ปัญจราศี ปุณณชัยยะ
ดร.นิตยา วงศ์ภินันท์วัฒนา
ผู้ช่วยศาสตราจารย์ประจำภาควิชาระบบสารสนเทศเพื่อการจัดการ
คณะพาณิชยศาสตร์และการบัญชี มหาวิทยาลัยธรรมศาสตร์

การสำรวจความต้องการด้านเนื้อหาของวิชาการวิเคราะห์และออกแบบระบบสารสนเทศ*

(A Survey on Educational Needs of System Analysis and Design)

[บทคัดย่อ]

งานวิจัยนี้เป็นการวิจัยเชิงสำรวจ (Survey) เพื่อสำรวจความต้องการด้านเนื้อหาวิชาการวิเคราะห์และออกแบบระบบสารสนเทศเพื่อการจัดการ ในการจัดทำวิจัยนี้คณะผู้วิจัยได้พัฒนาแบบสอบถามและจัดส่งแบบสอบถามไปยังหน่วยงานเอกชนตามรายชื่อที่ปรากฏในสรุปข้อสนเทศบริษัทจดทะเบียนตลาดหลักทรัพย์แห่งประเทศไทย บริษัท Software House ซึ่งเป็นหน่วยงานเอกชนที่ให้บริการด้านคอมพิวเตอร์ซอฟท์แวร์ ตามรายชื่อที่ปรากฏใน Computer Software Directory ที่จัดทำโดยกรมส่งเสริมการส่งออก กระทรวงพาณิชย์ รัฐวิสาหกิจที่เป็นสมาชิกในชมรมเทคโนโลยีสารสนเทศรัฐวิสาหกิจแห่งประเทศไทย และบริษัทนอกตลาดหลักทรัพย์แห่งประเทศไทย ผลการศึกษาแสดงให้เห็นว่า กิจการส่วนใหญ่ซื้อซอฟต์แวร์สำเร็จรูปแล้วนำมาปรับปรุงให้เหมาะสมกับกิจการอีกต่อหนึ่ง อนึ่ง ซอฟต์แวร์บางประเภทได้จากการพัฒนาขึ้นมาใช้ในกิจการเอง หรือพัฒนาตามความต้องการของลูกค้า หรือหน่วยงานอื่นๆ ภายในองค์กรเดียวกัน ในด้านวิธีการพัฒนาระบบพบว่า กิจการส่วนใหญ่ยังคงใช้วิธีพัฒนาระบบแบบ System Development Life Cycle หรือ SDLC และ Object-oriented Analysis/Design โดยเทคนิคที่นำมาช่วยในการพัฒนากิจการส่วนใหญ่ยังคงใช้เทคนิคที่มีการเรียนการสอนในชั้นเรียน กล่าวคือ ยังคงใช้เทคนิค Data Flow Diagram (DFD), Entity-Relationship (ER) อย่างไรก็ตามกิจการส่วนใหญ่มีแนวโน้มที่จะใช้ Object-Oriented techniques สำหรับพัฒนาระบบมากขึ้น อนึ่ง จะเห็นได้ว่ากิจการส่วนใหญ่โดยเฉพาะบริษัท Software House มีการนำโปรแกรมเครื่องมือที่ทำให้การทำงานด้านการวิเคราะห์และออกแบบระบบเป็นไปโดยอัตโนมัติน้อย โดยโปรแกรมที่กิจการนำมาใช้งานนั้น ไม่มีความหลากหลายมากนัก สำหรับภาษาคอมพิวเตอร์นั้น กิจการส่วนใหญ่ใช้ภาษา Visual Basic, C++ และ JAVA โดยกิจการส่วนใหญ่มีแนวโน้มที่จะใช้ภาษาคอมพิวเตอร์ด้าน Web page development (เช่น ASP/ASP.net, HTML, PHP เป็นต้น) มากขึ้น และความต้องการด้านบุคลากรที่กิจการส่วนใหญ่ต้องการสามอันดับแรก คือ พนักงานที่มีความรู้ด้านการบริหารโครงการ ทักษะด้านการวิเคราะห์ระบบ และความสามารถในการแก้ปัญหา

** งานวิจัยนี้ได้รับเงินสนับสนุนจากโครงการวิจัยเสริมหลักสูตร มหาวิทยาลัยธรรมศาสตร์*

1. บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันสถาบันการศึกษาในประเทศไทยที่มีการสอนทางด้านสารสนเทศเพื่อการจัดการ ได้จัดการเรียนการสอนเพื่อเตรียมความพร้อมของนักศึกษาให้มีความรู้และความสามารถที่จะปฏิบัติงานให้กับองค์กรต่างๆ ภายหลังสำเร็จการศึกษาโดยจัดให้มีการเรียนการสอนวิชาทางด้านสารสนเทศเป็นจำนวนมาก อย่างไรก็ตาม การผลิตนักศึกษาที่สามารถทำงานให้กับองค์กรได้อย่างมีประสิทธิผลนั้นสถาบันการศึกษาจำเป็นต้องจัดการเรียนการสอนวิชาที่ประกอบด้วยเนื้อหาที่ตรงกับความต้องการของผู้ว่าจ้างหรือองค์กร (Gupta et al., 1994) วิธีการหนึ่งที่จะทำให้แน่ใจว่าสถาบันการศึกษานั้นๆ ได้จัดเนื้อหาวิชาของการเรียนการสอนเหมาะสม คือ การสอบถามความต้องการจากผู้ว่าจ้างโดยตรง (Gupta et al., 1994)

แม้ว่าภาควิชาระบบสารสนเทศเพื่อการจัดการจะจัดให้มีการเรียนการสอนวิชาทางด้านสารสนเทศเพื่อการจัดการที่ทันต่อการเปลี่ยนแปลงทางด้านคอมพิวเตอร์อยู่เสมอ แต่ภาควิชายังไม่มีการสำรวจความต้องการด้านเนื้อหาของวิชาที่เปิดสอนอย่างเป็นระบบ เพื่อปรับเนื้อหาวิชาให้เหมาะสมกับความต้องการตลาดแรงงานในอนาคต โดยเฉพาะวิชาการวิเคราะห์และการออกแบบระบบสารสนเทศ ซึ่งในทางทฤษฎี ระเบียบวิธีการพัฒนา (Methodologies) เทคนิค (Techniques) และเครื่องมือ (Tools) ที่ใช้สำหรับการวิเคราะห์และออกแบบระบบสารสนเทศนั้นมีหลากหลาย อาทิเช่น ระเบียบวิธีการพัฒนาที่ใช้อาจเป็น System Development Life Cycle (SDLC) หรือ Waterfall Approach, Iterative Approach (เช่น Prototyping, Spiral, หรือ Participatory Design), Object-Oriented Analysis and Design, Structrued Methods ต่างๆ (เช่น Yourdon Structured Method, Soft Systems Method/Multiview, Structured Systems Analysis and Design Method (SSADM), และ Information Engineering (IE) เป็นต้น) ซึ่งการเลือกใช้ระเบียบวิธีการพัฒนาที่แตกต่างกัน ย่อมส่งผลต่อการเลือกใช้เทคนิคและเครื่องมือที่มีอยู่หลากหลาย กล่าวคือ เทคนิคที่เลือกใช้อาจเป็น Data Flow Diagram (DFD), Action Diagram, Function Diagram, State-transition Diagram, Event Diagram, Entity Model, Data Dictionary, Decision Table, Decision Tree, Structured chart, และ Unified Modeling Language เป็นต้น (Tudor and Tudor, 1997; Hoffer et al., 1996; Satzinger et al., 2000)

ดังนั้น ภาควิชาจึงมีความจำเป็นต้องกำหนดเนื้อหาของวิชาให้ครอบคลุมถึงระเบียบวิธีการพัฒนาเทคนิค และเครื่องมือซึ่งเป็นที่นิยมใช้กันอย่างแพร่หลายทั้งในปัจจุบันและในอนาคตขององค์กรต่างๆ ในประเทศไทยที่เป็นผู้ว่าจ้างหลักของบัณฑิตที่ภาควิชาระบบสารสนเทศเพื่อการจัดการผลิตออกไปสู่ตลาดแรงงาน

1.2 วัตถุประสงค์ของการศึกษา

เพื่อสำรวจระเบียบวิธีการพัฒนา เทคนิคและเครื่องมือที่ใช้สำหรับการวิเคราะห์และออกแบบระบบสารสนเทศที่องค์กรต่างๆ ในประเทศไทย (ซึ่งเป็นผู้ว่าจ้างหลักของบัณฑิตภาควิชาระบบสารสนเทศเพื่อการจัดการ) ใช้งาน และทำการวิเคราะห์ผลจากการสำรวจเพื่อนำไปปรับปรุงเนื้อหาวิชาการวิเคราะห์และออกแบบระบบสารสนเทศเพื่อการจัดการให้ตรงตามความต้องการของผู้ว่าจ้างต่อไป

1.3 ประโยชน์ที่คาดว่าจะได้รับ

ผลของงานวิจัยจะเป็นประโยชน์ในด้านการวางแผนจัดทำเนื้อหาของวิชาวิเคราะห์และออกแบบระบบ รวมทั้งจัดหากิจกรรมสำหรับการเรียนการสอนวิชานี้เพื่อเสริมทักษะที่จำเป็นสำหรับพัฒนาให้บัณฑิตสาขาวิชาระบบสารสนเทศสามารถพัฒนาตนเองให้เป็นนักวิเคราะห์ที่ประสบผลสำเร็จต่อไปในภายหน้าได้

2. แนวคิดและงานวิจัยที่เกี่ยวข้อง

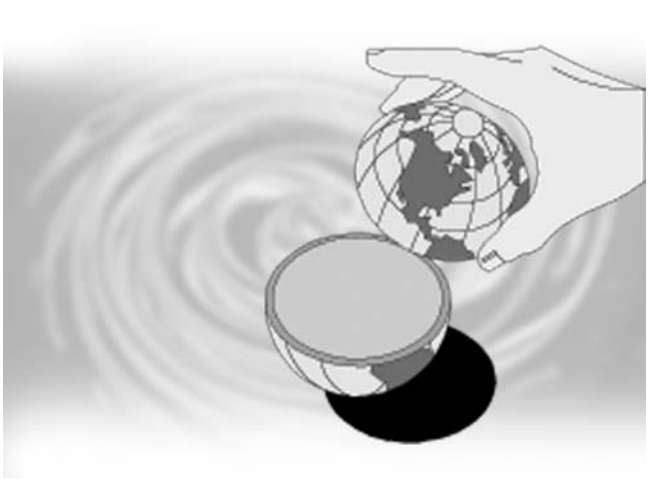
แนวคิดและงานวิจัยที่เกี่ยวข้องซึ่งคณะผู้วิจัยใช้เป็นพื้นฐานของการทำวิจัยนี้เป็นหัวเรื่องที่มีในเนื้อหาวิชาการวิเคราะห์และออกแบบระบบงานคอมพิวเตอร์ที่เป็นหลักสูตรของสถาบันที่คณะผู้วิจัยสอน หัวข้อเหล่านี้ ได้แก่ *(1) แหล่งของระบบสารสนเทศในองค์กร (2) ระเบียบวิธีการพัฒนา (Methodologies) เทคนิค (Techniques) เครื่องมือ (Tools) และภาษาคอมพิวเตอร์ (Computer Languages) ที่นำมาใช้ในการพัฒนาระบบงานคอมพิวเตอร์ (3) การพัฒนาระบบกับความสำเร็จของระบบสารสนเทศที่องค์กรจัดหามาใช้งาน และ (4) ทักษะทีมพัฒนาหรือความชำนาญที่ทำให้นักวิเคราะห์ประสบความสำเร็จในหน้าที่การทำงาน*

2.1 แหล่งที่มาของระบบสารสนเทศในองค์กร

องค์กรที่ต้องการนำระบบคอมพิวเตอร์มาใช้สนับสนุนการทำงานขององค์กรสามารถจัดหาระบบงานดังกล่าวได้โดยวิธีการใดวิธีการหนึ่ง คือ การพัฒนาเอง (In-House Development) การใช้บริการภายนอก (Outsourcing) หรือการซื้อโปรแกรมสำเร็จรูป (Software Package) (Laudon and Laudon, 2003)

2.1.1 การพัฒนาเอง ภายใต้วิธีการนี้บุคลากรภายในองค์กรสร้างระบบงานคอมพิวเตอร์ขึ้นมาเอง โดยการสร้างนั้นอาจเป็นการขยายหรือต่อเติมระบบงานเดิม หรือระบบงานปัจจุบันที่กำลังใช้งานอยู่ หรืออาจเป็นการสร้างระบบงานใหม่ที่ยังไม่เคยมีการใช้งานมาก่อนก็ได้ บุคลากรผู้สร้างระบบอาจเป็นบุคลากรที่อยู่ในหน่วยงานด้านระบบสารสนเทศที่ทำหน้าที่รับผิดชอบในการให้บริการด้านระบบงานคอมพิวเตอร์โดยตรง หรืออาจเป็นบุคลากรในฝ่ายผู้ใช้ระบบงาน ซึ่งโดยทั่วไปแล้วบุคลากรทั้งสองกลุ่มนี้จะใช้ระเบียบวิธีการพัฒนาที่แตกต่างกัน (รายละเอียดจะได้กล่าวต่อไปในหัวข้อ **"ระเบียบวิธีการพัฒนา เทคนิค เครื่องมือ และภาษาคอมพิวเตอร์"**) วิธีการนี้มักใช้ในกรณีที่เป็นองค์กรขนาดใหญ่ที่มีบุคลากรด้านระบบสารสนเทศที่มีความชำนาญเป็นของตนเอง รวมทั้งระบบงานที่ต้องจัดหานั้นเป็นระบบที่มีคุณลักษณะเฉพาะเป็นระบบที่มีความจำเป็นต่อการดำเนินงานขององค์กรและสามารถใช้เป็นกลยุทธ์ในการแข่งขันขององค์กรได้ (ศรีสมรัก, 2542; McFarlan, 1995)

2.1.2 การใช้บริการภายนอก มีอยู่หลายรูปแบบทั้งนี้ขึ้นอยู่กับขอบเขตข้อตกลงที่ทำสัญญากันระหว่างองค์กรและผู้ให้บริการ ตัวอย่างเช่น การจ้างองค์กรภายนอกที่มีความชำนาญด้านระบบสารสนเทศ ซึ่งอาจเป็นบริษัทผู้ผลิตซอฟต์แวร์ตามคำสั่งของลูกค้า (Custom Software House) หรือบริษัทให้คำปรึกษาด้านระบบ/เทคโนโลยีสารสนเทศ (เช่น บริษัท IBM, Oracle, Price Waterhouse-Coopers และ Anderson Consulting เป็นต้น) (Hoffer et al., 2002) สร้างระบบงานคอมพิวเตอร์ที่มีลักษณะเฉพาะให้กับองค์กร หรือการให้องค์กรภายนอกประมวลผลข้อมูลให้โดยองค์กรจะคอยรับรายงานที่ได้จากการประมวลผลและรับผิดชอบแต่เฉพาะในเรื่องการส่งข้อมูลนำเข้าเพื่อให้องค์กรภายนอกนำไปประมวลผลเท่านั้น หรือการใช้โปรแกรมประยุกต์ขององค์กรที่ให้บริการโปรแกรมประยุกต์ (Application Service Providers หรือ ASP) ซึ่งโปรแกรมประยุกต์ที่องค์กรเหล่านี้มีไว้ให้บริการลูกค้าอาจเป็นโปรแกรมที่ผู้ให้บริการนั้นพัฒนาขึ้นมาเอง หรือเป็นโปรแกรมสำเร็จรูปที่ผู้ให้บริการ



ซื้อมาจากบริษัทผู้ผลิตซอฟท์แวร์ (เช่น Oracle และ Microsoft) แล้วนำมาติดตั้งที่เครื่องเซิร์ฟเวอร์ (Server) เพื่อให้บริการกับองค์กรต่างๆ อีกต่อหนึ่ง (Hololan and Hall, 2000; Wilcox and Farmaer, 2000) หรือการใช้บริการในรูปแบบอื่นๆ อีก ซึ่งขึ้นอยู่กับขอบเขตข้อตกลงที่ทำสัญญากันระหว่างองค์กรและผู้ให้บริการ

2.1.3 การซื้อโปรแกรมสำเร็จรูป ในกรณีที่ระบบงานที่ต้องการจัดหาเป็นระบบสนับสนุนงานทั่วไปที่เงื่อนไขการทำงานและกระบวนงานของระบบไม่ได้มีคุณลักษณะเฉพาะมากนักองค์กรส่วนมากนิยมที่จะจัดหาซอฟต์แวร์โดยการซื้อโปรแกรมสำเร็จรูปที่บริษัทผู้ผลิตซอฟต์แวร์ (Software House) จัดทำขึ้นเพื่อจำหน่าย เพราะวิธีการนี้ทำให้ได้ระบบงานคอมพิวเตอร์มาใช้ภายในเวลาอันรวดเร็ว และมีค่าใช้จ่ายต่ำกว่าวิธีการพัฒนาขึ้นเอง (Laudon and Laudon, 2003) ซอฟต์แวร์ที่องค์กรซื้อมาใช้งานอาจเป็นโปรแกรมสำเร็จรูปที่เรียกว่า Off-The-Shelf Software ซึ่งมักเป็นโปรแกรมที่มีขนาดเล็กและมีความสลับซับซ้อนไม่มากนัก เช่น โปรแกรมบัญชีแยกประเภท Formula, ALL-In-One และ Express เป็นต้น หรืออาจเป็นการซื้อโปรแกรมสำเร็จรูปที่เรียกว่าระบบการวางแผนทรัพยากรของวิสาหกิจ (Enterprise Resource Planning หรือ ERP) เพื่อสนับสนุนหน้าที่งานหน้าที่ใดหน้าที่หนึ่งขององค์กรธุรกิจ (เช่น ด้านบัญชี ด้านการผลิต และด้านทรัพยากรบุคคล เป็นต้น)

