บทที่ 8
แนวความคิดและวิธีการเขียนโปรแกรมคอมพิวเตอร์

โครงร่างของบทนี้

8.7.3 การตรวจสอบ (Structured Walkthroughs)

8.7.4 การทดสอบ (Testing)

8.8 เอกสารโปรแกรม (Program Documentation)

8.9 การบำรุงรักษาโปรแกรม (Program Maintenance)

วัตถุประสงค์

เพื่อให้นักศึกษาเข้าใจพื้นฐานของการเขียนโปรแกรม ซึ่งวิเคราะห์จาก

(1) วิธีการพัฒนาการเขียนโปรแกรม

(2) กิจกรรมของระยะต่างๆ ในการเขียนโปรแกรม

(3) แนวความคิดและการประยุกต์ในการเขียนโปรแกรมโครงสร้าง

(4) การสร้างและการใช้เครื่องมือในการออกแบบโปรแกรม

หลังจากที่นักศึกษาได้อ่านและเรียนรู้จากบทเรียนนี้แล้ว นักศึกษาควรมีความสามารถดังต่อไปนี้

1. อธิบายได้ว่าทำไมผู้ใช้คอมพิวเตอร์ ควรมีความรู้พื้นฐานเกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์
2. กำหนดความเปลี่ยนแปลงหลักๆ ที่ใช้ในการเขียนโปรแกรม และอธิบายถึงผลที่เกิดขึ้นต่อผู้ใช้ และผู้เขียนโปรแกรมได้
3. สรุปหน้าที่แต่ละขั้นตอนของการเขียนโปรแกรมคอมพิวเตอร์
4. อธิบายถึงวิธีการต่างๆ ในการเขียนโปรแกรม ที่มีผลต่อการออกแบบ การเขียนคำสั่ง และการตรวจสอบความผิดพลาด ของโปรแกรม
5. อธิบายถึงวัตถุประสงค์ของการใช้เครื่องมือในการเขียนโปรแกรมต่างๆ เช่น structure and HIPO charts, layout forms, flowcharts, pseudocode และ decision table
6. สามารถเขียนผังงาน (flowcharts) และระบบพื้นฐานได้
7. สามารถกำหนด คำสั่งพื้นฐานชนิดต่างๆ เพื่อสั่งให้เครื่องปฏิบัติงานได้
8. สามารถบอกข้อผิดพลาดชนิดต่างๆ ที่อาจเกิดขึ้นในการเขียนโปรแกรม รวมทั้งสามารถอธิบายถึงการตรวจสอบและทดสอบ กิจกรรมต่างๆ ในการเขียนโปรแกรมได้
9. อธิบายถึงวัตถุประสงค์ในการจัดทำเอกสารโปรแกรมและการบำรุงรักษาได้

8.1 ทำไมต้องเรียนการเขียนโปรแกรมคอมพิวเตอร์

ความเข้าใจพื้นฐานของระบบคอมพิวเตอร์ รวมทั้งฮาร์ดแวร์ และซอฟต์แวร์ ในบทที่ผ่านๆ มา นั้นเป็นสิ่งที่สำคัญ แต่ยังไม่เพียงพอ สิ่งที่ควรเข้าใจ ซึ่งมีความสำคัญไม่แพ้กัน คือ ความเข้าใจพื้นฐานเกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์ ซึ่งคือ ขั้นตอนต่างๆ ที่เราต้องการให้คอมพิวเตอร์ปฏิบัติตาม

อย่างไรก็ตาม ถึงแม้ว่า ความจริงในปัจจุบันนี้ คนส่วนมากที่ใช้คอมพิวเตอร์ แน่นอนเลยว่า จะไม่พัฒนาโปรแกรมขึ้นมาใช้เอง จะใช้โปรแกรมสำเร็จรูปต่างๆ ที่มีขายตามท้องตลาด หรือใช้โปรแกรมที่พัฒนาจากกลุ่มของโปรแกรมเมอร์ในหน่วยงาน หรือจากบริษัทที่รับจ้างพัฒนาโปรแกรม แล้วทำไมจึงต้องเรียนการเขียนโปรแกรมด้วย เหตุผลก็คือ

(1) เพื่อให้สามารถติดต่อสื่อสารกับโปรแกรมเมอร์ ให้เข้าใจถึงปัญหาต่างๆ ที่เกิดขึ้นในธุรกิจ และวิธีการเพื่อแก้ไขปัญหาต่างๆ เหล่านั้น

(2) ผู้ใช้คอมพิวเตอร์ส่วนมากที่ใช้เครื่องไมโครคอมพิวเตอร์ หรือเทอร์มินัล ก็ตาม อาจต้องเขียนโปรแกรมง่ายๆ เพื่อทำงานพื้นฐานบนเครื่องคอมพิวเตอร์ของตนเอง

(3) ผู้ใช้คอมพิวเตอร์อาจพัฒนาโปรแกรมโดยใช้ nonprocedural language ซึ่งเป็นภาษาซึ่งผู้ใช้คอมพิวเตอร์สามารถสั่งให้คอมพิวเตอร์ปฏิบัติงานได้ โดยง่าย เป็นภาษาธรรมชาติ ที่ผู้ใช้สามารถโต้ตอบกับคอมพิวเตอร์ โดยไม่ต้องเขียนคำสั่งประมวลผลข้อมูลโดยละเอียด

ความรู้พื้นฐานในการเขียนโปรแกรมและซอฟต์แวร์คอมพิวเตอร์ ของผู้ใช้คอมพิวเตอร์ที่ควรเรียนมีดังนี้คือ

(1) พื้นฐานของวิธีการเขียนโปรแกรมคอมพิวเตอร์ (Computer Programming)

(2) คุณลักษณะเบื้องต้นของโปรแกรมภาษาที่นิยมใช้ในปัจจุบัน (programming language)

(3) การพัฒนาโปรแกรมเบื้องต้นโดยใช้โปรแกรมภาษาระดับสูง (high-level programming language)

(4) การใช้โปรแกรมสำเร็จรูปที่สำคัญชนิดต่างๆ (software packages)

8.2 ความเปลี่ยนแปลงในการเขียนโปรแกรมคอมพิวเตอร์ (Changes in Computer Prgramming)

การพัฒนาโปรแกรมคอมพิวเตอร์เพื่อนำมาใช้งานนั้น จะสังเกตว่าแต่ละโปรแกรมที่พัฒนาขึ้นมา มีบางโปรแกรมเท่านั้นที่ได้รับความนิยม บางโปรแกรมก็ยากที่ผู้ใช้จะใช้งานได้สะดวก ยากต่อการแก้ไข เปลี่ยนแปลง ทดสอบ และบำรุงรักษา และมีไม่น้อยที่การพัฒนาโปรแกรม

