บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาค้นคว้าด้วยตัวเอง ระบบวินิจฉัยโรคทางไกลโดยใช้ระบบผู้เชี่ยวชาญช่วยในการตั้งคำถาม มีทฤษฎีและงานวิจัยที่เกี่ยวข้องดังนี้

1. ระบบผู้เชี่ยวชาญ (Expert System)
2. ระบบฐานข้อมูล (Database)
3. เว็บเบสแอปพลิเคชั่น (Web-Base Application)
4. หลักการวิศวกรรมซอฟแวร์ (Software Engineering)
5. Telemedicine
6. งานวิจัยที่เกี่ยวข้อง

2.1 ระบบผู้เชี่ยวชาญ (Expert System)

ระบบผู้เชี่ยวชาญหรือเรียกว่า ระบบฐานข้อมูลความรู้ (knowledge-base) เป็นสาขาย่อยของปัญญาประดิษฐ์ (artificial intelligence) โดยการนำความรู้ทางด้านปัญญาประดิษฐ์มาผนวกเข้ากับภาษา หรือเครื่องมือที่ถูกสร้างให้ใกล้เคียงกับตรรกะของมนุษย์ ทำให้สามารถจำลองความเชี่ยวชาญของมนุษย์ในขอบเขตปัญหาที่สนใจได้ ความสำเร็จของระบบจะขึ้นอยู่กับคุณภาพของข้อมูลและความรู้ที่ได้มาจากผู้เชี่ยวชาญ การโปรแกรมแบบอิงกฎเป็นเทคนิคหนึ่งที่ใช้กันมาในการพัฒนาระบบผู้เชี่ยวชาญ ในการโปรแกรมแบบนี้กฎจะถูกใช้เพื่อแสดงวิทยาการศึกษาสำนึก (heuristics) ซึ่งระบุการกระทำในสถานการณ์ที่กำหนด กฎหนึ่งจะประกอบด้วยส่วนของ if และส่วนของ then ส่วนของ if จะเป็นชุดของรูปแบบที่ระบุข้อมูลหรือเงื่อนไขซึ่งทำให้กฎถูกใช้งาน ระบบผู้เชี่ยวชาญจะใช้กลไกที่เรียกว่า เครื่องอนุมาน (inference engine) จับคู่ข้อมูลกับรูปแบบอัตโนมัติและกำหนดกฎที่จะถูกใช้ ส่วนของ then จะเป็นชุดของการทำงานที่จะถูกเรียกเมื่อกฎถูกใช้ เครื่องอนุมานจะเลือกกฎและการกระทำของกฎที่ถูกเลือกจะถูกเรียกขึ้นมาทำงาน

ในปัจจุบันนี้มีระบบสารสนเทศอยู่มากมายโดยแต่ละระบบได้นำมาประยุคใช้ต่างกันไป ซึ่งระบบสารสนเทศที่จะศึกษาเป็นระบบที่นำมาประยุคใช้เกี่ยวกับการคิดวิเคราะห์ หาคำตอบ โดยมีการนำข้อมูลจากคลังความรู้ที่มี มาใช้เพื่อตัดสินใจ โดยระบบนี้ก็คือระบบผู้เชี่ยวชาญ (Expert System) ซึ่งระบบผู้เชี่ยวชาญ เป็นระบบสารสนเทศประเภทหนึ่งที่ถูกออกแบบมาให้สามารถคิดวิเคราะห์ เปรียบเทียบ คำตอบสำหรับสถานการณ์ใดๆ ลักษณะที่กล่าวมาของ ระบบผู้เชี่ยวชาญได้ถูกจำลองหรือลอกเรียนแบบมาจาก วิธีการคิด วิเคราะห์ของมนุษย์ หรือผู้เชี่ยวชาญในสาขานั้นๆ

ซึ่งจะเกี่ยวข้องกับการจัดการความรู้ (knowledge) มากกว่าสารสนเทศ และถูกออกแบบให้ช่วยในการแสดงข้อมูลความจริงจากองค์ความรู้ที่มี ซึ่งใช้วิธีเดียวกับผู้เชี่ยวชาญที่มนุษย์เป็น ซึ่งระบบผู้เชี่ยวชาญนี้จะมีลักษณะคล้ายกับ ระบบสนับสนุนการตัดสินใจ เนื่องจากจะช่วยในการแสดงองค์ความรู้แล้วยังสามารถช่วยในการตัดสินใจของมนุษย์ได้อีกด้วยโดยใช้เทคนิคต่างๆ ของระบบผู้เชี่ยวชาญ ไม่ว่าจะเป็นการอนุมานหรือการสรุปความ เช่น การอนุมานแบบไปข้างหน้า (Forward Chaining) การอนุมานแบบย้อนกลับ (Backward Chaining) และการนำความรู้มาใช้ หรือการแทนความรู้

2.1.1 โครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญโดยทั่วไปจะประกอบพื้นฐาน 5 ส่วน ที่เป็นหัวใจที่ขาดไม่ได้คือ ฐานความรู้และเครื่องอนุมาน (inference engine) รายละเอียดโดยย่อของแต่ละส่วนสามารถอธิบายได้ดังนี้

2.1.1.1 ฐานความรู้ (Knowledge Base)

ในส่วนนี้จะเปรียบเสมือนกับข้อมูลในซอฟต์แวร์ธรรมดา หรือฐานข้อมูล (Database) ในระบบสารสนเทศ (information system) เป็นส่วนที่เก็บความรู้ทุกประเภทไม่ว่าจะเป็นความรู้ที่ได้จากตำราหรือ ความรู้ที่เหมาะสม และที่ได้จากประสบการณ์ ปัญหาหลักในการสร้างฐานความรู้ก็คือ การเลือกวิธีการแสดงความรู้หรือโครงสร้างสำหรับเก็บความรู้ที่เหมาะสม

2.1.1.2 กลไกการอนุมาน หรือสรุปความ (Inference Engine)

เป็นส่วนที่จะนำความรู้จากองค์ความรู้ที่เก็บไว้ในฐานความรู้ (Knowledge Base) ไปใช้งาน ซึ่งจำเป็นต้องใช้โปรแกรมคอมพิวเตอร์สำหรับประมวลผล และควบคุมการสรุปความ

อาจจะกล่าวได้ว่ากลไกการอนุมานเป็นเครื่องมืออย่างหนึ่ง ที่ไม่เพียงแต่จะเกี่ยวข้องและสัมพันธ์โดยตรงกับการค้นหา (search) องค์ความรู้ทั้งหมดที่มีอยู่ในฐานความรู้ และการเปรียบเทียบรูปแบบ (Pattern Matching) แล้วยังมีการควบคุมการตรวจสอบ (Investigate) การขจัด (Eliminate) และการจับคู่ (Matching) ของกฎต่างๆ เพื่อให้ได้ผลลัพธ์ของการสรุปความที่เหมาะสม ซึ่งการอนุมาน หรือการสรุมความมีอยู่ด้วยกันหลายวิธี แต่จะกล่าวถึงในส่วนของ การอนุมาน หรือการสรุปความโดยใช้กฎ ที่จะกล่าวในต่อไปนี้

การสรุปความโดยใช้กฎ (Rule-Base System) กฎทุกข้อที่อยู่ในฐานความรู้ จะต้องตรวจสอบได้ว่าทั้งข้อสมมติฐาน และข้อสรุปนั้นสามารถนำมาใช้อ้างถึง หรือยืนยัน ระหว่างกฎด้วยกันได้ การสรุปความด้วยกฎไม่เพียงแต่จะมีความสัมพันธ์กับการโมดัสโพเน้นส์ (Modus Ponens) โดยตรงแล้วยังเกี่ยวข้องกับกลไกการค้นหา (Search Mechanism) อีกด้วย เช่น

IF มีน้ำมูกไหล AND เจ็บคอ
THEN อาจจะเป็นหวัด

วิธีในการควบคุมการสรุปความ ด้วยการใช้กฎสำหรับระบบผู้เชี่ยวชาญสามารถแบ่งออกได้เป็น 2 ชนิด ได้แก่ การสรุปความแบบไปข้างหน้า (Forward Chaining) และการสรุปความแบบย้อนกลับ (Backward Chaining) ซึ่งเปรียบได้กับขั้นตอนวิธี (Algorithm) เป็นส่วนที่ควบคุมการใช้ความรู้ในฐานความรู้เพื่อแก้ปัญหาได้อย่างมีประสิทธิภาพ โดยในการศึกษาระบบวินิจฉัยโรคทางไกลโดยใช้ระบบผู้เชี่ยวชาญช่วยในการตั้งคำถาม จะใช้การอนุมานแบบไปข้างหน้าเท่านั้น ซึ่งมีรายละเอียดดังนี้

การสรุปความแบบไปข้างหน้า (Forward Chaining) การอนุมานแบบไปข้างหน้าเป็นการอนุมานโดยเริ่มการตรวจสอบข้อมูล กับกฎเกณฑ์ที่มีอยู่ในระบบจนกว่าสามารถหากฎเกณฑ์ ที่สอดคล้องกับสถานการณ์ แล้วจึงดำเนินการตามเหมาะสม ซึ่งการสรุปความแบบไปข้างหน้าเป็นวิธีการมุ่งเน้นความสำคัญที่ข้อมูล (Data Driven) โดยได้เริ่มจากการค้นหาสารสนเทศที่มีอยู่ในฐานความรู้ หรือจากแนวความคิดพื้นฐานเพื่อให้ได้ซึ่งผลลัพธ์ของการสรุปความออกมา ทั้งนี้ในระบบผู้เชี่ยวชาญจะทำการวิเคราะห์ถึงปัญหา โดยการตรวจสอบข้อเท็จจริงดูว่าสอดคล้องกับส่วน IF ของกฎหรือไม่ด้วยการทดสอบกฎก่อนที่ โปรแกรมจะทำการประมวลผลจนกว่าจะได้ข้อสรุป

ในลักษณะของการทำงานก่อนหลังในระบบฐานกฎ (Rule-Base System) จะช่วยในการค้นหาเพื่อการสรุปความ แบบไปข้างหน้าและแบบย้อนกลับ (Forward and Backward Chaining) ทำงานได้ดียิ่งขึ้น ทั้งนี้ ก็ขึ้นอยู่กับวัตถุประสงค์ของการให้เหตุผลและขอบเขต ของการค้นหา

2.1.1.3 ส่วนการได้มาซึ่งความรู้ (Knowledge Acquisition Subsystem)

