คุณภาพของซอฟต์แวร์และกระบวนการพัฒนา

คุณภาพของซอฟต์แวร์คืออะไร

- มาตรฐานคุณภาพซอฟต์แวร์นานาชาติ ISO/IEC 9126 ได้กำหนดคุณสมบัติเชิงคุณภาพไว้ดังนี้

1. ประโยชน์ใช้สอย
2. ความน่าเชื่อถือ
3. การใช้งาน
4. ประสิทธิภาพ
5. การบำรุงรักษา
6. การโอนย้ายระบบ

มาตรฐานคุณภาพซอฟต์แวร์นานาชาติ

คุณสมบัติเชิงคุณภาพ	คุณสมบัติรอง
1. ประโยชน์ใช้สอย	ตรงความต้องการ ความถูกต้อง ความต่อเนื่อง ตรงตามมาตรฐาน ความปลอดภัย
2. ความน่าเชื่อถือ	ความสมบูรณ์ ระดับปัญหที่ยอมรับได้ การฟื้นฟู
3. การใช้งาน	ความเข้าใจ การเรียนรู้ การใช้ระบบ
4. ประสิทธิภาพ	ประสิทธิภาพในการ execute โปรแกรม ประสิทธิภาพในการใช้ทรัพยากร
5. การบำรุงรักษา	การวิเคราะห์ การเปลี่ยนแปลง การทดสอบ เสถียรภาพ
6. การโอนย้ายระบบ	ความเข้ากับสิ่งแวดล้อม งานโอนย้ายระบบ ความสอดคล้องกับมาตรฐาน

คุณภาพของซอฟต์แวร์

- ประโยชน์ใช้สอย หมายถึง ซอฟต์แวร์ต้องมีประโยชน์ ตรงตามความต้องการของลูกค้า
- เช่น ซอฟต์แวร์ต้องประมวลผลออกมาถูกต้อง มีความปลอดภัย
- การทำให้ซอฟต์แวร์มีประโยชน์ใช้สอยที่ดี ต้องเริ่มจากการหาให้ได้ว่าลูกค้าต้องการอะไร

คุณภาพของซอฟต์แวร์

- ความน่าเชื่อถือ หมายถึง ลูดค้าสามารถใช้งานซอฟต์แวร์ได้อย่างสบายใจ
- โดยทั่วไป ซอฟต์แวร์ที่ผ่านการใช้งานมากเท่าไร ซอฟต์แวร์นั้นก็จะผ่านการปรับปรุงแก้ไขให้สมบูรณ์มากขึ้นเท่านั้น เพราะเมื่อใช้งานไปความผิดพลาดที่ฝังอยู่ในตอนพัฒนาซอฟต์แวร์ หรือปัญหาที่คาดไม่ถึงจะปรากฏขึ้นมา

คุณภาพของซอฟต์แวร์

- การใช้งาน หมายถึง ซอฟต์แวร์ใช้งานง่าย เข้าใจง่าย จำง่าย

- ประสิทธิภาพ มีตัววัดหลายอย่าง เช่น Throughput , Response Time , Turnaround Time ฯลฯ

คุณภาพของซอฟต์แวร์

- การบำรุงรักษา เป็นคุณสมบัติที่สำคัญมากในซอฟต์แวร์ที่ใช้วงการธุรกิจ เพราะจะถูกนำไปใช้หลายปี เมื่อมีความจำเป็นต้องปรับปรุงซอฟต์แวร์ก็สามารถวิเคราะห์การทำงานของซอฟต์แวร์นั้นแล้วนำไปปรับปรุงทดสอบได้โดยง่าย

คุณภาพของซอฟต์แวร์

- การโอนย้ายระบบ เป็นคุณสมบัติที่สำคัญเมื่อจำเป็นต้องโอนย้ายระบบตามเทคโนโลยีใหม่ เช่น การเปลี่ยนไปใช้ระบบเว็บเบส(Web-Based)
- ซอฟต์แวร์ที่ดี ควรโอนย้ายระบบง่ายโดยไม่ต้องเขียนซอฟต์แวร์ใหม่

