

บทที่ 1

การแก้ปัญหาโปรแกรม

วัตถุประสงค์

1. เพื่อให้ นักศึกษาทราบถึงภาพรวมของคอมพิวเตอร์
2. เพื่อให้ นักศึกษาทราบถึงกระบวนการประมวลผลโปรแกรมภาษาระดับสูง
3. เพื่อให้ นักศึกษาทราบถึงขั้นตอนในการพัฒนาซอฟต์แวร์
4. เพื่อให้ทราบถึงคุณภาพของซอฟต์แวร์
5. เพื่อให้ทราบถึงการการพัฒนาซอฟต์แวร์ที่มีคุณภาพ

ตั้งแต่ปี ค.ศ.1940 เป็นต้นมา การพัฒนาของเทคโนโลยีคอมพิวเตอร์เติบโตขึ้นอย่างรวดเร็วส่งผลให้การดำเนินชีวิตของมนุษย์ในปัจจุบันให้ผูกติดกับเทคโนโลยีอย่างหลีกเลี่ยงไม่ได้ ไม่ว่าจะเป็นการส่งหรือรับข้อมูลข้อมูลผ่านทางไปรษณีย์อิเล็กทรอนิกส์(E-mail), การซื้อสินค้าทางอินเทอร์เน็ต(E-commerce) , การค้นหาข้อมูลจาก World Wide Web เพื่อทำงานวิจัยหรือทำการบ้านคอมพิวเตอร์กลายเป็นเครื่องมือที่สำคัญเพื่อช่วยมนุษย์ในการเก็บข้อมูล การดึงข้อมูลที่ต้องการ การประมวลผลข้อมูล รวมทั้งการสร้างสารสนเทศที่เป็นประโยชน์ไม่ว่าจะอยู่ในรูปของข้อความ ตัวเลข รายงาน ภาพนิ่ง ภาพเคลื่อนไหว กราฟิก หรือแม้กระทั่งเสียง ซึ่งการปฏิบัติงานของคอมพิวเตอร์มีความเร็วในการทำงานสูง และให้ความถูกต้องแม่นยำมากกว่าการปฏิบัติด้วยแรงงานมนุษย์ ซึ่งการสร้างสรรสิ่งที่เกิดเป็นประโยชน์ต่างๆเหล่านี้ โปรแกรมเมอร์ต้องมีการจัดเตรียมคำสั่งหรือโปรแกรมเพื่อให้เครื่องปฏิบัติงานได้ถูกต้องและมีประสิทธิภาพ ซึ่งมีหลายสิ่งหลายอย่างที่ควรเรียนรู้ดังต่อไปนี้

