บทที่ 2
การตรวจเอกสาร

ในบทนี้กล่าวถึงทฤษฎีที่ใช้ในงานวิจัยโดยสรุป ซึ่งประกอบไปด้วยหลักการสำหรับกระบวนการพัฒนาซอฟต์แวร์แบบวีโมเดล (V-Model) การหาความต้องการทางวิศวกรรมซอฟต์แวร์ และการสร้างแบบจำลองความถดถอยเพื่อหาสมการความสัมพันธ์เชิงเส้นแบบพหุคูณ (Multiple Linear Regressions) และทฤษฎีงานวิจัยที่เกี่ยวข้องกับการทำวิจัย

2. ทฤษฎีที่เกี่ยวข้องกับโครงการวิจัย

2.1 ทฤษฎีสำหรับกระบวนการพัฒนาซอฟต์แวร์ (Software Process)

ในช่วงปลายทศวรรษที่ 1960 ได้เกิดมีแนวความคิดขึ้นมาว่า กระบวนการพัฒนาซอฟต์แวร์ (Software Process) หรือระบบงานทางคอมพิวเตอร์นั้น มีความคล้ายคลึงกับกระบวนการที่ใช้ในทางวิศวกรรม (Engineering Process) กลุ่มผู้ควบคุมโครงการพัฒนาซอฟต์แวร์ต่างเล็งเห็นว่า การใช้กระบวนการจัดกิจกรรมของระบบงานทางวิศวกรรมในการดำเนินงานเรื่องต่าง ๆ ทำสามารถมองเห็นขั้นตอนการทำงานได้อย่างเด่นชัด แต่อย่างไรก็ตามกระบวนการการออกแบบและพัฒนาซอฟต์แวร์ด้วยกระบวนการทางวิศวกรรมนี้ก็ยังคงมีลักษณะเป็นการศึกษาหาวิธีดำเนินการที่เหมาะสมที่สุด เพราะการพัฒนาขั้นตอนการดำเนินการแบบต่าง ๆ ในการพัฒนาซอฟต์แวร์ในปัจจุบันปรากฏว่าไม่สามารถหาข้อสรุปรวมได้ว่า วิธีการใดเป็นวิธีการที่ดีที่สุด บางวิธีอาจจะดีกว่าอีกวิธีการหนึ่งในสถานการณ์ที่แตกต่างกันออกไป ดังนั้นจึงไม่อาจมีคำตอบแน่นอนสำหรับทุกสถานการณ์ได้ (จรณิต,2540)

กระบวนการออกแบบและพัฒนาระบบงานซอฟต์แวร์จำแนกตามรูปแบบหรือโครงสร้างของวิธีดำเนินการ (Models/Paradigms) ได้หลายรูปแบบด้วยกัน แต่ในโครงการค้นคว้าอิสระนี้จะนำเสนอกระบวนการพัฒนาซอฟต์แวร์แบบวีโมเดล (V-Model) เพื่อการดำเนินการพัฒนาวงจรชีวิตซอฟต์แวร์ (Software Life Cycle)

2.1.1 กระบวนการพัฒนาซอฟต์แวร์แบบ (V-Model)

กระบวนการพัฒนาซอฟต์แวร์แบบวี (V-Model) ดังแสดงในภาพที่ 1.7 กระบวนการที่ได้รับการพัฒนามาจากเทคนิคการออกแบบและพัฒนาซอฟต์แวร์แบบน้ำตก Waterfall Model และผู้เริ่มคิดค้นวิธีนี้ขึ้นมาคือ กระทรวงกลาโหมเยอรมันเมื่อปี 1992 (German Ministry of Defense) ข้อแตกต่างที่สำคัญจากการพัฒนาแบบน้ำตก (Waterfall Model) คือการพัฒนาแบบวี (V-Model) จะทำงานควบคู่กันไปพร้อมกับการทดสอบ โดยจะแบ่งออกเป็น 2 ส่วน มีฝั่งซ้ายและฝั่งขวาเป็นรูปคล้ายตัววี ฝั่งด้านซ้ายมือของตัววีจะเป็นกระบวนการวิเคราะห์และออกแบบระบบซอฟต์แวร์ ส่วนฝั่งขวาของตัววีจะเป็นการทดสอบและการบำรุงรักษาในกระบวนการผลิตซอฟต์แวร์ และในกระบวนการผลิตซอฟต์แวร์แบบวี (V-Model) ประกอบด้วยขั้นตอนการทำงานดังนี้

1. วิเคราะห์ความต้องการ (Requirement Analysis) ขั้นตอนการเก็บความต้องการจากลูกค้าและนำมาวิเคราะห์ความถูกต้องให้ตรงตามความต้องการที่ลูกค้าหรือผู้ใช้