เป็นส่วนของระบบผู้เชี่ยวชาญที่ช่วยในการดึงเอาความรู้จากตำราหรือฐานข้อมูล และจากผู้เชี่ยวชาญ การดึงเอาความรู้จากตำราหรือฐานข้อมูลเป็นสิ่งที่ทำได้ไม่ยาก แต่สิ่งที่ยากคือการดึงเอาความรู้จากผู้เชี่ยวชาญ ซึ่งจำเป็นต้องใช้เทคนิคต่างๆ เข้ามาช่วย หรือไม่ก็ทำให้ระบบผู้เชี่ยวชาญสามารถเรียนรู้ได้ด้วยตนเองในบางส่วนได้ ซึ่งเทคนิคต่างๆ มีอยู่หลายเทคนิคด้วยกัน ไม่ว่าจะเป็น ทฤษฎีเบส์ (Bayesian Theory) การให้ความเชื่อมั่น (Certainty Factors) เป็นต้น

2.1.1.4 ส่วนอธิบาย (Explanation Subsystem)

ทำหน้าที่อธิบายรายละเอียดของขั้นตอนการวินิจฉัยต่อผู้ใช้ว่าข้อสรุปหรือคำตอบนั้นได้มาอย่างไร

2.1.1.5 ส่วนเชื่อมต่อกับผู้ใช้ (User Interface)

เป็นส่วนที่เป็นตัวกลางระหว่างผู้ใช้กับระบบเพื่อทำให้การสื่อสารระหว่างผู้ใช้กับระบบเป็นไปอย่างราบรื่น เนื่องจากผู้ใช้มีความรู้ในงาน สารสนเทศที่แตกต่างกัน หรือผู้ใช้บางคนไม่เคยชินกับการรับคำแนะนำจากระบบสารสนเทศ ตลอดจนผู้ใช้มีความต้องการที่หลากหลาย ดังนั้นผู้พัฒนาระบบจึงต้องคำนึงถึงความสะดวกในการติดต่อระหว่างระบบผู้เชี่ยวชาญกับผู้ใช้

2.2 ระบบฐานข้อมูล (Database System)

เป็นกลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน เพื่อลดความซ้ำซ้อนของข้อมูลและเก็บข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน

การจัดการฐานข้อมูล (Database Management) คือ การบริหารแหล่งข้อมูลที่ถูกเก็บรวบรวมไว้ที่ศูนย์กลาง เพื่อตอบสนองต่อการใช้งานอย่างมีประสิทธิภาพและลดการซ้ำซ้อนของข้อมูล รวมทั้งลดความขัดแย้งของข้อมูลที่เกิดขึ้นภายในองค์กรด้วย

การจัดการฐานข้อมูลต้องอาศัยโปรแกรมที่ทำหน้าที่ในการกำหนดลักษณะข้อมูลที่จะเก็บไว้ในฐานข้อมูล อำนวยความสะดวกในการบันทึกข้อมูลลงในฐานข้อมูล กำหนดผู้ที่ได้รับอนุญาตให้ใช้ฐานข้อมูลได้ พร้อมกับกำหนดด้วยว่าให้ใช้ได้แบบใด เช่น ให้อ่านข้อมูลได้อย่างเดียวหรือให้แก้ไขข้อมูลได้ด้วย นอกจากนั้นยังอำนวยความสะดวกในการค้นหาข้อมูล การแก้ไขปรับปรุงข้อมูล ตลอดจนการจัดทำข้อมูลสำรองด้วย โดยอาศัยโปรแกรมที่เรียกว่า ระบบการจัดการฐานข้อมูล

(Database Management System: DBMS) ซึ่งโปรแกรมที่ได้รับความนิยมในการจัดการฐานข้อมูล ได้แก่ Microsoft Access, Oracle, Informix, dBase, FoxPro, และ Paradox เป็นต้น

2.2.1 ส่วนประกอบของตารางข้อมูลในฐานข้อมูล

โดยทั่วไปแล้วตารางข้อมูลที่ใช้งานกันจะประกอบด้วย แถว (Row) และคอลัมน์ (Column) ต่างๆ แต่ถ้ามองกันในรูปแบบของฐานข้อมูลแล้ว เราจะเรียกรายละเอียดในแถวว่า เรคคอร์ด (Record) และเรียกรายละเอียดในแนวคอลัมน์ว่า ฟิลด์ (Field)

ในฐานข้อมูล 1 ระบบ อาจประกอบด้วยตารางข้อมูลมากกว่า 1 ตาราง ฐานข้อมูลที่มีตารางข้อมูลมากกว่า 1 ตาราง และมีตารางตั้งแต่ 1 คู่ขึ้นไปที่มีความสัมพันธ์กันด้วยฟิลด์ใดฟิลด์หนึ่ง เราเรียกฐานข้อมูลประเภทนี้ว่า "ฐานข้อมูลเชิงสัมพันธ์" หรือ Relational Database

2.2.2 โครงสร้างของฐานข้อมูลประกอบด้วย

2.2.2.1 Character คือ ตัวอักขระแต่ละตัว / ตัวเลข / เครื่องหมาย

2.2.2.2 Field คือ เขตข้อมูล / ชุดข้อมูลที่ใช้แทนความหมายของสื่อโครงสร้าง เช่น ชื่อของบุคคล ชื่อของวัสดุสิ่งของ

2.2.2.3 Record คือ ระเบียน หรือรายการข้อมูล เช่น ระเบียนของพนักงานแต่ละคน

2.2.2.4 Table /File คือ ตาราง หรือแฟ้มข้อมูล ประกอบขึ้นด้วยระเบียนต่างๆ เช่น ตารางข้อมูลของบุคคล ตารางข้อมูลของวัสดุสิ่งของ

2.2.2.5 Database คือ ฐานข้อมูล ประกอบด้วยตาราง และแฟ้มข้อมูลต่างๆ ที่เกี่ยวข้อง หรือมีความสัมพันธ์กัน

2.3 เว็บเบสแอปพลิเคชั่น (Web-Base Application)

ระบบงานที่ถูกพัฒนาขึ้นใช้งานบนบราวเซอร์ผ่านระบบเครือข่าย ซึ่งทำงานได้ทั้งบนอินเตอร์เน็ต

2.3.1 ข้อดีของเว็บเบสแอพพลิเคชั่น

2.3.1.1 ข้อมูลต่าง ๆ ในระบบมีการไหลเวียนในแบบ Online ทั้งแบบ Local (ภายในวง LAN) และ Global (ออกไปยังเครือข่ายอินเตอร์เน็ต) ทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time

2.3.1.2 ระบบมีประสิทธิภาพ แต่ใช้งานง่าย เหมือนกับท่านทำกำลังท่องเว็บ

2.3.1.3 ระบบงานที่พัฒนาขึ้นมาจะตรงกับความต้องการกับหน่วยงาน หรือห้างร้านมากที่สุด ไม่เหมือนกับโปรแกรมสำเร็จรูปทั่วไป ที่มักจะจัดทำระบบในแบบกว้างๆ ซึ่งมักจะไม่ตรงกับความต้องการที่แท้จริง

2.3.1.4 ระบบสามารถโต้ตอบกับลูกค้า หรือผู้ใช้บริการแบบ Real Time ทำให้เกิดความประทับใจ เครื่องที่ใช้งานไม่จำเป็นต้องติดตั้งโปรแกรมใดๆ เพิ่มเติมทั้งสิ้น

2.3.2 ความรู้เกี่ยวกับภาษาพีเอชพี (PHP)

Rasmus Lerdorf ผู้สร้างภาษา PHP ได้เริ่มจากการเขียนสคริปต์ Perl CGI ใส่ไว้ในโฮมเพจประวัติส่วนตัว เพื่อบันทึกข้อมูลผู้ที่เข้าเยี่ยมชมโฮมเพจ แต่เนื่องจาก Lerdrf เห็นว่าการเขียน CGI ด้วย Perl นั้นออกจะเยิ่นเย้อเกินไป จึงได้ตัดสินใจเขียนโปรแกรมขึ้นใหม่ด้วยภาษา C ที่สามารถแยกส่วนที่เป็นภาษา HTML ออกจากส่วนที่เป็นภาษา C เพื่อแยกประมวลผลแล้วทำการสร้างโค้ด HTML ขึ้นมาใหม่ โดยตั้งชื่อโปรแกรมนี้ว่า Personal Home Page Tools (PHP-Tools) และได้เริ่มแจกจ่ายโค้ดออกไปในลักษณะฟรีแวร์ (ซึ่งในขณะนั้น Open source ยังไม่เป็นที่รู้จักกันมากนัก) ต่อมาจึงได้เริ่มเปิดให้ผู้ที่สนใจเข้าร่วมปรับปรุงและพัฒนา จนพัฒนาเป็น PHP/FI ที่เริ่มเป็นที่นิยมมากขึ้น

จนกระทั้ง Zeev Suraski และ Andi Gutmans ได้ร่วมกันเขียนโค้ดขึ้นใหม่โดยได้มีการปรับปรุงให้ดีขึ้นเป็นอย่างมากในหลายๆด้าน ทั้งด้านประสิทธิภาพ การสนับสนุนการโปรแกรมเชิงวัตถุ และในด้านอื่นๆ อีกมากมายหลายประการจนเกิดเป็น PHP 3 ซึ่งเป็นเวอร์ชั่นที่ได้รับความนิยมเป็นอย่างมาก

แต่เมื่อมีผู้ใช้เป็นจำนวนมาก จึงมีการนำไปใช้ในงานที่ซับซ้อนขึ้น ด้วยเหตุนี้ Zeev Suraski และ Andi Gutmans ผู้พัฒนา PHP 3 จึงตัดสินใจเขียนโค้ดขึ้นใหม่ทั้งหมด และได้ตั้งชื่อว่า Zend engine (มาจาก Zeev และ Andi) ซึ่งเป็นหัวใจของ PHP 4 ในปัจจุบัน ส่วน PHP 5 เป็นเวอร์ชันที่จัดได้ว่าเป็นการพลิกโฉมการโปรแกรมเชิงวัตถุด้วย PHP เนื่องจากมีการเปลี่ยนแปลงไปสู่การโปรแกรมเชิงวัตถุที่สมบูรณ์แบบมากยิ่งขึ้น

PHP เป็นภาษาสคริปต์แบบเซิร์ฟเวอร์ไซด์ (server-side scripting language) หมายถึงการประมวลผลจะเกิดขึ้นบนเครื่องแม่ข่าย หรือเซิร์ฟเวอร์ (server) แล้วจึงสร้างผลลัพธ์เป็นภาษา HTML ส่งมาให้กับเครื่องลูกข่าย หรือไคลเอนท์ (client) เพื่อแสดงผล ซึ่งลดภาระการส่งข้อมูลจำนวนมากเพื่อมาประมวลผลบนเครื่องลูกข่าย

