

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

การวิจัยในครั้งนี้เป็นการพัฒนานักศึกษาวิเคราะห์ความต้องการซอฟต์แวร์ รายวิชาโครงการศึกษาเฉพาะบุคคล นักศึกษาระดับปริญญาตรี สาขาวิชาบริหารระบบสารสนเทศทางคอมพิวเตอร์-พัฒนาซอฟต์แวร์ ชั้นปีที่ 4 คณะบริหารธุรกิจและศิลปศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ภาควิชาเทคโนโลยีสารสนเทศ ชั้นปีที่ 2 ปีการศึกษา 2554 ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องตามลำดับหัวข้อต่อไปนี้

1. ความหมายมาตรฐานการเก็บความต้องการซอฟต์แวร์ ISO 9126
2. องค์ประกอบของกระบวนการ ISO 9126
3. ลักษณะการจัดกิจกรรมการการเรียนรู้วิเคราะห์ความต้องการซอฟต์แวร์กระบวนการ ISO 9126
4. บ้านแห่งคุณภาพ (House of Quality: HOQ)
5. เทคนิคการกระจายหน้าที่เชิงคุณภาพ (Quality Function Deployment: QFD)
6. แผนผังการจัดลำดับความสำคัญ (Prioritization scheme)

### 2.1 มาตรฐานการเก็บความต้องการซอฟต์แวร์ ISO 9126

คุณภาพผลิตภัณฑ์ซอฟต์แวร์ได้ มีการเพิ่มขึ้นถึงความสำคัญในสังคมคอมพิวเตอร์ ความพยายามไม่ก็อย่างที่จะช่วยให้ให้ความช่วยเหลือในการปรับปรุงคุณภาพ ผลิตภัณฑ์ซอฟต์แวร์เป็นสิ่งที่พื้นฐานต้องไม่บรรลุข้อบกพร่องไว้และเพราะฉะนั้นจึงต้องเป็นสิ่งที่น่าสนใจให้เกิดความพึงพอใจกับเงื่อนไขที่มีความสำคัญ

ข้อกำหนดผลิตภัณฑ์ซอฟต์แวร์สามารถอ้างอิงถึงคุณภาพที่เพียงพอเท่านั้น เมื่อมันมีความต้องการที่พอใจสำหรับคุณลักษณะคุณภาพ อย่างเช่น การกำหนดฟังก์ชัน และ การง่ายต่อการใช้

ISO 9126 เป็นมาตรฐานสากลนานาชาติสำหรับประเมินคุณภาพซอฟต์แวร์ โดยวัตถุประสงค์พื้นฐานของการกำหนดมาตรฐานคือการระบุถึงบางสิ่งที่รู้ดีของความโน้มเอียงของมนุษย์ที่ตรงกันข้ามกับผลกระทบต่อการใช้งาน และ การเข้าใจของ โครงการพัฒนาซอฟต์แวร์ ซึ่งความโน้มเอียงนี้ได้รวมถึงการเปลี่ยนแปลงความสำคัญหลังจากการเริ่มไปแล้วของโครงการหรืออาจจะไม่ได้ระบุถึงจุดมุ่งหมายความสำเร็จอย่างชัดเจน



## 2.2 องค์ประกอบของกระบวนการ ISO 9126

โครงสร้างกลุ่มของคุณลักษณะและคุณลักษณะย่อย ดังนี้

### 2.2.1 Functionality

เป็นกลุ่มของคุณลักษณะแบกรับการมีอยู่ของกลุ่มฟังก์ชันและคุณสมบัติเฉพาะ การกำหนดหน้าที่(ฟังก์ชัน)เป็นสิ่งกำหนดสถานะความพอใจหรือถูกต้องถึงความถึงความต้องการ

1. Suitability
2. Accuracy
3. Interoperability
4. Compliance
5. Security

### 2.2.2 Reliability

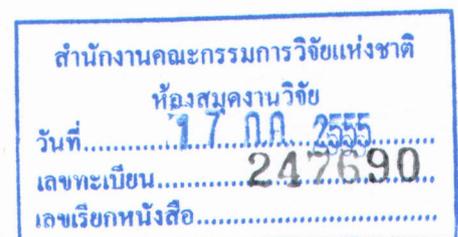
เป็นกลุ่มลักษณะที่แบกรับความสามารถของซอฟต์แวร์เพื่อที่จะสำคัญในการกำหนดลำดับสมรรถภาพภายใต้สถานะเงื่อนไขสำหรับสถานะในช่วงเวลาหนึ่ง

1. Maturity
2. Recoverability
3. Fault Tolerance

### 2.2.3 Usability

เป็นกลุ่มลักษณะที่แบกรับถึงความพยายามที่จะใช้ และเป็นเฉพาะการประเมินที่ใช้ โดยสถานะหรือการตีความกลุ่มของผู้ใช้งาน

1. Learnability
2. Understandability
3. Operability



#### 2.2.4 Efficiency

เป็นกลุ่มลักษณะที่แบกรับบนความสัมพันธ์ระหว่างระดับสมรรถภาพของซอฟต์แวร์และจำนวนของทรัพยากรที่ถูกใช้ภายใต้ สถานะเงื่อนไขการใช้

1. Time Behavior
2. Resource Behavior

#### 2.2.5 Maintainability

เป็นกลุ่มของคุณลักษณะที่แบกรับบนความพยายามที่ต้องการที่จะระบุการปรับปรุงรักษา

1. Stability
2. Analyzability
3. Changeability
4. Testability

#### 2.2.6 Portability

เป็นกลุ่มของคุณลักษณะที่แบกรับบนความสามารถของซอฟต์แวร์ที่โยกย้ายจากหนึ่งสถานะแวดล้อมไปอีกสถานะแวดล้อมอื่น

1. Installability
2. Replaceability
3. Adaptability

### 2.3 ลักษณะการจัดกิจกรรมการเรียนรู้วิเคราะห์ความต้องการซอฟต์แวร์ตามกระบวนการ ISO

#### 9126

มาตรฐานได้จัดทำกรอบการทำงานสำหรับองค์กรเพื่อที่จะ นิยามถึง โมเดลคุณภาพสำหรับผลิตภัณฑ์ซอฟต์แวร์ ซึ่งกำหนดแบบนี้ อย่างไรก็ตามมันได้ละทิ้งไปแต่ละองค์กร ตามหน้าที่เฉพาะแน่นอน ที่องค์กรนั้นเป็นเจ้าของ ซึ่งที่องค์กรจะทำได้ยกตัวอย่างเช่น การระบุถึงค่าของเป้าหมายสำหรับ เมตริกคุณภาพ แต่ละการประเมินการเข้าร่วมของคุณลักษณะคุณภาพ