1.1 ภาพรวมของคอมพิวเตอร์

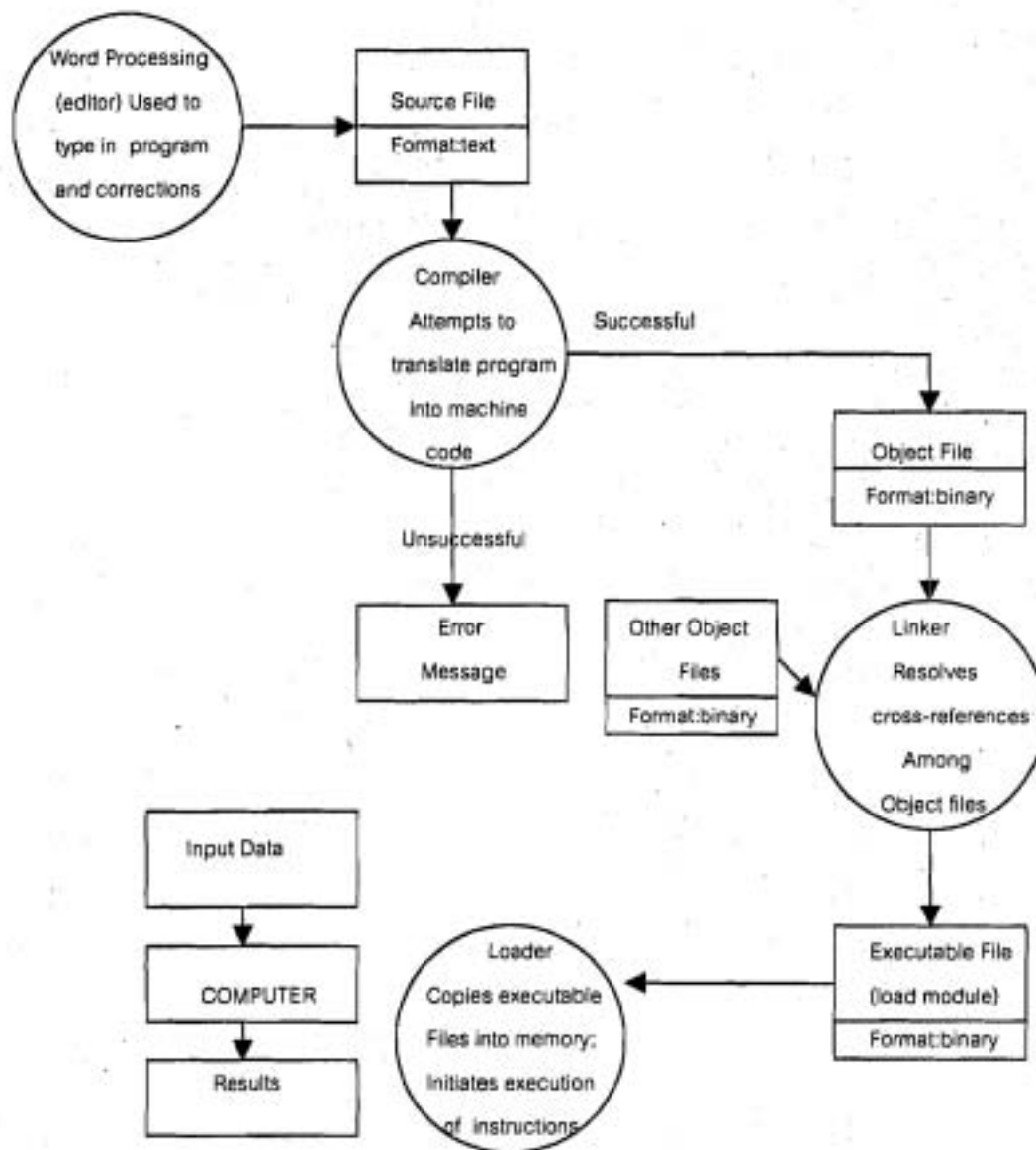
คอมพิวเตอร์คืออุปกรณ์ที่ใช้สำหรับประมวลผล ซึ่งพื้นฐานของดิจิทัลคอมพิวเตอร์ในปัจจุบันถูกออกแบบโดย Von Neumann ต่อมาได้มีการพัฒนามาเป็นลำดับทำให้คอมพิวเตอร์มีทั้งขนาดและประสิทธิภาพที่แตกต่างกัน องค์ประกอบหลักที่สำคัญของคอมพิวเตอร์ประกอบด้วย 2 ส่วนคือ ฮาร์ดแวร์ และซอฟต์แวร์ ฮาร์ดแวร์คืออุปกรณ์อิเล็กทรอนิกส์ต่างๆที่ใช้ในการปฏิบัติการ ส่วนซอฟต์แวร์เป็นโปรแกรมหรือชุดคำสั่งที่ใช้ในการติดต่อกับฮาร์ดแวร์นั่นเอง โดยการติดต่อระหว่างกันใช้ภาษาโปรแกรมที่เครื่องคอมพิวเตอร์เข้าใจเรียกว่าภาษาเครื่อง (machine language) ซึ่งประกอบด้วยเลขฐานสอง ในสมัยแรกๆโปรแกรมเมอร์ต้องเขียนภาษาเครื่องซึ่งยากต่อการพัฒนาและยากต่อการตรวจสอบและแก้ไข จึงมีการพัฒนาภาษาโปรแกรมต่อมาอีกหลายยุค ทำให้โปรแกรมเมอร์สามารถพัฒนางานประยุกต์ต่างๆได้ง่ายขึ้นเรื่อยๆมาเป็นลำดับ จนกระทั่งในปัจจุบันสามารถสั่งงานเครื่องเป็นภาษาโปรแกรมในรูปแบบของการเขียนโปรแกรมเชิงวัตถุหรือภาษาธรรมชาติ สำหรับตำราเล่มนี้มุ่งเน้นให้นักศึกษาสามารถเขียนโปรแกรมเพื่อให้เครื่องคอมพิวเตอร์ปฏิบัติการตามขั้นตอนที่กำหนดได้ (Procedural Programming) และสามารถเขียนโปรแกรมในแนวของเชิงวัตถุ(Object-Oriented Programming) ได้

1.2 การประมวลผลโปรแกรมภาษาระดับสูง

โปรแกรมที่โปรแกรมเมอร์เขียนขึ้นจากภาษาโปรแกรมระดับสูง ต้องผ่านการแปลโดย คอมไพเลอร์เพื่อตรวจสอบไวยากรณ์ของภาษานั้นๆเสียก่อน ต่อจากนั้นจะกระทำการสร้างแฟ้ม ภาษาเครื่องของโปรแกรมเพื่อให้สามารถปฏิบัติงานได้ โดยทั่วไปขั้นตอนในการเตรียมโปรแกรม ภาษาระดับสูงสำหรับประมวลผล สามารถแสดงได้ดังรูปที่ 1.1

