



บทที่ 2

ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 บทนำ

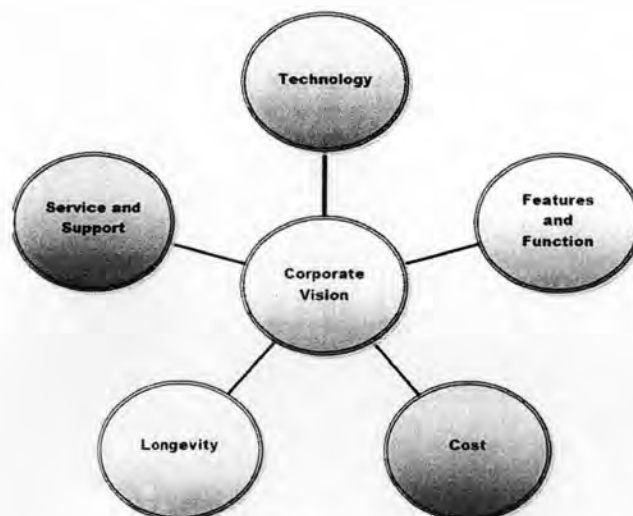
ในบทนี้เป็นการนำเสนอทฤษฎีและวรรณกรรมที่เกี่ยวข้องในแนวคิดการคัดเลือกซอฟต์แวร์ กระบวนการคัดเลือกซอฟต์แวร์ หลักการคัดเลือกซอฟต์แวร์สำเร็จของ R²ISC แผ่นตาราง (Spread Sheet) สำหรับคัดเลือกซอฟต์แวร์ และปัจจัยด้านต่างๆ ที่ใช้ในการคัดเลือกซอฟต์แวร์สำเร็จ

2.2 แนวคิดและทฤษฎีในการคัดเลือกซอฟต์แวร์

ในปัจจุบันนี้มีผู้ผลิตซอฟต์แวร์สำเร็จมากมายหลายบริษัท ที่ผลิตซอฟต์แวร์สำเร็จเพื่อรองรับการทำงานในการประกอบธุรกิจหลายๆ ประเภท มีผลิตภัณฑ์ซอฟต์แวร์สำเร็จ (Software Package) ออกวางจำหน่ายในท้องตลาดหลากหลายผลิตภัณฑ์ ด้วยคุณสมบัติและความสามารถที่แตกต่างกันออกไปถึงแม้การทำงานจะเป็นเรื่องเดียวกัน ซึ่งแต่ละองค์กรมีวิธีการทำงานที่แตกต่างกัน ดังนั้นจึงไม่ใช่เรื่องง่ายสำหรับการเลือกซื้อซอฟต์แวร์สำเร็จที่เหมาะสมกับธุรกิจหลายๆ องค์กรอยากได้ซอฟต์แวร์ที่นำมาช่วยอำนวยความสะดวกในการทำงานด้านต่างๆ แต่ก็ต้องการประหยัดงบประมาณ จึงมองหาซอฟต์แวร์สำเร็จที่มีขายในท้องตลาด (Commercial Off the Shelf Software: COTS) แทนการจัดหาพนักงานมาทำงานพัฒนาซอฟต์แวร์เอง อย่างไรก็ตามในการเลือกซอฟต์แวร์ที่ไม่เหมาะสม ทำให้ไม่สามารถรองรับการทำงานขององค์กร ความผิดพลาดของการคัดเลือกซอฟต์แวร์สำเร็จมักมาจากวิธีการคัดเลือกที่มีขั้นตอนการคัดเลือกแบบง่ายๆ ใช้ระยะเวลาในการคัดเลือกยาวนาน ทำให้เสียค่าใช้จ่ายในราคาสูง บางครั้งซอฟต์แวร์สำเร็จที่ตัดสินใจซื้อมาใช้งานอาจจะต้องถูกวางทิ้งไว้เฉยๆ อันเนื่องมาจากไม่สามารถตอบสนองหรือรองรับกับการทำงานขององค์กรได้ ส่งผลให้เสียทั้งงบประมาณ และเสียเวลาเป็นอย่างมาก (Michael, 2006)

โดยทั่วไปการคัดเลือกซอฟต์แวร์ จะมีปัจจัยที่สำคัญ คือ ซอฟต์แวร์นั้นต้องสามารถรองรับการทำงานได้มากที่สุด อย่างไรก็ตามการคัดเลือกซอฟต์แวร์สำเร็จนั้น หากพิจารณาฟังก์ชันการ

ทำงานของซอฟต์แวร์สำเร็จเพียงอย่างเดียวคงไม่เพียงพอ ยังมีอีกหลายปัจจัยที่จำเป็นต้องให้ความสำคัญ ตัวอย่างเช่น ปัจจัยในการคัดเลือกซอฟต์แวร์สำเร็จตามโมเดล R²ISC ของ Nathan (2000) มีปัจจัยในการคัดเลือกซอฟต์แวร์สำเร็จ 5 ปัจจัยคือ (1) ปัจจัยด้านฟังก์ชันของซอฟต์แวร์ที่ตรงกับความต้องการใช้งานของธุรกิจในปัจจุบัน (Current Requirement) (2) ปัจจัยด้านฟังก์ชันของซอฟต์แวร์รองรับความต้องการใช้งานของธุรกิจในอนาคต (Future Requirement) (3) ปัจจัยด้านการติดตั้งซอฟต์แวร์ (Implement Ability) (4) ปัจจัยด้านการดูแลบำรุงรักษาซอฟต์แวร์ (Support Ability) และ (5) ปัจจัยด้านราคาซอฟต์แวร์ (Cost) หรือตัวอย่างของสถาบัน Technology Group International ที่ให้คำปรึกษา แนะนำ แก่บริษัทต่างๆ ในการคัดเลือกซอฟต์แวร์ต้องใช้ปัจจัยในการคัดเลือกซอฟต์แวร์ 6 ปัจจัยได้แก่ (1) มุมมองภาพรวมของปัจจัยทั้งหมด (Corporate Vision) (2) ปัจจัยด้านเทคโนโลยีที่ใช้ในการพัฒนาซอฟต์แวร์สำเร็จ (Technology) (3) ปัจจัยด้านการให้ความช่วยเหลือและบริการหลังการขาย (Service and Support) (4) ปัจจัยด้านฟังก์ชันการใช้งาน (Features and Function) (5) อายุการใช้งาน (Longevity) และ (6) ปัจจัยด้านราคา (Cost) ดังรูปที่ 2.1 (Technology Group International ,2007)