ประสบผลล้มเหลว ไม่สามารถพัฒนาได้สำเร็จหรือล่าช้าเกินกว่าที่กำหนด ซึ่งปัญหาต่างๆ อาจจำแนกได้ดังนี้คือ

(1) Programmer productivity

เป็นปัญหาในกรณีที่โปรแกรมที่พัฒนาไม่สามารถเสร็จได้ทันตรงตามกำหนดเวลา ซึ่งอาจเกิดจากไม่สามารถควบคุมเวลาในการทำงาน และวัดความก้าวหน้าของโปรแกรมได้

(2) Programming quality

โปรแกรมที่พัฒนาขึ้นมา เป็นโปรแกรมที่มีคุณภาพไม่ดี อันเนื่องมาจากมีข้อผิดพลาด (error) มาก หรือ ไม่สามารถทำงานตามความต้องการของผู้ใช้ได้ หรือโปรแกรมมีความซับซ้อนมาก ยากต่อการทดสอบ การเปลี่ยนแปลง การบำรุงรักษา และเอกสารของโปรแกรมไม่ดีพอด้วย

(3) Programming cost

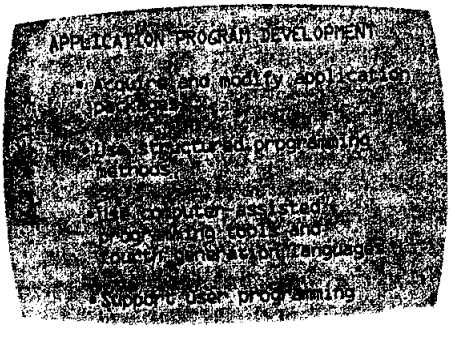
ค่าใช้จ่ายในการพัฒนาโปรแกรม การทดสอบ การบำรุงรักษา และการทำให้ถูกต้อง สูงเพิ่มขึ้นเรื่อยๆ

สำหรับผู้ใช้คอมพิวเตอร์ โดยเฉพาะเจ้าของบริษัท หรือหน่วยงาน แน่นอนว่าต้องการใช้โปรแกรมที่ดี มีคุณภาพ และราคาถูกที่สุด ดังนั้น จึงมีการพัฒนาการเขียนโปรแกรมวิธีใหม่ที่จะแก้ปัญหาในการพัฒนาโปรแกรม ซึ่งวิธีการเขียนแนวใหม่ ประกอบด้วย

(1) structured programming

(2) computer-assisted programming

(3) user programming



รูปที่ 8-1 แสดงรายละเอียดการเขียนโปรแกรมแนวใหม่

8.2.1 โปรแกรมโครงสร้าง (Structured Programming)

เป็นวิธีการเขียนโปรแกรมที่เป็นส่วนหนึ่งของ วิศวกรรมซอฟต์แวร์ (software engineering) ซึ่งคือการจัดการวิธีการพัฒนาซอฟต์แวร์ โดยออกแบบและพัฒนาซอฟต์แวร์ที่มีคุณภาพ โดยเกี่ยวข้องกับ คน เครื่องมือ และวิธีปฏิบัติ โดยใช้วิธีการในการจัดการสมัยใหม่

โปรแกรมโครงสร้าง เกี่ยวข้องกับวิธีการในการเขียนโปรแกรมโดยออกแบบเป็นลักษณะจากบนลงล่าง (top-down) โดยจำกัดจำนวนของโครงสร้างควบคุม (control structure) ไม่ให้มีมากเกินไป เพราะจะทำให้โปรแกรมซับซ้อน นอกจากนี้ การเขียนโปรแกรมที่ดีควรแบ่งโปรแกรมออกเป็นโมดูล เพื่อง่ายต่อการทดสอบ เปลี่ยนแปลง และแก้ไขได้ง่าย และควรมีหัวหน้าทีมในการเขียนโปรแกรมเพื่อควบคุมการเขียนโปรแกรมให้มีมาตรฐานเดียวกัน สรุปได้ว่า การเขียนโปรแกรมโครงสร้าง ควรที่จะมีคุณลักษณะต่อไปนี้

(1) top-down design

(2) modularity

(3) stepwise reefinement

(4) chief programmer team

เครื่องมือ (tool) ที่ช่วยในการพัฒนาโปรแกรม ได้แก่

(1) structure and HIPO chart

(2) structured coding

(3) pseudocode

(4) structured walkthroughs

ในปัจจุบันการเขียนโปรแกรมโครงสร้างเป็นที่นิยม เนื่องจากช่วยลดค่าใช้จ่ายในการพัฒนาและบำรุงรักษาโปรแกรมคอมพิวเตอร์ โปรแกรมที่พัฒนาขึ้นจะมีโครงสร้างที่ดีและเป็นโปรแกรมที่เป็นมาตรฐาน ซึ่งทำให้ง่าย (simplicity) ต่อการพัฒนาแก้ไข รวมทั้งมีความถูกต้อง (accuracy) สูง ซึ่งวิธีการนี้เองทำให้โปรแกรมเมอร์สามารถพัฒนาโปรแกรมให้มีประสิทธิภาพสูง ลดความซับซ้อนของโปรแกรม ซึ่งส่งผลให้

(1) Programming productivity ผลผลิตในการเขียนคำสั่งโปรแกรมของโปรแกรมเมอร์มากขึ้น และมีข้อผิดพลาดน้อย

(2) Programming economy ค่าใช้จ่ายและเวลาในการพัฒนาโปรแกรม และบำรุงรักษา (maintenance) ลดลง