จากการศึกษาต่างๆ ในสหรัฐอเมริกาพบว่า การจัดหาระบบงานด้วยวิธีการพัฒนาเองมีแนวโน้มลดลง และในขณะที่การจัดหาโดยวิธีการใช้บริการจากแหล่งภายนอก และการซื้อโปรแกรมสำเร็จรูปมาใช้งานจะมีแนวโน้มเพิ่มมากขึ้น จากการศึกษาของ Corbett Group พบว่า ในปี 1998 การใช้บริการจากแหล่งภายนอกมีแนวโน้มเพิ่มสูงขึ้น โดยสังเกตได้จากแนวโน้มของค่าใช้จ่ายของการใช้บริการจากแหล่งภายนอกที่เพิ่มขึ้น (ค่าใช้จ่ายของการใช้บริการจากแหล่งภายนอกในปี 1998 สูงกว่าในปี 1997 ถึง 97%) และความพึงพอใจของผู้บริหารในองค์กรที่ใช้บริการจากภายนอกสูงถึง 60% (Merrill, 1999) ในขณะที่ King และ Cole-Gomolski พบว่าเวลาและเงินที่ใช้ในการพัฒนาและบำรุงรักษาซอฟต์แวร์สำหรับระบบสารสนเทศในปี 1998 ลดลงน้อยกว่าปี 1997 ถึง 33% และการจัดหาซอฟต์แวร์โดยการซื้อซอฟต์แวร์สำเร็จรูปเพิ่มขึ้นถึง 3 เท่า เนื่องจากการพัฒนาโปรแกรมภายในเองทำให้องค์กรมีค่าใช้จ่ายในการบำรุงรักษามากกว่าการจัดหาซอฟต์แวร์ด้วยวิธีการซื้อซอฟต์แวร์สำเร็จรูป (Banker et al., 1998) แต่สำหรับในประเทศไทย จากการศึกษาของ ดร.สมบูรณ์วัลย์และคณะ (2539) พบว่าการจัดหาระบบงานคอมพิวเตอร์โดยใช้วิธีการพัฒนาเองภายในองค์กรเป็นวิธีการที่ถูกใช้มากที่สุด (42%) รองลงมาได้แก่ วิธีการจ้างให้หน่วยงานภายนอกพัฒนาระบบงานให้ (ประมาณ 27%) และการซื้อโปรแกรมสำเร็จรูปมาใช้งาน (ประมาณ 20%)

2.2 ระเบียบวิธีการพัฒนา เทคนิค เครื่องมือ และภาษาคอมพิวเตอร์

ในการพัฒนาระบบงานคอมพิวเตอร์แต่ละระบบหน่วยงานภายในองค์กรที่รับผิดชอบในการพัฒนาระบบงานคอมพิวเตอร์และบริษัทผู้ผลิตซอฟต์แวร์อาจเลือกใช้ระเบียบวิธีการพัฒนา เทคนิค และเครื่องมือสำหรับช่วยสนับสนุนการพัฒนาระบบที่แตกต่างกันไปดังต่อไปนี้

2.2.1 ระเบียบวิธีการพัฒนา ระเบียบวิธีการพัฒนาที่หน่วยงานระบบสารสนเทศและบริษัทผู้ผลิตซอฟต์แวร์นิยมใช้กันโดยทั่วไป ได้แก่ ระเบียบวิธีวงจรการพัฒนาระบบ (System Development Life Cycle หรือ SDLC) หรือ Waterfall Approaches การพัฒนาระบบอย่างรวดเร็ว (Rapid Application Development หรือ RAD) และการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis/Design) (Hoffer et al., 2002) แต่ถ้าบุคลากรในฝ่ายผู้ใช้เป็นผู้พัฒนาระบบงานเอง มักใช้การพัฒนาโดยผู้ใช้ขั้นปลาย (End User Development) เป็นระเบียบวิธีในการพัฒนาระบบงาน (Laudon and Laudon, 2003; Turban, 2002)

2.2.1.1 ระเบียบวิธีวงจรการพัฒนาระบบ ระเบียบวิธีวงจรการพัฒนาระบบ เป็นระเบียบวิธีการพัฒนาที่ถูกพัฒนามาหลายสิบปีและเป็นที่รู้จักกันดีในหลายๆ องค์กร วิธีการนี้มีคุณสมบัติเด่น คือ การแบ่งกระบวนการทำงานของโครงการพัฒนาระบบงานออกเป็นขั้นตอนต่างๆ หลายขั้นตอนเพื่อแสดงให้เห็นถึงความก้าวหน้าของการวิเคราะห์และออกแบบระบบ (Hoffer et al., 2002) โดยในแต่ละขั้นตอนจะกำหนดกิจกรรมพื้นฐานและผลลัพธ์ที่ต้องจัดทำให้แล้วเสร็จก่อนการทำงานในขั้นตอนถัดไปไว้อย่างชัดเจน (Bocij et al., 1999) และเมื่อทำงานในแต่ละขั้นตอนเสร็จ ผู้บริหารของฝ่ายผู้ใช้ต้องลงนามอย่างเป็นทางการ (Formal Sign-Offs or Agreement) ก่อนที่จะทำงานในขั้นตอนถัดไป ทั้งนี้ เพื่อให้เกิดการยอมรับในผลลัพธ์ที่เกิดขึ้นจากการทำงานในแต่ละขั้นตอน อนึ่ง การแก้ไขมติหรือเปลี่ยนแปลงข้อกำหนดที่ตกลง (Sign-Off) ไปแล้วในขั้นตอนก่อนหน้านี้ไม่สามารถทำได้ หรือทำได้ในขอบเขตที่จำกัดมาก (ด้วยเหตุนี้ระเบียบวิธีการพัฒนานี้จึงมีชื่อเรียกอีกชื่อหนึ่งว่า Waterfall Approach) นอกจากนี้ระเบียบวิธีการพัฒนานี้ยังมีการแบ่งหน้าที่ความรับผิดชอบระหว่างฝ่ายผู้ใช้และทีมพัฒนาระบบไว้อย่างชัดเจนในแต่ละขั้นตอนของกระบวนการพัฒนาระบบ (Hoffer et al., 2002) เช่น ในช่วงการวิเคราะห์ผู้ใช้มีหน้าที่ให้ข้อกำหนดความต้องการระบบส่วนทีมพัฒนาระบบมีบทบาทในการวิเคราะห์หาจุดอ่อนของระบบงานที่ใช้งานอยู่ พร้อมทั้งเสนอคุณลักษณะของระบบงานใหม่

ที่สามารถแก้ไขจุดอ่อนหรือปัญหาการทำงานของระบบปัจจุบันได้ เป็นต้น

การแบ่งขั้นตอนในกระบวนการพัฒนาระบบงานคอมพิวเตอร์ตามระเบียบวิธีการพัฒนานี้สามารถจัดแบ่งได้หลายขั้นตอนย่อย ซึ่งการแบ่งขั้นตอนย่อยของแต่ละองค์กรมักจะแตกต่างกันไป (Curtis and Cabham, 2002; Hoffer et al., 2002; Laudon and Laudon, 2002; Turban 2002; Oz, 2002; Bocij et al., 1999) แต่เมื่อพิจารณาเนื้อหาสาระสำคัญแล้ว กิจกรรมและผลลัพธ์ที่จะต้องทำในแต่ละขั้นตอนตลอดช่วงเวลาการพัฒนาระบบงานนั้นไม่แตกต่างกันมากนัก ดังนั้น ในการศึกษานี้จึงแบ่งขั้นตอนการพัฒนาระบบงานเป็น 6 ขั้นตอนตามลำดับการทำงานก่อนหลัง คือ การกำหนดและเลือกโครงการ, เริ่มต้นและวางแผนโครงการ, การวิเคราะห์, การออกแบบ, การปรับใช้, และการบำรุงรักษา (รายละเอียดแต่ละขั้นตอนศึกษาได้จาก Hoffer et al. (2002))

ระเบียบวิธีวงจรการพัฒนาระบบมักได้รับคำวิพากษ์วิจารณ์อย่างมากในประเด็นของความล่าช้าของกระบวนงาน (Procedures) ที่ค่อนข้างเป็นทางการอย่างมาก จนทำให้ระบบที่พัฒนาได้ ไม่สามารถสนองตอบความต้องการที่แท้จริงของผู้ใช้ซึ่งเปลี่ยนแปลงไปตามเงื่อนไขการทำงานทางธุรกิจได้ (Hoffer et al., 2002) ส่งผลทำให้มีค่าใช้จ่ายในการบำรุงรักษาระบบเกิดขึ้นเป็นจำนวนมาก ประมาณร้อยละ 40-70 ของต้นทุนการพัฒนาระบบงานทั้งระบบ (Dorfman และ Thayer, 1997) ดังนั้น จึงมีการคิดหาวิธีการ (Approaches) ต่างๆ มากมายเพื่อเพิ่มประสิทธิภาพของระเบียบวิธีการพัฒนานี้ วิธีการหนึ่งซึ่งเป็นที่นิยมกัน ได้แก่ วิธีการวิเคราะห์และออกแบบเชิงโครงสร้าง (Structured analysis and Design) หรือการพัฒนาระบบเชิงโครงสร้าง (Structured development) วิธีการนี้พยายามนำแนวคิดของการทำงานทางวิศวกรรมมาปรับใช้เพื่อทำให้การวิเคราะห์และออกแบบเป็นระเบียบแบบแผนทางวิทยาศาสตร์มากขึ้น และสามารถย้อนกลับไปแก้ไขงานและทบทวนการทำงานในขั้นตอนก่อนๆ ได้หากมีความจำเป็น (Curtis and Cabham,2002; Hoffer et al., 2002; Yourdon, 1989)

แม้ว่าจะมีการนำวิธีการวิเคราะห์และออกแบบระบบเชิงโครงสร้างมาใช้ เพื่อเพิ่มแบบแผนและประสิทธิภาพการทำงานให้กับกิจกรรมในขั้นตอนต่างๆ ของการพัฒนาระบบแล้วก็ตาม ระบบงานที่พัฒนาได้ก็ยังคงไม่สอดคล้องตรงตามความต้องการของผู้ใช้ เนื่องจากการแบ่งหน้าที่ความรับผิดชอบอย่างชัดเจนระหว่างฝ่ายผู้ใช้และทีมพัฒนาระบบ ในแต่ละขั้นตอนของการทำงานในกระบวนการพัฒนาระบบ การแบ่งหน้าที่ความรับผิดชอบดังกล่าว ทำให้เกิดข้อจำกัดของการที่ผู้ใช้จะเข้าไปมีส่วนร่วมในกระบวนการพัฒนาระบบอย่างจริงจัง โดยเฉพาะในขั้นตอนการออกแบบระบบ (Bocij et al., 1999) ดังนั้น จึงมีการพัฒนาระเบียบวิธีการพัฒนาใหม่ขึ้นอีกเป็นจำนวนมาก เพื่อให้ผู้ใช้ได้มีส่วนร่วมในการออกแบบระบบมากขึ้น และ/หรือเพื่อย่นระยะเวลาการสร้างระบบงานให้สั้นลง ตัวอย่างระเบียบวิธีการพัฒนาเหล่านี้ เช่น Participation Design (หรือ PD) และ Joint Application Design (หรือ JAD) เป็นต้น (Hoffer et al., 2002) อนึ่ง ระหว่างระเบียบวิธีการพัฒนาเหล่านี้ ระเบียบวิธีการพัฒนาระบบอย่างรวดเร็ว (Rapid application development หรือ RAD) (ซึ่งรายละเอียดจะได้กล่าวในหัวข้อถัดไป) เป็นระเบียบวิธีการหนึ่งที่ผู้พัฒนาระบบในหลายองค์กรคุ้นเคยกันเป็นอย่างดี

2.2.1.2 ระเบียบวิธีการพัฒนาระบบอย่างรวดเร็ว RAD เป็นระเบียบวิธีการพัฒนาที่เน้นในเรื่องการให้ผู้ใช้มีส่วนร่วมในการวิเคราะห์ ออกแบบ และสร้างระบบ โดยให้ผู้พัฒนาระบบและผู้ใช้ร่วมกันสร้างระบบต้นแบบระหว่างทำ (Working prototype) ให้ได้โดยเร็วเพื่อลดระยะเวลาการพัฒนาระบบให้น้อยลง (Whitten et al., 2001; Bocij et al., 1999) โดยหลักการพื้นฐานของ RAD คือ ชะลอการผลิตรายละเอียดของเอกสารข้อกำหนดคุณลักษณะของการออกแบบระบบออกไปจนกว่าจะได้ข้อกำหนดความต้องการระบบและข้อกำหนดคุณลักษณะของระบบที่ชัดเจนแล้ว (Hoffer et al., 2002) จากการที่ผู้ใช้ได้เห็นและทดลองใช้ระบบต้นแบบที่สร้างขึ้น (Whitten et al., 2001) ดังนั้น ระเบียบวิธีการพัฒนานี้จึงให้ความสำคัญกับการสร้างและปรับปรุงระบบต้นแบบตามความคิดเห็นของผู้ใช้ ซึ่งระบบต้นแบบดังกล่าวเป็นผลจากการทดลองใช้และประเมินระบบต้นแบบหลายๆ รอบ การกระทำดังกล่าวทำให้ได้รับการยอมรับจากผู้ใช้ในด้านส่วนต่อประสานของระบบที่กำลังพัฒนาและความสามารถหลักของระบบโดยเร็วที่สุด อนึ่ง ระบบต้นแบบที่สร้างขึ้นยังไม่คำนึงถึงประสิทธิภาพในการทำงานของระบบและการต่อเชื่อมกับระบบสารสนเทศอื่น (Curtis and Cabham, 2002; Whitten et al., 2001) โดยทั่วไปขั้นตอนการทำงานของระเบียบวิธีการพัฒนานี้น้อยกว่ากระบวนการทำงานของ SDLC กล่าวคือ RAD จะประกอบด้วยขั้นตอนเพียง 4 ขั้นตอน ได้แก่ *ขั้นตอนการวางแผน* (ซึ่งเทียบได้กับสองขั้นตอนของระเบียบวิธีการพัฒนาแบบ SDLC ที่กล่าวข้างต้น) *ขั้นตอนการออกแบบ* (ซึ่งเทียบได้กับขั้นตอนการวิเคราะห์และการออกแบบระบบในเชิงตรรกะของระเบียบวิธีการพัฒนาแบบ SDLC ที่กล่าวข้างต้น) *ขั้นตอนการพัฒนา/สร้าง* (ซึ่งเทียบได้กับขั้นตอนการออกแบบระบบในเชิงกายภาพการจัดทำ

โปรแกรมของระเบียบวิธีการพัฒนาแบบ SDLC ที่กล่าวข้างต้น) และ *ขั้นตอนการจบโครงการ* (ซึ่งประกอบด้วยการทดสอบระบบ และการอบรมผู้ใช้) (Curtis and Cabham, 2002; Hoffer et al., 2002)

2.2.1.3 การวิเคราะห์และออกแบบระบบเชิงวัตถุ การที่ระบบสารสนเทศมีความสำคัญอย่างมากต่อการดำเนินชีวิตและการดำเนินงานขององค์กรต่างๆ ทั้งในปัจจุบันและในอนาคต ทำให้องค์กรต่างๆ ต้องการที่จะได้ระบบสารสนเทศมาใช้งานโดยเร็ว การใช้ระเบียบวิธีการพัฒนาแบบ Structured SDLC และ RAD ซึ่งแยกข้อมูล (Data) และกระบวนการประมวลผล (Processes) และการดำเนินการ (Operate) กับข้อมูลออกจากกันอย่างชัดเจน โดยทั่วไปหากใช้ระเบียบวิธีการพัฒนาเหล่านี้ สิ่งที่ต้องพิจารณาอันดับแรก คือ ระบบสารสนเทศที่พัฒนาควรจะทำอะไรได้บ้าง จากนั้นจึงออกแบบกระบวนงานต่างๆ (Procedures) ของระบบ และโครงสร้างของข้อมูลในระบบที่สามารถใช้สนับสนุนกระบวนงานเหล่านั้นได้ แนวคิดนี้ทำให้การทำงานของระบบสารสนเทศที่ได้ขึ้นอยู่กับตัวแบบกรรมวิธีการทำงาน (Process model) ของกระบวนงานใดกระบวนงานหนึ่งและตัวแบบข้อมูล (Data model) ที่ถูกออกแบบไว้เฉพาะกับระบบใดระบบหนึ่ง ส่งผลให้ไม่สามารถนำส่วนประกอบของระบบสารสนเทศบางส่วนที่มีอยู่ในระบบเดิมกลับมาใช้งานได้อีก แม้ว่าส่วนประกอบบางส่วน ในระบบใหม่จะมีคุณสมบัติและคุณลักษณะที่เหมือนกับระบบเดิมก็ตาม ดังนั้น ทุกครั้งที่พัฒนาระบบสารสนเทศใหม่จึงต้องพัฒนาและปรับปรุงส่วนประกอบต่างๆ ของระบบงานใหม่ทั้งหมด ส่งผลให้การพัฒนาระบบงานใช้เวลาและค่าใช้จ่ายที่มากเกินไป (Laudon and Laudon, 2003)