2. ออกแบบระบบ (System Design) ขั้นตอนการออกแบบระบบในกระบวนการพัฒนาระบบซอฟต์แวร์เช่นการออกแบบโครงสร้างของซอฟต์แวร์ (Architecture Software System)

3. ออกแบบโปรแกรม (Program Design) เขั้นตอนการออกแบบรูปแบบการพัฒนาตัวโปรแกรม

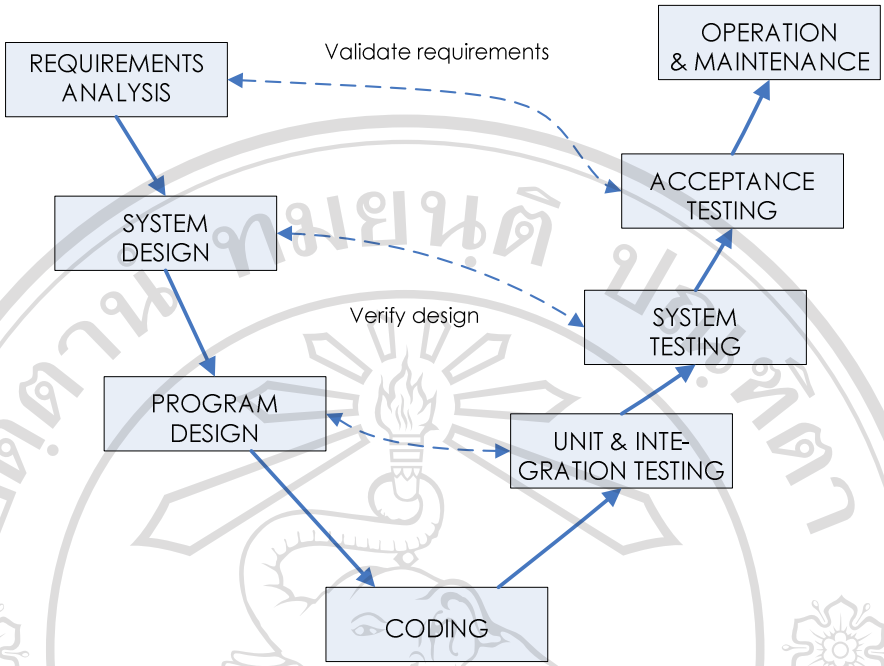
4. การเขียน (Coding) ขั้นตอนการเขียนโปรแกรมของซอฟต์แวร์

5. การทดสอบ (Unit & Integration Testing) ขั้นตอนการทดสอบโดยเป็นหน่วยย่อยและภาพรวมทั้งระบบของซอฟต์แวร์

6. ทดสอบระบบ (System Testing) ขั้นตอนการทดสอบในแต่ละส่วนของระบบซอฟต์แวร์

7. ทดสอบการยอมรับหรือข้อความที่เป็นทางการ (Acceptance Testing) ขั้นตอนการทดสอบซอฟต์แวร์เพื่อให้ตรงตามความต้องการและเพื่อการยอมรับ

8. การใช้งานและบำรุงรักษา (Operation & Maintenance) ขั้นตอนการใช้งานซอฟต์แวร์และการบำรุงรักษาเพื่อป้องกันการเกิดปัญหาเมื่อลูกค้านำไปใช้งาน



ภาพที่ 2.1 แสดงกระบวนการพัฒนาซอฟต์แวร์แบบวี V – Model

2.2 ทฤษฎีสำหรับวิศวกรรมการเก็บความต้องการ (Requirement Engineering)

เครื่องมือวัดระดับความสำเร็จของระบบซอฟต์แวร์ที่ได้ตามจุดประสงค์ที่กำลังมีการกล่าวถึงกันมากที่สุดคือ ระบบวิศวกรรมซอฟต์แวร์ความต้องการ (Requirement Engineering) และเป็นกระบวนการหรือจุดมุ่งหมายที่เห็นได้ชัดของกลุ่มผู้ให้แหล่งข้อมูล (Stakeholders) ว่าต้องการอะไร เอกสารที่ได้จากการเก็บความต้องการเหล่านี้เมื่อทำการวิเคราะห์เสร็จจะเป็นตัวเชื่อมโยงการนำไปสู่การติดตั้งระบบ(Implementation) ซึ่งกระบวนการดังกล่าวนี้จะรวมไปถึงการจ่ายเงิน ลูกค้า ผู้ใช้ระบบ ผู้พัฒนาระบบ หรือหลายอย่างที่เกี่ยวข้องที่แตกต่างกันออกไป ซึ่งสิ่งเหล่านี้เป็นเป้าหมายหลักในกรทำงานไม่ว่าจะเป็นความพึงพอใจของลูกค้าหรือนักพัฒนาระบบความขัดแย้งต่าง ๆ ที่เกิดขึ้นซึ่งเป็นผลต่อการพัฒนาระบบซอฟต์แวร์ทั้งสิ้น จะเห็นได้ว่าวิศวกรรมความต้องการ (Requirement Engineering) มีความสำคัญต่อการพัฒนาระบบเป็นอย่างมากเพราะว่าวิศวกรรมความต้องการ(Requirement Engineering) จะตอบคำถามได้ว่า ทำไมต้องทำ และขั้นตอนการทำหรือพัฒนาระบบจะตอบคำถามว่าทำอะไรและลำดับต่อมาเมื่อได้ความต้องการ มาแล้วจึง ทำการวิเคราะห์ ตรวจสอบความถูกต้อง และทำซอฟต์แวร์ข้อกำหนดต่อไป จากนั้นเมื่อได้ซอฟต์แวร์ข้อกำหนดจึงนำมาเขียนให้เป็นรูปแบบมาตรฐานในวิศวกรรมความต้องการ ซึ่งการทำ วิศวกรรม

