต้องการเพียงชุดย่อยของคำอธิบายไดอะแกรมที่พร้อมใช้งานสำหรับคุณ

13. แสดงแบบจำลองต่อสาธารณะ

คุณควรแสดงแบบจำลองของคุณต่อสาธารณะ โดยมักจะแสดงบนสิ่งที่เรียกว่า " modeling wall" หรือ " wall of wonder" สิ่งนี้สนับสนุนการสื่อสารที่เปิดกว้างและตรงไปตรงมาในทีมของคุณ เนื่องจากโมเดลปัจจุบันทั้งหมดสามารถเข้าถึงได้อย่างรวดเร็ว เช่นเดียวกับผู้มีส่วนได้ส่วนเสียของคุณ เนื่องจากคุณไม่ได้ซ่อนอะไรจากพวก modeling wall ของคุณเป็นที่ที่คุณโพสต์แบบจำลองของคุณให้ทุกคนได้เห็น ทีมพัฒนาและผู้มีส่วนได้ส่วนเสียอื่นๆ ควรเข้าถึง modeling wall ได้ modeling wall ของคุณอาจเป็นแบบทางกายภาพ อาจเป็นไวท์บอร์ดที่กำหนดไว้สำหรับไดอะแกรมสถาปัตยกรรมของคุณ หรือสถานที่ที่คุณติดเทปงานพิมพ์ของแบบจำลองข้อมูลทางกายภาพของคุณ การสร้าง modeling wall สามารถเป็นแบบเสมือนได้ เช่น เว็บเพจภายในที่ได้รับการอัพเดตด้วยภาพที่สแกน

ข้อดีและข้อเสียของการสร้างแบบจำลอง Agile

การสร้างแบบจำลองนำมาทั้งข้อดีและพร้อมกับข้อเสียด้วย

ข้อดี

- อำนวยความสะดวกในการสื่อสารที่มีประสิทธิภาพระหว่างทีมและลูกค้า
- เพิ่มความยืดหยุ่นของโครงการ จัดการการเปลี่ยนแปลงกะทันหันได้อย่างง่ายดายทุกเวลา
- ลดเวลาในการพัฒนาโดยรวม
- เพิ่มความพึงพอใจของลูกค้าด้วยการส่งมอบผลิตภัณฑ์ที่ใช้การได้อย่างต่อเนื่องและรวดเร็ว
- มอบซอฟต์แวร์ที่ใช้งานได้บ่อยครั้งในสัปดาห์แทนที่จะเป็นเดือน

ข้อเสีย

- ความสับสนระหว่างทีมอาจเกิดขึ้นเนื่องจากไม่เน้นเอกสาร ความไม่แน่นอนนี้อาจนำไปสู่การเปลี่ยนแปลงที่ยากลำบากระหว่างขั้นตอนต่างๆ
- บางครั้งก็เป็นการยากที่จะวัดว่าต้องใช้ความพยายามมากเพียงใดในการเริ่มวงจรชีวิตการพัฒนาของซอฟต์แวร์ที่ส่งมอบให้มีขนาดใหญ่ขึ้น
- หากขั้นตอนของโครงการของ stakeholder ได้ส่วนเสียไม่ไม่ตรงกัน โครงการจะล้มเหลว
- การสร้างแบบจำลองไม่เหมาะสำหรับมือใหม่ การตัดสินใจประเภทที่เกี่ยวข้องกับ Agile นั้นต้องการผู้ที่มีประสบการณ์และทักษะด้านการพัฒนาและการเขียนโปรแกรมที่แข็งแกร่ง

การวิเคราะห์ Method ตามกรอบ Agile values และ Agile principles

The 4 values of agile

1. **Individuals and Interactions over Process and Tools (ให้ความสำคัญกับ ‘คนและการปฏิสัมพันธ์กัน’ มากกว่า ‘ขั้นตอนและเครื่องมือ’)**