อนึ่ง ระเบียบวิธีการวิเคราะห์และออกแบบระบบเชิงวัตถุให้ความสำคัญกับกระบวนงานน้อยลง และเปลี่ยนวิธีการมองระบบสารสนเทศที่ประกอบด้วยองค์ประกอบ 2 ส่วนหลัก คือ ส่วนข้อมูลและส่วนกรรมวิธีการดำเนินการข้อมูล โดยที่องค์ประกอบสองส่วนนี้แยกออกจากกันแต่มีความสัมพันธ์กัน เป็นการมองว่าระบบสารสนเทศแต่ละระบบประกอบด้วยอ็อบเจ็กต์ (Objects) ต่างๆ หรือชุดของอ็อบเจ็กต์ชนิดต่างๆ ที่มีความสัมพันธ์กัน โดยอ็อบเจ็กต์แต่ละอ็อบเจ็กต์เกิดจากการนำส่วนของข้อมูล (Data element) บางตัวและกรรมวิธีการดำเนินการข้อมูลบางอย่างมารวมกัน (Laudon and Laudon, 2003)

2.2.1.4 การพัฒนาโดยผู้ใช้ขั้นปลาย การพัฒนาระบบโดยผู้ใช้ขั้นปลาย หมายถึง การที่บุคลากรในฝ่ายผู้ใช้ระบบสารสนเทศเป็นผู้พัฒนาระบบสารสนเทศขึ้นมาใช้เอง เช่น การใช้ซอฟต์แวร์กระดาษทำการอิเล็กทรอนิกส์ (Spreadsheet) จัดทำตัวแบบ What-if ที่ช่วยในการตัดสินใจเรื่องใดเรื่องหนึ่งในหน้าที่งานที่ตนรับผิดชอบ การเรียกข้อมูลจากฐานข้อมูลมาจัดทำรายงานโดยการใช้ภาษาสอบถาม และการจัดกำหนดตารางการทำงาน เป็นต้น โดยส่วนมากระบบที่พัฒนาด้วยวิธีการนี้มักเป็นระบบขนาดเล็กและเป็นระบบที่ใช้สนับสนุนการทำงานของหน่วยงานใดหน่วยงานหนึ่งในองค์กร (Departmental application) ในอนาคตระบบสารสนเทศในองค์กรที่พัฒนาด้วยวิธีการนี้มีแนวโน้มเพิ่มมากขึ้น ด้วยเหตุผลหลัก 2 ประการ *ประการแรก* คือ พัฒนาการทางด้านเทคโนโลยีสารสนเทศ คอมพิวเตอร์ส่วนบุคคล (Personal Computer หรือ PC) และอุปกรณ์คอมพิวเตอร์สำหรับ PC ที่นับวันจะยิ่งมีราคาถูกลงในขณะที่สมรรถนะการทำงานเพิ่มมากขึ้น มีการพัฒนาซอฟต์แวร์ที่มีส่วนต่อประสานกราฟิกกับผู้ใช้ที่นับวันจะใช้งานได้ง่ายขึ้น (Ease-of-Use) มากขึ้น ความก้าวหน้าของข่ายงานคอมพิวเตอร์ทำให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลขององค์กรซึ่งติดตั้งบนเครื่องคอมพิวเตอร์แพลตฟอร์มใดก็ได้ และ

ประการที่สอง คือ ความต้องการระบบสารสนเทศของหน่วยงานต่างๆ เพิ่มขึ้น ทำให้หน่วยงานสารสนเทศไม่สามารถให้บริการทันกับความต้องการระบบงานของผู้ใช้ นอกจากนี้ การพัฒนาระบบโดยวิธีการข้างต้นทำให้ได้ระบบสารสนเทศที่ตรงตามความต้องการมาใช้ภายในเวลาอันรวดเร็ว ส่งผลให้สามารถลดค่าใช้จ่ายโดยรวมของระบบสารสนเทศขององค์กรได้ ทำให้วิธีการนี้มีแนวโน้มที่จะได้รับความนิยมเพิ่มขึ้นในอนาคต (Bocij et al., 1999)

2.2.2 เทคนิค เทคนิคการพัฒนาระบบแต่ละเทคนิคถูกพัฒนาขึ้นโดยมีวัตถุประสงค์ที่แตกต่างกัน เช่น Joint Application Design (หรือ JAD) เป็นเทคนิคที่สร้างขึ้นเพื่อให้ผู้ที่มีบทบาทต่อการพัฒนาระบบงานมีส่วนร่วมในการแสดงความคิดเห็นเกี่ยวกับเรื่องใดเรื่องหนึ่งเพื่อหาข้อยุติในเรื่องนั้นๆ ส่วนเทคนิคแผนภาพ (Diagram Techniques) เป็นเทคนิคที่ถูกพัฒนาขึ้นเพื่อใช้ในการจัดทำตัวแบบ (Model) ของระบบสารสนเทศที่พัฒนาเพื่อให้ทีมพัฒนาระบบสามารถทำความเข้าใจกับระบบสารสนเทศนั้นในมิติต่างๆ ได้โดยง่าย เทคนิคบางเทคนิคสามารถนำมาใช้ได้ในขั้นตอนบางขั้นตอนของการพัฒนาระบบ โดยที่บางเทคนิคสามารถนำมาใช้ได้กับหลายขั้นตอน เช่น Entity Relationship Diagram หรือ ERD และ Structured English สามารถนำมาใช้ได้ทั้งในขั้นตอนการวิเคราะห์และออกแบบ ในขณะที่เทคนิคบางเทคนิคสามารถนำมาใช้ได้เฉพาะแต่ในขั้นตอนใดขั้นตอนหนึ่งเท่านั้น เช่น Structured Chart เป็นเทคนิคที่สามารถนำมาใช้ในขั้นตอนการออกแบบเท่านั้น และ Gantt Chart สามารถนำมาใช้ได้เฉพาะแต่ในขั้นตอนการวางแผนเท่านั้น

นอกจากนี้เทคนิคบางเทคนิค (โดยเฉพาะเทคนิคแผนภาพ) ถูกพัฒนามาเพื่อใช้กับระเบียบวิธีการพัฒนาระเบียบวิธีใดระเบียบวิธีหนึ่งโดยเฉพาะ เช่น Data Flow Diagram และ Structured Chart เป็นเทคนิคที่ถูกพัฒนาขึ้นเพื่อใช้กับระเบียบวิธีการพัฒนาระบบเชิงโครงสร้าง ส่วน Use Cases Diagram เป็นเทคนิคที่ถูกพัฒนาขึ้นเพื่อใช้กับระเบียบวิธีการพัฒนาระบบเชิงวัตถุ ในขณะที่เทคนิคบางเทคนิคที่พัฒนาขึ้นมาสำหรับใช้กับระเบียบวิธีการพัฒนาวิธีหนึ่ง แต่สามารถนำไปใช้ได้กับระเบียบวิธีการพัฒนาอื่นๆ ได้ เช่น การสัมภาษณ์ การสังเกตการณ์ และการจัดทำแบบสอบถาม เป็นเทคนิคพื้นฐานที่สามารถใช้ได้กับทุกระเบียบวิธีการพัฒนา และ JAD สามารถใช้ได้ทั้งในระเบียบวิธีการพัฒนาแบบ SDLC และ RAD เป็นต้น

2.2.2.1 Structured development มีเทคนิคแผนภาพจำนวนมากที่ถูกพัฒนาขึ้นมาใช้ในการจัดโครงสร้างความต้องการระบบและ/หรือข้อกำหนดคุณลักษณะของระบบ ซึ่งเป็นกิจกรรมที่เกิดขึ้นในขั้นตอนการวิเคราะห์และ/หรือขั้นตอนการออกแบบของระเบียบวิธีการพัฒนาแบบ Structured SDLC อนึ่ง งานวิจัยนี้จะศึกษาบางเทคนิคที่ปรากฏในตำราด้านการวิเคราะห์และออกแบบระบบส่วนมากเท่านั้น เทคนิคเหล่านี้ คือ Data Flow Diagram (DFD), Decision Table, Decision Tree, Structured English, State Transition Diagram, Entity-Relationship (ER), Action Diagram, Dialogue Design Diagram, Hierarchy Chart, Structured Chart, System Flow Chart, และHierarchical Input-Process-Output II (HIPO II) (รายละเอียดแต่ละเทคนิคศึกษาได้จาก Hoffer et al. (2002); Kendall and Kendall (2002); Jerva (2001); Whitten et al. (2001); Dennis and Wixom (2000); Satzinger et al. (2000); Shelly et al. (1998); Tudor and Tudor (1997); Hoffer et al. (1996); Martin (1987))

นอกจากเทคนิคแผนภาพที่ใช้ในการจัดโครงสร้างระบบสารสนเทศแล้ว ยังมีเทคนิคอื่นๆ อีกที่ใช้ในกระบวนการพัฒนาระบบ เทคนิคเหล่านั้น ได้แก่ การสัมภาษณ์, การสังเกตการณ์, การใช้แบบสอบถาม, Joint Application Design (หรือ JAD), Data Dictionary, และ Project Management Techniques เป็นต้น

2.2.2.2 Object-oriented Analysis and design เนื่องจากแนวคิดของการวิเคราะห์และออกแบบเชิงวัตถุแตกต่างจากแนวคิดของการวิเคราะห์และออกแบบเชิงโครงสร้างอย่างมาก เทคนิคแผนภาพต่างๆ ที่ใช้ในการจัดทำตัวแบบระบบของการวิเคราะห์และออกแบบเชิงโครงสร้างจึงไม่สามารถนำมาใช้กับการวิเคราะห์และออกแบบเชิงวัตถุ ดังนั้น จึงมีการพัฒนาเทคนิคแผนภาพขึ้นมาใหม่ที่สำคัญ ได้แก่ แผนภาพของ Booch, แผนภาพ OMT (Object Modeling Technique) ของ Rumbaugh, และแผนภาพ OOSE (Object-Oriented Software Engineering) ของ Jacobson แต่ความหลากหลายของเทคนิคทำให้เป็นอุปสรรคต่อการพัฒนาระบบสารสนเทศโดยใช้แนวคิดนี้ ดังนั้น ปี 1997 Object Management Group (หรือ OMG) ซึ่งเป็นกลุ่มขององค์กร บริษัทซอฟต์แวร์ และผู้พัฒนาระบบในสหรัฐอเมริกาจำนวนมากกว่า 800 ราย จึงกำหนดให้ UML (Unified Modeling Language) เป็นมาตรฐานสำหรับการจัดทำตัวแบบระบบของการวิเคราะห์และออกแบบเชิงวัตถุ ตัวอย่างแผนภาพของ UML ที่ใช้เป็นเทคนิคในการจัดทำตัวแบบระบบสารสนเทศ ประกอบด้วย Use Case Diagram, Class Diagram, Object Diagram, State Transition Diagram หรือ State Diagram, Sequence Diagram, และ Component Diagram (รายละเอียดแต่ละเทคนิคศึกษาได้จาก Hoffer et al. (2002);

Whitten et al. (2001); Kendall and Kendall (2002); Satzinger et al. (2000))

2.2.3 เครื่องมือเนื่องจากมีกิจกรรมที่ต้องจัดทำในขั้นตอนต่างๆ ของกระบวนการพัฒนาระบบงานสารสนเทศเป็นจำนวนมาก (ไม่ว่าจะเลือกใช้ระเบียบวิธีการพัฒนาวิธีใดก็ตาม) ฉะนั้นถ้าไม่มีการนำเครื่องมือใดมาช่วยทำงาน การทำกิจกรรมแต่ละกิจกรรม (โดยเฉพาะกิจกรรมการจัดทำตัวแบบระบบโดยใช้เทคนิคแผนภาพต่างๆ) ให้แล้วเสร็จต้องใช้ระยะเวลานาน ซึ่งอาจส่งผลให้ได้ระบบเพื่อนำไปใช้งานช้ากว่ากำหนด เครื่องมือที่นิยมนำมาใช้ในกระบวนการพัฒนาระบบ มีดังนี้

2.2.3.1 CASE (Computer-aided software engineering) เป็นซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นเพื่อใช้สนับสนุนให้การทำกิจกรรมต่างๆ ของกระบวนการพัฒนาระบบสารสนเทศเป็นไปโดยอัตโนมัติ ตัวอย่างของซอฟต์แวร์ที่ใช้สนับสนุนกิจกรรมต่างๆ เช่น Designer 2000 ของ Oracle, Power Designer ของ Powersoft, และ Visible Analyst ของ Visible System เป็นต้น (Hoffer et al., 2002; Whitten et al., 2001)

CASE สามารถจำแนกได้เป็น 3 ประเภทหลัก ดังนี้ *(1) Front-End Case Tool* หมายถึง ซอฟต์แวร์ที่ช่วยให้การทำงานตั้งแต่ขั้นตอนวางแผนจนกระทั่งถึงขั้นตอนการออกแบบให้เป็นไปอย่างอัตโนมัติ *(2) Back-End Case Tool* หมายถึง ซอฟต์แวร์ที่ช่วยให้การทำงานตั้งแต่ขั้นตอน การลงรหัสโปรแกรมจนกระทั่งถึงขั้นการจัดทำเอกสารเป็นไปอย่างอัตโนมัติ *(3) Integrated Case Tool* หรือ *Cross Life Cycle CASE Tool* หมายถึง ซอฟต์แวร์ที่ช่วยให้การทำงานตั้งแต่ขั้นวางแผนจนกระทั่งถึงขั้นการปรับใช้เป็นไปอย่างอัตโนมัติ โดยทั่วไปซอฟต์แวร์ที่ถูกจัดเป็น *CASE ประเภทแรก* จะเป็นโปรแกรมที่สามารถช่วยนักวิเคราะห์ในการจัดทำเทคนิคแผนภาพต่างๆ ที่ใช้ในขั้นตอนวางแผน วิเคราะห์และออกแบบ สร้างฟอร์ม (จอภาพสำหรับให้ผู้ใช้ป้อนและแสดงข้อมูล) และรายงาน ตรวจสอบความถูกต้องและความสมบูรณ์ของข้อกำหนดคุณลักษณะของระบบในแผนภาพ รวมทั้งตรวจสอบความสอดคล้องระหว่างแผนภาพที่จัดทำกับฟอร์มและรายงานให้โดยอัตโนมัติ นอกจากนี้ ยังสามารถจัดทำเอกสารประกอบระบบได้ ส่วน CASE *ประเภทที่สอง* จะเป็นโปรแกรมที่สามารถช่วยโปรแกรมเมอร์ในการสร้างรหัสชุดคำสั่งของโปรแกรมต่างๆ และสร้างรหัสคำนิยามของฐานข้อมูลจากแผนภาพ ฟอร์ม และรายงานต่างๆ ที่นักวิเคราะห์ออกแบบไว้ให้โดยอัตโนมัติ ส่วน *CASE ประเภทสุดท้าย* เป็นโปรแกรมที่นอกจากจะสามารถทำงานต่างๆ ที่ Front-End Case Tool และ Back-End Case Tool ทำได้แล้ว CASE ประเภทนี้ยังมีเรโพสิโทรี (Repository) ที่เก็บรวบรวมสารสนเทศต่างๆ ของระบบ เพื่อนำมาอ้างอิงหรือนำกลับมาใช้งานใหม่หรือบำรุงรักษาต่อไปได้โดยง่าย สารสนเทศเหล่านั้น ได้แก่ โปรแกรม มอดูลต่างๆ คำอธิบายเกี่ยวกับฮาร์ดแวร์ที่เป็นส่วนประกอบของระบบ ข้อกำหนดคุณลักษณะการวิเคราะห์และออกแบบของระบบที่อยู่ในรูปของเทคนิคแผนภาพต่างๆ พร้อมคำอธิบายเกี่ยวกับข้อมูลทั้งในเรื่องโครงสร้างข้อมูล มุมมองข้อมูล (Views) แฟ้มข้อมูล และฐานข้อมูล ความสัมพันธ์ระหว่างโปรแกรมและข้อมูล (หรือมุมมองต่างๆ ของข้อมูล) ในฐานข้อมูล ฟอร์ม รายงานที่ออกแบบ รวมทั้งเอกสารประกอบระบบที่จัดทำ (Whitten et al., 2001; Shelly et al., 1998) เนื่องจากในทางปฏิบัติไม่สามารถแยกขั้นตอนการวิเคราะห์ออกจากขั้นตอนการออกแบบได้อย่างชัดเจนจึงทำให้ในความเป็นจริงไม่สามารถแบ่งแยก Upper CASE และ Lower CASE ออกจากกันได้อย่างเด็ดขาด ดังนั้น CASE ที่มีจำหน่ายในตลาด CASE ของบางบริษัทจึงมีความสามารถบางส่วนของ Upper CASE ตามคำนิยามข้างต้น เช่น สามารถช่วยนักวิเคราะห์ในการจัดทำเทคนิคแผนภาพต่างๆ ที่ใช้ในขั้นตอนการออกแบบระบบ รวมทั้งช่วยออกแบบฟอร์ม และรายงานต่างๆ ในระบบ (แต่ไม่สามารถช่วยการทำงานของนักวิเคราะห์ในขั้นตอนการวางแผน) แต่ในขณะเดียวกันก็มีความสามารถของ Lower CASE ตามคำนิยามข้างต้นด้วย หรือ CASE ของบางบริษัทสามารถ