ความต้องการจำเป็นอย่างมากที่จะต้องมีความรู้พื้นฐานดังต่อไปนี้เพื่อประกอบการทำ วิศวกรรมความต้องการ

• จิตวิทยา (*Cognitive psychology)* เพื่อช่วยในการทำซึ่งเป็นพื้นฐานที่ควรมีในการเก็บรวบรวมเพราะผู้ที่เข้าไปเก็บความต้องการ ต้องเข้าไปสัมภาษณ์กว่าจะมาซึ่งปัญหาและความต้องการ

• มนุษยวิทยา (*Anthropology* provides)

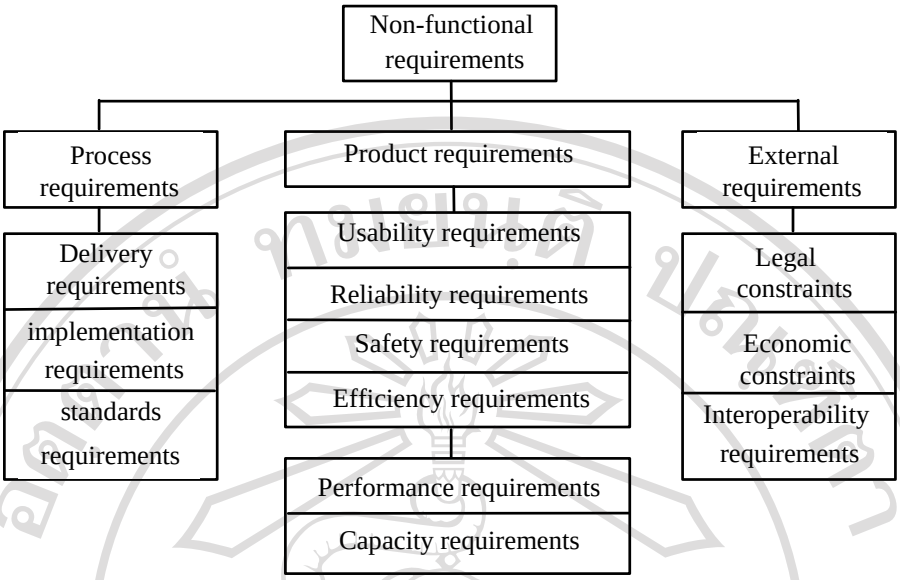
• สังคมวิทยา (*Sociology)*

• ภาษาศาสตร์ (*Linguistics)*

ก่อนการออกแบบและพัฒนาระบวนการผลิตซอฟต์แวร์จะต้องมีความเข้าใจว่าอะไรคือความต้องการ เพื่อที่จะได้ทำการออกแบบและสร้างได้อย่างถูกต้อง แต่การได้มาซึ่งความต้องการนั้นจะได้ทราบได้อย่างไรว่าเป็นความต้องการที่ถูกต้อง ไม่คลาดเคลื่อน เพราะจุดที่ผิดพลาดแต่ละจุด สามารถนำไปสู่ความล้มเหลวได้การเก็บความต้องการจึงเหมือนกับกิจกรรมทางวิศวกรรมซอฟต์แวร์ (Software Engineering) กิจกรรม อื่น ๆ ที่สามารถปรับเปลี่ยนให้เข้ากับความต้องการของ Process, Project, Product และงานที่ทำได้ การปรับเปลี่ยนไม่ได้แปลว่าหลีกเลี่ยง แต่เป็นความจำเป็น ที่จะต้องทำความเข้าใจ ความต้องการของปัญหาก่อนที่จะแก้ปัญหา รูปแบบงานของวิศวกรรมความต้องการ (Requirement Engineering) เป็นการจัดหาวิธีการที่เหมาะสมที่จะนำมาใช้เพื่อที่จะเข้าใจในสิ่งที่ ลูกค้า (Customer) ต้องการและนำมาวิเคราะห์ความต้องการ, การประเมินความเป็นไปได้ , การเจรจาในการแก้ปัญหาอย่างมีเหตุ การอธิบายการแก้ปัญหาให้ชัดเจน ซึ่งสามารถแบ่งชนิดของความต้องการออกได้ 3 แบบดังนี้คือ

