

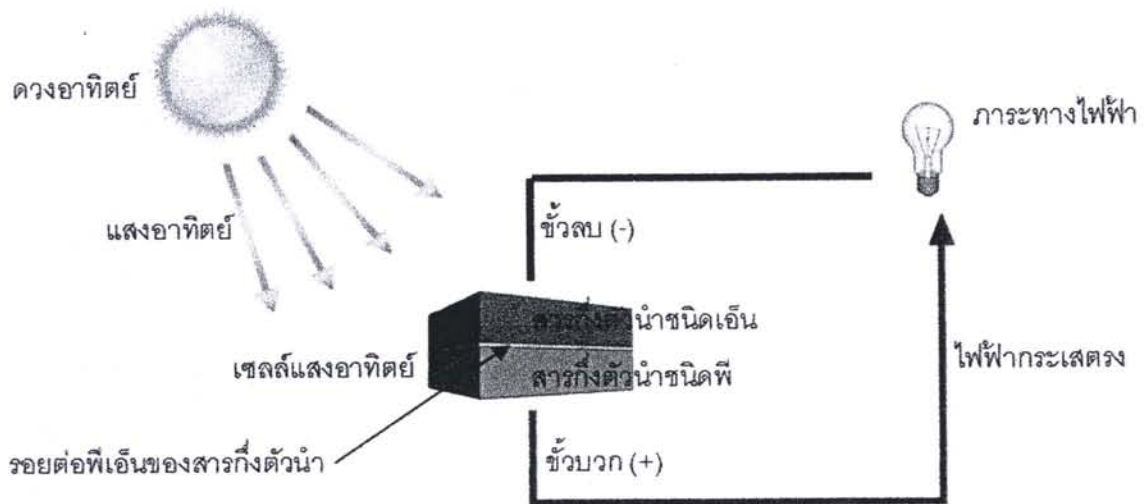
บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

เซลล์แสงอาทิตย์

เซลล์แสงอาทิตย์ คือ สิ่งประดิษฐ์ที่ทำมาจากสารกึ่งตัวนำ (Semiconductor) ที่สามารถเปลี่ยนพลังงานแสงอาทิตย์ให้ไปเป็นไฟฟ้าได้ โดยไฟฟ้าที่ได้นั้นจะเป็นไฟฟ้ากระแสตรง เซลล์แสงอาทิตย์โดยทั่วไป ประกอบด้วยสารกึ่งตัวนำชนิดซิลิกอนที่มีคุณสมบัติแตกต่างกันมาต่อกันแบ่งเป็นชั้น n - type ซิลิกอน ซึ่งอยู่ด้านหน้าของเซลล์ คือ สารกึ่งตัวนำที่ได้รับการเจือปนด้วยสารฟอสฟอรัส มีคุณสมบัติเป็นตัวให้อิเล็กตรอนเมื่อรับพลังงานจากแสงอาทิตย์ และ p - type ซิลิกอน คือสารกึ่งตัวนำที่ได้รับการเจือปนด้วยสารโบรอน เรียกรอยต่อนี้ว่า รอยต่อพีเอ็น หรือ P-N Junction ทำหน้าที่สร้างสนามไฟฟ้าภายในเซลล์แสงอาทิตย์ ซึ่งเมื่อมีแสงอาทิตย์ตกกระทบเซลล์แสงอาทิตย์ จะเกิดพาหะนำไฟฟ้าขึ้นสองชนิด คือ อิเล็กตรอน (ประจุลบ) และ โฮลอิสระ (ประจุบวก) สนามไฟฟ้าบริเวณรอยต่อพีเอ็นจะแยกอิเล็กตรอนที่เกิดขึ้นให้ไหลไปทางขั้วลบ (n-type) และแยกโฮลให้ไหลไปทางขั้วบวก (p-type) มีผลทำให้เกิดแรงดันไฟฟ้าแบบกระแสตรงขึ้นที่ขั้วทั้งสอง ดังนั้นเมื่อเราต่อขั้วดังกล่าวเข้ากับภาระทางไฟฟ้า (เครื่องใช้ไฟฟ้ากระแสตรง) ก็จะทำให้เกิดการไหลของกระแสไฟฟ้าขึ้นภายในวงจร ดังภาพ 1

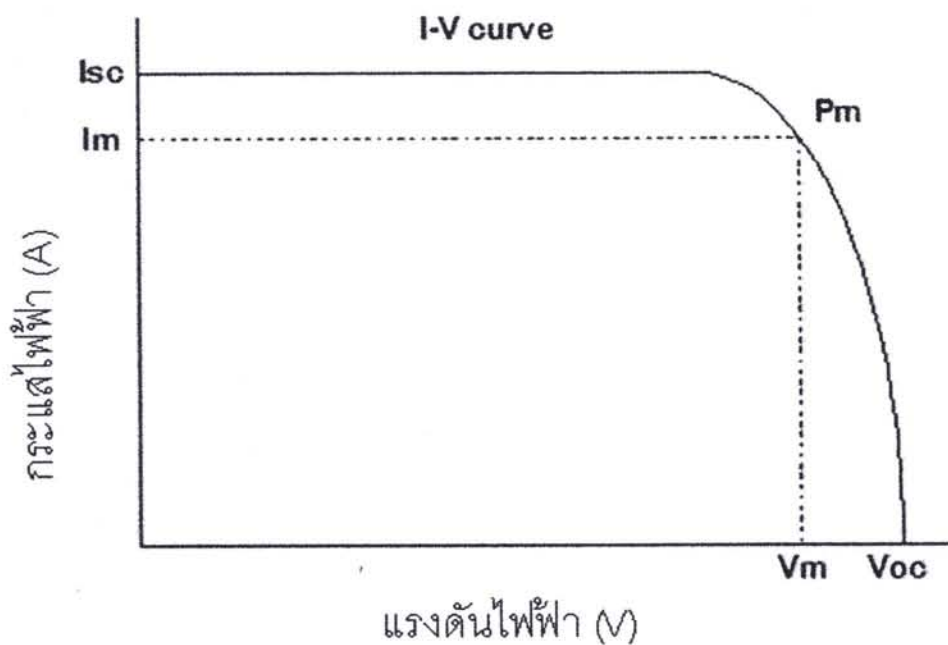
เซลล์แสงอาทิตย์โดยทั่วไปจะให้แรงดันไฟฟ้าประมาณ 0.5 โวลต์ ที่สภาวะวงจรเปิดและไม่มีภาระทางไฟฟ้า ส่วนค่ากระแสไฟฟ้าของเซลล์แสงอาทิตย์จะขึ้นอยู่กับประสิทธิภาพและ ขนาดของเซลล์แสงอาทิตย์ (พื้นที่หน้าตัด) และยังขึ้นอยู่กับค่าความเข้มรังสีแสงอาทิตย์ที่ตกกระทบบนพื้นผิวของเซลล์แสงอาทิตย์ ตัวอย่างเช่น เซลล์แสงอาทิตย์จากโรงงานขนาดพื้นที่หน้าตัด 160 ตารางเซนติเมตร จะสามารถผลิตไฟฟ้าได้สูงสุดประมาณ 2 วัตต์ ที่สภาวะความเข้มแสงอาทิตย์ประมาณ 1,000 วัตต์ต่อตารางเมตร และเมื่อความเข้มแสงอาทิตย์ลดต่ำลงเหลือประมาณ 400 วัตต์ต่อตารางเมตร เซลล์แสงอาทิตย์นี้ก็จะผลิตไฟฟ้าได้เพียงประมาณ 0.8 วัตต์ [2]



ภาพ 1 ลักษณะทั่วไปและการทำงานของเซลล์แสงอาทิตย์

คุณสมบัติทางไฟฟ้าของแผงเซลล์แสงอาทิตย์

คุณสมบัติทางไฟฟ้าของแผงเซลล์แสงอาทิตย์สามารถอธิบายได้โดยใช้กราฟแสดงคุณสมบัติกระแส-แรงดันไฟฟ้าของแผงเซลล์แสงอาทิตย์ (IV-Curve) ลักษณะของ IV-Curve ไม่ว่าจะของเซลล์ (cell) โมดูล (module) อาร์เรย์ (array) จะมีลักษณะที่เดียวกัน ดังแสดงภาพ 2



ภาพ 2 คุณสมบัติกระแสไฟฟ้าและแรงดันไฟฟ้าของเซลล์แสงอาทิตย์

1. ค่ากระแสไฟฟ้าสูงสุด (I_m) คือ ค่ากระแสไฟฟ้าที่จุดกำลังไฟฟ้าสูงสุดของเซลล์แสงอาทิตย์ในขณะที่ยังต่ออยู่กับภาระทางไฟฟ้า
2. ค่าแรงดันไฟฟ้าสูงสุด (V_m) คือค่าแรงดันไฟฟ้าที่จุดกำลังไฟฟ้าสูงสุดของเซลล์แสงอาทิตย์ในขณะที่ยังต่ออยู่กับภาระทางไฟฟ้า
3. ค่ากระแสไฟฟ้าลัดวงจร (I_{sc}) คือค่ากระแสไฟฟ้าของเซลล์แสงอาทิตย์ในขณะที่เกิดการลัดวงจร
4. ค่าแรงดันไฟฟ้าวงจรเปิด (V_{oc}) คือค่าแรงดันไฟฟ้าของเซลล์แสงอาทิตย์ในกรณีที่ไม่มีภาระทางไฟฟ้า
5. ค่ากำลังไฟฟ้าสูงสุด (P_m) คือค่ากำลังไฟฟ้าสูงสุดที่เซลล์แสงอาทิตย์จ่ายออกมาในขณะที่มีภาระทางไฟฟ้า
6. ค่าฟิลล์แฟคเตอร์ (F.F) คือ ค่าอัตราส่วนของกำลังไฟฟ้าสูงสุดต่อผลคูณระหว่างค่ากระแสไฟฟ้าลัดวงจรกับแรงดันไฟฟ้าวงจรเปิดสามารถเขียนเป็นสมการได้ดังต่อไปนี้

$$F.F = P_m / (I_{sc} \times V_{oc}) = (I_m \times V_m) / (I_{sc} \times V_{oc}) \quad (1)$$

เซลล์แสงอาทิตย์ที่ดีควรมีค่าฟิลล์แฟคเตอร์มากกว่า 0.7 ขึ้นไป เพื่อให้จะให้จุดทำงานมีค่าใกล้เคียงกับจุดกำลังไฟฟ้าสูงสุด

7. ประสิทธิภาพสูงสุด (η_m) คือ ค่าอัตราส่วนกำลังไฟฟ้าสูงสุดต่อพลังงานที่ได้รับของแผงเซลล์แสงอาทิตย์ ซึ่งหาค่าได้จากสมการ

$$\eta_m = [P_m / A_m G_T] \times 100\% \quad (2)$$

เมื่อ A_m คือ พื้นที่รับแสงของเซลล์แสงอาทิตย์ (m^2)

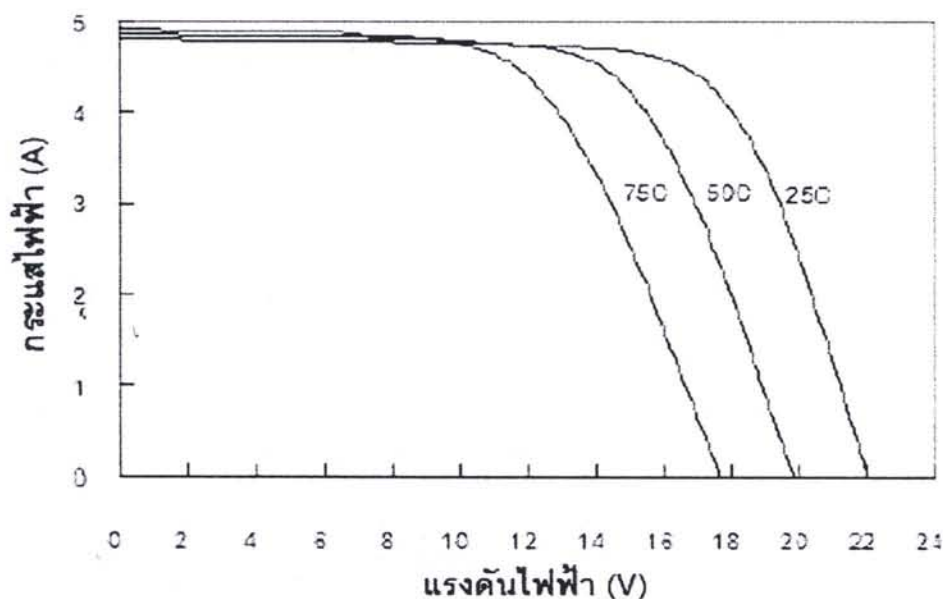
G_T คือ ค่าความเข้มรังสีดวงอาทิตย์ (W/m^2)