   ใน Agile modeling การสร้างแบบจำลอง Agile ส่งเสริมการสื่อสารระหว่างสมาชิกในทีม นักพัฒนา และผู้มีส่วนได้ส่วนเสีย ซึ่งใน Agile modeling ระบุไว้ว่าการสื่อสารเป็นกุญแจสู่ความสำเร็จในการสร้างแบบจำลองคือการมีการสื่อสารที่มีประสิทธิภาพระหว่างผู้มีส่วนได้ส่วนเสียทั้งหมด

2. **Customer Collaboration over Contracts (ให้ความสำคัญกับ ‘การร่วมมือทำงานกับลูกค้า’ มากกว่า ‘การต่อรองให้เป็นไปตามสัญญา’)**

   ใน Agile modeling สมาชิกในทีมมีการใช้ไดอะแกรมเพื่อสื่อสารความคิดของพวกเขาเพื่อช่วยให้ผู้มีส่วนได้ส่วนเสียสามารถให้ข้อเสนอแนะได้อย่างรวดเร็ว ซึ่งจะช่วยลดเวลาการทำงานของโครงการ และการทำงานอย่างใกล้ชิดกับลูกค้า เพื่อทำความเข้าใจความต้องการ เพื่อวิเคราะห์ความต้องการเหล่านั้น เพื่อนำไปพัฒนาระบบที่ตรงกับความต้องการของพวกเขา ซึ่งสอดคล้องกับข้อสองที่เป็นการร่วมมือทำงานกับลูกค้า

3. **Working Software over Documentation (ให้ความสำคัญกับ ‘ซอฟต์แวร์ที่นำไปใช้งานได้จริง’ มากกว่า ‘เอกสารที่ครบถ้วนสมบูรณ์’)**

   ใน Agile modeling ได้มีการกำหนดหลักในการนำไปปฏิบัติไว้ในข้อที่ 10 ซึ่งได้กำหนดไว้ว่า “ซอฟต์แวร์ที่ใช้งานได้คือเป้าหมายหลัก” ได้ให้คำอธิบายไว้ว่า เป้าหมายของการพัฒนาซอฟต์แวร์คือการผลิตซอฟต์แวร์คุณภาพสูง ซอฟต์แวร์สามารถทำงานโดยตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสียได้อย่างมีประสิทธิภาพ และ ในข้อที่ 11 ได้กำหนดไว้ว่า เอกสารควรมีเท่าที่จำเป็นหรือเพียงพอกับแบบจำลองที่กำลังพัฒนา หากมีเอกสารน้อยเกินไป อาจจะทำให้ทีมพัฒนาหลงทาง ถ้ามีมากเกินไป ทีมพัฒนาอาจลืมไปว่าเป้าหมายหลักไม่ใช่การเขียนเอกสาร แต่เป็นการสร้างซอฟต์แวร์และแบบจำลองที่เหมาะสม ซึ่งสอดคล้องกับข้อสามที่เป็นการทำซอฟต์แวร์ที่นำไปใช้งานได้จริง’ มากกว่า ‘เอกสารที่ครบถ้วนสมบูรณ์’

4. **Responding to Change over Following a Plan (ตอบรับกับ ‘การเปลี่ยนแปลง’ มากกว่า ‘การทำตามแผนที่วางไว้)**

ใน Agile modeling ได้มีการกำหนด Values ไว้ว่า ต้องมีการกล้าตัดสินใจ โดยได้มีการอธิบายไว้ว่าต้องมีความกล้าหาญในการตัดสินใจที่ยากลำบากและการเปลี่ยนทางเลือกแม้ว่าทีมจะใช้เวลาและทรัพยากรไปมากกับงานแล้วก็ตาม ซึ่งหมายความว่าเมื่อมีการเปลี่ยนแปลงขึ้นในโปรเจคที่กำลังพัฒนาอยู่ถึงแม้ว่าทีมจะใช้ทรัพยากรและเวลาไปกับงานไปมากแล้วแต่เมื่อมีการเปลี่ยนแปลงเกิดขึ้นต้องกล้าหาญและตัดสินใจที่จะยอมรับการเปลี่ยนแปลงที่เกิดขึ้น และได้มีการกำหนดไว้ใน หลักการ ในข้อที่ 3 ไว้ว่า ให้ยอมรับการเปลี่ยนแปลง ซึ่งสอดคล้องกับข้อสี่ที่เป็นการตอบรับกับ ‘การเปลี่ยนแปลง’ มากกว่า ‘การทำตามแผนที่วางไว้