1. ความต้องการหลัก (Functional Requirement) เป็นความต้องการที่ต้องมีอยู่ในกระบวนการผลิตซอฟต์แวร์นั้น ๆ ซึ่งจะเป็นความต้องการขั้นพื้นฐานที่ระบบ หรือ ผู้ใช้ ต้องการและจำเป็นต้องมีซึ่งจะขาดไม่ได้ในการทำระบบของซอฟต์แวร์

2. ความต้องการที่ไม่ใช่ความต้องการหลัก (Non – Functional Requirement) เป็นความต้องการของระบบหรือผู้ใช้ที่มีในระบบแล้วสามารถทำให้ระบบหรือซอฟต์แวร์นั้นทำงานได้ในเชิงความสามารถด้านคุณภาพและการทำงานให้มีประสิทธิภาพอาจจะเป็นเงื่อนไขของเวลา, มาตรฐานที่ใช้พัฒนาระบบจะอยู่ในรูปแบบของกระบวนการหรือข้อมูลที่เป็นส่วนช่วยสนับสนุนให้ความต้องการหลัก (Function Requirement) มีการทำงานที่สมบูรณ์หรือทำให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น ซึ่งตัวอย่างของความต้องการที่ไม่ใช่ความต้องการหลัก (Non – Functional Requirement) ดังแสดงในภาพที่ 2.2

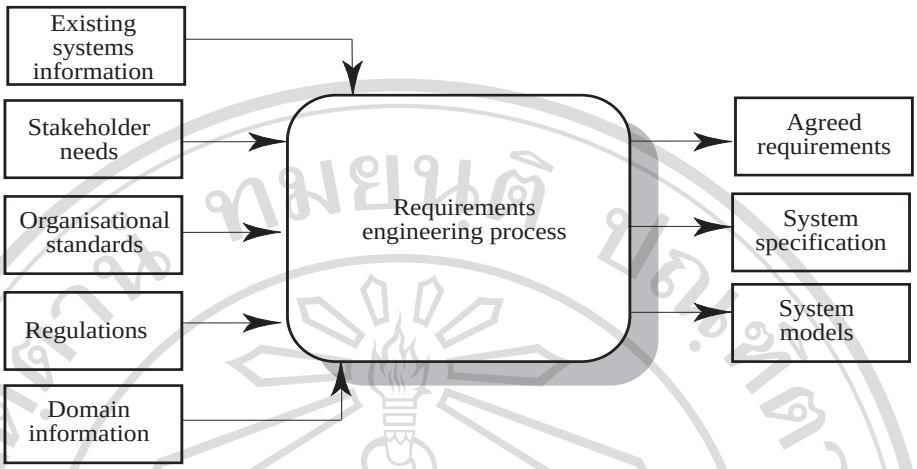


ภาพที่ 2.2 แสดงตัวอย่างของ Non – Functional Requirement

3. ความต้องการสำคัญหลัก (Domain Requirement) เป็นความต้องการที่ผู้พัฒนาระบบซอฟต์แวร์ต้องให้ความสำคัญเป็นอันดับแรกคือต้องทำระบบนี้ก่อนจึงสามารถทำระบบอื่นต่อไปได้ ซึ่ง Domain Requirement จะรวมถึง Functional Requirement ที่สำคัญและต้องเลือกทำก่อนเพื่อที่จะได้พัฒนาระบบที่เกี่ยวเนื่องต่อจากความต้องการสำคัญนี้ได้

2.2.1 กระบวนการหาวิศวกรรมความต้องการ (Requirement Engineering Process)

เป็นทฤษฎีหรือกระบวนการที่จะได้มาซึ่งความต้องการที่ถูกต้องในการเก็บความต้องการและก่อนที่จะไปเก็บความต้องการจากลูกค้าจำเป็นต้องมีกระบวนการ Requirement Engineering process ที่แน่นอนเพื่อเตรียมการทำงานให้มีประสิทธิภาพและสิ่งที่อยากได้จากขั้นตอนและวิธีการของ Requirement Engineering Process คือ Input และ Out Put ดังแสดงในภาพที่ 2.3