รูปที่ 2.1 แสดงปัจจัยที่ใช้เป็นเกณฑ์คัดเลือกซอฟต์แวร์สำเร็จ

2.3 กระบวนการคัดเลือกซอฟต์แวร์สำเร็จ (Software Selection Process)

Komoski และคณะ (1995) ได้ศึกษาและแบ่งกระบวนการคัดเลือกซอฟต์แวร์สำเร็จ โดยแบ่งออกเป็น 7 ขั้นตอน ประกอบด้วย (1) ขั้นตอนวิเคราะห์ความต้องการ (Analyze Needs) ของธุรกิจของบริษัท (2) ขั้นตอนกำหนดความต้องการของระบบ (Specify Requirements) (3) ขั้นตอนการเลือกซอฟต์แวร์ที่ใกล้เคียงกับความต้องการ (Identify Promising Software) (4) ขั้นตอนการศึกษารายละเอียดของซอฟต์แวร์สำเร็จที่เลือกไว้ (Read Relevant Reviews) (5) ขั้นตอนทดลองใช้งานซอฟต์แวร์สำเร็จ (Preview Software) (6) ขั้นตอนสรุปผลการใช้งานซอฟต์แวร์ (Make Recommendations) และ (7) ขั้นตอนการเก็บข้อมูลเกี่ยวกับการใช้งานซอฟต์แวร์สำเร็จ หลังจากนั้นจึงตัดสินใจซื้อและเริ่มใช้งาน (Get Post Use Feedback) โดยมีรายละเอียดดังต่อไปนี้

2.3.1 ขั้นตอนวิเคราะห์ความต้องการ (Analyze Needs) เป็นการศึกษาในรายละเอียดการทำงานในองค์กร ศึกษาพฤติกรรมการทำงานของพนักงานในองค์กร ไม่ได้มองเพียงแต่ว่าหน่วยงานในองค์กรนั้น ทำงานอะไร มีหน้าที่อะไร แต่ต้องศึกษาในรายละเอียดการทำงานและพฤติกรรมการทำงานจริงๆ ของพนักงาน สาเหตุที่ต้องทำเช่นนี้ก็เพราะว่า ในตำแหน่งงานเดียวกันของแต่ละองค์กร อาจจะทำเนื้องานหรือมีกิจกรรมการทำงานที่แตกต่างกันออกไป ดังนั้น หากต้องการซอฟต์แวร์สำเร็จที่เหมาะสมกับองค์กรมากที่สุด จะต้องเข้าใจถึงพฤติกรรมการทำงาน กระบวนการทำงานของแต่ละหน่วยงานให้มากที่สุด เพื่อตอบสนองให้ตรงกับความต้องการของหน่วยงานนั้นๆ

2.3.2 ขั้นตอนกำหนดความต้องการของระบบ (Specify Requirements) เป็นการระบุความต้องการของระบบที่จำเป็นสำหรับสนับสนุนการทำงานของแต่ละหน่วยงานในองค์กร กำหนดคุณสมบัติที่จำเป็นและฟังก์ชันการทำงานของซอฟต์แวร์ที่จะขาดไม่ได้ โดยจะต้องกำหนดให้สอดคล้องกับสภาพการทำงานที่กำหนดไว้แล้วก่อนหน้านี้ การกำหนดความต้องการซอฟต์แวร์นี้ต้องรวมไปถึง การกำหนดคุณสมบัติของฮาร์ดแวร์ที่จำเป็นต้องใช้งานในซอฟต์แวร์งบประมาณในการจัดซื้อซอฟต์แวร์สำเร็จ และ ความสามารถในการทำความเข้าใจของผู้ใช้งานที่มีต่อซอฟต์แวร์นั้น

2.3.3 ขั้นตอนการเลือกซอฟต์แวร์ที่ใกล้เคียงกับความต้องการ (Identify Promising Software) เกิดจากเหตุการณ์ที่หาฟังก์ชันที่ตรงกับความต้องการใช้งานไม่ได้ ต้องคัดเลือกซอฟต์แวร์สำเร็จที่มีฟังก์ชันการใช้งานใกล้เคียงกับความต้องการขององค์กรมากที่สุด โดยทำรายงานบันทึกรายชื่อซอฟต์แวร์สำเร็จ (Long List) ทุกอันที่ใกล้เคียงกับความต้องการ เพื่อใช้สำหรับประกอบการตัดสินใจคัดเลือกซอฟต์แวร์สำเร็จในลำดับต่อไป

2.3.4 ขั้นตอนการศึกษารายละเอียดของซอฟต์แวร์สำเร็จที่เลือกไว้ (Read Relevant Reviews) เป็นการศึกษาในรายละเอียดของซอฟต์แวร์สำเร็จที่คัดเลือกไว้ ศึกษาคุณสมบัติการใช้งานของซอฟต์แวร์สำเร็จ และยังรวมถึงการศึกษบทความที่เกี่ยวข้องของในประเด็นการใช้งานซอฟต์แวร์สำเร็จนั้นๆ โดยอาจจะมือนักวิชาการ หรือหน่วยงานอื่นๆ ที่เคยใช้ซอฟต์แวร์ดังกล่าว ได้เขียนบทความ หรือให้ความเห็นไว้

2.3.5 ขั้นตอนทดลองใช้งานซอฟต์แวร์สำเร็จ (Preview Software) เป็นการทดลองใช้งานจริงจากซอฟต์แวร์สำเร็จที่สนใจ ประเด็นนี้ถือว่ามีค่ามากก่อนทำการตัดสินใจเลือกซื้อซอฟต์แวร์สำเร็จ เพราะการอ่านบทความ หรือศึกษาคุณสมบัติของซอฟต์แวร์เพียงอย่างเดียว อาจจะยังไม่เพียงพอสำหรับการตัดสินใจ เมื่อได้ทดลองใช้งาน ก็จะทำให้สามารถมองภาพการใช้งานได้ชัดเจนมากยิ่งขึ้น ทั้งนี้ขึ้นอยู่กับนโยบายของผู้ขายซอฟต์แวร์สำเร็จแต่ละซอฟต์แวร์ ด้วยว่าจะมีนโยบายอย่างไร ตัวอย่างเช่น บางบริษัทอาจจะอนุญาตให้ทดลองใช้งานฟรีในบางฟังก์ชันการทำงาน หรือบางบริษัทอาจจะเพียงแค่อธิการทำงานในบางฟังก์ชันงานเท่านั้น นอกจากนี้แล้วยังทำให้องค์กรสามารถศึกษาการให้บริการที่มีต่อองค์กรของบริษัทผู้ขายซอฟต์แวร์สำเร็จในแต่ละรายได้ บริษัทผู้ขายให้ความสนใจและบริการเป็นที่น่าพึงพอใจต่อองค์กรได้มากน้อยเพียงใด

