

Project Management

Unit 6

Key Methodologies for IT Project Management

อาจารย์กীরติบุตร กาญจนเสถียร

เนื้อหา

- Waterfall
- Spiral
- Agile

Methodology

Methodology แปลว่า รูปแบบ หรือวิธีในการดำเนินการ หรือรูปแบบวิธีปฏิบัติในการทำสิ่งใดๆ ซึ่ง methodology ในการพัฒนาซอฟต์แวร์ หมายถึงรูปแบบที่ทีมพัฒนาซอฟต์แวร์ใช้ในการบริหารจัดการทรัพยากรต่างๆไม่ว่าจะเป็นทรัพยากรวัสดุ สิ่งของ คน รวมไปถึงขั้นตอนการทำงานต่างๆ เพื่อให้การพัฒนาซอฟต์แวร์นั้นสำเร็จลุล่วงไปด้วยดี ซึ่งเรียกอีกอย่างหนึ่งว่า software development model หรือ software development life cycle (SDLC)

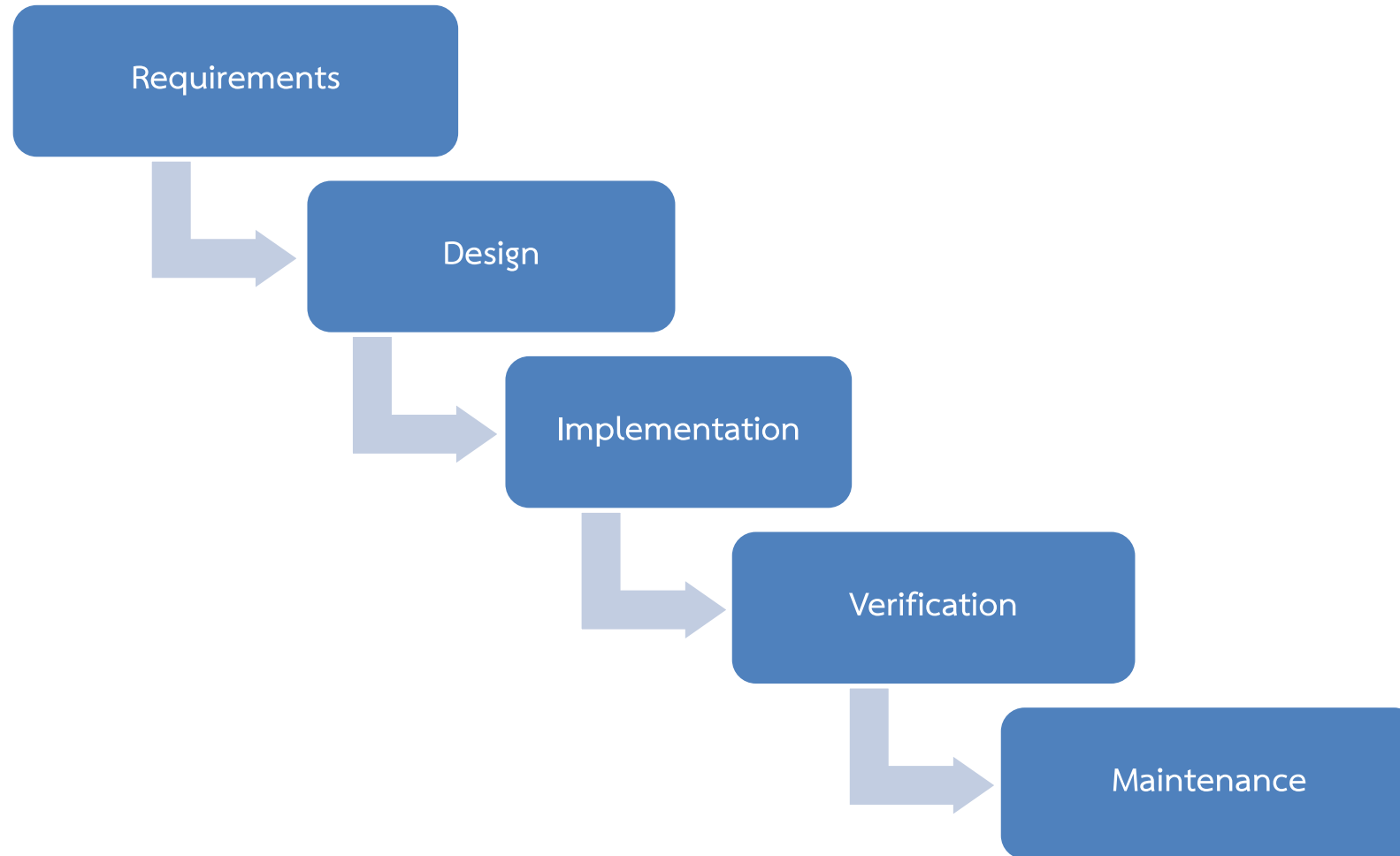
1. Waterfall Model

Waterfall model ถือเป็นโมเดลเชิงเส้นที่ตรงไปตรงมา มีการทำงานที่ไม่ซับซ้อน เป็นโมเดลดั้งเดิมที่มีการใช้งานมายาวนานตั้งแต่การพัฒนาระบบ IT ในยุคก่อนหน้า โดยโมเดลประกอบไปด้วยขั้นตอนที่เรียงกันตามลำดับ ได้แก่

1. requirements
2. design
3. implementation
4. verification
5. maintenance

แต่ละขั้นตอนจะต้องเสร็จสิ้น 100% ก่อนที่จะไปดำเนินการในขั้นตอนต่อไปได้ ซึ่งหากใช้โมเดลนี้ในการพัฒนาซอฟต์แวร์ จะไม่มีขั้นตอนหรือแนวทางในการย้อนกลับไปแก้ไข

Waterfall Model



1. Waterfall Model

ข้อดี: เนื่องจากโมเดลมีขั้นตอนการทำงานเป็นเส้นตรง ส่งผลให้ง่ายต่อการบริหารจัดการ เหมาะสมกับโปรเจ็คหรือโครงการที่มีเป้าหมายและความต้องการ (requirement) ที่ชัดเจน ผู้คุมโครงการไม่จำเป็นที่จะต้องมีการประสานงานในการคุมโครงการมากนักก็สามารถใช้เทคนิคนี้ได้เนื่องจากมีความตรงไปตรงมาไม่ซับซ้อน เหมาะสำหรับงาน เช่น งานราชการ งานที่บุคลากรมีตำแหน่งหน้าที่ชัดเจน งาน freelance ที่ทำคนเดียว

ข้อสังเกต: การดำเนินงานตามโมเดลนี้จะยึดแผนการทำงานและ deadline เป็นหลัก ซึ่งในหลายครั้งส่งผลให้เกิดการล่าช้าเนื่องจากธรรมชาติของการทำโครงการมักที่จะเกิดการล่าช้าจากการบริหารความเสี่ยงที่ไม่ครอบคลุม อีกทั้งไม่มีวิธีการรองรับการบริหารบุคลากรหากมีความจำเป็นต้องมีการสลับตำแหน่ง จะทำให้การทำงานในส่วนอื่นเกิดข้อบกพร่องได้

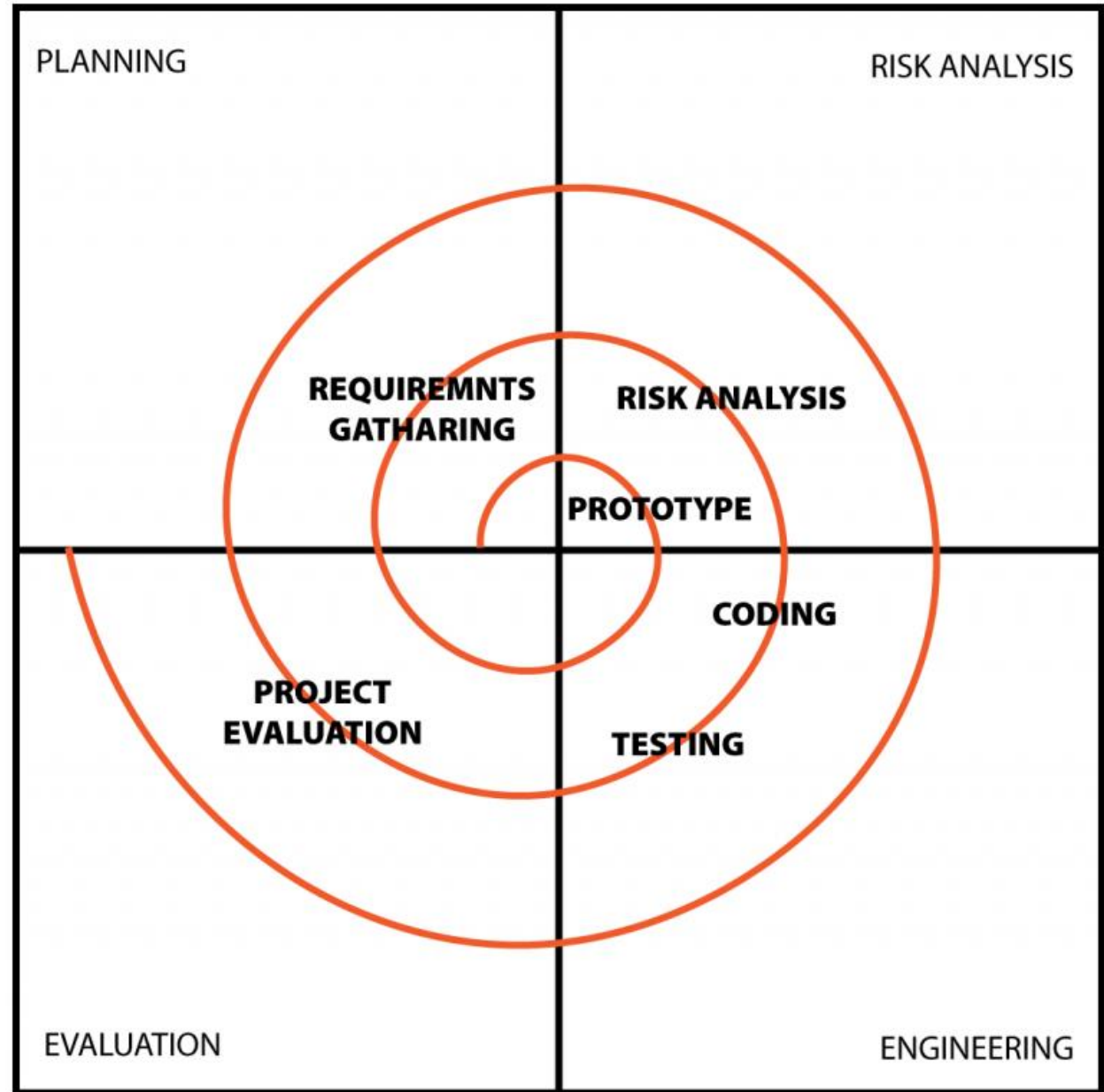
2. Spiral Model

Spiral model หรือ meta model เป็นโมเดลที่มีลักษณะเป็นรูปก้นหอยลากเส้นโค้งวนจากด้านในออกมาด้านนอก ซึ่งจะมีลักษณะการทำงานเป็นลูปคือสามารถวนลูปได้เรื่อยๆจนกว่าการพัฒนาซอฟต์แวร์จะเสร็จสิ้น โดยแต่ละลูปจะเรียกว่าเฟส “phase” ของการพัฒนาซอฟต์แวร์ โดยจะแบ่งเป็น 4 เฟส ได้แก่

1. การได้มาซึ่งความต้องการของระบบ (requirements)
2. การวิเคราะห์ปัญหาหรือความเสี่ยง (risk management)
3. การพัฒนาระบบ (implementation)
4. การทดสอบและประเมินผล (testing and evaluation)

ซึ่งโมเดลนี้สามารถใช้ในการบริหารความเสี่ยงได้ดีเนื่องจากการบริหารความเสี่ยงในทุกขั้นของการทำงาน อีกทั้งยังมีช่วงให้สามารถปรับแก้ไข requirements อีกด้วย การพัฒนาซอฟต์แวร์ที่แบ่งเป็นฟิเจอร์โดยการพัฒนาไปทีละส่วนจะเหมาะสมกับการใช้โมเดลนี้

Spiral Model



2. Spiral Model

ข้อดี: การรับมือกับความเสี่ยงถือเป็นจุดเด่นของโมเดลนี้ ซึ่งจัดว่าเป็นหนึ่งในโมเดลที่ดีที่สุดในด้านการวิเคราะห์รับมือ และจัดการกับความเสี่ยงซึ่งจะทำในทุกๆ เฟสของการพัฒนาซอฟต์แวร์ อีกทั้งยังมีความยืดหยุ่นในการจัดการกับความต้องการของระบบ (requirements) เนื่องจากจะมีการตรวจปรับความต้องการอยู่ในทุกๆ เฟส เช่นเดียวกัน เป็นโมเดลที่เหมาะสมสำหรับโครงการขนาดใหญ่ที่มีความซับซ้อน

Cons: ไม่เหมาะสมสำหรับโครงการเล็กๆ หรือโครงการที่มีงบประมาณน้อยเนื่องจากการบริหารความเสี่ยง หรือ การเก็บ requirements ในทุกๆ เฟสของการพัฒนาระบบจะมีค่าใช้จ่ายที่มาก เป็นโมเดลที่มีความซับซ้อนกว่าโมเดลแบบอื่นๆ เนื่องจากการทำซ้ำกิจกรรมเดิมๆ ในทุกๆ เฟสส่งผลให้เปลืองทรัพยากร ต้องใช้ผู้เชี่ยวชาญในการทำงานเนื่องจากการวิเคราะห์ความเสี่ยงอยู่เป็นประจำ บริหารจัดการการทำงานด้วยโมเดลนี้ได้ยาก

3. Agile Model

Agile methodology เป็นรูปแบบการพัฒนาซอฟต์แวร์ที่นิยมใช้ในยุคปัจจุบัน เนื่องจากรูปแบบของการพัฒนาซอฟต์แวร์ในปัจจุบันมีความแตกต่างจากในอดีตเป็นอย่างมาก ในอดีตระบบ IT จะเป็นระบบที่ไม่มีระบบย่อยมาก เช่น พัฒนาโปรแกรมจำนวน 1 โปรแกรม พัฒนาเว็บไซต์จำนวน 1 เว็บไซต์ หรือพัฒนาระบบ ERP จำนวน 1 ระบบ ซึ่งสามารถบริหารจัดการให้เสร็จสิ้นโดยใช้โมเดลแบบ waterfall ให้เสร็จได้อย่างราบรื่นโดยที่ไม่มีการเปลี่ยนแปลงเนื้อหาหรือ requirement

แตกต่างจากระบบ IT ในปัจจุบันที่มีการทำงานเป็นแบบ micro-service มีการเชื่อมต่อกับระบบภายนอก มีจำนวนผู้ใช้หลายกลุ่ม มีการใช้ API จากผู้ให้บริการที่หลากหลาย อีกทั้งแรงกดดันจากคู่แข่งที่มีการพัฒนาเช่นเดียวกัน ส่งผลให้ความต้องการของระบบหรือ requirement อาจต้องมีการเปลี่ยนแปลงในระหว่างทางของการพัฒนา ซึ่งการพัฒนาซอฟต์แวร์ตามแนวคิดแบบ Agile จะรองรับและสนับสนุนการทำงานในลักษณะนี้

3. Agile Model

Agile ยังถูกใช้เป็นวิธีการทำงานขององค์กรในภาพรวม หรือในบางองค์กรจะมีวัฒนธรรมทำงานเป็นแบบ agile โดย agile จะยึดผู้นำของทีมหรือผู้บริหารองค์กรเป็นคนที่สำคัญซึ่งมีหน้าที่นำพาองค์กรหรือทีมไปสู่จุดหมาย ความต้องการของลูกค้าหรือผู้ใช้ถือเป็นความสำคัญอันดับสูงสุดของ agile อีกทั้งยังเป็น methodology ที่สนับสนุนให้คนในทีมได้สื่อสารกันมากขึ้น ซึ่งปกติจะมีการพบปะเพื่อพูดคุยหรือรายงานผลการทำงานกันเป็นรายวัน ขั้นตอนใน Agile มี 6 ขั้นตอน ได้แก่

1. requirements รวบรวมความต้องการของระบบ
2. design ออกแบบระบบ
3. development พัฒนาระบบ
4. testing ทดสอบระบบ
5. deployment ติดตั้งระบบหรือนำไปใช้
6. review รับ feedback จากลูกค้า/ผู้ใช้

3. Agile Model

ข้อดีของ Agile คือ

1. ทุกคนในทีมได้มีส่วนร่วมและได้มีโอกาสได้พูดคุยสื่อสารกัน
2. สนับสนุนการได้มาซึ่ง feedback จากลูกค้า
3. มีความยืดหยุ่น รองรับการปรับเปลี่ยน requirements
4. มีการติดต่อสื่อสารกับผู้ใช้หรือลูกค้าอยู่เสมอ

ข้อจำกัดของ Agile คือ

1. การระบุ deadline ที่แน่นอนอาจทำได้ยาก เนื่องจากมีการพูดคุย และการพูดคุยนำมาซึ่งการเปลี่ยนแปลง ดังนั้นยังมีการพูดคุยกับลูกค้ามากขึ้น อาจส่งผลให้มีการปรับเปลี่ยนกำหนดการบ่อยขึ้น เช่น งานเสร็จเร็วกว่ากำหนด หรือมีการเพิ่มงาน
2. อาจเป็นการยากที่จะให้ทุกคนในทีมหรือองค์กรมีปฏิสัมพันธ์กันเพื่อให้การพัฒนาซอฟต์แวร์ไปต่อได้

3. Agile Model

จากคำกล่าวที่ว่า agile นั้นคาดการณ์ deadline ได้ยาก เนื่องจากเมื่อได้ requirements แล้วจะเข้าสู่รูปของการพัฒนาซอฟต์แวร์ และเมื่อถึงขั้นตอนสุดท้ายที่มี review เพื่อรับ feedback จากผู้ใช้ ผู้ใช้อาจยังไม่พอใจกับงานที่ได้ จึงเกิดเป็น requirements เพิ่มเติม ซึ่งก็จะวนกลับเข้าสู่รอบรอบ แต่ถ้าหากลูกค้ามีความพึงพอใจแล้วก็จะถือว่าเสร็จสิ้นกระบวนการพัฒนาซอฟต์แวร์ของ agile



3. Agile Model

จากรูปแผนภาพของ agile จะเห็นว่ามีลักษณะการทำงานคล้ายกับ spiral model แต่ความแตกต่างอยู่ที่ Agile จะต้องมีส่วนร่วมได้ส่วนเสียหรือลูกค้าหรือผู้ใช้ เข้ามามีส่วนร่วมในขั้นตอนต่างๆ อยู่เสมอมากที่สุดเท่าที่จะเป็นไปได้ อีกทั้ง agile ยังเน้นไปที่การพัฒนาระบบ ปรับปรุงและพัฒนาไปจนกว่าลูกค้าจะพอใจ ต่างกับ 2 model ก่อนหน้านี้ที่จะพัฒนาระบบให้เสร็จสิ้นในการทำงานครั้งเดียว ซึ่ง agile จะสามารถประกอบไปด้วยทีมมากกว่า 1 ทีมก็ได้ที่มีส่วนร่วมในการพัฒนาซอฟต์แวร์ตัวเดียวกัน แต่มีผู้นำเพียง 1 คนที่คอยบริหารจัดการแบ่งกลุ่มบุคคลในการทำงานแบบ agile ได้ดังนี้

1. stakeholders คือ ผู้มีส่วนได้ส่วนเสีย เช่น ลูกค้า ผู้ใช้งาน เจ้าของงบประมาณ
2. developer คือ นักพัฒนาระบบ
3. product owner คือ เจ้าของระบบ
4. project manager คือ ผู้นำทีมพัฒนาระบบ

End of UNIT 6

Key Methodologies for IT Project Management