ตารางที่ 2.1 คุณลักษณะความต้องการซอฟต์แวร์ตามมาตรฐาน ISO 9126 [12]

Order	Characteristic Name	คำบรรยาย
1	Functionality	การทำงานตามหน้าที่(Functionality) คือ จุดประสงค์ที่จำเป็นเพื่อกำหนดผลิตภัณฑ์และบริการที่เกี่ยวข้อง ยกตัวอย่างเช่น การถอดสมอเรือเป็นฟังก์ชันที่บริการให้เรือตั้งอยู่กับที่, ตู้ATM มีฟังก์ชันซอฟต์แวร์การถอนฝากที่มีความยุ่งยากขึ้นและได้ถูกระบุเจาะจงไว้ เช่น ระบบควรสามารถบันทึกข้อมูลลูกค้าเพื่อให้สามารถอ้างอิงเอกสารการทำรายการได้
ลักษณะย่อย		
Order	ชื่อลักษณะย่อย	คำบรรยาย
1	Suitability	การกำหนดฟังก์ชันที่อ้างอิงได้เหมาะสม เพื่อระบุรายละเอียดของฟังก์ชัน ซอฟต์แวร์
2.	Accurateness	การกำหนดฟังก์ชันที่แน่นอน เช่น เครื่อง ATM ต้องสามารถระบุฟังก์ชันจ่ายเงินสดที่มีความถูกต้องด้วย
3	Interoperability	การกำหนดฟังก์ชันให้ซอฟต์แวร์สามารถทำงานร่วมกับระบบอื่นได้
4	Compliance	การกำหนดฟังก์ชันตามกฎหมายซอฟต์แวร์ ซึ่งบังคับจากภาครัฐ ให้มีความรับผิดชอบต่อแนวทาง และระบุถึงความยินยอมการกระทำของซอฟต์แวร์
5	Security	การกำหนดฟังก์ชันให้มีความปลอดภัย เกี่ยวข้องกับการเข้าถึงสิทธิ์การใช้งานฟังก์ชันซอฟต์แวร์

ตารางที่ 2.1 คุณลักษณะความต้องการซอฟต์แวร์ตามมาตรฐาน ISO 9126 [12] (ต่อ)

Order	Characteristic Name	คำบรรยาย
2	Reliability	ความน่าเชื่อถือของซอฟต์แวร์ขณะทำงาน เป็นลักษณะกำหนดถึงความสามารถของระบบให้รักษาการบริหารที่จัดทำให้ตามภายใต้เงื่อนไขที่กำหนดตามช่วงเวลา หลักเกณฑ์ด้านหนึ่งของลักษณะความน่าเชื่อถือ คือ ความสามารถของซอฟต์แวร์ที่ทนต่อความล้มเหลว ยกตัวอย่างเช่น เมื่อเครือข่ายล้มลงระบบสามารถที่จะทำให้อัตโนมัติเป็นปกติและดำเนินต่อไปตามฟังก์ชันเดิมได้
ลักษณะย่อย		
Order	ชื่อลักษณะย่อย	คำบรรยาย
1	Maturity	ความน่าเชื่อถือของซอฟต์แวร์ขณะทำงาน สามารถทนต่อความล้มเหลวซอฟต์แวร์ซึ่งกำหนดตามความถี่ของเหตุการณ์ที่เกิดความล้มเหลว
2	Fault tolerance	ความน่าเชื่อถือของซอฟต์แวร์ขณะทำงาน สามารถทนต่อความล้มเหลวซอฟต์แวร์จากสิ่งแวดล้อม ที่ก่อให้เกิดความล้มเหลว
3	Recoverability	ความน่าเชื่อถือของซอฟต์แวร์ขณะทำงาน สามารถฟื้นคืนสภาพเดิมของซอฟต์แวร์ และการเชื่อมต่อข้อมูลกับเครือข่าย
Order	Characteristic Name	คำบรรยาย
3	Usability	ความสะดวกการใช้งานซอฟต์แวร์จากการทำงานตามฟังก์ชัน ยกตัวอย่างเช่นการถอนเงิน ATM ความสะดวกการใช้งานซอฟต์แวร์จะไม่กระทบต่อฟังก์ชันหลักการถอนเงิน แต่เพิ่มความสะดวกในการใช้งานฟังก์ชันให้ผู้ใช้เรียนรู้วิธีใช้งานระบบได้รวดเร็วขึ้น

ตารางที่ 2.1 คุณลักษณะความต้องการซอฟต์แวร์ตามมาตรฐาน ISO 9126 [12] (ต่อ)

ลักษณะย่อย		
Order	ชื่อลักษณะย่อย	คำบรรยาย
1	Understand ability	ความสะดวกการใช้งานซอฟต์แวร์ให้มีผู้ที่มีความเข้าใจต่อฟังก์ชันระบบ โดยใช้แบบจำลองทางใจ คอมพิวเตอร์มีปฏิริยาต่อมนุษย์ผู้ใช้
2	Learnability	ความสะดวกการใช้งานซอฟต์แวร์สำหรับความพยายามของผู้ใช้ในการเรียนรู้ ซึ่งแบ่งประเภทของผู้ใช้ เช่น ผู้ใช้เริ่มต้น ผู้ใช้ที่บังเอิญเข้ามาใช้
3	Operability	ความสะดวกการใช้งานซอฟต์แวร์สำหรับในสภาพแวดล้อมต่างๆให้ใช้งานซอฟต์แวร์โดยง่าย
Order	Characteristic Name	คำบรรยาย
4	Efficiency	<p>ประสิทธิภาพการใช้งานซอฟต์แวร์ ลักษณะนี้เกี่ยวข้องกับทรัพยากรระบบที่ใช้เตรียมให้พร้อมสำหรับการทำงานตามฟังก์ชัน เช่น พื้นที่ดิส, พื้นที่หน่วยความจำ, เครือข่าย การเตรียมไว้เป็นสิ่งที่ดีของลักษณะด้านประสิทธิภาพ</p> <p>ลักษณะด้านประสิทธิภาพการใช้งานซอฟต์แวร์(Efficiency)อาจมีการซ้อนทับกับด้านความสะดวกการใช้งานซอฟต์แวร์(Usability) ยกตัวอย่างเช่น ด้านความสะดวกการใช้งานซอฟต์แวร์อาจได้ได้ อิทธิพล โดยการกระทำของระบบที่มีประสิทธิภาพการใช้งานซอฟต์แวร์ที่ไม่ดีเพียงพอ เช่น ในกรณีระบบใช้ 3 ชั่วโมงเพื่อได้ตอบระบบจะไม่สะดวกง่ายต่อการใช่ แม้ว่าปัญหา มาจากลักษณะประสิทธิภาพของระบบ</p>