2.3.6 ขั้นตอนสรุปผลการใช้งานซอฟต์แวร์ (Make Recommendations) เป็นการทำรายงานสรุปผล หลังจากทดลองใช้งาน หรือหลังจากได้รับชมการสาธิตการใช้งานจากบริษัทผู้ขายของแต่ละซอฟต์แวร์สำเร็จ ในรายงานจะระบุเหตุผลประกอบว่า ซอฟต์แวร์สำเร็จของผู้ผลิตแต่ละรายมีความน่าสนใจ และความสามารถในการตอบสนองความต้องการขององค์กรได้มากน้อยเพียงใด เหตุใดที่ต้องซื้อกับผู้ผลิตรายนี้ และเหตุใดถึงไม่ควรซื้อกับผู้ผลิตรายนี้ หลังจากที่ได้

รายงานดังกล่าวแล้ว ก็จะทำให้ได้รายชื่อบริษัทที่มีผลิตภัณฑ์ หรือมีซอฟต์แวร์สำเร็จที่ใกล้เคียงกับความต้องการขององค์กรมากที่สุด

2.3.7 ขั้นตอนการเก็บข้อมูลเกี่ยวกับการใช้งานซอฟต์แวร์สำเร็จ หลังจากตัดสินใจซื้อและเริ่มใช้งาน (Get Post Use Feedback) เป็นการเก็บบันทึกประวัติการใช้งาน ตั้งแต่ติดตั้ง เริ่มต้นการใช้งาน และระหว่างการทำงาน หากพบปัญหาหรืออุปสรรคต่างๆ ก็ให้เก็บบันทึกประวัติไว้ เพื่อใช้เป็นข้อมูลในการปรับปรุงซอฟต์แวร์สำเร็จให้มีความเหมาะสมกับองค์กรมากที่สุด

2.4 วิธีการคัดเลือกซอฟต์แวร์สำเร็จ R²ISC ตามทฤษฎีของ Nathan

การคัดเลือกซอฟต์แวร์สำเร็จที่ไม่ตรงกับความต้องการในการใช้งานขององค์กร ไม่เพียงแค่ว่าจะทำให้ซอฟต์แวร์สำเร็จนั้นถูกวางทิ้งไว้บนชั้นวางเฉยๆ เพียงอย่างเดียวเท่านั้น (Shelf ware) แต่ยังหมายถึง การส่งผลกระทบต่อการทำงานของพนักงานในองค์กรอีกด้วย เพราะนอกจากจะสิ้นเปลืองงบประมาณในการจัดซื้อแล้ว ยังไม่สามารถนำมาใช้งาน ตามความต้องการที่มุ่งหวังไว้ได้อีกด้วย ภายใต้แนวคิดที่นี้ Nathan (2000) จึงสร้างวิธีการสำหรับการคัดเลือกซอฟต์แวร์สำเร็จ ตั้งชื่อว่า R²ISC ซึ่งเป็นการนำอักษรตัวแรกของแต่ละปัจจัยของวิธีการคัดเลือกมาตั้งเป็นชื่อ R²ISC นี้ สามารถปรับมาใช้ได้กับทุกองค์กร ไม่ว่าจะองค์กรจะมีขนาดใหญ่หรือเล็กก็สามารถนำ R²ISC นี้ ไปใช้ในการคัดเลือกซอฟต์แวร์สำเร็จได้ โดย R²ISC มี 5 ปัจจัย ดังนี้

2.4.1 ปัจจัยด้านฟังก์ชันของซอฟต์แวร์ที่ตรงกับความต้องการใช้งานของธุรกิจในปัจจุบัน (Current Requirements (R อักษรตัวแรกใน R²ISC)) เป็นการพิจารณาคุณสมบัติของฟังก์ชันงานในซอฟต์แวร์สำเร็จให้ตรงกับความต้องการใช้งานในธุรกิจขององค์กรหรือไม่ Nathan (2000) เปรียบการพิจารณาปัจจัยนี้เหมือนกับการซื้อบ้านหนึ่งหลัง หากต้องการซื้อบ้านหนึ่งหลัง บ้านที่จะซื้อนั้นจะต้องมีอะไรบ้าง เช่น (1) บ้านที่อยากซื้อเป็นแบบประเภทไหน บ้านเดี่ยวชั้นเดียว หรือสองชั้น (2) เพื่อนบ้านจะต้องเป็นอย่างไร และ (3) จำนวนห้องที่อยากจะได้ในบ้านหลังใหม่ เป็นต้น

2.4.2 ปัจจัยด้านฟังก์ชันของซอฟต์แวร์รองรับความต้องการใช้งานของธุรกิจในอนาคต (Future Requirements (R อักษรตัวที่สองใน R²ISC)) เป็นการพิจารณาคุณสมบัติของซอฟต์แวร์ที่

มีฟังก์ชันรองรับการเปลี่ยนแปลงงานธุรกิจของบริษัทในอนาคต ซอฟต์แวร์สำเร็จที่มีความสามารถในการรองรับการเปลี่ยนแปลงมาน้อยเพียงใด Nathan (2000) เปรียบชั้นตอนนี้กับการเลือกซื้อบ้านว่า หากต้องการจะขยายขนาดของห้องนอนให้มีขนาดใหญ่ขึ้น บ้านใหม่ที่จะซื้อจะสามารถรองรับความต้องการนี้ได้มาน้อยเพียงใด

2.4.3 ปัจจัยด้านการติดตั้งซอฟต์แวร์ (Implement Ability (I อักษรตัวที่สามใน R²ISC)) เป็นการพิจารณาคุณสมบัติในการติดตั้งซอฟต์แวร์สำเร็จ มีความยากหรือง่าย ในการติดตั้ง ใช้เวลามากน้อยเพียงใด

2.4.4 ปัจจัยด้านการดูแลบำรุงรักษาซอฟต์แวร์ (Support Ability (S อักษรตัวที่สี่ใน R²ISC)) เป็นการพิจารณาคุณสมบัติการดูแลบำรุงรักษาในซอฟต์แวร์สำเร็จของผู้ขายหรือตัวแทนจำหน่ายซอฟต์แวร์สำเร็จ ซึ่งเป็นบริการหลังการขายซอฟต์แวร์สำเร็จ โดยพิจารณาถึงขอบเขตในการบริการ ประเภทบริการ รวมไปถึงระยะเวลาที่ให้บริการของผู้ขายหรือตัวแทนจำหน่ายซอฟต์แวร์สำเร็จ มีความแตกต่างกันมาน้อยเพียงใด