นำตัวแบบที่อยู่ในรูปของเทคนิคแผนภาพต่างๆ เป็นข้อมูลนำเข้าของโปรแกรม เพื่อนำไปประมวลสร้างเป็นฟอร์มและรายงานให้โดยอัตโนมัติ เป็นต้น

องค์กรหลายองค์กรนำ CASE มาใช้ เนื่องจากเชื่อว่าการใช้ CASE สามารถลดเวลาที่ใช้ในการพัฒนาระบบลง ทำให้สามารถพัฒนาซอฟต์แวร์ได้ในปริมาณที่เพิ่มขึ้น และมีการพัฒนากระบวนการให้เป็นมาตรฐานเดียวกัน ส่งผลให้มีความมั่นใจในคุณภาพของกระบวนการที่ใช้ในการพัฒนาระบบทำให้มีการจัดทำเอกสารประกอบระบบที่สมบูรณ์และมีแหล่งสารสนเทศเกี่ยวกับระบบงานที่พัฒนาอย่างครบถ้วน ส่งผลให้ง่ายต่อการบำรุงรักษาระบบในภายหน้า แต่ในขณะเดียวกันก็มีองค์กรอีกหลายองค์กรที่ไม่ใช้ CASE ในการพัฒนาระบบ เนื่องจากการใช้ CASE มีค่าใช้จ่ายที่สูงมาก เพราะนอกจาก CASE จะมีราคาแพงแล้ว องค์กรยังต้องเสียค่าใช้จ่ายในการอบรมบุคลากรเพื่อให้เกิดการใช้งาน CASE ได้อย่างมีประสิทธิภาพ นอกจากนี้องค์กรยังไม่มีความมั่นใจว่าโปรแกรมที่เป็นผลลัพธ์ที่ได้จาก CASE จะมีความถูกต้อง (Hoffer et al., 2002)

2.2.3.2 Development Tools **หรือ** Rapid Application Development Tool เป็นซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นเพื่อช่วยให้การสร้างโปรแกรมสามารถทำได้ง่ายและเร็วขึ้น โดยซอฟต์แวร์เหล่านี้จะมีสิ่งอำนวยความสะดวกต่างๆ ให้ผู้พัฒนาใช้ในการสร้างส่วนต่อประสานกับผู้ใช้ (User Interface) ได้โดยง่ายและรวดเร็ว เช่น มีเมนูเอดดิเตอร์ (Menu Editor) สำหรับสร้างเมนูของระบบ มีอ็อบเจ็กต์ต่างๆ (ที่คาดว่าจะเป็นส่วนประกอบของระบบสารสนเทศทุกๆ ระบบ) เตรียมไว้ให้ผู้พัฒนาระบบดึงมาเป็นส่วนประกอบของระบบสารสนเทศที่พัฒนาได้ เช่น เลือกวินโดว์ ปุ่ม (Button) สำหรับบันทึกและยกเลิก ช่องกรอก/แสดงข้อมูลมาประกอบกันเพื่อสร้างจอภาพที่ใช้เป็นฟอร์มสำหรับป้อนหรือปรับปรุงข้อมูล เป็นต้น ผู้พัฒนาสามารถเลือกว่าจะให้อ็อบเจ็กต์ใดบนจอภาพสัมพันธ์กับข้อมูลหรือมุมมองของข้อมูลใดในฐานข้อมูล โดยหลังจากที่ผู้พัฒนาออกแบบผัง (Layout) และกำหนดรายละเอียดการเชื่อมโยงอ็อบเจ็กต์ในจอภาพกับฐานข้อมูลแล้ว เครื่องมือประเภทนี้จะแปลงจอภาพที่ออกแบบเป็นโปรแกรมมอดูลให้โดยอัตโนมัติ ซึ่งจะทำให้ผู้พัฒนาสามารถดูรูปแบบและการทำงานของหน้าจอที่สร้างได้ทันที ในทำนองเดียวกับการสร้างจอภาพ อนึ่ง ผู้พัฒนาสามารถใช้สิ่งอำนวยความสะดวกในเครื่องมือประเภทนี้เพื่อสร้างผังของรายงาน และรายการฟิลด์ต่างๆ ในรายงานโดยกำหนดว่าฟิลด์ใดเชื่อมโยงกับข้อมูลใด จากนั้นเครื่องมือประเภทนี้จะใช้ผังรายงานที่ออกแบบและรายละเอียดต่างๆ ที่ผู้พัฒนากำหนดแปลงเป็นโปรแกรมมอดูลสำหรับการจัดทำรายงานให้โดยอัตโนมัติ โดยผู้พัฒนาสามารถดูรูปแบบและเนื้อหาของรายงานที่สร้างได้ทันทีเหมือนกับที่จะปรากฏในระบบงานจริง ดังนั้น เครื่องมือประเภทนี้จึงมีบทบาทอย่างมากกับระเบียบวิธีการพัฒนาแบบ RAD ตัวอย่างซอฟต์แวร์ต่างๆ เช่น Developer ของ Oracle, Power Builder ของ Powersoft, Visual BASIC และ Visual Studio ของ Microsoft, และ Dephi ของ Borland International เป็นต้น (Hoffer et al., 2002)

2.2.3.3 Normalization Analysis Tool เป็นซอฟต์แวร์ ที่ได้รับการพัฒนาขึ้นเพื่อแปลงแผนภาพความสัมพันธ์ระหว่างข้อมูล (ERD) เป็นตาราง (Tables) ต่างๆ ของฐานข้อมูลเชิงสัมพันธ์ ตามกฎของการทำให้ข้อมูลเป็นบรรทัดฐาน (Normalization) เพื่อลดปัญหาที่อาจเกิดจากการปรับปรุงข้อมูล นอกจากนี้ เครื่องมือนี้ยังสามารถตรวจวิเคราะห์ความเป็นบรรทัดฐานของฐานข้อมูลเชิงกายภาพที่ถูกออกแบบไว้ว่ามีความถูกต้องตามกฎเกณฑ์เพียงใด เพื่อให้แน่ใจว่าฐานข้อมูลเชิงกายภาพที่ออกแบบจะไม่ประสบกับปัญหาในการปรับปรุงข้อมูล

นอกจากเครื่องมือที่กล่าวแล้ว ในทางปฏิบัติการพัฒนาระบบงานยังนำซอฟต์แวร์ประยุกต์ประเภทอื่นๆ ที่ถูกพัฒนาใช้งานด้วยซอฟต์แวร์เหล่านั้น ได้แก่ *(1) Project Management Aid Tool* เป็นซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นเพื่อใช้สำหรับการบริหารโครงการ ซอฟต์แวร์ประเภทนี้ช่วยในการจัดทำ PERT, Gantt Chart, ปฏิทิน, ตารางการใช้งานทรัพยากรต่างๆ ของโครงการ ติดตามค่าใช้จ่ายต่างๆ ที่เกิดกับโครงการ และการวิเคราะห์ต้นทุน-ผลประโยชน์ของโครงการ ตัวอย่างซอฟต์แวร์ประเภทนี้ เช่น Microsoft Project เป็นต้น *(2) Human Resource Allocation* เป็นซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นเพื่อใช้ในการจัดสรรและบริการบุคลากร และ *(3) Document Management System* เป็นซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นเพื่อให้การจัดเก็บและเรียกใช้เอกสารเป็นไปอย่างมีประสิทธิภาพ

2.2.4 ภาษาคอมพิวเตอร์ การสร้างมอดูล (หรือชุดรหัสคำสั่ง) แต่ละมอดูลของโปรแกรมต่างๆ ในระบบสารสนเทศอาจเลือกใช้ภาษาคอมพิวเตอร์ใดภาษาหนึ่ง ดังต่อไปนี้

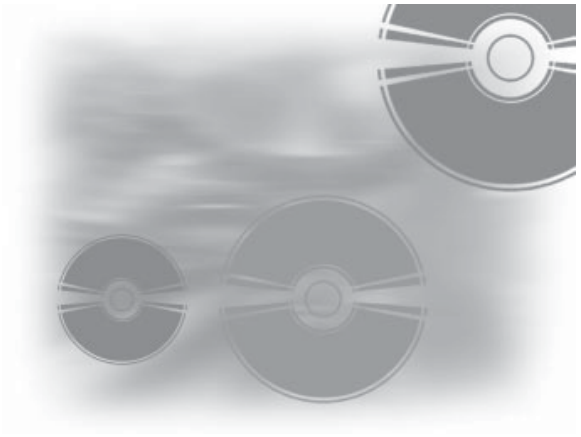
2.2.4.1 Procedural Language เป็นภาษาคอมพิวเตอร์ยุคที่ 3 ซึ่งผู้เขียนโปรแกรมเป็นผู้กำหนดขั้นตอน

วิธีต่างๆ (Algorithms) ของการทำงานในแต่ละมอดูลเองทั้งหมด ดังนั้นการใช้ภาษาประเภทนี้นอกจากผู้เขียนโปรแกรมจะต้องบอกว่าทำอะไรแล้ว ยังต้องบอกด้วยว่าทำอย่างไรจึงจะได้ผลลัพธ์นั้น (เช่น ต้องนำข้อมูลใดมาเป็นข้อมูลนำเข้า และจะจัดดำเนินการกับข้อมูลนำเข้า อย่างไรเพื่อให้ได้ผลลัพธ์ที่ต้องการ) ภาษาที่นำมาใช้งานกันโดยทั่วไป ได้แก่ ภาษา Basic, COBOL, C, และ FORTRAN (Shelly et al., 2003; Oz, 2002)

2.2.4.2 Object-oriented programming language เป็นภาษาคอมพิวเตอร์ที่ถูกพัฒนาขึ้นเพื่อให้สามารถประยุกต์ใช้แนวคิดเชิงวัตถุกับการสร้างโปรแกรมของระบบสารสนเทศได้ ภาษาคอมพิวเตอร์กลุ่มนี้ช่วยให้โปรแกรมเมอร์สร้างและจัดดำเนินการ (เช่น ปรับปรุง) กับอ็อบเจ็กต์ต่างๆ ที่มีในระบบได้ โดยทั่วไปภาษาประเภทนี้เป็น Event driven โปรแกรม กล่าวคือ โปรแกรมจะโต้ตอบอย่างไรเมื่อมีเหตุการณ์ต่างๆ เกิดขึ้น เช่น เมื่อผู้ใช้ระบบกดปุ่ม F1 บนแป้นพิมพ์ (Keyboard) ภาษาประเภทนี้ได้แก่ ภาษา C++ และ JAVA

2.2.4.3 Visual Programming Language เป็นภาษาที่ให้โปรแกรมเมอร์สามารถสร้างรหัสชุดคำสั่งต้นฉบับ (Source code) ของโปรแกรมต่างๆ ในภาษาคอมพิวเตอร์บางภาษาโดยผ่านส่วนต่อประสานกราฟิกที่โปรแกรมเมอร์สามารถเห็นภาพได้ (ตัวอย่างเช่น ถ้าโปรแกรมเมอร์ต้องการเขียนปุ่มบันทึก (Save Button) ไว้ตรงตำแหน่งมุมขวาด้านล่างจอภาพด้วยภาษา C++ แทนที่โปรแกรมเมอร์จะต้องเขียนประโยคคำสั่งของภาษา C++ เพื่อสร้างปุ่มและกำหนดตำแหน่งของปุ่มบันทึกที่ต้องการ โปรแกรมเมอร์เพียงแต่เลือกสัญรูป (Icon) ปุ่มในแถบเครื่องมือแล้วนำมาวางในตำแหน่งที่ต้องการก็จะได้ประโยคคำสั่งของภาษา C++ เพื่อสร้างปุ่มและกำหนดตำแหน่งของปุ่มนั้นโดยอัตโนมัติ ดังนั้นจึงอาจกล่าวได้ว่าภาษานี้มีคุณสมบัติที่ใกล้เคียงกับ Development tools ที่กล่าวแล้วในข้างต้น ภาษาประเภทนี้ที่มีการใช้งานกันโดยทั่วไป ได้แก่ Visual Studio, Power Builder, Delphi Power Builder, Visual Basic และ Visual Aid for JAVA

2.2.4.4 Web Page Development เป็นภาษาที่ถูกพัฒนาขึ้นเพื่อใช้สร้างหน้าเว็บของระบบสารสนเทศที่ทำงานบนข่ายงานอินเทอร์เน็ตภาษาประเภทนี้ที่มีการใช้งานกันโดยทั่วไป ได้แก่ ASP/ASP.net, HTML, WML, XHTML, XML และ PHP



2.2.5 การเลือกใช้ระเบียบวิธีการพัฒนาระบบและเทคนิคการพัฒนาระบบ

2.2.5.1 ระเบียบวิธีการพัฒนาระบบ Cabham พบว่า วิธีการพัฒนาระบบที่มีการใช้งานมากที่สุดได้แก่ วิธีการพัฒนาระบบเชิงโครงสร้าง (42% ของโครงการพัฒนาระบบสารสนเทศที่สำรวจ) รองลงมา ได้แก่ วิธีการพัฒนาระบบเชิงวัตถุ (20%) ส่วนวิธีการ RAD มีการนำไปใช้ประมาณ 8.5% ของโครงการพัฒนาระบบสารสนเทศที่สำรวจ (Curtis and Cabham, 2002)

ปัจจุบันมีแนวโน้มที่ผู้ใช้ต้องการได้ระบบสารสนเทศมาใช้งานภายในเวลาที่รวดเร็ว ในขณะที่ระบบสารสนเทศมีขนาดใหญ่ มีโครงสร้าง องค์ประกอบและการทำงานที่สลับซับซ้อนมากขึ้น ประกอบกับซอฟต์แวร์ที่ใช้เป็นเครื่องมือพัฒนาระบบให้เป็นอัตโนมัติมีความสามารถมากขึ้น ทำให้ผู้พัฒนาไม่เห็นความจำเป็นของการแบ่งแยกขั้นตอนการทำงานในกระบวนการพัฒนาระบบให้ชัดเจน ระเบียบวิธีการพัฒนาแบบ Waterfall ผนวกกับวิธีการพัฒนาระบบเชิงโครงสร้างจึงไม่เพียงพอที่จะนำมาใช้ในการพัฒนาระบบสารสนเทศภายใต้สภาพแวดล้อมใหม่ได้ ซึ่งต่างจากระเบียบวิธีการพัฒนาแบบ RAD และระเบียบวิธีการวิเคราะห์และออกแบบระบบเชิงวัตถุ ที่พัฒนามาจากแนวคิดที่สอดคล้องกับสภาพแวดล้อมของระบบสารสนเทศในปัจจุบันและในอนาคต

การที่หน่วยงานในองค์กรหรือบริษัทผู้ผลิตซอฟต์แวร์จะเลือกระเบียบวิธีการพัฒนาใดสำหรับพัฒนาระบบสารสนเทศใดนั้นขึ้นอยู่กับชนิดของระบบที่จะพัฒนา (Olive, 1983) ซึ่งโดยทั่วไปแล้วหากใช้เกณฑ์งานที่ระบบสารสนเทศให้การสนับสนุน

สามารถแบ่งระบบสารสนเทศออกได้เป็น 4 ชนิด คือ Transaction Processing หรือ TPS, Report/Inquiry Systems, Decision Support Systems/Executive Information Systems (DSS/EIS), และ Expert Systems (รายละเอียดศึกษาได้จาก Hoffer et al. (2002); Laudon and Laudon (1998/2003))

อนึ่ง องค์ประกอบและคุณลักษณะ (Characteristics) ที่แตกต่างกันของระบบเหล่านี้จะส่งผลกระทบต่อการเลือกใช้ระเบียบวิธีการพัฒนาที่แตกต่างกัน (Laudon and Laudon, 2003; Anderson and Post, 2000) เช่น TPS ซึ่งเป็นระบบที่เกี่ยวข้องกับการปฏิบัติงานประจำวันขององค์กร เป็นระบบขนาดใหญ่ และความผิดพลาดของระบบจะส่งผลอย่างมากต่อการดำเนินงานขององค์กร ดังนั้น การพัฒนา TPS มักเลือกใช้ระเบียบวิธีการพัฒนาแบบ SDLC นอกจากนี้ SDLC ยังนำมาใช้พิจารณาเพื่อขยายระบบงานเดิมหรือนำระบบเดิมมาพัฒนาให้สอดคล้องกับเทคโนโลยีใหม่ เนื่องจากมีกระบวนงานและระเบียบวิธีการปฏิบัติงานที่แน่นอนแล้ว ซึ่งส่งผลให้ข้อกำหนดความต้องการระบบมีความชัดเจน แต่ในกรณีที่เป็นระบบใหม่อาจเลือกใช้ SDLC ผสมกับการสร้างระบบต้นแบบเนื่องจากข้อกำหนดความต้องการยังไม่สามารถทราบได้แน่ชัด อนึ่ง TPS ที่ทำงานแบบ Online Real Time ซึ่งให้ความสำคัญกับส่วนต่อประสานกับผู้ใช้ อาจเลือกวิธีการสร้างระบบต้นแบบเป็นส่วนเสริมของการเลือกใช้ SDLC ในทางตรงกันข้ามระบบ DSS ที่พัฒนาเพื่อใช้เฉพาะกับผู้บริหารบางรายจะเป็นระบบขนาดเล็ก ในขณะที่ข้อกำหนดความต้องการระบบไม่แน่ชัด ซึ่งส่งผลอย่างมากต่อประสิทธิผลการทำงานของระบบ ดังนั้น การพัฒนา DSS จึงมักใช้ระเบียบวิธีการพัฒนาแบบ RAD เป็นส่วนมาก (Anderson and Post, 2000)