ดัชนีวัดประสิทธิภาพ

- Throughput
- Response Time
- Turnaround Time

Throughput

- หมายถึง ปริมาณงานที่สามารถประมวลผลได้ในหนึ่งหน่วยเวลา เช่น ใน 1 นาที
- เป็นดัชนีที่เหมาะสมสำหรับงานแบบ Batch Processing(ประมวลผลเป็นชุดใหญ่ ๆ ต่อเนื่องกัน)
- เช่น คำนวณค่าบริการตอนสิ้นเดือน
- Throughput หมายถึง ใน 1 ชั่วโมง คำนวณค่าบริการลูกค้าได้กี่ราย

Response Time

- หมายถึง ระยะเวลานับตั้งแต่อินพุตข้อมูลลงไปในหน้าจอ จนถึงหน้าจอแสดงเอาต์พุตออกมา
- เป็นดัชนีที่เหมาะสมสำหรับการประมวลผลแบบออนไลน์
- เช่น เมื่อกดตู้ ATM เพื่อสอบถามยอดเงินคงเหลือในบัญชี ใช้เวลานานเท่าไรจึงเห็นยอดเงิน

Turnaround Time

- หมายถึง เวลาทั้งหมดนับตั้งแต่สั่งอินพุตจนได้เอาท์พุตออกมา ดัชนีเหมาะสมสำหรับการทำงานแบบรอบ เช่น สาขาย่อยร้องขอให้บริษัทแม่พิมพ์รายงานในสิ้นเดือน Turnaround Time จะนับเวลาทั้งหมดตั้งแต่ที่สาขาร้องขอนจนถึงได้รายงาน ซึ่งก็รวมเวลาที่บริษัทแม่รอจนถึงสิ้นเดือน เวลาในการประมวลผล และเวลาในการส่งรายงาน

ได้อย่างก็ต้องเสียอย่าง

รถยนต์ที่คุณภาพดีต้องมีคุณสมบัติหลายอย่าง เช่น ปลอดภัย อัตราเร่งดี ประหยัดน้ำมัน มีศูนย์บริการมาก กว้างขวาง นั่งสบาย