the 12 Principles of Agile

1. **The highest priority is to satisfy the customer (มุ่งเน้นทำให้ได้ตามที่ลูกค้าพึงพอใจ)**

ใน Agile modeling ได้มีการกำหนด หลักการ ในข้อ 6 ไว้ว่า ให้เพิ่มผลตอบแทนจากการลงทุนของผู้มีส่วนได้ส่วนเสียสูงสุด ซึ่งหมายความว่า ผู้มีส่วนได้ส่วนเสียในโครงการกำลังลงทุนทรัพยากร เวลา เงิน สิ่งอำนวยความสะดวก และอื่นๆ เพื่อพัฒนาซอฟต์แวร์ที่ตรงตามความต้องการของพวกเขา ผู้มีส่วนได้ส่วนเสียควรลงทุนทรัพยากรของตนในวิธีที่ดีที่สุดเท่าที่จะเป็นไปได้และอย่าให้ทรัพยากรถูกทีมงานทิ้ง นอกจากนี้ พวกเขาสมควรที่จะได้ข้อสรุปว่าทรัพยากรเหล่านั้นมีการลงทุนหรือไม่ลงทุนอย่างไร ซึ่งหมายความว่า ทีมงานต้องรับผิดชอบในการผลิตซอฟต์แวร์เพื่อเพิ่มผลตอบแทนสูงสุดให้กับลูกค้า ทีมงานต้องใช้ความพยายามอย่างเต็มที่ในการพัฒนาซอฟต์แวร์ที่ตรงกับความต้องการของผู้มีส่วนได้ส่วนเสีย และข้อ 8 ผลิตผลงานที่มีคุณภาพ ถ้าผลิตผลงานที่ไม่มีคุณภาพอาจทำให้ผู้ใช้ปลายทางจะไม่ชอบงานที่ต่ำกว่ามาตรฐาน เนื่องจากมีแนวโน้มว่าจะทำงานไม่ถูกต้องหรือไม่เป็นไปตามความคาดหวังของพวกเขา ดังนั้นแล้ว ใน Agile modeling จึงได้ให้ความสำคัญในการมุ่งเน้นความพึงพอใจของลูกค้าสูงสุดโดยมีการกำหนดในหลักการทำแบบจำลองถึง 2 ข้อ

2. **Welcome changing requirements (ตอบสนองต่อการเปลี่ยนแปลง)**

ใน Agile modeling ได้มีการกำหนด หลักการ ข้อ 3 ให้ยอมรับความเปลี่ยนแปลง ซึ่งหมายความว่าความต้องการนั้นพัฒนาไปตามกาลเวลา ความเข้าใจของผู้คนเกี่ยวกับข้อกำหนดนั้นมีการเปลี่ยนแปลงอยู่ตลอดเวลา ผู้มีส่วนได้ส่วนเสียสามารถที่จะเปลี่ยนแปลงความต้องการได้ เมื่อโครงการได้เริ่มทำไปสักพักแล้ว ผู้มีส่วนได้ส่วนเสียสามารถเปลี่ยนมุมมองได้ ซึ่งพวกเขาอาจเปลี่ยนเป้าหมายและเกณฑ์ความสำเร็จ