2.2.5.2 เทคนิคการพัฒนาระบบ เนื่องจากเทคนิคการพัฒนาระบบมีเป็นจำนวนมาก (ดังที่กล่าวแล้วข้างต้น) การพัฒนาระบบสารสนเทศระบบใดระบบหนึ่ง นักวิเคราะห์และออกแบบระบบไม่สามารถใช้เทคนิคทุกเทคนิคได้ ดังนั้น โดยทั่วไปนักวิเคราะห์และออกแบบระบบจะเลือกใช้บางเทคนิคเท่านั้น

จากการศึกษาของ Callaghan and Leigh พบว่า เทคนิคการพัฒนาระบบที่มีการใช้งานมาก 5 อันดับแรกในขั้นการวิเคราะห์ ได้แก่ Entity-Relationship (ER), Data flow diagram (DFD), Flow Chart, Normalization และ Class/Object Diagram ส่วนเทคนิคการพัฒนาระบบที่มีการใช้งานมาก 5 อันดับแรกในขั้นการออกแบบ ได้แก่ Entity-Relationship (ER), Data flow diagram (DFD), Flow Chart, Structured English และ Normalization (Curtis and Cabham, 2002)

2.3 การพัฒนาระบบกับความสำเร็จของระบบสารสนเทศ

ความสำเร็จของการพัฒนาระบบงานสามารถวัดได้จากความสำเร็จของการบริหารโครงการซึ่งโดยทั่วไปมีตัวชี้วัดที่สำคัญ 2 ตัว คือ เวลาและงบประมาณที่ใช้ในโครงการ ดังนั้น หากสามารถควบคุมให้เวลาและงบประมาณที่ใช้ในการพัฒนาระบบเป็นไปตามแผนงานที่กำหนด ในมิติหนึ่งอาจกล่าวได้ว่าการพัฒนาระบบงานนั้นประสบผลสำเร็จ (Saarinen, 1996) นอกจากตัวชี้วัดในเรื่องการบริหารโครงการแล้ว ความสำเร็จของการพัฒนาระบบสามารถวัดจาก

(1) คุณภาพของระบบสารสนเทศซึ่งเป็นผลลัพธ์ของกระบวนการพัฒนาระบบ โดยตัวชี้วัดคุณภาพของระบบ ได้แก่ ความพึงพอใจของผู้ใช้ (ระบบสามารถทำงานบรรลุความต้องการของผู้ใช้หรือไม่) และความถูกต้องของระบบ (ระบบที่พัฒนาได้ไม่มีข้อผิดพลาดในการทำงานและในสารสนเทศที่ผลิตให้กับผู้ใช้) (DeLone and McLean, 1992)

(2) ความสามารถในการบำรุงรักษาระบบ (ระบบที่พัฒนาง่ายต่อการปรับปรุงและบำรุงรักษา โดยดูที่ความครบถ้วนของเอกสารประกอบระบบ เนื้อหาในเอกสารประกอบระบบสามารถให้รายละเอียดเกี่ยวกับวิธีการทำงานของระบบ วิธีการใช้ระบบ วิธีการให้การสนับสนุนกับผู้ใช้ระบบ และวิธีการบำรุงรักษาระบบได้มากน้อยเพียงใด)

(3) ความสำเร็จของการติดต่อสื่อสารระหว่างนักวิเคราะห์และผู้ที่เกี่ยวข้องกับโครงการพัฒนาระบบ ได้แก่ ผู้บริหาร ผู้ใช้ เจ้าหน้าที่เทคนิค (Liebowiz, 1999)

2.4 ทักษะทีมพัฒนากับความสำเร็จในหน้าที่การทำงาน

ทักษะในการติดต่อสื่อสารของทีมพัฒนาระบบส่งผลต่อความสำเร็จของกระบวนการพัฒนาระบบ ซึ่งเป็นปัจจัยที่ทำให้ระบบสารสนเทศประสบความสำเร็จ (Liebowitz, 1999; Saarinen, 1996) ทักษะทางด้านการสื่อสารหมายรวมถึง ทักษะทั้งทางด้านการเขียน การพูด และการทำงานเป็นทีม (Hoffer et al., 2002) นอกจากทักษะด้านการสื่อสารแล้ว นักวิเคราะห์และออกแบบระบบยังต้องมีทักษะที่จะช่วยให้ประสบความสำเร็จในการทำงานในด้านต่างๆ ดังนี้

(1) ทักษะด้านเทคนิค ได้แก่ ความรู้ความชำนาญในด้านเทคโนโลยีต่างๆ เช่น ระบบเครือข่าย ฐานข้อมูลและระบบบริหารจัดการฐานข้อมูล การเขียนโปรแกรม การใช้ระเบียบวิธีการพัฒนา เทคนิคและเครื่องมือสำหรับพัฒนาระบบ (Hoffer et al., 2002; Vitalari, 1985) การศึกษาของ Trauth et al. (1993) แสดงให้เห็นว่า ความชำนาญด้านเทคนิคที่จำเป็นสำหรับการทำงาน 3 กลุ่ม ได้แก่ การสื่อสารข้อมูลและระบบเครือข่าย การเข้าถึงข้อมูลและการบริหารจัดการข้อมูล ระบบสนับสนุนการตัดสินใจและ CASE (Trauth et al., 1993) นอกจากนี้ ประเทศไต้หวันและสหรัฐอเมริกาให้ความสำคัญกับความชำนาญด้านเทคนิคที่สำคัญ ได้แก่ การวิเคราะห์และออกแบบระบบ และการสื่อสารข้อมูลและระบบเครือข่าย (Gupta et al., 1994)

(2) ทักษะด้านการบริหาร ได้แก่ การบริหารโครงการ การบริหารทรัพยากร และการบริหารความเสี่ยง (Hoffer et al., 2002)

(3) ทักษะการคิดอย่างเป็นระบบ ซึ่งหมายรวมถึง ความรู้ความชำนาญเกี่ยวกับองค์กรเจ้าของระบบสารสนเทศที่จะพัฒนาในด้านต่างๆ ได้แก่ โครงสร้างองค์กร หน้าที่การทำงานของหน่วยงานในองค์กร คุณลักษณะขององค์กร สภาพแวดล้อมขององค์กร (คู่แข่ง ลูกค้า และอื่นๆ) วัฒนธรรมและการเมืองภายในองค์กร นโยบายองค์กร สไตล์การบริหารงานของผู้บริหาร (Hoffer et al., 2002; Vitalari, 1985) การวิเคราะห์เพื่อระบุปัญหาและความสามารถในการแก้ปัญหา

ในระหว่างทักษะทั้ง 4 ด้านข้างต้น ถ้าต้องการให้ระบบงานที่พัฒนาประสบความสำเร็จ (ซึ่งก็หมายถึงความสำเร็จของนักวิเคราะห์ด้วย) นักวิเคราะห์ควรให้ให้ความสำคัญกับทักษะเหล่านี้เท่าๆ กัน หรือควรให้ความสำคัญกับทักษะในด้านใดเป็นพิเศษ การศึกษาของ Gupta et al. (1994) ได้ข้อสรุปว่าทักษะด้านเทคนิคมีความสำคัญน้อยกว่าทักษะในด้านอื่นๆ ซึ่งสอดคล้องกับการศึกษาของ Shrout, Henry, Arvey และ Hoyle, Khan และ Kukalis, Hunte และ Benbasat et al. ที่เคยทำในอดีต (Vitalari, 1985) และการศึกษาของ Gupta et al. (1994) แต่จากการศึกษาของ Vitalari (1985) สรุปว่า นักวิเคราะห์จะใช้ทักษะในด้านใดมากหรือน้อยขึ้นอยู่กับงานที่ทำและขั้นตอนการพัฒนาระบบงาน นั่นหมายความว่านักวิเคราะห์จะให้ความสำคัญกับทักษะใดมากหรือน้อยอย่างไรขึ้นอยู่กับ ณ ขณะนั้นนักวิเคราะห์กำลังทำงานอะไรในขั้นตอนใดของกระบวนการพัฒนาระบบงาน (Vitalari, 1985)

นอกจากนี้ มีการศึกษาที่พบว่า ความรับผิดชอบ ความใส่ใจ ความตั้งใจจริง และความกระตือรือร้น ของบุคลากรในหน่วยงานระบบสารสนเทศในการให้บริการกับผู้ใช้ส่งผลต่อการใช้งานและความพึงพอใจของผู้ใช้ ซึ่งเป็นปัจจัยที่ส่งผลต่อความสำเร็จของระบบงานสารสนเทศ (Pett et al., 1995) และเนื่องจากระบบสารสนเทศส่งผลกระทบอย่างมากต่อการทำงานขององค์กรและผู้ที่มีส่วนได้เสียกับองค์กร (พนักงาน ลูกค้า ผู้ถือหุ้น เป็นต้น) การทำงานโดยขาดจรรยาบรรณของบุคลากรด้านระบบสารสนเทศสามารถส่งผลกระทบต่อคนหลายคน (Laudon and Laudon, 1998; Bocij et al., 1999) ดังนั้น นักวิเคราะห์ที่จะประสบความสำเร็จจึงควรมีคุณลักษณะเหล่านี้รวมทั้งเป็นผู้ที่มีจริยธรรมในการทำงานด้วย

3. วิธีดำเนินงานวิจัย

3.1 การเก็บข้อมูลและกลุ่มตัวอย่าง

งานวิจัยนี้ใช้ระเบียบวิธีวิจัยเชิงสำรวจ (Survey) ซึ่งจะจัดทำกับนายจ้างที่เป็นบริษัทเอกชนและรัฐวิสาหกิจ (ไม่เน้นที่กิจการขนาดกลางและขนาดเล็ก) เหตุผลที่เลือกสำรวจเฉพาะบริษัทเอกชนและรัฐวิสาหกิจ เนื่องจากเป้าหมายหลักของภาควิชามุ่งที่จะผลิตบัณฑิตป้อนสู่ตลาดแรงงานในภาคเอกชนและรัฐวิสาหกิจ อนึ่ง การจัดเก็บข้อมูลสำหรับงานวิจัยกระทำในช่วงเดือนกุมภาพันธ์ถึงมีนาคม 2547 แบบสอบถามที่เกี่ยวข้องกับการสำรวจความต้องการด้านเนื้อหาของวิชาการวิเคราะห์และออกแบบระบบสารสนเทศถูกจัดส่งไปยัง *(1) หน่วยงานเอกชน* ตามรายชื่อที่ปรากฏในสรุปข้อสนเทศบริษัทจดทะเบียน ตลาดหลักทรัพย์แห่งประเทศไทย 2544 โดยคัดเลือกบริษัทขนาดใหญ่จำนวนทั้งสิ้น 209 บริษัท (จากจำนวนทั้งหมด 360 บริษัท) ซึ่งประกอบด้วยธนาคารพาณิชย์ รัฐวิสาหกิจ สถาบันการเงินและบริษัทประกันภัย ซึ่งเป็นธุรกิจอุตสาหกรรมการผลิต และให้บริการที่ไม่รวมบริษัทที่ให้บริการด้านคอมพิวเตอร์ซอฟท์แวร์ (Software House) และรัฐวิสาหกิจ *(2) บริษัท Software House* ซึ่งเป็นหน่วยงานเอกชนที่ให้บริการด้านคอมพิวเตอร์ซอฟท์แวร์ตามรายชื่อที่ปรากฏใน Computer Software Directory ที่จัดทำโดยกรมส่งเสริมการส่งออก กระทรวงพาณิชย์ ปี พ.ศ. 2544 จำนวนทั้งสิ้น 149 บริษัท *(3) รัฐวิสาหกิจ* ที่เป็นสมาชิกในชมรมเทคโนโลยีสารสนเทศรัฐวิสาหกิจแห่งประเทศไทย [(IT State Enterprise Club of Thailand (ITSEC))] ปี พ.ศ. 2544 จำนวน 46 บริษัท และ *(4) บริษัทนอก*

ตลาดหลักทรัพย์แห่งประเทศไทย จำนวน 38 บริษัท รวมแบบสอบถามที่จะจัดส่งทั้งหมดสำหรับงานวิจัยนี้เท่ากับ 442 บริษัท อนึ่ง บริษัทที่จัดส่งแบบสอบถามต้องเป็นบริษัทที่มีผู้รับผิดชอบด้านการพัฒนาระบบสารสนเทศให้กับหน่วยงานของตนเอง หรือมีการพัฒนาระบบสารสนเทศเพื่อจำหน่ายเท่านั้น ไม่รวมบริษัทที่ให้บริการด้านการจัดหาโปรแกรมสำเร็จรูปเพื่อนำไปติดตั้งให้กับบริษัทลูกค้า ในการจัดส่งแบบสอบถามนี้ งานวิจัยคัดเลือกบริษัทที่จัดส่งแบบสอบถามด้วยวิธีการสุ่ม

ในการจัดส่งแบบสอบถามสำหรับงานวิจัยนี้กระทำ 2 รอบ รอบแรกเป็นการจัดส่งแบบสอบถามไปยังบริษัทในตลาดหลักทรัพย์แห่งประเทศไทย บริษัท Software House ซึ่งมีชื่ออยู่ใน Computer Software Directory และรัฐวิสาหกิจที่ปรากฏชื่อในชมรมเทคโนโลยีสารสนเทศรัฐวิสาหกิจแห่งประเทศไทย จดหมายที่ตีกลับเนื่องจากกิจการบางแห่งปิดกิจการจำนวน 38 ชุด ถูกจัดส่งไปใหม่ให้กับหน่วยงานเอกชนที่อยู่นอกตลาดหลักทรัพย์แห่งประเทศไทยซึ่งเป็นกิจการขนาดใหญ่และมีหน่วยงานด้านคอมพิวเตอร์ อนึ่ง เมื่อสิ้นสุดเวลาตอบกลับแบบสอบถาม ผู้วิจัยได้โทรศัพท์ทวงถามแบบสอบถามที่ยังไม่ส่งกลับคืนด้วย จากแบบสอบถามจำนวน 442 ชุด มีแบบสอบถามตอบกลับมาจำนวน 57 ชุด คิดเป็นร้อยละ 12.90 ในจำนวนนี้เป็นแบบสอบถามจากบริษัท Software House ซึ่งมีชื่ออยู่ใน Computer Software Directory ซึ่งจัดทำโดยกรมส่งเสริมการส่งออก กระทรวงพาณิชย์จำนวน 14 แห่ง และบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทย และชมรมเทคโนโลยีสารสนเทศรัฐวิสาหกิจแห่งประเทศไทยจำนวน 43 แห่ง แบบสอบถามที่ตอบกลับมาทั้งหมดถูกนำมาใช้ในการวิเคราะห์ผลของการวิจัยในขั้นต่อไป

3.2 เครื่องมือที่ใช้ในงานวิจัย

ในการดำเนินงานวิจัย ผู้ทำการวิจัยจะพัฒนาแบบสอบถามจากแบบสอบถามของ Gupta et al. (1994) และ Ching et al. (2000) โดยหลังจากจัดทำแบบสอบถามเรียบร้อยแล้ว แบบสอบถามดังกล่าวจะนำไปทดสอบความเหมาะสมเบื้องต้น (Pre-test) เพื่อทำการปรับปรุงแก้ไขก่อนจัดส่งแบบสอบถามไปยังหน่วยงานเอกชนต่างๆ ที่คัดเลือกในกลุ่มตัวอย่างข้างต้น

เนื่องจากงานวิจัยนี้เป็นงานวิจัยเชิงปริมาณ จึงใช้แบบสอบถามเป็นเครื่องมือที่ใช้ในงานวิจัย และเพื่อให้แบบสอบถามครอบคลุมวัตถุประสงค์ และปัจจัยทั้งหมดที่ต้องการศึกษา จึงแบ่งแบบสอบถามออกเป็น 4 ส่วนดังนี้

ส่วนที่ 1 ข้อมูลทั่วไปเกี่ยวกับผู้ตอบแบบสอบถาม ได้แก่ ตำแหน่ง ประสบการณ์การทำงาน และส่วนงานที่ทำงาน คำถามในส่วนนี้มีจำนวนทั้งสิ้น 3 ข้อ

ส่วนที่ 2 ข้อมูลทั่วไปเกี่ยวกับกิจการ ได้แก่ ประเภทธุรกิจ รายได้เฉลี่ย กิจกรรมหลักของกิจการ จำนวนพนักงาน จำนวนของทรัพย์สินถาวรด้านคอมพิวเตอร์ที่มีไว้ใช้ในการทำงาน วิธีการได้มาซึ่งซอฟต์แวร์ คุณสมบัติขั้นต่ำของพนักงานพัฒนาระบบงานคอมพิวเตอร์ งบประมาณเกี่ยวกับเทคโนโลยีสารสนเทศ คำถามในส่วนนี้มีจำนวนทั้งสิ้น 11 ข้อ