2.4.5 ปัจจัยด้านราคาซอฟต์แวร์ (Cost (C อักษรตัวสุดท้ายใน R²ISC)) เป็นการพิจารณาราคาของซอฟต์แวร์สำเร็จ (รวมค่าใช้จ่ายในการติดตั้งซอฟต์แวร์สำเร็จ) และราคาค่าใช้จ่ายการดูแลรักษาในซอฟต์แวร์สำเร็จ

เมื่อใช้วิธีการคัดเลือกตามแนวคิด R²ISC แล้วจะช่วยแก้ปัญหาในเรื่องการคัดเลือกซอฟต์แวร์สำเร็จ ตัวอย่างเช่น ลดงานในเรื่องการขอให้ส่งข้อเสนอ (Request for Proposal: RFP) เมื่อผู้ซื้อมีความต้องการระบบที่มาสนับสนุนงาน ผู้ซื้อจะติดต่อถึงผู้ขายซอฟต์แวร์เพื่อให้ส่งข้อเสนอมาและตรวจสอบการสาธิตของซอฟต์แวร์สำเร็จ (Demo) ตรงกับการใช้งานในธุรกิจของบริษัทจริง เชื่อถือได้และเป็นประโยชน์ในการเจรจาซื้อซอฟต์แวร์สำเร็จ (Nathan, 2000)

2.5 แผ่นตาราง (Spread Sheet) สำหรับคัดเลือกซอฟต์แวร์

ศึกษารูปแบบการคัดเลือกซอฟต์แวร์โดยใช้แผ่นตาราง (Spread Sheet) เป็นเครื่องมือช่วยในการคัดเลือก

2.5.1 แผ่นตาราง (Spread Sheet) สำหรับคัดเลือกซอฟต์แวร์ ของสถาบัน

Technology group International

สถาบัน Technology group International (TGI) ได้พัฒนาแผ่นตารางเพื่อใช้เป็นเครื่องมือสำหรับการคัดเลือกซอฟต์แวร์อีอาร์พี (Enterprise Resource Planning: ERP) ลักษณะการทำงานของเครื่องมือนี้จะให้หน่วยงานจัดตั้งทีมในการคัดเลือกซอฟต์แวร์ขึ้นมา โดยให้ทีมดังกล่าวพิจารณาการคัดเลือกซอฟต์แวร์ตามปัจจัยในการคัดเลือกซอฟต์แวร์ที่กำหนดไว้ในวิธีการคัดเลือกซอฟต์แวร์ของสถาบัน ตัวอย่างเช่น ปัจจัยด้านเทคโนโลยีและโครงสร้างพื้นฐาน (Technology and Infrastructure) ตามตารางที่ 2.1

ตารางที่ 2.1 แผ่นตารางการคัดเลือกซอฟต์แวร์ตามปัจจัยด้านเทคโนโลยีและโครงสร้างพื้นฐาน

Technology and Infrastructure							
		Vendor A Responses		Vendor B Responses		Vendor C Responses	
Functional Requirement	Project Team's Priority Level	Vendor A	Vendor Weighted Average	Vendor B	Vendor Weighted Average	Vendor C	Vendor Weighted Average
Supplier Offers Project Management services	3	3	9	3	9	5	15
Supplier Offers End User Training services	3	9	27	4	12	7	21
Supplier Offers Technical Training services	3	4	12	3	9	1	3
.....
Weighted Average Total	49		432		432		408

จากตารางที่ 2.1 ปัจจัยด้านเทคโนโลยีและโครงสร้างพื้นฐานจะประกอบด้วยรายละเอียดในการพิจารณาหลายหัวข้อ เช่น การให้บริการเพื่อสนับสนุนการบริหารโครงการ (Supplier Offers

Project Management services) เป็นต้น ทีมงานให้ระดับคะแนนความต้องการของหน่วยงานใน คอลัมน์ Project Team's Priority Level ตามระดับการให้คะแนนในตารางที่ 2.2

ตารางที่ 2.2 แสดงระดับการให้คะแนนของ TGI

คะแนน	หมายถึง
0	ไม่มี Feature ของซอฟต์แวร์เข้ากับองค์กร
1	อาจจะใช้ Feature ของซอฟต์แวร์บ้าง
2	ปรารถนาใน Feature ของซอฟต์แวร์
3	ต้องการ Feature ของซอฟต์แวร์

ต่อจากนั้นทีมงานจะให้คะแนนซอฟต์แวร์ของผู้ขายแต่ละราย เช่น ในคอลัมน์ Vendor A ทีมงานพิจารณาจากการสนับสนุนตามรายละเอียดที่กำหนดอยู่ในปัจจัยด้านเทคโนโลยีและโครงสร้างพื้นฐาน โดยให้ระดับคะแนนอยู่ในช่วง 0 ถึง 10 หลังจากนั้นแผ่นตารางจะคำนวณระดับคะแนนเฉลี่ยแสดงไว้ในคอลัมน์ Vendor Weighted Average จากนั้นแผ่นตารางจะสรุปผลรวมระดับคะแนนเฉลี่ยของปัจจัยด้านเทคโนโลยีและโครงสร้างพื้นฐานเพื่อใช้ในการพิจารณาคัดเลือกซอฟต์แวร์

เมื่อคำนวณปัจจัยในการคัดเลือกซอฟต์แวร์ครบแล้ว จะนำผลรวมของระดับคะแนนในการคัดเลือกซอฟต์แวร์จากตัวแทนจำหน่ายแต่ละรายมาเปรียบเทียบ โดยระดับคะแนนในการคัดเลือกซอฟต์แวร์ของตัวแทนจำหน่ายซอฟต์แวร์ที่สูงที่สุดจะมีความเหมาะสมต่อการใช้งานขององค์กร ตามตารางที่ 2.3 แผ่นตาราง (Spread Sheet) สรุปการคัดเลือกซอฟต์แวร์