ภาพที่ 2.3 แสดง Requirement Engineering Process Input and Out Put

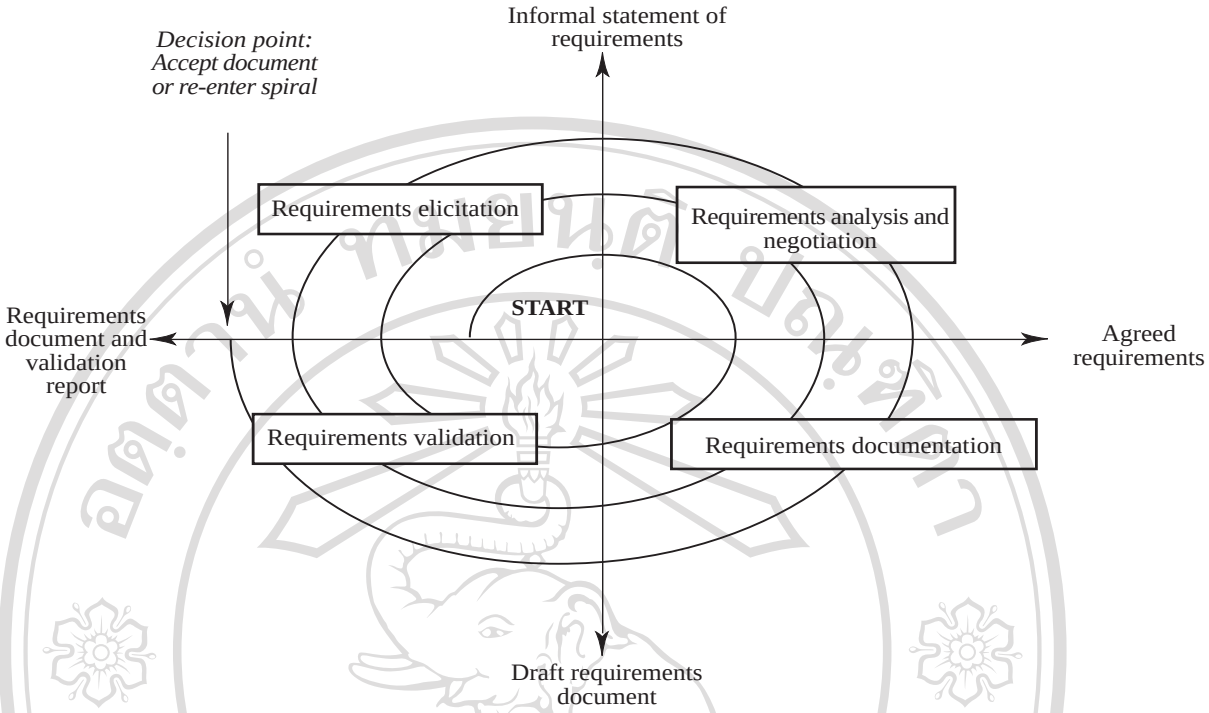
กระบวนการวิศวกรรมความต้องการที่กลุ่มนักพัฒนาซอฟต์แวร์ส่วนใหญ่เลือกใช้กันมากในปัจจุบันคือ แบบเกลียวก้นหอย (Spiral Model Requirement Engineering Process) ดังแสดงในภาพที่ 2.4 ประกอบด้วย 4 ขั้นตอนหลัก ๆ ดังนี้

1. การรวบรวมความต้องการ (Requirement Elicitations) เป็นขั้นตอนการรวบรวมความต้องการของผู้ที่เกี่ยวข้องกับระบบทั้งหมด เพื่อให้ได้ความต้องการที่คลอบคลุมทั้งระบบงาน

2. การเจรจาและการวิเคราะห์ความต้องการ (Requirements Analysis and Negotiations) เป็นขั้นตอนการวิคราะห์ข้อมูลที่ได้จากแหล่งข้อมูลที่เกี่ยวข้อง และเมื่อมีข้อขัดแย้งเกิดขึ้นก็ต้องทำการเจรจาเกี่ยวกับระบบที่อาจเปลี่ยนแปลงให้เข้าใจไปในทิศทางเดียวกัน

3. การทำเอกสารความต้องการ (Requirements Documentation) เป็นขั้นตอนการทำเอกสารต่าง ๆ เกี่ยวกับความต้องการหรือ การทำข้อกำหนดของข้อมูลที่ได้มา

4. การตรวจสอบความต้องการ (Requirements Validation) เป็นขั้นตอนการตรวจสอบเอกสารหรือสิ่งที่ได้กำหนดไว้มีความถูกต้องของข้อมูล หรือจะเป็นการทำซ้ำในส่วนที่ไม่แน่ใจเพื่อให้มั่นใจว่า Software Requirements พร้อมที่จะนำไปพัฒนาเป็นต่อไป