การประมวลผลเริ่มจากแฟ้มต้นฉบับ(source file)ที่บรรจุภาษาโปรแกรมภาษาระดับสูง ซึ่งสร้างจากโปรแกรมกระตาดาทด(editor) สมมุติว่าโปรแกรมเมอร์เลือกภาษา C++ ต้องตั้งชื่อ แฟ้มโปรแกรมและเขียนคำสั่งในการปฏิบัติงานต่างๆเก็บในแฟ้มนี้ สมมุติว่าชื่อแฟ้ม myprog.cpp การปฏิบัติงานเริ่มจาก ตัวแปลภาษา(compiler) ตรวจสอบไวยากรณ์ของภาษา C++ ถ้าไม่ถูกต้องโปรแกรมเมอร์ต้องแก้ไขให้ถูกต้อง ถ้าไวยากรณ์ภาษาถูกต้องตัวแปลภาษา จะทำการสร้างแฟ้มภาษาเครื่อง (object file) ซึ่งมีการแปลงคำสั่งภาษา C++ ที่โปรแกรมเมอร์ เขียนขึ้นเป็นเลขฐานสอง ในที่นี้คือแฟ้ม myprog.obj แฟ้มนี้ยังไม่สามารถทำงานได้เพราะเป็น คำสั่งภาษาเครื่องที่ไม่สมบูรณ์ ในปัจจุบันการเขียนโปรแกรมแนวใหม่จะมีการเรียกใช้โมดูลใน การปฏิบัติการที่ได้เตรียมไว้ให้ซึ่งอยู่ภายนอกโปรแกรมซึ่งเป็นภาษาเครื่อง เช่นโมดูลที่เป็น ฟังก์ชันในการหาค่า หรือการปฏิบัติการใดๆที่มีการเรียกใช้บ่อยๆ เป็นต้น ดังนั้นจึงต้องมี โปรแกรมสำหรับเชื่อมโยง(linker program) ทำหน้าที่ในการนำโมดูลที่มีการเรียกใช้อยู่ภายนอก โปรแกรมมาเชื่อมโยงคำสั่งภาษาเครื่องที่พัฒนาขึ้นเพื่อให้โปรแกรมสมบูรณ์และพร้อมทำงานได้ และสร้างเป็นแฟ้มใหม่ชื่อว่า myprog.exe สำหรับการทำงานนั้นต้องมีโหลดเดอร์ (loader) เป็น โปรแกรมระบบที่ทำหน้าที่ในการนำโปรแกรมภาษาเครื่องไปเก็บในหน่วยความจำและบอกให้ ซีพียูปฏิบัติการตามคำสั่งโปรแกรม การประมวลผลโปรแกรมซีพียูต้องตรวจสอบคำสั่งแต่ละ คำสั่งในหน่วยความจำ และปฏิบัติงานคำสั่งเป็นลำดับ ซึ่งระหว่างการประมวลผลคำสั่งอาจมี การนำข้อมูลเข้าไปเก็บในหน่วยความจำ หรือมีการปฏิบัติการกับข้อมูลต่างๆเหล่านั้น อาทิเช่น การคำนวณ การเปรียบเทียบ เป็นต้น รวมทั้งคำสั่งในการนำข้อมูลซึ่งเป็นผลลัพธ์ออกจาก หน่วยความจำมาแสดงผลตามรูปแบบที่ต้องการ รูปที่ 1.2 แสดงการประมวลผลโปรแกรมคิด คำนวณ แบ่งเป็น 3 ขั้นตอนดังนี้

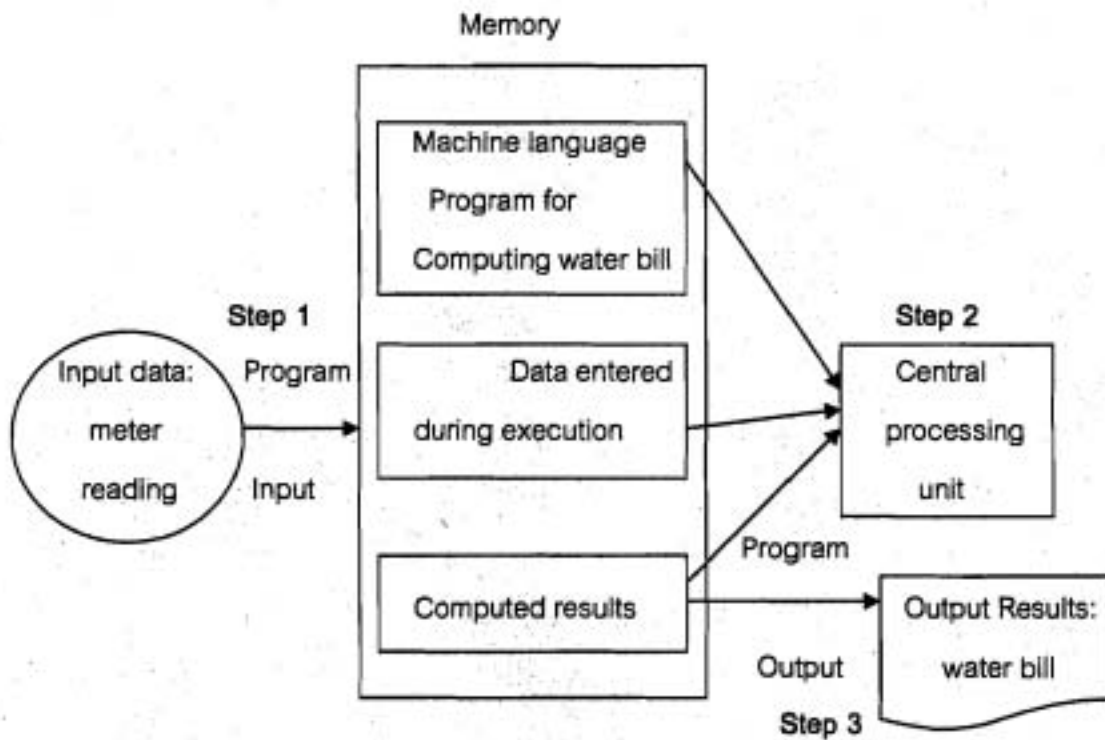
ขั้นตอนที่ 1 โปรแกรมภาษาเครื่องในการคิดคำนวณ และข้อมูลที่ใช้ในการประมวลผลคือ จำนวนของยูนิตคำนวณที่ใช้ จะถูกเก็บในหน่วยความจำ



รูปที่ 1.1 แสดงการเตรียมโปรแกรมภาษาระดับสูงสำหรับการประมวลผล

ขั้นตอนที่ 2 โปรแกรมจะทำการประมวลผลโดยนำจำนวนของยูนิคของน้ำมาคำนวณหา
ค่าน้ำที่ต้องจ่าย และเก็บผลลัพธ์ไว้ในหน่วยความจำ