ภาษา PHP เป็นโอเพ่นซอร์ส (Open Source) ซึ่งสามารถดาวน์โหลด PHP (พร้อม source code) มาใช้งานได้ฟรีจากเว็ปไซต์ของ PHP (www.php.net/download.php) ส่วนคู่มือการใช้งาน (PHP Manual) นั้นสามารถเรียกดูได้จาก www.php.net/docs.php ซึ่งสามารถเรียกดูในแบบออนไลน์ได้ทันที (HTML) หรือหากต้องการดาวน์โหลดก็มีให้เลือกทั้งในรูปแบบเว็บเพจ (HTML) และไฟล์ช่วยเหลือในแบบของ Windows (.chm) โดยสามารถดาวน์โหลดได้ที่ www.php.net/download-docs.php

2.3.3 MySQL

เป็นโปรแกรมฐานข้อมูล มีหน้าที่เก็บข้อมูลอย่างมีโครงสร้าง และรองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูลอย่างมืออาชีพ ยังมีเครื่องมืออีกหลายอย่าง ที่ท่านต้องใช้ร่วมกันอย่างสอดคล้อง จึงจะนำไปพัฒนาระบบฐานข้อมูลซับซ้อน ตามความต้องการของผู้ใช้ได้สำเร็จสมประสงค์ เช่น การบริการเว็บ ภาษาสำหรับพัฒนาเว็บ ระบบปฏิบัติการ และคอมพิวเตอร์ที่เหมาะสม การใช้ MySQL ในฐานะนักเรียน เป็นเพียงจุดเริ่มต้นของการสร้างระบบที่สมบูรณ์ แม้นักเรียนจะพัฒนาระบบฐานข้อมูลเป็นโครงงานก่อนจบได้สมบูรณ์ แต่นั่นก็เป็นเพียงระบบหนึ่ง การหาเวลาศึกษาหลายๆ ระบบจะทำให้นักเรียนเข้าใจระบบฐานข้อมูลมากขึ้น

2.3.3.1 ข้อดีของ MySQL

1. MySQL มีขนาดเล็กและใช้ทรัพยากรน้อย
2. เป็นฐานข้อมูลที่ได้รับความนิยมมากที่สุดในการเขียนแอพพลิเคชัน PHP
3. มีเครื่องมือมากมายในการจัดการ ทั้งแบบที่เป็นกราฟฟิกและเว็บ

4. สามารถติดตั้งบนวินโดวส์เช่นเดียวกันกับ Linux/FreeBSD

2.4 หลักการวิศวกรรมซอฟแวร์ (Software Engineering)

กระบวนการในการออกแบบและพัฒนาซอฟต์แวร์ด้วยวิธีการทางวิศวกรรมนี้ก็ยังคงมีลักษณะเป็นการศึกษาหาวิธีดำเนินงานที่เหมาะสมที่สุด เพราะเท่าที่ได้พัฒนาขั้นตอนการดำเนินการแบบต่างๆ ตลอดมา ปรากฏว่าเราไม่อาจหาข้อสรุปรวมได้ว่า วิธรการใดเป็นวิธรการที่ดีที่สุด บางวิธีอาจจะดีกว่าวิธีการหนึ่งในสถานการณ์ที่แตกต่างกันออกไป ดังนั้นจึงไม่อาจมีคำตอบที่แน่นอนสำหรับทุกสถานการณ์ได้

2.4.1 Software Process มีอยู่ 4 ขั้นตอนใหญ่ๆ ดังนี้

2.4.1.1 Specification คือ ขั้นตอนการกำหนดคุณสมบัติและความต้องการของระบบ

2.4.1.2. Design คือ ขั้นตอนการออกแบบระบบ หรือ ออกแบบตัว Software

2.4.1.3. Validation คือ รวมตั้งแต่การพัฒนาขึ้นมาและทดสอบว่ามันใช้งานได้จริงตามต้องการหรือป่าว

2.4.1.4. Evolution คือพัฒนาการของ Software ที่มีการปรับเปลี่ยนไปตามความต้องการของผู้ใช้ หรือความเหมาะสมให้มีประสิทธีภาพมากขึ้น

2.4.1.1 ขั้นตอนที่เกี่ยวข้องกับความต้องการ (Requirement)

- ศึกษาความเป็นไปได้ เป็นสิ่งแรกที่ควรจะทำในการพัฒนาระบบหรือซอฟต์แวร์ใดๆ ซึ่งจะเป็นการศึกษาว่าเป็นไปได้หรือไม่ที่จะพัฒนาหรือพัฒนาและจะพบกับปัญหาอะไรมากน้อยแค่ไหน
- เก็บรวบรวมและวิเคราะห์ความต้องการ เกิดจากการเลือกใช้เครื่องมือหรือเทคนิคในการเก็บรวบรวมข้อมูล ซึ่งเทคนิคหนึ่งคือการสัมภาษณ์ผู้ใช้หรือผู้มีส่วนได้ส่วนเสียกับการพัฒนาระบบซอฟต์แวร์ และนำมาวิเคราะห์ข้อมูล ผลลัพธ์ที่ได้คือ โมเดลระบบ (System Model)
- กำหนด Specification ซึ่งเป็นกระบวนการกำหนด Specification และอาจมีการเก็บ Requirement เพิ่มเติมในกรณีที่ไม่ถูกต้องหรือไม่เพียงพอ ทำให้ได้ User Requirement และ System Requirement ซึ่งมีลักษณะการอธิบายเป็นหัวข้ออย่าง

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

ชัดเจนว่าต้องการให้ระบบทำอะไรบ้าง

- ตรวจสอบ Requirement ว่าถูกต้อง ไม่คลุมเครือ และนำไปให้ผู้ใช้หรือผู้มีส่วนได้ส่วนเสียกับการพัฒนาระบบซอฟต์แวร์อ่านและตรวจสอบว่าตรงตามความต้องการหรือไม่ ซึ่งถ้าตรงความต้องการ จะนำไปเขียนเป็นเอกสารความต้องการ (Requirement Document)

2.4.1.2 กระบวนการออกแบบระบบ (System Design)

เราสามารถนำขั้นตอนของ Requirement Specification มาเขียนและออกแบบได้ดังต่อไปนี้

- การออกแบบสถาปัตยกรรมระบบ (System architecture) หรือมักจะเรียกเป็น Block Diagram ซึ่งจะบอกว่าระบบประกอบด้วยส่วนสำคัญอะไรบ้าง มีการเชื่อมโยงกันอย่างไร ต่อมาก็เขียน Software Specification (หรือ Functional Specification) ซึ่งบอกว่าซอฟต์แวร์สามารถทำอะไรได้บ้าง
- การออกแบบ Interface ได้เป็น Interface Specification ซึ่งจะเป็นรายละเอียดของ Interface โดยเฉพาะ เช่น ลักษณะหน้าจอเป็นอย่างไร ใช้อุปกรณ์อะไรเป็น Input / Output ตรงส่วนไหนใช้เมาส์ คีย์บอร์ด หรือกดปุ่มไหนทำอะไรบ้าง
- การแบ่งระบบทั้งหมดออกเป็นระบบย่อยๆ (Component Specification) ว่าส่วนไหนทำอะไรบ้าง มีฟังก์ชั่นอะไร
- การออกแบบและจัดการฐานข้อมูล (Database System) จะวิเคราะห์ส่วนของข้อมูลว่าควรจะมีการจัดเก็บอย่างไรในรูปแบบอะไรและใช้ตัวเชื่อมโยงอะไร
- การออกแบบระบบอัลกอริทึม ซึ่งคือลำดับขั้นตอนของความคิดในการแก้ปัญหาการทำงานของโปรแกรม

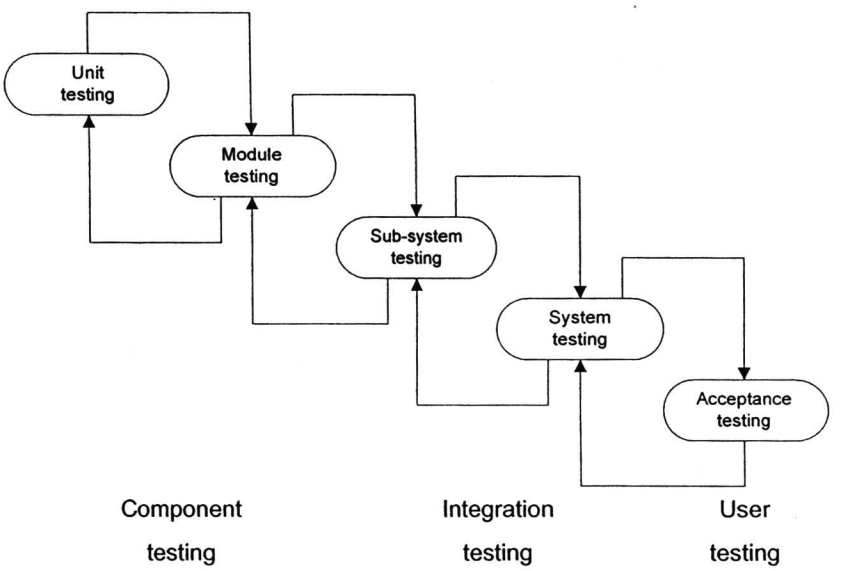
2.4.1.3 ขั้นตอนการออกแบบระบบการตรวจสอบ (Validation)

แบ่งออกเป็นส่วน 3 ส่วนใหญ่ๆ ดังนี้

- ทดสอบแต่ละส่วนประกอบของระบบ (Component testing)
- ทดสอบการนำส่วนประกอบของระบบมารวมเข้าด้วยกัน (Integration testing)
- ทดสอบใช้งานจริงโดยผู้ใช้ (User testing)

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย
วันที่ 22 มิ.ย. 2555
เลขทะเบียน 246110
เลขเรียกหนังสือ

การเลือกว่าจะใช้วิธีไหนในการทดสอบขึ้นอยู่กับความเหมาะสมของขนาดของ Software และความต้องการระดับของความละเอียดของโปรแกรม



ภาพ 2-2 ขั้นตอนการตรวจสอบ

Unit testing: เป็นการทดสอบว่ามีการกำหนดฟังก์ชั่นในการเรียกใช้อะไรบ้าง โดยบอกได้ว่ารับ input อะไรเข้ามา แล้วได้ output อะไรออกไปบ้าง

Module testing: เป็นการทดสอบความถูกต้องของฟังก์ชันการทำงานในแต่ละส่วนย่อยๆ ว่าทำงานได้ถูกต้องและมีประสิทธิภาพเพียงพอหรือไม่

Sub-system testing: เป็นการทดสอบระบบย่อยๆ ก่อนนำเข้ามารวมเป็นระบบใหญ่ เช่น ระบบบัญชี ระบบการเงิน ระบบบุคลากร