ตารางที่ 2.3 แผ่นตาราง (Spread Sheet) สรุปการคัดเลือกซอฟต์แวร์

Requirements Section	Item Per Section	Point Value of Responses			Percentage Value of Responses		
		Vendor A	Vendor B	Vendor C	Vendor A	Vendor B	Vendor C
Technology/Infrastructure	49	432	432	408	99%	99%	93%
Application Summary	24	196	196	196	96%	96%	96%
Accounts Payable	47	390	366	387	98%	92%	98%
Purchase Order & RFQ Management	108	858	687	801	99%	79%	92%
Order Management & Quotation Processing	193	1575	1495	1485	100%	95%	94%
.....
Weighted Average Total	1134	9385	8885	8636	98.53%	93.28%	90.67%

จากตารางที่ 2.3 ตัวแทนจำหน่าย A มีผลรวมระดับคะแนนเฉลี่ยสูงสุดเท่ากับ 98.53% จึงสรุปได้ว่าซอฟต์แวร์ของตัวแทนจำหน่าย A มีความเหมาะสมต่อการใช้งานขององค์กรมากที่สุด

2.5.2 แผ่นตาราง (Spread Sheet) R²ISC สำหรับคัดเลือกซอฟต์แวร์

Nathan (2000) ได้สร้างแผ่นตาราง (Spread Sheet) ไว้รองรับทฤษฎี R²ISC เพื่อให้กระบวนการคัดเลือกซอฟต์แวร์สามารถดำเนินการให้เสร็จสิ้นภายใน 4-5 สัปดาห์ จากเดิมที่เคยใช้เวลาในการคัดเลือกซอฟต์แวร์อยู่ประมาณ 3-4 เดือน ลักษณะของแผ่นตารางที่สร้างขึ้นประกอบด้วย 6 หน้าหลัก มี 5 หน้าหลักเป็นการพิจารณาปัจจัยของ R²ISC คือ (1) ปัจจัยด้านฟังก์ชันของซอฟต์แวร์ที่ตรงกับความต้องการใช้งานของธุรกิจในปัจจุบัน (Current Requirement) (2) ปัจจัยด้านฟังก์ชันของซอฟต์แวร์รองรับความต้องการใช้งานของธุรกิจในอนาคต (Future Requirement) (3) ปัจจัยด้านการติดตั้งซอฟต์แวร์ (Implement Ability) (4) ปัจจัยด้านการดูแลบำรุงรักษาซอฟต์แวร์ (Support Ability) และ (5) ปัจจัยด้านราคาซอฟต์แวร์ (Cost) และหน้าที่หกเป็นหน้าสรุปผลแต่ละปัจจัยของ R²ISC

ตัวอย่างเช่น หน้าของการพิจารณาปัจจัยด้านฟังก์ชันของซอฟต์แวร์ที่ตรงกับความต้องการใช้งานของธุรกิจในปัจจุบัน (Current Requirements) เป็นการพิจารณาคุณสมบัติของ

ฟังก์ชันงานในซอฟต์แวร์สำเร็จตรงกับความต้องการใช้งานในธุรกิจขององค์กรหรือไม่ มีวิธีการพิจารณาการให้คะแนนปัจจัย โดยเริ่มจากการให้คะแนนน้ำหนักความต้องการของบริษัทในคอลัมน์ Company Need มาคูณกับน้ำหนักคะแนนความเหมาะสมของแต่ละฟังก์ชันในคอลัมน์ Raw Score โดยเปรียบเทียบระหว่างซอฟต์แวร์สำเร็จจากผู้ขายซอฟต์แวร์ 4 รายในที่นี่คือ Vendor1 Vendor2 Vendor3 และ Vendor4 ผลลัพธ์ที่ได้จากการคูณจะอยู่ในคอลัมน์ Weight Score แถว Total Score เป็นการรวมคะแนนของปัจจัยด้านฟังก์ชันของซอฟต์แวร์ที่ตรงกับความต้องการใช้งานของธุรกิจในปัจจุบันแต่ละซอฟต์แวร์ ต่อจากนั้นนำค่าของ Total Score แต่ละซอฟต์แวร์ มาหาร ผลรวมของ Company Need คูณจำนวนเต็มของ Raw Score เพื่อเป็นค่าเปอร์เซ็นต์ในการเปรียบเทียบ รายละเอียดตามตารางที่ 2.4

เขียนเป็นสูตรได้ดังนี้ :

$$\text{Percentage Fit Current Requirements} = \frac{\text{Total Score แต่ละซอฟต์แวร์}}{\text{Total Score ของ Company Need * ค่าเต็ม Raw Score}}$$

ปัจจัยความต้องการใช้งานของธุรกิจในปัจจุบัน (Current Requirements) ซอฟต์แวร์ของ Vendor1 ที่ตรงกับความต้องการใช้งาน 9 คะแนน จาก 30 คะแนน ถ้าค่าคะแนนเต็ม (Score) 10 จะได้กี่เปอร์เซ็นต์

แทนค่า ซอฟต์แวร์ของ Vendor1 :

$$\text{Percentage Fit Current Requirements} = \frac{9}{30 \times 10} = \frac{3}{100}$$

Percentage Fit Current Requirements = 3% ของความต้องการใช้งานของธุรกิจในปัจจุบัน

สรุปได้ว่า ซอฟต์แวร์ของ Vendor1 สนับสนุนองค์กรได้ 3% ของความต้องการใช้งานของธุรกิจในปัจจุบัน

ตารางที่ 2.4 แสดงหน้า Current Requirements ของแผ่นตาราง (Spread Sheet) R²ISC

Current Requirements - Business Area Summary									
Business Areas	Company Need	Raw Score (1-10)				Weighted Score			
		Vendor 1	Vendor 2	Vendor 3	Vendor 4	Vendor 1	Vendor 2	Vendor 3	Vendor 4
Business Area #1	10	2	2	2	10	20	20	20	100
Business Area #2	5	1	1	1	1	5	5	5	5
Business Area #3	2	2	3	5	6	4	6	10	12
Business Area #4	4	2	3	5	6	8	12	20	24
Business Area #5	9	2	3	5	6	18	27	45	54
	-----	-----	-----	-----	-----	-----	-----	-----	-----
Total Score	30	9	12	18	41				195
Percentage Fit Current Requirements						3%	4%	6%	65%

หน้าสรุปผลในแผ่นตาราง R²ISC จะให้ระดับความสำคัญของปัจจัย R²ISC ใน คอลัมน์ Company's Need ซึ่งผลรวมระดับความสำคัญจะได้เท่ากับ 100% ดังตารางที่ 2.5