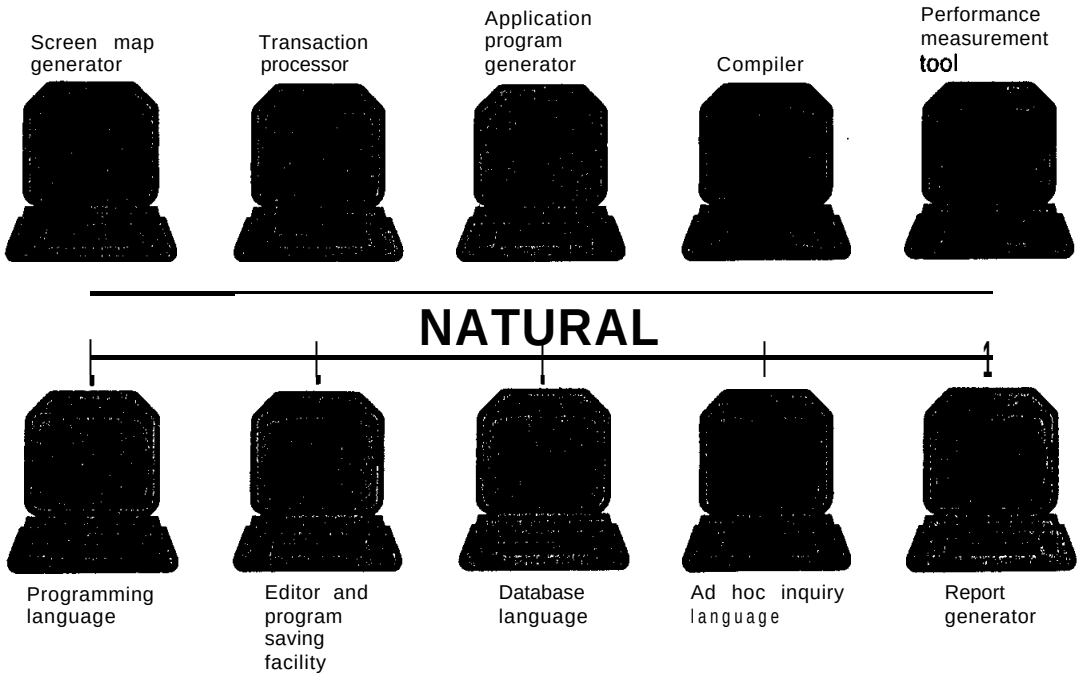
(3) Programming simplicity หมายถึง โปรแกรมที่พัฒนาขึ้น อ่าน เขียน แก้ไข บำรุงรักษาง่าย

8.2.2 โปรแกรมช่วยเหลือ (Computer-Assisted Programmingg)

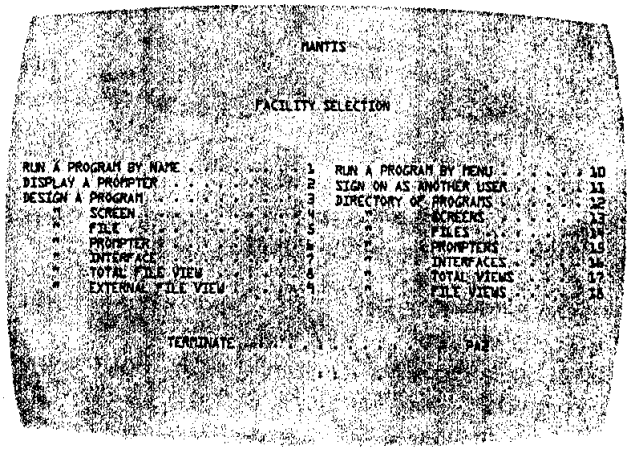
นอกจากการเขียนโปรแกรมโครงสร้างที่ช่วยเพิ่มผลผลิต เพิ่มคุณภาพและลดค่าใช้จ่ายแล้ว ยังมีอีกวิธีหนึ่งซึ่งเป็นที่นิยมคือ รูปแบบของ automate เป็นการเขียนโปรแกรมที่มีการปฏิบัติการโต้ตอบโดยอัตโนมัติ โดยผู้ใช้คอมพิวเตอร์ หรือโปรแกรมเมอร์ สามารถ ออกแบบ ซึ่งอาจเป็นหน้าจอป้อนข้อมูล หน้าจอแสดงผลลัพธ์ หรือ คำสั่งประมวลผลทางตรก โดยใช้ ส่วนช่วยเหลือ ซึ่งมีให้ในระบบคอมพิวเตอร์ นอกจากนี้ ยังสามารถช่วยโปรแกรมเมอร์ในการเขียนคำสั่ง แปล ทดสอบ ตรวจแก้ไขจุดบกพร่องของโปรแกรม อีกด้วย ซึ่งถือได้ว่า โปรแกรมช่วยเหลือเป็นเครื่องมือซอฟต์แวร์ (software tool) สำหรับ ระบบพัฒนางานประยุกต์ (application development systems or application generators) ซึ่งเตรียมส่วนช่วยเหลือต่างๆ ให้กับโปรแกรมเมอร์ รวมทั้ง menus, prompts และกราฟฟิค ซึ่งทำให้วิธีการเขียนโปรแกรมง่ายและเป็นไปโดยอัตโนมัติ

ระบบพัฒนางานประยุกต์ (Application Development System หรือ ADS) ที่นิยมในปัจจุบัน เช่น Database Management System (DBMS) เป็นซอฟต์แวร์ที่ช่วยโปรแกรมเมอร์ ในการสร้างฐานข้อมูล บำรุงรักษาฐานข้อมูล การดึงข้อมูลจากฐานข้อมูล โดยง่าย ซึ่ง ซอฟต์แวร์นี้จะประกอบด้วยโปรแกรมต่างๆ ซึ่งเรียกว่า programming tool เป็นเครื่องมือในการเขียนโปรแกรม ซึ่งสนับสนุนผู้เขียนโปรแกรม ในส่วนของ การออกแบบทางตรรก การแก้ไข การเขียนคำสั่ง การทดสอบ การตรวจแก้ไขจุดบกพร่องของโปรแกรม และการบำรุงรักษา

NATURAL BY SOFTWARE AG



MANTIS BY CINCOM



รูป 8-2 เป็นเครื่องมือต่างๆ ที่ช่วยในการพัฒนางานประยุกต์

นอกจากนี้โปรแกรมช่วยเหลือที่สำคัญอีกโปรแกรมหนึ่ง คือ fourth-generation languages (4GL) หรือ nonprocedural languages เป็นการเขียนโปรแกรมภาษาชนิดใหม่ ที่อนุญาตให้ผู้ใช้และโปรแกรมเมอร์ ระบุถึงผลลัพธ์ ที่ต้องการ โปรแกรมจะกำหนด ลำดับของคำสั่งต่างๆ ในการปฏิบัติงานได้ จนได้ผลลัพธ์ที่ต้องการ ซึ่งแตกต่างจากการเขียนโปรแกรมด้วยภาษาระดับสูงอื่นๆ ซึ่ง ความต้องการของผู้ใช้จะถูกพัฒนาเป็นลำดับของคำสั่ง ซึ่งคอมพิวเตอร์ต้องปฏิบัติตามจนกระทั่งได้ผลลัพธ์ 4GL นี้ช่วยการประมวลผลโปรแกรมง่ายและเร็วขึ้น ส่วนมากเป็นโปรแกรมที่สนับสนุน ของ DBMS และโปรแกรมสำเร็จรูป ADS ซึ่งช่วยการประมวลผลฐานข้อมูลให้ง่ายขึ้น