สำหรับโปรเจค ดังนั้น เมื่อสภาพแวดล้อมของโครงการนั้นมีการเปลี่ยนไปตามความคืบหน้าในการพัฒนาโปรเจคของทีม และด้วยเหตุนี้ แนวทางในการพัฒนาของทีมจะต้องสะท้อนและมีการรองรับหรือวางแผนสำหรับความเป็นจริงในข้อนี้ และทีมจะต้องมีแนวทางที่คล่องตัวและยืดหยุ่นในการจัดการการเปลี่ยนแปลง ยิ่งทีมมีความเข้าใจโปรเจ็กต์มากขึ้นเท่าไร ก็ยิ่งมีโอกาสเปลี่ยนแปลงมากขึ้นเท่านั้น แทนที่จะต่อสู้กับการเปลี่ยนแปลง จงยอมรับและกล้าที่จะปรับตัวและสร้างใหม่ Agile modeling จึงมีการกำหนดหลักการคือให้ยอมรับการเปลี่ยนแปลงและปรับตัวเข้ากับการเปลี่ยนแปลงที่เกิดขึ้น

3. **Business people & developers together daily (ดึงเอาทีมงานส่วนธุรกิจเข้ามาทำงานกับนักพัฒนาฯ ให้ได้ตลอดช่วงของโครงการ)**

ใน Agile modeling ได้มีการกำหนด ไว้ใน แนวปฏิบัติ ข้อที่ 1 การมีส่วนร่วมของผู้มีส่วนได้ส่วนเสียอย่างแข็งขัน เพื่อให้มีผู้มีส่วนได้ส่วนเสียในโครงการ ซึ่งรวมถึงผู้ใช้โดยตรง ฝ่ายบริหาร ผู้บริหารระดับสูง เจ้าหน้าที่ฝ่ายปฏิบัติการ และพนักงานสนับสนุน (ฝ่ายช่วยเหลือ) มีส่วนร่วมอย่างแข็งขันในโครงการ ซึ่งรวมถึงการตัดสินใจในการจัดหาทรัพยากรอย่างทันท่วงทีโดยผู้บริหารระดับสูง การมีส่วนร่วมอย่างแข็งขันในการดำเนินงาน และการสนับสนุนพนักงานในการพัฒนาข้อกำหนดและแบบจำลองที่เกี่ยวข้องกับพื้นที่ของตน ในการนำ Agile modeling มาใช้สามารถส่งเสริมการมีส่วนร่วมของผู้มีส่วนได้ส่วนเสียอย่างแข็งขันในโครงการได้อย่างง่ายดายหากทีมใช้เทคนิคการสร้างแบบจำลองที่ครอบคลุมมาใช้ ดังนั้นสรุปได้ว่า Agile modeling เน้นการมีส่วนร่วมของทีมงานส่วนธุรกิจเข้ามาทำงานกับนักพัฒนาฯ โดย Stakeholders ต้องให้ข้อมูล ตัดสินใจเกี่ยวกับผลิตภัณฑ์ในเวลาที่เหมาะสม และมีส่วนร่วมอย่างแข็งขันในกระบวนการพัฒนาให้มากที่สุดโดยใช้เครื่องมือและเทคนิคที่ครอบคลุม

4. **Deliver working software frequently (ส่งมอบบ่อยขึ้น เร็วขึ้น ในช่วงเวลาที่สั้น)**

ใน Agile modeling ได้มีการกำหนดแนวปฏิบัติในข้อ 7 แบบจำลองเพิ่มทีละน้อย โดยการพัฒนาที่เพิ่มขึ้นเรื่อย ๆ ทีมจะพยายามที่จะจัดระเบียบจากแบบจำลองที่ใหญ่โดยแบ่งให้เป็นส่วนย่อย ๆ ที่ทีมจะสามารถปล่อยเมื่อเวลาผ่านไปได้ โดยหวังว่าจะเพิ่มขึ้นทีละหลายสัปดาห์หรือหนึ่งหรือสองเดือน ซึ่งจะเพิ่มความคล่องตัวของทีมโดยช่วยให้ทีมพัฒนาสามารถส่งมอบซอฟต์แวร์ให้ถึงมือผู้ใช้ได้เร็วยิ่งขึ้น ซึ่งหมายความว่าการที่ทีมแบ่งเป็นส่วนย่อย ๆ ในการทำ และเพิ่มขึ้นเรื่อย ๆ และสามารถเพิ่มความคล่องตัวในการพัฒนาโปรเจค ส่งผลให้ทีมพัฒนาสามารถที่จะส่งมอมซอฟต์แวร์ให้กับลูกค้าได้อย่างรวดเร็วยิ่งขึ้น ซึ่งสอดคล้องกับการ ส่งมอบบ่อยขึ้น เร็วขึ้น ในช่วงเวลาที่สั้น