ตารางที่ 2.1 คุณลักษณะความต้องการซอฟต์แวร์ตามมาตรฐาน ISO 9126 [12] (ต่อ)

ลักษณะย่อย		
Order	ชื่อลักษณะย่อย	คำบรรยาย
1	Time behavior	ประสิทธิภาพการใช้งานซอฟต์แวร์ลักษณะย่อยการตอบสนองด้านเวลา ผลลัพธ์เป็นอัตราเร็วด้านเวลา เช่น transaction rate
2	Resource behavior	ประสิทธิภาพการใช้งานซอฟต์แวร์ลักษณะย่อยทรัพยากรที่ใช้ เช่น หน่วยความจำ, หน่วยประมวลผล(cpu), การใช้ดิส และ เครือข่าย พฤติกรรมทรัพยากรที่ใช้ ลักษณะทรัพยากรที่ใช้ เช่น หน่วยความจำ, หน่วยประมวลผล(cpu), การใช้ดิสและเครือข่าย
Order	Characteristic Name	คำบรรยาย
5	Maintainability	การบำรุงรักษาการใช้งานซอฟต์แวร์ ลักษณะนี้เกี่ยวกับความสามารถซ่อมความความผิดพลาดภายในส่วนประกอบซอฟต์แวร์ คืออะไรตำแหน่งใด และลักษณะการบำรุงรักษา ลักษณะย่อยด้านนี้ถูกกำหนดเป็นลักษณะการสนับสนุน ซึ่งอาจถูกผลกระทบ โดยการอ่านรหัส หรือ ความซับซ้อนจากเกณฑ์ที่วัด สิ่งที่ช่วยค้นหาสาเหตุของความผิดพลาด และ ตรวจสอบความผิดพลาด อย่างเช่น testability ก็เป็นลักษณะด้านหนึ่งของการบำรุงรักษาการใช้งานซอฟต์แวร์
ลักษณะย่อย		
Order	ชื่อลักษณะย่อย	คำบรรยาย
1	Analyzability	การบำรุงรักษาการใช้งานซอฟต์แวร์สามารถวิเคราะห์ค้นหาสาเหตุของปัญหาความล้มเหลวในซอฟต์แวร์
2	Changeability	การบำรุงรักษาการใช้งานซอฟต์แวร์สามารถยินยอมให้เปลี่ยนแปลงหน้าที่ฟังก์ชันระบบในอนาคต

ตารางที่ 2.1 คุณลักษณะความต้องการซอฟต์แวร์ตามมาตรฐาน ISO 9126 [12] (ต่อ)

Order	ชื่อลักษณะย่อย	คำบรรยาย
3	Stability	การบำรุงรักษาการใช้งานซอฟต์แวร์เมื่อให้ระบบซอฟต์แวร์แก่ผู้ใช้ ผู้ใช้จะพบถึงความรู้สึกด้านลบในการใช้งาน ซึ่งนั่นอาจจะมีสาเหตุให้เกิดการเปลี่ยนแปลงระบบแล้วระบบยังคงมั่นคงไม่กระทบต่อการทำงานร่วมกับฟังก์ชันอื่น
4	Testability	การบำรุงรักษาการใช้งานซอฟต์แวร์มีความจำเป็นต้องพิสูจน์และทดสอบระบบเมื่อเกิดการเปลี่ยนแปลงระบบ
Order	Characteristic Name	คำบรรยาย
6	Portability	ความสามารถใช้งานร่วมกับสิ่งแวดล้อมอื่น เป็นลักษณะที่อ้างถึงการสร้างซอฟต์แวร์ให้รองรับรับการใช้งานเมื่อเปลี่ยนแปลงสภาพแวดล้อมเกินจากที่ความต้องการซอฟต์แวร์ระบุไว้ ลักษณะย่อยของลักษณะด้านนี้ได้ระบุถึงการดัดแปลง(adaptability) การออกแบบเชิงวัตถุ(Object oriented design) และ เครื่องมือที่ช่วยสนับสนุนเพื่อแสดงขอบเขตของระบบที่ให้
ลักษณะย่อย		
Order	ชื่อลักษณะย่อย	คำบรรยาย
1	Adaptability	ความสามารถใช้งานร่วมกับสิ่งแวดล้อมอื่นต้องอาศัยความสามารถของระบบเพื่อเปลี่ยนให้เข้ากับสิ่งแวดล้อมใหม่ได้
2	Install ability	ความสามารถใช้งานร่วมกับสิ่งแวดล้อมอื่นต้องอาศัยความพยายามในการติดตั้งซอฟต์แวร์
3	Conformance	ความสามารถใช้งานร่วมกับสิ่งแวดล้อมอื่นอาศัยความพยายามในการทำงานของฟังก์ชันที่ใช้ทรัพยากรอื่นด้าน โครงสร้างร่วมกัน เช่น SQL ใช้ค้นหาข้อมูลร่วมกับ ฐานข้อมูลอื่นๆ ได้
4	Replace ability	ความสามารถใช้งานร่วมกับสิ่งแวดล้อมอื่นอาศัยความง่ายในการแลกเปลี่ยนทรัพยากรของประกอบภายในซอฟต์แวร์ที่ใช้ร่วมกับภาพแวดล้อมอื่น