System testing: เป็นการทดสอบระบบใหญ่หรือระบบที่เป็นโครงการทั้งหมด ภายหลังจากที่นำระบบย่อยที่ผ่านการทดสอบมารวมกันแล้ว

Acception-test: เป็นการทดสอบความถูกต้องและการยอมรับครั้งสุดท้ายจาก user ก่อนการส่งมอบ เช่น การทดสอบส่วน user interface

2.4.1.4 ขั้นตอนการพัฒนาซอฟต์แวร์ ที่มีการปรับเปลี่ยนตามความต้องการของผู้ใช้

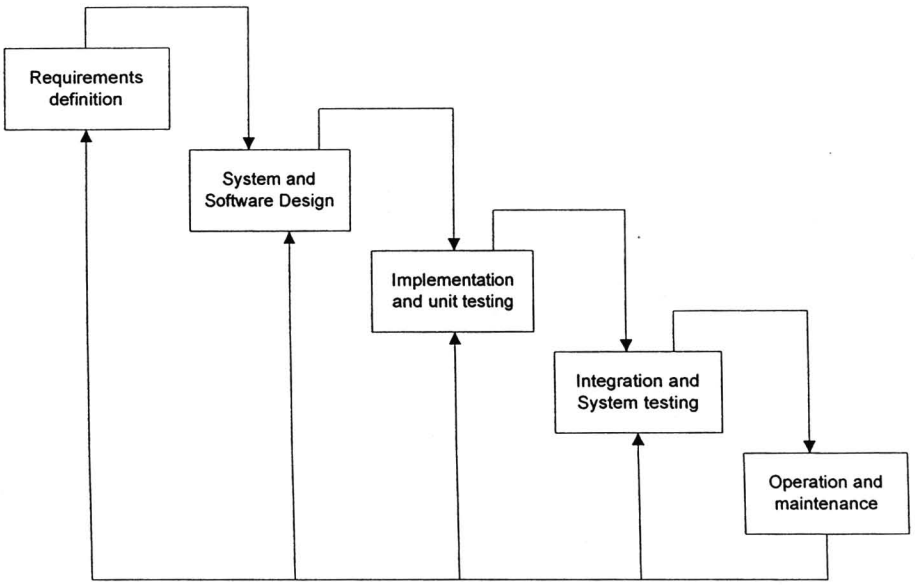
เป็นขั้นตอนการตรวจสอบระบบที่พัฒนาอยู่ในปัจจุบันมาวิเคราะห์กับความต้องการที่ระบบได้ผ่านการวิเคราะห์และลงความเห็นว่าเป็นความต้องการจากผู้ใช้ที่ถูกต้อง เทียบกับความต้องการจากผู้ใช้ที่เกิดการเปลี่ยนแปลงในระหว่างการพัฒนา เพื่อให้ทราบว่าระบบมีข้อบกพร่องเมื่อเทียบกับความต้องการเดิมหรือไม่ หรือระบบควรมีการปรับปรุงเปลี่ยนแปลงหรือมีส่วนไหนบ้างที่จำเป็นต้องเพิ่มเติม ซึ่งจะได้ความต้องการใหม่และจะเป็นกระบวนการกระทำซ้ำจนกว่าจะได้ความต้องการที่ถูกต้องสมบูรณ์

2.4.2 กระบวนการทางซอฟต์แวร์ทั่วไป (Generic Software Process)

ส่วนใหญ่ที่นิยมใช้กันอย่างแพร่หลาย มี อยู่ 3 กระบวนที่สำคัญ

2.4.2.1 Waterfall Model แบบจำลองน้ำตก บางครั้งถูกเรียกว่า วงจรแบบฉบับ (Classic Life Cycle) ซึ่งหมายถึง แบบระเบียบวิธีเรียงลำดับเป็นระบบในการพัฒนาซอฟต์แวร์ เริ่มด้วยการกำหนดความต้องการของลูกค้า และก้าวหน้าไปสู่การวางแผน การสร้างแบบจำลอง การสร้างซอฟต์แวร์ ตามด้วยการให้การช่วยเหลือในการใช้งาน ซึ่งแบบจำลองน้ำตก มี 5 ขั้นตอนดังนี้

- Requirement Definition: เป็นการนิยามหรือกำหนดความต้องการของระบบและคุณสมบัติ
- System and Software Design: การออกแบบซอฟต์แวร์และการออกแบบระบบ
- Implementation and Unit testing: การพัฒนาตามที่ได้ออกแบบและการทดสอบระบบ
- Integration and System testing: การนำระบบย่อยมารวมกันและทดสอบระบบทั้งหมด
- Operation and Maintenance: การติดตั้งใช้งานจริงและขั้นตอนการบำรุงรักษา โดยที่สามารถย้อนกลับไปทำขั้นตอนก่อนหน้าเมื่อพบปัญหา



ภาพ 2-3 The software life cycle **กระบวนการออกแบบและพัฒนาระบบแบบ** Waterfall

วิธีการนี้ได้รับการพัฒนาขึ้นมาเป็นโมเดลแรกเริ่มตั้งแต่ที่มีการพัฒนาซอฟต์แวร์ โดยเทคนิคนี้จะทำการแบ่งการดำเนินงานของระบบทั้งหมดเป็นขั้นตอนย่อยๆ ที่จำเป็น เช่น การกำหนดความต้องการ การออกแบบระบบ การพัฒนาและการทดสอบระบบ ซึ่งในแต่ละขั้นตอนจะสามารถย้อนกลับไปทำยังขั้นตอนก่อนหน้าได้เมื่อพบปัญหา จึงมีลักษณะการทำงานเป็นขั้นๆ เหมือนกับน้ำตก จึงเรียกกระบวนการพัฒนาแบบนี้ว่า waterfall ซึ่งทำให้เกิดวงจรชีวิตของซอฟต์แวร์

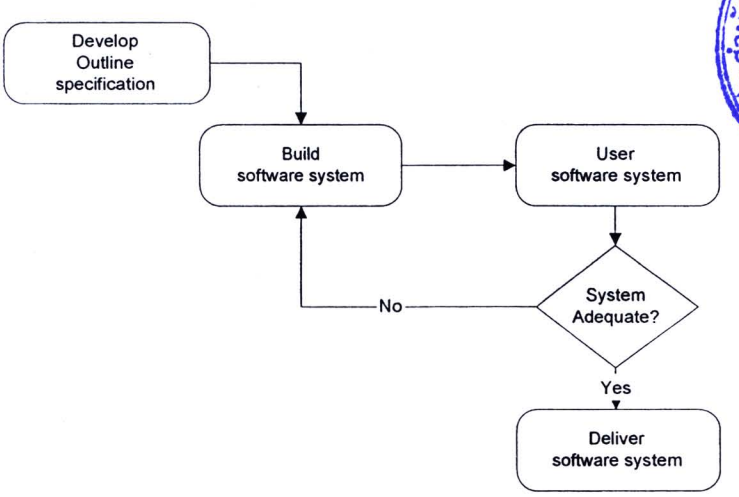
2.4.2.2 การพัฒนาแบบ Evolutionary การพัฒนา Softwareจัดเป็นวิวัฒนาการ โดยมีการปรับเปลี่ยน Requirement Specification (Spec.) ไปเรื่อยๆ และพัฒนาปรับเปลี่ยน Specification เป็นลักษณะของ Version

ขั้นแรกเป็นการกำหนดขอบเขตก่อน จากนั้นค่อยกำหนด Spec. ขึ้นมา และพร้อมที่จะพัฒนาได้ในระยะเวลาอันสั้นและทำการตรวจสอบว่า Version แรกถูกต้องและมีอะไรปรับเปลี่ยนหรือไม่ หลังจากได้ version 1 แล้ว เราก็ทำการเพิ่ม Specification ต่างๆ ลงไปหรือการทำงานในมุมต่างๆที่เราเคยตัดทิ้งไปในตอนแรก เราก็จะรวมเข้าไปและทำการพัฒนาต่อไปอีกได้เป็น Version ถัดไป และทำเป็นกระบวนการซ้ำจนกว่าจะได้ Version ที่สมบูรณ์แล้ว

เทคนิคการออกแบบและพัฒนาซอฟต์แวร์นี้ ตั้งอยู่บนแนวความคิดที่จะพัฒนาระบบซึ่งจะทำงานกับระบบที่ทำงานเร็วที่สุดเท่าที่จะเป็นไปได้ให้เสร็จก่อน หลังจากนั้นจึงจะทดลองใช้และ

ปรับปรุงแก้ไขจนกว่าระบบจะทำงานได้ตรงตามความต้องการ ตัวอย่างที่เห็นได้ชัดคือ ในกระบวนการสร้างระบบ "ปัญญาประดิษฐ์" (Artificial Intelligence: AI) ซึ่งเป็นระบบที่เราไม่สามารถกำหนดรายละเอียดเบื้องต้นทุกอย่างได้ ในการออกแบบระบบงานโดยเทคนิคนี้ผู้ออกแบบระบบจะคำนึงถึงความเหมาะสมของการทำงาน (adequacy) มากกว่าความสมบูรณ์ถูกต้อง (correctness)

ดังนั้นมีผู้โต้แย้งว่าทุกระบบงานก็จะเริ่มต้นจากโครงร่างคร่าวๆ ทั้งนั้น แต่อย่างไรก็ตาม การออกแบบ exploratory programming นี้เหมาะอย่างยิ่งกับการออกแบบระบบที่เราไม่รู้ว่ารายละเอียดควรเป็นอย่างไร



สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

ภาพ 2-4 กระบวนการออกแบบและพัฒนาระบบแบบ Exploratory programming

การออกแบบพัฒนาระบบซอฟต์แวร์โดยใช้เทคนิคนี้ อาจจะเหมาะสมสำหรับระบบงานบางอย่างโดยเฉพาะ แม้ว่าอาจใช้ไม่ได้ในการพัฒนาระบบงานทั่วๆ ไป แต่อย่างไรก็ตามเทคนิคนี้ก็อาจนำมาประยุกต์ใช้ได้ดีในการบำรุงรักษาระบบงานทั่วไปที่ยังไม่มีความชัดเจนว่าจะใช้วิธีการใดในการปรับแต่งแก้ไขปัญหาที่เกิดขึ้น โดยสรุปแล้วเทคนิค Exploratory programming เหมาะที่จะถูกเลือกใช้ใน 3 กรณีหลักดังต่อไปนี้