R²ISC จะนำคะแนนรวมจากปัจจัยทั้ง 5 คือ (1) ปัจจัยด้านฟังก์ชันของซอฟต์แวร์ที่ตรงกับความต้องการใช้งานของธุรกิจในปัจจุบัน (Current Requirement) (2) ปัจจัยด้านฟังก์ชันของซอฟต์แวร์รองรับความต้องการใช้งานของธุรกิจในอนาคต (Future Requirement) (3) ปัจจัยด้านการติดตั้งซอฟต์แวร์ (Implement Ability) (4) ปัจจัยด้านการดูแลบำรุงรักษาซอฟต์แวร์ (Support Ability) และ (5) ปัจจัยด้านราคาซอฟต์แวร์ (Cost) มารวมในหน้าผลสรุป ตามตารางที่ 2.6

ตารางที่ 2.5 แสดงตัวอย่างการให้ระดับความสำคัญในแผ่นตาราง R²ISC

R ² ISC Worksheet					
R ² ISC Criteria	Company's Need	Package A Raw Score	Package A Weighted Score	Package B Raw Score	Package B Weighted Score
Current Requirements	30%
Future Requirements	25%
Implementability	20%
Supportability	15%
Cost	10%
Package Total Score	100%				

นำค่าการคำนวณปัจจัยต่างๆ เช่น การนำค่า Current Requirements ในตารางที่ 2.4 ซอฟต์แวร์ของ Vendor1 ที่ได้คำนวณได้มาเป็น 3% มาใส่ในตารางที่ 2.6 เพื่อเปรียบเทียบค่าปัจจัยทั้งหมด

ตารางที่ 2.6 แสดงหน้า Summary ของแผ่นตาราง (Spread Sheet) R²ISC

HIGH LEVEL REQUIREMENTS	Company Need	Raw Score				Weighted Score			
		Package 1	Package 2	Package 3	Package 4	Vendor 1	Vendor 2	Vendor 3	Vendor 4
		1	2	3	4	1	2	3	4
Current requirements	30	3%	4%	6%	65%	1%	1%	2%	20%
Future requirements	10	7%	8%	30%	7%	1%	1%	3%	1%
Implement ability	10	40%	41%	30%	38%	4%	4%	3%	4%
Supportability	30	39%	54%	61%	39%	12%	16%	18%	12%
Cost	20	50%	30%	30%	25%	10%	6%	6%	5%
R ² ISC Rating	100					27%	28%	32%	41%

จากตารางที่ 2.6 สรุปผลการคัดเลือกครั้งนี้ได้ว่า ซอฟต์แวร์ของตัวแทนจำหน่ายที่ 4 มีผลรวมระดับคะแนนเฉลี่ยสูงสุดเท่ากับ 41% แสดงว่าสนับสนุน 41% จากความต้องการในองค์กร 100% ซึ่งสนับสนุนความต้องการไม่ถึงครึ่งของความต้องการทั้งหมด ควรหาซอฟต์แวร์มาคัดเลือกใหม่ แต่ถ้ามีความจำเป็นที่ต้องใช้ซอฟต์แวร์เพื่อสนับสนุนงานในองค์กรมีให้เลือกเพียง 4 ซอฟต์แวร์ ซอฟต์แวร์ของตัวแทนจำหน่ายที่ 4 เหมาะสมกับองค์กรมากที่สุดในการคัดเลือกครั้งนี้

2.6 ปัจจัยด้านความต้องการที่ไม่เป็นฟังก์ชัน (Non Functions)

Bandor (2006) ได้ศึกษาปัจจัยที่ใช้ในการคัดเลือกซอฟต์แวร์ โดยนำปัจจัยด้านความต้องการที่ไม่เป็นฟังก์ชัน (Non Functions) มาใช้ในการพิจารณาคัดเลือกซอฟต์แวร์ Bandor ใช้ชื่อปัจจัยที่ไม่มีตัวตน (Intangible Factor) แทนปัจจัยด้านความต้องการที่ไม่เป็นฟังก์ชัน (Non Functions) โดยมีวิธีการพิจารณาปัจจัยด้านความต้องการที่ไม่เป็นฟังก์ชัน (Non Functions) เพื่อใช้คัดเลือกซอฟต์แวร์จากการนับจำนวนความถี่ของการตอบคำถามในลักษณะให้เลือกตอบว่า ใช่หรือไม่ใช่ จากทีมในการคัดเลือกซอฟต์แวร์ ตามตารางที่ 2.7

ตารางที่ 2.7 แสดง Intangible Factor

Intangible Factor	Consideration
Can other people work on it?	Does the software require specialized language training or techniques to use it or integrate it into the system?
Are you going to change?	Are your organization's business processes/requirements subject to a large amount of change?
What is the scope?	Is this software being applied to only one area of the system, or is it being reused across many areas?
Is it overkill?	Are you buying more "bells and whistles" than you really need? You may be paying for many features that can't be used or that could have a detrimental effect on the architecture.
Remember the end user.	The end user is the person who is most likely affected by your decision. Will integrating this software require additional training or changes to the process?
What is the additional time/cost to modify or interface to the software?	Interface development may still be needed to integrate the software or fully take advantage of its features.
How well does it integrate or "play well" with the other applications within the architecture?	If the software doesn't integrate well, it may be necessary to make a significant change to the architecture. Remember that time is not on your side!
What kind of documentation and support are available?	If there is a lack of documentation and support, the integration may be difficult and your organization may need a significant amount of time to understand how the software works.

ในการวิเคราะห์เพื่อคัดเลือกซอฟต์แวร์ Bandor ใช้แผนตารางทำการวิเคราะห์ตัดสินใจเลือก (Decision Analysis Spreadsheet) โดยให้ทีมในการคัดเลือกซอฟต์แวร์เป็นผู้ให้คะแนน ในคอลัมน์ Weight เป็นการให้น้ำหนักความสำคัญของเกณฑ์ในการตัดสินใจ (Decision Criterion) โดยที่ผลรวมของน้ำหนักทั้งหมดจะเท่ากับ 100% ในคอลัมน์ Weight เป็นความถี่ในการตอบคำถามคิดเป็นร้อยละของปัจจัยความต้องการที่ไม่เป็นฟังก์ชันทั้งหมดที่ใช้ในการพิจารณาคัดเลือกซอฟต์แวร์ ในคอลัมน์ Raw เป็นการให้คะแนนความต้องการซอฟต์แวร์ตามความพอใจของทีมในการเลือกตามตารางที่ 2.8

ตารางที่ 2.8 แสดงแผ่นตารางทำการวิเคราะห์ตัดสินใจเลือก (Decision Analysis Spreadsheet)