ขั้นตอนที่ 3 นำผลลัพธ์ที่คำนวณเก็บไว้ในหน่วยความจำ แสดงผลออกมาเป็นบิลค่าน้ำ



รูปที่ 1.2 แสดงการประมวลผลโปรแกรมคิดค่าน้ำ

1.3 ขั้นตอนในการพัฒนาซอฟต์แวร์

โปรแกรมเมอร์สามารถแบ่งขั้นตอนในการพัฒนาซอฟต์แวร์ (software development method) เป็น 6 ขั้นตอน เพื่อช่วยในการแก้ปัญหาโปรแกรม ดังนี้

1. ระบุถึงความต้องการ (problem)
2. วิเคราะห์ปัญหา (analysis)
3. ออกแบบอัลกอริทึมเพื่อแก้ปัญหา (design)
4. สร้างโปรแกรม (implementation)
5. ทดสอบและตรวจสอบความสมบูรณ์ของโปรแกรม (testing)
6. บำรุงรักษาและแก้ไขโปรแกรม (maintenance)

ขั้นตอนแรกเป็นการระบุถึงความต้องการของปัญหา ซึ่งความต้องการต่างๆอาจเป็นปัญหาที่เกิดขึ้นซึ่งผู้พัฒนาสามารถสอบถามได้จากบุคคลต่างๆที่เกี่ยวข้อง ความต้องการที่ดีต้องชัดเจน ไม่คลุมเครือ มีความจำเป็น สามารถทวนสอบหรือตรวจสอบได้ ติดตามได้ ต่อจากนั้นจะนำความต้องการต่างๆเหล่านี้มาวิเคราะห์ถึงหนทางในการแก้ปัญหา โดยนำปัญหาต่างๆเหล่านี้มากำหนดเป็นนิยาม(abstraction)ของข้อมูลเข้า ,ข้อมูลนำออก ,การคำนวณ เงื่อนไขในการปฏิบัติการต่างๆ ต่อจากนั้นออกแบบอัลกอริทึมในการแก้ปัญหาโดยมีการแบ่งปัญหาออกเป็นปัญหาย่อยๆเพื่อให้ง่ายต่อการพัฒนาซึ่งเรียกว่า top-down design หรือ divide and conquer ในแต่ละปัญหาย่อยจะกำหนดเป็นลำดับขั้นตอนในการแก้ปัญหาโดยละเอียด เรียกว่า algorithm refinement ต่อจากนั้นจะนำไปเขียนเป็นภาษาโปรแกรมให้ถูกต้องตามไวยากรณ์ของภาษานั้นๆ นำโปรแกรมที่สร้างขึ้นไปทดสอบโดยกำหนดข้อมูลทดสอบหลายๆชุด เพื่อให้แน่ใจว่าการทำงานถูกต้อง และสามารถนำไปปฏิบัติการในการแก้ปัญหาได้ การบำรุงรักษาจะเริ่มต้นเมื่อโปรแกรมที่พัฒนาขึ้นได้ถูกนำไปใช้งานเรียบร้อยแล้วขั้นตอนนี้เป็นการป้องกันและแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นในขณะทำงาน

กรณีศึกษา : การแปลงไมล์เป็นกิโลเมตร

ปัญหา ต้องการแปลงระยะทางที่เป็นไมล์ให้เป็นกิโลเมตร

วิเคราะห์ ต้องหาข้อมูลว่าอะไรคือข้อมูลเข้า อะไรคือผลลัพธ์ที่ต้องการ และต้องมีการประมวลผลอย่างไร ในที่นี้ข้อมูลเข้าคือระยะทางที่เป็นไมล์ ผลลัพธ์ที่ต้องการคือระยะทางที่เป็นกิโลเมตร สำหรับการประมวลผลเป็นการคำนวณที่ต้องทราบว่า 1 ไมล์มีค่าเท่ากับ 1.609 กิโลเมตร ต่อจากนั้นนิยามข้อมูลต่างๆเหล่านี้เพื่อให้สามารถเก็บในหน่วยความจำของเครื่องคอมพิวเตอร์ได้

DATA REQUIREMENTS

Problem Input

miles ระยะทางเป็นไมล์

Problem Output

kms ระยะทางเป็นกิโลเมตร

Formular

1 ไมล์ = 1.609 กิโลเมตร

ออกแบบ ขั้นตอนในการแก้ปัญหาแบ่งออกเป็น 3 ขั้นตอนใหญ่ ประกอบด้วย

ALGORITHM

1. รับระยะทางเป็นไมล์
2. แปลงระยะทางจากไมล์เป็นกิโลเมตร
3. แสดงระยะทางเป็นกิโลเมตร

พิจารณาแต่ละขั้นตอนว่าชัดเจนและสมบูรณ์หรือไม่ ปรากฏว่าในขั้นตอนที่ 1 และขั้นตอนที่ 3 เป็นขั้นตอนที่เป็นการรับและแสดงผลข้อมูลพื้นฐาน แต่ขั้นตอนที่ 2 ยังไม่ทราบว่ามีวิธีการอย่างไรในการแปลงดังนั้นต้องกำหนดในรายละเอียดเพื่อให้ชัดเจนขึ้นดังนี้

ALGORITHM WITH REFINEMENTS

1. รับระยะทางเป็นไมล์
2. แปลงระยะทางจากไมล์เป็นกิโลเมตร
 - 2.1 ระยะทาง 1 ไมล์มีค่าเท่ากับ 1.609 กิโลเมตร
3. แสดงระยะทางเป็นกิโลเมตร