1. โครงสร้างของระบบงานที่กำลังพัฒนาเป็นไปในลักษณะที่ต้องมีการนำไปทดลองใช้อย่างสม่ำเสมอ การปรับแต่งแก้ไขเพื่อความเหมาะสมจึงถือได้ว่าเป็นการประเมินความก้าวหน้าของโปรแกรมไปตามลำดับ
2. ระบบงานที่แม้จะมีโครงสร้างที่ไม่ชัดเจน แต่ผู้พัฒนาสร้างระบบและผู้แก้ไขระบบภายหลังเป็นบุคคลเดียวกัน แต่อย่างไรก็ตามในกรณีที่เป็นระบบงานใหญ่ และต้อง

ใช้ภายในหน่วยงานเป็นเวลานาน เทคนิคนี้จะไม่เหมาะสมเพราะคนสร้างระบบและคนปรับแต่งแก้ไขระบบในภายหลังมักจะไม่ใช่คนเดียวกัน ซึ่งอาจมีผลทำให้โครงสร้างและรูปแบบของระบบบิดเบือนไปจากจุดเดิมจนเสียรูปได้

3. แม้ว่ายังไม่มีงานวิจัยยืนยันลักษณะของทีมงานที่เหมาะสมสำหรับการออกแบบโดยใช้เทคนิคนี้ แต่มีผู้แนะนำให้กว้างๆ ว่าทีมงานที่จะใช้วิธีการออกแบบชนิดนี้ได้ดีน่าจะเป็นทีมงานขนาดเล็กที่ประกอบด้วยผู้มีทักษะเชี่ยวชาญ และเป็นผู้ที่มีแรงจูงใจในการทำงานสูง

ข้อดี: เกิดจากการที่เราค่อยๆ ทำทีละ Version ทำให้เกิดการขัดเกลาความต้องการได้หลายรอบ ทำให้ความต้องการ (Requirement) มีความผิดพลาดน้อย
ข้อเสีย: บอกความคืบหน้าของงานได้ลำบาก คือ ยากที่จะบอกว่างานดำเนินการไปถึงไหน หรือ การ Check Milestone ได้ค่อนข้างลำบาก

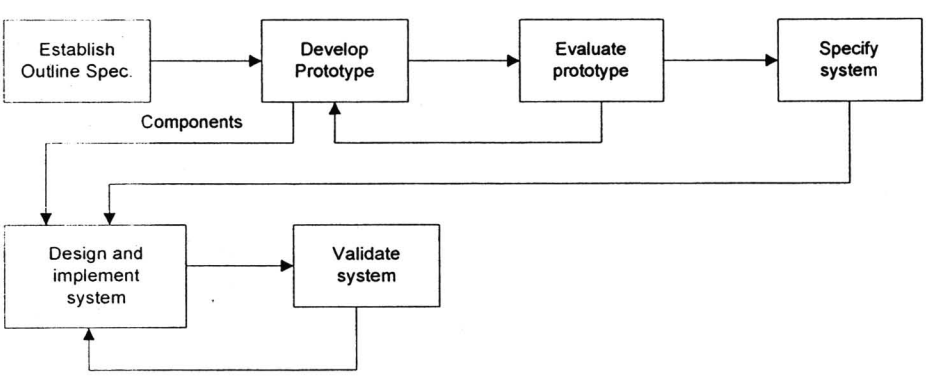
ข้อแตกต่างที่สำคัญระหว่างการออกแบบ Exploratory programming และแบบ Waterfall การออกแบบ Exploratory programming จะเน้นรายละเอียดของระบบตรงขั้นตอนการตรวจสอบและยืนยันความถูกต้อง (V&V) ทั้งนี้เพราะ V แรก (Verification) เป็นการตรวจสอบว่าระบบที่พัฒนาขึ้นมาตรงตามรายละเอียดที่ได้ออกแบบไว้หรือไม่อย่างไร ดังนั้นเมื่อเราไม่สามารถกำหนดรายละเอียดเบื้องต้นได้ ขั้นตอนการตรวจสอบนี้ก็เป็นไปไม่ได้ ส่วนกระบวนการตรวจสอบความถูกต้องของ V ที่สอง (Validation) จึงเป็นการตรวจสอบความถูกต้องเหมาะสมของการทำงานของโปรแกรม มากกว่าเป็นการตรวจสอบความเที่ยงตรงตามโครงสร้างที่ได้ออกแบบไว้

ความถูกต้องเหมาะสมเป็นสิ่งที่วัดได้ยาก และค่อนข้างเป็นการติดสินใจตามผู้ตรวจสอบ (Subjective) เราไม่สามารถรับรองได้ว่าพฤติกรรมของมนุษย์เป็นความถูกต้อง แต่สามารถบอกได้ว่า เราพอใจกับพฤติกรรมที่ทำงานสอดคล้องกับงานที่ต้องการจะให้ทำ

การสร้างแบบจำลอง (Prototyping) เป็นเทคนิคการออกแบบลักษณะ exploratory programming และลักษณะ prototyping มีส่วนที่เหมือนกันคือ เริ่มต้นจากการพัฒนาและนำเสนอระบบงานให้ผู้ใช้ทดลองใช้ แม้ว่าระบบยังไม่สมบูรณ์และยังต้องแก้ไขและปรับเสริมแต่งเพิ่มเติมภายหลัง แต่วิธีการทั้งสองก็มีข้อแตกต่างกันที่สำคัญ คือ exploratory programming เริ่มต้นจากความเข้าใจในรายละเอียดของระบบที่ยังไม่ชัดเจน หลังจากทดลองใช้แล้วจึงมีการเสริม

แต่งระบบงานจนเป็นระบบที่ใช้งานได้ดี (executable system) แม้ว่าบางระบบอาจไม่สามารถระบุรายละเอียดได้เลยก็ตาม ส่วนวิธีการแบบ prototyping เริ่มจากโครงร่างของระบบอย่างคร่าวๆ (outline requirements) เช่นเดียวกัน แต่การทดลองสร้างระบบขึ้นมาใช้งานมีจุดมุ่งหมายที่จะค้นหารายละเอียดของระบบ (system specification) ที่พึงประสงค์อาจเรียกได้ว่าเป็นต้นแบบ (prototype) เมื่อได้นำระบบต้นแบบแรกมาทดลองประเมิน และปรับแต่เพิ่มเติมเป็นระบบต้นแบบในลำดับต่อๆ มาตามลำดับจนเห็นว่าใช้การได้แล้ว (executable) อาจทิ้งระบบต้นแบบสุดท้ายนั้นแล้วสร้างระบบงานจริงใหม่อีกครั้งจากต้นแบบที่ได้รับผ่านมา เพื่อให้ได้ระบบงานที่มีคุณภาพตรงตามความต้องการที่แท้จริงของผู้ใช้ระบบ

กระบวนการพัฒนาซอฟต์แวร์เป็นการขยายขั้นตอนการดำเนินการแบบต้นแบบ (Prototyping) โดยมีจุดมุ่งหมายที่จะลดค่าใช้จ่ายในการพัฒนาระบบงาน รูปแบบของระบบต้นแบบตัวอย่างจะมีลักษณะแบบใช้แล้วทิ้ง (throw-away) หลังจากได้ทดลองใช้ศึกษาคุณลักษณะของระบบที่ต้องการแล้วแสดงขั้นตอนการดำเนินงานตามลำดับคือ การพัฒนาระบบตัวอย่าง จะถูกพัฒนามาจากโครงร่างของระบบอย่างง่าย (outline specification) เมื่อระบบตัวอย่างดังกล่าวถูกนำมาศึกษา ทดลองใช้และประเมินผล จนผู้ใช้รู้สึกพอใจในขั้นตอนและวิธีการทำงานแล้ว กระบวนการพัฒนาซอฟต์แวร์ตามปกติก็จะเริ่มต้น โดยการนำเอารายละเอียดของระบบ (system specification) และโครงสร้างหลักของระบบต้นแบบตัวอย่าง (prototype components) มาประกอบการดำเนินการสร้างระบบงานที่ต้องการขึ้นมา และระบบงานใหม่ดังกล่าวนี้จะถูกทดสอบความถูกต้องสมบูรณ์ (validate) จนกว่าจะเป็นที่ยอมรับได้



ภาพ 2-5 ขั้นตอนดำเนินการของกระบวนการต้นแบบ

ในบางกรณีมีการนำเอาระบบงานต้นแบบ มาจัดเข้าเป็นส่วนหนึ่งของระบบที่สร้างขึ้นมาโดยไม่ต้องดำเนินการจัดสร้างใหม่ ซึ่งเป็นการลดขั้นตอนการพัฒนาสร้างระบบใหม่จาก

ระบบต้นแบบ นั่นคือไม่ต้องมีการสร้างส่วนของระบบงานที่เหมือนกับระบบตัวอย่างต้นแบบที่ได้ทดลองแล้วซ้ำใหม่อีกครั้ง แต่การดำเนินการดังกล่าวนี้อาจมีปัญหาและข้อควรระวังอยู่บ้าง คือ

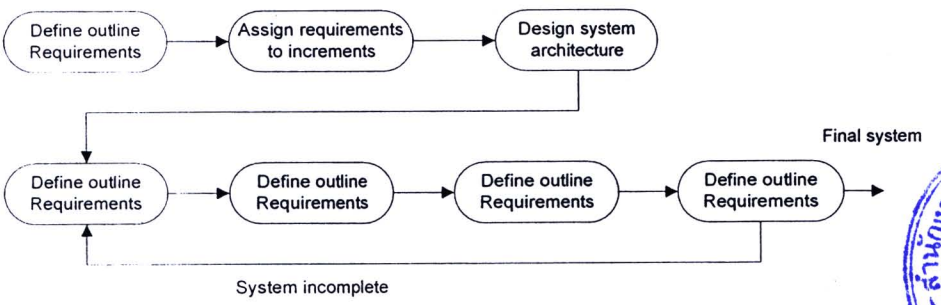
1. ลักษณะสำคัญบางอย่างของระบบ อาจถูกละเว้นไปในขณะทำเป็นระบบงานต้นแบบเพื่อทำให้การทดลองสะดวกและทำงานได้
2. อาจมีปัญหาทางกฎหมายในเรื่องของการนำเอาระบบต้นแบบไปใช้งานในที่ทำงานจริงและยังคงเป็นปัญหาในเรื่องของลิขสิทธิ์
3. ผู้ใช้ระบบอาจใช้ระบบต้นแบบไม่เหมือนกับตอนที่ระบบงานถูกพัฒนาแล้วและใช้งานจริง ตัวอย่างเช่น ในขณะที่เป็นระบบงานต้นแบบช่วงเวลาในการทำงานอาจช้าและผู้ใช้ระบบปรับตัวจนเคยชินกับระบบต้นแบบแล้ว เมื่อพัฒนาเป็นระบบงานจริงเราต้องการทำงานที่เร็วขึ้น ผู้ใช้ระบบอาจต้องเผชิญหน้ากับความไม่คุ้นเคยและทำให้ระบบไม่ประสบผลสำเร็จเท่าที่ควร