			Software Alternatives			
			System 1		System 2	
Item	Decision Criterion	Weight	Raw	Weighted	Raw	Weighted
A	Rule-based presentation	20%	1.0	20.00%	1.0	20.00%
B	Reliable fault tolerant	10%	1.0	10.00%	1.0	10.00%
C	Scalable	10%	1.0	10.00%	1.0	10.00%
D	Product/vendor maturity	10%	0.5	5.00%	1.0	10.00%
E	Vendor support	10%	0.5	5.00%	1.0	10.00%
F	Low total cost of ownership	10%	0.0	0.00%	1.0	10.00%
G	Extensible	10%	1.0	10.00%	1.0	10.00%
H	Single-vendor solution	5%	-0.5	-2.50%	1.0	5.00%
I	Visual rules definition/administration	15%	1.0	15.00%	0.5	7.50%
Total		100%		72.5%		92.50%

จากตารางที่ 2.8 ในคอลัมน์ Weighted ได้จากการนำคอลัมน์ Weight คูณกับคอลัมน์ Raw จากนั้นจะรวมค่าของคอลัมน์ Weighted ของแต่ละซอฟต์แวร์ แล้วนำค่าที่ได้มาเปรียบเทียบเพื่อใช้ในการตัดสินใจคัดเลือกซอฟต์แวร์ที่เหมาะสมกับองค์กร ซึ่งการให้ระดับคะแนนความพอใจของทีมงานในคอลัมน์ Raw ตามตารางที่ 2.8

ตารางที่ 2.8 แสดงความหมายของระดับคะแนนความพอใจ

คะแนน	หมายถึง
1.0	พอใจความต้องการทางธุรกิจทั้งหมด
0.5	พอใจความต้องการบางส่วนทางธุรกิจ
0.0	เฉยๆหรือไม่แสดงความต้องการทางธุรกิจ
-0.5	ไม่พอใจความต้องการบางส่วนทางธุรกิจ
-1.0	ไม่พอใจความต้องการทางธุรกิจทั้งหมด

กรณีศึกษาเรื่องใช้วิธีเชิงคุณภาพสำหรับการเลือกซอฟต์แวร์สำเร็จของ Franch และ Carvallo (2003) ใช้การเปรียบเทียบความต้องการของผู้ใช้ซอฟต์แวร์ โดยนำ ไอเอสไอ 9126-1 (International Organization for Standardization and International Electrotechnical Commission 9126-1) ที่ว่าด้วยเรื่องคุณภาพของซอฟต์แวร์มาตรฐานมาใช้เป็นมาตรวัดในกรณีศึกษาของ Franch และ Carvallo ด้วยเหตุผลเป็นคุณภาพของซอฟต์แวร์มาตรฐานทั่วไปสามารถปรับรูปแบบตามซอฟต์แวร์สำเร็จเฉพาะได้ เป็นมาตรฐานที่สร้างลำดับชั้นคุณภาพได้และเป็นมาตรฐานที่รู้จักกันทั่วไป โดยนำคุณลักษณะและคุณลักษณะย่อยของซอฟต์แวร์ของ ไอเอสไอ 9126-1 ตามตารางที่ 2.10

ตารางที่ 2.10 แสดง Non Functional ของซอฟต์แวร์สำเร็จที่เป็นมาตรฐาน (ISO/IEC 9126-1)

ISO/IEC 9126-1 characteristics and subcharacteristics	
Characteristics	Subcharacteristics
Functionality	Suitability Accuracy Interoperability Security Functionality compliance
Reliability	Maturity Fault tolerance Recoverability Reliability compliance
Usability	Understandability Learnability Operability Attractiveness Usability compliance
Efficiency	Time behavior Resource utilization Efficiency compliance
Maintainability	Analyzability Changeability Stability Testability Maintainability compliance
Portability	Adaptability Installability Coexistence Replaceability Portability compliance

กรณีศึกษาความต้องการของผู้ใช้โปรแกรมบริการจดหมาย (Mail Servers) ของ Franch และ Carvallo (2003) โดยนำความต้องการที่ไม่เป็นฟังก์ชัน (Non Functions) ของ ไอเอสไอ 9126-1 เป็นมาตรวัด ใช้รูปแบบคุณภาพ 6 ขั้นตอนคือ (1) กำหนดขอบเขตสิ่งที่สนใจของซอฟต์แวร์ (2) คัดการณ์ลักษณะคุณภาพของซอฟต์แวร์ (3) กำหนดลำดับชั้นของคุณลักษณะ

ย่อยของซอฟต์แวร์ (4) แยกคุณสมบัติของคุณลักษณะย่อยของซอฟต์แวร์ (5) แยกที่มาของคุณสมบัติพื้นฐานของซอฟต์แวร์ (6) กำหนดความสัมพันธ์ระหว่างคุณภาพเอนทิตี (Entities)

2.7 ปัจจัยด้านผู้ขายซอฟต์แวร์

ผู้ขายซอฟต์แวร์ (Vendors) นับเป็นอีกหนึ่งปัจจัยที่มีความสำคัญที่ใช้สำหรับประกอบการตัดสินใจคัดเลือกซอฟต์แวร์สำเร็จนอกเหนือจากการพิจารณาด้านฟังก์ชันการใช้งานและต้นทุนของซอฟต์แวร์สำเร็จ ถึงแม้ว่าส่วนประกอบเหล่านั้นจะมีความสำคัญต่อกระบวนการในการคัดเลือกก็ตาม แต่ก็ไม่ควรละเลยในเรื่องการส่งมอบสินค้า และบริการหลังการขาย หากเกิดปัญหาขณะใช้งาน ผู้ขายซอฟต์แวร์ให้ความสนใจและดูแลมากน้อยเพียงใด ดังนั้น ผู้ขายซอฟต์แวร์จึงเป็นอีกหนึ่งปัจจัยที่องค์กรที่ต้องการซื้อซอฟต์แวร์สำเร็จไม่ควรมองข้ามอีกเช่นกัน

2.7.1 หลักเกณฑ์การพิจารณาปัจจัยด้านผู้ขายซอฟต์แวร์

หลักเกณฑ์ที่ใช้ประกอบการพิจารณาผู้ขายซอฟต์แวร์ มี 9 ข้อดังต่อไปนี้ (Taleo Research, 2009)