ในการพัฒนาโปรแกรมนั้น ส่วนมากทำงานเป็นกลุ่ม มีผู้ร่วมงานในทีมงานหลายคน การมี development centers หรือศูนย์กลางการพัฒนาถือว่าเป็นการช่วยเหลือการพัฒนาโปรแกรมวิธีหนึ่ง เพราะเป็นศูนย์กลางของหน่วยงานที่สนับสนุนกลุ่มของโปรแกรมเมอร์ ในการกำหนดมาตรฐานในการเขียนโปรแกรมให้มีคุณภาพ รวมทั้งให้บริการ และเป็นที่ปรึกษาการพัฒนางานประยุกต์โดยผู้ที่รับผิดชอบ ศูนย์กลางการพัฒนาต้องเป็นบุคคลที่มีความรู้ มีความชำนาญ และประสบการณ์สูง นอกจากนี้ บุคลากรเหล่านี้ทำหน้าที่วิเคราะห์ถึงคุณภาพและผลผลิตของการเขียนโปรแกรม แนะนำวิธีการ หรือทรัพยากร และช่วยสร้างโปรแกรมที่ดีอีกด้วย

โปรแกรมช่วยเหลือที่กล่าวมาทั้งหมดนี้ ช่วยสนับสนุนผู้ใช้และโปรแกรมเมอร์ สรุปได้ดังนี้คือ

(1) Hardware support โปรแกรมสนับสนุนผู้ใช้ เช่น intelligent terminal ให้มีความสามารถเพิ่มขึ้นในการแสดงผล และกราฟฟิค

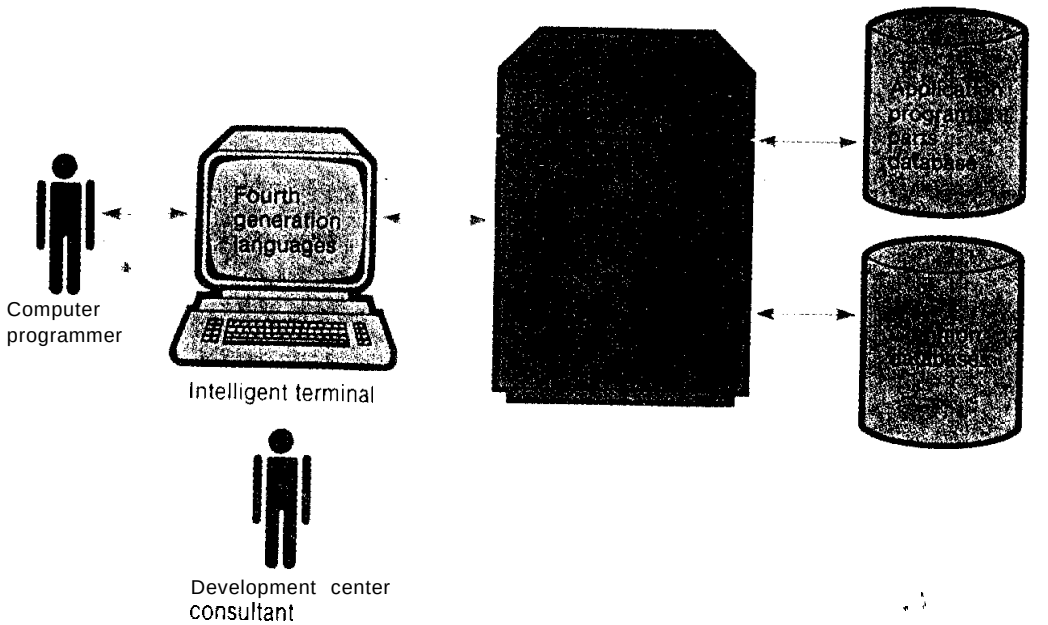
(2) Programming tools เป็นโปรแกรมที่ใช้เป็นเครื่องมือในระบบพัฒนางานประยุกต์โดยช่วยในการพัฒนาโปรแกรม

(3) Fourth-generation languages เป็นภาษาคอมพิวเตอร์ที่เป็น nonprocedural programming ที่ช่วยให้การพัฒนาโปรแกรมง่ายและเร็วขึ้น

(4) Application program parts เป็นส่วนของโปรแกรมที่เก็บในโปรแกรมประยุกต์ฐานข้อมูล

(5) Database management systems เป็นระบบจัดการฐานข้อมูลที่ใช้ในระบบ realtime ซึ่งใช้ฐานข้อมูลร่วมกัน

(6) Development centers เป็นองค์กรพิเศษ ซึ่งสนับสนุนการเขียนโปรแกรม ซึ่งช่วยให้คุณภาพและผลผลิต ดีขึ้น



รูปที่ 8-3 โปรแกรมช่วยเหลือซึ่งสนับสนุนระบบคอมพิวเตอร์

การเขียนโปรแกรมในปัจจุบันนี้ ส่วนมากจะใช้โปรแกรมช่วยเหลือ ช่วยในการพัฒนาโปรแกรม ซึ่งจะแทนที่การพัฒนาโปรแกรมในรูปแบบเก่าๆ อย่างไรก็ตาม ผู้ใช้คอมพิวเตอร์ต้องมีความรู้ในกิจกรรมหลักๆ ในการที่จะพัฒนาโปรแกรมคอมพิวเตอร์ให้สำเร็จไม่ว่าจะเป็นวิธีใดก็ตาม

8.2.3 ผู้ใช้ (User)

ในการพัฒนาโปรแกรมให้มีคุณภาพดี ผลผลิตสูง และเสียค่าใช้จ่ายต่ำนั้น นอกจากการพัฒนาการเขียนโปรแกรมของโปรแกรมเมอร์ ในการใช้เทคนิคใหม่ๆ แล้วนั้น ผู้ใช้โปรแกรม (User) ก็เป็นปัจจัยที่สำคัญหนึ่งที่จะสนับสนุน ช่วยเหลือ และแนะนำ ในการพัฒนาระบบ ในแง่ต่างๆ ได้ เช่น

(1) Hardware support

ระบบการทำงานในปัจจุบัน มีการติดต่อสื่อสารข้อมูลผ่านระบบเครือข่าย (network) ผู้

ใช้โปรแกรมต้องทำหน้าที่เตรียมข้อมูล ซึ่งอาจเป็นการป้อนข้อมูลต่างๆ บนเครื่อง intelligent workstation หรือเครื่องไมโครคอมพิวเตอร์ เพื่อส่งข้อมูลไปยังฐานข้อมูลส่วนกลาง เพื่อประมวลผลข้อมูล และในทำนองเดียวกัน ข้อมูลจากฐานข้อมูลส่วนกลาง อาจถูกดึงไปใช้ได้โดยผู้ใช้อื่นๆ ในระบบ ซึ่งเป็นการช่วยเหลือการทำงานของโปรแกรม