ในกรณีที่รูปกราฟ IV-Curve เปลี่ยนแกนไปอยู่ใน Quadrant ที่สองหรือสี่ (ค่าแรงดันและกระแสไฟฟ้าเป็นลบ) จะหมายความว่าเกิดการสูญเสียกำลังไฟฟ้า ซึ่งสาเหตุอาจเกิดจากการที่เซลล์แสงอาทิตย์มีอุณหภูมิที่สูงมาก และใกล้ที่จะเสียหาย

โดยปกติการทดสอบคุณสมบัติทางไฟฟ้าของเซลล์แสงอาทิตย์เพื่อรับรองคุณภาพจะกระทำที่เงื่อนไขมาตรฐานเรียกว่า Standard Testing Condition (STC) ซึ่งเงื่อนไขดังกล่าวเป็นการทดสอบเพื่อหาค่าคุณสมบัติทางไฟฟ้าของเซลล์แสงอาทิตย์ภายใต้เงื่อนไขการทดสอบคือ ค่าความเข้มรังสีดวงอาทิตย์ $1,000 W/m^2$ ที่ความยาวอากาศ 1.5 และ อุณหภูมิเซลล์ $25^\circ C$

เซลล์แสงอาทิตย์จะมีลักษณะเฉพาะที่น่าสนใจ คือ กระแสไฟฟ้าที่ผลิตได้จะเพิ่มขึ้นอย่างสม่ำเสมอเมื่อความเข้มรังสีดวงอาทิตย์เพิ่มขึ้น ซึ่งส่งผลต่อ IV-Curve ของเซลล์แสงอาทิตย์ที่

สภาวะความเข้มรังสีอาทิตย์คงที่ กับที่สภาวะอุณหภูมิแตกต่างกัน กำลังไฟฟ้าที่ผลิตได้ของเซลล์แสงอาทิตย์จะตกลงเมื่ออุณหภูมิเซลล์แสงอาทิตย์สูงขึ้น ส่วนค่ากระแสนั้นจะเพิ่มขึ้นเล็กน้อยในขณะที่ค่าแรงดันไฟฟ้าจะตกลงอย่างมาก [3]



ภาพ 3 คุณสมบัติกระแสไฟฟ้าและแรงดันไฟฟ้าของเซลล์แสงอาทิตย์ เมื่อค่าความเข้มรังสีดวงอาทิตย์คงที่และอุณหภูมิเซลล์เพิ่มขึ้น

หลักการในกระบวนการทางวิศวกรรมซอฟต์แวร์

วิศวกรรมซอฟต์แวร์ หมายถึง การนำแนวทางหรือหลักการทางวิศวกรรมมาประยุกต์ใช้ในการพัฒนา ปฏิบัติการ และบำรุงรักษาซอฟต์แวร์อย่างเป็นระบบ มีระเบียบ กฎเกณฑ์ และสามารถวัดผลในเชิงปฏิบัติได้ [4]

นักพัฒนาจำเป็นต้องใช้หลักการทางวิศวกรรมในการพัฒนาระบบซอฟต์แวร์ ซึ่งโครงสร้างของซอฟต์แวร์จะไม่เหมือนกับผลงานทางวิศวกรรมด้านอื่นๆ การพัฒนาจึงมีกระบวนการทางวิศวกรรมเป็นของตัวเอง กระบวนการทางวิศวกรรมซอฟต์แวร์ที่ดี จะต้องสามารถทำให้การพัฒนาซอฟต์แวร์เสร็จได้ทันเวลา โดยใช้งบประมาณตามที่ตั้งไว้และซอฟต์แวร์ตรงตามความต้องการของผู้ใช้ [5]

ในการพัฒนา ผู้พัฒนาจำเป็นต้องใช้หลักการต่างๆ ประกอบกันเพื่อให้ได้ซอฟต์แวร์ที่ดี ซึ่งหลักการในการพัฒนา ประกอบไปด้วยหลักการต่อไปนี้ [4]

1. ความสามารถในการบำรุงรักษา (Maintainability)

ซอฟต์แวร์จะต้องง่ายต่อการบำรุงรักษา สามารถเปลี่ยนแปลง ปรับเปลี่ยนให้เหมาะสม และตอบสนองได้อย่างรวดเร็วและทันท่วงที ในกรณีที่เกิดปัญหาหรือข้อผิดพลาดจากความ ต้องการที่เปลี่ยนแปลงของผู้ใช้ซอฟต์แวร์

2. ความน่าเชื่อถือ (Dependability)

ซอฟต์แวร์จะต้องคงไว้ซึ่งความสามารถในการสร้างความน่าเชื่อถือ ได้อย่างมีประสิทธิภาพ โดยจะต้องผ่านกระบวนการทดสอบและตรวจรับการทำงานของฟังก์ชันทั้งหมด รวมไปถึงการป้องกันและรักษาความปลอดภัยอันเกิดจากสถานการณ์ที่ไม่คาดคิดมาก่อน

3. ประสิทธิภาพ (Efficiency)

ซอฟต์แวร์ที่พัฒนาต้องมีค่าใช้จ่ายที่ประหยัด หรือสิ้นเปลืองน้อยที่สุด สามารถใช้ทรัพยากรต่างๆ ได้อย่างคุ้มค่าและเหมาะสม ในระดับที่ไม่เกินขีดความสามารถของทรัพยากรที่มีอยู่ ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ ขนาดของพื้นที่จัดเก็บข้อมูล ความรวดเร็วในการประมวลผล หรือแม้แต่ความรวดเร็วในการตอบสนองกับผู้ใช้งาน ทั้งนี้ขึ้นอยู่กับโครงสร้างหรือสถาปัตยกรรมของซอฟต์แวร์ที่ถูกออกแบบไว้

4. ความสามารถในการใช้งาน (Usability)

ซอฟต์แวร์จะต้องสะดวกและง่ายต่อการใช้งาน สามารถทำให้ผู้ใช้เรียนรู้การทำงานของซอฟต์แวร์ได้อย่างรวดเร็ว ไม่ว่าจะเป็นการออกแบบจอภาพที่นำทางการใช้งานของผู้ใช้ได้ หรือแม้แต่มือประกอบการติดตั้งและใช้งานที่เหมาะสม เป็นต้น

คุณลักษณะของโปรแกรมที่ดี [6]

1. มีความถูกต้องเชื่อถือได้ (Correctness and Reliability)

โปรแกรมคอมพิวเตอร์ที่ดีต้องให้ผลลัพธ์ที่ถูกต้องแม่นยำ ไม่คลาดเคลื่อน โปรแกรมคอมพิวเตอร์จึงต้องมีความสมบูรณ์มากที่สุด คือ ผ่านการทดสอบที่ครอบคลุม การทำงานของโปรแกรมคอมพิวเตอร์ต้องนิ่ง มีเสถียรภาพ ไม่ค่อยเกิดปัญหา เพราะอาจถูกนำไปใช้ในการตัดสินใจสำคัญของผู้ใช้งาน

2. ความเป็นมิตรต่อผู้ใช้ (User-friendliness)

ปัจจุบันมีผู้ใช้โปรแกรมคอมพิวเตอร์ในการทำงานจำนวนมากขึ้น ในจำนวนนี้มีผู้ใช้ที่เป็นผู้เริ่มต้นหรือไม่มีพื้นฐานทางด้านคอมพิวเตอร์เลย และมีแนวโน้มเพิ่มมากขึ้น การสร้างโปรแกรมคอมพิวเตอร์ให้ผู้ใช้สามารถใช้งานได้ง่ายและสะดวกจึงเป็นเรื่องที่สำคัญ

3. ค่าใช้จ่ายต่ำ (Low Cost)

ก่อนการพัฒนาโปรแกรมคอมพิวเตอร์ ต้องมีการวางแผนและประเมินค่าใช้จ่าย เมื่อมีการพัฒนาโปรแกรมคอมพิวเตอร์ก็ต้องควบคุมค่าใช้จ่ายให้เป็นไปตามแผนค่าใช้จ่ายในการพัฒนา ซึ่งในมุมมองของผู้ใช้โปรแกรมคอมพิวเตอร์จะมองว่าโปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นคุ้มกับค่าใช้จ่าย ส่วนในมุมมองของผู้พัฒนาโปรแกรมคอมพิวเตอร์ ค่าใช้จ่ายในการพัฒนาจะต้องต่ำกว่าราคาที่เสนอให้ผู้ใช้งานโปรแกรมคอมพิวเตอร์

4. ต้องอ่านง่ายและสามารถนำกลับมาใช้ใหม่ (Readability and Reusability)

โปรแกรมคอมพิวเตอร์ที่สามารถทำงานได้หลายอย่างหรือมีความสามารถมาก มักมีโครงสร้างที่ใหญ่และซับซ้อน จึงควรมีการออกแบบเป็นโมดูล(Module) ย่อยที่มีอิสระต่อกันและมีการเขียนโปรแกรมคอมพิวเตอร์แบบเรียบง่าย ไม่ยุ่งยากซับซ้อน เพื่อให้ผู้พัฒนาโปรแกรมคอมพิวเตอร์คนอื่นสามารถเข้าใจสามารถนำไปพัฒนาต่อให้เหมาะกับยุคสมัยได้ง่าย

5. มีความปลอดภัย (Security)

ข้อมูลสำคัญต่างๆส่วนใหญ่มีการเก็บไว้ในคอมพิวเตอร์กันมากขึ้น ทำให้เกิดความเสี่ยงในเรื่องของความปลอดภัยของข้อมูล เช่น การแก้ไขข้อมูลโดยไม่ได้รับอนุญาต การลักลอบขโมยข้อมูล รวมไปถึงการทำลายข้อมูลได้ โปรแกรมคอมพิวเตอร์ที่ดีจึงต้องมีความปลอดภัยในการเข้าถึงข้อมูลต่างๆสูง

6. ใช้เวลาในการพัฒนาไม่นาน

ปัจจุบันเทคโนโลยีและวิธีการทำงานเปลี่ยนแปลงไปอย่างรวดเร็ว ทำให้ไม่สามารถใช้เวลานานเกินไปในการพัฒนาโปรแกรมคอมพิวเตอร์ ไม่เช่นนั้น กว่าโปรแกรมคอมพิวเตอร์จะเสร็จความต้องการในการใช้โปรแกรมคอมพิวเตอร์ที่ทำการพัฒนาขึ้น อาจมีการเปลี่ยนแปลงไป และที่สำคัญโปรแกรมที่พัฒนาขึ้นต้องเป็นไปตามระยะเวลาที่ประเมินไว้

การออกแบบหน้าจอแสดงผลสำหรับโปรแกรม

การออกแบบหน้าจอแสดงผลหรือการออกแบบกราฟิกเป็นการเรียนรู้และใช้ทักษะด้านการออกแบบ ที่เกี่ยวข้องกับการสื่อสารรอบตัวที่มองเห็นด้วยสายตา การออกแบบการสื่อสารต่างๆ ที่ต้องการการออกแบบ เพื่อให้ได้ผลที่สมบูรณ์ที่สุด ผ่านแนวความคิด เป็นการประยุกต์ใช้ศิลปะให้สอดคล้องกับการออกแบบ ด้วยหลักการทฤษฎีและกลวิธีการต่างๆ [7]

ความหมายของกราฟิก

ความหมาย "กราฟิก" (Graphic) หมายถึงการขีด เขียนบันทึก ภาพ ด้วยวัสดุ และสื่อ มีความหมายทั้งการเขียนด้วยตัวหนังสือและการสื่อความหมายโดยการใช้เส้นวัสดุกราฟิกจะ