ตรวจสอบอัลกอริทึมก่อนกระทำในขั้นตอนต่อไป ถ้าในขั้นตอนที่ 1 ถ้ารับระยะทางเท่ากับ 10.0 ไมล์ ในขั้นตอนที่ 2.1 จะทำการแปลงโดยนำ $1.609 \cdot 10.00$ นั่นคือ 16.09 กิโลเมตรนั่นเอง ต่อจากนั้นจะแสดงผลออกมาเป็นขั้นตอนที่ 3

เขียนโปรแกรม ทำการเปลี่ยนอัลกอริทึมเป็นภาษา C++ ดังนี้

```
//miles.cpp
//Converts distance in miles to kilometers.
#include <iostream>
using namespace std;
```

```

int main()                                //start of main function
{
    const float KM_PER_MILE=1.609        //1.609 km in a mile
    float miles,                          //input : distance in miles
          kms;                             //output: distance in kilometers
    //Get the distance in miles.
    cout << "Enter the distance to miles: ";
    cin >> miles;
    // Convert the distance to kilometers.
    kms = KM_PER_MILE * miles;
    // Display the distance to kilometers.
    cout << " The distance in kilometers is " << kms << endl;
    return 0;
}

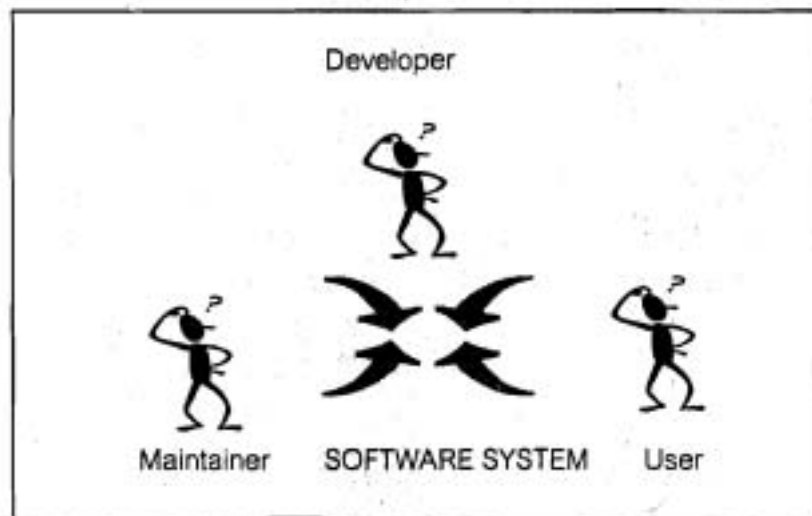
```

ทดสอบ ถ้าเขียนโปรแกรมถูกต้อง โดยการป้อนระยะทางเป็นไมล์เท่ากับ 10.0 จะได้ผลลัพธ์เป็นระยะทางเท่ากับ 16.09 กิโลเมตร ดังนี้

<pre> Enter the distance in miles: 10.0 The distance in kilometer is 16.09 </pre>

1.4 คุณภาพของซอฟต์แวร์

วิศวกรรมซอฟต์แวร์คือกลยุทธ์ของการผลิตซอฟต์แวร์ที่มีคุณภาพ ความแตกต่างของคุณภาพที่ว่าดีหรือไม่ดีนั้น เราวัดจากอะไร? โดยทั่วไปแล้วคุณภาพของซอฟต์แวร์ที่ดีนั้นขึ้นอยู่กับปัจจัยหลายๆอย่าง ผู้ที่ตัดสินใจอาจเป็นผู้ใช้ระบบ ซึ่งวัดจากความง่ายในการเรียนรู้ ง่ายในการใช้งาน ถ้าเป็นผู้วิเคราะห์และออกแบบระบบหรือโปรแกรมเมอร์ วัดคุณภาพจากการบำรุงรักษาโปรแกรม การแก้ไขปรับปรุงระบบเก่าให้มีประสิทธิภาพดีขึ้น รูปภาพที่ 1.3 แสดงให้เห็นว่าซอฟต์แวร์ที่มีคุณภาพสูง ต้องมีคุณลักษณะที่เชื่อมโยงกับผู้ใช้งาน, ผู้พัฒนาระบบ และ ผู้บำรุงรักษาระบบ