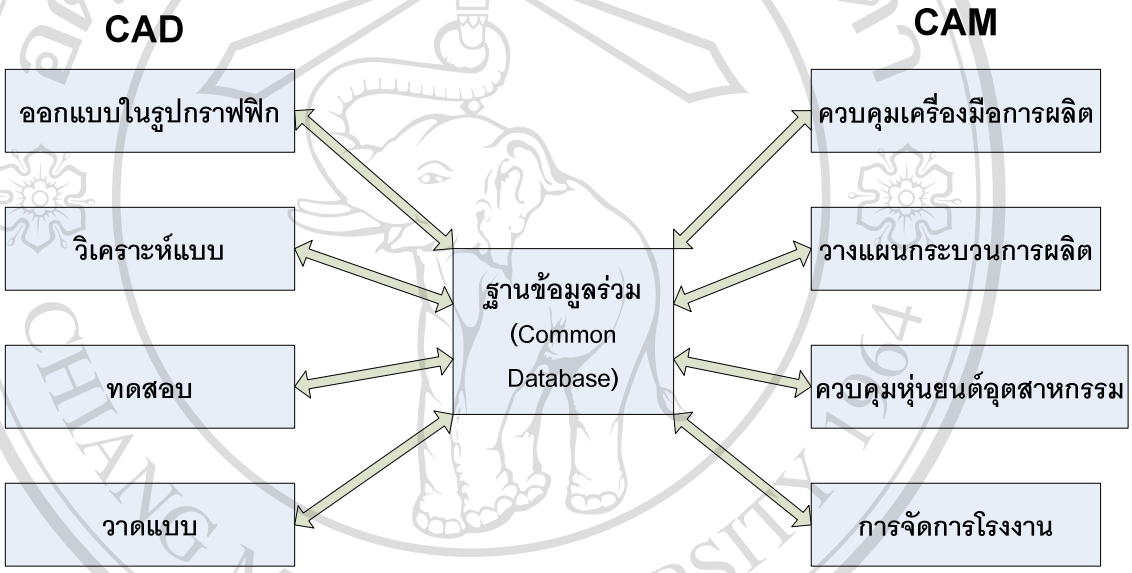
ภาพที่ 2.4 แสดงแผนภาพของกระบวนการหาความต้องการแบบเกลียวก้นหอย (Spiral Model Requirement Engineering Process)

2.3 คอมพิวเตอร์ในการพัฒนาอุตสาหกรรมและธุรกิจด้าน CAD/CAM

นับตั้งแต่ปี ค.ศ. 1950 เป็นต้นมาได้มีการประยุกต์ใช้คอมพิวเตอร์ในงานวิศวกรรม และช่วยงานอุตสาหกรรมมากขึ้นโดยเริ่มจากการใช้คอมพิวเตอร์เพื่อช่วยในการคำนวณเป็นส่วนใหญ่ ด้วยความก้าวหน้าทางด้านไมโครอิเล็กทรอนิกส์ ทำให้คอมพิวเตอร์มี ขนาดเล็กลงแต่ขีดความสามารถสูงขึ้นและที่สำคัญคือราคาถูกลงจึงมีการประยุกต์ใช้คอมพิวเตอร์ในงานต่าง ๆ กว้างขวางขึ้นซึ่งก่อให้เกิดการพัฒนาทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์เพื่อประยุกต์ใช้งานต่าง ๆ จนเกิดเทคโนโลยีใหม่ ๆ เช่น คอมพิวเตอร์กราฟิก (Computer Graphics) ซึ่งเป็นการใช้คอมพิวเตอร์เพื่อสร้างภาพในทางวิศวกรรมก็ได้มีการพัฒนาเทคโนโลยีใหม่ขึ้นโดยอาศัยพื้นฐานทางด้านคอมพิวเตอร์กราฟิกเพื่อช่วยในการสร้างแบบโดยใช้ชื่อว่า“การใช้คอมพิวเตอร์ช่วยในการออกแบบ” หรือ CAD (ซึ่งมาจาก Computer Aided Design หรือ Computer Aided Drafting) และต่อมาก็ได้มีการพัฒนาใช้คอมพิวเตอร์ช่วยในการผลิตด้วย โดยการใช้ควบคุมอุปกรณ์ที่ทำหน้าที่ผลิตสินค้าหรือชิ้นงานในโรงงานและได้เรียกเทคนิคนี้ว่า“การใช้คอมพิวเตอร์ช่วยในงานผลิต”หรือ CAM (มาจาก Computer Aided Manufacturing) และเนื่องจากคอมพิวเตอร์ช่วยในงานผลิตนั้นต้องอาศัยข้อมูลจากคอมพิวเตอร์ช่วยในการออกแบบดังนั้น งานทั้งสองอย่างนี้ จึงต้องใช้งานร่วมกัน

