บทที่ 2
สรุปสาระสำคัญจากเอกสารที่เกี่ยวข้อง

ในการการพัฒนาระบบติดตามงานวิจัยโดยใช้สถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง ผู้วิจัยได้ศึกษาเกี่ยวกับปัญหาของระบบงานเดิม(as is) เพื่อที่นำไปออกแบบระบบใหม่ที่เป็นระบบแบบ (to be) รวมไปถึงเทคโนโลยีที่เกี่ยวข้องกับการพัฒนาระบบ โดยเฉพาะสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง ที่จะช่วยให้การพัฒนาทำได้รวดเร็วและปรับแก้หรือนำไปใช้ใหม่ได้ง่าย อีกทั้งเมื่อได้พัฒนาไปแล้วสามารถนำเอาต้นแบบไปประยุกต์ใช้ได้อีก โดยแบ่งออกเป็นหัวข้อต่างๆ ดังนี้

2.1 สถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง

2.2วัฏจักรการพัฒนาของเอ็มดีเอ

2.3 เครื่องมือของเอ็มดีเอ

2.4 การพัฒนาซอฟต์แวร์โดยวิธีเอจายด์

2.5 การพัฒนาซอฟต์แวร์ด้วยวิธีสกรัม

2.6ระบบติดตามงานวิจัย

2.1 สถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง

สถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง (MDA: Model-Driven Architecture หรือ เอ็มดีเอ) เป็นรูปแบบและวิธีในการพัฒนาซอฟต์แวร์ ซึ่งถูกพัฒนาจาก โอเอ็มจี (OMG: The Object Management Group) กระบวนการพัฒนาซอฟต์แวร์ของเอ็มดีเอ ประกอบไปด้วย การวิเคราะห์ การออกแบบ การเขียนโปรแกรม และการทดสอบ ซึ่งการพัฒนาและออกแบบของเอ็มดีเอใช้สัญลักษณ์พื้นฐานของยูเอ็มแอล (UML) แต่สัญลักษณ์ที่ใช้ยังล้าหลังจากมาตรฐานยูเอ็มแอลที่ใช้กันอยู่ในปัจจุบัน เนื่องจากต้องการให้โมเดลที่ใช้สามารถครอบคลุมทุกสิ่งในกระบวนการพัฒนาด้วยเอ็มดีเอ

เอ็มดีเอประกอบไปด้วยโมเดลสามแบบและจะใช้ขั้นตอนการพัฒนาจากบนลงล่าง (Top-Down Development) โดยเริ่มจากการสร้างโมเดลแบบจำลองลงไปจนถึงขั้นตอนการเขียนโปรแกรมและทดสอบ การพัฒนาเริ่มจากการวิเคราะห์ จากนั้นเป็นการทำงานกับโมเดลที่เป็นองค์ประกอบของธุรกิจหรือเรียกว่า โดเมนโมเดล โดยในขั้นตอนการทำงานจะมีการตรวจสอบอย่างต่อเนื่อง เพื่อยืนยันความถูกต้องของโดเมนโมเดลของธุรกิจ หลังจากได้โดเมนโมเดลที่ถูกต้องแล้ว จะทำการเปลี่ยนโดเมนโมเดลที่ได้ เป็นโมเดลที่จำเพาะเจาะจงกับเทคโนโลยีแบบใดแบบหนึ่ง

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

หรือเรียกว่าเทคโนโลยีโมเดล และเมื่อได้เทคโนโลยีโมเดลแล้ว ก็จะทำการออกแบบและพัฒนาซอฟต์แวร์ในส่วนของเทคโนโลยีโมเดล ในขั้นตอนนี้จะเป็นการจัดการกับฟังก์ชันการทำงานและการจัดการกับส่วนติดต่อกับผู้ใช้ หลังจากทำการออกแบบและพัฒนาส่วนเทคโนโลยีโมเดลเสร็จแล้ว ขั้นตอนต่อไปจะใช้เครื่องมือเปลี่ยนเทคโนโลยีโมเดลไปเป็นโค้ด ซึ่งโค้ดที่ได้ นักพัฒนาสามารถเขียนโค้ดโปรแกรม ทำให้โปรแกรมทำงานตามความต้องการของระบบ ในขั้นตอนการแปลงโมเดล เครื่องมือจะทำการแปลงให้โดยอัตโนมัติ การทำงานของเครื่องมือแบบนี้ ทำให้กระบวนการพัฒนาด้วยเอ็มดีเอสะดวก และง่ายกว่าการเขียนโปรแกรมเองด้วยมือ ซึ่งจะลดระยะเวลาของกระบวนการพัฒนาซอฟต์แวร์และทำให้ซอฟต์แวร์ที่ได้มีคุณภาพยิ่งขึ้น

กระบวนการพัฒนาซอฟต์แวร์ด้วยเอ็มดีเอ ถูกขับเคลื่อนโดยกิจกรรมของการสร้างแบบจำลอง (หรือเรียกว่า โมเดล) เพื่อสร้างซอฟต์แวร์ ของธุรกิจในรูปแบบต่างๆ แนวคิดของเอ็มดีเอนั้นจะมองไปยังโครงสร้างของธุรกิจและนำเอากระบวนการและองค์ประกอบต่างๆ ของธุรกิจนั้นออกมาในเชิงนามธรรม (Abstract) แล้วนำมาสร้างเป็นโมเดล โดยปกติแล้วธรรมชาติของแต่ละธุรกิจจะมีกระบวนการและองค์ประกอบคล้ายๆกัน หรือเหมือนกัน ซึ่งเอ็มดีเอจะสร้างโมเดลจากองค์ประกอบพื้นฐานเหล่านั้นและให้ครอบคุลมถึงกระบวนการและองค์ประกอบต่างๆ ที่มีอยู่ในแต่ละธุรกิจ เมื่อต้องการสร้างซอฟต์แวร์ให้ธุรกิจ ก็นำเอาองค์ประกอบโมเดลเหล่านั้นมาปรับปรุงให้สามารถใช้กับธุรกิจนั้นๆ ที่จะพัฒนาแล้วสร้างซอฟต์แวร์จากโมเดลเหล่านั้น ทำให้การพัฒนาซอฟต์แวร์เป็นไปอย่างรวดเร็วและถูกต้องเมื่อเทียบกับการพัฒนาแบบเดิมที่จะต้องเก็บความต้องการของลูกค้าแล้วมาเขียนโมเดลใหม่ทุกครั้ง