(2) Software support

ระบบซอฟต์แวร์สนับสนุนกลุ่มโปรแกรมเมอร์ โดยจัดหาเครื่องมือการเขียนโปรแกรม เช่น ADS, fourth-generation nonprocedural language, database management system query, report generator language, โปรแกรมสำเร็จรูปอื่นๆ เช่น graphic, integrated package เท่าที่จัดหาได้ เพื่อสนับสนุนการเขียนโปรแกรม

(3) Organizational support

ผู้ใช้โปรแกรมสามารถช่วยเหลือหน่วยงานโดยจัดหาข้อมูลใหม่ๆ ให้แก่ศูนย์กลางข้อมูล (information center) ของหน่วยงาน ซึ่งอาจเป็นข้อมูลการพัฒนาโปรแกรมประยุกต์ใหม่ๆ เพื่อช่วยแก้ปัญหาในการทำงาน

โดยปกติแล้ว ศูนย์กลางข้อมูล (information center) ของหน่วยงาน นี้จะสนับสนุน และช่วยเหลือ ผู้ใช้ระบบของหน่วยงาน โดยเตรียมฮาร์ดแวร์ให้กับผู้ใช้ระบบ เช่น microcomputer, intelligent terminals advanced graphic terminals, high-speed printers, poltter เป็นต้น ในส่วนของซอฟต์แวร์ จัดเตรียมโปรแกรมสำเร็จรูปต่างๆ รวมทั้งโปรแกรมช่วยเหลือต่างๆ เช่น DBMS, nonprocedural languages เป็นต้น นอกจากนี้ ยังช่วยเหลือบุคลากร โดยจัดเตรียมกลุ่มบุคคลเพื่อให้เป็นที่ปรึกษาสำหรับผู้ใช้ระบบ ซึ่งกลุ่มบุคคลเหล่านี้ อาจเป็นผู้วิเคราะห์ระบบ หรือโปรแกรมเมอร์ ซึ่งฝึกสอนให้ผู้ใช้ระบบสามารถทำงานในระบบได้อย่างมีประสิทธิภาพ

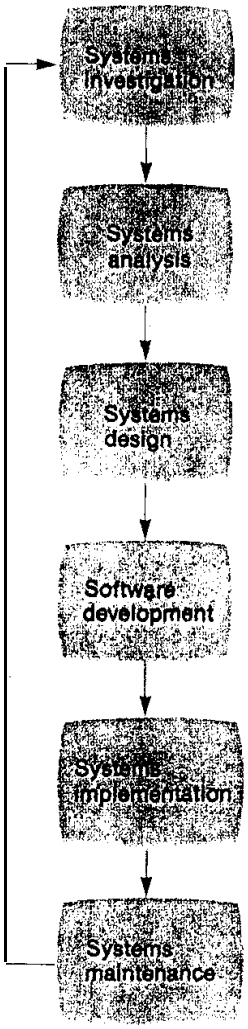
8.3 วิธีการเขียนโปรแกรม (The programming process)

โปรแกรมต่างๆ ที่ได้ถูกพัฒนาขึ้นมาใช้นั้น ก่อนที่จะเสร็จสมบูรณ์ต้องผ่านวิธีการต่างๆ มาหลายขั้นตอน ในหัวข้อนี้ จะกล่าวถึงขั้นตอนต่างๆ ในการพัฒนาซอฟต์แวร์

8.3.1 วัฏจักรในการพัฒนาระบบ (The Systems Development Cycle)

วิธีการในการพัฒนาระบบประมวลผลข้อมูล เรียกว่า system development cycle หรือ application developmeent หรือ system analysis and design ในบทที่ 2 ได้กล่าวไว้ว่าซอฟต์แวร์เป็นทรัพยากรที่สำคัญในระบบประมวลผลข้อมูล นอกจากนี้ ฮาร์ดแวร์ และบุคลากร ก็จัด

เป็นทรัพยากรที่จำเป็นเช่นกัน เพราะเป็นทรัพยากรที่แปลงรูป (transform) ข้อมูลให้เป็นผลลัพธ์ที่ต้องการ ดังนั้น วิธีการในการพัฒนาระบบที่จะกล่าวต่อไปนี้ ครอบคลุมถึงทรัพยากรซอฟต์แวร์ ฮาร์ดแวร์ บุคลากร ข้อมูล และผลลัพธ์



- Determine whether a new or Improved information system IS required Make a preliminary study of the information needs of prospective users and the feasibility of the proposed system

- Analyze in detail the information needs of prospective users, the user organization, and any system presently used Develop the logical input processing, output, storage, and control requirements of the proposed system

- Design an information system that meets the system requirements of the analysis stage Develop specifications for the hardware (equipment and media), software (programs and procedures). people (specialists and users). data resources, and information products required to implement the input, processing, output, storage and control functions of the new system

- Develop computer programs that meet the software specifications of the design stage

- Acquire and install hardware and software Test and document the system Train people to operate and use the system Begin lo use the new system

- Monitor, evaluate. and modify the system as needed

รูปที่ 8-4 แสดงถึงวัฏจักรในการพัฒนาระบบ

วัฏจักรในการพัฒนาระบบ แบ่งออกเป็น 6 ขั้นตอน

(1) System investigation

เป็นขั้นตอนในการศึกษาความต้องการของผู้ใช้ ซึ่งจะนำข้อมูลต่างๆ ที่ได้มากำหนดความต้องการของระบบ และศึกษาความเป็นไปได้ในการพัฒนาระบบ กรณีที่สามารถพัฒนาระบบงานได้ตามความต้องการของผู้ใช้ จะดำเนินงานตามขั้นตอนขั้นต่อไป

(2) System analysis

เป็นขั้นตอนในการวิเคราะห์ในรายละเอียดถึงความต้องการต่างๆ ของผู้ใช้ระบบ รวมทั้งความต้องการของหน่วยงาน และระบบอื่นๆ ที่ใช้อยู่ในปัจจุบัน ในด้านของการประมวลผลทางด้านข้อมูลเข้า (input) ข้อมูลออก (output) หน่วยความจำ (storage) และควบคุมให้ได้ตรงตามความต้องการซึ่งเป็นจุดมุ่งหมายของระบบ