หมายถึงวัสดุใดๆ ที่แสดงความต้องการสื่อสารความคิด โดยใช้ภาพวาด ภาพเขียน และอักษร ข้อความรวมกัน ภาพวาดอาจจะเป็น แผนภาพ(Diagram) ภาพสเก็ตช์ (Sketch) หรือแผนสถิติ (Graph) หรืออาจเป็นคำที่ใช้เป็นหัวเรื่อง (Title) คำอธิบายเพิ่มเติมของแผนภูมิ แผนภาพ แผนสถิติ และภาพโฆษณา อาจวาดเป็นการ์ตูนประเภทต่างๆ ภาพสเก็ตช์ สัญลักษณ์ และภาพถ่าย สามารถใช้เป็นวัสดุกราฟิกเพื่อสื่อความหมายในเรื่องราวข้อเท็จจริงต่างๆได้ ปัจจุบันการออกแบบกราฟิกใช้คอมพิวเตอร์มาช่วยในการออกแบบมากขึ้น [7]

หลักการ KISS : Keep It Simple and Straightforward

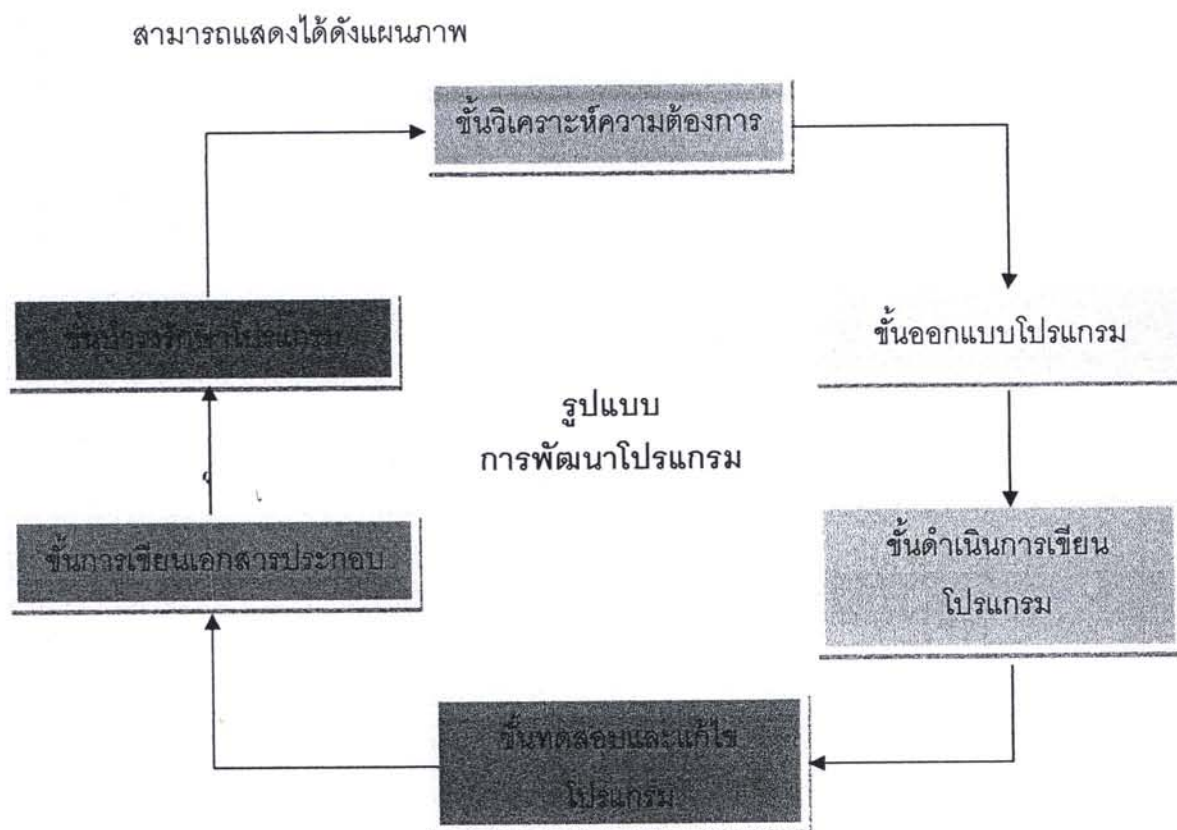
หลักการ KISS หรือ Keep It Simple and Straightforward เป็นหลักการที่มีความเรียบง่าย เข้าใจง่าย ไม่ซับซ้อน บางครั้งใช้คำว่า Keep It Simple, Stupid หรือ Keep It Short and Simple ตามหลักการ KISS จะคิดถึงความเรียบง่ายและหลีกเลี่ยงสิ่งที่ซับซ้อนทั้งหลายเป็นหลัก ซึ่งหลักการ KISS เป็นการมองที่เป้าหมายเพื่อให้ผลลัพธ์ที่ออกมาใช้ทรัพยากรที่ประหยัดกว่า แต่ทำงานได้เร็วกว่าและดีกว่าการใช้สามัญสำนึกในการแก้ไขปัญหาที่ยุ่งยากซับซ้อน แนวคิดนี้ทำให้สามารถประหยัดเวลา เงินลงทุนและกำลังคนที่ใช้ในการทำงาน [8]

หลักการ KISS สามารถช่วยให้ออกแบบได้เร็ว มีความยืดหยุ่น ใช้งานได้หลากหลาย ใช้ทรัพยากรน้อยแต่ทำงานได้ดี ไม่ยึดติดกับฐานข้อมูล และง่ายในการออกแบบและเขียนโปรแกรม ซึ่งการใช้หลักการ KISS ในการออกแบบ ทำให้หน้าจอแสดงผลง่ายในการทำความเข้าใจ การปรับปรุงให้มีขนาดใหญ่ขึ้น การปรับปรุงให้มีความเหมาะสมในการใช้งานและการบูรณาการในการนำไปใช้งานต่างๆ [9]

วงจรการพัฒนาโปรแกรม [6]

ในการพัฒนาโปรแกรม ผู้พัฒนาโปรแกรมจะต้องมีการเตรียมงานเกี่ยวกับการพัฒนาโปรแกรมอย่างเป็นขั้นตอน เรียกขั้นตอนเหล่านี้ว่า วงจรการพัฒนาโปรแกรม (Program Development Life Cycle: PDLC) ประกอบด้วย 6 ขั้นตอน ดังนี้

1. ขั้นวิเคราะห์ความต้องการ (Requirement Analysis and Feasibility Study)
2. ขั้นตอนออกแบบโปรแกรม (Program Design)
3. ขั้นตอนการเขียนโปรแกรม (Program Coding)
4. ขั้นตอนทดสอบและแก้ไขโปรแกรม (Program Testing and Debugging)
5. ขั้นตอนการเขียนเอกสารประกอบ (Program Documentation)
6. ขั้นบำรุงรักษาโปรแกรม (Program Maintenance)



ภาพ 4 วงจรการพัฒนาโปรแกรม

รูปแบบหรือขั้นตอนในการพัฒนาโปรแกรม จะช่วยให้การพัฒนาโปรแกรมดำเนินการได้อย่างมีประสิทธิภาพและได้ผลลัพธ์ตามที่มุ่งหมาย เพราะแต่ละขั้นตอนจะช่วยให้เกิดความเป็นระเบียบ การเรียบเรียงแนวคิดมีความชัดเจน ไม่สับสน และเกิดความง่ายต่อการพัฒนาโปรแกรม ถึงแม้ปัจจุบันจะมีวิธีการพัฒนาโปรแกรมใหม่ๆ เกิดขึ้น แต่รูปแบบหรือขั้นตอนในการพัฒนาโปรแกรม ยังเป็นสิ่งจำเป็นสำหรับการพัฒนาโปรแกรม เพราะจะสามารถช่วยให้แนวคิดในการพัฒนาโปรแกรมเป็นระเบียบไม่สับสน

1. ขั้นวิเคราะห์ความต้องการ (Requirement Analysis and Feasibility Study)

ความต้องการ หมายถึง เงื่อนไขหรือความสามารถที่โปรแกรมควรจะต้องมี ส่วนการวิเคราะห์ความต้องการ คือ การค้นหา จัดการ ทำเอกสารความต้องการของโปรแกรมและการพัฒนาโปรแกรม และดูแลความต้องการระหว่างผู้ใช้โปรแกรมกับผู้พัฒนาโปรแกรมในการเปลี่ยนแปลงความต้องการของโปรแกรม [5]

การวิเคราะห์ความต้องการ ในงานวิจัยนี้ เป็นการทำความเข้าใจกับปัญหาและค้นหาสิ่งที่ต้องการ นั่นคือการศึกษาวเคราะห์โดยละเอียดว่า ต้องการผลลัพธ์อะไร ต้องใช้ข้อมูลอะไรเพื่อให้ได้ผลลัพธ์ที่ต้องการ และมีขั้นตอนในการประมวลผลอย่างไร อาจเรียกขั้นตอนนี้ว่า การวิเคราะห์ปัญหาหรือการวิเคราะห์งาน ถือเป็นขั้นตอนที่สำคัญที่สุด ก่อนที่จะดำเนินวางแผนแก้ปัญหาหรือเขียนโปรแกรม

การวิเคราะห์ความต้องการ เป็นการศึกษ ผลลัพธ์(Output) ข้อมูลนำเข้า (Input) และวิธีการประมวลผล (Process) ที่จะใช้ในการพัฒนาโปรแกรม สามารถแบ่งวิธีการเป็น 5 หัวข้อหลัก [6] ดังนี้

1.1 สิ่งที่ต้องการ เป็นการพิจารณาเกี่ยวกับงานที่ต้องการให้โปรแกรมที่พัฒนาทำและผลลัพธ์ที่ต้องการสำหรับงานที่ต้องการผลลัพธ์มากกว่าหนึ่งอย่าง จะเขียนแยกเป็นข้ออย่างชัดเจน

1.2 รูปแบบของผลลัพธ์ เป็นการศึกษาลักษณะของผลลัพธ์ที่ต้องการให้โปรแกรมแสดงออกมา เช่น แสดงออกทางจอภาพ เป็นเสียง หรือพิมพ์ออกทางเครื่องพิมพ์ และอาจต้องมีการกำหนดรายละเอียดอย่างชัดเจน เช่น การแสดงผลทางจอภาพ ต้องมีรายการอะไรบ้าง เช่น มีการแสดงเป็นกราฟระหว่างค่าที่ต้องการ มีการวิเคราะห์ผลจากกราฟ หรือมีการแสดงผลจากการคำนวณอะไรบ้าง จึงจำเป็นที่ผู้วิเคราะห์งานต้องทำความเข้าใจกับผู้ที่นำโปรแกรมไปใช้งาน

1.3 ข้อมูลนำเข้า เมื่อทราบรูปแบบของผลลัพธ์ที่ต้องการแล้ว จะต้องพิจารณาถึงข้อมูลที่จะนำมาเป็นข้อมูลเข้า (Input) ว่าครบถ้วนสำหรับการประมวลผลหรือไม่ ซึ่งจะต้องคำนึงถึงประเภทของข้อมูลด้วย เช่น เลขจำนวนเต็มหรือทศนิยม เป็นต้น

1.4 กำหนดตัวแปรที่ใช้ เป็นการกำหนดชื่อแทนความหมายของข้อมูลนำเข้า เพื่อความสะดวกต่อการนำเอาข้อมูลไปวิเคราะห์หาขั้นตอนการประมวลผล มีกฎการตั้งชื่อตัวแปร ดังนี้