อัตราเร่งดี

ประหยัดน้ำมัน

ปลอดภัย

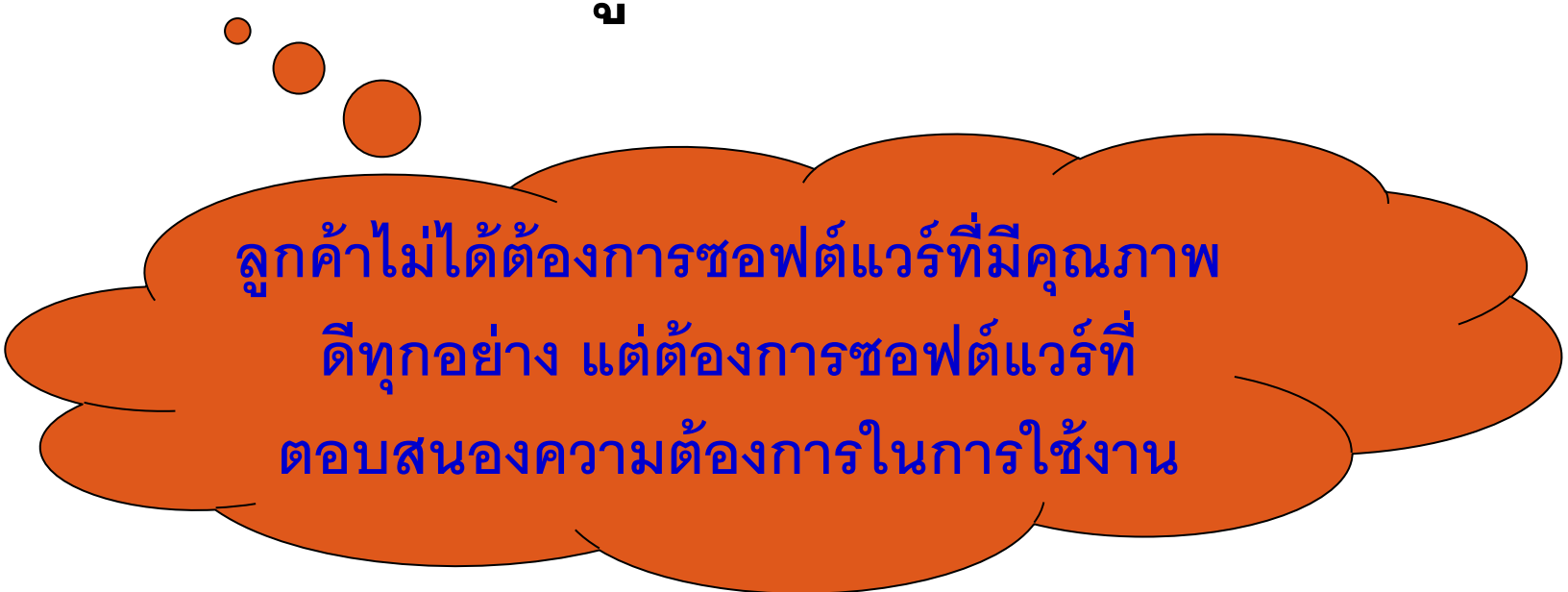
กว้างขวาง , นั่งสบาย

มีศูนย์บริการมาก

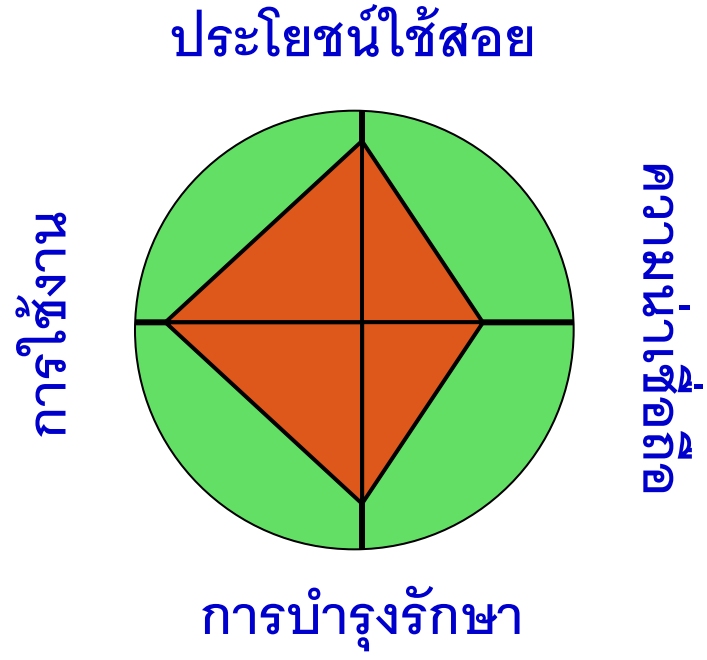
แต่ในความเป็นจริง แม้ว่าเราทุกคนอยากได้รถยนต์ที่มีคุณภาพดี แต่เราต้องเลือกรถยนต์ที่มีคุณสมบัติบางอย่าง

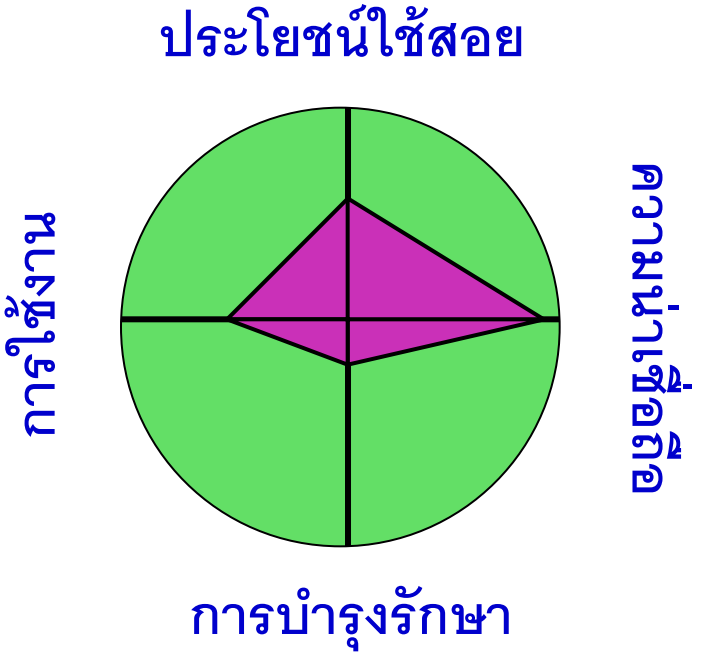
นิยามของซอฟต์แวร์ที่มีคุณภาพดี

- “คุณภาพของซอฟต์แวร์ ก็คือ ระดับการตอบสนองความต้องการของลูกค้า”
- ดังนั้นตัววัดคุณภาพของซอฟต์แวร์ที่แท้จริงคือ “ความต้องการของลูกค้า” แต่ละราย