ด้วยสาเหตุดังกล่าวข้างต้น ในการพัฒนาระบบงานใหญ่ๆ จึงควรมีการสร้างและพัฒนาระบบงานจริงขึ้นมาใหม่จากระบบงานต้นแบบที่ได้ทดลองใช้ โดยไม่ควรนำเอาระบบต้นแบบมาพัฒนาใช้งานเลย (เพื่อลดค่าใช้จ่าย) ทั้งนี้ด้วยเหตุผลดังนี้

1. ลักษณะการทำงานบางอย่าง เช่น พฤติกรรมของระบบและผู้ใช้ระบบ ความมั่นคงปลอดภัยของระบบ ความน่าเชื่อถือของข้อมูลในระบบอาจถูกมองข้ามไปอย่างจงใจในระบบงานต้นแบบเมื่อเวลาทดลองใช้ แต่ในระบบงานจริงจะละเว้นลักษณะของระบบดังกล่าวไม่ได้ และหลายๆ กรณีเราไม่สามารถเพิ่มลักษณะดังกล่าวเข้าในระบบต้นแบบ
2. ในขณะพัฒนาระบบงานต้นแบบอาจมีการเปลี่ยนแปลงรูปแบบอยู่ตลอดเวลาเพื่อให้ตรงตามความต้องการของผู้ใช้ระบบ การเปลี่ยนแปลงดังกล่าวอาจเกิดขึ้นในสภาพที่ไร้การควบคุม มีการเปลี่ยนรูปแบบไปเรื่อยๆ จนกว่าจะพอใจ และเมื่อนำเอาระบบต้นแบบที่พอใจแล้วมาใช้งานจริง ก็อาจจะประสบปัญหาการบำรุงดูแลรักษาระบบในภายหลังได้
3. การเปลี่ยนแปลงระบบต้นแบบบ่อยครั้ง อาจเป็นการลดโครงสร้างที่จำเป็นต้องมีในระบบงานจริง เพื่อทำให้สะดวกและง่ายมักจะทดลองคุณลักษณะนั้นๆ ของระบบการเปลี่ยนแปลงมากเกินไปอาจทำให้เราไม่สามารถบำรุงรักษาระบบต้นแบบนั้นในภายหลัง

การใช้ prototype ทำให้เกิดประโยชน์อย่างยิ่งในการนำไปทำความเข้าใจเกี่ยวกับทิศทางที่จะออกแบบระบบรวมและ interface ของระบบ โดยมีการปรับเปลี่ยนไปตามความต้องการให้มีประสิทธิภาพ (Nauman and Jenkins, 1982)

2.4.2.3 การพัฒนาแบบเพิ่มขึ้น (Incremental Development) การพัฒนาแบบเพื่อขึ้น (Incremental Development) จะมีลักษณะที่สอดคล้องกับ Model ต่างๆ ที่กล่าวไว้เป็นส่วนใหญ่ โดยแนวคิดของการพัฒนาแบบเพื่อขึ้น คือ แทนที่เราจะทำ Software ให้จบทีเดียว เราก็จะทำเป็นทีละส่วนย่อยๆ แล้วนำไปทำการทดสอบหรือไม่ก็ส่งงานไปเหมือนกับเป็น Milestone ย่อยๆ ไป



ภาพ 2-6 การพัฒนาแบบ Incremental

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

จากรูปที่ 5 ในขั้นตอนแรกจะเป็นการกำหนด Requirement คร่าวๆ แล้วขั้นต่อมาจะพิจารณาหาส่วนที่จะเพิ่ม requirement เข้าไป หลังจากนั้นก็จะเป็นขั้นตอนการออกแบบให้เห็นภาพรวมของระบบ แล้วจึงมาออกแบบงานชิ้นย่อยชิ้นแรกโดยละเอียดและตรวจสอบ แล้วก็วนกลับมาทำ incremental ที่ 2 แล้วตรวจสอบ แล้วนำมารวมกับ increment ที่ 1 แล้วตรวจสอบภาพรวมว่าทำงานได้เป็นอย่างไร แล้วจึงวนกลับไปทำ increment ที่ 3 และ increment ที่เหลือซ้ำจนจบทุก increment ที่กำหนดไว้จนได้ออกมาเป็น Product สุดท้าย

ข้อดี:

1. increment แรกๆ จะโดนตรวจสอบหลายรอบ
2. ถ้า requirement เปลี่ยนแปลงก็สามารถนำรวมเข้าไปได้เรื่อยๆ

2.5 โครงการ Telemedicine คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล

การแพทย์ทางไกล (Telemedicine) เป็นการนำเอาความก้าวหน้าด้านการสื่อสารโทรคมนาคม มาประยุกต์ใช้กับงานทางการแพทย์ โดยการส่งสัญญาณผ่านสื่อซึ่งอาจจะเป็นสัญญาณดาวเทียม (Satellite) หรือใยแก้วนำแสง (Fiber optic) แล้วแต่กรณีควบคู่ไปกับเครือข่ายคอมพิวเตอร์ แพทย์ต้นทางและปลายทางสามารถติดต่อกันด้วยภาพเคลื่อนไหวและเสียง ทำให้สามารถแลกเปลี่ยนข้อมูลคนไข้ระหว่างกันและกัน ทั้งทางด้านภาพเช่นฟิล์มเอกซเรย์และเสียงสัญญาณจากเครื่องมือแพทย์เช่นการเต้นของหัวใจลื่นหัวใจ (ECG) พร้อมๆ กับแลกเปลี่ยนประสบการณ์ และปรึกษาหารือกันเสมือนกับแพทย์ต้นทางแพทย์ปลายทางและคนไข้อยู่ในห้องเดียวกัน

นอกจากนั้นการแพทย์ทางไกลยังนำมาใช้ในการประชุมปรึกษาหารือกันทางไกล (Video Conferencing) การศึกษาต่อเนื่องทางไกล (Distance Learning) และการเชื่อมโยงเครือข่ายคอมพิวเตอร์ระหว่างส่วนกลางและส่วนภูมิภาคอีกด้วย กระทรวงสาธารณสุขได้ตระหนักถึงปัญหาขาดแคลนแพทย์ผู้เชี่ยวชาญในท้องที่ชนบทห่างไกล ซึ่งทำให้ประชาชนที่เจ็บป่วยมุ่งเข้ามารับการรักษาพยาบาลในโรงพยาบาลใหญ่ๆ ในเมือง อันทำให้เป็นภาระหนักของโรงพยาบาลเหล่านั้น และเป็นภาระทางด้าน ค่าใช้จ่ายและเวลาของประชาชนที่ต้องเดินทางเข้ามารับบริการ กระทรวงสาธารณสุขจึงได้ริเริ่มดำเนินงานโครงแพทย์ทางไกลผ่านดาวเทียมขึ้น โดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพในการให้บริการรักษาพยาบาลของ โรงพยาบาลชุมชนที่มีแพทย์ประจำอยู่จำกัด และเพื่อพัฒนาบุคลากรทางการแพทย์และสาธารณสุขในท้องถิ่นห่างไกล โดยจัดการเรียนการสอนทางไกล เพื่อให้บุคลากรเหล่านั้นได้มีโอกาสศึกษาต่อเนื่องเพิ่มเติมโดยไม่ต้องลาเรียนต่อและได้อยู่ปฏิบัติงานที่หน่วยงานต่อไปได้

หมายความของการแพทย์ทางไกล ไม่จำเป็นจะต้องผ่านดาวเทียม ผ่านสายโทรศัพท์หรือเคเบิลอย่างอื่น รวมทั้งอินเตอร์เนทด้วยก็ได้ เช่นคนไข้อยู่แม่ฮ่องสอนแต่แพทย์ที่จะให้คำปรึกษาอยู่เชียงใหม่ ข้อมูลต่างๆ เช่นความดันเลือด ชีพจร คลื่นหัวใจ รวมทั้งเอกซ์เรย์ ผลการตรวจเลือด ฯลฯ ตลอดจนภาพผู้ป่วย ดูที่เชียงใหม่ได้ทันตา หรือที่เรียกว่า real time ไม่เสียเวลาส่งข้อมูลเลย

จุดอ่อนของการแพทย์ทางไกลอยู่ในด้านการตรวจ เพราะการตรวจร่างกายคนไข้นั้นใช้การดู คลำ เคาะ กับฟัง การดูกับการฟัง (เช่นดูท่าทางคนไข้ การหายใจ รวมทั้งดูแผล ดูตา หรือฟังเสียงหัวใจ ฟังเสียงหายใจ) นั้นพอทำทางไกลได้ถ้าอุปกรณ์ดีพอ แต่การคลำกับการเคาะนั้นต้องอาศัยจากปากคำของแพทย์หรือผู้ช่วยแพทย์ที่ยืนอยู่ข้างๆคนไข้ แพทย์ที่อยู่ที่เชียงใหม่ไม่มีทางรู้ว่าก้อน

ในท้องของผู้ป่วยที่แม่ฮ่องสอนนั้นมีลักษณะอย่างไร หรือกดท้องตรงไหนเจ็บ กล้ามเนื้อหน้าท้องเกร็งหรือไม่ เป็นต้น

2.6 งานวิจัยที่เกี่ยวข้อง

ในการดำเนินการเพื่อพัฒนาระบบวินิจฉัยโรคทางไกลโดยใช้ระบบผู้เชี่ยวชาญช่วยในการตั้งคำถาม ผู้ศึกษาได้ทำการศึกษาเอกสารที่เกี่ยวข้องเพื่อเป็นพื้นฐานและแนวทางสำหรับการดำเนินงาน โดยสามารถสรุปได้ดังนี้

ภัสชรัชย์ ชาวบ้านกร่าง (2546) ได้ศึกษาระบบผู้เชี่ยวชาญด้านโรคพืช โดยโครงงานระบบผู้เชี่ยวชาญด้านโรคพืชถูกพัฒนาขึ้น เนื่องจากผู้จัดทำโครงงานได้เล็งเห็นถึงปัญหาและความสำคัญของโรคพืช ที่นับเป็นปัญหาหลักของเกษตรกรและผู้ที่สนใจในการเพราะปลูกตลอดจนนักวิชาการด้านโรคพืชที่ต้องการเผชิญอย่างหลีกเลี่ยงไม่ได้ ระบบดังกล่าวถูกออกแบบและพัฒนาขึ้น เพื่อช่วยเหลือ และตอบสนองความต้องการ ในการแก้ปัญหาข้างต้น โดยระบบของผู้จัดทำโครงงานมีความสามารถในการ เพิ่มเติมข้อมูล แก้ไขข้อมูล หรือลบข้อมูลโรคพืช และสามารถค้นหาข้อมูลโรคพืช หรือโดยการเปรียบเทียบอาการของโรคได้ ในทฤษฎีที่เกี่ยวข้อง ผู้ศึกษาได้อ้างอิงข้อมูลจากเอกสารอื่นมากมาย ซึ่งนำมาใช้กับการทำโครงงานนี้ อย่างเช่น ทฤษฎีของ อินเตอร์เน็ต ASP JavaScript VBScript ภาษา SQL โครงสร้างข้อมูล ปัญญาประดิษฐ์กับฐานข้อมูลความรู้ และทฤษฎีที่เกี่ยวกับ โรคพืช เป็นต้น