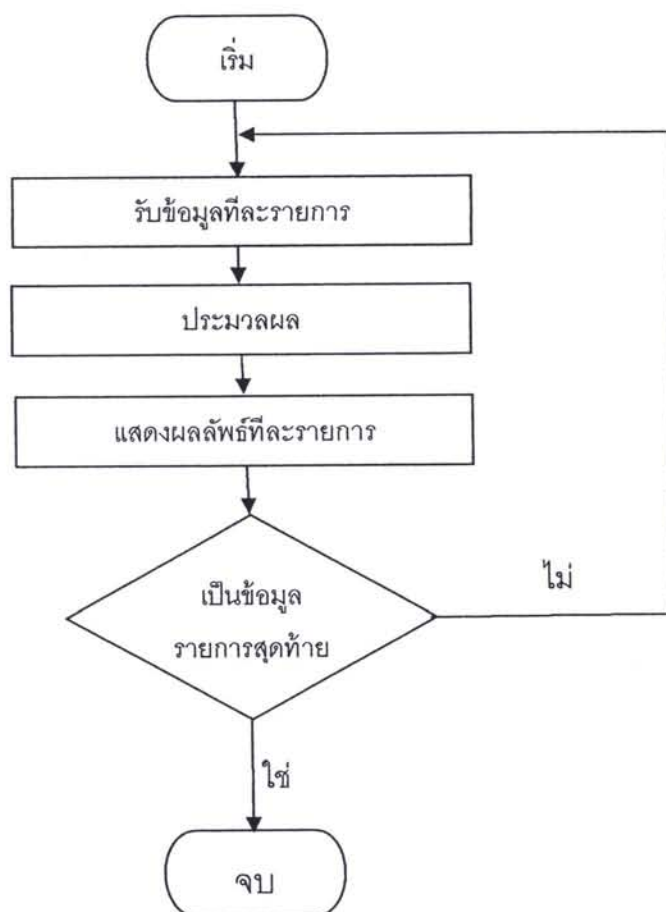
1.4.1 ตั้งชื่อตัวแปรตรงกับกฎการตั้งชื่อตัวแปรของภาษาที่จะใช้ในการพัฒนาโปรแกรม

1.4.2 ต้องตั้งชื่อตัวแปรไม่ตรงกับคำศัพท์เฉพาะของภาษาที่เขียน เช่น คำสั่งหรือฟังก์ชันการทำงานของภาษาโปรแกรมที่ใช้ในการพัฒนา

1.4.3 ตั้งชื่อตัวแปรให้สื่อความหมายกับค่าที่ใช้แทนค่า เช่น V หมายถึงแรงดันไฟฟ้า

1.5 วิธีการประมวลผล คือ ขั้นตอนการประมวลผลเพื่อให้ได้ผลลัพธ์ตามรูปแบบที่ต้องการโดยใช้ข้อมูลนำเข้าและตัวแปรต่างๆที่ได้กำหนดไว้ ขั้นตอนวิธีการประมวลผล ขึ้นอยู่กับการแสดงผลลัพธ์ โดยทั่วไปมี 2 วิธี คือ

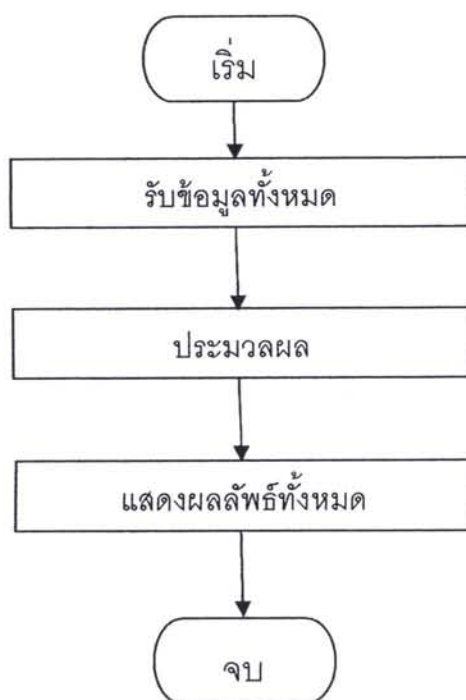
1.5.1 รับข้อมูลเข้าสู่โปรแกรมทีละรายการ แล้วประมวลผลและแสดงผลลัพธ์ทีละรายการ เมื่อรับข้อมูลแรกเข้ามา โปรแกรมจะดำเนินการประมวลผลและแสดงผลลัพธ์ออกมา หลังจากนั้นก็ทำการรับข้อมูลใหม่เข้ามา เพื่อนำไปประมวลผลและแสดงผลลัพธ์อีก ทำแบบนี้ไปเรื่อยๆจนกว่าจะหมดชุดข้อมูลที่ต้องการประมวลผล สามารถอธิบายวิธีการประมวลผลดังภาพ 5



ภาพ 5 การรับข้อมูลทีละรายการ เพื่อประมวลผล

1.5.2 รับข้อมูลทั้งหมดเข้าสู่โปรแกรมในครั้งเดียว และประมวลผลในครั้งเดียว การรับข้อมูลนำเข้าวิธีนี้จะทำเพียงครั้งเดียว จากนั้นโปรแกรมจะทำการประมวลผลเพื่อแสดงผลลัพธ์ทั้งหมดออกมา วิธีนี้จะต้องใช้หน่วยความจำมากขึ้นเพื่อทำการประมวลผล สามารถแสดงได้ดังแผนภาพ 6

สำนักงานคณะกรรมการวิจัยแห่งชาติ	
ห้องสมุดวิจัย	
วันที่.....	๐ 5 ต.ค. 2555
เลขทะเบียน.....	249075
เลขเรียกหนังสือ.....	



ภาพ 6 การรับข้อมูลทั้งหมดครั้งเดียว เพื่อประมวลผล

2. ขั้นตอนออกแบบโปรแกรม (Program Design)

การออกแบบโปรแกรมเป็นการจัดลำดับการทำงานของโปรแกรมตามขั้นตอนการประมวลผล และต้องสอดคล้องกับวิธีการทำงานของโปรแกรมที่ทำการพัฒนา ขั้นตอนนี้นักเรียกว่า การออกแบบอัลกอริทึม ซึ่งเป็นแนวทางที่ทำให้นักพัฒนาทำงานได้ง่ายขึ้น โดยมีหลายวิธีที่นำมาช่วยในการออกแบบ เช่น การเขียนอัลกอริทึม การเขียนชุดูโค้ด (Pseudo Code) หรือ รหัสเทียม และการเขียนผังงาน (Flowchart) เป็นต้น

2.1. การเขียนอัลกอริทึม อัลกอริทึม หมายถึง [6] ลำดับขั้นตอนเชิงคำนวณซึ่งแปลข้อมูลขาเข้าของปัญหา ไปเป็นผลลัพธ์ที่ต้องการ ซึ่งขั้นตอนต่างๆในอัลกอริทึมต้องเป็นขั้นตอนที่ใช้หลักการคำนวณ หรือขั้นตอนต่างๆในอัลกอริทึมสามารถแปลงไปเป็นคำสั่งที่ทำงานด้วยภาษาที่โปรแกรมหรือซอฟต์แวร์เข้าใจได้

การเขียนอัลกอริทึมเป็นการนำเอาลำดับขั้นตอนการทำงานที่ออกแบบไว้ มาเขียนในรูปแบบภาษาเขียนหรืออาจเขียนในรูปแบบภาษาอังกฤษที่จัดโครงสร้างไว้ โดยเน้นที่ลำดับขั้นตอนการทำงานเป็นหลัก จึงมีการตัดทอนรายละเอียดปลีกย่อยออก

2.2. การเขียนผังงาน เป็นการนำเอาขั้นตอนการทำงานมาเขียนเป็นแผนภาพหรือสัญลักษณ์ โดยเปลี่ยนจากคำพูดหรือข้อความในการวิเคราะห์งานมาเป็นรูปภาพ แล้วใช้คำอธิบายข้อความนั้นเขียนเพิ่มเติมในรูปหรือสัญลักษณ์ดังกล่าว

2.3 การเขียนซูดโคด (Pseudo Code) ซูดโคด หรือ รหัสเทียม เป็นการนำคำภาษาอังกฤษและคำภาษาคอมพิวเตอร์ที่เข้าใจง่ายมาแสดงลำดับการทำงานของโปรแกรม เพื่อเป็นแนวทางให้นักพัฒนาโปรแกรมนำไปแปลงเป็นภาษาโปรแกรมคอมพิวเตอร์ต่อไป ซึ่งจะช่วยให้นักพัฒนาทำงานได้ง่ายขึ้นกว่าการอ่านสัญลักษณ์ภาพจากผังงาน นอกจากนี้ยังเป็นต้นแบบให้นักพัฒนาโปรแกรมสามารถนำไปเขียนด้วยภาษาต่างๆได้

3. ขั้นตอนการเขียนโปรแกรม (Program Coding)

การเขียนโปรแกรม เป็นการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่งขึ้นมา โดยเลือกใช้ภาษาที่เหมาะสมกับลักษณะและประเภทของงาน และความถนัดของผู้พัฒนา

4. ขั้นตอนทดสอบและแก้ไขโปรแกรม (Program Testing and Debugging)

เป็นการตรวจสอบข้อผิดพลาดของโปรแกรมหรือเรียกว่า บั๊ก (Bugs) ที่เกิดขึ้น และดำเนินการแก้ไขข้อผิดพลาดนั้น หรือเรียกว่า Debugs โปรแกรมที่ทำงานไม่ตรงตามวัตถุประสงค์ของการพัฒนา เรียกว่า โปรแกรมมี Error ซึ่งการเกิด Error ของโปรแกรมมาจาก 2 สาเหตุหลัก คือ

4.1 ข้อผิดพลาดทางไวยากรณ์ของภาษา (Syntax Error) คือ ข้อผิดพลาดที่เกิดจากการเขียนโปรแกรมคำสั่ง (Source Code) ไม่ตรงกับไวยากรณ์ (Syntax) ของภาษาที่ใช้พัฒนา ข้อผิดพลาดนี้เป็นข้อผิดพลาดที่ง่ายต่อการตรวจสอบและแก้ไข เนื่องจากในระหว่างที่มีการแปลภาษาโปรแกรม ถ้ามีข้อผิดพลาดประเภทนี้อยู่ ตัวแปลภาษาโปรแกรมจะแจ้งให้ทราบทันที

4.2 ข้อผิดพลาดที่เกิดจากการตีความหมายของปัญหาผิดไป (Logical Error) เป็นข้อผิดพลาดที่เกิดจากการออกแบบโปรแกรมให้ทำงานผิดจากวัตถุประสงค์หรือความต้องการ การแก้ไขข้อผิดพลาดประเภทนี้จะต้องทำการวิเคราะห์ปัญหาใหม่เพื่อหาข้อผิดพลาดนั้นให้พบ

5. ขั้นตอนการเขียนเอกสารประกอบ

เอกสารประกอบ คือ การอธิบายรายละเอียดของโปรแกรมว่า จุดประสงค์ของโปรแกรมคืออะไร มีขั้นตอนการทำงานของโปรแกรมเป็นอย่างไร ผู้พัฒนาโปรแกรมที่ดีควรมีเอกสารประกอบโปรแกรมทุกขั้นตอนตั้งแต่การออกแบบโปรแกรม จนถึงการทดสอบโปรแกรม เนื่องจากอาจต้องมีการนำโปรแกรมนั้นมาปรับปรุง เปลี่ยนแปลงเพื่อให้ตรงกับความต้องการ ผู้พัฒนาที่รับช่วงต่อ เมื่อดูเอกสารแล้วจะสามารถทำความเข้าใจได้ง่ายขึ้น

เอกสารประกอบโปรแกรมมี 2 แบบ คือ

5.1 คู่มือผู้ใช้ (User Manual) คู่มือนี้จะเน้นเรื่องการใช้งานโปรแกรมเป็นหลัก เช่น วิธีการเปิดใช้งานโปรแกรม โปรแกรมทำงานอย่างไร ป้อนข้อมูลอย่างไร

5.2 คู่มือสำหรับผู้พัฒนา (Programmer Manual) คู่มือนี้จะอธิบายรายละเอียดต่างๆ เกี่ยวกับการพัฒนาโปรแกรมเป็นหลัก เช่น ใช้ภาษาอะไรในการพัฒนา โปรแกรมหลักทำหน้าที่อะไร แต่ละโปรแกรมย่อยทำหน้าที่อะไร คู่มือนี้จะมีประโยชน์ต่อการปรับปรุงแก้ไขโปรแกรมในอนาคต