- 1) บริษัทผู้ขายซอฟต์แวร์สำเร็จมีผลิตภัณฑ์ที่ตอบสนองความต้องการขององค์กรมากน้อยเพียงใด
- 2) การให้บริการทั้งก่อน และหลังการขาย มีเงื่อนไขอย่างไร รวมไปถึงการรับประกันซอฟต์แวร์สำเร็จนั้นยาวนานเพียงใด บริษัทผู้ขายซอฟต์แวร์แต่ละรายอาจจะมีเงื่อนไขเหล่านี้แตกต่างกัน
- 3) ประมาณมูลค่าการลงทุน หากเลือกซื้อซอฟต์แวร์สำเร็จกับผู้ขายซอฟต์แวร์รายนี้แล้วจะต้องมีค่าใช้จ่ายเกิดขึ้นมากน้อยเพียงใด คุ่มค่ากับการลงทุนหรือไม่
- 4) ตรวจสอบจำนวนบริษัทอื่น ที่เลือกซื้อซอฟต์แวร์สำเร็จกับผู้ขายซอฟต์แวร์รายนี้ เพราะหากมีจำนวนลูกค้ามาก ก็ย่อมเสริมสร้างความเชื่อมั่นได้ว่า ซอฟต์แวร์สำเร็จนี้มีความน่าเชื่อถือจึงส่งผลให้มีลูกค้าจำนวนมาก

5) พิจารณาข้อมูลบริษัทหรือองค์กรอื่นที่กำลังใช้งานซอฟต์แวร์สำเร็จของผู้ขายซอฟต์แวร์แต่ละราย ว่ามีความพึงพอใจมากน้อยเพียงใด ตอบสนองความต้องการ รวมไปถึงการให้บริการเป็นไปตามเงื่อนไขที่ระบุไว้หรือไม่

6) ดูแผนการพัฒนาระบบ หากเลือกซื้อซอฟต์แวร์สำเร็จกับผู้ขายซอฟต์แวร์รายนี้แล้ว จะต้องใช้เวลาในการติดตั้ง รวมไปถึงการฝึกอบรม จนถึงขั้นตอนการใช้งานจริง จะต้องใช้ระยะเวลา มากน้อยเพียงใด

7) เงื่อนไข และข้อจำกัดของซอฟต์แวร์สำเร็จ มีรายละเอียดอย่างไรบ้าง มีข้อกำหนดใดที่เป็นการผู้มัดผู้ซื้อมากเกินไปหรือไม่ จึงเป็นสิ่งที่ต้องตรวจสอบให้รอบคอบ เพราะเงื่อนไข และข้อจำกัดของซอฟต์แวร์สำเร็จแต่ละรายอาจจะมีเหมือนกัน

8) ความสามารถในการฝึกอบรมก่อนการเริ่มใช้งาน ผู้ขายที่ขายซอฟต์แวร์สำเร็จแต่ละราย มีเงื่อนไขในการฝึกอบรมอย่างไรบ้าง ควรตรวจสอบให้รอบคอบก่อนการตัดสินใจซื้อ

9) ข้อมูลบริษัทผู้ขาย เช่น สถานะทางการเงินของบริษัทเป็นอย่างไร มีความมั่นคงในการดำเนินธุรกิจมากน้อยแค่ไหน มีที่ตั้งสำนักงานอยู่สถานที่ใดบ้าง หากเกิดปัญหาบริษัทสามารถตอบสนองความต้องการได้รวดเร็วเพียงใด

และในงานวิจัยของ Tamrat และคณะ (2002) ได้กล่าวถึงการคัดเลือกซอฟต์แวร์ตัวอย่างจากผู้ขายหรือตัวแทนจำหน่ายซอฟต์แวร์ ต้องให้ทีมคัดเลือกปัจจัยคุณสมบัติผู้ขายหรือตัวแทนจำหน่ายซอฟต์แวร์จำนวน 4 ข้อ คือ

- 1) เอกสาร (Documentation)
- 2) สนับสนุนการบำรุงรักษา (Maintenance support)
- 3) ประวัติของซอฟต์แวร์/บริษัทผู้ผลิต (Pedigree)
- 4) พื้นฐานจัดซื้อ (Pre-purchase Facility)

แสดงให้เห็นว่าผู้ขายหรือตัวแทนจำหน่ายซอฟต์แวร์ (Vendors) เป็นปัจจัยในการคัดเลือกซอฟต์แวร์สำเร็จเช่นกัน

2.7.2 การประเมินข้อเสนอของผู้ขาย (Evaluating vendor proposals) หลังจากผู้ขายซอฟต์แวร์เสนอข้อเสนอ ผู้ซื้อจัดทำกรประเมินและคัดเลือกซอฟต์แวร์ มีวิธีการ 2 วิธี ที่จะช่วยในกระบวนการประเมิน ดังนี้

1) การให้คะแนนผู้ขายซอฟต์แวร์ (Vendor rating systems) เป็นเครื่องมือที่ใช้กันอย่างแพร่หลายในการประเมินผู้ขายซอฟต์แวร์ คือระบบซึ่งผู้ขายจะถูกให้คะแนนตามเกณฑ์ที่กำหนดในการคัดเลือกซอฟต์แวร์ระบบงาน (An example of a weighted-criterion approach for evaluating competing software products) ซึ่งแสดงในตารางที่ 2.11

ตารางที่ 2.11 แสดงตัวอย่างเกณฑ์ในการเลือกแบบค่าถ่วงน้ำหนัก

ปัจจัยในการเลือก (Selection criterion)	ค่าน้ำหนัก (1-10) (Weight)	คะแนนบริษัทผู้ขาย A. (Vendor score A.)	คะแนนบริษัทผู้ขาย B. (Vendor score B.)
ความง่ายในการใช้งาน (Ease of use)	10	5x10=50	10x10=50
ความเร็วของการประมวลผล (Speed)	7	8x7=56	6x7=42
ขีดความสามารถ (Capacities)	7	8x7=56	5x7=35
ราคา (Cost)	4	4x4=16	7x4=28
การบริการ (Support)	5	3x5=15	7x5=35
เอกสารประกอบ (Documentation)	5	5x5=25	7x5=35
จำนวนงานที่ทำได้ (Functionality)	10	7x10=70	8x10=80
คะแนนรวม (Total score)		288	355

2) การเปรียบเทียบกับมาตรฐาน (Benchmark tests) หมายถึง การทดสอบเพื่อวัดความสามารถในการประมวลหรือการทำงานของโปรแกรมใดโปรแกรมหนึ่งโดยเทียบเคียงกับเกณฑ์มาตรฐาน (Parker and Case, 1993 อ้างถึงใน จันทรัตน์, 2551)