## 2.4 บ้านแห่งคุณภาพ (House of Quality: HOQ)

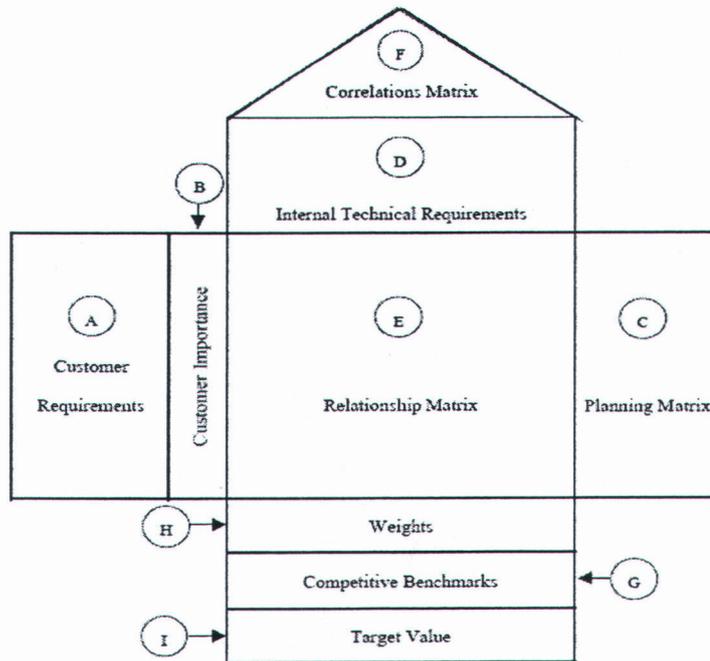
House of Quality : HOQ เป็นการรวบรวมความคิดเห็นของลูกค้าว่าต้องการให้มีคุณลักษณะอะไรบ้างในตัวของผู้ผลิตภัณฑ์หรือการบริการ (Voice of Customer) เพื่อให้ทราบว่าลูกค้าต้องการอะไร (“What” of customer desire) แล้วให้นำน้ำหนักความสำคัญแก่แต่ละคุณลักษณะ ซึ่งวิธีการรวบรวมความคิดเห็นของลูกค้านี้ทำได้หลายวิธี เช่น วิธีการกรอกแบบสอบถาม การสัมภาษณ์แบบตัวต่อตัว การสัมภาษณ์แบบ focus group เป็นต้น

ซึ่งเป็นการประเมินผลการตอบสนองจากลูกค้า หลังจากนั้นจะแปลความหมายของความ ต้องการของลูกค้า (Voice of Customer) ให้เป็นคำแทนคุณลักษณะทางด้านคุณภาพ (Substitute Quality Characteristic : SQCs) ซึ่งเป็นศัพท์ทางเทคนิคหรือข้อกำหนดที่ใช้กันภายในองค์กร เพื่อแสดงว่าจะทำอย่างไร จึงจะทำให้ได้สิ่งที่ลูกค้าต้องการ จากนั้นจัดลำดับความสำคัญว่า ทีมงานควรเริ่มที่การพัฒนา SQCs ตัวใดก่อน โดยเริ่มที่ SQCs ที่มีความสำคัญมากที่สุดที่ส่งผลกระทบต่อความต้องการของลูกค้า ในบางกรณีอาจเริ่มที่ SQCs ที่มีความจำเป็นหรือที่เป็นไปได้ที่สุดก่อน ระหว่าง SQCs เหล่านี้อาจมีความสัมพันธ์ระหว่างกัน จะต้องสามารถระบุได้ว่า SQCs ตัวใดสัมพันธ์กันอย่างไร ชัดแย้งหรือเสริมกันมากน้อยเพียงใด

จะเห็นว่าการสร้าง House of Quality นี้จำเป็นต้องอาศัยความรู้ความเชี่ยวชาญจากหลายฝ่ายในองค์กร ดังนั้นการติดต่อสื่อสารระหว่างหน่วยงานจึงสำคัญต่อการสร้าง House of Quality มา ผู้บริหารจึงควรสนับสนุนการติดต่อสื่อสารนี้ รวมทั้งทุกฝ่ายภายในองค์กรควรให้ความร่วมมือในการสร้าง House of Quality นี้ด้วย ซึ่ง House of Quality : HOQ ประกอบด้วยส่วนสำคัญ ดังนี้ (ดูรูปที่ 2.1 ประกอบ)

### 2.4.1 ข้อมูลนำเข้าลูกค้า (Customer Input)

Voice of Customer จะถูกรอกลงในส่วนทางซ้ายของ House of Quality : HOQ (A) ซึ่งควรมีการแบ่ง Voice of Customer ออกเป็นลำดับขั้นประมาณ 2-3 ชั้น เพื่อง่ายต่อการอ่าน



รูปที่ 2.1 ส่วนประกอบของ House of Quality : HOQ

คะแนนความสำคัญโดยเฉลี่ยที่ลูกค้าให้แต่ละ Voice of Customer จะ ถูกกรอกลงในสดมภ์ (B) คะแนนที่ลูกค้าให้แก่ผลิตภัณฑ์ของบริษัทเราเมื่อเปรียบเทียบกับคู่แข่ง (Customer rating หรือ customer assessment) โดยอาจให้คะแนนตั้งแต่ 1 ถึง 5 คะแนนนี้จะถูกกรอกในส่วนทางขวาของ House of Quality : HOQ (C) หรือเรียก ส่วนนี้ว่า Planning Matrix อาจจะมีการขยายส่วนนี้เพิ่มเติมได้โดยการเพิ่มสดมภ์รายการอื่นอีกเช่น ข้อมูลคำติเตียนจากลูกค้าที่แสดงความถี่ของปัญหาที่ได้รับรายงานจากลูกค้าเกี่ยวกับ Voice of Customer นั้นๆ

#### 2.4.2 ข้อมูลนำเข้าทางเทคนิค (Technical Input)

Substitute Quality Characteristics : SQCs จะถูกกรอกลงในส่วนบนของ House of Quality : HOD (D) ตัวอย่างของ SOQs เช่น ตัววัดผลงาน (Performance Measurement) หน้าที่ของผลิตภัณฑ์ (Product Function) ระบบย่อยของผลิตภัณฑ์ (Product Subsystems) และขั้นในกระบวนการ (Process Steps)

ความสัมพันธ์ระหว่าง Voice of Customer และ SQCs มักจะถูกแสดงโดยสัญลักษณ์เพื่อระบุว่ามีความสัมพันธ์กันมากน้อยเพียงใด สัญลักษณ์นี้จะถูกกรอกลงในส่วนของ House of Quality : HOQ (E) หรือเรียกส่วนนี้ว่า Relation Matrix ตัวอย่างสัญลักษณ์ที่นิยมใช้แสดงดังรูปที่ 2.2

สัญลักษณ์	คะแนน	ความสำคัญ
△	1	น้อย
○	3	ปานกลาง
◎	9	มาก

รูปที่ 2.2 ตัวอย่างสัญลักษณ์ที่ใช้ในการระบุระดับความสัมพันธ์ (Rao et al, 1996 : 397)