6. ขั้นบำรุงรักษาโปรแกรม

เป็นขั้นตอนหลังจากที่ใช้งานโปรแกรมที่พัฒนาขึ้นไประยะหนึ่ง หากโปรแกรมที่ทำการพัฒนาทำงานไม่ตรงตามความต้องการของผู้ใช้ ผู้พัฒนาจำเป็นต้องแก้ไขหรือปรับปรุงให้โปรแกรมที่พัฒนาขึ้นสามารถใช้งานได้ดีและโปรแกรมที่พัฒนาดังกล่าวอยู่ในสภาพที่พร้อมใช้งานได้เสมอ

การพัฒนาโปรแกรม

โปรแกรม MATLAB

MATLAB [10] คือ โปรแกรมการคำนวณเชิงตัวเลขที่มีสิ่งแวดล้อมในการคำนวณของตัวเอง (Numerical Computing Environment) และมีภาษาเฉพาะตัวในการเขียนโปรแกรมได้ โดย MATLAB มาจากคำ 2 คำรวมกัน คือ Matrix และ Laboratory ซึ่งหมายถึง ห้องปฏิบัติการเมตริกซ์

MATLAB [11] เป็นโปรแกรมคอมพิวเตอร์สมรรถนะสูงเพื่อใช้ในการคำนวณทางเทคนิค MATLAB ได้รวมการคำนวณ การเขียนโปรแกรมและการแสดงผลรวมกันอยู่ในตัวโปรแกรมเดียวได้อย่างมีประสิทธิภาพ และอยู่ในลักษณะที่ง่ายต่อการใช้งาน นอกจากนี้ลักษณะของการเขียนสมการในโปรแกรมก็จะเหมือนการเขียนสมการคณิตศาสตร์ที่เราคุ้นเคยดีอยู่แล้ว ลักษณะงานทั่วไปที่ใช้ MATLAB เช่น การคำนวณทั่วไป การสร้างและการทดสอบแบบจำลอง การวิเคราะห์ข้อมูล การแสดงผลในรูปกราฟทั้งโดยทั่วไปและกราฟทางด้านทางวิทยาศาสตร์และวิศวกรรม สามารถสร้างโปรแกรมในลักษณะที่ติดต่อกับผู้ใช้ทางกราฟฟิกส์

การทำงานของ MATLAB จะสามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง (Interactive) คือการเขียนคำสั่งเข้าไปทีละคำสั่ง เพื่อให้ MATLAB ประมวลผลไปเรื่อยๆ หรือสามารถที่จะรวบรวมชุดคำสั่งเหล่านั้นเป็นโปรแกรมก็ได้ ข้อสำคัญอย่างหนึ่งของ MATLAB ก็คือ ข้อมูลทุกตัวจะถูกเก็บในลักษณะของอาร์เรย์ (Array) คือในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็กๆ ขึ้น (หรือจะได้รับการแบ่งเป็น element นั่นเอง) ซึ่งการใช้ตัวแปรเป็นอาร์เรย์ใน MATLAB นี้เราไม่จำเป็นต้องจอง dimension เหมือนกับการเขียนโปรแกรมในภาษาโปรแกรมทั่วไป ซึ่งทำให้เราสามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะของ matrix และ vector ได้

โดยง่าย ซึ่งทำให้เราลดเวลาการทำงานลงได้อย่างมากเมื่อเทียบกับการเขียนโปรแกรมโดยภาษาอื่น

MATLAB เป็นโปรแกรมที่ใช้กันอย่างแพร่หลายในแวดวงของนักวิทยาศาสตร์และวิศวกรในปัจจุบัน โดย MATLAB นั้นได้เริ่มต้นขึ้นเพื่อต้องการให้เราสามารถแก้ปัญหาตัวแปรที่มีลักษณะเป็น Matrix ได้ง่ายขึ้น สำหรับในปัจจุบันนี้ MATLAB ได้ถูกพัฒนาโปรแกรมให้มีประสิทธิภาพสูงขึ้น ซึ่งทำให้ทุกวันนี้ MATLAB เป็นโปรแกรมที่สุดยอดในการคำนวณที่คำนวณด้านmatrix สำหรับงานทางวิทยาศาสตร์และวิศวกรรมโปรแกรมหนึ่ง

นอกเหนือจากตัวโปรแกรม MATLAB แล้ว ยังมีการพัฒนากล่องเครื่องมือหรือที่เรียกว่า toolbox ซึ่งเป็นโปรแกรมที่เขียนขึ้นเพื่อประกอบกับการใช้ MATLAB สำหรับงานที่จำเพาะเจาะจงหลายประเภท กล่องเครื่องมือที่มีนั้นเป็นการนำเอาโปรแกรมที่เขียนขึ้นเป็นฟังก์ชันสำหรับ MATLAB เพื่อรวมเข้าเพื่อให้ผู้ใช้งานมีความสะดวกในการเรียกใช้มากขึ้น ทำให้ผู้ใช้ไม่จำเป็นที่จะต้องสร้างโปรแกรมขึ้นมาใช้งานเอง โดยกล่องเครื่องมือที่สร้างขึ้นจะครอบคลุมการทำงานด้านต่างๆมากมาย เช่น Signal Processing, Control Systems, Neural Networks, Fuzzy Logic, Wavelets, Simulation เป็นต้น

ความสามารถหลักของ MATLAB ที่ทำให้เป็นโปรแกรมที่เหมาะสมกับการทำงานทางด้านวิศวกรรมด้วยเหตุผลดังนี้

1. MATLAB เป็นโปรแกรมเพื่อการคำนวณและแสดงผลได้ทั้งตัวเลขและรูปภาพซึ่งมีประสิทธิภาพสูง โดยผู้ผลิตได้ให้นิยามว่าเป็น High-Performance Numeric Computation and Visualization Software
2. MATLAB จะควบคุมการทำงานด้วยชุดคำสั่งและยังสามารถรวบรวมชุดคำสั่งเป็นโปรแกรมได้อีกด้วย
3. MATLAB มีฟังก์ชันที่เหมาะสมกับงานทางวิศวกรรมพื้นฐานมากมาย นอกจากนั้นผู้ใช้อย่างยังสามารถเขียนฟังก์ชันหรือคำสั่งการทำงานขึ้นมาใหม่โดยสามารถใช้ประโยชน์จากฟังก์ชันที่มีอยู่แล้วใน MATLAB เพื่อให้เหมาะสมกับงานของผู้ใช้แต่ละกลุ่ม
4. ลักษณะการเขียนโปรแกรมใน MATLAB จะใกล้เคียงการเขียนสมการทางคณิตศาสตร์ที่เราคุ้นเคยจึงง่ายกว่าการเขียนโปรแกรมโดยใช้ภาษาอื่นๆ
5. MATLAB มีความสามารถในการเขียนกราฟและรูปภาพทั้ง 2 มิติและ 3 มิติได้อย่างมีประสิทธิภาพ

6. MATLAB สามารถทำงานติดต่อกับโปรแกรมอื่นๆได้ไม่ว่าจะเป็น Microsoft Word, Microsoft Excel หรือโปรแกรมอื่นๆที่ร่วมทำงานอยู่บนระบบปฏิบัติการ Windows

7. MATLAB มีกล่องเครื่องมือหรือชุดคำสั่งพิเศษสำหรับผู้ใช้ที่ต้องการใช้งานเฉพาะทางหรืองานด้านวิศวกรรมขั้นสูงอื่นๆ

ประโยชน์ของ MATLAB [10]

การพัฒนาโปรแกรมด้วย MATLAB มีความง่ายและเร็วกว่าภาษาอื่นๆ เพราะ MATLAB จะมีไลบรารีจำนวนมากรองรับ และด้วยลักษณะการทำงานเชิงเมตริกซ์ทำให้ในการพัฒนาจะสามารถจัดการกับอาร์เรย์ได้อย่างง่ายดาย การเขียนโปรแกรมสั้นกะทัดรัด เหมาะกับการสร้างและทดสอบระเบียบวิธีใหม่ๆ รองรับการทำงานกับกราฟิก รวมถึงสามารถพัฒนาในรูปแบบ GUI ทำให้สะดวกในการป้อนค่าและแสดงผล นอกจากนั้นยังสามารถติดต่อกับฮาร์ดแวร์ และโปรแกรมภาษาอื่นๆได้ โดยสามารถแบ่งประโยชน์ของ MATLAB ออกเป็น 3 ข้อหลักๆได้แก่

1. เป็นโปรแกรมเพื่อการคำนวณ

1.1 เชิงตัวเลข (Numeric) สามารถใช้ในการคำนวณธรรมดาหรือใช้งานฟังก์ชันทางคณิตศาสตร์ขั้นสูงได้

1.2. เชิงสัญลักษณ์ (Symbolic) สามารถใช้คำนวณในเชิงตัวแปรได้ เช่น การอินทิเกรต หรือการแก้สมการต่างๆแบบติดตัวแปร

2. สามารถสร้างเป็นโปรแกรมใหม่เพื่อใช้ในงานเฉพาะทางได้

2.1 สามารถพัฒนาโปรแกรมขึ้นใหม่โดยเขียนโปรแกรมได้ทั้งแบบสคริปต์ เพื่อให้ทำงานในลักษณะชุดคำสั่งต่อเนื่อง หรือแบบฟังก์ชัน เพื่อให้ทำงานและแสดงผลออกมาได้

2.2 สามารถใช้งานได้แบบการแปลเป็นภาษาเครื่องทีละคำสั่ง (Interpret) หรือแบบแปลเป็นภาษาเครื่องหมดทั้งโปรแกรม (Compile) เพื่อที่จะให้โปรแกรมที่พัฒนาขึ้นมาทำงานไปตามแบบที่เขียน

2.3 มี GUI (Graphical User Interface) เป็นการพัฒนาที่สร้างขึ้นด้วยเครื่องมือด้านกราฟิก (graphical object) แบบต่าง ๆ เช่น ปุ่มกดต่างๆ เมนูแสดงการทำงานต่างๆ ซึ่งการพัฒนาโดยใช้ GUI นี้ จะทำให้ผู้ใช้งานโปรแกรมเข้าใจถึงความหมายและวิธีการใช้ได้ดียิ่งขึ้น ซึ่งจะทำให้สามารถพัฒนาโปรแกรมโดยมีวิธีการกำหนดให้เกิดขั้นตอนต่อ ๆ ไป ตามที่เราต้องการได้ดียิ่งขึ้น

2.4 รองรับการพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ทั้งการใช้คลาสของ MATLAB เองหรือคลาสของภาษาอื่นๆ เช่น Java หรือ .NET

2.5 สามารถตรวจสอบข้อผิดพลาด (Debug) ของโปรแกรมได้ และในส่วนของติดต่อกับภาษาอื่นๆ สามารถแปลเป็นภาษาเครื่องหมดทั้งโปรแกรม (Compile) เพื่อนำไปทำการตรวจสอบข้อผิดพลาดในโปรแกรมอื่น เช่น Visual Studio ได้ด้วย

3. สามารถติดต่อ หรือใช้งานร่วมกับโปรแกรม ฮาร์ดแวร์ หรือเพิ่มข้อมูลรูปแบบต่างๆได้

3.1 สามารถเชื่อมต่อกับภาษา หรือโปรแกรมอื่นๆได้เช่น Java, C/C++, .NET, MS Excel โดยการพัฒนาอาจให้โปรแกรมหลักพัฒนาโดยใช้ MATLAB แล้วเรียกใช้งานภาษา หรือโปรแกรมอื่น หรือให้ภาษาอื่นเป็นโปรแกรมหลักแล้วเรียกใช้งาน MATLAB ก็ได้

3.2 สามารถอ่านหรือเขียนเพิ่มข้อมูลสื่อสารแบบมาตรฐานได้ เช่น ข้อความ รูปภาพ เสียง วิดีโอ เป็นต้น

3.3 สามารถติดต่อกับฮาร์ดแวร์ต่างๆ เช่น กล้องวิดีโอ บอร์ด DSP (Digital Signal Processor) เป็นต้น ได้