เครื่องมือในการพัฒนาซอฟต์แวร์ที่ใช้กระบวนการของเอ็มดีเอมีหลากหลาย ซึ่งมีการทำงานที่ซับซ้อน ในแต่ละกระบวนการของเอ็มดีเอ นักพัฒนาจำเป็นต้องเลือกให้เหมาะกับลักษณะของแต่ธุรกิจ เนื่องจากแต่ละกระบวนการของเอ็มดีเอ เครื่องมือจะทำงานให้เกือบทั้งหมด เช่น นำไปใช้กับการเก็บรวบรวมความต้องการของระบบ การสร้างโดเมนโมเดล และการเขียนโค้ดโปรแกรม รวมถึงการสร้างส่วนประกอบอื่นๆ ด้วย เครื่องมือของเอ็มดีเอแต่ละแบบมีความสามารถแตกต่างกันไป สามารถแบ่งชนิดของเครื่องมือเอ็มดีเอเป็นแบบเชิงพาณิชย์หรือคอมเมอเชียล และแบบเปิดเผยโค้ดหรือโอเพนซอส ซึ่งเครื่องมือแบบโอเพนซอสจะไม่มีค่าใช้จ่ายสามารถนำมาใช้ได้โดยอยู่ภายใต้เงื่อนไขของแต่ละผู้ผลิต ส่วนเครื่องมือแบบคอมเมอเชียลจะมีค่าใช้จ่ายในการซื้อเครื่องมือมาใช้ โดยแบบคอมเมอเชียลจะมีประสิทธิภาพสูงและมีเครื่องมือย่อยช่วยมากมาย ส่วนในแบบโอเพนซอสจะมีความสามารถแบบกลางๆ เหมาะกับนักพัฒนาหรือธุรกิจใหญ่ ซึ่งจะต้องใช้เวลาและความพยายามสูงเพื่อศึกษาการใช้งานของเครื่องมือให้เข้าใจ เครื่องมือของเอ็มดีเอในปัจจุบันที่ใช้กันส่วนใหญ่ได้แก่

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- ArcStylerเป็นแบบคอมเมอเชียลจากค่าย Interractive Objects ซึ่งติดตั้ง Magic draw UML-Tool มาด้วย
- MOFScriptเป็นแบบโอเพนซอสเป็นเครื่องมือที่ทำงานลักษณะแผลข้อความคำสั่ง (Script command) โดยใช้กับ Eclipse plug-in
- OpenMDXเป็นแบบโอเพนซอส สามารถเชื่อมต่อกับอีกหลายเครื่องมือและเป็นแบบสร้างโค้ดได้อย่างเดียว ไม่สามารถสร้างโมเดลจากโค้ดได้ สามารถใช้กับเทคโนโลยีเจทูอีอี (J2EE) หรือ ดอทเน็ต (.Net)
- AndroMDAเป็นแบบโอเพนซอสสามารถใช้กับจาวาเทคโนโลยี สามารถสร้างโค๊ดจาก UML/XML
- OptimalJเป็นแบบคอมเมอเชียลจากค่าย Compuware Corporation ใช้สัญลักษณ์ยูเอ็มแอล(UML: Unified Model Language) สำหรับการสร้างพีไอเอ็ม (PIM: Platform Independent Model) และพีเอสเอ็ม (PSM: Platform Specific Model) มีเครื่องมือสำหรับวิเคราะห์และสร้างยูเอ็มแอล์ไดอะแกรม
- Enterprise Core Object (ECO)เป็นแบบคอมเมอเชียลจากค่าย Capable Objects รองรับการพัฒนาบนเอเอสพี ดอทเน็ตเฟรมเวิคASP .Net Frameworkใช้สัญลักษณ์ยูเอ็มแอล (UML: Unified Model Language) สำหรับการสร้างพีไอเอ็ม (PIM: Platform Independent Model) และพีเอสเอ็ม (PSM: Platform Specific Model)

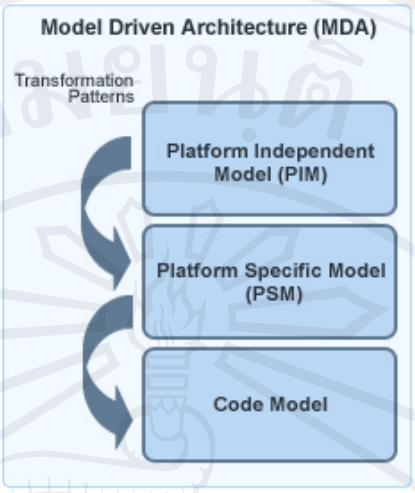
2.2วัฏจักรการพัฒนาของเอ็มดีเอ(MDA Development Life Cycle)

การพัฒนาซอฟต์แวร์ด้วนเอ็มดีเอนั้นแยกระดับการออกแบบโมเดลเป็น 3 ระดับ แสดงให้เห็นขั้นตอนการเปลี่ยนรูปแบบของโมเดลในแต่ละระดับ โดยการพัฒนาในแต่ละระดับของเอ็มดีเอได้แก่

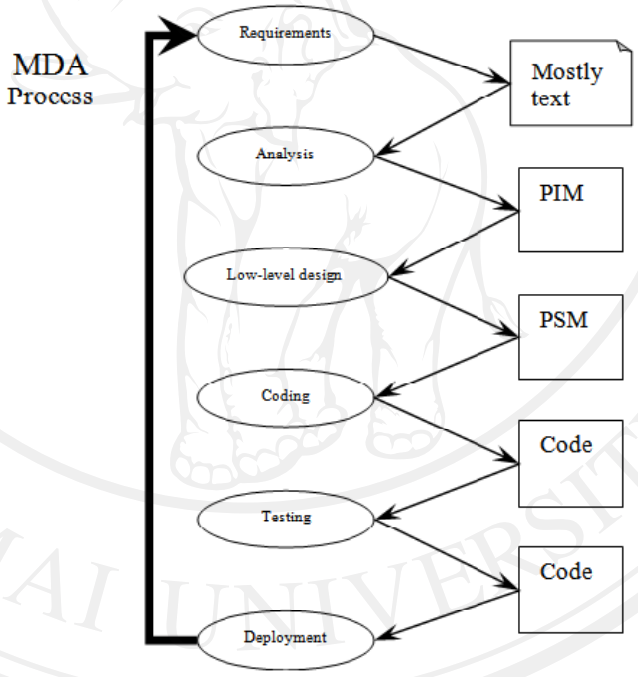
- พีไอเอ็ม(PIM: Platform Independent Model)
- พีเอสเอ็ม(PSM: Platform Specific Model)
- โค้ด (Code)

วัฏจักรการพัฒนาของเอ็มดีเอจะอยู่ภายใต้กระบวนการ 3 ระดับนี้ โดยเริ่มตั้งแต่การพัฒนาจากพีไอเอ็ม พีเอสเอ็มและการเขียนโค้ดโปรแกรม ดังรูปต่อไปนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved



รูปที่ 2.1 แสดง MDA Implemented



รูปที่ 2.2แสดง MDA Software Development Lifecycles

1) พีไอเอ็ม(PIM: Platform Independent Model)

พีไอเอ็มเป็นระดับบนสุดของวัฏจักรการพัฒนาซอฟต์แวร์ด้วยเอ็มดีเอ โดยแสดงถึงองค์ประกอบของธุรกิจในลักษณะนามธรรมที่ไม่ยึดติดกับเทคโนโลยีใดๆ พีไอเอ็มจะสร้างโมเดลจากองค์ประกอบต่างๆ ที่มีอยู่ในธุรกิจ โมเดลที่ได้จะรองรับความต้องการต่างๆ ของธุรกิจ ซึ่งในขั้นตอนนี้ เป็นขั้นตอนที่มีความสำคัญมาก โมเดลที่ได้จากขั้นตอนนี้สามารถนำไปใช้ได้กับอีกหลายธุรกิจได้หากออกแบบมาได้ดี เนื่องจากธรรมชาติของแต่ละธุรกิจจะมีบางส่วนที่คล้ายคลึงกันหรือเหมือนกันสามารถนำมาใช้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ร่วมกันได้ โมเดลที่ได้จะไม่ขึ้นกับเทคโนโลยีใดๆ เช่นเทคโนโลยีของฐานข้อมูล (MSSQL, MySQL, Oracle) ภาษาโปรแกรมที่ใช้ (ASP .NET, JSP, Servlet) และโปรแกรมประยุกต์ย่อยต่างๆ ที่นำมาใช้ในการพัฒนา พีไอเอ็มประกอบไปด้วยสัญลักษณ์และส่วนต่างๆ ของยูเอ็มแอล เช่น ยูสเคส คลาสไดอแกรม และอื่นๆ