ความสัมพันธ์ระหว่าง SQCs แต่ละตัว มักจะถูกแสดงในรูปของสัญลักษณ์เพื่อระบุว่ามีความสัมพันธ์กันอย่างไร สนับสนุนกันหรือขัดแย้งกันในทิศทางใด สัญลักษณ์จะถูกกรอกลงในส่วนยอดของ House of Quality : HOQ (F) หรือเรียกส่วนนี้ว่า Correlation Matrix ตัวอย่างของสัญลักษณ์ที่ใช้แสดงดังรูปที่ 2.3

การประเมินความสามารถทางเทคนิคในการผลิตหรือการให้บริการ โดยเปรียบเทียบระหว่างองค์กรเราและคู่แข่ง (Technical Assessment หรือ Competitive Benchmarks) โดยให้คะแนนตั้งแต่ 1 ถึง 5 ข้อมูลนี้จะแสดงอยู่ในส่วนของ (G)

สัญลักษณ์	ความสำคัญ
✓ ✓	Strong position impact
✓	Moderate position impact
<blank>	No impact
✗	Moderate negative impact
✗ ✗	Strong negative impact

รูปที่ 2.3 ตัวอย่างสัญลักษณ์ที่ใช้ในการระบุถึงระดับและทิศทางของความสัมพันธ์

(Cohen, 1995: 155)

### 2.4.3 ผลลัพธ์ที่ได้จาก House of Quality : HOQ

ค่าน้ำหนักสัมบูรณ์ของ SQCs หรือลำดับความสำคัญของ SQCs คือ ผลรวมของผลคูณระหว่างค่าระดับความสัมพันธ์กับคะแนนของ Voice of Customer แต่ละตัวในแต่ละสดมภ์ของ SQCs ซึ่งค่านี้จะถูกกรอกลงในส่วนฐานของ House of Quality : HOQ (H)

ค่าเป้าหมาย (Target Value) ที่ทีมงานตกลงร่วมกันสำหรับแต่ละ SQCs หรืออย่างน้อยเฉพาะ SQCs ที่สำคัญที่สุดที่มีความเป็นไปได้ที่จะสามารถทำการปรับปรุง กรอกค่านี้ลงใน (I)

ข้อดีอย่างหนึ่งของ House of Quality : HOQ ที่เห็นได้ชัด คือ เป็นเมตริกที่แสดงถึงความสัมพันธ์ระหว่างข้อมูลต่างๆ ได้ภายในตารางเพียงตารางเดียว ซึ่งสิ่งนี้จะช่วยทีมงานเมื่อกลับมาอ่านอีกครั้งจะสามารถทำความเข้าใจความสัมพันธ์ระหว่างข้อมูลต่างๆ ได้อย่างสะดวกและรวดเร็ว

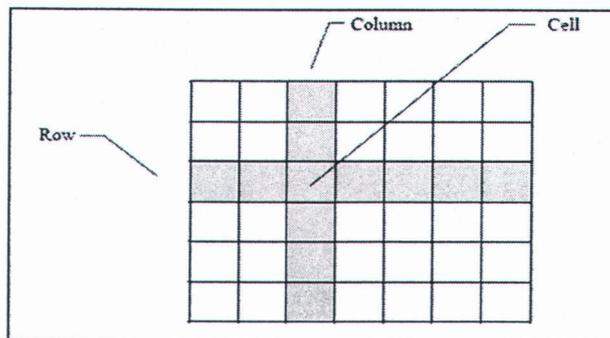
### 2.5 เทคนิคการกระจายหน้าที่เชิงคุณภาพ (Quality Function Deployment : QFD)

QFD เป็นเทคนิคที่ได้รับการพัฒนาขึ้นมาใช้เป็นที่แรกในประเทศญี่ปุ่นที่อุตสาหกรรมของบริษัท มิตซูบิชิ จังหวัดโกเบ ต่อมา ในปี ค.ศ. 1960 บริษัทโตโยต้าประเทศญี่ปุ่นได้นำ QFD มาปรับปรุงและพัฒนาเพื่อใช้ในบริษัทโตโยต้าและบริษัทในเครือ ทำให้ QFD เป็นที่นิยมแพร่หลายในญี่ปุ่นมากขึ้น มีบริษัทยักษ์ใหญ่ในอเมริกาหลายบริษัทได้ให้ความสนใจและนำเทคนิคของ QFD ไปใช้ด้วย

เทคนิคการกระจายหน้าที่เชิงคุณภาพ (Quality Function Deployment : QFD) เป็นเทคนิคที่นำมาช่วยในด้านการวางแผนออกแบบและพัฒนาผลิตภัณฑ์ของผู้ผลิต เพื่อให้สอดคล้องต่อความต้องการของลูกค้า โดยอาศัยความคิดเห็นของลูกค้าและทีมงานเป็นหลัก ซึ่งเป้าหมายของเทคนิคคือ สนองความต้องการของลูกค้าหรือผู้ใช้งาน และบอกถึงระดับของความพึงพอใจของความต้องการตัวนั้นได้ด้วย เพื่อนำไปใช้รวมความคาดหวังจากความต้องการทางกระบวนการซอฟต์แวร์ (Software Process Requirement) โดยนำเมตริกเข้ามาช่วยในการดำเนินงาน และทำการจัดลำดับความสำคัญ อีกทั้งผสมผสานของความต้องการเข้าด้วยกัน เพื่อการพัฒนาความต้องการทางกระบวนการซอฟต์แวร์ (Software Process Requirement) ให้สามารถตอบสนองต่อความคาดหวังของลูกค้าได้เป็นอย่างดี

### 2.5.1 รูปแบบพื้นฐานของ QFD Matrix

เมตริกเป็นเครื่องมือที่ง่ายแต่มีอนุภาพมาก และเป็นหัวใจสำคัญของ QFD (Cohen, 1995 : 57-64) ซึ่งเมตริกเป็นแผนผังรูปสี่เหลี่ยมผืนผ้าที่ถูกแบ่งออกเป็นแถวตั้งหรือสดมภ์ (Columns) หลายๆ สดมภ์ และแถวนอน (Rows) หลายๆ แถว ช่องที่ตัดกันระหว่างสดมภ์และแถวนอนแต่ละคู่ เรียกว่า เซล (Cell) แสดงดังรูปที่ 2.4 เพื่อแสดงความสัมพันธ์ระหว่างปัจจัยในแถวนอนและปัจจัยในสดมภ์



รูปที่ 2.4 ลักษณะแผนผังของเมตริก (Cohen, 1995: 58)