(3) System design

เป็นขั้นตอนในการออกแบบระบบ โดยระบุถึงฮาร์ดแวร์ที่ใช้ในระบบ เช่น อุปกรณ์และสื่อต่างๆ ที่ใช้ รวมทั้งซอฟต์แวร์ เช่น โปรแกรม และวิธีดำเนินงาน (procedure) เป็นต้น บุคลากรในระบบ เช่น ผู้ใช้ และผู้เชี่ยวชาญ รวมทั้งออกแบบโครงสร้างของข้อมูล ทั้งในด้านข้อมูลเข้าและข้อมูลออก การประมวลผลข้อมูล หน่วยเก็บข้อมูล (storage) และฟังก์ชันควบคุมของระบบใหม่

(4) Software development

เป็นขั้นตอนในการพัฒนาโปรแกรม โดยสร้างโปรแกรมขึ้นมาเพื่อให้สามารถทำงานได้ตามที่ได้ออกแบบระบบไว้

(5) System implementation

เป็นขั้นตอนของการใช้งาน โดยการนำเอาโปรแกรมที่พัฒนาเสร็จสมบูรณ์ ไปติดตั้ง ทำการทดสอบระบบ รวมทั้งฝึกฝน ให้ผู้ใช้ระบบ ให้สามารถปฏิบัติงานโดยใช้ระบบใหม่นี้ได้

(6) System maintenance

เป็นขั้นตอนในการบำรุงรักษาระบบ โดยตรวจสอบหรือควบคุมการทำงานของระบบคอมพิวเตอร์ และแก้ไขระบบเมื่อต้องการ

8.3.2 ระยะต่างๆ ของการเขียนโปรแกรม (The Stages of the Programming Process)

การเขียนโปรแกรมคอมพิวเตอร์ คือ วิธีการ ในการพัฒนาโปรแกรมคอมพิวเตอร์ ซึ่งประกอบด้วย กลุ่มของคำสั่งซึ่งสั่งให้คอมพิวเตอร์ทำการประมวลผล หรือกิจกรรมต่างๆ ซึ่งเกี่ยวข้องกับการเขียนคำสั่ง ในภาษาโปรแกรมต่างๆ ซึ่งสามารถแบ่งได้เป็นหลายระยะ คือ

(1) Program analysis

เป็นระยะของการวิเคราะห์ถึงจุดประสงค์ของงานประยุกต์ โดยกำหนดถึงหน้าที่ต่างๆ ที่จะให้โปรแกรมทำงานได้

(2) Program design

เป็นระยะของการวางแผน และออกแบบ ถึงคุณลักษณะของข้อมูลเข้า (input) ข้อมูลออก (output) หน่วยเก็บข้อมูล วิธีดำเนินการประมวลผล

(3) Program coding

เป็นระยะของการเขียนคำสั่งภาษาโปรแกรม ซึ่งเปลี่ยนจาก Program design เป็นโปรแกรมคอมพิวเตอร์ที่สมบูรณ์

(4) Program verification

เป็นระยะของการตรวจทาน ทดสอบ โปรแกรมที่เขียนขึ้น ให้ถูกต้องและสมบูรณ์ ตรงตามความต้องการของระบบ ซึ่งเรียกว่า debugging และ testing

(5) Program documentation

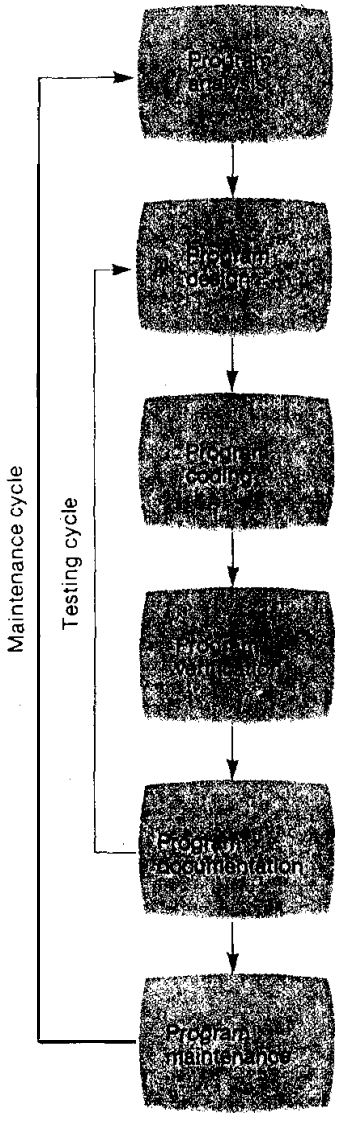
เป็นระยะของการบันทึกรายละเอียดของการออกแบบ และรายละเอียดของโปรแกรม โดยจัดทำเป็นคู่มือ และเอกสาร ของระบบ

(6) Program maintenance

เป็นระยะของการปรับปรุง หรือ สร้างโปรแกรมคอมพิวเตอร์ เพื่อพัฒนาให้ดีขึ้น ซึ่งอาจจะขยายขีดความสามารถ หรือ ปรับปรุงให้ถูกต้องยิ่งขึ้น

รูปที่ 8-5 แสดงถึงระยะต่างๆ ของการเขียนโปรแกรม

A. THE TRADITIONAL PROGRAMMING PROCESS



. Analyzing a proposed computer application to determine what the computer program is expected to accomplish.

Planning and designing the input/output characteristics, data storage specifications, processing procedures and control methods required by the proposed application.

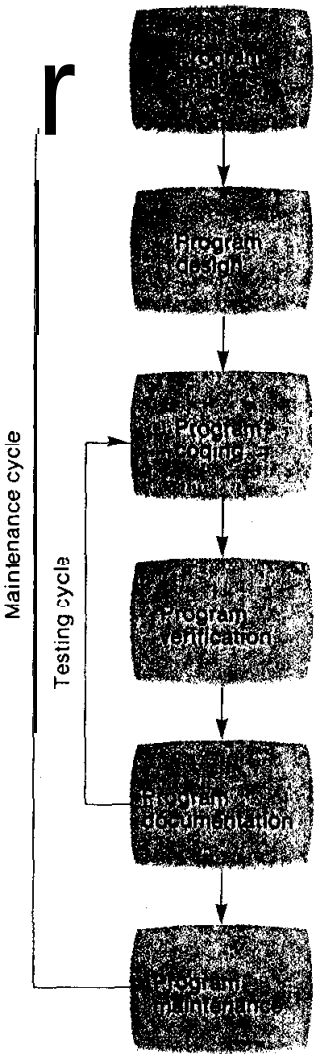
. Developing a detailed set of programming language statements that converts the program design into a complete computer program.

• Checking, testing, and correcting a newly written computer program (commonly called "debugging").

. Recording a detailed description of the design and instruction content of a computer program.

• Revising an established computer program that requires improvement such as updating, expansion, or correction.

B. THE COMPUTER-ASSISTED PROGRAMMING PROCESS



• A proposed computerapplication isanalyzed todeterminewhat the computer program is expected to accomplish.

. Interactive computer-assisted design of data entry and output display screens, documents, and reports; database access and maintenance procedures; and error handling and security procedures.

. Only complex and unusual procedures need coding. All other program code is automatically generated.

. Testing is minimal, since automatically generated code is usually error-free.

. Most documentation IS automatically created during the design and coding stages.

. Revisions can be done easily, since program specifications, code, and documentation are computerized.

8.4 การวิเคราะห์โปรแกรม (Program Analysis)

การวิเคราะห์โปรแกรมเป็นขั้นตอนแรกในการเขียนโปรแกรมคอมพิวเตอร์โดยการวิเคราะห์ถึงหน้าที่ต่างๆ ของโปรแกรม โดยแบ่งเป็นงาน หรือฟังก์ชัน ฟังก์ชันหนึ่งอาจปฏิบัติการได้ ก็ต่อเมื่อปฏิบัติงานอีกฟังก์ชันหนึ่งเสร็จก่อน การวิเคราะห์ปัญหาต่างๆ ของโปรแกรมอาจเป็นปัญหาสั้นๆ พื้นฐาน หรือปัญหาทางคณิตศาสตร์ที่ซับซ้อน ซึ่งจะต้องกำหนดปัญหา (problem definition) และกำหนดรายละเอียดของปัญหา (problem specification) ในการปฏิบัติการให้ชัดเจน ในกรณีที่งานประยุกต์เป็นการประมวลผลข้อมูล การวิเคราะห์โปรแกรมควรวิเคราะห์ถึงข้อกำหนดรายละเอียดของซอฟต์แวร์ (software specification) ในระยะของการออกแบบ (design stage) หรือความต้องการในรายละเอียดของโปรแกรม (program specification) อย่างเช่น

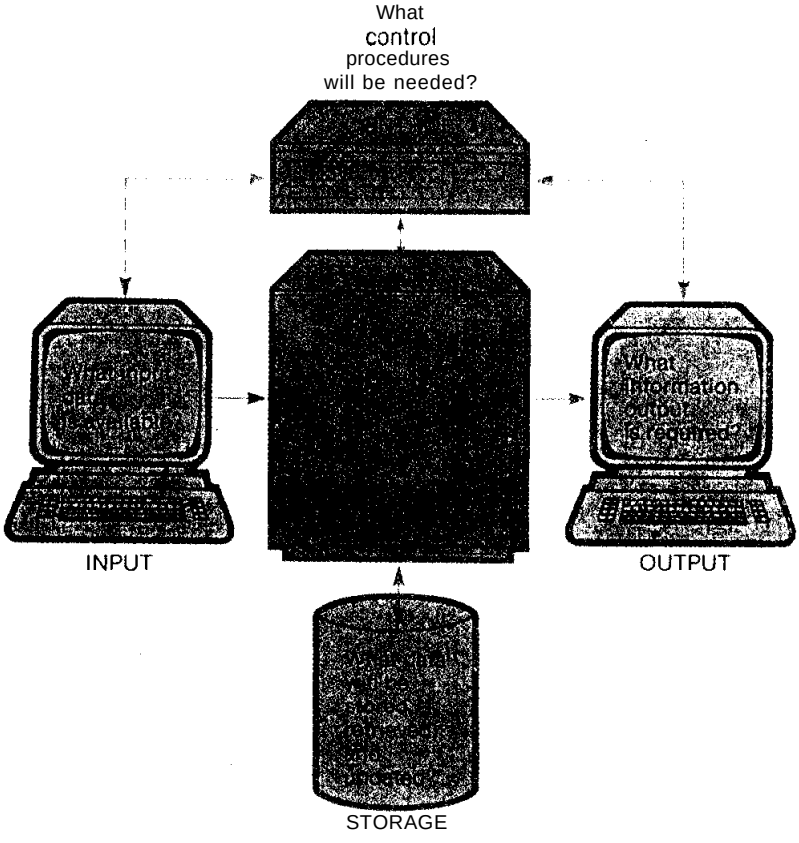
(1) Output โดยวิเคราะห์ว่าผลลัพธ์ที่ต้องการคืออะไรบ้าง

(2) Input โดยวิเคราะห์ว่าข้อมูลที่สามารถเรียกหาได้ (available) มีอะไรได้บ้าง

(3) Storage โดยวิเคราะห์ว่าข้อมูลจะเก็บ (store) หรือ ดึง (retrieved) หรือแก้ไข ในหน่วยเก็บข้อมูลอะไร

(4) Proceessing โดยวิเคราะห์ถึงวิธีการประมวลผลต่างๆ ซึ่งอาจเป็นการคำนวณทางคณิตศาสตร์ การเปรียบเทียบ และกรรมวิธี (proceduree) อื่นๆ

(5) Control Procedure โดยวิเคราะห์วิธีการควบคุมการทำงานของโปรแกรม



รูปที่ 8-6 เป็นรูปแสดงการวิเคราะห์โปรแกรม

8.5 การออกแบบโปรแกรม (Program design)

ระยะของการออกแบบโปรแกรม เป็นระยะของการวางแผนและออกแบบ โดยระบุคุณลักษณะของข้อมูลเข้า (Input) ข้อมูลออก (Output) กรรมวิธีประมวลผล กำหนดรายละเอียดของหน่วยเก็บข้อมูล และวิธีการควบคุม ซึ่งค่าของความพยายาม (effort) ในการวิเคราะห์และออกแบบโปรแกรม ขึ้นอยู่กับความซับซ้อนของงานประยุกต์ และจำนวนของงานในระบบ โดยปกติจะเป็นการกำหนดกฎเกณฑ์ทางตรรก และคำสั่งที่ระบุถึงการปฏิบัติงาน ซึ่งเรียกว่า อัลกอริทึม ซึ่งเป็นวิธีการในการแก้ปัญหา เพื่อให้โปรแกรมสามารถทำงานได้ตรงตามวัตถุประสงค์