วิธีดำเนินการจัดทำโครงงาน มีการทำงานที่เป็นลำดับขั้น โดยจะทำการศึกษาและเก็บรวบรวมตัวอย่างข้อมูลโรคพืชที่สามารถอ้างอิงที่มาได้ อย่างเช่น เอกสารทางวิชาการ และสื่ออิเล็กทรอนิกส์จาก Web Site ต่างๆ มาจำแนกตามชนิดของพืชเป็น 4 กลุ่ม ได้แก่ กลุ่มพืชผัก กลุ่มผลไม้ กลุ่มพืชไร่ และกลุ่มไม้ดอก-ไม้ประดับ จากนั้นทำการศึกษาโครงสร้าง และส่วนประกอบต่างๆ ของข้อมูลโรคพืช ซึ่งจะประกอบไปด้วยชื่อพืช ชื่อโรค ชื่อสามัญ สาเหตุ อาการ วิธีป้องกันแก้ไข และแหล่งข้อมูลอ้างอิง ส่วนของโปรแกรมผู้ศึกษาได้ทำการศึกษาเครื่องมือ วิธีการ เทคนิคตลอดจน อัลกอริทึม ในการวิเคราะห์ระบบจะทำการสร้างฐานข้อมูลโรคพืชที่จัดเก็บชื่อพืช ชื่อโรค ชื่อโรคสามัญ สาเหตุ อาการ วิธีป้องกันแก้ไข และแหล่งข้อมูลอ้างอิง การออกแบบระบบจะมีการแบ่งออกเป็น 2 ส่วน คือ การจัดเตรียมข้อมูล โดยจะทำการรวบรวมข้อมูลโรคพืชซึ่งประกอบด้วย ชื่อโรค ชื่อโรคสามัญ สาเหตุ อาการ วิธีป้องกันแก้ไข และแหล่งข้อมูลอ้างอิง จะมีกระบวนการอยู่ 4 อย่าง คือ เพิ่มเติมข้อมูลโรคพืช แก้ไขชื่อพืช แก้ไขข้อมูลโรคพืช และลบข้อมูลโรคพืช ในการออกแบบระบบลำดับถัดไป คือ การค้นหาข้อมูล โดยจะแบ่งการค้นหาเป็น 2 รูปแบบ อันดับแรก

คือ ทำการค้นหาโดยชื่อโรค จะเป็นการค้นหาโดยป้อนชื่อพืช จากนั้นจะแสดงชื่อโรคที่เกิดกับพืชดังกล่าว อันดับที่สอง คือ ค้นหาโดยอาการ ผู้ใช้ป้อนชื่อพืชที่ต้องการ โปรแกรมจะทำการเปรียบเทียบอาการที่ผู้ใช้เลือกกับอาการจากฐานข้อมูลในระบบ ทำการประมวลผล และแสดงชื่อโรคซึ่งสามารถดูการไหลของข้อมูลได้จาก DFD จากเอกสารต้นฉบับ ในการพัฒนาโปรแกรมจะมีในส่วนของการจัดเตรียมข้อมูล และการค้นหาข้อมูล ซึ่งจะมีการแสดงใน ผังระบบงาน (System Flowchart) และขั้นตอนสุดท้ายคือทำการทดสอบระบบ ซึ่งจะมีการทดสอบอยู่ 2 อย่าง อันดับแรก คือ การทดสอบผลการดำเนินงานของโปรแกรมโดยผู้จัดทำโครงงาน ซึ่งจะทำการทดสอบโดยวิธีการแบบ Black Box Testing ซึ่งเป็นการทดสอบโดยให้ความสนใจกับข้อมูลที่นำเข้า และผลลัพธ์ที่ได้รับจากข้อมูลที่นำเข้า อันดับที่สองคือ การทดสอบประสิทธิภาพของโปรแกรมโดยผู้เชี่ยวชาญ ซึ่งจะให้เครื่องมือในการประเมินประสิทธิภาพของโปรแกรม โดยการใช้วิธีการแบบ Black Box Testing

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมระบบผู้เชี่ยวชาญด้านโรคพืช ได้แก่ Active Server Page (ASP) VBScript JavaScript และระบบจัดการฐานข้อมูล Microsoft Access ซึ่งมีผลการดำเนินงาน และประสิทธิภาพของโปรแกรมอยู่ในเกณฑ์ดีมาก

สรุปผลการทำงานของระบบผู้เชี่ยวชาญด้านโรคพืชได้พัฒนาขึ้น เพื่อช่วยแก้ปัญหาโรคพืช และเพื่อเผยแพร่ข้อมูลโรคพืช พร้อมวิธีการป้องกันแก้ไขสำหรับเกษตรกร ผู้เชี่ยวชาญด้านโรคพืช ตลอดจนบุคคลทั่วไปที่สนใจในการเพราะปลูก และกำลังประสบปัญหาด้านโรคพืช ซึ่งโปรแกรมนี้สามารถ เพิ่มเติม แก้ไข และลบข้อมูลโรคพืชได้ สามารถค้นหาข้อมูลโรคพืชโดยชื่อโรค สามารถค้นหาข้อมูลโรคพืช โดยเปรียบเทียบอาการของโรค ดับอาการจากฐานข้อมูลในระบบ

สุวัฒน์ ธาดาวุธ (2547) ซึ่งในงานวิจัยนี้ผู้วิจัยได้ศึกษาเกี่ยวกับโรคโลหิตจางธาลัสซีเมีย เพื่อนำมาใช้พัฒนาระบบวินิจฉัยโรคธาลัสซีเมียจากผลการตรวจค่าฮีโมโกลบิน และผู้วิจัยได้อ้างอิงถึงงานวิจัยที่เกี่ยวข้องหลายงานวิจัย ซึ่งงานวิจัยที่เกี่ยวข้องกับระบบผู้เชี่ยวชาญมีบทบาทค่อนข้างมากในการวินิจฉัยทางการแพทย์ ในงานเหล่านี้ก็มีอิทธิพลอย่างมากต่อการพัฒนาของระบบผู้เชี่ยวชาญในขั้นต่อมา อย่างเช่น มัยซิน (Mycin) ซึ่งเป็นระบบผู้เชี่ยวชาญที่เก่าแก่ที่สุด และเป็นต้นแบบในการพัฒนาระบบผู้เชี่ยวชาญทางการแพทย์ต่อมาอีกหลายระบบ วีเอ็ม (VM: The Ventialtor Manager) เป็นระบบผู้เชี่ยวชาญที่ใช้ในการเฝ้าสังเกตผู้ป่วยหลังจากการผ่าตัดหัวใจ ระบบนี้พัฒนาขึ้นโดยมีอิทธิพลมาจากระบบมัยซิน ไกดอน (Guidon) ระบบผู้เชี่ยวชาญอีกระบบหนึ่งซึ่งมีความสัมพันธ์กับมัยซิน คือ ไกดอน ซึ่งเป็นระบบผู้เชี่ยวชาญทางด้านการสอนพัฒนาโดยนักวิจัยของมหาวิทยาลัยสแตนฟอร์ด ไกดอนเป็นระบบที่พัฒนาส่วนการอธิบายของมัยซินต่อจาก

เดิมเพื่อหวังว่าจะสามารถใช้สอนนักศึกษาแพทย์ในการวินิจฉัยการติดเชื้อในโลหิต ระบบสนับสนุนการตัดสินใจสำหรับการแพทย์ทางไกลโดยอาศัยความชำนาญหลายทาง ซึ่งเป็นงานวิจัยที่มีจุดมุ่งหมายในการจัดการความชำนาญที่มาจากแพทย์ผู้เชี่ยวชาญจากหลากหลายประเทศ เพื่อที่จะนำมาสร้างเป็นระบบสนับสนุนการวินิจฉัยทางการแพทย์ ดีเอ็กเพลน (DXplain) เป็นเครื่องมือที่ช่วยสนับสนุนการตัดสินใจทางการแพทย์อีกตัวหนึ่งซึ่งมีฐานข้อมูลของอาการกว่า 5,000 รายการ ที่สัมพันธ์กับโรคกว่า 2,000 ชนิด และ อิลียด (Iliad) เป็นเครื่องมือการวินิจฉัยทางการแพทย์ที่ใช้สอนในวิทยาลัยแพทย์ และใช้ปรึกษาให้คำแนะนำในโรงพยาบาล ในส่วนทฤษฎีที่เกี่ยวข้องได้อ้างอินถึง ระบบผู้เชี่ยวชาญซึ่งเป็นทฤษฎีหลักที่ใช้ในการทำวิจัย ซึ่งจะพูดถึงโครงสร้างพื้นฐานของระบบผู้เชียวชาญ อันดับแรกคือ ฐานความรู้ (Knowledge Base) ซึ่งจะเป็นส่วนที่ใช้เก็บความรู้ทุกประเภท ทั้งที่ได้จากตำราหรือความรู้ที่ได้จากประสบการณ์ อันดับที่สอบ เครื่องอนุมาน (Inference Engine) ซึ่งเปรียบได้กับขั้นตอนวิธี (Algorithm) เป็นส่วนที่ควบคุมการใช้ความรู้ อันดับที่สาม ส่วนที่ได้มาซึ่งความรู้ ส่วนอธิบาย และส่วนต่อประสานผู้ใช้ และมีทฤษฎีของโรคโลหิตจางธารัสซีเมีย

ในส่วนของวิธีการดำเนินงาน หรือกระบวนการในการพัฒนาระบบผู้เชี่ยวชาญ โดยจะมีการนำปัญหามาวิเคราะห์ การเลือกเครื่องมือ จะพิจารณาความสามารถของการแสดงความรู้ เครื่องอนุมาน การติดต่อกับผู้พัฒนาระบบ และการติดต่อกับผู้ใช้ การถอดความรู้ การสร้างต้นแบบ การทดสอบ การปรับปรุง และการได้มาซึ่งความรู้ เป็นการดึงความรู้ในการสกัดและเรียบเรียงความรู้จากผู้เชี่ยวชาญเพื่อนำมาใช้ในระบบปัญญาประดิษฐ์ อย่างเช่น การสัมภาษณ์ ซึ่งจะมีการสัมภาษณ์แบบไร้โครงสร้าง เป็นการสัมภาษณ์ที่วิศวกรความรู้จะให้ผู้เชี่ยวชาญแนะนำเกี่ยวกับแนวคิด การสัมภาษณ์แบบปลายเปิด การสังเกตการณ์ทำงานของผู้เชี่ยวชาญ การออกแบบสอบถาม และการรายงานผู้เชี่ยวชาญ