สำหรับตัวอย่างที่แสดงในรูปที่ 2.5 แสดงให้เห็นว่า ปัจจัย “c” ในแถวนอนมีความสัมพันธ์กับปัจจัย “3” ในสดมภ์ อีกตัวอย่างหนึ่ง ดังแสดงในรูปที่ 4.3 แสดงให้เห็นว่าแต่ละปัจจัยในแถวนอนมีความสัมพันธ์กับหลายๆ ปัจจัยในสดมภ์ เช่น ปัจจัย “c” ในแถวนอนมีความสัมพันธ์กับปัจจัย “1” “3” “4” และ “7” ในสดมภ์ เป็นต้น

	1	2	3	4	5	6	7
A							
B							
C			●				
D							
E							
F							

รูปที่ 2.5 การใช้สัญลักษณ์แสดงความสัมพันธ์ระหว่างคู่ของเมตริก 1 คู่ (Cohen, 1995: 59)

	1	2	3	4	5	6	7
A	●				●		
B	●	●					
C	●		●	●			●
D						●	
E				●			
F			●				

รูปที่ 2.6 การใช้สัญลักษณ์แสดงความสัมพันธ์ระหว่างคู่ของเมตริกมากกว่า 1 คู่ (Cohen, 1995: 59)

ลักษณะความสัมพันธ์ที่แสดงในรูปที่ 2.5 และ 2.6 แสดงให้เห็นถึงแต่ละคู่ของความสัมพันธ์มีระดับความสัมพันธ์เท่ากันทุกคู่ แต่โดยทั่วไปแล้วการวิเคราะห์หาแนวทางในการแก้ไขปัญหาและทางเลือกที่เหมาะสมนั้น ความสัมพันธ์ระหว่างปัจจัยแต่ละคู่จะมีระดับความสัมพันธ์ที่มากน้อยต่างกันไป ขึ้นอยู่กับเงื่อนไขและเป้าหมายที่กำหนดไว้ แผนผังเมตริกแบบลำดับความสำคัญ (Prioritization Matrix) เป็นเมตริกที่มีการแบ่งระดับความสัมพันธ์ระหว่างปัจจัยให้มีความแตกต่างกันใน QFD จะใช้ทั้งสัญลักษณ์และตัวเลขแสดงระดับความสัมพันธ์ของปัจจัยแต่ละคู่ดังรูปที่ 2.6 แผนผังใช้สัญลักษณ์แทนระดับความสัมพันธ์ เช่น

ปัจจัย “A” ในแนวนอนมีความสัมพันธ์ในระดับปานกลางกับปัจจัย “1” ในสดมภ์

ปัจจัย “A” ในแนวนอนมีความสัมพันธ์อย่างมากกับปัจจัย “2” และ “4” ในสดมภ์

ปัจจัย “A” ในแนวนอนมีความสัมพันธ์น้อยมากกับปัจจัย “5” ในสดมภ์

การเลือกใช้สัญลักษณ์หรือตัวเลขแสดงระดับความสัมพันธ์ระหว่างปัจจัยนั้น ไม่มีข้อกำหนดตายตัวขึ้นอยู่กับพิจารณาเลือกของผู้ใช้งาน รูปที่ 2.7 เขียนแผนผังของเมตริกใหม่ โดยการใช้ตัวเลขแสดงระดับความสัมพันธ์ระหว่างปัจจัยแทนการใช้สัญลักษณ์ โดยตัวเลขที่แสดงอยู่ด้านล่างของเมตริกแสดงถึงผลบอกราคาของระดับความสัมพันธ์ของแต่ละปัจจัยในสดมภ์ จากรูปที่ 2.7 ผลรวมของระดับความสัมพันธ์ของปัจจัย “4” ในสดมภ์เท่ากับ 30 ซึ่งมิต่ำมากที่สุด จึงถือว่าปัจจัย “4” นี้มีความสำคัญมากที่สุด

	1	2	3	4	5	6	7
A	○	●		●	△		
B					○		
C	△	○		●			●
D		○		●		○	△
E		△		○	●		
F							

- △ Slight or possible relationship
- Moderate relationship
- Strong relationship

รูปที่ 2.7 ลักษณะการใช้สัญลักษณ์เพื่อระบุระดับความสัมพันธ์ในเมตริก (Cohen, 1995: 61)

	1	2	3	4	5	6	7
A	3	9		9	1		
B					3		
C	1			9			9
D		3		9		3	1
E		3		3	9		
F		1					
	4	16	0	30	13	3	10

รูปที่ 2.8 การใช้ตัวเลขเพื่อแสดงระดับความสัมพันธ์ในเมตริก (Cohen, 1995: 62)

### 2.5.2 การสำรวจความต้องการของลูกค้า (The Voice of Customer)

เทคนิคการกระจายหน้าที่เชิงคุณภาพมีพื้นฐานอยู่ตรงการรับฟังเสียงของลูกค้า โดยคำนึงว่า “อะไร” ที่ลูกค้าต้องการจะได้รับจากสินค้าและบริการ ความสำเร็จของเทคนิคการกระจายหน้าที่เชิงคุณภาพจะขึ้นอยู่กับความชัดเจนของเป้าหมายในการดำเนินการและสามารถตอบได้ว่า ลูกค้าที่

แท้จริงหรือผู้ใช้งานคนสุดท้ายที่ได้รับผลกระทบโดยตรงจากเทคนิคการกระจายหน้าที่เชิงคุณภาพคือใคร เนื่องจากเป็นจุดเริ่มต้นในการรวบรวมข้อมูลเพื่อการตัดสินใจว่า ความต้องการที่แท้จริงของลูกค้าที่มีต่อสินค้าและบริการคืออะไร

ขั้นตอนแรกของเทคนิคการกระจายหน้าที่เชิงคุณภาพ คือ การระบุลูกค้าหรือผู้ใช้งาน และดำเนินการรวบรวมหรือจัดกลุ่มลูกค้าให้เหมาะสม เนื่องจากลูกค้าแต่ละกลุ่มอาจมีความต้องการที่แตกต่างกันได้ซึ่งกระบวนการของเทคนิคการกระจายหน้าที่เชิงคุณภาพนั้น ต้องการข้อมูลจากลูกค้าที่แสดงถึงระดับความสำคัญที่ลูกค้าพิจารณาไว้ในแต่ละคุณสมบัติของสินค้าและบริการ ขั้นตอนการหาข้อมูล มีดังนี้

1) กำหนดคุณสมบัติของสินค้าและบริการ (Qualitative Data)