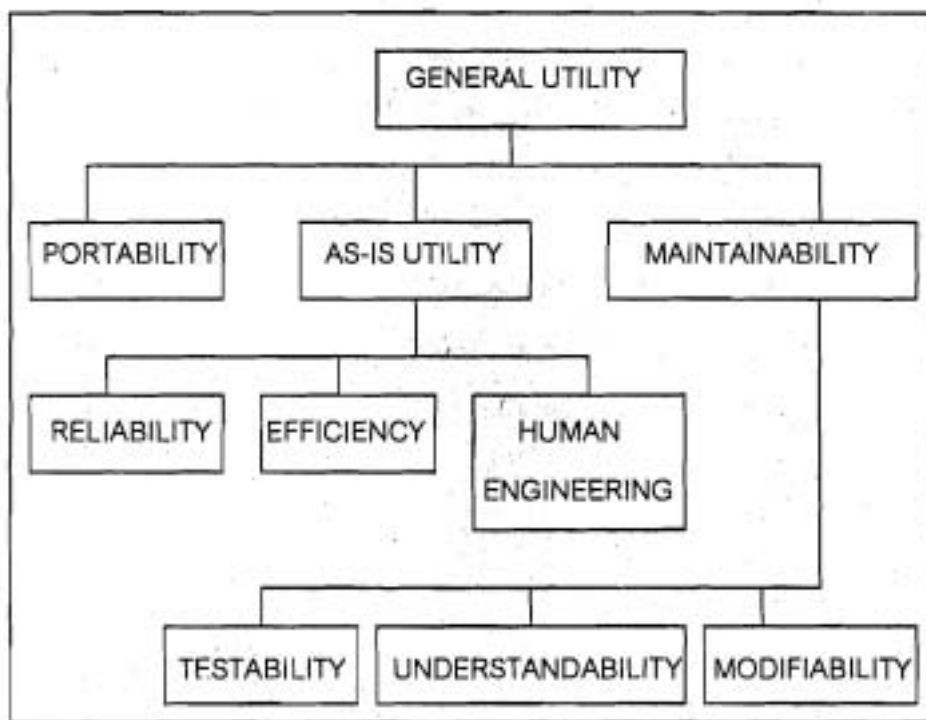
รูปภาพที่ 1.3 Judges of Software Quality

Boehm([BOE78]) ได้กล่าวถึงคุณลักษณะของซอฟต์แวร์ที่มีคุณภาพโดยวัดจากผู้เกี่ยวข้องกับซอฟต์แวร์ 3 กลุ่ม คือ กลุ่มแรกคือผู้ใช้ วัดจากประโยชน์ต่อการใช้งาน(useful)เป็นสำคัญ ซอฟต์แวร์ที่มีคุณภาพจะต้องสามารถกระทำงานตามที่ลูกค้าต้องการได้อย่างถูกต้อง กลุ่มที่สองคือผู้บำรุงรักษาระบบ วัดคุณภาพจากการอัปเดตและการเปลี่ยนแปลงระบบ เช่น ผู้ใช้ต้องการใช้ระบบจากสถานที่ตั้งอื่นๆ หรือจากคอมพิวเตอร์เครื่องอื่นๆที่ต่างกัน ผู้บำรุงรักษาระบบต้องจัดการได้โดยง่าย สามารถย้ายซอฟต์แวร์จากเครื่องคอมพิวเตอร์ระบบเก่า ไปยัง

เครื่องคอมพิวเตอร์ระบบใหม่ที่มีการปรับเปลี่ยนอุปกรณ์ฮาร์ดแวร์หรือระบบซอฟต์แวร์ได้โดยปราศจากความผิดพลาดหรือความยุ่งยาก ตัวอย่าง ถ้าตัวแปลภาษาโปรแกรมภาษาซี ถูกแทนที่ด้วยตัวแปลภาษาซีอีกตัวหนึ่งที่พัฒนาขึ้นใหม่อาจเป็นรุ่นใหม่ๆ ผู้บำรุงรักษาซอฟต์แวร์ต้องสามารถติดตั้งซอฟต์แวร์นี้ได้โดยง่ายการทำงานของซอฟต์แวร์สามารถกระทำตามได้ถูกต้องและต้องไม่ลด ประสิทธิภาพลงจากเดิม กลุ่มสุดท้ายคือโปรแกรมเมอร์ผู้บำรุงรักษาระบบ ทำหน้าที่เปลี่ยนแปลงระบบตามที่ลูกค้าต้องการ ป้องกันและแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นซอฟต์แวร์ที่มีคุณภาพต้องเอื้อประโยชน์ให้กับโปรแกรมเมอร์โดยสามารถให้โปรแกรมเมอร์ทราบตำแหน่งของ คำสั่งที่ผิดพลาดได้ง่าย เข้าใจและแก้ไขความผิดพลาดได้ง่าย

ผู้ใช้ทั้งสามกลุ่มมีความคาดหวังเหมือนกันว่า ระบบต้องมีความน่าเชื่อถือ และมีประสิทธิภาพ ความน่าเชื่อถือวัดได้จากระดับของความถูกต้องของผลลัพธ์จากการทำงานของระบบ ซอฟต์แวร์ที่มีคุณภาพต้องมีระดับของความถูกต้องอยู่ในเกณฑ์สูง นอกจากนี้ต้องใช้เวลาในการปฏิบัติงาน หรือการตอบสนองกับผู้ใช้อย่างรวดเร็วภายในเวลาที่ผู้ใช้ยอมรับได้รวมทั้งซอฟต์แวร์ต้องสามารถเข้าใจและเรียนรู้ได้ง่าย ระบบบางระบบมีความน่าเชื่อถือและมีประสิทธิภาพที่ดีเยี่ยมแต่ผู้ใช้ไม่สามารถเข้าใจได้ง่าย ใช้งานยาก ก็มีผลให้ซอฟต์แวร์นี้ไม่เป็นที่ยอมรับ เราสามารถสรุปได้ว่าซอฟต์แวร์ที่มีคุณภาพต้องมีคุณลักษณะดังนี้

- (1) สามารถทำงานตามที่ผู้ใช้ต้องการให้กระทำได้อย่างถูกต้อง
- (2) ใช้ทรัพยากรคอมพิวเตอร์อย่างถูกต้องและมีประสิทธิภาพ
- (3) ง่ายต่อการเรียนรู้ และใช้งาน
- (4) ผู้พัฒนาซอฟต์แวร์ สามารถออกแบบ ทดสอบ ทดสอบ และ บำรุงรักษา ระบบได้ง่าย