5. **Build projects around motivated individuals (การพัฒนาโปรเจคโดยรวมคนทำงานที่มีแรงจูงใจ และกระตือลือล้นในการทำงาน)**
   ตัว agile modeling มีเน้นไปที่การทำความเข้าใจเกี่ยวกับตัวโปรเจคและยังรวมไปถึงการกระจายหน้าที่เหมาะสมกับความสามารถของบุคคลนั้น ซึ่งเมื่อบุคคลนั้นทำงานที่ตนถนัดย่อมทำให้เกิดความตั้งใจที่จะทำงานนั้นให้เสร็จลุล่วงอย่างมีประสิทธิภาพ

6. **face-to-face conversation (วิธีการที่มีประสิทธิภาพและประสิทธิผลนั้น ให้มีการแลกเปลี่ยนข้อมูล ระหว่างทีมพัฒนาด้วยกันเอง หรือกับลูกค้า ด้วยการสื่อสารแบบตัวต่อตัวเห็นหน้า)**
   ตัว agile modeling เน้นไปที่การอำนวยความสะดวกในการสื่อสารที่มีประสิทธิภาพระหว่างทีมและลูกค้า อีกทั้งยังเน้นไปที่การเพิ่มความพึงพอใจของลูกค้าด้วยการส่งมอบผลิตภัณฑ์ที่ใช้การได้อย่างต่อเนื่องและรวดเร็ว และการติดต่อสื่อสารกันในทีมทำให้การทำงานในตัวโปรเจคเกิดก้าวหน้าไปได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

7. **Working software is the primary measure of progress (การสร้างหรือพัฒนาซอฟต์แวร์ (หรือผลิตภัณฑ์/บริการ/กระบวนการทำงาน)ที่มีคุณค่า เป็นตัววัดความก้าวหน้าของการทำงาน)**
   ตัว agile modeling เน้นการติดต่อสื่อสารกันในทีมทำให้การทำงานในตัวโปรเจคเกิดก้าวหน้าไปได้อย่างรวดเร็ว ซึ่งการติดต่อสื่อสารกันช่วยลดการทำงานที่ทับซ้อนกัน ช่วยให้การทำงานพัฒนาไปได้ไวยิ่งขึ้น

8. **Promote sustainable development (จะเป็นการส่งเสริมการพัฒนาที่ยั่งยืน ดังนั้นผู้สนับสนุน นักพัฒนา และกลุ่มผู้ใช้ จะต้องทำงานด้วยระดับความเร็วที่สม่ำเสมอ ไม่ช้าเกินไป หรือ ไปเร่งงานช่วงท้ายของการพัฒนา)**
   ตัว agile modeling ถูกออกแบบมาเพื่อเพิ่มความยืดหยุ่นของโครงการ จัดการการเปลี่ยนแปลงกะทันหันได้อย่างง่ายดายทุกเวลา ทำให้การพัฒนาสามารถทำได้อย่างรวดเร็ว อีกทั้งการติดต่อสื่อสารกันในทีมยังทำให้การแบ่งหน้าที่ในการทำงานได้ง่าย ทำให้การพัฒนาสามารถเกิดขึ้นได้อย่างต่อเนื่อง