2) พีเอสเอ็ม(PSM: Platform Specific Model)

ในขั้นตอนพีเอสเอ็มโดเมนโมเดลจะถูกเปลี่ยนรูปเป็นเทคโนโลยีโมเดล โดยจะเลือกเทคโนโลยีที่จะใช้ก่อน เช่น ดอทเน็ทเทคโนโยลีจากไมโครซอฟต์ (.Net) หรือ จาวาเทคโนโลยี (JSP, Servlet) โดยพีไอเอ็มหนึ่งๆ จะเป็นต้นแบบที่สามารถเปลี่ยนไปใช้ได้กับเทคโนโลยีอื่นได้อีกหลายแบบ ซึ่งนักพัฒนาตจะต้องมีความรู้ ความสามารถกับเทคโนโลยีที่แตกต่างกันไป เมื่อถึงขั้นตอนนี้ นักพัฒนาจะเลือกเทคโนโลยีที่จำนำมาใช้สำหรับการพัฒนา ในขั้นตอนการแปลงนี้จะใช้เครื่องมือ สำหรับการเปลี่ยนรูปจากพีไอเอ็มไปเป็นพีเอสเอ็ม ซึ่งเครื่องมือจะนำเอาต้นแบบโครงสร้างมาตรฐานที่มีอยู่แปลงพีไอเอ็มเป็นพีเอสเอ็ม ซึ่งจะได้โมเดลและส่วนประกอบต่างๆ จากเทคโนโลยีที่เลือกใช้ ในแต่ละเทคโนโลยีโมเดล ยังมีแบบโครงสร้างซอฟต์แวร์รูปแบบต่างๆให้เลือกใช้ ซึ่งการเลือกแบบโครงสร้างซอฟต์แวร์จะต้องเลือกให้เหมาะสมกับขนาดของโครงการ ซึ่งโครงการที่มีขนาดไม่ใหญ่เลือกแบบโครงสร้างซอฟต์แวร์ที่เล็กไม่ซับซ้อนทำให้การพัฒนาง่ายและเร็วกว่าการเลือกแบบโครงสร้างซอฟต์แวร์ที่มีขนาดใหญ่และซับซ้อน

3) โค้ด(Code)

ขั้นตอนสุดท้ายคือการเขียนโค้ดโปรแกรม เครื่องมือจะแปลงโค้ดจากพีเอสเอ็มซึ่งจะได้โครงสร้างโค้ดของโปรcdi,ตามภาษาและเทคโนโลยีที่เลือกใช้ โครงสร้างโค้ดของโปรแกรมที่ได้จะประกอบไปด้วยคลาส คุณสมบัติต่างๆ และเมทอดหรือฟังก์ชันการทำงาน (Classes, Attributes, Methods) และไฟล์คุณสมบัติต่างๆ ที่จำเป็นสำหรับการติดตั้งซอฟต์แวร์ โค้ดที่ได้ โปรแกรมเมอร์สามารถเขียนคำสั่งลงไปตามส่วนต่างๆ ซึ่งจะเขียนเพิ่มได้เฉพาะส่วนที่เรียกว่าส่วนอิสระ (Free Block) เท่านั้น โค้ดส่วนที่ไม่สามารถเขียนได้เรียกว่าส่วนป้องกัน (Guarded Block) เนื่องจากเป็นโค้ดจากการสร้างของเครื่องมือ โค้ดส่วนนี้มาจากขั้นตอนการออกแบบพีไอเอ็มและพีเอสเอ็มเท่านั้น การสร้างโค้ดของเครื่องมือส่วนใหญ่เมื่อสร้างเสร็จแล้วจะสามารถทำงานในเบื้องต้นได้เลย ทำให้โปรแกรมเมอร์ สามารถทดสอบและแก้ไขส่วนต่างๆของโค้ด ให้ตรงกับความต้องการต่อไป

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.3เครื่องมือของเอ็มดีเอ (MDA Tools)

กระบวนการของเอ็มดีเอ มีความแตกต่างจากธรรมชาติของกระบวนการพัฒนาซอฟต์แวร์ในแบบเดิม ซึ่งแต่เดิมการแปลงจากโมเดลไปยังอีกโมเดล หรือการแปลงจากโมเดลไปเป็นโค้ดนั้นจะถูกกระทำด้วยมือโดยนักพัฒนา ส่วนเครื่องมือที่ช่วยแปลงเป็นโมเดลไปเป็นโค้ดสามารถช่วยได้เป็นบางส่วน ทำให้โค้ดส่วนใหญ่ยังถูกสร้างมาจากนักพัฒนา

ปัจจุบันการพัฒนาซอฟต์แวร์ด้วยเอ็มดีเอรวดเร็วขึ้น เนื่องจากโมเดลหนึ่งถูกเปลี่ยนไปเป็นโมเดลอีกระดับหนึ่งได้โดยการใช้เครื่องมือ ซึ่งเรียกว่าขั้นตอนการแปลงแบบอัตโนมัติ (Automation of the Transformation Steps) การแปลงของเอ็มดีเอจะใช้เครื่องมือเสมอเพื่อแปลงจากโมเดลไปเป็นโมเดลหรือจากโมเดลไปเป็นโค้ดดังแสดงในรูปที่ 2.3 มีหลายเครื่องมือที่สามารถแปลงจากพีเอสเอ็มไปยังโค้ด ซึ่งเป็นเรื่องปกติเพราะตัวพีเอสเอ็มเองจะมีองค์ประกอบและโครงสร้างตามภาษาที่ต้องการพัฒนาอยู่แล้ว และมีความใกล้เคียงกับโค้ด ไม่ได้เป็นสิ่งใหม่หากในเอ็มดีเอได้นำเสนอการแปลงจากพีไอเอ็มไปยังพีเอสเอ็ม ซึ่งจะเป็นไปอย่างอัตโนมัติ แสดงให้เห็นถึงประสิทธิภาพของเอ็มดีเอได้อย่างชัดเจน หากเทียบกับจำนวนของเวลาและความพยายามเพื่อสร้างฐานข้อมูลจากการออกแบบระดับสูง รวมถึงจำนวนงานที่ใช้ไปสำหรับการออกแบบของโปรแกรม การแปลงของเอ็มดีเอ จะช่วยประหยัดเวลาในขั้นตอนนี้ได้เป็นอย่างมาก