2) หาค่าความสำคัญของแต่ละคุณสมบัติของสินค้าและบริการที่ลูกค้าพิจารณาให้(Quantitative Data) สำหรับเทคนิคการกระจายหน้าที่เชิงคุณภาพต้องการข้อมูลทั้งในส่วนของข้อมูลเชิงคุณภาพและข้อมูลเชิงปริมาณ ซึ่งเรียกคุณสมบัติของสินค้าและบริการ ข้อมูลคุณลักษณะในตัวผลิตภัณฑ์หรือบริการที่ลูกค้าต้องการนี้ว่า ข้อมูลเชิงคุณภาพ (Qualitative Data) และข้อมูลทางตัวเลขที่แสดงระดับความสำคัญที่ลูกค้าให้แก่แต่ละคุณลักษณะของความ ต้องการว่า ข้อมูลเชิงปริมาณ (Quantitative Data)

วิธีในการเข้าถึงเสียงของลูกค้าได้จะต้องดำเนินการรวบรวมข้อมูล ทั้งข้อมูลเชิงปริมาณและข้อมูลเชิงคุณภาพ ซึ่งบอกถึงความต้องการและระดับความสำคัญที่ลูกค้าพิจารณาไว้ในแต่ละความต้องการ โดยใช้เทคนิคในการรวบรวมข้อมูล เช่น การตอบแบบสอบถาม การถามตรง เป็นต้น

### 2.5.3 ประโยชน์ที่ได้รับจากการใช้เทคนิคการกระจายหน้าที่เชิงคุณภาพ

1. เน้นที่ความสำคัญของลูกค้า โดยเทคนิคการกระจายหน้าที่เชิงคุณภาพ นำข้อมูลป้อนกลับจากลูกค้ามาศึกษาและพยายามระบุความต้องการที่แท้จริงของลูกค้าออกมา และค้นหาวิธีการในการที่จะทำให้บรรลุความต้องการดังกล่าวอย่างเหมาะสม ทำให้ความผิดพลาดเนื่องจากผลิตภัณฑ์ใหม่มีคุณภาพไม่ตรงกับความต้องการของลูกค้าลดลง

2. การลดระยะเวลาการออกสู่ตลาดของผลิตภัณฑ์ เนื่องจากเทคนิคการกระจายหน้าที่เชิงคุณภาพช่วยจัดการเกี่ยวกับความไม่แน่นอนในการออกแบบ ทำให้ปัญหาความล่าช้าในการออกแบบผลิตภัณฑ์ใหม่ลดลง อันเนื่องมาจากจำนวนการเปลี่ยนแปลงแบบและการทบทวนแบบลดลง

3. ช่วยจัดโครงสร้างของข้อมูลให้อยู่ในรูปแบบที่รัดกุม ซึ่งสามารถนำมาใช้ได้โดยง่าย และเมื่อมีการเปลี่ยนแปลงคณะทำงาน โครงสร้างของข้อมูลเหล่านี้จะทำให้การทำงานดำเนินต่อไปได้โดยไม่หยุดชะงัก นอกจากนี้เทคนิคการกระจายหน้าที่เชิงคุณภาพยังยึดหยุ่นต่อการเปลี่ยนแปลงเนื่องจากโครงสร้างของเมตริกเอื้ออำนวยต่อการเพิ่มหรือลดขนาดได้ตามข้อมูลที่ป้อนเมตริก

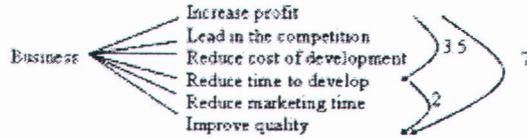
## 2.6 แผนผังการจัดลำดับความสำคัญ (Prioritization scheme)

ระหว่างขั้นตอนดึงความต้องการของโครงการพัฒนากระบวนการซอฟต์แวร์ ความต้องการซอฟต์แวร์ถูกเก็บจาก กลุ่มของผู้มีส่วนได้ส่วนเสียซอฟต์แวร์ซึ่งมีมุมมองความคาดหวังที่จะนำมาแก้ไขปัญหาของเขาเอง ซึ่งในส่วนที่แล้วแนะนำไว้ว่า การเก็บรวบรวมความต้องการซอฟต์แวร์เป็นกลุ่ม จากแต่ละกลุ่มของผู้มีส่วนได้ส่วนเสีย กรอบการทำงานนี้เรียกว่า ความคาดหวัง(Perspective) แต่ละความคาดหวังซอฟต์แวร์ที่ได้เก็บของเขาเหล่านั้นควรที่จะนำไปใช้พัฒนาซอฟต์แวร์

ความสำคัญจากการเก็บความต้องการจากหลายหลายกลุ่มของผู้มีส่วนได้ส่วนเสียซอฟต์แวร์คือมุมมองความคาดหวังต่อซอฟต์แวร์แต่ละด้านจะเป็นอิสระต่อมุมมองด้านอื่นๆ และมุมมองความคาดหวังทุกด้านนั้นจึงความสำคัญทั้งหมด และ ไม่ควรที่จะนำมุมมองความคาดหวังด้านใดๆ ออกไป เพราะมุมมองความคาดหวังต่อซอฟต์แวร์หนึ่งอาจมีผลกระทบสัมพันธ์โดยอ้อมกับมุมมองความคาดหวังอื่น

ปัญหาจะเกิดเมื่อเราให้ความมุมมองคาดหวังเป็นตัวแปรในกระบวนการเก็บความต้องการซอฟต์แวร์ และได้นำสองความคาดหวังสองด้านมาสัมพันธ์กันในความรู้สึกพึงพอใจในสิ่งหนึ่งสิ่ง อาจมีความสัมพันธ์ต่อกันหนึ่ง ค่าที่ใส่ให้แต่ละมุมมองไม่เท่ากัน เพื่อให้เกิดความมั่นใจว่าสิ่งที่จะได้รับเป็นสิ่งสะท้อนออกมาเป็นความต้องการซอฟต์แวร์มากที่สุดจากทุกๆ มุมมองของความคาดหวังซอฟต์แวร์

CMUS จึงถูกพัฒนาเพื่อให้ลำดับถึงความสำคัญและรวมความต้องการซอฟต์แวร์จากมุมมองความคาดหวัง เป็นเครื่องมือเพื่อที่จัดลำดับและค้นหาความต้องการวิกฤตที่มากที่สุดเพื่อที่จะวิเคราะห์จัดหาทรัพยากรที่มีคุณค่าให้พอเพียงกับความต้องการนั้น