และจะเรียกเทคโนโลยีด้านนี้ ด้วยความเคยชินว่า CAD/CAM ซึ่ง CAD/CAM ถือว่าเป็นพื้นฐานของการประยุกต์ใช้คอมพิวเตอร์ในงานอุตสาหกรรม

หากจะใช้กระบวนการทำงานด้าน CAD/CAM ให้ได้ผลเต็มที่จะต้องสามารถส่งข้อมูลถึงกันได้ กล่าวคือ ข้อมูลที่ออกแบบโดยคอมพิวเตอร์ช่วยในการออกแบบซึ่งเป็นข้อมูลในลักษณะรูปภาพกราฟิกจะสามารถนำไปใช้ในการผลิตชิ้นงานซึ่งมีลักษณะและขนาดเช่นเดียวกับที่ออกแบบไว้ทุกประการดังนั้นหากมองในลักษณะการใช้ข้อมูลร่วมกันลักษณะของงาน CAD/CAM จะเป็นลักษณะที่แสดงในภาพที่ 2.5



ภาพที่ 2.5 แสดงความสัมพันธ์ของการใช้ข้อมูลร่วมกันระหว่างหน่วยงานต่าง ๆ ของระบบงาน CAD/CAM ในงานอุตสาหกรรม

ข้อดีของการใช้ระบบ CAD/CAM ที่นำเข้ามาช่วยในด้านการผลิต คือ การลดเวลาในการออกแบบ โดยผลงานที่ออกมามีความถูกต้องเชื่อถือได้อีกทั้งมีความยืดหยุ่นต่อการเปลี่ยนแปลงแก้ไขค่าใช้จ่ายในการสร้างผลงานอยู่ในเกณฑ์ต่ำและเมื่อมีการนำข้อมูลที่ออกแบบไว้ไปใช้งานในการผลิตจะได้ข้อมูลที่ต้องการเพราะเป็นข้อมูลชุดเดียวกันไม่ต้องมีการใส่ข้อมูลใหม่จึงลดเวลาในการผลิตลงได้มาก

2.4 ทฤษฎีการหาสมการความสัมพันธ์เชิงเส้นแบบพหุคูณ (Multiple Linear Regressions)

ปัญหาด้านวิศวกรรมและวิทยาศาสตร์โดยทั่วไปมักเกี่ยวข้องกับตัวแปรที่มีความสัมพันธ์กันตั้งแต่สองตัวขึ้นไป ปกติแล้วสมมติว่าตัวแปรตาม (Dependent Variable, Y) หรือผลตอบสนอง (Response) มีอยู่เพียงตัวเดียว ที่ขึ้นกับค่าของตัวแปรอิสระ (Independent Variable, X) จำนวน k ตัว ($X_1, X_2,..., X_k$) ความสัมพันธ์ระหว่างตัวแปรเหล่านี้ถูกกำหนดโดยแบบจำลองคณิตศาสตร์เรียกว่าแบบจำลองการถดถอย (Regression Model) ซึ่งแบบจำลองนี้จะถูกสร้างขึ้นมาให้เหมาะสมกับข้อมูลตัวอย่างเซตหนึ่ง ในบางครั้งผู้ทดลองทราบถึงรูปแบบที่แน่นอนของความสัมพันธ์ในลักษณะที่เป็นฟังก์ชันระหว่าง Y และ $X_1, X_2,..., X_k$ หรือ $Y=\phi(X_1, X_2,..., X_k)$ แต่ส่วนมากรูปแบบที่แน่นอนของความสัมพันธ์เหล่านี้เป็นสิ่งที่เราไม่ทราบ ดังนั้นผู้ทดลองต้องเลือกฟังก์ชั่นที่เหมาะสมเพื่อประมาณ ϕ ซึ่งโดยมากจะใช้แบจำลองพหุนามอันดับต่ำ (Low Order).ในการประมาณฟังก์ชัน

ในการสร้างแบบจำลองการถดถอยแบบเชิงเส้น จากข้อมูลการทดลองสามารถแสดงความสัมพันธ์ได้ดังนี้