ส่วนที่ 3 ข้อมูลเกี่ยวกับการพัฒนาระบบ ได้แก่ วิธีที่องค์กรหรือส่วนงานด้านคอมพิวเตอร์ใช้พัฒนาระบบงานคอมพิวเตอร์ เปอร์เซ็นต์ของระยะเวลาที่จัดสรรในการพัฒนาระบบโดยใช้วิธี System Development Life Cycle (SDLC) เปอร์เซ็นต์การใช้เทคนิคการพัฒนาระบบขององค์การหรือส่วนงานด้านคอมพิวเตอร์ โปรแกรมหรือเครื่องมือที่กิจการนำมาใช้สนับสนุนการทำงานในขั้นการวิเคราะห์และออกแบบระบบให้เป็นไปโดยอัตโนมัติ ความถี่ของการใช้ภาษาคอมพิวเตอร์สำหรับพัฒนา/บำรุงรักษาระบบงานคอมพิวเตอร์ของกิจการ ชนิดและลักษณะของระบบที่ใช้วิธีการจัดสร้าง Prototype ประโยชน์ที่กิจการได้รับจากการทำ Prototype ปัญหาของการทำ Prototype สิ่งที่แสดงความสำเร็จในการพัฒนาระบบงานโดยใช้วิธีพัฒนาของกิจการในปัจจุบัน คำถามในส่วนนี้มีจำนวนทั้งสิ้น 9 ข้อ

ส่วนที่ 4 ข้อมูลเกี่ยวกับโปรแกรมการศึกษาด้านระบบสารสนเทศเพื่อการจัดการ เป็นคำถามที่ต้องการทราบถึงลำดับความสำคัญของทักษะที่กิจการต้องการจากพนักงาน และข้อเสนอแนะอื่นๆ คำถามในส่วนนี้มีจำนวนทั้งสิ้น 2 ข้อ

3.3 การประเมินความตรงและความเที่ยงของเครื่องมือ

แบบสอบถามดังกล่าวข้างต้นได้ถูกจัดส่งไปยังผู้ทรงคุณวุฒิซึ่งเป็นเจ้าของกิจการผลิตและจำหน่ายซอฟต์แวร์จำนวน 1 ท่าน เพื่อขอความเห็นเกี่ยวกับความครอบคลุมของเนื้อหา ลำดับของคำถาม ความเข้าใจในคำถามและความง่ายต่อการตอบคำถามในแบบสอบถามก่อนนำออกใช้จริง หลังจากแก้ไขแบบสอบถามแล้ว ผู้วิจัยได้จัดส่งแบบสอบถามไปยังกลุ่มเป้าหมายดังกล่าวข้างต้น

3.4 การวิเคราะห์ข้อมูล

ข้อมูลจากแบบสอบถามที่จัดส่งกลับมา ถูกนำไปประมวลผลทางสถิติโดยใช้โปรแกรมสำเร็จรูปทางสถิติ SPSS อนึ่งก่อนการนำข้อมูลไปประมวลผลทางสถิติ ผู้วิจัยได้ตรวจสอบความถูกต้องของการแปลงข้อมูลจากแบบสอบถามให้อยู่ในรูปของข้อมูลทางคอมพิวเตอร์ก่อนนำไปประมวลผลทางสถิติ โดยเปรียบเทียบข้อมูลที่ป้อนกับแบบสอบถาม ต่อจากนั้นจัดทำสถิติโดยใช้ค่าร้อยละ ค่าเฉลี่ย และตารางไขว้ (Cross-Tabulation) เพื่อแสดงความสัมพันธ์ระหว่างข้อมูล ในการวิเคราะห์จะแยกวิเคราะห์ข้อมูลของบริษัท Software House และบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจออกจากกัน เพื่อแสดงให้เห็นถึงเนื้อหาของวิชาการวิเคราะห์และออกแบบระบบงานคอมพิวเตอร์ (รวมถึงระเบียบวิธีการพัฒนา เทคนิค และเครื่องมือที่ใช้สำหรับการวิเคราะห์และออกแบบระบบสารสนเทศ) ที่หน่วยงานต่างๆ ต้องการ

4. ผลการวิจัย

4.1 ลักษณะทางประชากรศาสตร์ของกลุ่มตัวอย่าง

โดยสรุป ผู้ตอบแบบสอบถามของบริษัท Software House ส่วนใหญ่เป็นกรรมการผู้จัดการ ด้านการบริหารและมีประสบการณ์การทำงานระหว่าง 5-9 ปี อนึ่ง บริษัท Software House ส่วนใหญ่ที่ตอบแบบสอบถามเป็นกิจการที่ให้บริการด้านการพัฒนาระบบงานให้กับหน่วยงานธุรกิจต่างๆ หรือผลิตซอฟต์แวร์เพื่อจำหน่าย ซึ่งกิจการส่วนใหญ่มีรายได้ต่อปีประมาณ 20 ล้านบาท ในด้านจำนวนพนักงานบริษัท Software House ที่ตอบแบบสอบถามส่วนใหญ่เป็นกิจการขนาดเล็กที่มีพนักงานประมาณ 10 ถึง 30 คน ซึ่งในจำนวนนี้เป็นพนักงานในส่วนงานด้านคอมพิวเตอร์ต่ำกว่า 10 คน กิจการส่วนใหญ่จะมีทรัพย์สินด้านคอมพิวเตอร์และอุปกรณ์ประกอบคอมพิวเตอร์ โปรแกรมที่ใช้เป็นเครื่องมือสำหรับช่วยพัฒนาระบบงาน และซอฟต์แวร์ประยุกต์สำหรับระบบงานต่างๆ ต่ำกว่า 10 หน่วยและมีการจัดสรรงบประมาณต่อปีเกี่ยวกับเทคโนโลยีสารสนเทศต่อรายได้ของบริษัท มากกว่าร้อยละ 10

สำหรับบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจ ผู้ตอบแบบสอบถามส่วนใหญ่เป็นผู้จัดการฝ่ายคอมพิวเตอร์ซึ่งดูแลด้านการพัฒนาระบบงานคอมพิวเตอร์ มีประสบการณ์การทำงานในตำแหน่งดังกล่าวน้อยกว่า 5 ปี บริษัทที่ตอบแบบสอบถามสามลำดับแรก คือ รัฐวิสาหกิจ สถาบันการเงินและอื่นๆ กิจการส่วนใหญ่เป็นกิจการขนาดใหญ่ซึ่งมีรายได้ต่อปีประมาณ 100-10,000 ล้านบาท มีพนักงานทั้งหมดในองค์กรประมาณ 200 คน ซึ่งในจำนวนนี้เป็นพนักงานที่ทำงานด้านคอมพิวเตอร์ต่ำกว่า 10 คน การดำเนินงานของศูนย์คอมพิวเตอร์ส่วนใหญ่จะให้บริการด้านคอมพิวเตอร์กับผู้ใช้ภายในองค์กร ในด้านจำนวนคอมพิวเตอร์และอุปกรณ์ประกอบคอมพิวเตอร์ โปรแกรมที่ใช้เป็นเครื่องมือสำหรับช่วยพัฒนระบบงาน และซอฟต์แวร์ประยุกต์สำหรับระบบงานต่างๆ กิจการส่วนใหญ่จะมีทรัพย์สินดังกล่าวอยู่ระหว่าง 10-30 หน่วย และมีการจัดสรรงบประมาณต่อปีเกี่ยวกับเทคโนโลยีสารสนเทศต่อรายได้ของบริษัทน้อยกว่าร้อยละ 2.5

เมื่อพิจารณาค่าเฉลี่ยเปอร์เซ็นต์วิธีการได้มาของซอฟต์แวร์ที่องค์กรหรือส่วนงานด้านคอมพิวเตอร์ใช้งาน พบว่ากิจการที่ตอบกลับส่วนมากมีค่าเฉลี่ยเปอร์เซ็นต์ของการได้มาของซอฟต์แวร์จาก การจัดซื้อโปรแกรมสำเร็จรูปและพัฒนาเองสูงกว่าวิธีการได้มาของซอฟต์แวร์ลักษณะอื่นๆ (ค่าเฉลี่ยเปอร์เซ็นต์วิธีการได้มาของซอฟต์แวร์ มีดังนี้ การซื้อโปรแกรมสำเร็จรูป 48.85% และพัฒนาเอง 44.61 % สำหรับบริษัท Software House และการซื้อโปรแกรมสำเร็จรูป 42.81% และพัฒนาเอง 34.30% สำหรับบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ)

4.2 ลักษณะของบุคลกรทางคอมพิวเตอร์ที่ต้องการ

บริษัทส่วนมากต้องการจ้างนักวิเคราะห์ระบบที่มีประสบการณ์ที่เกี่ยวข้องมากกว่าการจ้างพนักงานโปรแกรมเมอร์ DBA และพนักงานสนับสนุนด้านเทคนิคซึ่งมีประสบการณ์ที่เกี่ยวข้อง ในด้านระดับการศึกษาของเจ้าหน้าที่ด้านคอมพิวเตอร์พบว่า ความต้องการขั้นต่ำสุดของนักวิเคราะห์ระบบของบริษัท Software House คือ ปริญญาตรีระบบสารสนเทศเพื่อการจัดการและปริญญาโทด้านระบบสารสนเทศเพื่อการจัดการและด้านคอมพิวเตอร์ ส่วนความต้องการขั้นต่ำสุดของโปรแกรมเมอร์คือปริญญาตรีด้านคอมพิวเตอร์และด้านอื่นๆ สำหรับบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจเป็นไปในทำนองเดียวกับบริษัท Software house แตกต่างกันที่ความต้องการขั้นต่ำสุดของโปรแกรมเมอร์คือ ปริญญาตรีด้านระบบสารสนเทศเพื่อการจัดการและปริญญาตรีด้านอื่นๆ

4.3 การพัฒนาระบบ

4.3.1 วิธีพัฒนาระบบ

จากค่าเฉลี่ยเปอร์เซ็นต์ของวิธีพัฒนาระบบพบว่า บริษัท Software House และบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจส่วนใหญ่ยังคงพัฒนาระบบงานโดยใช้วิธี System Development Life Cycle หรือ SDLC โดยบริษัท Software House มีแนวโน้มของค่าเฉลี่ยเปอร์เซ็นต์ของวิธีพัฒนาที่จะเป็นแบบ Object-Oriented Analysis/Design มากขึ้น สำหรับบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจนั้น พบว่าวิธีพัฒนาระบบอื่นๆ (เช่น Rapid Application Development, Structured Development, และ End-User Development) นอกเหนือจาก SDLC มีค่าเฉลี่ยใกล้เคียงกัน (วิธีพัฒนาระบบแบบ SDLC 40.45% และ Object-Oriented Analysis/Design 38.64% สำหรับบริษัท Software House และวิธีพัฒนาระบบแบบ SDLC 48.66% และ Object-Oriented Analysis/Design 16.09% สำหรับบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ)

เมื่อพิจารณาระยะเวลาที่จัดสรรในแต่ละขั้นตอนของวงจรพัฒนาระบบของวิธีพัฒนาแบบ SDLC ของกิจการที่ตอบแบบสอบถามพบว่า ค่าเฉลี่ยเปอร์เซ็นต์ระยะเวลาที่จัดสรรมากที่สุดในสามอันดับแรก คือ การปรับใช้จริง การออกแบบ และการวิเคราะห์ตามลำดับ แสดงให้เห็นว่าบริษัททั้งสองกลุ่มให้ความสำคัญกับแต่ละขั้นตอนของวงจรพัฒนาระบบดังกล่าว ดังนั้นผู้สอนควรให้ความสำคัญหรือจัดให้มีการเรียนการสอนที่เน้นขั้นตอนดังกล่าวมากขึ้น (ค่าเฉลี่ยเปอร์เซ็นต์ของระยะเวลาที่จัดสรรในขั้นตอนของวงจรพัฒนาระบบมีดังนี้ การปรับใช้จริง 28.18% การออกแบบ 25.46% และการวิเคราะห์ 18.82% สำหรับบริษัท Software house และการปรับใช้จริง 28.76% การออกแบบ 22.27% และการวิเคราะห์ 19.97% สำหรับบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ)

เมื่อพิจารณาวิธีพัฒนาระบบในรูปแบบของการจัดสร้าง Prototype พบว่าบริษัท Software House จะจัดสร้าง Prototype สำหรับระบบประมวลผลรายการแบบ Real Time, Decision Support Systems/Executive Information Systems, Expert System, และ Report Systems ซึ่งเป็นระบบใหม่ที่ไม่เคยมีการพัฒนามาก่อน ที่เป็นดังนี้ เนื่องจากบริษัท Software House จะพัฒนาระบบให้กับบริษัทลูกค้าที่ตนยังไม่ทราบลักษณะของระบบงานการจัดสร้าง Prototype ทำให้เข้าใจลักษณะของระบบงานได้ดีขึ้น แต่ถ้าเป็นระบบประมวลผลรายการแบบ Batch แล้วจะจัดสร้าง Prototype สำหรับระบบเดิมที่เคยมีการพัฒนาหรือใช้งานแล้ว จะเห็นได้ว่าระบบประมวลผลรายการแบบ Batch เป็นระบบที่มีความซับซ้อนน้อย ดังนั้นจึงสามารถนำรูปแบบของระบบแบบเดิมมาใช้งานได้ สำหรับบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจ พบว่ากิจการจะจัดสร้าง Prototype ของระบบใหม่ที่ไม่เคยมีการพัฒนามาก่อน ให้กับ Decision Support Systems/Executive Information Systems และ Expert System ซึ่งเป็นระบบที่มักมีความซับซ้อนและมักมีความแตกต่างจากระบบเดิมที่เคยจัดทำอย่างมาก ส่วนระบบประมวลผลรายการแบบ Batch, Real Time, และ Report (Inquiry) Systems กิจการมักจัดสร้าง Prototype สำหรับระบบเดิมที่เคยมีการพัฒนาหรือใช้งานแล้ว

อนึ่ง ค่าเฉลี่ยจากการสำรวจข้อมูลประโยชน์ที่องค์กรได้รับจากการจัดทำ Prototype แสดงให้เห็นว่า บริษัท Software House รับรู้ถึงประโยชน์ที่ได้รับจากการจัดทำ Prototype ในระดับปานกลาง สำหรับ *(1) การทำให้ทราบถึงการเปลี่ยนแปลงต่างๆ จากการสำรวจข้อมูลก่อนหน้า (2) การทำให้ผู้ใช้มีส่วนร่วมในการพัฒนาระบบมากขึ้น และ (3) การทำให้ทราบความต้องการของผู้ใช้ดีขึ้น* นอกจากนี้ บริษัทยังรับรู้ถึงประโยชน์ที่ได้รับในระดับน้อยสำหรับ *(1) ทำให้ลดระยะเวลาการพัฒนาระบบ (2) ทำให้ระบบที่พัฒนามีต้นทุนต่ำ และ (3) เป็นการฝึกอบรมผู้ใช้ขั้นต้น* ท้ายที่สุด จะเห็น

ได้ว่าบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจ แสดงผลในทำนองเดียวกันกับบริษัท Software House ดังกล่าวข้างต้น ตารางที่ 1 แสดงค่าเฉลี่ยประโยชน์ที่ได้รับจากการจัดทำ Prototype

สำหรับปัญหาที่กิจการซึ่งจัดทำ Prototype พบนั้น กิจการทั้งสองกลุ่มส่วนใหญ่เห็นในทำนองเดียวกันว่าปัญหา *อันดับแรก* คือ ทำให้การวิเคราะห์ระบบไม่เพียงพอ *อันดับสอง* คือ ทำให้ผู้ใช้คาดหวังว่าการพัฒนาระบบจะใช้เวลาน้อยลง และ *อันดับสุดท้าย* คือ ทำให้เอกสารประกอบระบบไม่ดี (ค่าเฉลี่ยเปอร์เซ็นต์ของปัญหาที่กิจการพบจากการจัดทำ Prototype มีดังนี้ ทำให้การวิเคราะห์ระบบไม่เพียงพอ 44.44% ทำให้ผู้ใช้คาดหวังว่าการพัฒนาระบบจะใช้เวลาน้อยลง 44.44% และทำให้เอกสารประกอบระบบไม่ดี 11.12% สำหรับบริษัท Software House และทำให้การวิเคราะห์ระบบไม่เพียงพอ 52.63% ทำให้ผู้ใช้คาดหวังว่าการพัฒนาระบบจะใช้เวลาน้อยลง 36.84% และทำให้เอกสารประกอบระบบไม่ดี 10.53% สำหรับบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ) จากข้อมูลดังกล่าวแสดงให้เห็นว่าวิธีการพัฒนาระบบโดยจัดทำ Prototype นั้น มีทั้งข้อดีและข้อเสีย ดังนั้น การเรียนการสอนในส่วนนี้ควรสอนให้ผู้เรียนทราบถึงเรื่องดังกล่าวด้วย

4.3.2 เทคนิคที่นำมาใช้

นอกเหนือจากวิธีพัฒนาระบบดังกล่าวแล้ว เทคนิคที่นำมาใช้ในการจัดเก็บข้อมูล โปรแกรมเครื่องมือ และภาษาคอมพิวเตอร์สำหรับการวิเคราะห์และออกแบบระบบของแต่ละวิธีพัฒนาระบบก็มีความสำคัญเช่นกัน ในที่นี้จะพิจารณาผลของการสำรวจเทคนิคที่ใช้ในการพัฒนาระบบออกเป็น เทคนิคสำหรับการจัดทำ Fact Finding และเทคนิคที่เป็น Diagram Techniques ซึ่งแบ่งเป็นส่วนย่อยได้ 2 ประเภท คือ Structure และ Object-Oriented Techniques ส่วนภาษาคอมพิวเตอร์สำหรับพัฒนา/บำรุงรักษางานคอมพิวเตอร์จะแบ่งออกเป็น Procedural Language, Object-Oriented Programming Language, Visual Programming Language, และ Web Page Development เป็นต้น