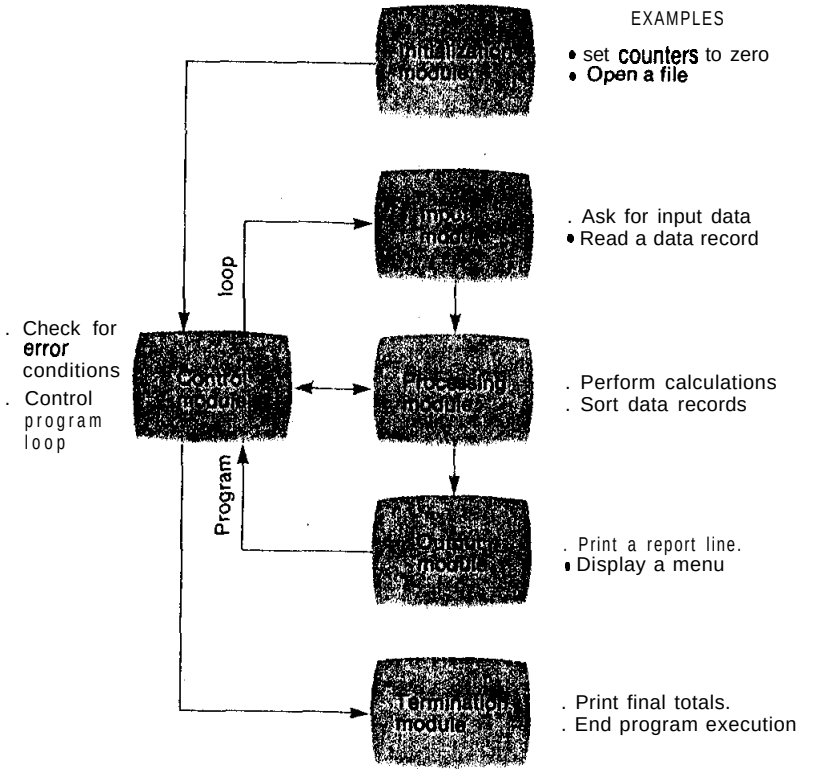
ในการออกแบบโปรแกรม โดยปกติจะแบ่งโปรแกรมออกเป็น ส่วนย่อยๆ เรียกว่า โมดูล (modules หรือ subdivisions) โดยแต่ละโมดูล จะมีส่วนของการกำหนดค่าเริ่มต้น (initialization), ข้อมูลเข้า (input), ประมวลผล (processing) และส่วนแสดงผล (output) และส่วนของการสิ้นสุดหรือเลิกใช้ (termination) โมดูล โปรแกรมส่วนมากมีโมดูลควบคุม ,ใช้สำหรับตรวจสอบและควบคุมการทำงานต่างๆ เช่น

(1) ลำดับของการประมวลผล (order of processing)

(2) ขั้นตอนการทำงานซ้ำๆ (looping)

(3) เงื่อนไขยกเว้น (exceptional conditions) เช่น ข้อผิดพลาดต่างๆ (errors)

(4) สิ่งเบี่ยงเบนจากการประมวลผลปกติ (other deviations from normal processing requirements)



รูปที่ 8-17 เป็นตัวอย่างของตัวอย่างโมดูลโปรแกรม

IDEAL's framework divides a commercial application program into the following components:

I. General declarative information, including—

- A declaration of the application and its inputs and outputs.
- The logical database definition (or traditional file record layout for applications that use conventional file access methods).
- Possible report definitions.
- Screen panel layouts and definitions for online screen-oriented applications.
- Input and output parameters.

II. The application program itself, consisting of—

- The definition of working data (data local to the program).
- The logic, computations, terminal interaction and database maintenance rules, procedures, and actions.

Since all but the last component are highly declarative or descriptive in nature. the approach in IDEAL is to use special-purpose, fill-in-the-blank screen formats, or "panels." These panels are processed interactively and eliminate the need for a textual language. All logic, computations, and database maintenance are expressed in a language. IDEAL/PDL, and in a manner that solves traditional problems by offering a comprehensive. yet simple, unified, structured. and very high-level language.

Source: Courtesy Applied Data Research.

รูปที่ 8-8 เป็นส่วนประกอบของโปรแกรมช่วยเ

8.5. I Top-down design

เป็นวิธีการออกแบบโปรแกรม ซึ่งเป็นส่วนหลักของการเขียนโปรแกรมโครงสร้าง ซึ่งประกอบด้วยขั้นตอนต่างๆ ดังนี้

(1) โปรแกรมเมอร์ต้องกำหนดผลลัพธ์ (output) ที่ต้องการ, ข้อมูลเข้า (input) และงานประมวลผลหลักๆ ที่จำเป็นในการเปลี่ยนข้อมูลเข้าเป็นผลลัพธ์

(2) งานประมวลผลหลักๆ จจะถูกแบ่งออกเป็นโมดูล ซึ่งเป็นอิสระต่อกัน

(3) กำหนดการประมวลผลทางตรรก หรือ อัลลกอริทึม ของแต่ละโมดูล โดยกำหนดโมดูลหลักก่อนเป็นโมดูลแรก ต่อจากนั้นจึงเป็นโมดูลในระดับต่อมา

เครื่องมือที่ใช้สำหรับออกแบบ อย่างเช่น structure chart, HIPO charts, flowcharts, pseudocode, decision tables และ input/output and storage layout forms เป็นเครื่องมือที่สามารถเลือกนำไปออกแบบแต่ละโมดูลของโปรแกรมได้

ข้อจำกัดของ Top-down design

(1) แต่ละโมดูล ควรมีทางเข้า 1 ทาง และทางออก 1 ทาง เท่านั้น

(2) แต่ละโมดูลควรมีหน้าที่การทำงานเดียวเท่านั้น (only one program function) เช่น การอ่านแฟ้มข้อมูลหลัก

(3) แต่ละโมดูลควรมีจำนวนคำสั่งพอประมาณ เมื่อแปลเป็นของโปรแกรมภาษาแล้วไม่ควรเกิน 50 บรรทัด

วัตถุประสงค์ของการกำหนดข้อจำกัดต่างๆ เหล่านี้ เพื่อให้เป็นมาตรฐานและเพื่อให้ง่ายต่อการอ่าน การทดสอบ และการแก้ไขให้ถูกต้องสมบูรณ์

8.5.2 Structure and Hierachy Charts

เป็นวิธีการออกแบบโปรแกรมซึ่งแสดงถึงแผนภูมิความสัมพันธ์ของโมดูลต่างๆ แบบเป็นลำดับชั้น (hierarchical) โดยแสดงการไหลของตรรกะ (flow of logic) ในโปรแกรมโดยใช้ โครงสร้างต้นไม้ (tree) เชื่อมต่อระหว่างโมดูล ซึ่งสามารถอ้างถึงได้ในกรณีจัดทำเอกสารของโปรแกรม