GUI ของโปรแกรม MATLAB

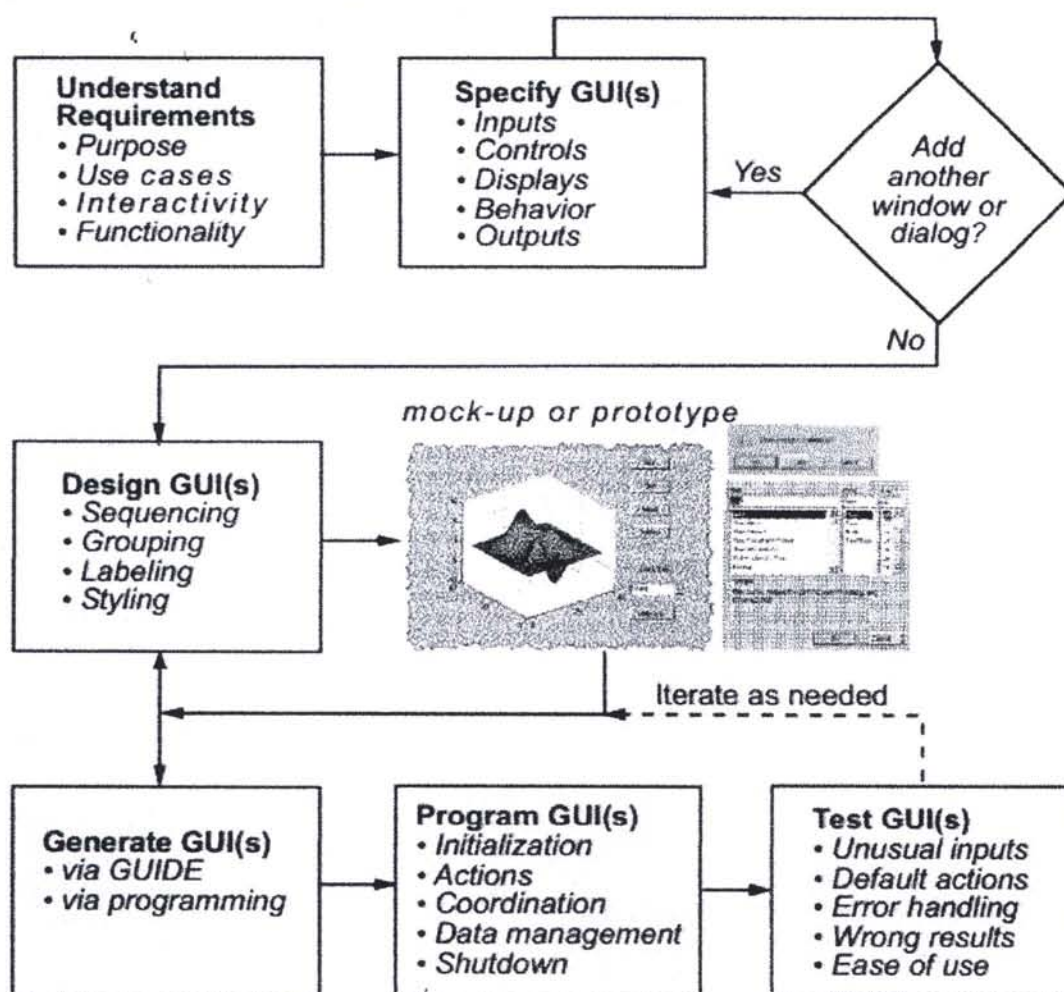
ส่วนติดต่อผู้ใช้แบบกราฟิก (Graphical User Interface: GUI) เป็นวิธีการเชื่อมต่อระหว่างผู้ใช้และคอมพิวเตอร์ ซึ่งเป็นส่วนประกอบของโปรแกรมที่จะทำงานระหว่างโปรแกรมหลักกับผู้ใช้โปรแกรม โดยโปรแกรมคอมพิวเตอร์จะทำงานก็ต่อเมื่อผู้ใช้งานได้ป้อนข้อมูลที่ต้องการให้โปรแกรม โปรแกรมจะแสดงตัวอักษรและกราฟิกต่างๆทางจอภาพ ช่วยเพิ่มความสมบูรณ์และความสะดวกให้แก่ผู้ใช้งานมากขึ้น [10]

ส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) [12] คือ การแสดงผลกราฟิกในหนึ่งหน้าต่างหรือมากกว่าหนึ่งหน้าต่าง ที่มีตัวควบคุมที่เรียกว่าองค์ประกอบที่สามารถช่วยให้ผู้ใช้ดำเนินการโต้ตอบกับโปรแกรมตามความต้องการ ผู้ใช้งาน GUI ไม่ต้องทำการสร้างสคริปต์หรือคำสั่งการทำงาน ซึ่งจะแตกต่างกับการเขียนโปรแกรมเพื่อใช้ในการทำงาน ผู้ใช้งาน GUI ไม่จำเป็นต้องเข้าใจรายละเอียดของวิธีการทำงานของโปรแกรมการทำงาน GUI จะรวมองค์ประกอบต่างๆเช่น Menus, Tools Bar, ปุ่มกด, ปุ่มเลือก, กล่องรายการและSliders เป็นต้น ไว้เพื่อให้ผู้ใช้งานใช้โปรแกรมได้สะดวก

GUIs ที่สร้างขึ้นโดยใช้เครื่องมือของ ® MATLAB จะสามารถดำเนินการคำนวณได้ในหลายรูปแบบของการคำนวณใดๆ สามารถอ่านและเขียนไฟล์ข้อมูลอื่น ๆได้ เชื่อมต่อหรือติดต่อสื่อสารกับ GUIs อื่นๆหรือในภาษาคอมพิวเตอร์อื่นๆได้ และยังแสดงข้อมูลเป็นตารางหรือในรูปแบบกราฟได้ [12]

ในการออกแบบซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์ จะต้องเข้าใจวัตถุประสงค์และความต้องการของผู้ใช้งานก่อน เพื่อให้การออกแบบ GUIs จะตอบสนองความต้องการของผู้ใช้งานได้อย่างสมบูรณ์ ซึ่งจะรวมไปถึงการระบุรายละเอียดหรือการกำหนดค่าต่างๆ เช่น ค่าที่ป้อนให้โปรแกรม ค่าที่คำนวณได้ การแสดงผล และการทำงานของโปรแกรมที่จะต้องให้อยู่ในการควบคุม และหลังจากที่ออกแบบ GUI เรียบร้อยแล้วจะต้องทำการเขียนโปรแกรมควบคุมแต่ละส่วนให้ทำงานได้อย่างถูกต้องตามความต้องการ และทำการทดสอบโปรแกรมว่าการทำงานอยู่ภายใต้เงื่อนไขที่ออกแบบไว้หรือไม่ แสดงได้ดังแผนภาพ 7 [12]

Designing a Graphical User Interface



ภาพ 7 การออกแบบส่วนติดต่อผู้ใช้แบบกราฟิก

การพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ

การพัฒนาโปรแกรมเชิงวัตถุ [13] หรืออาจเรียกว่า "Agile Development" มีกระบวนการที่คล้ายกับการออกแบบโปรแกรม คือ การวิเคราะห์ความต้องการ การออกแบบโปรแกรม การเขียนโปรแกรม การทดสอบและแก้ไขโปรแกรม การเขียนเอกสาร และการบำรุงรักษาโปรแกรม การพัฒนาโปรแกรมเชิงวัตถุจะเป็นการมุ่งเน้นที่ลำดับของการดำเนินการและผลลัพธ์ที่เกิดจากกระบวนการดังกล่าว เพื่อช่วยให้ประสิทธิภาพในการดำเนินงานและช่วยลดการผิดพลาดในแต่ละขั้นตอนได้มากกว่าวิธีอื่น

Agile Development [14] เป็นกลุ่มของกระบวนการพัฒนาซอฟต์แวร์ โดยทั่วไปจะตรงข้ามกับกับวิธีการพัฒนาซอฟต์แวร์ที่มีการวางแผน โดยวิธีการแบบ Agile จะเน้นไปยังการปรับปรุงซอฟต์แวร์อย่างรวดเร็ว หรือเน้นการพัฒนาโดยรองรับการเปลี่ยนแปลงในอนาคต ทั้งนี้ไม่ได้หมายความว่า การพัฒนาด้วยวิธีนี้ไม่ได้มีการวางแผนใดๆ

โดย Agile ได้ทำการแบ่งงานให้เป็นงานเล็กๆ และมีการวางแผนในช่วงสั้นๆ โดยได้แบ่งการทำงานออกเป็นช่วงสั้นๆ ซึ่งในแต่ละช่วงการทำงานจะใช้รูปแบบวงจรการพัฒนาซอฟต์แวร์เต็มขั้น ซึ่งรวมถึงการวางแผน การเก็บข้อมูลความต้องการ การออกแบบซอฟต์แวร์ การพัฒนา และการทดสอบซอฟต์แวร์ และในเมื่อหมดช่วงการทำงานนั้นๆ ทีมผู้พัฒนาจะแสดงผลงานที่พัฒนาในช่วงสั้นๆ ให้ได้รับทราบ

ข้อดีของการพัฒนาเชิงวัตถุ [5]

การพัฒนาเชิงวัตถุ เป็นเทคโนโลยีที่ได้รับการพัฒนาอย่างต่อเนื่อง โดยสรุปข้อดีของการพัฒนาเชิงวัตถุได้ดังนี้ คือ

1. ทำให้นักพัฒนาสามารถออกแบบโปรแกรมได้อย่างครบถ้วนมากยิ่งขึ้น

การออกแบบโปรแกรม นักพัฒนาสามารถออกแบบโปรแกรมจากการวิเคราะห์ปัญหาได้หลายรูปแบบ โดยแต่ละรูปแบบมีลักษณะการทำงานไม่เหมือนกันได้ ในการพัฒนาเชิงวัตถุผู้พัฒนาจะใช้โมเดลการออกแบบเดียวกันตลอดการพัฒนา การกำหนดคลาสและวัตถุในโมเดลที่ออกแบบจากการวิเคราะห์ จะทำให้นักพัฒนาสามารถดูความสัมพันธ์ระหว่างการกำหนดปัญหากับวิธีการแก้ปัญหา และลงมือพัฒนาได้ง่าย

2. ช่วยเพิ่มความเข้าใจในประเด็นของการพัฒนา จะทำให้นักพัฒนาสามารถดำเนินการไปในทิศทางที่ถูกต้อง

ในมุมมองการพัฒนาเชิงวัตถุนั้น จะเป็นการมองจากระดับสูงและใกล้เคียงประเด็นปัญหาจากการวิเคราะห์มาก ซึ่งการออกแบบการพัฒนาเชิงวัตถุนั้น แทบจะไม่มีมุมมองทางด้าน

คอมพิวเตอร์เข้าไปเกี่ยวข้อง ส่งผลให้โปรแกรมที่ได้รับการพัฒนาจะมีการแยกแยะระหว่างสิ่งที่เป็นอิสระต่อกัน กล่าวคือวัตถุที่ได้รับการออกแบบจะแยกออกจากกันอย่างชัดเจน แต่ก็จะแสดงความสัมพันธ์ที่ใกล้ชิดกันระหว่างสิ่งที่ต้องเกี่ยวข้องกัน เช่น ข้อมูลและกระบวนการนำข้อมูลไปใช้ เป็นต้น

3. ช่วยเพิ่มเสถียรภาพของการเปลี่ยนแปลง เมื่อมีความต้องการเปลี่ยนแปลงความสามารถของโปรแกรม นักพัฒนาสามารถแก้ไขปรับปรุงได้อย่างรวดเร็ว

การพัฒนาเชิงวัตถุ จะมีความเสถียรมากกว่า การพัฒนาแบบโครงสร้าง เมื่อโปรแกรมมีการเปลี่ยนแปลงความต้องการก็จะมีเพิ่มวัตถุหรือทำการเปลี่ยนแปลงในวัตถุหนึ่งๆที่ต้องการเปลี่ยนเท่านั้น โดยที่ไม่ต้องเปลี่ยนแปลงโครงสร้างของโปรแกรมทั้งหมด

4. มีการออกแบบที่สนับสนุนการนำกลับมาใช้ (Reuse)