ระดับการตอบสนองความต้องการของลูกค้า



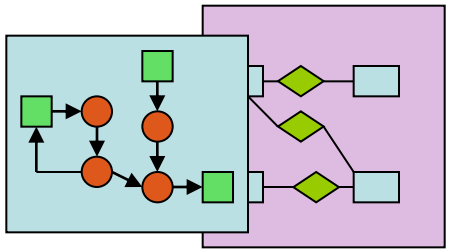


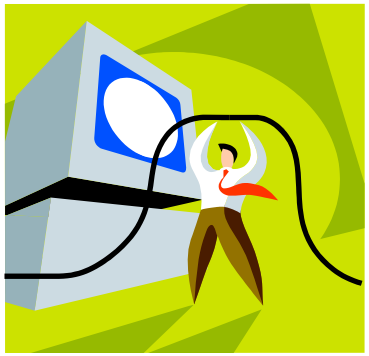
ลูกค้าต้องการแบบนี้

เราน่าจะพัฒนาแบบนี้

ข้อบกพร่องของซอฟต์แวร์เป็นสิ่งที่เห็นได้ยาก

นิยามความต้องการ → ออกแบบ → เขียนโปรแกรม → ทดสอบ





ซอฟต์แวร์ส่วนใหญ่จะไม่รู้ว่ามีข้อบกพร่อง จนกว่าจะได้ทดสอบหรือใช้งานจริง

วิธีในการทบทวนการออกแบบ

ในกระบวนการออกแบบต้องมีการทบทวนการออกแบบโดยการเขียนผลการออกแบบลงในเอกสาร แล้วให้ผู้อื่นช่วยตรวจทานให้ การทบทวนการออกแบบมี 3 วิธีใหญ่ ๆ ดังนี้

- Inspection
- Work Through
- Round Robin

Inspection

- เป้าหมายหลักคือ การค้นหาข้อบกพร่องให้พบ
- วิธีการก็คือ การทบทวนในลักษณะของการประชุม โดยมีผู้ที่เรียกว่า Modulator เป็นผู้จัดการประชุม และเพื่อให้การทบทวนมีประสิทธิผล ต้องมีการกำหนดขอบเขตที่จะทบทวน และทบทวนอย่างเข้มข้นในเวลาอันสั้น

เกณฑ์ในการทบทวนแบบ Inspection

- มี Modulator ควบคุมการประชุมแต่ละครั้ง
- มีเลขานุการบันทึกการประชุม
- ให้เวลาในการเตรียมตัวที่เหมาะสมก่อนการประชุมแต่ละครั้ง
- มีการควบคุมการบันทึกข้อบกพร่องที่ค้นพบ
- ไม่นำข้อบกพร่องที่ค้นพบไปใช้ในการประเมินผลการทำงานหรือลงโทษคนที่เกี่ยวข้อง

Inspection

- การทบทวนแบบ Inspection เป็นการทบทวนแบบทั่วไป จึงสามารถใช้ในการพัฒนาซอฟต์แวร์ใดก็ได้ และใช้ในขั้นตอนไหนก็ได้
- รวมทั้งในขั้นตอนการเขียนโปรแกรมและทดสอบโปรแกรมด้วย Inspection มาตรฐานมีประสิทธิภาพในการป้องกันข้อบกพร่องสูงมาก
- ถือเป็นเครื่องมือที่ดีที่สุดอันหนึ่งในการปรับปรุงคุณภาพ

Work Through

- หมายถึง การจินตนาการการทำงานของโปรแกรมพร้อมกับการค้นหาข้อบกพร่องโดยการดูสเปกหรือตัวโปรแกรมไปด้วย
- ซึ่งเหมือนกับ “การดีบักเป็นกลุ่มโดยการอ่านซอส์ดโคด แต่ไม่ได้รันโปรแกรมจริง”

