



- 8) On-line update คุณสมบัติในการปรับปรุงข้อมูลแบบออนไลน์
- 9) Complex processing ความซับซ้อนในการประมวลผล
- 10) Usability in other applications มีความยืดหยุ่นที่จะใช้ร่วมกับแอปพลิเคชันอื่น
- 11) Installation ease ความง่ายในการติดตั้ง
- 12) Operational ease ความง่ายในการใช้งาน
- 13) Multiple sites มีการรับบริการที่งานเป็นหลายที่หรือไม่
- 14) Facilitate change สามารถปรับเปลี่ยนซอฟต์แวร์ได้ง่าย

การเคลื่อนย้ายในเครื่องเองขนาดโดยใช้ส่วนประกอบในการประมวลผลของซอฟต์แวร์ โดยถือเอาว่าส่วนประกอบของระบบคือสิ่งที่ผู้ใช้ใช้งานหรือมองเห็นได้ (เรียก User Function Types by IBM) และมีการกำหนดการวัดขนาดที่ใช้วัดดังนี้

- 1) ข้อมูลเข้าจากภายนอก (External Input)
- 2) ข้อมูลที่ส่งออกสู่ภายนอก (External Output)
- 3) ข้อมูลที่ดึงมาจากภายนอก (External Inquiries)
- 4) ข้อมูลที่ต้องการจากภายนอก (External Interface Files)
- 5) ข้อมูลเชิงตรรกะภายใน (Internal Logical Files)

เป็นเหตุทำให้มีนักศึกษาการวิเคราะห์ระบบนำเข้มาศึกษาด้วยระบบที่เป็นระบบ Batch ที่จำนวนอินพุตและเอาต์พุต (Input / Output) ที่เกี่ยวข้องเก็บฟังก์ชันพออยู่ที่สามารถนับค่าได้ หากพัฒนาระบบนี้เป็นแบบออนไลน์มีข้อมูลอินพุต/เอาต์พุตเท่าเดิมจะเกิดปัญหาขึ้น เพราะหากใช้ FPA แบบ Albrecht ค่าก็ย่อมออกมาเท่ากัน กรณีปัญหานี้บอกให้รู้ว่า FPA ยังไม่ครอบคลุมไปในเรื่องที่เกี่ยวข้องกับความสัมพันธ์ทางระบบเทคโนโลยี

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
 2.10 แนวคิดยูนิไฟแอดโพรเซส (Unified Process, UP)
 J.J. Albrecht (1979) ผู้ที่ได้อธิบายหรือผู้ (Unified Process) ได้ศึกษาวิจัย
 แล้ว คือ กระบวนการทางวิศวกรรมซอฟต์แวร์ที่เกิดจากการรวมเอาสิ่งที่ผู้เชี่ยวชาญทางการพัฒนา
 ซอฟต์แวร์ (Software engineer) ที่แยกกันมาใช้และใช้แล้วได้ผลในการพัฒนาซอฟต์แวร์นั้นมา
 รวมกัน โดยเลือกแต่เทคนิคที่ดีและขั้นตอนหลักที่เหมาะสมมา และใช้ได้อย่างมีประสิทธิภาพของ
 แต่ละผู้เชี่ยวชาญดังกล่าว รวมกัน (Unity) และกำหนดให้ชื่อใหม่ว่า “กระบวนการพัฒนา
 ซอฟต์แวร์แบบรวมกันหนึ่งเดียว (Unified Process)” บางครั้งเราจะพบว่ามีการพัฒนาระบบนี้

Copyright © by Chiang Mai University
 All rights reserved

ในแวดวงทางวิศวกรรมซอฟต์แวร์ เช่น Rational Unified Process (RUP) ซึ่งเป็นของบริษัทยักษ์ใหญ่ในวงการพัฒนาซอฟต์แวร์ที่ชื่อว่า “Rational Rose Corporation” และถูกจดลิขสิทธิ์ด้วยหลักการหรือแนวคิดจะคล้ายกันแต่จะแตกต่างที่รายละเอียดของกระบวนการมากกว่า สำหรับแนวคิดที่มีลักษณะร่วมกันหรือเหมือนกันของยูพี ได้แก่ การพัฒนาแบบวนกลับ (Iterative Development), การจัดการกับความต้องการ (Requirement Management) และการใช้เครื่องมือช่วยทางวิศวกรรมซอฟต์แวร์ (CASE Tools) เป็นต้น

เป้าหมายหรือวัตถุประสงค์ของยูพี คือ การที่ได้ซอฟต์แวร์ที่มีคุณภาพและสอดคล้องกับความต้องการของผู้ใช้ โดยอยู่ภายใต้งบประมาณและเวลาที่สามารถคาดเดาได้ ยูพีนั้นจะเน้นการกำหนดบทบาท (Role) ไปที่ทีมพัฒนางานมากกว่าแต่ละบุคคล กล่าวคือจะมีการกำหนดว่าในแต่ละช่วง (Phase) ของการพัฒนานั้นๆ ควรประกอบไปด้วยใคร (Who) แต่ละคนมีหน้าที่รับผิดชอบอะไร (What) จะทำงานที่รับผิดชอบนั้นเมื่อไหร่ (When) และปฏิบัติอย่างไร (How) ที่กล่าวมาเป็นลักษณะที่เป็นนามธรรมกระบวนการยูพี ซึ่งอาจจะทำให้เข้าใจภาพยังไม่ชัดเจน กลยุทธ์ที่ใช้ในยูพีรวมแล้วจะเรียกมันว่า “Best Practice Model” หรือ “Best Practice” ก็ได้ มีการปฏิบัติ 6 อย่าง ดังนี้

- 1) การพัฒนาซอฟต์แวร์ควรเป็นการพัฒนาแบบวนกลับ (Iterative Development)
- 2) การพัฒนาซอฟต์แวร์ใดๆ ควรมีการจัดการความต้องการได้ (Requirement Management)
- 3) การใช้แนวคิดสถาปัตยกรรมแบบองค์ประกอบ (Component-based Model Architecture)
- 4) การสร้างต้นแบบของระบบที่สามารถมองเห็นได้ (Visual Model) ด้วยภาษา UML
- 5) การตรวจสอบคุณภาพของซอฟต์แวร์ที่พัฒนาอย่างต่อเนื่อง (Continuously Verify)
- 6) การจัดการการเปลี่ยนแปลง (Change Management)