ในช่วงแรกที่เอ็มดีเอยังใหม่อยู่นั้น เครื่องมือสำหรับการแปลงของเอ็มดีเอยังไม่สามารถแปลงจากโมเดลไปโมเดลหรือจากโมเดลไปยังโค้ดได้อย่างสมบูรณ์ บางครั้งยังต้องใช้นักพัฒนาช่วยเขียนเพิ่มลงไปด้วย แต่ในปัจจุบันนี้เครื่องมือของเอ็มดีเอพัฒนาไปอย่างมาก โดยสามารถสั่งให้โปรแกรมทำงานจากโมเดลได้เลยอย่างง่ายๆ ซึ่งสามารถช่วยให้นักพัฒนานำเสนอและได้ผลตอบรับจากลูกค้าได้อย่างรวดเร็ว ซึ่งจะทำให้นำมาปรับปรุงแก้ไขได้อย่างรวดเร็ว

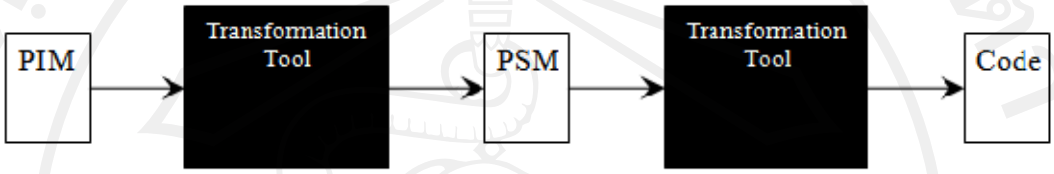
1) ขั้นตอนการแปลงแบบอัตโนมัติ(Automation of the Transformation Steps)

ขั้นตอนการแปลงของเครื่องมือเริ่มจากการแปลงพีไอเอ็มที่ได้จากนักออกแบบไปเป็นพีเอสเอ็ม โดยการแปลงนี้จะต้องเลือกเทคโนโลยีที่จะใช้ก่อน ซึ่งเครื่องจากแปลงพีไอเอ็มโดยใช้ต้นแบบโครงสร้างเทคโนโลยีที่เครื่องมือมีอยู่แปลงไปเป็นพีเอสเอ็ม หลังจากได้พีเอสเอ็ม นักพัฒนาจะใช้โมเดลนี้สำหรับการพัฒนาระบบ แต่ยังไม่สามารถเขียนโปรแกรมได้เนื่องจากโมเดลพีเอสเอ็ม ยังไม่มีโครงสร้างของโค้ด การสร้างโค้ดของเครื่องมือจะแปลงพีเอสเอ็ม ซึ่งเป็นโครงสร้างโมเดลที่ขึ้นอยู่กับเทคโนโลยีอยู่แล้วออกมาเป็นโครงสร้างโค้ดโปรแกรมให้

ขั้นตอนการแปลงอัตโนมัติ นักออกแบบ นักพัฒนาและนักทดสอบ จะใช้เครื่องมือสำหรับการแปลง ซึ่งจากเดิมจะต้องแปลงด้วยมือ ทำให้เสียเวลาในการพัฒนา

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ปัจจุบันการพัฒนาด้วยเอ็มดีเอจะใช้การแปลงแบบอัตโนมัติด้วยเครื่องมือและไม่มีการแปลงด้วยมือแล้ว การแปลงด้วยเครื่องมือนี้ปัจจุบันมีเครื่องมือสำหรับการแปลงมากมาย เนื่องด้วยเทคโนโลยีที่พัฒนาไปอย่างรวดเร็ว เครื่องมือการแปลงของเอ็มดีเอแต่ละแบบก็มีโทคโนโลยีที่แตกต่างกันไป แต่ยังอยู่บนพื้นฐานของมาตรฐานโอเอ็มจี (OMG) ผู้กำหนดมาตรฐานของเอ็มดีเอ รูปที่ 2.3 แสดงให้เห็นว่าโมเดลพีไอเอ็มและพีเอสเอ็ม ถูกเปลี่ยนรูปโดยเครื่องมือ



รูปที่ 2.3 แสดง The Three Major Steips in the MDA Development process

2.4 การพัฒนาซอฟต์แวร์โดยวิธีเอจายล์(Agile software development)

2.4.1 หลักการพัฒนาซอฟแวร์โดยวิธีเอจายล์

เอจายล์เป็นหลักการในการพัฒนา ซอฟต์แวร์ที่ไม่มีกฎตายตัวว่าต้องทำตามขั้นตอนใดขั้นตอนหนึ่งแต่จะเน้นที่ความเหมาะสมกับคนหรือวัฒนธรรมขององค์กร เอจายล์ไม่เคร่งรัดในเรื่องของหลักการโดยจะเน้นไปที่เรื่องของคน การสื่อสารระหว่างทีมให้มีประสิทธิภาพ การสื่อสารกับลูกค้าและเน้นความพึ่งพอใจของลูกค้าเป็นหลัก ลดขั้นตอนใดที่สร้างความยุ่งยาก หรือไม่เหมาะสมกับวัฒนธรรมหรือองค์กร ก็จะสามารถปรับแต่งให้เหมาะสม หรือแม้กระทั่งเรื่องของเอกสารถ้าเอกสารใดต้องเสียเวลาทำจำนวนมากก็สามารถพิจารณาไม่ทำเอกสารนั้นตามความเหมาะสมหรือ ขั้นตอนการดำเนินงาน (Process) อื่น ๆ ที่ไม่จำเป็นมากนักซึ่งหลักของ เอจายล์มีไว้ขับเคลื่อนการทำงานให้เสร็จ กระบวนการทำงานไหนที่ทำให้งานช้าลงต้องได้รับการปรับปรุงหรือพิจารณายกเลิก ที่สำคัญต้องยอมรับความต้องการของระบบที่มีการเปลี่ยนแปลงอยู่เสมอจากลูกค้าและเน้นการออกแบบที่มีความง่าย ไม่สลับซับซ้อนทำให้การดูแลแก้ไขง่ายเมื่อพบการเปลี่ยนแปลง