Inspection

- โดยมาก Work Through จะเป็นการทบทวนแบบไม่เป็นทางการ โดยมีโปรแกรมเมอร์ที่รับผิดชอบเป็นผู้เรียกประชุมเมื่อมีโอกาส
- เป็นวิธีการที่เหมาะสมสำหรับการทบทวน Procedure หรือ การไหล เช่น การไหลของการควบคุม การไหลของข้อมูล และเหมาะสมกับการทบทวนชิ้นงานขั้นสุดท้ายก่อนเป็นสินค้าจริง

Round Robin

- เป็นการแบ่งงานออกเป็นส่วนย่อยๆ เท่าๆ กัน ตามจำนวนผู้เข้าร่วมทบทวน แล้วกระจายให้ผู้เข้าร่วมทบทวนแต่ละคนนำเสนอตามลำดับ แล้วช่วยกันพิจารณา
- วิธีการนี้มีประสิทธิภาพดีในแง่ของการมีส่วนร่วมของทุกคน มีการแลกเปลี่ยนความคิดเห็นซึ่งกันและกัน ทำให้ทุกคนมีความสามารถสูงขึ้น จึงนิยมใช้เพื่อการอบรมมากกว่าการทบทวนที่มีประสิทธิภาพ

ตัวอย่างเป้าหมายในการทบทวน

1	ทบทวนนิยามความต้องการ	หลงลืมความต้องการของลูกค้าหรือไม่ เข้าใจความต้องการของลูกค้าผิดหรือไม่ เน้น Why และ What แต่ยังไม่เน้น How
2	ทบทวนการออกแบบภายนอก	นำความต้องการไปเป็นฟังก์ชันได้อย่างเหมาะสมหรือไม่
3	ทบทวนการออกแบบภายใน	ตรรกะที่สร้างฟังก์ชันเหมาะสมหรือไม่ โครงสร้าโปรแกรมดีหรือไม่ แตกฟังก์ชันเหมานะสมหรือไม่
4	ทบทวนการเขียนโปรแกรม	ซอส์ดโคดเหมาะสมหรือไม่ อ่านซอส์ดโคดรู้เรื่องหรือไม่ เขียนโปรแกรมตามเกณฑ์หรือมาตรฐานหรือไม่
5	ทบทวนการทดสอบ	ขั้นตอนและเนื้อหารในการทดสอบเหมาะสมหรือไม่ กรณีที่ทดสอบครอบคลุมเพียงพอหรือไม่

กระบวนการทดสอบ

- “การทดสอบเป็นส่วนหนึ่งของการพัฒนาโปรแกรม โดยมีขึ้นเพื่อค้นหาข้อผิดพลาดของโปรแกรม”
- กระบวนการทดสอบช่วยให้เกิดความชัดเจนว่าซอฟต์แวร์จะมีคุณภาพดีหรือมีข้อบกพร่องอะไรบ้าง
- กระบวนการทดสอบจะใช้เวลามากหรือน้อยขึ้นกับว่ามีข้อบกพร่องมากน้อยเพียงไร