9. **Continuous attention to technical excellence and good design (การพัฒนาความรู้เชิงเทคนิคให้ดีเยี่ยมอย่างต่อเนื่อง และการออกแบบที่ดี เมื่อมีทั้งสองสิ่งนี้จะทำให้เกิดการคล่องตัวในการพัฒนาผลิตภัณฑ์ได้อย่างรวดเร็ว)**

   ตัว agile modeling เน้นไปที่การอำนวยความสะดวกในการสื่อสารที่มีประสิทธิภาพระหว่างทีมและลูกค้า อีกทั้งยังเน้นไปที่การเพิ่มความพึงพอใจของลูกค้าด้วยการส่งมอบผลิตภัณฑ์ที่ใช้การได้อย่างต่อเนื่องและรวดเร็ว ทำให้การออกแบบพัฒนาดีและตรงกับความต้องการของลูกค้า

10. **Simplicity (พยายามทำงาน หรือพัฒนาผลิตภัณฑ์ให้เรียบร้อย ไม่ซับซ้อน แต่ยังมีประสิทธิภาพการใช้งานได้สูงสุดดีกว่า)**

    ตัว agile modeling เน้นไปที่การติดต่อสื่อสารทั้งในทีมและกับลูกค้า ทำให้การพัฒนาผลิตภัณฑ์ให้เรียบร้อย ไม่ซับซ้อน สามารถทำได้ง่ายและมีประสิทธิภาพ

11. **self-organizing teams (คนในทีมมีการรับผิดชอบในงานของตัวเองอย่างดีที่สุด สามารถคิดและหาวิธีการทำงานที่มีประสิทธิภาพ และแก้ปัญหาได้โดยไม่ต้องรอให้ใครมาบริหารจัดการ นอกจากจัดการตัวเองให้พร้อม และพัฒนาตัวเองอยู่เสมอ)**

    ตัว agile modeling เน้นไปที่การทำความเข้าใจเป้าหมายของโครงการซึ่งการเข้าใจในเป้าหมายย่อมส่งผลให้การทำงานเกิดประสิทธิภาพที่สูงขึ้น และการติดต่อสื่อสารกันทำให้การแบ่งงานที่เหมาะสมกับบุคลากรได้อย่างมีประสิทธิภาพ ส่งผลให้เกิดการพัฒาและการแก้ปัญหาที่ตรงจุด

12. **the team reflects on how to become more effective (ในช่วงเวลาปกติ ทีมต้องมีการแลกเปลี่ยน หรือให้ข้อมูลป้อนกลับทั้งให้กับตัวเอง และทีมงาน ซึ่งจะผ่านการให้ feedback กันในทีมก็ได้ เพื่อปรับเปลี่ยนพฤติกรรม แนวทางการทำงาน เพื่อให้เกิดการพัฒนาขึ้นอย่างต่อเนื่องและสม่ำเสมอ)**

    อย่างที่ได้กล่าวไปตัวagile modeling เน้นไปที่การติดต่อสื่อสารกันในทีม ทำให้การแลกเปลี่ยน หรข้อมูลป้อนกัน การตอบรับหรือส่งความเห็น feedback กันในทีมได้ เกิดการปรับเปลี่ยนพฤติกรรม แนวทางการทำงาน ทำให้เกิดการพัฒนาขึ้นอย่างต่อเนื่องและสม่ำเสมอ

อ้างอิง

Alden, B. (n.d.). Agile Modeling. The Agile Methodologies. Retrieved February 23, 2022, from https://www.umsl.edu/~sauterv/analysis/Fall2013Papers/Buric/agile-methodologies/agile-modeling.html

Ambler, S. W. (n.d.). The Principles of Agile Modeling (AM). Agilemodeling.Com. Retrieved February 23, 2022, from http://agilemodeling.com/principles.htm

Simplilearn. (2021, August 28). Agile Modeling: Core Principles, Advantages, And Best Practices Explained. Retrieved September 23, 2022, from https://www.simplilearn.com/agile-modelling-article