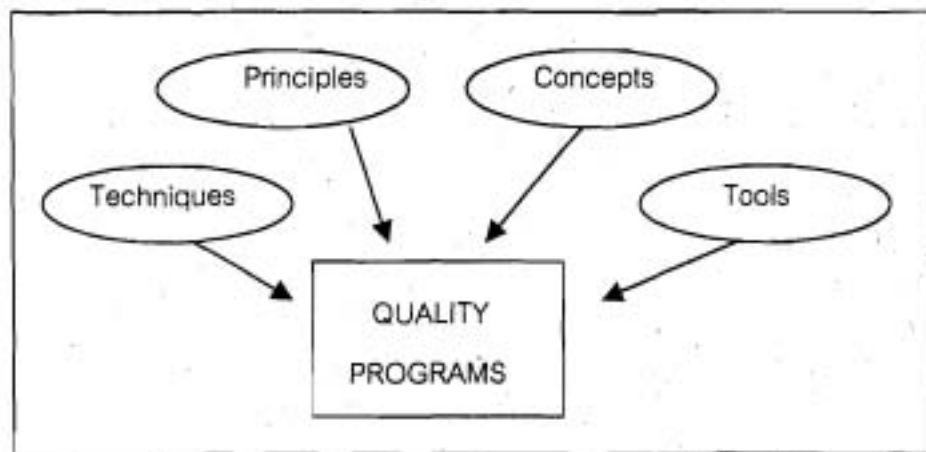
รูปภาพที่ 1.4 Characteristics of software

การผลิตซอฟต์แวร์ที่มีคุณภาพนั้น ถึงแม้ว่าเราจะทราบถึงหลักการในการสร้างหรือมีความเชี่ยวชาญภาษาโปรแกรมที่ใช้ในการพัฒนาซอฟต์แวร์เป็นอย่างดีก็ยังไม่เพียงพอ เรายังต้องมีความรู้ในเรื่องอื่นๆอีก ตามรูปภาพที่ 1.5 แสดงให้เห็นว่าเราต้องเรียนรู้ถึงแนวความคิด เทคนิค และเครื่องมือต่างๆที่ช่วยในการออกแบบและพัฒนาโปรแกรม เราจึงสามารถสร้างโปรแกรมที่มีคุณภาพได้

งานและความรับผิดชอบ

การพัฒนาโครงการซอฟต์แวร์เริ่มจากการติดต่อสื่อสารระหว่างลูกค้าและผู้พัฒนาซอฟต์แวร์ โดยทำความเข้าใจถึงความต้องการและความจำเป็นในการสร้างระบบให้ต้องแท้เสียก่อน จาก การสอบถาม รวบรวมข้อมูล สังเกต ออกแบบสอบถาม เพื่อทำความเข้าใจกับบุคคลต่างๆที่เกี่ยวข้องกับระบบทั้งหมด ก่อนที่จะสร้างระบบขึ้นมา

จำนวนของบุคคลในทีมงานพัฒนาซอฟต์แวร์จะขึ้นกับขนาดและระดับความยากง่ายของโครงการ ซึ่งหน้าที่และความรับผิดชอบของแต่ละบุคคลจะแตกต่างกันออกไปขึ้นอยู่กับขนาดของโครงการ ถ้าเป็นโครงการขนาดใหญ่เราอาจได้รับมอบหมายให้ทำเฉพาะงาน แต่ถ้าเป็นโครงการขนาดเล็กเราอาจได้ทำในหลายๆหน้าที่ โดยทั่วไปแล้วผู้ที่เกี่ยวข้องในโครงการทั้งหมดจะแบ่งเป็น 3 กลุ่มใหญ่ๆ คือ ลูกค้า ผู้ใช้ระบบ และผู้พัฒนาระบบ ลูกค้าในที่นี้อาจเป็นบริษัท องค์กร หรือบุคคลที่ลงทุนเสียค่าใช้จ่ายในการพัฒนาระบบซอฟต์แวร์ ผู้พัฒนาระบบอาจเป็นบริษัทซอฟต์แวร์ หรือเป็นองค์กร หรือเป็นบุคคลใดๆที่รับจ้างสร้างระบบซอฟต์แวร์ตามที่ลูกค้าต้องการ สำหรับผู้ใช้ระบบคือบุคคลใดๆ หรือเจ้าหน้าที่ หรือพนักงานที่ใช้ระบบในการทำงานเป็นประจำ โดยทั่วไปนั่งอยู่หน้าเครื่องเทอร์มินัล



รูปภาพที่ 1.5 How to Build Quality Programs

ความสัมพันธ์ของบุคลากรทั้งสามกลุ่มนั้น เริ่มจากลูกค้าเจรจาตกลงกับผู้พัฒนาระบบ โดยมีการพูดคุยถึงความต้องการต่างๆในระบบ ซึ่งผู้พัฒนาระบบต้องทำความเข้าใจถึงสิ่งที่ลูกค้าและผู้ใช้ระบบต้องการ เพราะในบางกรณีผู้ที่มาติดต่ออาจไม่ใช่ผู้ใช้ระบบจริงๆ ผู้พัฒนาต้องไปติดต่อกับผู้ใช้ระบบจริงเพื่อให้ได้ข้อมูลที่ถูกต้องและเป็นจริงที่สุด และก่อนที่จะตกลงทำสัญญา ผู้พัฒนาต้องทราบระบบ(system) อย่างกว้างๆเสียก่อนว่าจะมีแนวทางในการสร้างหรือแก้ปัญหา และกำหนดผลลัพธ์ในสภาพแวดล้อมที่ลูกค้าต้องการได้อย่างถูกต้อง