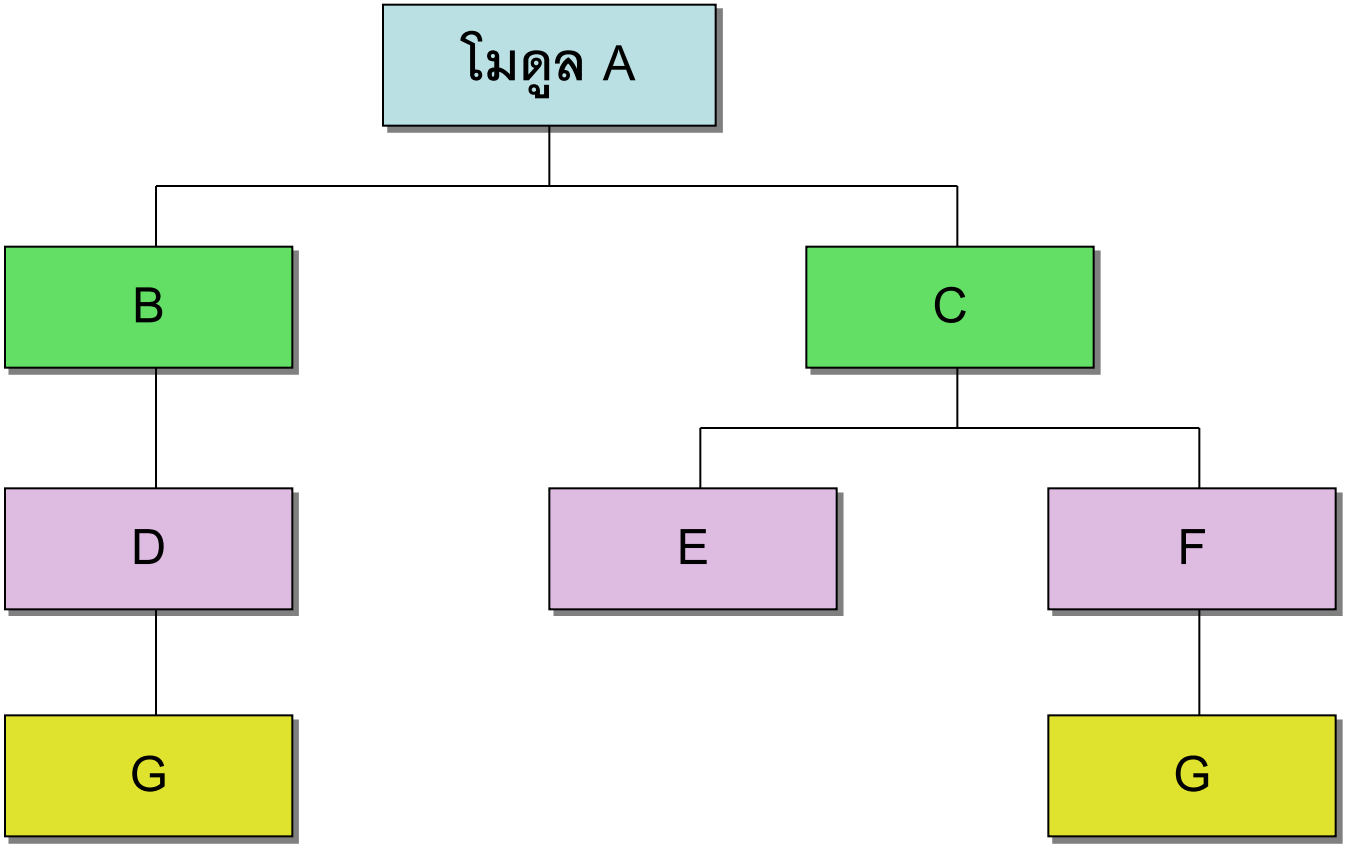
การทดสอบโมดูล(Module Test)

- เป็นการทดสอบทีละโมดูล โดยให้โมดูลนั้นทำงานอย่างอิสระ แล้วตรวจสอบตรรกะภายในโมดูล
- มีชื่อเรียกอีกอย่างหนึ่งว่า Unit Test
- ปกติโปรแกรมเมอร์จะเป็นผู้ทำหน้าที่ทดสอบ
- การทดสอบทำโดยการอ่านซอส์ดโคด สร้างกรณีทดสอบ(Test Case) แล้วลองให้โมดูลทำงานดูว่าได้ผลออกมาตามความคาดหมายหรือไม่
- ข้อผิดพลาดหลัก ๆที่ค้นพบในขั้นตอนนี้ได้แก่ ข้อผิดพลาดในการเขียนโปรแกรม เรียกว่า “บัก”