การพัฒนาเชิงวัตถุสนับสนุนการนำกลับมาใช้ใหม่ โดยให้ผู้พัฒนาที่จะใช้ทำการเพิ่มและขยายคุณสมบัติของคลาสหรือวัตถุนั้นได้เลยโดยไม่ต้องทำการเปลี่ยนแปลงซอร์สโค้ดเก่าเลย หรือบางครั้งหากต้องการให้วัตถุนั้นมีความสามารถเพิ่มขึ้น นักพัฒนาที่ต้องการใช้สามารถเพิ่มชุดคำสั่งหรือแก้ไขคุณสมบัติต่างๆได้

5. สนับสนุนการปรับเปลี่ยนขนาดของโปรแกรม (Scalability)

การพัฒนาเชิงวัตถุ จะมีการออกแบบการทำงานที่แยกออกจากกันอย่างชัดเจนและยังมีการใช้สัญลักษณ์หรือเครื่องหมายแทนตัวเลขหรือการคำนวณเดียวกันตลอดกระบวนการของการพัฒนา จะสามารถทำให้นักพัฒนาสามารถเปลี่ยนจากการวิเคราะห์เป็นการออกแบบได้ง่าย

6. สนับสนุนการออกแบบระบบที่เชื่อถือได้และความปลอดภัย (Reliability and Safety)

การพัฒนาด้วยวิธีเชิงวัตถุจะมีการแยกการทำงานอย่างมีสัดส่วน ทำให้การสื่อสารระหว่างวัตถุที่ไม่เกี่ยวข้องกันถูกจำกัดการทำงานอยู่ด้วยอินเตอร์เฟส (Interface) ที่ผู้พัฒนากำหนดไว้เท่านั้น ทำให้การพัฒนาเชิงวัตถุมีความน่าเชื่อถือได้มากกว่า เพราะว่าผู้พัฒนาสามารถควบคุมการติดต่อระหว่างวัตถุได้

7. สนับสนุนการทำงานแบบพร้อมกัน (Concurrency)

การพัฒนาด้วยวิธีเชิงวัตถุสนับสนุนการทำงานแบบพร้อมกันในตัวเองและในรายละเอียดของงาน สามารถทำการเชื่อมโยงระหว่างงาน (Synchronization) แล้วแสดงออกมาในแผนภาพของ State charts, วัตถุที่ทำงานแบบแอคทีฟ (Active Object) และการส่งข้อความ

(Message) ระหว่างวัตถุ ซึ่งแผนภาพเหล่านี้เป็นเครื่องมือที่เหมาะสมสำหรับการพัฒนาโปรแกรมที่มีเงื่อนไขทางด้านประสิทธิภาพ

หลักการพัฒนด้วยวิธีเชิงวัตถุ [5]

เทคโนโลยีการพัฒนด้วยวิธีเชิงวัตถุเป็นหลักการทางวิศวกรรม ซึ่งบรรจุไว้ด้วยองค์ประกอบต่างๆ ที่เรียกว่า โมเดลของวัตถุ (Object Model) ในโมเดลของวัตถุนั้นก็แบ่งออกเป็น การแยกแยะเอกลักษณ์ (Abstraction) การซ่อนรายละเอียด (Encapsulation) การรวมกลุ่มความสัมพันธ์ (Modularity) ลำดับชั้น (Hierarchy) ชนิด (Typing) การทำงานพร้อมกัน (Concurrency) และการรักษาสถานะ (Persistence) สำหรับการวิเคราะห์และออกแบบโปรแกรมด้วยวิธีเชิงวัตถุ นั้น จะขึ้นอยู่กับภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ และในภาษาสำหรับการเขียนโปรแกรมเชิงวัตถุแต่ละภาษาก็มีรายละเอียดปลีกย่อยที่ต่างกันด้วย

หลักการของวัตถุ จะมีการแบ่งปัญหาออกเป็นวัตถุ ซึ่งจะมีความผูกพันกันน้อย (Loose Couple) ทำให้การดูแลและแก้ไขส่วนต่างๆ ของโปรแกรมหรือวัตถุสามารถทำได้ง่ายและมีผลกระทบต่อส่วนอื่นๆ ของโปรแกรมน้อยที่สุด

1. การวิเคราะห์ด้วยวิธีเชิงวัตถุ (Object-Oriented Analysis, OOA) จะเน้นการมองในความเป็นจริงโดยใช้มุมมองของการพัฒนด้วยวิธีเชิงวัตถุ การวิเคราะห์ด้วยวิธีเชิงวัตถุเป็นการวิเคราะห์และตรวจสอบความต้องการโดยใช้มุมมองจากคลาสและวัตถุ หลังจากการวิเคราะห์แล้วจะได้โมเดลของโปรแกรมเชิงวัตถุ ซึ่งจะนำไปใช้เป็นหลักในการออกแบบ และเมื่อออกแบบเสร็จจะได้โครงสร้างหลักสำหรับการเขียนโปรแกรมเชิงวัตถุต่อไป

2. การออกแบบด้วยวิธีเชิงวัตถุ (Object-Oriented Design, OOD) เป็นวิธีการในการออกแบบกระบวนการในการแยกย่อย และเป็นการใช้สัญลักษณ์แบบการพัฒนด้วยวิธีเชิงวัตถุสำหรับโมเดลทั้งในทางตรรกะ (Logic) และทางกายภาพ (Physical) รวมถึงโมเดลการทำงานแบบสถิต (Static) และไดนามิก (Dynamic) สำหรับโปรแกรมที่ต้องการออกแบบ โดยจะเป็นการออกแบบโดยยึดวัตถุเป็นหลัก

3. การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming, OOP) เป็นวิธีการสร้างโปรแกรม โปรแกรมที่ถูกสร้างขึ้นจะถูกจัดให้มีโครงสร้างในการทำงานร่วมกันของวัตถุ ซึ่งวัตถุนั้นจะเป็นตัวตน (Instance) ของคลาส และคลาสก็เป็นส่วนหนึ่งของลำดับชั้น (Hierarchy) ของคลาสที่มีความสัมพันธ์ในการสืบทอดคุณสมบัติ (Inheritance) โดยภาษาที่จะเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุได้นั้นต้องมีคุณสมบัติ 3 ข้อดังนี้

1. เป็นภาษาที่ใช้วัตถุเป็นหลัก ไม่ใช้อัลกอริทึมเป็นหลักในการแก้ปัญหา

2. วัตถุที่ปรากฏในการเขียนโปรแกรมจะต้องเป็นตัวตนของคลาส

3. การสร้างคลาสจะต้องมีความสัมพันธ์กับคลาสอื่นด้วยการสืบทอดคุณสมบัติ

คลาสกับวัตถุ เป็นสองสิ่งที่มีความเกี่ยวข้องกันโดยตรง คลาสคือโค้ดที่เราเขียนขึ้นเพื่อทำหน้าที่เป็นโครงสร้างหลักหรือพิมพ์เขียวของวัตถุ การสร้างวัตถุจากคลาสเรียกว่าการทำให้ instantiation

วัตถุที่ถูกสร้างขึ้นเรียกว่าหนึ่งอินสแตนซ์ ซึ่งสามารถสร้างวัตถุได้หลายๆอินสแตนซ์จากคลาสเพียงคลาสเดียว

การใช้งานคลาสทำได้สองวิธี วิธีแรกคือการนำไปสร้างวัตถุ และเรียกใช้งานผ่านวัตถุ อีกวิธีหนึ่งคือการเรียกใช้คลาสโดยตรงไม่ต้องสร้างวัตถุ เรียกว่า "Static Class" การใช้งานแบบ Static Class จะไม่มีการสร้างวัตถุมาทำงานในโปรแกรม ทำให้ดูเหมือนไม่ตรงหลักการ OOP แต่ในบางสถานการณ์ก็ถือว่ามีความเหมาะสมดี

การใช้งานวัตถุ ต้องทำได้ง่ายโดยไม่จำเป็นต้องรู้รายละเอียดการทำงานภายในของมัน เมื่อต้องการเรียกใช้งานวัตถุให้ทำงานบางอย่าง เราจะเรียกเมธอด (Method) ของมัน วัตถุที่ถูกเรียกจะรวมกระบวนการ (Method) และข้อมูล (Information) ไว้ภายในตัวของมันเอง โดยไม่จำเป็นต้องรู้ว่ามันเก็บไว้ที่ไหน อย่างไร รู้เพียงวิธีการดึงข้อมูลที่ได้ออกมาใช้งานก็พอ

งานวิจัยที่เกี่ยวข้อง

Kobsak Sriprapha, et al. [1] ได้ทำการศึกษาในเรื่องการพึ่งพาคุณสมบัติในการดูดซึมแสงของเซลล์แสงอาทิตย์ชนิดฟิล์มบางซิลิกอน เพื่อที่จะค้นหาการดูดซึมแสงที่ดีที่สุด ในสภาวะที่แผงเซลล์แสงอาทิตย์มีอุณหภูมิสูง โดยการเปรียบเทียบระหว่างเซลล์แสงอาทิตย์ชนิดอะมอร์ฟัสซิลิกอนแบบเจือไฮโดรเจน (hydrogenated amorphous), ชนิดผลึกดั้งเดิม (protocrystalline) และชนิดผลึกขนาดเล็ก (microcrystalline) ที่ผลิตด้วยวิธีการเคลือบไอเคมี โดยการใช้พลาสมาทำให้ไอเคมีแตกตัว แล้วไปเคลือบผิวซับสเตรต (plasma-enhanced chemical vapor deposition หรือ PECVD) และวิธีการเคลือบไอเคมีด้วยการใช้เส้นความร้อน (hot-wired chemical vapor deposition หรือ HWCVD) ลักษณะของ J-V วัดจากเครื่องจำลองแสงอาทิตย์ (Solar Simulator) ซึ่งอุณหภูมิโดยรอบอยู่ระหว่าง 25 – 75 องศาเซลเซียส เซลล์แสงอาทิตย์ชนิดผลึกซิลิคอนเซลล์แบบดั้งเดิม (protocrystalline) ให้ค่าสัมประสิทธิ์อุณหภูมิต่ำสุดสำหรับประสิทธิภาพของเซลล์ ในขณะที่เซลล์แสงอาทิตย์ชนิดผลึกซิลิคอนขนาดเล็กแบบดั้งเดิม (microcrystalline) จะมีปฏิกิริยาเร็วต่ออุณหภูมิ ผลการทดลองพบว่าเซลล์แสงอาทิตย์ชนิดผลึก

ซิลิคอนเซลล์แบบดั้งเดิม (protocrystalline) เป็นวัสดุที่มีแนวโน้มเพื่อใช้เป็นเซลล์แสงอาทิตย์ชนิดฟิล์มบางซิลิคอนที่สามารถดูดซับแสงอาทิตย์และทำงานได้ดีสำหรับการดำเนินงานในภูมิภาคที่มีอุณหภูมิสูง ซึ่งค่าสัมประสิทธิ์อุณหภูมิที่ประเมินออกมาได้ จะสามารถบอกถึงการเปลี่ยนแปลงคุณสมบัติทางไฟฟ้าของเซลล์แสงอาทิตย์ ณ ที่อุณหภูมิต่างๆได้ โดยมีวิธีการประเมินค่าสัมประสิทธิ์อุณหภูมิมีสูตรการคำนวณ ดังนี้

$$TC(Part/^{\circ}C) = \frac{1}{z} \frac{\delta z}{\delta T} \Big|_{T_n} = 2.5^{\circ}C \quad (3)$$

เมื่อ TC คือ สัมประสิทธิ์อุณหภูมิ (Temperature Coefficient)

Z คือ ค่าของตัวแปรที่ต้องการหาค่า TC เช่น Isc , Voc หรือ Pmax

Tn คือ อุณหภูมิอ้างอิงปกติ 25 °C

$\frac{\delta z}{\delta T}$ คือ อัตราการเปลี่ยนแปลงของตัวแปร Z ต่ออุณหภูมิที่เปลี่ยนแปลงจากอุณหภูมิอ้างอิง 25 °C