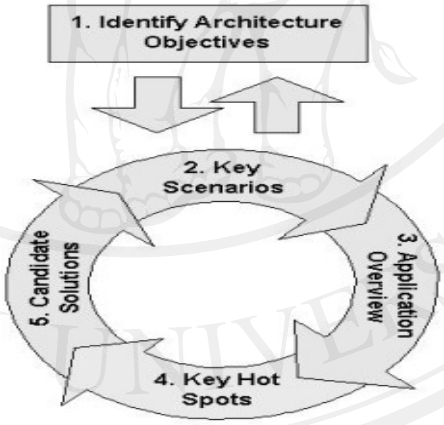
2.4.2 วัตถุประสงค์ของ Agile Method

วิธีการพัฒนาซอฟท์แวร์ทุกวิธีมีเป้าหมายเดียวกันคือเพื่อให้ได้ซอฟต์แวร์ที่สามารถทำงานได้ตามความต้องการที่กำหนดพัฒนาเสร็จภายในเวลาที่กำหนด อยู่ในงบประมาณที่ตั้งไว้และกระบวนการพัฒนาเป็นไปตามมาตรฐานทางวิชาการอันเป็นที่ยอมรับอย่างไรก็ดีการพัฒนาโดยใช้ Agile Method มีวัตถุประสงค์ในรายละเอียดเพิ่มเติมดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- เน้นความถนัดของแต่ละบุคคลและการพูดคุยสื่อสารกันมากกว่าการยึดติดที่เครื่องมือและกระบวนการในการทำงาน
- ทำงานโดยยึดที่ผลผลิตหรือซอฟต์แวร์เป็นหลักเช่น เดิมอาจเน้นเอกสาร แต่ Agile Method ไม่เน้นมากนักแต่จะเน้นว่ามีซอฟต์แวร์ส่งให้ผู้ใช้หรือยัง
- ให้ความสำคัญเรื่องการสื่อสารระหว่างทีมงานด้วยกัน และระหว่างทีมงานกับผู้ใช้
- ยอมรับความเปลี่ยนแปลงที่เกิดขึ้น เช่นเดิมต้องวางแผนล่วงหน้าให้ครบถ้วน และทำตามแผนที่กำหนดไว้ให้ได้ แต่ Agile Method ให้ความสำคัญกับการทำตามแผนน้อยลงแต่เน้นการสนองตอบต่อความเปลี่ยนแปลงที่เกิดขึ้นได้

การพัฒนาซอฟต์แวร์โทรศัพท์เคลื่อนที่โดยวิธีเอจายล์(Agile Software mobile development) เป็นการนำเอาเอจายล์มาพัฒนาใช้งานเฉพาะเจาะจงในการพัฒนาซอฟต์แวร์ทางด้านแอพพลิเคชั่น โดยมีจุดเด่น คือ การลดต้นทุนในการพัฒนาซอฟต์แวร์ ลดความยุ่งยากซับซ้อนในการพัฒนาและมุ่งเน้นในการพัฒนาซอฟต์แวร์เพื่อตอบสนองการใช้งาน โดยมีลักษณะจุดเด่นจาก แผนภาพข้างล่าง รูปที่ 2.4ซึ่งจะเน้นการพัฒนาที่เรียบง่าย โดยใช้วิธีการเอจายล์และนำการพัฒนาแบบสครัม (Scrum methodology) มาประยุกต์ใช้



รูปที่ 2.4 แสดงความสัมพันธ์ Agile Architecture Method

(http://blogs.msdn.com/jmeier/archive/2008/11/06/agile-architecture-method.aspx)

2.4.3 เปรียบเทียบ Agile Method กับวิธีการพัฒนาซอฟต์แวร์แบบอื่น

Agile Method เป็นวิธีการพัฒนาซอฟท์แวร์แบบตรงกันข้ามกับวิธีที่เน้นการวางแผนแบบเป็นระบบ และมีระเบียบ การระบุเช่นนี้อาจจะทำให้เกิดการเข้าใจผิดว่าเอจายล์เมธอด ไม่มีการวางแผนและไม่มีระเบียบ ซึ่งจริง ๆ แล้วไม่ใช่ ดังนั้นถ้าต้องการเห็นความแตกต่าง จึงควรจะ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

แยกแยะว่าเป็นแบบปรับเปลี่ยนได้หรือแบบตามแผนงาน ในที่นี้ เอจายล์เมธอด ถือว่าเป็นการพัฒนาแบบปรับเปลี่ยนได้

Agile Method เน้นความสามารถในการปรับเปลี่ยนได้อย่างรวดเร็วตามความเป็นจริงที่เปลี่ยน แปลงอยู่เสมอ เมื่อความต้องการของผู้ใช้เปลี่ยนแปลงไป ทีมงานพัฒนาก็ต้องพร้อมที่จะรับการเปลี่ยนแปลงตามไปด้วย ทีมงานพัฒนาดังกล่าวไม่สามารถบอกได้ว่าจะเกิดอะไรขึ้นในอนาคต ยิ่งวันที่ต้องการทราบว่าอะไรจะเกิดขึ้นไกลออกไปเท่าไร ความกำกวมของวิธีการและขั้นตอนในการตอบสนองก็จะยิ่งมีมากขึ้นเท่านั้น ทีมงานพัฒนาจะสามารถบอกถึงรายละเอียดของงานที่จะทำในสัปดาห์ต่อไปได้ แต่จะสามารถบอกได้เพียงแค่ลักษณะเด่นที่ได้วางแผนไว้ในเดือนถัดไปเท่านั้น

เมื่อเปรียบเทียบกันแล้ว วิธีการแบบตามแผนงานจะเน้นการวางแผนของอนาคตโดยละเอียด ทีมงานพัฒนาที่ใช้วิธีการแบบนี้จะสามารถบอกได้ถึงรายละเอียดทั้งหมดที่วาง เอาไว้ตลอดทั้งช่วงระยะเวลาของโครงการ แต่ก็ไม่สามารถปรับเปลี่ยนทิศทางการดำเนินโครงการได้ง่าย ๆ ทั้งนี้เพราะแผนงานที่วางไว้จะเป็นแผนงานที่เหมาะสมที่สุดที่สามารถนำไปสู่ จุดประสงค์ การปรับเปลี่ยนการดำเนินงานอาจทำให้งานทั้งหมดล้มเหลวได้ ทีมงานพัฒนาของวิธีการแบบนี้มักจะตั้งกรรมการที่รับผิดชอบการเปลี่ยนแปลง เพื่อรองรับความเปลี่ยนแปลงบางอย่างที่ได้มีการกำหนดเอาไว้ก่อนแล้วเช่นกัน