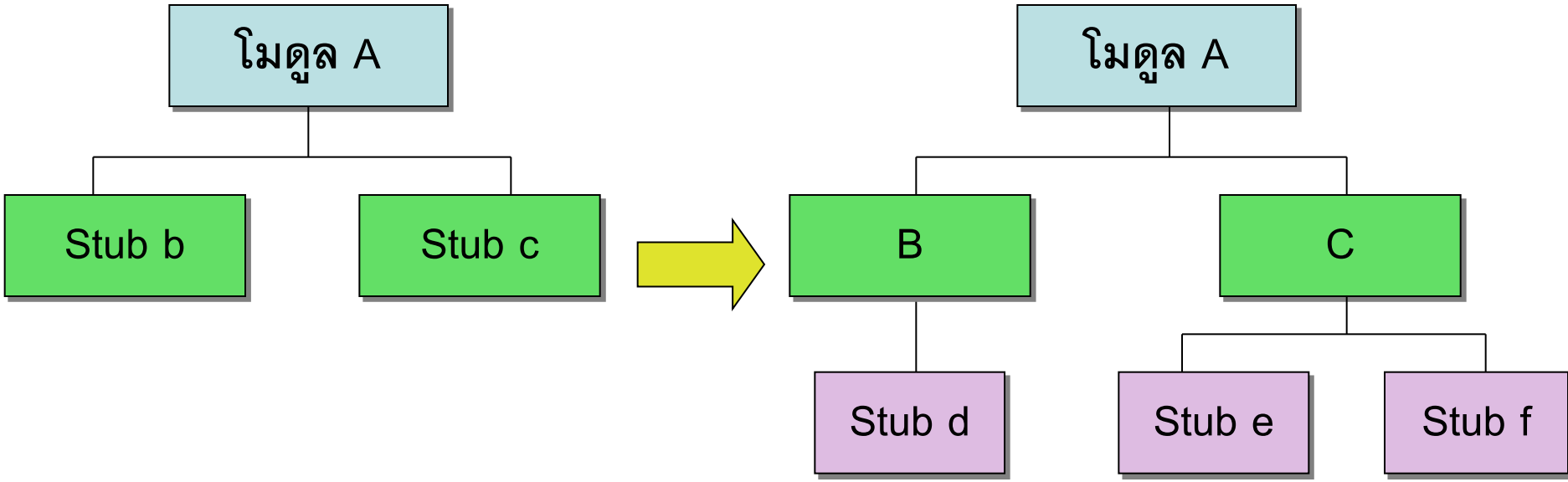
การทดสอบการเชื่อมต่อโมดูล

- เป็นการนำโมดูลมาเชื่อมต่อกัน แล้วลองให้ทำงานดู เป็นการทดสอบว่าการทำงานโดยรวมถูกต้องหรือไม่ แต่โมดูลที่นำมาเชื่อมต่อกันต้องผ่านการทดสอบโมดูลในขั้นตอนแรกก่อน
- ความผิดพลาดที่ค้นพบในขั้นตอนนี้คือ อินเทอร์เฟสที่ไม่สมบูรณ์ระหว่างโมดูล เช่น จำนวนหรือลำดับของข้อมูลที่รับส่งระหว่างโมดูลไม่ถูกต้อง ลืมตั้งค่าเริ่มต้นของตัวแปรที่จะส่งข้ามโมดูล หรือลืมคำนวณบางอย่าง เป็นต้น