(a) Requirements hierarchy and relative dominance

Perspective	Requirements	Priorities		
		Local	Raw Initial Global	Initial Global
Business 3	Increase profit	7	3*7=21	0.1615
	Lead in competition	5	3*5=15	0.1154
	Reduce cost of development	5	3*5=15	0.1154
	Reduce time to develop	2	3*2=6	0.0462
	⋮	⋮	⋮	⋮
Mngt 2	Within budget	7	2*7=14	0.1077
	On schedule	6	2*6=12	0.0923
	High customer satisfaction	4	2*4=8	0.0615
	Increase productivity	2	2*2=4	0.0308
⋮	⋮	⋮	⋮	

(b) Local priorities and initial global priorities

รูปที่ 2.9 การจัดเรียงความสำคัญความต้องการซอฟต์แวร์

ในขั้นตอนเก็บความต้องการและความคาดหวัง ได้เก็บค่าตามลำดับชั้น โดยมีโครงสร้างการเก็บค่าตามลำดับชั้น ตามรูปแบบจัดความสำคัญระดับตามรูปภาพ 2.9 โดยความต้องการและความคาดหวังถูกให้ค่าความสำคัญเริ่มต้นจากผู้มีส่วนได้ส่วนเสียซอฟต์แวร์ ระหว่าง 1 และ 5 ซึ่งค่าความแตกต่างความคาดหวังในโครงสร้างการเก็บค่าตามลำดับชั้นจะมีค่าต่างกัน ดังนั้นควรกำหนดความสัมพันธ์ให้มีสมรรถภาพเช่นเดียวกันมีขั้นตอนดังนี้

**ขั้นตอนที่ 1** สร้างการเชื่อมต่อระหว่างแต่ละคู่ของความคาดหวัง โดยสามารถระบุระดับค่าความสำคัญความคาดหวังหนึ่งสัมพันธ์กับความคาดหวังด้านอื่นนั้น

หากพิจารณาถึงภายในความคาดหวังหรือกลุ่มความต้องการ ความเป็นไปได้ที่บางความต้องการบางตัวสะท้อนค่าความสำคัญที่สับสนแล้วส่วนทางกับกับตัวอื่นเป็นเหตุให้เราไม่สามารถทำให้ได้ความพึงพอใจที่สมบูรณ์ได้ ดังนั้นระดับความพึงพอใจสมควรที่จะปรับได้เช่น

ถ้าลูกค้าสามารถระบุจำนวนร้อยละ (percentage) ของระดับความพึงพอใจ (satisfaction degree) ของความคาดหวัง ( $P_i$ ) นั้นเป็นสิ่งที่ต้องการให้เพิ่มขึ้นเพื่อ ชดเชยสำหรับการลดลงในระดับความพึงพอใจ (satisfaction degree) ของอีกความคาดหวังหนึ่ง ( $P_j$ ) ดังนั้นส่วนสัมพันธ์ความสัมพันธ์ (relative dominance) ของ  $P_i$  บน  $P_j$  แสดงโดยค่า  $rd_{ij}$  ซึ่งสามารถใช้ตามสมการดังนี้

$$rd_{ij} = n_j / n_i$$

เมื่อ  $n_i$  เป็นจำนวนร้อยละ(percentage)ของการเพิ่มขึ้นในระดับความพึงพอใจ(satisfaction degree)ของความคาดหวัง ( $P_i$ ), และ  $n_j$  เป็นจำนวนร้อยละ(percentage)ของที่ลดลงในระดับความพึงพอใจ(satisfaction degree)ของความคาดหวัง( $P_j$ )

**ขั้นตอนที่ 2** หลังการสร้างส่วนสำคัญความสัมพันธ์(relative dominance) ของความคาดหวังหนึ่งบนความคาดหวังด้านอื่น ทำให้ความสำคัญเฉพาะแห่ง(local priority)ของความคาดหวังด้านอื่นๆสามารถถูกคำนวณได้

เพื่อช่วยความเข้าใจโครงสร้าง ความสำคัญเฉพาะแห่ง (local priority) เราให้ความแตกต่างด้านความคาดหวังถูกเรียกว่า ความสำคัญเฉพาะแห่งของความคาดหวัง(perspective local priorities) และเราสมมติว่าเรามีความคาดหวัง  $n$  perspectives  $P_1, P_2, \dots, P_n$  โดยเรียงลำดับค่าความสำคัญของความคาดหวังจากมากไปหาน้อย

เมื่อกำหนดให้  $W_{P_i}$  แสดงตัวเลขค่าความสำคัญของความคาดหวัง  $P_i$  เริ่มแรกเรากำหนด  $W_{P_n}$  (มีความสำคัญของความคาดหวัง  $P_n$  ที่มีค่าต่ำที่สุด) แล้วความสำคัญเฉพาะแห่งของความคาดหวัง (perspective local priorities) ที่เหลือทั้งหมดถูกกำหนดซ้ำตามสมการ

$$W_{P_{i-1}} = W_{P_i} * rd_{i-1,i} \quad \text{สำหรับ } 1 < i \leq n;$$

**ขั้นตอนที่ 3** สำหรับแต่ละความต้องการ การคำนวณค่าเริ่มต้นความสำคัญทั้งหมด(raw initial global priority) เป็นสิ่งสะท้อนทั้ง ความสำคัญเฉพาะแห่ง(local priorities)และ ความสำคัญของความคาดหวัง(priority of the perspective) ที่มันเป็นเจ้าของ

สำหรับความต้องการ  $X$  จากความคาดหวัง  $Y$  การคำนวณค่าความสำคัญทั้งหมดแรกเริ่มดิบ (raw initial global priority) เป็นการคำนวณโดยการคูณ ความสำคัญเฉพาะแห่งของความต้องการ (requirement local priorities) และ ความสำคัญเฉพาะแห่งของความคาดหวัง(perspective local priorities) ตามนี้



$$\text{Raw Initial Global Priority (X)} = \text{Perspective Local Priority (Y)} * \text{Requirement Local Priority (X)}$$

**ขั้นตอนที่ 4** ค่าความสำคัญทั้งหมดแรกเริ่มดิบ (raw initial global priority) ของความต้องการทั้งหมดได้ถูกทำให้เป็นมาตรฐานผ่านความคาดหวังไปได้มา ความสำคัญทั้งหมดแรกเริ่ม (initial global priority)

$$\text{Initial Global Priority (X)} = \frac{\text{Raw Initial Global Priority (X)}}{\sum_{1}^n \text{Raw Initial Global Priority (n)}}$$