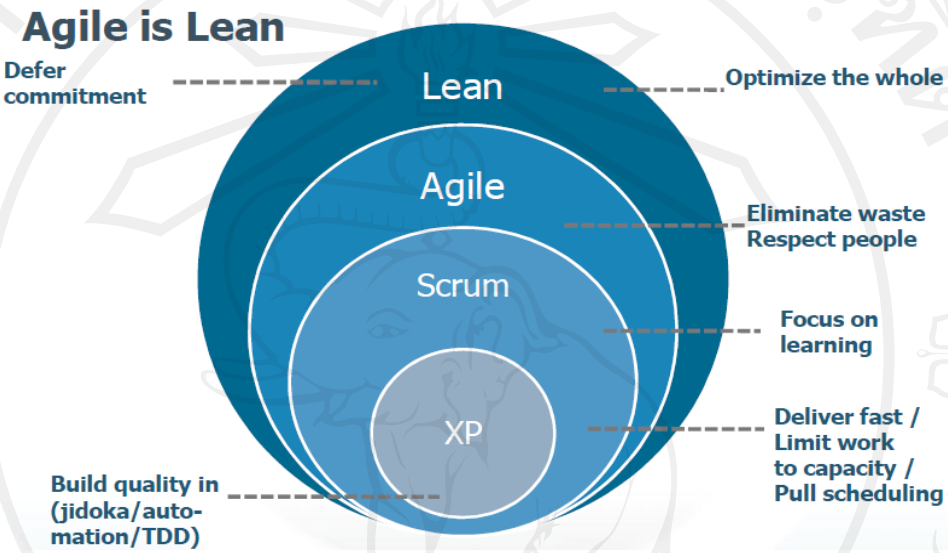
2.4.4เปรียบเทียบกับ Waterfall Model

อาจกล่าวได้ว่า Agile Method ไม่มีอะไรที่เหมือนกับ Waterfall Model เลย แต่ Waterfall Model ก็ยังเป็นวิธีการพัฒนาระบบที่ใช้กันอยู่ค่อนข้างแพร่หลายซึ่งวิธีนี้จะมีลำดับขั้นตอนที่ตายตัว เริ่มตั้งแต่รวบรวมข้อมูลกำหนดความต้องการของผู้ใช้ วิเคราะห์ทางเลือก ออกแบบ เขียนโปรแกรม ทดสอบระบบ และสุดท้ายทำการติดตั้งระบบโดยแต่ละส่วนของขั้นตอนดังกล่าวจะถือเป็นตัววัดความก้าวหน้าของงานปัญหาสำคัญของ Waterfall Model คือขั้นตอนของการพัฒนาที่ไม่ยืดหยุ่น เพราะตัวงานจะแบ่งเป็นช่วงๆแบบตายตัวทำให้มีข้อผูกมัดตั้งแต่เริ่มโครงงานและไม่สามารปรับเปลี่ยนความต้องการผู้ใช้ได้ หมายความว่าการพัฒนาโดยใช้ Waterfall Model นั้นไม่เหมาะกับงานที่ความต้องการของผู้ใช้เข้าใจยากและมีแนวโน้มว่าจะเปลี่ยนแปลงตลอดเวลา

ในทางกลับกัน Agile Method จะแบ่งงานออกเป็นส่วนย่อยๆ แล้วค่อย ๆทดสอบไปเรื่อยๆ ทุกสัปดาห์ หรือทุกสองสัปดาห์ทั้งนี้จะเน้นสร้างส่วนย่อยที่สุดของงานทั้งหมดที่สามารถใช้งานได้ทีละชิ้นเพื่อให้ส่งมอบได้รวดเร็วและทำการปรับปรุงเพิ่มเติมอย่างต่อเนื่องตลอดช่วงเวลาของโครงงานดังรูป 2.5เปรียบเทียบความสัมพันธ์ของแต่ละวิธีการพัฒนาซอฟต์แวร์ ด้วยหลักการเอ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

จายล์ เป็นพื้นฐานของการศึกษาการทำงานแต่ละวิธีการ ซึ่งความสัมพันธ์ของขอบเขตการทำงานแต่ละซับเซตวิธี ที่ได้แสดงถึงความสำคัญของลักษณะการทำงานและการเชื่อมโยงการทำงานที่สามารถทำงานเชื่อมโยงหากันได้



รูปที่ 2.5 เปรียบเทียบความสัมพันธ์ของวิธีการพัฒนาซอฟต์แวร์ด้วยเอจายล์

(http://blog.crisp.se/henrikkniberg/)

2.4.5 ความเหมาะสมของการเลือกAgile Method

ถึงแม้ว่าหลายๆ วิธีที่ถือว่าเป็นวิธีพัฒนาแบบเอจายล์(Agile Method)จะแตกต่างกันออกไป แต่ก็ยังมีหลายๆส่วนที่เหมือนกัน เช่นเป็นการพัฒนาระบบแบบทำซ้ำ เน้นการปฏิสัมพันธ์การสื่อสารและการลดปริมาณชิ้นงานที่ต้องใช้ทรัพยากรมากความเหมาะสมของวิธีการพัฒนาแบบเอจายล์ สามารถระบุได้จากหลายมุมมองถ้าเป็นการระบุความเหมาะสมตามชนิดของงาน วิธีการพัฒนาแบบ Agile Method จะเหมาะสมกับงานที่ความต้องการของลูกค้ามีการเปลี่ยนแปลงอยู่เสมอ แต่จะไม่เหมาะสมกับระบบที่เน้นด้านความปลอดภัยและต้องการให้ระบบดังกล่าวน่าเชื่อถือในระดับสูงสุดอีกมุมมองหนึ่งในการระบุความเหมาะสมก็คือมุมมองขององค์กรปัจจัยที่จะนำมาซึ่งความสำเร็จขององค์กรมีดังต่อไปนี้

- วัฒนธรรมองค์กรนั้น ๆ จะต้องยอมรับและสนับสนุนการเจรจาต่อรอง
- บุคลากรภายในองค์กรจะต้องเชื่อถือได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- จำนวนบุคลากรต้องไม่มากจนเกินไป และทุกคนควรจะเป็นบุคลากรที่มีคุณภาพเมื่อเปรียบเทียบกับบุคลากรขององค์กรอื่น ๆ
- องค์กรจะต้องให้ความสำคัญหรือยอมรับการตัดสินใจของผู้พัฒนาระบบ
- องค์กรจะต้องมีสภาพแวดล้อมในองค์กรที่เกื้อหนุนการติดต่อสื่อสารที่รวดเร็วระหว่างกันของสมาชิกในทีมพัฒนา

ปัจจัยที่สำคัญที่สุดน่าจะเป็นขนาดของตัวโครงงานยิ่งโครงงานใหญ่มากเท่าไร การพบปะสนทนาก็จะยิ่งทำได้ยากขึ้นเท่านั้นดังนั้นวิธีการพัฒนาแบบเอจายล์จึงเหมาะสำหรับโครงงานที่มีทีมพัฒนาขนาดเล็ก (ไม่เกิน 20 – 40 คน)

ปัญหาใหญ่อีกปัญหาหนึ่งก็คือการด่วนสรุปหรือการเร่งรัดเอาคำตอบในเรื่องของความต้องการของผู้ใช้อาจจะนำมาซึ่งการแก้ปัญหาที่ไม่ตรงจุดโดยเฉพาะเมื่อการตัดสินใจในเรื่องดังกล่าวไม่ได้มาจากการไตร่ตรองโดยละเอียดของผู้ใช้ในบางครั้งผลลัพธ์ที่ได้จากโครงงานอาจเกิดขึ้นจากการพูดจาหว่านล้อมของผู้พัฒนาระบบเอง ทำให้การตัดสินใจเหล่านั้นไม่เหมาะสมไม่ได้แก้ปัญหาที่เกิดขึ้นและนำไปใช้งานจริงไม่ได้

ปัญหาดังกล่าวสามารถบรรเทาได้โดยการแบ่งแยกการรวบรวมความต้องการของผู้ใช้ออกเป็นช่วงๆแล้วพยายามวนเวียนเข้าหาผู้ใช้อย่างสม่ำเสมอเพื่อให้ผู้ใช้ได้ทดลองใช้และประเมินผลลัพธ์ของแต่ละช่วงซึ่งการกระทำดังกล่าวนี้ก็ยังประสบปัญหาอีกปัญหาหนึ่ง คือผู้ใช้ส่วนใหญ่จะไม่ลงทุนเสียเวลามากและการตรวจสอบคุณภาพก็ทำได้ยากกับงานที่สำเร็จไปทีละขั้น การระบุว่าวิธีการพัฒนาแบบเอจายล์นั้นเหมาะสมหรือไม่จะต้องมีตัวช่วยในการวิเคราะห์ที่ค่อนข้างซับซ้อนเข้ามาช่วยโดยตัวช่วยวิเคราะห์ดังกล่าวอาจสร้างตัวเลือกหรือแนะนำว่าควรใช้วิธีการแบบใดกับการพัฒนาโครงงานนั้น ๆวิธีการพัฒนาแบบเอจายล์แต่ละแบบนั้นเหมาะสมกับช่วงของการพัฒนาระบบที่แตกต่างกันออกไปโดยลักษณะเหล่านี้จะเป็นตัวที่ช่วยในการเลือกวิธีพัฒนาให้เหมาะสมกับโครงงานต่าง ๆ ได้