Dogan Ibrahim [15] ได้ทำการศึกษาเกี่ยวกับการเรียนการสอนในหลักสูตรวิศวกรรม ซึ่งจำเป็นที่จะต้องให้นักศึกษาในหลักสูตรวิศวกรรมได้สัมผัสกับเครื่องมือและอุปกรณ์จริงทางวิศวกรรม ในงานวิจัยนี้ได้ทำการจำลองเครื่องมือและอุปกรณ์ที่สามารถใช้ในห้องปฏิบัติการได้ เรียกว่า ห้องปฏิบัติการทางวิศวกรรมเสมือนจริง สามารถจำลองสิ่งแวดล้อม ข้อดีและข้อเสียได้ โดยใช้โปรแกรม MATLAB ในการจำลองการศึกษาในหลักสูตรวิศวกรรมไฟฟ้าและอิเล็กทรอนิกส์ พบว่า การจำลองเครื่องมือทางวิศวกรรม เป็นสิ่งสำคัญอย่างหนึ่งในการศึกษาทางด้านวิศวกรรม แม้ว่าการจำลองนี้ นักศึกษาวิศวกรรมจะไม่ได้สัมผัสเครื่องมือและอุปกรณ์จริง แต่สามารถทำให้นักศึกษาเข้าใจในหลักการใช้เครื่องมือและอุปกรณ์ทางวิศวกรรมได้ ซึ่งในงานวิจัยได้ใช้ตัวอย่างการจำลองการทำงานของวงจรไฟฟ้ามาแสดงให้เห็น โดยใช้เครื่องมือและส่วนประกอบในการจำลองจากโปรแกรม MATLAB

Joaquim Jaumota, et al. [16] ได้ทำการศึกษาเกี่ยวกับการใช้ภาพกราฟิกในการติดต่อสื่อสารกับผู้ใช้งานให้ผู้ใช้งานสามารถใช้งานได้ง่าย (graphical user-friendly interface) สำหรับการแก้สมการเชิงเส้นที่มีหลายตัวแปร (Multivariate Curve Resolution) โดยใช้วิธีการสลับกันของสี่เหลี่ยมให้น้อยที่สุด (Alternating Least Squares) หรือ (MCR-ALS) ซึ่งจะทำให้การทำงานติดต่อกันระหว่างภาพกราฟิกและการเลือกข้อมูลที่ใช้ในการวิเคราะห์ เช่น ข้อมูลที่แสดงผลได้หลาย

มิติ หรือ ข้อมูลที่ต้องใช้หลายเทคนิคในการวิเคราะห์ มีประสิทธิภาพมากขึ้น และยังสามารถเลือกข้อจำกัดต่างๆที่ใช้ในการวิเคราะห์ข้อมูลได้ง่ายอีกด้วยแล้ว พบว่าการแก้สมการเชิงเส้นที่มีหลายตัวแปร (MCR) ของชุดข้อมูลจะเป็นกระบวนการต่อเนื่องที่สามารถทำการวิเคราะห์ได้หลายครั้งภายใต้เงื่อนไขที่แตกต่างกันในการเลือกคำสั่งให้วิเคราะห์เพียงครั้งเดียว การใช้ภาพกราฟิกในการติดต่อสื่อสารกับผู้ใช้งานโดยใช้ MCR-ALS จะทำให้ผู้ใช้ใช้งานได้ง่าย อีกทั้งยังสามารถปรับเปลี่ยนเงื่อนไขในการวิเคราะห์ได้สะดวก โดยพัฒนาจากชุดคำสั่งของโปรแกรม MATLAB

C.S. Chin, A. Babu, W. McBride [17] ได้ทำการศึกษาเกี่ยวกับการออกแบบ การสร้างแบบจำลองและการทดสอบสำหรับแผงเซลล์แสงอาทิตย์แบบติดตามดวงอาทิตย์ชนิดแกนเดียวโดยใช้ MATLAB/Simulink ในงานวิจัยจะใช้ตัวต้านทานไวแสงในการตรวจจับความเข้มแสงอาทิตย์ติดตั้งที่กึ่งกลางของโครงแผงเซลล์ทั้งด้านบนและด้านล่าง โดยระบบติดตามดวงอาทิตย์จะสามารถทำงานได้ในสภาพที่แตกต่างกันเพื่อความยืดหยุ่นสำหรับการใช้งานในสภาพอากาศหรือพื้นที่ที่แตกต่างกัน หรือสำหรับการตั้งค่าที่แตกต่างกันของผู้ใช้งาน และจะทำงานในขณะที่มีแสงเท่านั้น ในเวลากลางคืนจะถูกปรับให้อยู่ในสภาพหยุดทำงาน แบบจำลองที่ได้จากการวิจัยจะสามารถนำไปใช้ในการประมาณค่าหรือตรวจสอบประสิทธิภาพการผลิตไฟฟ้าของแผงเซลล์แสงอาทิตย์ได้ ก่อนการดำเนินการจริง พบว่าแม้จะสามารถจำลองเพื่อประมาณค่าหรือตรวจสอบประสิทธิภาพการผลิตไฟฟ้าของแผงเซลล์แสงอาทิตย์ได้ แต่ผลลัพธ์ที่ได้จากแบบจำลองไม่มีตัวแปรของสิ่งแวดล้อมภายนอกอยู่ด้วย เช่น แรงเสียดทานทางกล แรงลม และผลเฉื่อยที่ได้ในแต่ละชั่วโมง ดังนั้นในการพัฒนาในอนาคต การสร้างแบบจำลองจะต้องมีการเพิ่มค่าตัวแปรของสิ่งแวดล้อมภายนอกเข้าไปด้วยเพื่อให้ได้แบบจำลองที่มีประสิทธิภาพสูงขึ้น

Tetsuyuki Ishii, et al. [18] ได้ทำการศึกษาในเรื่องของการประเมินค่าสัมประสิทธิ์อุณหภูมิที่กำลังไฟฟ้าสูงสุดของแผงเซลล์แสงอาทิตย์ในเวลาที่แตกต่างกัน พบว่าค่าสัมประสิทธิ์อุณหภูมิของเซลล์แสงอาทิตย์ชนิดอะมอร์ฟัสซิลิกอนจะมีค่าเป็นลบจากการวัดในอาคาร ซึ่งจะตรงกันข้ามกับการวัดค่านอกตัวอาคารที่มีอุณหภูมิเพิ่มขึ้น เพื่อแก้ปัญหาที่เกิดขึ้นนี้ จึงทำการตรวจสอบค่าสัมประสิทธิ์อุณหภูมิของแผงเซลล์แสงอาทิตย์ชนิดอะมอร์ฟัสซิลิกอน โดยใช้ 2 วิธีการที่ต่างกัน พบว่าในระยะเวลาลั้นจะมีค่าสัมประสิทธิ์อุณหภูมิน้อยกว่าที่มีระยะเวลายาวซึ่งเป็นผลจากความร้อนและแสงที่ส่องนานขึ้น แสดงให้เห็นว่าผลที่ได้จากการคำนวณสัมประสิทธิ์อุณหภูมิจะบอกถึงพฤติกรรมของแผงเซลล์แสงอาทิตย์ชนิดอะมอร์ฟัสซิลิกอนเป็นผลมาจากการเปลี่ยนแปลงความร้อนของแผงเซลล์แสงอาทิตย์และแสงที่ตกกระทบ เพื่อใช้เป็นตัวช่วยในการคำนวณประสิทธิภาพของแผงเซลล์แสงอาทิตย์ได้ถูกต้องยิ่งขึ้น

Sulaiman Shaari, et al. [19] ได้ทำการศึกษาเรื่องค่าสัมประสิทธิ์ที่เกี่ยวข้องกับอุณหภูมิของเซลล์แสงอาทิตย์ชนิดอะมอร์ฟัสซิลิกอนและแบบผลึกและทำการทดสอบในพื้นที่ประเทศมาเลเซีย ซึ่งใช้เทคนิคการลดลงเชิงเส้นในการวิเคราะห์ โดยได้ทำการติดตั้งระบบแผงเซลล์แสงอาทิตย์กับแบตเตอรี่ขนาด 62 Wp สำหรับเซลล์แสงอาทิตย์ชนิดอะมอร์ฟัสซิลิกอน , 225 Wp สำหรับชนิดผลึกผสม และ 225 Wp สำหรับชนิดผลึกเดี่ยว ระบบที่ติดตั้งได้ออกแบบเพื่อให้รองรับกับบ้านในชนบทที่มีขนาดการใช้ไฟฟ้าที่ 200W 500W และ 550W พบว่ากำลังไฟฟ้าปกติที่ออกมาต่อการเปลี่ยนแปลงของอุณหภูมิของชนิดอะมอร์ฟัสซิลิกอนเท่ากับ 0.037 ต่อ 1 องศาเซลเซียส, ชนิดผลึกผสมเท่ากับ 0.0225 ต่อ 1 องศาเซลเซียส และชนิดผลึกเดี่ยวเท่ากับ 0.0263 ต่อ 1 องศาเซลเซียส นอกจากนั้นที่ความเข้มแสงอาทิตย์มีค่า 500 W/m^2 จะมีค่ากระแสไฟฟ้า, แรงดันไฟฟ้า, กำลังไฟฟ้าและค่าสัมประสิทธิ์ประสิทธิภาพของการเปลี่ยนแปลงอุณหภูมิที่ได้รับจากการลดลงเชิงเส้นชนิดอะมอร์ฟัสซิลิกอนเท่ากับ $37.0 \text{ mA/}^\circ\text{C}$, $-31.8 \text{ mV/}^\circ\text{C}$, $-0.1036 \text{ W/}^\circ\text{C}$ และ $-0.0214\%/^\circ\text{C}$, ชนิดผลึกผสมเท่ากับ $22.5 \text{ mA/}^\circ\text{C}$, $-39.4 \text{ mV/}^\circ\text{C}$, $-0.2525 \text{ W/}^\circ\text{C}$ และ $-0.072\%/^\circ\text{C}$ และชนิดผลึกเดี่ยวเท่ากับ $26.3 \text{ mA/}^\circ\text{C}$, $-32.6 \text{ mV/}^\circ\text{C}$, $-0.1742 \text{ W/}^\circ\text{C}$ และ $-0.0523\%/^\circ\text{C}$ การศึกษานี้จะส่งผลโดยตรงกับการออกแบบขนาดของระบบในภูมิภาคที่มีสภาพภูมิอากาศคล้ายคลึงกัน ดังนั้นจึงเป็นข้อสนับสนุนที่ดีสำหรับการออกแบบขนาดของระบบเซลล์แสงอาทิตย์ในเขตอากาศร้อนชื้นได้ทั่วโลก

อนุสรณ์ แสงประจักษ์ [20] ได้ทำการศึกษาอุณหภูมิสูงต่อค่าแรงดันไฟฟ้าวงจรเปิดของแผงเซลล์แสงอาทิตย์ที่ใช้งานในประเทศไทย พบว่าการใช้งานแผงเซลล์แสงอาทิตย์ชนิดผลึกเดี่ยวและผลึกผสมในประเทศไทยภายใต้อุณหภูมิไม่เกิน 65 องศาเซลเซียสจะทำให้ได้แรงดันไฟฟ้าวงจรเปิดออกมาไม่แตกต่างกัน โดยการทำงานของแผงเซลล์แสงอาทิตย์ในช่วงอุณหภูมิ 55 ถึง 65 องศาเซลเซียส จะทำให้แรงดันวงจรเปิดลดลงไปจากที่ผู้ผลิตระบุไว้ถึง 22 ถึง 25 เปอร์เซ็นต์ ดังนั้นการลดอุณหภูมิของแผงเซลล์แสงอาทิตย์ขณะใช้งานให้มีค่าต่ำกว่าช่วงอุณหภูมิ 55 ถึง 65 องศาเซลเซียส จะมีผลช่วยให้แรงดันไฟฟ้าวงจรเปิดที่ได้จากแผงขณะใช้งานมีค่าคงที่ ซึ่งส่งผลให้แรงดันไฟฟ้าจริงที่จ่ายให้กับโหลดมีเสถียรภาพดีขึ้นด้วย เนื่องจากอุณหภูมิขณะใช้งานที่ต่ำจะทำให้กระแสไฟฟ้าอิ่มตัวมีค่าต่ำลงไปด้วย ทำให้แรงดันไฟฟ้าวงจรเปิดที่ได้มีค่าสูงกว่าเมื่อใช้งานเซลล์แสงอาทิตย์ที่อุณหภูมิสูง