

รูปแบบการพัฒนาซอฟต์แวร์



กระบวนการพัฒนาซอฟต์แวร์ (Software Process) ประกอบด้วย 4 ขั้นตอนดังนี้

1. จัดทำข้อกำหนดของซอฟต์แวร์ (Software Specification)
2. การออกแบบซอฟต์แวร์ (Software Design)
3. การทดสอบซอฟต์แวร์ (Software Validation)
4. วิวัฒนาการของซอฟต์แวร์ (Software Evolution)

จากกระบวนการดังกล่าวมีรูปแบบการพัฒนาซอฟต์แวร์ อยู่ 4 แบบคือ

1. รูปแบบน้ำตก (Waterfall model) รูปแบบการพัฒนานี้ได้ถูกนำมาใช้เป็นครั้งแรกตั้งแต่ปี ค.ศ.1970 ที่เรียกว่าแบบน้ำตก เพราะมีการส่งผลการจากขั้นตอนเริ่มต้นสู่ขั้นตอนที่อยู่ถัดๆ ไป คล้ายการไหลของน้ำที่ตกเป็นชั้นๆ นอกจากนี้แต่ละขั้นตอนยังสามารถย้อนกลับไปขั้นตอนก่อนหน้าได้ถ้าพบว่ามี ความผิดพลาดหรือไม่สมบูรณ์ รูปแบบการพัฒนาแบบน้ำตกมีขั้นตอนดังนี้

- การวิเคราะห์และกำหนดความต้องการ เป็นการวิเคราะห์ความต้องการของผู้ใช้อย่างเป็นระบบ รวมทั้งกระบวนการทำงานและข้อกำหนดต่างๆ จากนั้นก็จะนำมาสร้างเป็นตัวแบบ และข้อกำหนดต่อไป

- การออกแบบระบบและซอฟต์แวร์ คือการออกแบบระบบทั้งฮาร์ดแวร์และซอฟต์แวร์สถาปัตยกรรมของระบบ

- การพัฒนาและการทดสอบหน่วยย่อย การพัฒนาจะเริ่มจากหน่วยย่อยๆ และทำการทดสอบแต่ละหน่วยว่าถูกต้องตามข้อกำหนดหรือไม่

- การรวบรวมและการทดสอบระบบ เมื่อแต่ละหน่วยย่อยผ่านการทดสอบแล้วก็จะนำมาวมกันเป็นระบบใหญ่ และทดสอบในภาพรวมอีกครั้ง

- การทำงานและการบำรุงรักษา คือการนำระบบที่ทำการทดสอบแล้วมาใช้งานจริง และวางแผนการบำรุงรักษาระบบให้สามารถทำงานได้อย่างราบรื่น

ปัญหาของการพัฒนาแบบน้ำตก มีดังนี้

- ลำดับขั้นตอนของระบบจะตึงเครียดจนเกินไป ไม่สามารถแบ่งแยกโครงการย่อยๆ ได้
- การเปลี่ยนแปลงความต้องการของผู้ใช้จะทำได้ยาก
- รูปแบบนี้เหมาะสำหรับโครงการที่มีความเข้าใจในความต้องการของผู้ใช้เป็นอย่างดี

2. การพัฒนาแบบวิวัฒนาการ (Evolutionary Development) ไม่ได้มีการจัดทำข้อกำหนดอย่างชัดเจน แต่เป็นการพัฒนาควบคู่กันไป ใช้หลักการพัฒนาตัวอย่างก่อน แล้วจึงให้ผู้ใช้ทำการทดสอบ จากนั้นก็ผลไปทำการปรับปรุงซอฟต์แวร์ให้ถูกต้องตามความต้องการของผู้ใช้ ระบบนี้เหมาะสำหรับระบบขนาดเล็กและกลาง หรือเป็นระบบที่มีการตอบสนองหรือใช้สำหรับระบบที่มีอายุสั้น

ปัญหาของการพัฒนาแบบวิวัฒนาการ มีดังนี้

- ขาดการมองภาพรวมของระบบ
- ระบบไม่มีโครงสร้างที่แน่นอน
- ต้องใช้ความสามารถพิเศษในการพัฒนา

3. การพัฒนาระบบอย่างมีรูปแบบ (Formal Systems Development) เป็นการพัฒนาระบบโดยใช้รูปแบบทางคณิตศาสตร์ ซึ่งจะถูกลดทอนไปเป็นการวางระบบ เป็นวิธีการที่คล้ายกับรูปแบบน้ำตก แต่มีการประยุกต์ใช้หลักการทางคณิตศาสตร์เข้ามาเปลี่ยนรูปข้อกำหนดให้เป็นโปรแกรมที่ทำงานได้ ในการเปลี่ยนรูปนั้นใช้หลักการ Formal Mathematical เพื่อสร้างตัวแทนของระบบการเปลี่ยนแปลงและแก้ไข โดยใช้ตัวแบบที่สร้างขึ้น



ปัญหาของการพัฒนาแบบ Formal System Development

- จำเป็นต้องอาศัยความเชี่ยวชาญเฉพาะ และการฝึกอบรมมาก
- มีความยากในการหาตัวแบบซึ่งสอดคล้องกับระบบที่ต้องการ

4. การพัฒนาระบบแบบการนำกลับมาใช้ใหม่ (Reuse-Based Development) โดยระบบจะมีองค์ประกอบของสิ่งที่เคยมีอยู่แล้วและนำกลับมาใช้ใหม่ เช่น การนำเข้าข้อมูล การพิมพ์รายงาน เป็นต้น

ดังนั้น การนำคอมพิวเตอร์หรือซอฟต์แวร์กลับมาใช้และปรับแต่งให้เข้ากับระบบใหม่จึงเป็นวิธีการที่ทำได้อย่างรวดเร็ว ซึ่งมีขั้นตอนดังนี้

- การวิเคราะห์คอมพิวเตอร์ ทำการค้นหาคอมพิวเตอร์ที่เหมาะสมกับข้อกำหนดซึ่งอาจจะไม่ตรงกันทั้งหมด แต่ถ้ามีบางฟังก์ชันพอที่จะใช้ได้ก็สามารถนำมาประยุกต์ได้

- การตัดแปลงความต้องการจากคอมพิวเตอร์ที่ได้นำมาแก้ไข ตัดแปลง ให้ตรงกับความต้องการตามข้อกำหนด

- การออกแบบระบบแบบนำกลับมาใช้ เป็นการออกแบบกระบวนการทำงานของที่จะนำคอมพิวเตอร์มารวมกัน

- การพัฒนาและการรวมระบบ เป็นการรวมระบบทั้งหมดเข้าด้วยกัน ซึ่งในบางครั้งอาจจะต้องรวมเอาคอมพิวเตอร์ที่เป็นเชิงการค้าหรือซอฟต์แวร์ไลบรารีมาใช้ประกอบด้วยก็ได้

การพัฒนาวิธีนี้นับวันจะยิ่งมีความสำคัญมาก แต่ปัญหาคือการขาดแคลนผู้ที่มีความรู้ประสบการณ์ด้านนี้