2.4.6 จุดเด่นของวิธีการพัฒนาแบบเอจายล์(Agile Method)

จุดเด่นของวิธีการพัฒนาแบบเอจายล์คือการปรับตัวเข้ากับระบบที่สามารถทำได้อย่างรวดเร็วเมื่อองค์กรต้องการเปลี่ยนแปลงและปรับปรุงการทำงานของทีมงานทีมปรับปรุงจะมีความยากที่จะบรรยายสิ่งต่างๆที่จะปรับเปลี่ยนในช่วงเวลาข้างหน้าซึ่งมีความไม่แน่นอนสูงและมีรู้ว่าเมื่อปรับเปลี่ยนแล้วจะเกิดผลเป็นอย่างไรภายในช่วงเวลานั้นๆ ในระยะยาวทีมปรับปรุงจะสามารถที่จะรายงานที่แม่นยำในการทำงานในช่วงเวลาที่ทำภายในระยะประมาณ 1 สัปดาห์แต่เมื่อมีคำถามทีมงานจะสามารถที่จะบรรยายภารกิจและมูลค่าที่คาดว่าจะได้รับกับต้นทุนที่ได้ลงทุนไป

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การพยากรณ์วิธีการ(Methods)ในการเปรียบเทียบจุดเด่นเป็นการวางแผนรายละเอียดปลีกย่อยทีมพยากรณ์สามารถที่จะรายงานความแม่นยำว่างานอะไรจะเกิดขึ้นเป็นการวางแผนภายในช่วงเวลาภายใต้กระบวนการพัฒนาเป็นงานยากที่จะควบคุมการวางแผนอย่างดีสำหรับจุดมุ่งหมายและการเปลี่ยนแปลงทิศทางสามารถทำให้การทำงานเป็นไปอย่างสมบูรณ์ และลดความยากลำบากของงานโดยสามารถสรุปได้ให้เห็นชัดเจน เป็นลำดับข้อ ดังนี้

- สามารถของเขต(Scope)งานขั้นต่อไปได้อย่างรวดเร็ว โดยดูแนวโน้มทั้งด้านธุรกิจและเทคโนโลยีด้วย
- ส่งมอบชิ้นงานได้รวดเร็ว
- มีประสิทธิภาพถึงแม้ว่างานนั้น ๆ จะยังไม่ได้มีการตกลงกันในเรื่องของความต้องการของผู้ใช้
- รองรับความเปลี่ยนแปลงที่อาจเกิดขึ้นได้ตลอดเวลา ทำให้การพัฒนามีความยืดหยุ่นมากกว่า
- มุ่งเน้นความง่ายของตัวโครงงาน ในขณะเดียวกันก็ตัดส่วนที่ทำให้เกิดความสูญเปล่าทิ้งไป
- ลูกค้าพึงพอใจมากกว่าเพราะตรงกับความต้องการในตลอดทุกขั้นตอนการทำงาน
- แนวโน้มว่าจะต้องแก้งานในส่วนที่เสร็จสิ้นแล้วน้อยมาก เพราะมีการอัพเดทความต้องการของผู้ใช้อยู่เสมอ

2.4.7 ข้อจำกัดของวิธีการพัฒนาแบบเอจายล์ (Agile Method)

- ทีมพัฒนาอาจใช้วิธีนี้ไม่ได้ ถ้าลูกค้าไม่มีความไว้เนื้อเชื่อใจในตัวทีมพัฒนา
- สมาชิกในทีมพัฒนาจะต้องมีทักษะด้านการปฏิสัมพันธ์ที่ดีเพราะต้องติดต่อพบปะกับลูกค้าเสมอ
- เนื่องจากเป็นวิธีแบบทำซ้ำที่ต้องมีการพูดคุยเรื่องความต้องการหลาย ๆรอบทำให้มีโอกาสสูงที่ขอบเขตของโครงงานจะกว้างเกินไปไม่มีที่สิ้นสุด
- ส่งผลเสียต่อทีมพัฒนาในเรื่องของตารางเวลานัดหมาย และส่งผลเสียต่อผู้ใช้ในเรื่องของค่าใช้จ่าย
- ลูกค้าส่วนใหญ่ต้องการเห็นตัวงานเป็นชิ้นเป็นอันในเวลาที่ค่อนข้างสม่ำเสมอและอยู่ในขอบเขตงบประมาณที่กำหนดไว้ตายตัวซึ่งตรงกันข้ามกับแนวคิดของวิธีพัฒนาแบบ Agile

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- ต้องเข้าใจถึงวัฒนธรรมองค์กรมากกว่าวิธีแบบอื่น ๆ รวมถึงต้องตัดสินใจว่าควรใช้วิธีใดให้เข้ากับสภาพแวดล้อมขององค์กรดังกล่าว
- เนื่องด้วยขั้นตอนที่ยุ่งยากและเสียเวลาของวิธีแบบ Agile ทำให้ผู้บริหารหัวเก่าบางท่านไม่นิยมและพร้อมที่จะเลือกใช้วิธีทันทีถ้าวิธีดังกล่าวสามารถสร้างผลลัพธ์ได้ใกล้เคียงกันและมีขั้นตอนที่ง่ายกว่า

2.5การพัฒนาซอฟต์แวร์ด้วยวิธีสครัม

สครัม (Scrum) เป็นวิธีการหนึ่งของการพัฒนาซอฟต์แวร์ด้วยหลักการเอจายล์ โดยมีรูปแบบการจัดการแบบที่แต่ละคนต่างผลักดันซึ่งกันและกันสามารถนำไปปรับ ใช้ในการบริหารจัดการและควบคุมทั้งโครงงานแบบทำซ้ำและแบบเพิ่มรายละเอียดในแต่ละขั้นตอน การพัฒนาแบบสครัม ผู้ใช้ระบบจะต้องทำงานร่วมกับทีมพัฒนาอย่างใกล้ชิด เพื่อที่จะระบุและจัดลำดับความสำคัญก่อนหลังของหน้าที่ของระบบ แล้วเก็บรวบรวมไว้ในรูปแบบที่เรียกว่า Product Backlog ข้อมูลที่อยู่ภายใน Product Backlog ประกอบด้วยลักษณะเด่น การแก้ไขข้อผิดพลาดต่าง ๆ และความต้องการของผู้ใช้ในส่วนอื่น ๆ เป็นต้น การส่งมอบงานของวิธีสครัม จะเป็นชิ้นงานที่ทำงานได้บางส่วนที่เกิดจากการแบ่งงานทั้งงานออกเป็นส่วนย่อย ๆ โดยระยะเวลาระหว่างการส่งมอบงานจะอยู่ในช่วงประมาณ 30 วัน ดังนั้น สครัมจึงเป็นกระบวนการพัฒนาที่อยู่บนพื้นฐานของสปรินท์ (Sprint) หมายถึงวิธีการที่ให้เราทำงานเป็นช่วง ๆ คือ ออมแรงไว้ก่อน แล้วสปรินท์เป็นช่วง ๆ ตามหลักการ คือ 2 - 4 สัปดาห์