$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\in \quad \ldots (2.1)$$

ในการหาค่าความสัมพันธ์ของปัญหาที่มีค่าตัวแปรมากกว่าสองตัวขึ้นไปนั้นในการสร้างแบบจำลองการทดลองจะอาศัยการวิเคราะห์การถดถอยแบบเชิงเส้นพหุเพื่อใช้ในการทดลองและได้รับความนิยมมากและให้ค่าความเชื่อมั่นในการวิเคราะห์หาความสัมพันธ์ของปัจจัยที่มีผลต่อต่อการพยากรณ์ ซึ่งการทดลองแบบถดถอยเชิงเส้นแบบพหุนี้ประกอบด้วยตัวแปรอิสระตั้งแต่ 2 ตัว และจะเรียกตัวแปรอิสระว่าตัวแปรทำนาย (Predictor Variable) หรือตัวถดถอย (Repressor) คำว่าเชิงเส้นถูกนำมาใช้เนื่องจากสมการดังกล่าวเป็นฟังก์ชันเชิงเส้นของพารามิเตอร์ไม่ทราบค่า β_0, β_1และ β_2 แบบจำลองแสดงให้เห็นถึงระนาบเกิน 2 มิติของ x_1, x_2 พารามิเตอร์ β_0 จะเป็นตัวกำหนดจุดตัดของระนาบ ดังสมการ 2.2, 2.3, และ สมการ 2.4

สมการที่ 2.1 แสดงสมการในรูปประชากร

$$Y=\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_k X_k+\varepsilon \quad \ldots (2.2)$$

สมการที่ 2.2 แสดงสมการในรูปของตัวอย่าง

$$Y=b_0+b_1 x_1+b_2 x_2+\ldots+b_k x_k+e \quad \ldots (2.3)$$

สมการที่ 2.3 แสดงสมการในรูปการพยากรณ์ (การประมาณค่า)

$$\hat{Y} = b_0 + b_1x_1 + b_2x_2 + \ldots + b_kx_k \quad \ldots (2.3)$$

ซึ่งสมการรูปแบบนี้เรียกว่า แบบจำลองการถดถอยแบบเชิงเส้นพหุคูณที่มีตัวแปรถดถอย k ตัว (Multiple Linear Regression Model with k Repressors Variables) พารามิเตอร์ β_j, j = 0, 1, …., k ถูกเรียกว่า สัมประสิทธิ์การถดถอย พารามิเตอร์ β_j แสดงถึงการเปลี่ยนแปลงที่เกิดขึ้นกับผลตอบ Y ต่อหนึ่งหน่วยของการเปลี่ยนแปลงที่เกิดขึ้นกับ x_j เมื่อตัวแปรอิสระที่เหลือทั้งหมด x_i (i=j) มีค่าคงตัว

2.4.1 เงื่อนไขการวิเคราะห์ความถดถอยเชิงพหุ

เงื่อนไขของการวิเคราะห์ความถดถอย เป็นเงื่อนไขที่เกี่ยวข้องกับความคลาดเคลื่อน (Error or Residual) ซึ่งจะต้องทำการตรวจสอบเงื่อนไขของการวิเคราะห์ความถดถอยเกี่ยวกับค่าคลาดเคลื่อนดังนี้ (กัลยา, 2546)

1 ค่าความคลาดเคลื่อน e เป็นตัวแปรที่มีการแจกแจงปกติ

2 ค่าเฉลี่ยความคลาดเคลื่อนเป็นศูนย์ E (e) =0

3 ค่าความแปรปรวนของความคลาดเคลื่อนเป็นค่าคงที่ V (e) = σ_e^2

4 ค่าความคลาดเคลื่อนต้องเป็นอิสระต่อกัน

5 ตัวแปรอิสระต้องเป็นอิสระต่อกัน

ทฤษฎีการออกแบบการทดลองและแบบจำลองการถดถอยเชิงเส้นพหุคูณดังที่กล่าวมาข้างต้นจะถูกนำมาหาความสัมพันธ์ของปัจจัยที่มีผลกระทบในการทำงานด้าน CAD/CAM ต่อการประเมินราคาและระยะเวลาของการผลิตชิ้นงานต้นแบบแม่พิมพ์

2.5 ทฤษฎีการคำนวณค่าไฟฟ้าที่ใช้ในมอเตอร์สามเฟส

ค่าพลังงานไฟฟ้า หมายถึง ค่ากำลังไฟฟ้าของเครื่องใช้ไฟฟ้าที่เปิดใช้งานในหน่วยกิโลวัตต์คูณด้วย จำนวนชั่วโมงที่ใช้งาน มีหน่วยเป็น กิโลวัตต์-ชั่วโมง หรือเรียกว่า 1 หน่วย หรือ 1 ยูนิต กรณีการคิดค่าพลังงานที่ใช้ในมอเตอร์สามเฟสมีสูตรในการคำนวณ ดังต่อไปนี้

ค่าไฟฟ้าที่ใช้ในหนึ่งเดือน = ((Power x ชั่วโมงการใช้งานในหนึ่งเดือน)/1000) x อัตราค่าไฟฟ้าต่อหน่วย..... (บาท/เดือน ... (2.5)