ผลจากการสำรวจข้อมูลเทคนิคที่บริษัทต่างๆ นำมาใช้ในการจัดทำระบบงาน แสดงให้เห็นว่าเทคนิคที่กิจการนำมาใช้ช่วยในการจัดทำ Fact Finding ซึ่งประกอบด้วย Joint Application Design, Observe, Questionnaire และ Interview นั้น กิจการส่วนมากนำมาใช้น้อยกว่าหรือเท่ากับ 50 เปอร์เซ็นต์ ยกเว้นเทคนิค Joint Application Design ที่บริษัท Software House ไม่นำมาใช้งานด้านการพัฒนาระบบงาน สำหรับ Structure Techniques ลักษณะการใช้งานของกิจการเป็นไปในทำนองเดียวกับเทคนิค

ตารางที่ 1 ค่าเฉลี่ยประโยชน์ที่ได้รับจากการจัดทำ Prototype

รายการ	บริษัท Software house	บริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ
ทำให้ทราบถึงการเปลี่ยนแปลงต่างๆ จากการสำรวจข้อมูลก่อนหน้า	3.07	2.84
ทำให้ลดระยะเวลาการพัฒนาระบบ	2.86	2.79
ทำให้ผู้ใช้มีส่วนร่วมในการพัฒนาระบบมากขึ้น	3.14	3.09
ทำให้ระบบที่พัฒนามีต้นทุนต่ำ	2.79	2.26
เป็นการฝึกอบรมผู้ใช้ขั้นต้น	2.64	2.67
ทำให้ทราบความต้องการของผู้ใช้ดีขึ้น	3.64	3.23
อื่นๆ	-	-

หมายเหตุ *ค่าเฉลี่ยมี 5 ระดับคือ 1=น้อยที่สุด 2=น้อย 3=ปานกลาง 4=มาก 5=มากที่สุด*

Fact Finding กล่าวคือกิจการส่วนมากนำมาใช้น้อยกว่าหรือเท่ากับ 50 เปอร์เซ็นต์ ยกเว้น Data Dictionary และ Entity Relationship Diagram ซึ่งบริษัท Software House ส่วนมากนำมาใช้งานมากกว่า 50 เปอร์เซ็นต์ นอกจากนี้กิจการยังไม่ใช้เทคนิค Dialogue Design Diagram และ Hierarchy Chart ในการพัฒนาระบบงาน สำหรับ Object-Oriented Techniques การใช้งานเป็นไปในทำนองเดียวกับเทคนิคต่างๆ ดังกล่าวข้างต้น กล่าวคือ กิจการต่างๆ มีเปอร์เซ็นต์การนำมาใช้น้อยกว่าหรือเท่ากับ 50 เปอร์เซ็นต์

ในด้านโปรแกรมเครื่องมือที่องค์กรใช้เพื่อให้การทำงานในขั้นการวิเคราะห์และออกแบบระบบเป็นไปโดยอัตโนมัติ ผลจากการสำรวจข้อมูลพบว่ากิจการทั้ง 2 กลุ่ม (Software House และบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจ) มีการใช้โปรแกรมที่หลากหลาย เช่น โปรแกรม Power Designer, Oracle Designer, Sybase, Rational Rose และ MS Project เป็นต้น ซึ่งส่วนมากกิจการจะมีความพึงพอใจในเครื่องมือที่ใช้งานอยู่ ยกเว้นโปรแกรมเครื่องมือบางโปรแกรมที่กิจการจะเปลี่ยนไปใช้โปรแกรมอื่นเนื่องจากล้าสมัย เช่น MS Project และ J-Builder เป็นต้น อนึ่ง มีบางโปรแกรมที่กิจการไม่พอใจในผลงานของโปรแกรมและคาดว่าจะเปลี่ยนไปใช้โปรแกรมอื่นๆ เช่น Visio เป็นต้น อย่างไรก็ตามกิจการบางแห่งของทั้ง 2 กลุ่มตัวอย่าง ไม่นำโปรแกรมเครื่องมือมาช่วยในการวิเคราะห์และออกแบบระบบดังกล่าว โดยเหตุผลหลักที่ไม่นำโปรแกรมเครื่องมือมาใช้ เนื่องจากไม่มีความคุ้นเคยกับโปรแกรมเครื่องมือนั้นๆ อย่างไรก็ตามกิจการบางแห่งคาดว่าจะหันมาใช้โปรแกรมเครื่องมืออัตโนมัติประมาณ 1 ปีข้างหน้า ดังนั้น ถ้าต้องการให้กิจการหันมาให้ความสนใจกับการใช้โปรแกรมเครื่องมือเพื่อช่วยงานการพัฒนาระบบงานให้เป็นไปโดยอัตโนมัติ อันจะส่งผลให้การพัฒนาระบบงานมีประสิทธภาพและประสิทธิผลนั้น ควรฝึกฝนเจ้าหน้าที่ในองค์กรนั้นๆ ให้คุ้นเคยกับเครื่องมือก่อน

กรณีของภาษาคอมพิวเตอร์ที่ใช้งานในการพัฒนาบำรุงรักษาระบบงานคอมพิวเตอร์ ผลจากการสำรวจข้อมูลพบว่า กิจการทั้งสองกลุ่มส่วนมากไม่ใช้ Procedural language เช่นภาษา Basic, COBOL, C เป็นต้น ซึ่งเป็นภาษาคอมพิวเตอร์ที่มีการใช้ในอดีตเป็นจำนวนมาก อย่างไรก็ตามปัจจุบันกิจการส่วนมากจะหันมาใช้ภาษาคอมพิวเตอร์ Object-Oriented Programming Language เช่น C++ และ JAVA เป็นต้น อนึ่ง บริษัท Software House จะมี

เปอร์เซ็นต์ของการใช้โปรแกรม C++ มากกว่าบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทยและรัฐวิสาหกิจซึ่งส่วนมากไม่ใช้ภาษาคอมพิวเตอร์ดังกล่าว ส่วนภาษาคอมพิวเตอร์ Visual programming language เช่น Delphi, Power Builder, และ Visual Basic นั้น พบว่ากิจการทั้งสองกลุ่มจะใช้ภาษาคอมพิวเตอร์ Visual Basic ในระดับปานกลางถึงใช้บ่อย ท้ายที่สุดสำหรับภาษาคอมพิวเตอร์ประเภท Web Page Development เช่น ASP, HTML, PHP และ XML เป็นต้น ข้อมูลจากการสำรวจแสดงให้เห็นว่า กิจการทั้งสองกลุ่มหันมาให้ความสำคัญกับภาษาคอมพิวเตอร์ที่ใช้สำหรับการสร้าง Web Page โดยกิจการส่วนมากจะใช้ภาษาคอมพิวเตอร์ เช่น ASP/ASP.net, HTML, และ PHP เป็นต้น ในระดับปานกลางถึงใช้บ่อย ส่วนภาษาอื่นๆ เช่น WML, XHTML, XML เป็นภาษาคอมพิวเตอร์ที่ใช้ในระดับปานกลางถึงไม่ใช้ จากข้อมูลดังกล่าวจะเห็นได้ว่ากิจการส่วนใหญ่ให้ความสำคัญกับโปรแกรมที่ใช้สำหรับการสร้าง Web มากขึ้น ดังนั้น การจัดการเรียนการสอนควรเน้นภาษาคอมพิวเตอร์ดังกล่าวให้มากขึ้น

4.3.3 โปรแกรมการศึกษาด้านระบบสารสนเทศเพื่อการจัดการ

จากเปอร์เซ็นต์ของทักษะที่ต้องการ แสดงให้เห็นว่าเปอร์เซ็นต์สูงสุดสามอันดับแรก ของบริษัท Software House *อันดับที่หนึ่ง* คือ การบริหารโครงการ ทักษะด้านการวิเคราะห์ระบบและความสามารถในการแก้ปัญหาตามลำดับ *อันดับสอง* คือ ความกระตือรือร้น ทักษะด้านการทำงานเป็นทีม และทักษะ

ด้านเครือข่าย ตามลำดับ และ *อันดับที่สาม* คือ ทักษะด้านฐานข้อมูล การมีจริยธรรมในการทำงานและทักษะด้านการสนทนา ส่วนเปอร์เซ็นต์สูงสุดสามอันดับแรกของทักษะที่ต้องการของบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ *อันดับแรก* คือ ทักษะด้านการวิเคราะห์ระบบ ทักษะด้านการทำงานเป็นทีม และอื่นๆ *อันดับสอง* คือ ทักษะด้านการเขียน ทักษะด้านการเขียนโปรแกรม และทักษะด้านการสนทนา *อันดับที่สาม* คือ ความสามารถในการบริหารโครงการ ทักษะด้านการเขียนโปรแกรม และระบบเครือข่าย เมื่อพิจารณาอันดับของทักษะที่ต้องการของกิจการทั้งสองกลุ่ม พบว่ากิจการสองกลุ่มมีความต้องการทักษะที่เหมือนกัน กล่าวคือ ต้องการทักษะด้านการสื่อสาร การทำงานเป็นทีม การบริหารโครงการ ทักษะด้านฐานข้อมูล การวิเคราะห์ระบบและเครือข่าย โดยทักษะที่มีความแตกต่างกันคือทักษะด้านการเขียน ความกระตือรือร้น ความสามารถในการแก้ปัญหา และการมีจริยธรรมในการทำงาน ความแตกต่างของทักษะที่ต้องการเกิดจากลักษณะของกลุ่มกิจการที่แตกต่างกัน กล่าวคือ บริษัท Software House ซึ่งพัฒนาโปรแกรมให้บริษัทลูกค้าต้องการพนักงานที่มีทักษะด้านความสามารถในการแก้ปัญหาซึ่งมีความหลากหลายมากกว่าบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจที่พัฒนาระบบงานให้หน่วยงานภายในกิจการ

4.3.4 ความสำเร็จในการพัฒนาระบบงาน

เมื่อพิจารณาถึงปัจจัยที่กิจการคิดว่า กิจการประสบความสำเร็จในการพัฒนาระบบงานพบว่า กิจการทั้งสองกลุ่มส่วนมากเห็นว่าปัจจัยที่ทำให้การพัฒนาระบบงานประสบความสำเร็จปานกลาง คือ การบรรลุความต้องการของผู้ใช้ ระบบที่พัฒนาได้ ง่ายต่อการปรับปรุงและบำรุงรักษา เอกสารประกอบระบบครบถ้วน และการสื่อสารกับผู้บริหาร ผู้ใช้ และเจ้าหน้าที่ด้านเทคนิคประสบผลสำเร็จ ส่วนปัจจัยที่มีผลต่อการประสบความสำเร็จน้อยได้แก่ พัฒนาระบบเสร็จตามเวลาที่กำหนด พัฒนาระบบเสร็จตามงบประมาณที่กำหนด ลดระยะเวลาในการเขียนโปรแกรม ระบบที่พัฒนาได้ไม่มีข้อผิดพลาด อนึ่ง บริษัทในและนอกตลาดและรัฐวิสาหกิจมีค่าเฉลี่ยของปัจจัยการพัฒนาระบบเสร็จตามงบประมาณที่กำหนดในระดับปานกลาง และค่าเฉลี่ยของเอกสารประกอบระบบครบถ้วนในระดับน้อย

ตารางที่ 2 แสดงค่าเฉลี่ยความสำเร็จในการพัฒนาระบบงาน

รายการ	บริษัท Software house	บริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ
บรรลุความต้องการของผู้ใช้	3.79	3.48
พัฒนาระบบเสร็จตามเวลาที่กำหนด	2.64	2.81
พัฒนาระบบเสร็จตามงบประมาณที่กำหนด	2.86	3.08
ลดระยะเวลาในการเขียนโปรแกรมลง	2.93	2.56
ระบบที่พัฒนาได้ไม่มีข้อผิดพลาด	2.43	2.67
ระบบที่พัฒนาได้ ง่ายต่อการปรับปรุงและบำรุงรักษา	3.29	3.13
เอกสารประกอบระบบครบถ้วน	3.07	2.56
การสื่อสารกับบุคคลเหล่านี้ประสบผลสำเร็จ:		
ผู้บริหาร	3.50	3.00
ผู้ใช้	3.57	3.33
เจ้าหน้าที่ด้านเทคนิค	3.07	3.25
อื่นๆ	-	-

หมายเหตุ *ค่าเฉลี่ยมี 5 ระดับคือ 1=น้อยที่สุด 2=น้อย 3=ปานกลาง 4=มาก 5=มากที่สุด*

4.4 ความสัมพันธ์ระหว่างตัวแปร

4.4.1 ความสัมพันธ์ระหว่างตัวแปรวิธีพัฒนาระบบกับความสำเร็จในการพัฒนาระบบงาน

ผลจากการจัดทำ Crosstabs ของบริษัททั้ง 2 กลุ่ม (บริษัท Software House และบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจ) แสดงให้เห็นว่า ตัวแปรส่วนมากในกลุ่มวิธีพัฒนาระบบกับความสำเร็จในการพัฒนาระบบไม่มีความสัมพันธ์กัน อย่างไรก็ตามมีบางตัวแปรในกลุ่มวิธีพัฒนาระบบและความสำเร็จในการพัฒนาระบบมีความสัมพันธ์กัน แต่ความสัมพันธ์ของตัวแปรดังกล่าวจะแตกต่างกันระหว่างกลุ่มบริษัททั้งสอง กล่าวคือ กลุ่มบริษัทในและนอกตลาดหลักทรัพย์และรัฐวิสาหกิจแสดงให้เห็นความสัมพันธ์อย่างมีนัยสำคัญทางสถิติระหว่าง *(1) วิธีพัฒนาระบบด้าน End User Development กับความสำเร็จในการพัฒนาระบบด้านการบรรลุความต้องการของผู้ใช้ (2) วิธีพัฒนาระบบด้าน Structured Development กับความสำเร็จในการพัฒนาระบบด้านระบบที่พัฒนาไม่มีข้อผิดพลาด และ (3) วิธีพัฒนาระบบด้าน Structured Development กับความสำเร็จในการพัฒนาระบบด้านเอกสารประกอบระบบครบถ้วน* นอกจากนี้ยังแสดงให้เห็นถึงความสัมพันธ์ที่มีนัยสำคัญส่วนเพิ่มทางสถิติ (Marginal Significance) สำหรับความสำเร็จในการพัฒนาระบบด้านการพัฒนาระบบเสร็จตามเวลาที่กำหนด ส่วนกลุ่มบริษัท Software House แสดงให้เห็นความสัมพันธ์อย่างมีนัยสำคัญทางสถิติระหว่าง *(1) วิธีพัฒนาระบบด้าน System Development Life cycle (SDLC) กับความสำเร็จในการพัฒนาระบบ* โดยลดระยะเวลาในการเขียนโปรแกรมลง และ *(2) วิธีพัฒนาระบบแบบ Rapid Application Development* เช่น Prototype กับความสำเร็จในการพัฒนาระบบด้านระบบที่พัฒนาได้ไม่มีข้อผิดพลาด อนึ่ง ความสำเร็จในการพัฒนาระบบด้านการสื่อสารกับผู้บริหารและผู้ใช้มีความสัมพันธ์กับวิธีพัฒนาระบบด้าน Object-Oriented Analysis/Design อย่างมีนัยสำคัญทางสถิติ และสัมพันธ์กับวิธีพัฒนาแบบ SDLC อย่างมีนัยสำคัญส่วนเพิ่มทางสถิติ จะเห็นได้ว่าการที่กลุ่มบริษัททั้งสองแสดงความสัมพันธ์ระหว่างตัวแปรที่แตกต่างกันอาจเนื่องมาจากลักษณะและขนาดธุรกิจที่แตกต่างกัน นอกจากนี้ ผลการวิจัยแสดงให้เห็นว่าวิธีการพัฒนาระบบที่แตกต่างกันส่งผลต่อความสำเร็จในการพัฒนาระบบที่แตกต่างกันออกไปด้วย

4.4.2 ความสัมพันธ์ระหว่างวิธีพัฒนาระบบกับขนาดของกิจการ

งานวิจัยนี้วัดขนาดของกิจการโดยใช้เกณฑ์จำนวนพนักงานทั้งหมดของกิจการ กล่าวคือกิจการที่มีจำนวนพนักงานมากกว่า 100 คน จะเป็นกิจการขนาดใหญ่ อนึ่ง ข้อมูลทางสถิติของบริษัททั้ง 2 กลุ่ม แสดงให้เห็นว่าตัวแปรวิธีพัฒนาระบบไม่มีความสัมพันธ์กับขนาดของกิจการ เมื่อพิจารณาค่าเฉลี่ยเปอร์เซ็นต์ของวิธีพัฒนาระบบซึ่งกล่าวมาแล้วข้างต้น พบว่ากิจการต่างๆ ทั้งสองกลุ่มตัวอย่าง ใช้วิธีพัฒนาระบบหลายรูปแบบผสมผสานกันไป อย่างไรก็ตาม วิธีพัฒนาระบบที่นิยมใช้กันมากที่สุด คือ SDLC รองลงมาคือ Object-oriented analysis/design ดังนั้น อาจกล่าวได้ว่าการเรียนการสอนที่เกี่ยวกับวิธีการพัฒนาควรเน้นที่การเรียนการสอน SDLC และ Object-oriented analysis/design