มีสาระสำคัญของสครัม มี 3 อย่างคือ

1. ทีมงาน มีลักษณะที่สำคัญ 3 ประการคือ

- Scrum Teamคือ กลุ่มบุคคลที่ปฏิบัติงานจริงประมาณ 5 - 9 คน แต่ละคนไม่ได้กำหนดงานตายตัว สามารถทดแทนกันได้เสมอ นอกจากนี้แต่ละคนในทีมงานมีหน้าที่ประเมินเวลาของงานที่ต้องทำ แบ่งงานและส่งงานกันเอง

- Product Ownerคือ ผู้ที่ทำหน้าที่จัดการเรื่อง product backlog คิด รวบรวม เผยแพร่ให้ทุกคนรับรู้ เพื่อให้คนในทีมเห็นแนวปฏิบัติในอนาคตว่ามีงานอะไรบ้าง ซึ่งบุคคลคนนี้จะเป็นคนเขียนรายละเอียดและความต้องการของผู้ใช้ด้วย

- Scrum Masterคือผู้ที่ทำหน้าที่ดูแลทีมงาน เป็นโค้ชของทีมงาน เป็นคนที่รับผิดชอบคุณภาพของผลงาน จัดลำดับความสำคัญของงาน แตกงานให้ที่ประชุมตัดสินใจตามความเหมาะสม

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2. กระบวนการทำงาน มีลักษณะที่สำคัญ 3 ประการ คือ

- **Backlog**เป็นรายการของคุณลักษณะที่ต้องทำซึ่งรวมถึงความต้องการจากผู้ใช้ การแก้ไขข้อผิดพลาด และกำหนดคุณลักษณะพิเศษเฉพาะของตัวผลงานโดยคนที่ทำคือ product owner ซึ่งจะจัดลำดับคุณลักษณะตามความสำคัญ จัดรายการเพื่อนำเข้า sprint และจัดการกับรายละเอียดต่าง ๆ ของคุณลักษณะเหล่านั้น

- **Sprint phase**คือช่วงที่จะต้องทำงานซ้ำ ๆ โดยมีกำหนดไม่เกิน 30 วัน ก่อนเริ่ม sprint ก็จะมีการนำ product backlog มาจัดลำดับความสำคัญเพื่อเลือกมาเป็น sprint backlog จากนั้น scrum team จะดู backlog และแตกเป็นงานย่อย ๆ ออกมาและทำการประมาณเวลาที่ใช้ในแต่ละงาน หลังจากได้เวลาและต่อรองกันระหว่างทีมงานแล้ว ก็จะได้รายการของงานและรายการของ backlog

- **Daily scrum**มีลักษณะเป็นการประชุมอย่างไม่เป็นทางการ อาจเป็นการยืนคุยกัน โดยทุก ๆ วัน scrum master และ scrum team จะพบปะพูดคุยกันเพื่อทบทวนว่าเมื่อวานทำอะไรไปบ้าง และวันนี้จะทำอะไร มีการถกกันเพื่อแก้ไขปัญหาที่พบเมื่อวาน และจัดการกำหนดงานให้กับทีมงานแต่ละคน

3. การสาธิตและการประเมินผลงาน

จุดเด่นของ Scrum คือสามารถวัดผลการทำงานได้ มีผังการทำงานที่เรียบง่าย และธรรมดา แต่ทำให้เห็นภาพของ sprintได้อย่างชัดเจนโดยหลักการแล้วก็คือกราฟของงานโดยแกน y แทนจำนวนงานที่เหลือ ส่วนแกน x แทนวันแต่ละวันของ sprint โดยในแต่ละรายการจะมีการปรับเปลี่ยนกราฟ เพื่อแสดงให้เห็นภาพความคืบ หน้าของงาน หลังจากจบ sprint ก็จะเอากราฟนี้มาประเมินผลงานของทีมงาน

2.6ระบบติดตามงานวิจัย

ระบบติดตามงานวิจัยโดยใช้สถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง เป็นระบบที่จะพัฒนาขึ้นเพื่อใช้สำหรับติดตามความก้าวหน้าของโครงการวิจัย ที่อยู่ในระหว่างการดำเนินการในแต่ละขั้นตอนของคณะทำงานและอนุกรรมการวิจัย ซึ่งเป็นขั้นตอนสำคัญ ผู้ดำเนินโครงการวิจัยจะต้องรายงานความก้าวหน้าในแต่ละงวดงานของโครงการวิจัย ให้กับคณะทำงานติดตามและ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

คณะอนุกรรมการวิจัย คณะอนุกรรมการวิจัยจะมีข้อเสนอแนะการทำงานและบันทึกผลการติดตามความก้าวหน้าเพื่อเป็นข้อมูลอ้างอิงในการติดตามงวดงานในครั้งต่อไป นอกจากนี้ยังต้องการสารสนเทศที่สรุปการติดตามความก้าวหน้าในด้านของงานที่ทำและงบประมาณที่เบิกจ่ายในแต่ละงวด โดยระบบจะต้องสามารถทำงานได้ดังต่อไปนี้

- ระบบสามารถจัดเก็บรายละเอียดข้อมูลโครงการวิจัยที่รับผิดชอบโดยสำนักวิจัย การบันทึกโครงการวิจัยซึ่งเนื้อหาสอดคล้องกับแผนงานวิจัย แผนงานประจำปี ครอบคลุมถึงโครงการที่ทางสถาบันดำเนินการเองและภายนอกขอรับทุนสนับสนุนการวิจัย ข้อมูลนักวิจัย ข้อมูลงานวิจัยที่เผยแพร่

- ระบบสามารถสนับสนุนการทำงานตามขั้นตอนการทำงานของสำนักวิจัย ตั้งแต่การติดตามงานวิจัยและสรุปรายงานตามงวดงาน โดยรูปแบบการติดตามมีทั้งด้านการใช้จ่ายงบประมาณและด้านของเนื้องานที่ได้ดำเนินการ

- เจ้าหน้าที่ผู้รับผิดชอบในส่วนของกลุ่มงานวิจัยสามารถเข้าไปบันทึกสถานะต่างๆของโครงการได้ตามงวดงานที่กำหนด สามารถแนบเอกสารและบันทึกข้อมูลประกอบในรายละเอียดแต่ละงวดงานที่ได้ดำเนินการวิจัยในแต่ละโครงการ

- คณะอนุกรรมการติดตามโครงการวิจัยสามารถเรียกดูเอกสารหรือรายละเอียดของโครงการและสามารถแนบเอกสารหรือข้อความคำแนะนำได้ตามงวดงานของโครงการ

-ระบบสามารถรายงานผลในเรื่องของงบประมาณที่เบิกจ่ายและผลการดำเนินการให้กับเจ้าหน้าที่ประจำโครงการและคณะอนุกรรมการติดตามงานในรายโครงการ สรุปรายงานภาพรวมทุกโครงการให้กับผู้บริหาร

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved