

Written by Thapanapong Rukkanchanunt

# Software Development

# บทที่ 2 กระบวนการพัฒนาซอฟต์แวร์เบื้องต้น

เค้าโครงเนื้อหา

1. บทนำ
2. กระบวนการพัฒนาซอฟต์แวร์โดยรวม
3. โครงสร้างแบบน้ำตก (Waterfall)
4. วิธีการพัฒนาซอฟต์แวร์แบบคล่องแคล่วว่องไว (Agile)

# Programming vs Software Development

- Programming หมายถึงการเขียนโปรแกรมคอมพิวเตอร์เพื่อแก้ปัญหาในชีวิตประจำวัน
- Software Development หมายถึงการพัฒนาโปรแกรมคอมพิวเตอร์อย่างเป็นระบบ
- เมื่อเราต้องเขียนโปรแกรมขนาดใหญ่ มีผู้ใช้มากมาย และมีจำนวนบรรทัดของคำสั่งหลายพันหลายหมื่น เราควรจะต้องวางแผนให้ดีและทำงานอย่างเป็นระบบ ดังนั้นการเรียนรู้กระบวนการที่เกี่ยวข้องกับ Software Development จึงมีความจำเป็นอย่างยิ่ง
- ในคลาสนี้เราจะได้เรียนรู้กระบวนการหลัก และวงจรชีวิตของการพัฒนาซอฟต์แวร์

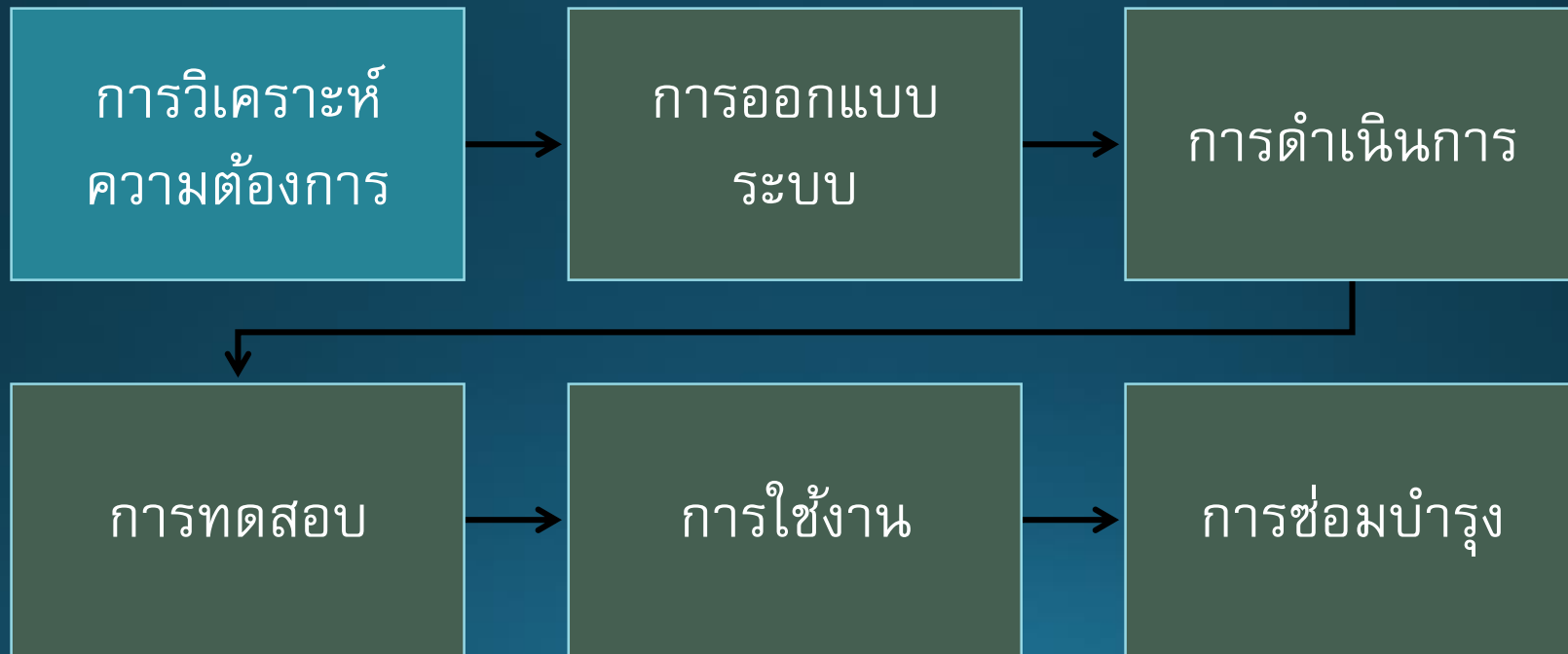
# กระบวนการพัฒนาซอฟต์แวร์โดยรวม

1. การกำหนดปัญหา คือกำหนดว่าโปรแกรมของเราจะทำไปแก้ไขปัญหาอะไร
2. การวิเคราะห์ปัญหา คือกำหนดว่าข้อมูลเข้าคืออะไร ข้อมูลออกคืออะไร
3. การออกแบบ คือกำหนดว่าจะมีไฟล์อะไรบ้าง หน้าตาของโปรแกรมเป็นอย่างไร
4. การพัฒนาระบบงาน คือเขียนโปรแกรมตามแผนที่วางไว้
5. การทดสอบ คือการตรวจสอบความถูกต้อง ค้นหาจุดผิดพลาด
6. การติดตั้ง คือการนำโปรแกรมไปใช้งานจริง
7. การบำรุงรักษา คือการนำเอาจุดด้อยมาแก้ไข คอยปรับเปลี่ยนให้ทันสมัย

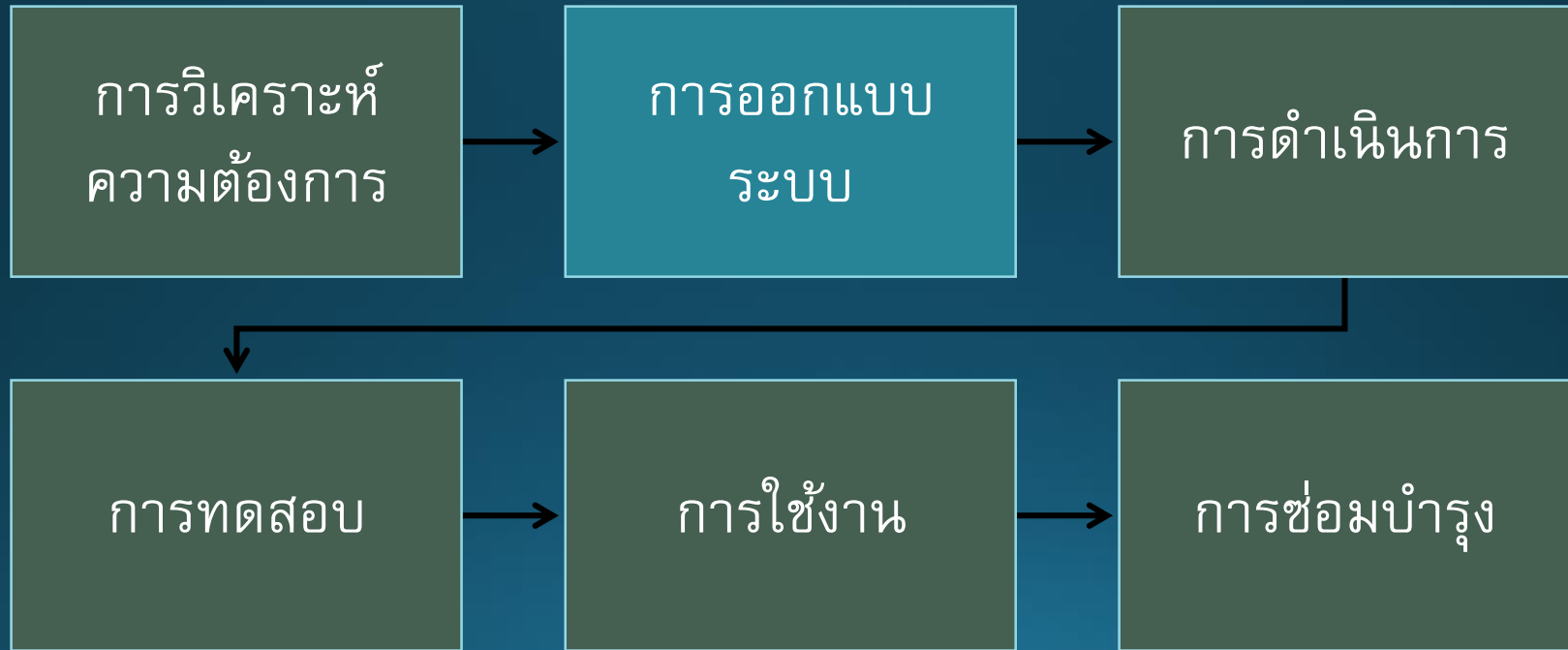
# โครงสร้างแบบน้ำตก

- โครงสร้างแบบน้ำตก หรือ The Waterfall Model เป็นวงจรชีวิตของการพัฒนาซอฟต์แวร์แรกที่น่ามาใช้ในการพัฒนาซอฟต์แวร์ ซึ่งมีลักษณะเป็นขั้นตอนลำดับ ทำให้ง่ายต่อการเข้าใจและนำไปใช้
- แต่ละขั้นตอนของกระบวนการนี้จะเสร็จสิ้นก่อนจะเริ่มขั้นตอนต่อไป ดังนั้นจะไม่มี การคาบเกี่ยวกันระหว่างขั้นตอนต่าง ๆ
- เหมาะสำหรับงานที่มีการกำหนดขอบเขตอย่างชัดเจน ทำให้มีการวางแผนที่แน่นอนและงานดำเนินไปในทิศทางที่ถูกต้อง

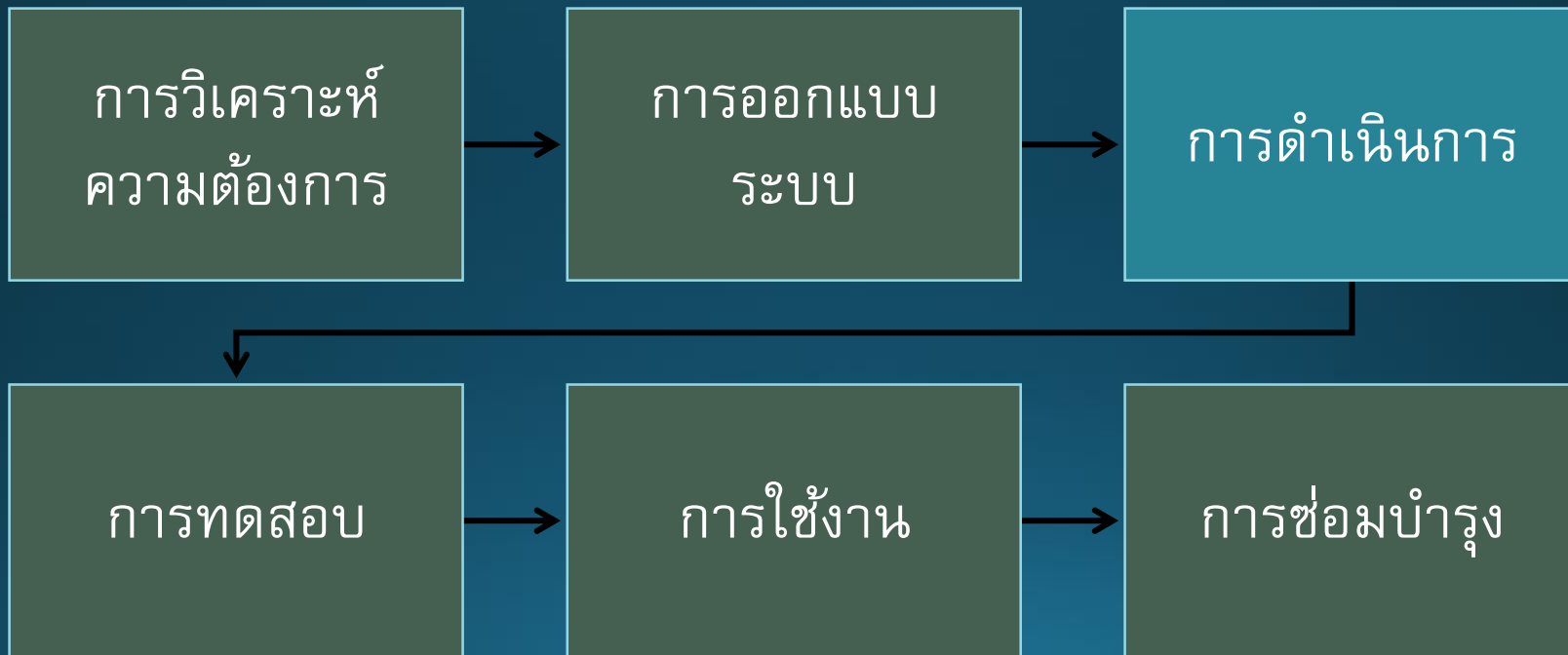
# Waterfall – Requirement Analysis



# Waterfall – System Design

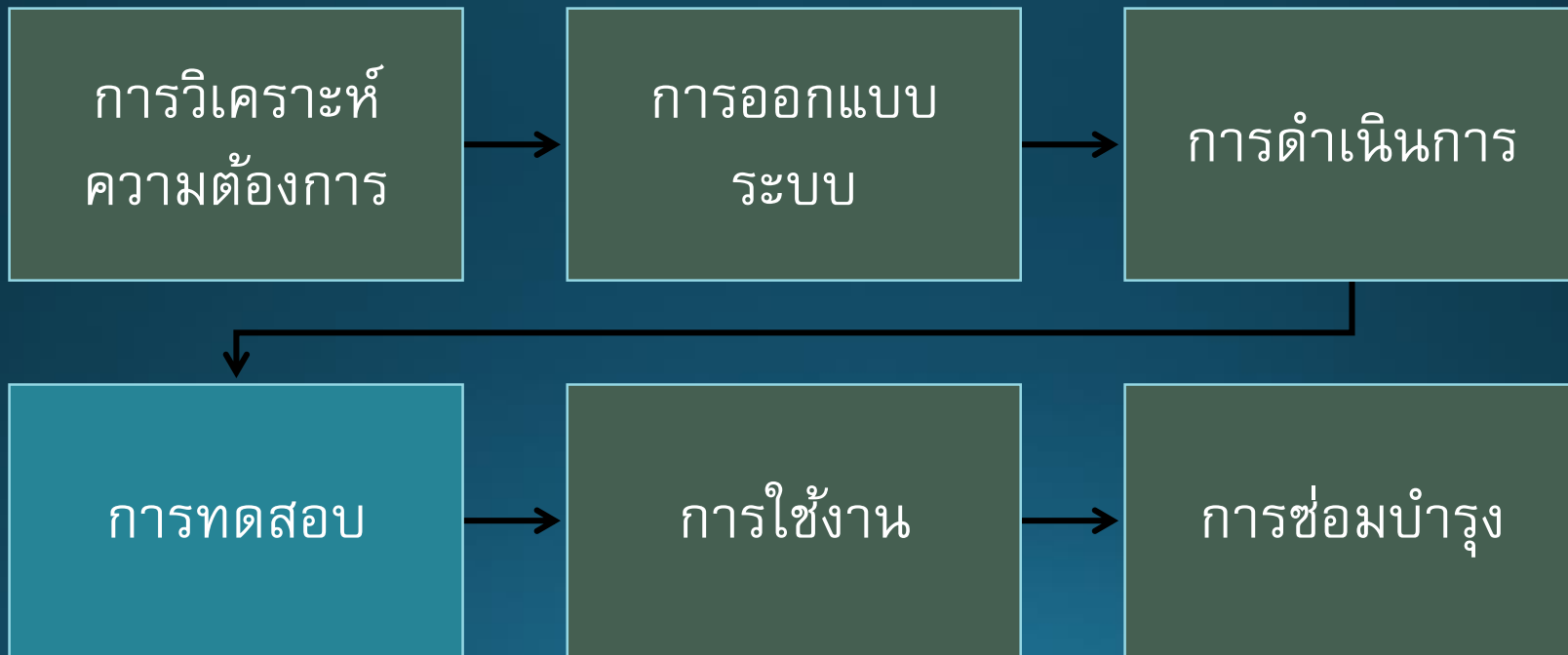


# Waterfall – Implementation

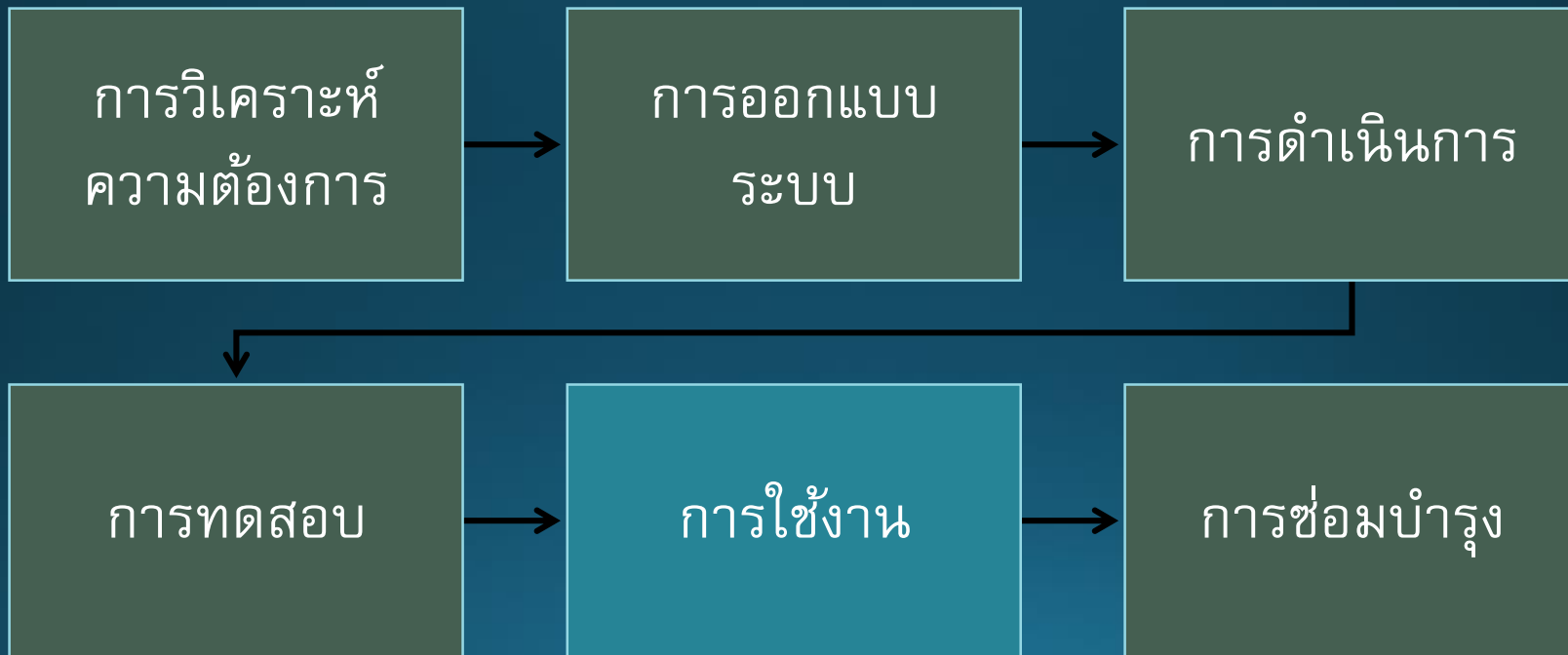




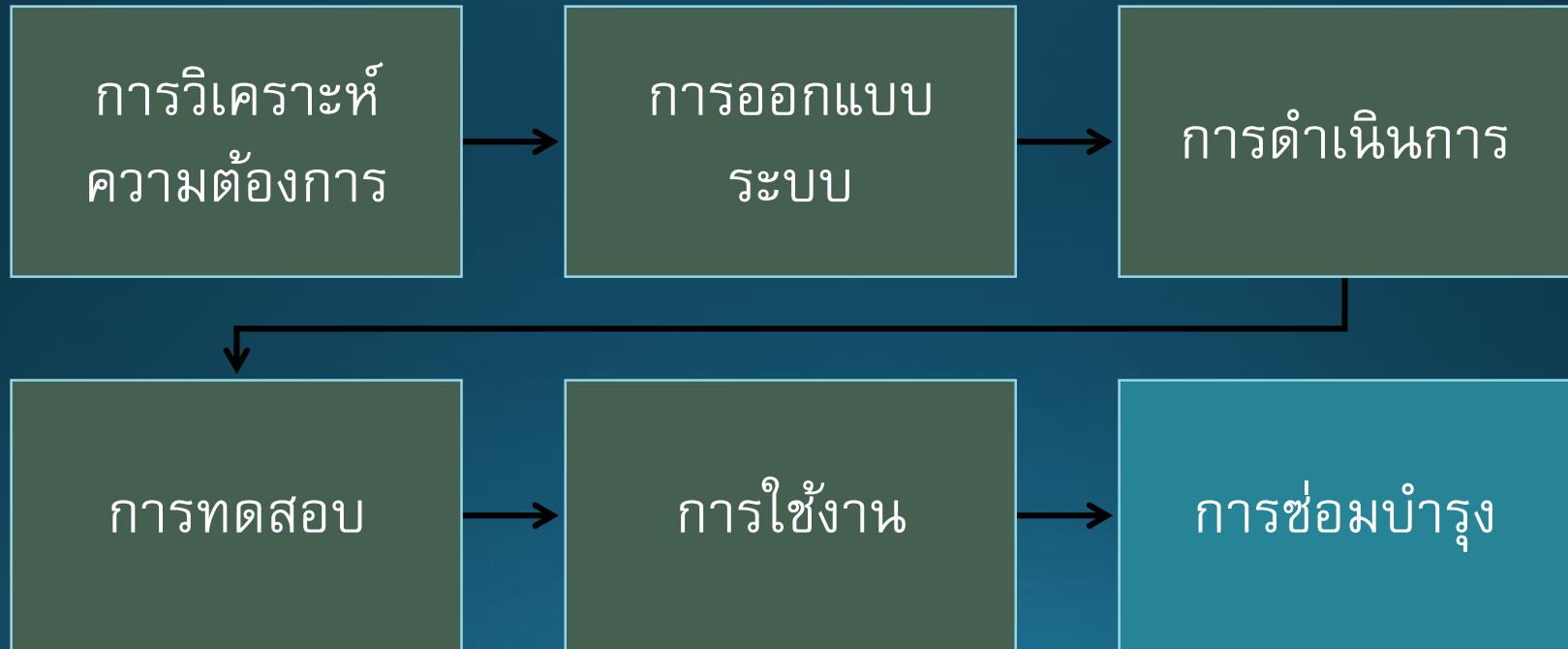
# Waterfall – Integration and Testing



# Waterfall – Deployment of System



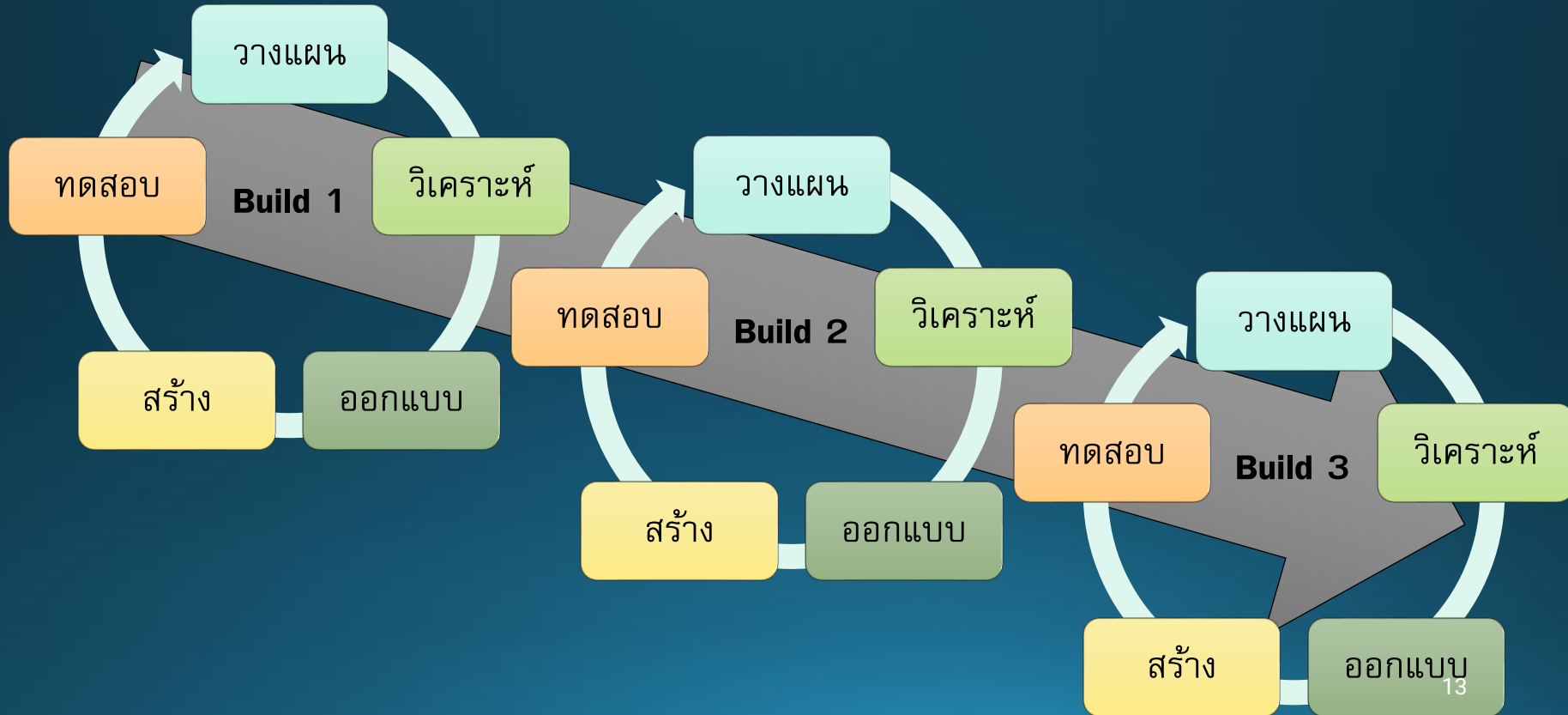
# Waterfall – Maintenance



# Agile Methods

- วิธีการพัฒนาซอฟต์แวร์แบบคล่องแคล่วว่องไว หรือ Agile Methods เป็นวงจรชีวิตของการพัฒนาซอฟต์แวร์ที่ใช้หลักการจาก Waterfall โดยทำให้กระบวนการทั้งหมดสั้นลงแล้วทำซ้ำเป็นรอบ ๆ
- Agile Methods แบ่งระบบงานออกเป็น รอบเล็ก ๆ ที่เรียกว่า Build โดยแต่ละ Build จะมีระยะเวลาประมาณ 1 – 3 อาทิตย์ และจะมีการลงความเห็นเมื่อแต่ละ Build เสร็จสิ้น
- ในแต่ละ Build จะมีการเพิ่มคุณลักษณะขึ้นมาในซอฟต์แวร์ตัวอย่าง ซึ่งคุณลักษณะนั้นอาจจะทำไปพร้อม ๆ กันได้ใน Build สุดท้ายจะมีคุณลักษณะทุกอย่างที่เราต้องการ

# Agile Methods Diagram



# หลักสี่ประการของ Agile Methods

1. **การประสานงานร่วมกัน** งานจะถูกแบ่งออกเป็น Build ย่อย ๆ ซึ่งแต่ละกลุ่มจะต้องช่วยกันทำงานและมีการประชุมร่วมกันสม่ำเสมอ
2. **ซอฟต์แวร์ตัวอย่าง** ในแต่ละ Build เราสามารถนำซอฟต์แวร์ที่ได้มาแสดงตัวอย่าง ถึงแม้ว่าจะมีคุณลักษณะไม่ครบก็ตาม ซึ่งเป็นสิ่งที่ผู้บริหารหรือลูกค้าชอบ
3. **ความร่วมมือระหว่างลูกค้า** หลังเสร็จสิ้นแต่ละ Build ลูกค้าสามารถเข้าร่วมในการแสดงตัวอย่างซอฟต์แวร์ และแสดงความคิดเห็นได้ ทำให้แก้ไขงานได้ใน Build ถัดไป
4. **การตอบสนองต่อการเปลี่ยนแปลง** ซอฟต์แวร์มีการพัฒนาอย่างต่อเนื่องและสามารถเปลี่ยนได้ตลอดเวลาเนื่องจากการตรวจสอบทุก 1 - 3 อาทิตย์

# ข้อดีและข้อเสียของ Agile Methods

## ข้อดี

- กฎระเบียบน้อย นักพัฒนาทำงานง่าย
- ส่งเสริมการทำงานเป็นทีม
- พังค์ชันมีการพัฒนาและสาธิตได้รวดเร็ว
- ใช้ทรัพยากรอย่างประหยัด
- เหมาะกับสภาพแวดล้อมที่เปลี่ยนแปลงบ่อย

## ข้อเสีย

- ไม่เหมาะกับงานที่มีการอ้างอิงที่ซับซ้อน
- เข้มงวดกับการส่งงานให้ทันตามกำหนด
- การทำงานขึ้นอยู่กับลูกค้า ถ้าลูกค้าไม่ชัดเจนอาจทำให้งานเดินไปผิดทาง
- มีการจัดทำเอกสารน้อยเนื่องจากต้องใช้เวลาพัฒนาซอฟต์แวร์เป็นส่วนใหญ่

# คำถามท้ายบทเรียน

- จงบอกความแตกต่างระหว่าง Programming และ Software Development
- จงบอกข้อดีของ The Waterfall Model
- หลักสี่ประการของ Agile Methods มีอะไรบ้าง
- เพราะเหตุใด ในปัจจุบัน Agile Methods จึงได้รับความนิยมมากกว่า The Waterfall Model
- สมมติว่าเราต้องการสร้างเกมออนไลน์บนมือถือ เราควรใช้การพัฒนาซอฟต์แวร์แบบใด
- สมมติว่าเราต้องการสร้างโปรแกรมยืมคีนหนังสือให้ห้องสมุด เราควรใช้การพัฒนาซอฟต์แวร์แบบใด