ในการออกแบบฐานความรู้ จะทำการเก็บรวบรวมและตั้งสมมติฐานของเกณฑ์ที่ใช้ในการวินิจฉัย ซึ่งจะรวบรวมและจำแนกรายการของคำวินิจฉัยและเกณฑ์ของคำวินิจฉัยแต่ละรายการจากแพทย์ผู้เชี่ยวชาญ สืบเนื่องจากการที่แพทย์ผู้เชี่ยวชาญไม่สามารถถ่ายทอดความรู้ ความชำนาญการในการวินิจฉัยโรคธาลัสซีเมียให้ผู้อื่นเข้าใจได้ง่าย ผู้วิจัยได้ศึกษาจากตำราการแพทย์และสอบถามจากผู้เชี่ยวชาญทำให้ได้เกณฑ์ของคำวินิจฉัยแต่ละรายการ

ปนัดดา สรรพงชัยพงษ์ (2544) ได้ทำการศึกษาพัฒนาระบบผู้เชี่ยวชาญในการตรวจวินิจฉัยและรักษาโรคตาแดง เพื่อให้คำปรึกษาในการตรวจ วินิจฉัยโรค และรักษาโรคตาแดง ได้

เช่นเดียวกับผู้เชี่ยวชาญด้านจักษุวิทยา ระบบนี้ได้จัดเก็บความรู้แบบกฎแล้วนำมาสร้างเป็นระบบผู้เชี่ยวชาญที่ใช้เทคนิคการอนุมานทั้งแบบเดินหน้าและย้อนกลับซึ่งพัฒนาด้วยโปรแกรมวิชวลเบสิก ที่สามารถวิเคราะห์เพื่อวินิจฉัยและให้คำแนะนำในการรักษาโรคตาแดงได้ วิธีการใช้ระบบนี้ทำได้โดย ผู้ใชระบบจะซักถามประวัติหรือตรวจผู้ป่วยที่มีกาอการแสดงของตาแดง เพื่อให้ได้ข้อมูลของผู้ป่วยเกี่ยวกับอาการ อาการแสดง ผลการตรวจตา และผลการตรวจทางห้องปฏิบัติการ แล้วระบบจะนำข้อมูลนั้นไปวิเคราะห์เพื่อแสดงรายชื่อโรคที่เกี่ยวข้องสุดท้ายจึงจะแสดงรายละเอียดของโรคและให้คำแนะนำในการรักษาโรคนั้นๆ ในบางกรณีที่ระบบไม่สามารถระบบโรคได้ ระบบก็จะแสดงคำแนะนำให้ไปปรึกษาผู้เชี่ยวชาญทางด้านจักษุวิทยา หรือให้ย้อนกลับไปเริ่มทำการซักประวัติและผู้ป่วยใหม่อีกครั้งเพื่อเก็บข้อมูลเพิ่มเติม

วัชรชัย วิริยะสุทธิวงศ์ (2542) ได้ทำการออกแบบและพัฒนาระบบผู้เชี่ยวชาญต้นแบบสำหรับวินิจฉัยโรคทางคลินิก โดยนทฤษฎีฟัซซี่เซตมาประยุกต์ใช้ในกระบวนการอนุมานและการแทนความรู้ การออกแบบได้จัดวางโครงสร้างและการทำงานของกลไกการวินิจฉัยโรค ให้มีลักษณะใกล้เคียงกับกระบวนการตัดสินใจของแพทย์ โดยแบ่งการวินิจฉัยออกเป็น 3 ส่วน ส่วนแรกเป็นการวินิจฉัยเพื่อหากลุ่มโรค ซึ่งใช้กลไกการอนุมานแบบฟัซซี่ และการแทนความรู้ในรูปกฎฟัซซี่ การวินิฉัยสองส่วนหลังเป็นการวินิจฉัยเพื่อหาโรคและกำหนดแนวทางการรักษา ซึ่งใช้กลไกการอนุมานเป็นลูกโซ่แบบไปข้างหน้า และการแทนความรู้ในรูปกฎโพรดักชัน การติดต่อกับผู้ใช้มีลักษณะเป็นเมนูให้เลือกแบบถามตอบกับผู้ใช้ และใช้การวิเคราะห์กระจายคำร่วมกับชุดของคำที่เตรียมไว้

สถาปัตยกรรมของระบบผู้เชี่ยวชาญประกอบด้วย ส่วนวินิจฉัยโรค ฐานความรู้ส่วนเชื่อมโยงกับผู้ใช้ ส่วนเพิ่มเติมความรู้ ส่วนให้คำอธิบาย ส่วนสอบถามความรู้ และหน่วยความจำใช้งานระบบได้รับการพัฒนาขึ้นบนไมโครคอมพิวเตอร์ โดยใช้ภาษา Common LISP การติดต่อกับผู้ใช้และวินิจฉัยโรคทำได้ 2 ภาษาคือ ภาษาไทย และอังกฤษ ฐานความรู้ประกอบด้วยความรู้จำนวน 1122 กฎ และสามารถวินิจฉัยโรคได้ประมาณ 120

จำลอง แสนคำ (2548) ได้ทำการวิจัยเกี่ยวกับระบบตรวจวัดคลื่นไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคลสำหรับการแพทย์ทางไกล วิทยานิพนธ์นี้นำเสนอการวิจัยและพัฒนาเครื่องมือวัดและส่งสัญญาณคลื่นไฟฟ้าหัวใจ 12 สายสัญญาณ ด้วยคอมพิวเตอร์ส่วนบุคคลในลักษณะเวลาจริงไปบนเครือข่ายโทรศัพท์พื้นฐานที่มีใช้แพร่หลายทั่วไปและค่าบริการไม่แพง เพื่อช่วยเพิ่มความสามารถของแพทย์ที่เชี่ยวชาญเฉพาะทางในการให้บริการทางการแพทย์แก่

โรงพยาบาลในชนบท และเพื่อพัฒนาอุปกรณ์เครื่องมือแพทย์ขึ้นใช้ภายในประเทศ ระบบต้นแบบประกอบด้วย 1) เครื่องปรับปรุงสัญญาณคลื่นไฟฟ้าหัวใจ 8 ช่องสัญญาณ 1 เครื่อง (8 channels ECG Signal Conditioning, ESC) 2) คอมพิวเตอร์ส่วนบุคคล 2 เครื่อง 3) หน่วยรับสัญญาณคลื่นไฟฟ้าหัวใจ (Data Acquisition Card, DAC) 1 อัน 4) โมเด็ม 2 เครื่อง และ 5) โปรแกรม TeleECG

ในการทดสอบระบบที่พัฒนาขึ้นเบื้องต้นระหว่างโรงพยาบาลเขาสวนกวาง อำเภอเขาสวนกวาง จังหวัดขอนแก่น และศูนย์หัวใจสิริกิติ์ภาคตะวันออกเฉียงเหนือ จังหวัดขอนแก่น ที่อยู่ห่างกันเป็นระยะทางประมาณ 50 กิโลเมตร เพื่อหาค่าความผิดพลาดและความเพี้ยนในการส่งสัญญาณคลื่นไฟฟ้าหัวใจ โดยการนำข้อมูลของสัญญาณคลื่นไฟฟ้าหัวใจจากทางด้านรับและด้านส่ง มาเปรียบเทียบหาความแตกต่างของข้อมูล พบว่าไม่มีความแตกต่างของชุดข้อมูลที่ใช้ทดสอบ (Percent Root-Mean-Square Difference, PRD = 0 %) เมื่ออัตราการส่งข้อมูลของสายโทรศัพท์มากกว่า 26.4 kbps. และค่าเวลาหน่วงมีน้อยมาก และไม่มีผลเสียต่อการวินิจฉัยโรคของแพทย์

Carlisle Housing Association และ Carlisle and District Primary Care (2549) โครงการ telemedicine ประเทศอังกฤษ ได้นำระบบการรักษาแบบทางไกลของแพทย์มาใช้ ช่วยลดเวลาการเข้ารักษาตัวของคนไข้ในโรงพยาบาลลงได้มาก โดยแพทย์สามารถมอนิเตอร์อาการของคนไข้ได้จากที่บ้าน ไม่ต้องอยู่ดูอาการในโรงพยาบาลนาน ๆ เหมือนในอดีต

แพทย์อังกฤษใช้การรักษาดังกล่าวกับผู้ป่วยที่เป็นโรคเรื้อรัง ซึ่งช่วยให้ผู้ป่วยใช้เวลารักษาตัวอยู่ที่โรงพยาบาลน้อยลงจากเดิม 10 วัน เหลือเพียง 5.5 วันเท่านั้น โดยผู้ป่วยจะได้รับเครื่อง telemedicine monitors ซึ่งก็คืออุปกรณ์ที่สามารถวัดระดับอุณหภูมิในร่างกาย อัตราการเต้นของหัวใจ ระบบการหายใจ และความดันโลหิตได้ ซึ่งคนไข้จะต้องใช้วัดตามที่แพทย์สั่ง จากนั้น ระบบก็จะส่งผลการวัดทั้งหมดกลับไปยังเซิร์ฟเวอร์ของโรงพยาบาล (ผ่านสายโทรศัพท์พื้นฐาน) และเก็บบันทึกไว้เป็นข้อมูลของคนไข้ สำหรับให้หมอและพยาบาลได้ใช้ประโยชน์ในการรักษาครั้งต่อไป หรือใช้ในกรณีฉุกเฉิน

ระบบได้พัฒนาขึ้นมาสำหรับรองรับให้คนไข้เป็นคนวัดค่าต่าง ๆ ด้วยตัวเองได้ ในกรณีที่อยู่คนเดียว และสำหรับคนไข้ที่มีอายุระหว่าง 35 - 82 ปี พบว่า 94 เปอร์เซ็นต์ให้การยอมรับระบบดังกล่าว ผู้เชี่ยวชาญของสถานพยาบาลได้นำระบบ telemedicine มาใช้ และพบว่ามันสามารถใช้ร่วมไปกับการรักษาผู้ป่วยที่ป่วยเป็นโรค COPD (Chronic Obstructive Pulmonary Disease)ได้ และในสถานพยาบาลที่นำไปใช้ แพทย์สามารถให้คนไข้กลับไปรักษาตัวที่บ้านได้เร็วขึ้น