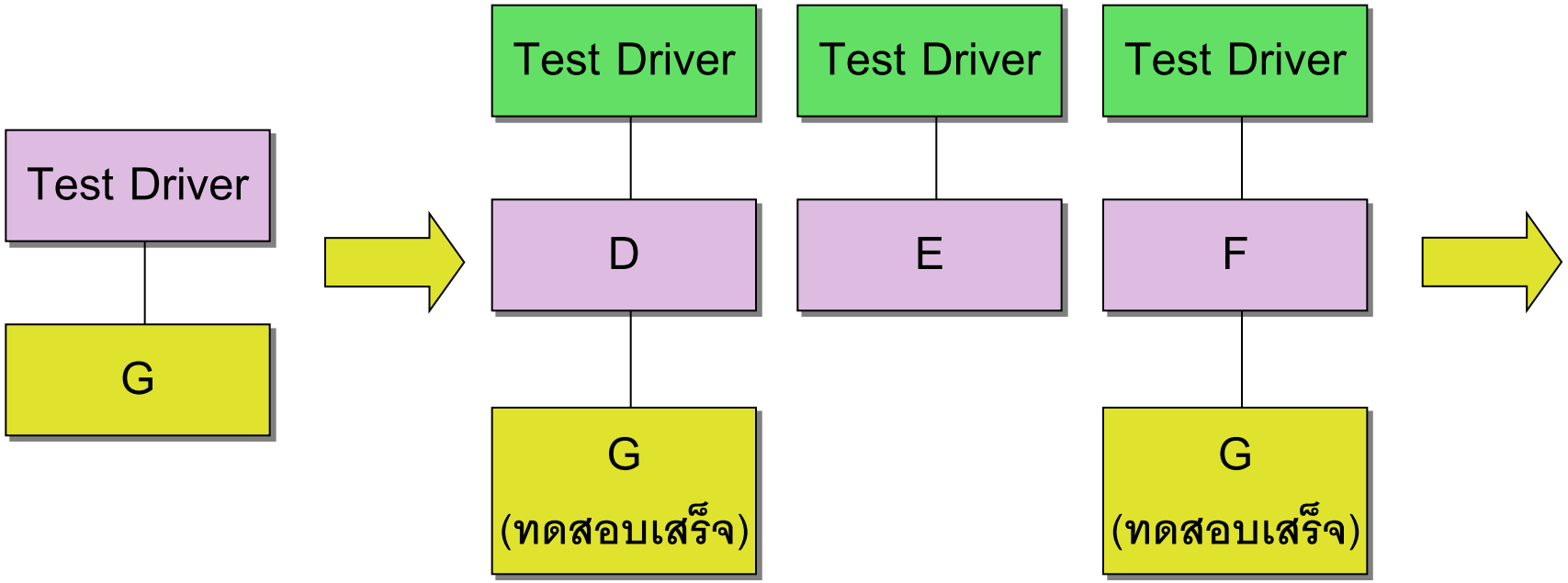
การทดสอบการเชื่อมต่อโมดูล



การทดสอบการเชื่อมต่อโมดูลแบบบนลงล่าง



การทดสอบการเชื่อมต่อโมดูลแบบล่างขึ้นบน



เปรียบเทียบการทดสอบแบบบนลงล่างและล่างขึ้นบน

การทดสอบ	ข้อดี	ข้อเสีย
แบบบนลงล่าง	•ค้นพบข้อผิดพลาดที่สำคัญได้เร็ว เช่น อินเทอร์เฟสผิดพลาด •โมดูลบนที่มีความสำคัญลูงจะผ่านการทดสอบหลายรอบ ทำให้ความน่าเชื่อถือโดยรวมสูง •ไม่ต้องมีทูล Test Driver	•ยากในการแบ่งงานให้ทำพร้อมกันหลายคน •ต้องสร้าง Stub
แบบล่างขึ้นบน	•สามารถแบ่งงานเป็นส่วน ๆแล้วทำการทดสอบพร้อม ๆกันไปได้ จึงทดสอบโมดูลจำนวนมากได้รวดเร็ว •สามารถเลือกทดสอบโมดูลที่ใช้งานร่วมก่อน เพราะมีผลกระทบมากหากเกิดข้อผิดพลาด •ไม่ต้องสร้าง Stub	•ไม่ค่อยเห็นข้อมูลผิดพลาดที่สำคัญในระยะแรก ๆของการทดสอบ •ต้อมีทูล Test Driver

การทดสอบระบบ

- เป็นการตรวจสอบประสิทธิภาพและการทำงานของระบบโดยรวม
- ต้องพยายามทดสอบภายใต้สิ่งแวดล้อมที่ใกล้เคียงกับการทำงานจริงมากที่สุด

เทคนิคการทดสอบ

- White Box Test : เป็นการทดสอบที่คำนึงถึงโครงสร้างภายในระบบ หรืออาจเรียกว่า Program-Based Test เพราะเป็นการสร้าง Test Case ตามตรรกะของโปรแกรม
- Black Box Test : เป็นการทดสอบที่ไม่คำนึงถึงโครงสร้างภายในระบบ หรืออาจเรียกว่า Specification-Based Case เพราะเป็นการสร้าง Test Case ตาม Function Specification

White Box Testing

- เป็นการทดสอบเพื่อดูโครงสร้างของโปรแกรม หรือทางเดินในโปรแกรม
- ต้องสร้างชุดทดสอบเฉพาะสำหรับการทดสอบในเงื่อนไขต่าง ๆ
- ชุดทดสอบจะต้องประกอบด้วยชุดที่สามารถประมวลผลอย่างปกติและไม่ปกติ

Black Box Testing

- เป็นการทดสอบโดยไม่คำนึงถึงคำสั่งภายในโปรแกรม
- เป็นการทดสอบ Function ต่าง ๆ ของโปรแกรมตาม Requirements ที่มี
- เป็นการทดสอบโดยดูค่า Output จาก Input ที่ให้กับโปรแกรมต้องมีความสอดคล้องกัน