5. บทสรุป

5.1 สรุปงานวิจัย

งานวิจัยนี้เกิดขึ้นจากการที่ผู้วิจัยเห็นว่าปัจจุบันเทคนิคที่นำมาใช้ในการพัฒนาระบบงานคอมพิวเตอร์มีการเปลี่ยนแปลงอย่างมาก แม้ว่าภาควิชาระบบสารสนเทศเพื่อการจัดการจะจัดให้มีการเรียนการสอนวิชาทางด้านสารสนเทศเพื่อการจัดการโดยเฉพาะวิชาการวิเคราะห์และออกแบบระบบสารสนเทศที่ทันต่อการเปลี่ยนแปลงทางด้านคอมพิวเตอร์อยู่เสมอ แต่ภาควิชายังไม่มีการสำรวจความต้องการด้านเนื้อหาของวิชาที่เปิดสอนอย่างเป็นระบบเพื่อปรับเนื้อหาวิชาให้เหมาะสมกับความต้องการของผู้ว่าจ้างหรือองค์กร ดังนั้น วัตถุประสงค์หลักของงานวิจัยนี้เพื่อสำรวจความต้องการขององค์กรต่างๆ ในประเทศไทยที่เป็นผู้ว่าจ้างหลักของบัณฑิตภาควิชาระบบสารสนเทศเพื่อการจัดการที่มีต่อวิธีการ ระเบียบวิธีการพัฒนา เทคนิคและเครื่องมือที่ใช้สำหรับการวิเคราะห์และออกแบบระบบสารสนเทศ โดยทำการวิเคราะห์ผลจากการสำรวจเพื่อนำไปปรับปรุงเนื้อหาวิชาการวิเคราะห์และออกแบบระบบสารสนเทศให้ตรงตามความต้องการของผู้ว่าจ้างต่อไป

งานวิจัยนี้ใช้วิธีการวิจัยเชิงสำรวจ (Survey) เพื่อสำรวจความต้องการด้านเนื้อหาวิชาการวิเคราะห์และออกแบบระบบสารสนเทศเพื่อการจัดการ ในการจัดทำวิจัยนี้คณะผู้วิจัยได้พัฒนาแบบสอบถาม และจัดส่งแบบสอบถามไปยังหน่วยงานเอกชนตามรายชื่อที่ปรากฏในสรุปข้อสนเทศบริษัทจดทะเบียนตลาดหลักทรัพย์แห่งประเทศไทย 2544 โดยคัดเลือกบริษัทขนาดใหญ่จำนวนทั้งสิ้น 209 บริษัท (จากจำนวนทั้งหมด 360 บริษัท) บริษัท Software house ซึ่งเป็นหน่วยงานเอกชนที่ให้บริการด้านคอมพิวเตอร์ซอฟท์แวร์ตามรายชื่อที่ปรากฏในComputer Software Directory ที่จัดทำโดยกรมส่งเสริมการส่งออก กระทรวงพาณิชย์ ปี พ.ศ. 2544 จำนวนทั้งสิ้น 149 บริษัท รัฐวิสาหกิจที่เป็นสมาชิกในชมรมเทคโนโลยีสารสนเทศรัฐวิสาหกิจแห่งประเทศไทย (IT State Enterprise Club of Thailand (ITSEC) ปี พ.ศ. 2544 จำนวน 46 บริษัท และบริษัทนอกตลาดหลักทรัพย์แห่งประเทศไทยจำนวน 38 บริษัท รวมแบบสอบถามที่จะจัดส่งทั้งหมดสำหรับงานวิจัยนี้จำนวน 442 บริษัท ซึ่งมีแบบสอบถามตอบกลับมาจำนวน 57 ชุด คิดเป็นร้อยละ 12.90 ในจำนวนนี้เป็นแบบสอบถามจากบริษัท Software House ซึ่งมีชื่ออยู่ใน Computer Software Directory ซึ่งจัดทำโดยกรมส่งเสริมการส่งออก กระทรวงพาณิชย์จำนวน 14 แห่ง และบริษัทในและนอกตลาดหลักทรัพย์แห่งประเทศไทย และชมรมเทคโนโลยีสารสนเทศรัฐวิสาหกิจแห่งประเทศไทยจำนวน 43 แห่ง แบบสอบถามที่ตอบกลับมาทั้งหมดถูกนำมาใช้ในการวิเคราะห์ผลโดยใช้เปอร์เซ็นต์ ค่าเฉลี่ยและ Crosstabs ด้วยโปรแกรมสำเร็จรูป SPSS

ผลการศึกษาแสดงให้เห็นว่ากิจการส่วนใหญ่ซื้อซอฟต์แวร์สำเร็จรูปแล้วนำมาปรับปรุงให้เหมาะสมกับกิจการอีกต่อหนึ่ง อนึ่ง ซอฟต์แวร์บางประเภทได้จากการพัฒนาขึ้นมาใช้ในกิจการเองหรือพัฒนาตามความต้องการของลูกค้าหรือหน่วยงานอื่นๆ ภายในองค์กรเดียวกัน ในด้านวิธีการพัฒนาระบบพบว่ากิจการส่วนใหญ่ยังคงใช้วิธีพัฒนาระบบแบบ System Development Life Cycle หรือ SDLC และ Object-Oriented Analysis/Design โดยเทคนิคที่นำมาช่วยในการพัฒนากิจการส่วนใหญ่ยังคงใช้เทคนิคที่มีการเรียนการสอนในชั้นเรียน กล่าวคือ ยังคงใช้เทคนิค Data flow diagram (DFD), Entity-Relationship (ER) อย่างไรก็ตาม กิจการส่วนใหญ่มีแนวโน้มที่จะใช้ Object-Oriented Techniques สำหรับพัฒนาระบบมากขึ้น อนึ่ง จะเห็นได้ว่ากิจการส่วนใหญ่โดยเฉพาะบริษัท Software House มีการนำโปรแกรมเครื่องมือที่ทำให้การทำงานด้านการวิเคราะห์และออกแบบระบบเป็นไปโดยอัตโนมัติน้อย โดยโปรแกรมที่กิจการนำมาใช้งานนั้น ไม่มีความหลากหลายมากนัก สำหรับด้านภาษาคอมพิวเตอร์นั้น กิจการส่วนใหญ่ใช้ภาษา Visual Basic, C++ และ JAVA โดยกิจการส่วนใหญ่มีแนวโน้มที่จะใช้ภาษาคอมพิวเตอร์ด้าน Web Page development (เช่น ASP/ASP.net, HTML, PHP เป็นต้น) มากขึ้น และความต้องการด้านบุคลากรที่กิจการส่วนใหญ่ต้องการสามอันดับแรก คือ พนักงานที่มีความรู้ด้านการบริหารโครงการ ทักษะด้านการวิเคราะห์ระบบ และความสามารถในการแก้ปัญหา

5.2 ประโยชน์ของงานวิจัย

ผลที่ได้รับจากงานวิจัยแสดงให้เห็นว่าเนื้อหาของวิชาการวิเคราะห์และออกแบบระบบงานควรเน้นวิธีพัฒนาแบบ System Development Life Cycle (SDLC) โดยขั้นตอนที่ควรให้ความสนใจ คือ ขั้นตอนการปรับใช้จริง การออกแบบ และการวิเคราะห์ นอกจากนี้ควรเน้นวิธีพัฒนาแบบ Object-Oriented analysis/design ส่วนเทคนิคการพัฒนาคงเป็นเทคนิคที่มีการเรียนการสอนสำหรับวิธีการพัฒนาแบบ SDLC โดยเน้นที่เทคนิค Data dictionary และ Entity Relationship Diagram

สำหรับโปรแกรมเครื่องมือที่องค์กรใช้เพื่อให้การทำงานในขั้นการวิเคราะห์และออกแบบระบบเป็นไปโดยอัตโนมัตินั้น เนื่องจากกิจการที่ตอบแบบสอบถามยังมีการนำมาใช้งานน้อย ดังนั้น ผู้สอนอาจจะจัดให้มีการเรียนการสอนสำหรับเครื่องมือดังกล่าวเพิ่มเติมหรือไม่ก็ได้ อย่างไรก็ตาม กิจการที่ตอบแบบสอบถามหลายแห่งกำลังให้ความสนใจที่จะนำเครื่องมือดังกล่าวมาใช้งานเพิ่มมากขึ้น

ในด้านภาษาคอมพิวเตอร์นั้น สถาบันการศึกษาควรเน้นการเรียนการสอนที่ Object-Oriented Programming Language เช่น C++ และ JAVA เป็นต้น Visual Programming Language เช่น Delphi, Power Builder, และ Visual Basic เป็นต้น และ Web page development เช่น ASP, HTML, PHP และ XML เป็นต้น

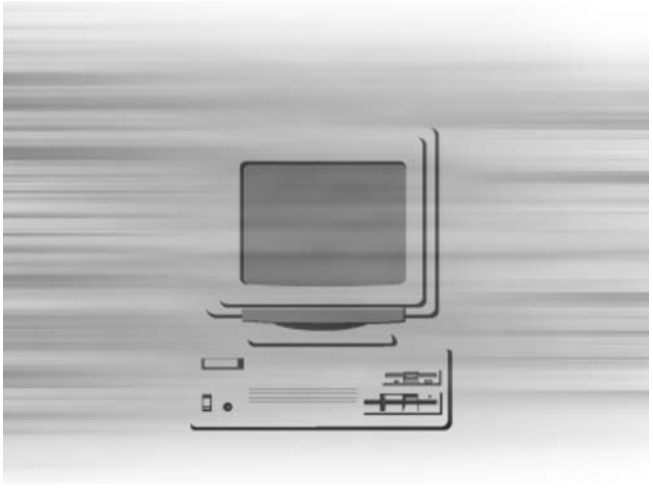
ท้ายที่สุด ในการกำหนดการเรียนการสอนควรกำหนดหลักสูตรที่เน้นให้ผู้เรียนมีความรู้ด้านการบริหารโครงการ ทักษะ

ด้านการวิเคราะห์ระบบและความสามารถในการแก้ปัญหา นอกจากนี้ ควรเน้นการเรียนการสอนที่จะทำให้นักศึกษามีประสบการณ์ในเนื้อหาวิชาที่ศึกษาด้วย

5.3 ข้อจำกัดของงานวิจัย

งานวิจัยนี้มีข้อจำกัดด้านจำนวนของกลุ่มตัวอย่างที่ตอบกลับแบบสอบถามมีจำนวนน้อยกว่าที่คาดไว้ แม้ว่าคณะผู้วิจัยจะได้ติดตามและทวงถามโดยโทรศัพท์ไปยังกิจการต่างๆ เพื่อให้จำนวนการตอบกลับแบบสอบถามมีจำนวนมากขึ้นก็ตาม การที่จำนวนแบบสอบถามมีจำนวนไม่มากนัก อาจส่งผลให้ผลของการวิจัยครอบคลุมเฉพาะกลุ่มนายจ้างซึ่งตอบแบบสอบถามกลับมาเท่านั้น ทั้งนี้ กิจการที่ไม่ตอบแบบสอบถามอาจมีลักษณะที่แตกต่างจากกิจการที่ตอบแบบสอบถาม ดังนั้น การใช้ผลของงานวิจัยนี้ควรกระทำด้วยความระมัดระวัง

อนึ่ง ความต้องการของนายจ้างเป็นเพียงปัจจัยหนึ่งที่นำมาใช้ในการพัฒนาปรับปรุงหลักสูตรเท่านั้น ยังมีปัจจัยอื่นๆ เช่น เป้าหมายการผลิตบัณฑิตที่พึงประสงค์ และการเตรียมความพร้อมของบัณฑิตสำหรับการเปลี่ยนแปลงในอนาคต เป็นต้น ที่ผู้ใช้ข้อมูลควรนำมาร่วมพิจารณาสำหรับการพัฒนาปรับปรุงหลักสูตรด้วย



5.4 งานวิจัยต่อเนื่อง

ผู้ที่สนใจอาจจัดทำวิจัยต่อเนื่องโดยปรับเปลี่ยนวิธีการวิจัยเป็นวิจัยเชิงคุณภาพ กล่าวคือ จัดเก็บข้อมูลโดยการใช้วิธีการสัมภาษณ์ หรือจัดทำ Focus group หรือจัดเก็บตัวอย่างจากวิสาหกิจขนาดกลางและขนาดย่อม ซึ่งเป็นกิจการที่รัฐบาลกำลังส่งเสริมเพื่อพิจารณาความต้องการด้านเนื้อหาของวิชาระบบสารสนเทศ พร้อมทั้งเปรียบเทียบความต้องการด้านเนื้อหาของกิจการที่มีขนาดและลักษณะที่แตกต่างกันออกไป นอกจากนี้อาจจัดทำงานวิจัยในทำนองเดียวกันแต่เน้นที่วิชาที่แตกต่างออกไป

บรรณานุกรม

ภาษาไทย

ศรีสมรัก อินทุจันทร์ยง, 2542, "การใช้บริการระบบสารสนเทศจากแหล่งภายนอก." **วารสารบริหารธุรกิจ,** ปีที่ 22 ฉบับที่ 83, กรกฎาคม-กันยายน 2542:1-20.

สมบูรณ์วัลย์ สัตยารักษ์วิทย์, ปรีชา วิจิตรธรรมรส, บุษยา วีรกุล, เกสร ชินเมธีพิทักษ์, สุภา กีร์ติบุตร, ปัญจราศี ศรไชย, และวิพร เกตุแก้ว, 2539, **รายงานวิจัยการศึกษาการใช้เทคโนโลยีสารสนเทศในประเทศไทย.**

ภาษาอังกฤษ

Anderson, D. L. and Post, G. V., 2000, **Management Information Systems.** McGraw-Hill, USA.

Banker, R.D., Davis, G.B. and Slaughter, S.A., 1998, "Software Development Practices, Software Complexity, and Software Maintenance Performance: A Field Study." **Management Science,** Volume 44(4) 1998:433-450.

Bocij, P., Chaffey, D., Greasley, A., and Hickie, S., 1999, **Business Information System.** England: Prentice Hall, USA.

Curtis, G. and Cabham, D. 2002, **Business Information Systems.** Pearson Education, USA.

Delone, W. H. and McLean, E. R., 1992, "Information Systems Success: The Quest for the Dependent Variable." **Information System Research,** Volume 3(1) 1994:60-95.

Dennis, A. and Wixom, B. H., 2000, **Systems Analysis and Design.** John Wiley & Sons, Inc., USA.

Dorfman, M. and Thayer, R.M., 1997, **Software Engineering,** Los Alamitos. IEEE Computer Society Press, CA.

Gupta, J. N. D., Wang, P., and Ravichandran, R., 1994, "An Assessment of Information Systems Education Needs in Taiwan." **International Journal of Information Management,** Volume 14 1994:369-384.

Hoffer, J.A., George, J.F., Valacich, J.S., 1996, **Modern Systems Analysis and Design (1st edition).** The Benjamin/cummings Publishing, CA.

_____, 2002, **Modern Systems Analysis and Design (3rd edition).** Pearson Education, Prentice Hall, New Jersey.

Hololan, M., and Hall, M., 2000, "Update: Oracle Embrances ASPs". **ComputerWorld** (www.computerworlld.com), July 31.

Jerva, M., 2001, "Systems Analysis and Design Methodologies: Practicalities and Use in Today's Information Systems Development Efforts." **Top Health Information Management,** Volume 21(4) 2001:13-20.

Kendall, K. E., and Kendall, L. E., 2002. **Systems Analysis and Design, (5th edition).** Prentice Hall, New Jersey.

Laudon, J. P. and Laudon, K. C., 1998, **Management Information Systems** (5th edition). Prentice Hall, New Jersey.

_____, 2003, **Management Information Systems (7th edition).** Prentice Hall, New Jersey.

Liebowiz, J., 1999, "A look at why information systems fail." **Kybernetes,** Volume 28(1) 1999:61-67.

Martin, J., 1987, **Recommend Diagramming Standards for Analysts & Programmers.** Prentice-Hall, NY.

McFarlan, W. F. and Nolan, R. L., 1995, "How to manage an IT outsourcing alliance." **Sloan Management Review,** Winter 1995:9-23.

Merrill, K., 1999, Poll: "IT Outsourcing Shows No Signs of Slowing." **TechWeb** (www.techweb.com), March 31.

Oz, E., 2002, **Management Information System (3rd edition).** Course Technology,Thomson Learning, Canada.

Pett, L. F., Watson, K. and Kavan, C.B., 1995, **Service Quality: A Measure of Information System Effectiveness,** MIS Quarterly 1995:173-187.

Saarinen, T., 1996, "An Expanded Instrument for Evaluating Information System Success." **Information and Management,** Volume 31 1996: 103-118.

Satzinger, J. W., Jackson, R. B., and Burd, S. D., 2000, **Systems Analysis and Design in a Changing World.** Thomson Learning, USA.

Shelly, G. B., Thomas, J., Cashman, K. and Harry, J. R., 1998, **Systems Analysis and design (3rd edition).** International Thomson Publishing, Cambrige MA.

_____, Cashman, A. and Misty, E. V., 2003, **Discovering Computers 2004. Course Technology,** International Thomson Publishing, USA.

Stair, R. M., 1996, **Principles of Information Systems, (2nd edition).** Course Technology, International Thomson Publishing, USA.

Trauth, E. M., Farwell, D. and Lee, D., 1993, **"The IS Expectation Gap: Industry Expectations Versus Academic Preparation." MIS Quarterly,** September 1993:293-307.

Tudor, D. J. and Tudor, I. J., 1997, **Systems Analysis and Design: A Comparison of Structured Methods.** London.

Vitalari, N. P., 1985, "Knowledge as a Basic for Expertise in Systems Analysis: An Empirical Study." **MIS Quarterly,** September 1985:221-241.

Whitten, J. L., Bentley, L. D., and Dittman, K. C., 2001, **Systems Analysis and Design Methods (5th edition).** Irvin McGraw-Hill. Yourdon, E., 1989, Modern Structured Analysis. McGraw-Hill.