สรุป

1. คอมพิวเตอร์คืออุปกรณ์ที่ใช้สำหรับประมวลผล ประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ โดยการติดต่อระหว่างกันใช้ภาษาโปรแกรมที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้เรียกว่าภาษาเครื่อง (machine language) โปรแกรมที่โปรแกรมเมอร์เขียนขึ้นจากภาษาโปรแกรมระดับสูง ต้องผ่านการแปลโดยคอมไพเลอร์เพื่อแปลเป็นภาษาเครื่อง
2. ขั้นตอนในการพัฒนาซอฟต์แวร์ (software development method) แบ่งเป็น 6 ขั้นตอน ดังนี้
 - 2.1 ระบุความต้องการ (problem) โดยสอบถามได้จากบุคคลต่างๆที่เกี่ยวข้อง เพื่อให้ได้ความต้องการที่ดีไม่คลุมเครือ , จำเป็น และถูกต้อง
 - 2.2 วิเคราะห์ปัญหา (analysis) เป็นการวิเคราะห์ถึงปัญหาต่างๆที่เกิดขึ้นจากความต้องการ
 - 2.3 ออกแบบอัลกอริทึมเพื่อแก้ปัญหา (design) เป็นการหาหนทางในการแก้ปัญหา
 - 2.4 สร้างโปรแกรม (implementation) เขียนคำสั่งโปรแกรมเพื่อให้คอมพิวเตอร์ปฏิบัติงาน
 - 2.5 ทดสอบและตรวจสอบความสมบูรณ์ของโปรแกรม (testing)
 - 2.6 บำรุงรักษาและแก้ไขโปรแกรม (maintenance)
3. วิศวกรรมซอฟต์แวร์คือกลยุทธ์ของการผลิตซอฟต์แวร์ที่มีคุณภาพ ซอฟต์แวร์ที่มีคุณภาพสูง ต้องมีคุณลักษณะที่เอื้อประโยชน์ต่อผู้ใช้ , ผู้พัฒนาระบบ และ ผู้บำรุงรักษาระบบ ซอฟต์แวร์ที่มีคุณภาพจะต้องสามารถกระทำงานตามที่ถูกคำสั่งได้อย่างถูกต้อง สามารถปรับเปลี่ยนอุปกรณ์ฮาร์ดแวร์หรือระบบซอฟต์แวร์ได้โดยปราศจากความผิดพลาดหรือความยุ่งยาก ซอฟต์แวร์ที่มีคุณภาพต้องเอื้อประโยชน์ให้กับโปรแกรมเมอร์ โดยสามารถให้โปรแกรมเมอร์ทราบตำแหน่งของ คำสั่งที่ผิดพลาดได้ง่าย เข้าใจและแก้ไขความผิดพลาดได้ง่าย

4. ซอฟต์แวร์ที่มีคุณภาพต้องมีคุณลักษณะดังนี้
 - (1)สามารถทำงานตามที่ใช้ต้องการให้กระทำได้อย่างถูกต้อง
 - (2)ใช้ทรัพยากรคอมพิวเตอร์อย่างถูกต้องและมีประสิทธิภาพ
 - (3)ง่ายต่อการเรียนรู้ และใช้งาน
 - (4)ผู้พัฒนาซอฟต์แวร์ สามารถออกแบบ ถอดรหัส ทดสอบ และ บำรุงรักษาระบบได้ง่าย
5. โดยทั่วไปแล้วผู้ที่เกี่ยวพันในโครงการทั้งหมดจะแบ่งเป็น 3 กลุ่มใหญ่ๆ คือ ลูกค้า ผู้ใช้ระบบ และผู้พัฒนาระบบ

แบบฝึกหัด

1. ขั้นตอนในการพัฒนาซอฟต์แวร์มีกี่ขั้นตอน อะไรบ้าง
2. จงอธิบายคำศัพท์ต่อไปนี้มาพอเข้าใจ
 - 2.1 compiler
 - 2.2 machine language
 - 2.3 linking
 - 2.4 loading
 - 2.5 execution
3. จงวิเคราะห์และออกแบบ การแปลงจำนวนนิ้วโตๆ ให้เป็นจำนวนฟุตและจำนวนนิ้วที่เหลือ
4. จงวิเคราะห์และออกแบบ การแปลงจำนวนหลา จำนวนฟุต และจำนวนนิ้วโตๆ ให้เป็นจำนวนนิ้วทั้งหมด
5. จงวิเคราะห์และออกแบบ การแปลงจำนวนวินาทีโตๆ ให้เป็นจำนวนชั่วโมง จำนวนนาที และจำนวนวินาทีที่เหลือ
6. จงวิเคราะห์และออกแบบ การหาจำนวนแบงค์ที่น้อยที่สุดในการแลกจากจำนวนเงินเหรียญที่หยิบออกมาจากกระปุกออมสิน
7. วิศวกรรมซอฟต์แวร์คืออะไร จงอธิบายพอเข้าใจ
8. ซอฟต์แวร์ที่มีคุณภาพควรมีลักษณะอย่างไร
9. การพัฒนาซอฟต์แวร์ต้องเกี่ยวข้องกับกลุ่มบุคลากรในกลุ่มใดบ้าง จงอธิบายถึงความเกี่ยวข้องพันธะของแต่ละกลุ่มพอเข้าใจ