

# การพัฒนาและสร้างระบบในส่วนของซอฟต์แวร์ (CMM)

## หลักการพื้นฐานของกระบวนการพัฒนาซอฟต์แวร์

กระบวนการพัฒนาซอฟต์แวร์นั้นวัตถุประสงค์หลักคือพัฒนาซอฟต์แวร์ให้มีคุณภาพที่ดี โดยมีประสิทธิภาพและความสามารถตรงตามแผนที่กำหนดไว้ โดยกระบวนการพัฒนาซอฟต์แวร์ (Software Process) นั้นประกอบด้วยบุคลากร เครื่องมือ และวิธีการดำเนินงานเข้าไว้ด้วยกัน ดังรูปภาพที่ 1 ซึ่งเป็นกลุ่มของกิจกรรม วิธีการ การปฏิบัติ และการแปรรูป ที่บุคคลได้ใช้ในการพัฒนาและรักษาซอฟต์แวร์และผลิตภัณฑ์ที่เกี่ยวข้อง เช่น แผนงานของโครงการ เอกสารการออกแบบ การเขียนโปรแกรม กรณีที่ใช้ทดสอบโปรแกรม และคู่มือสำหรับผู้ใช้ เป็นต้น

ความสามารถของกระบวนการซอฟต์แวร์ (Software process capability) หมายถึงขอบเขตของผลลัพธ์ที่คาดหวังที่สามารถบรรลุได้โดยกระบวนการซอฟต์แวร์ หรือองค์กรสามารถที่จะพยากรณ์ผลลัพธ์ขององค์กรได้ใกล้เคียงกับผลลัพธ์ที่เกิดขึ้นจริง และสามารถทำผลลัพธ์ให้ใกล้เคียงกับความคาดหวังได้มากที่สุดในโครงการต่อไปขององค์กร

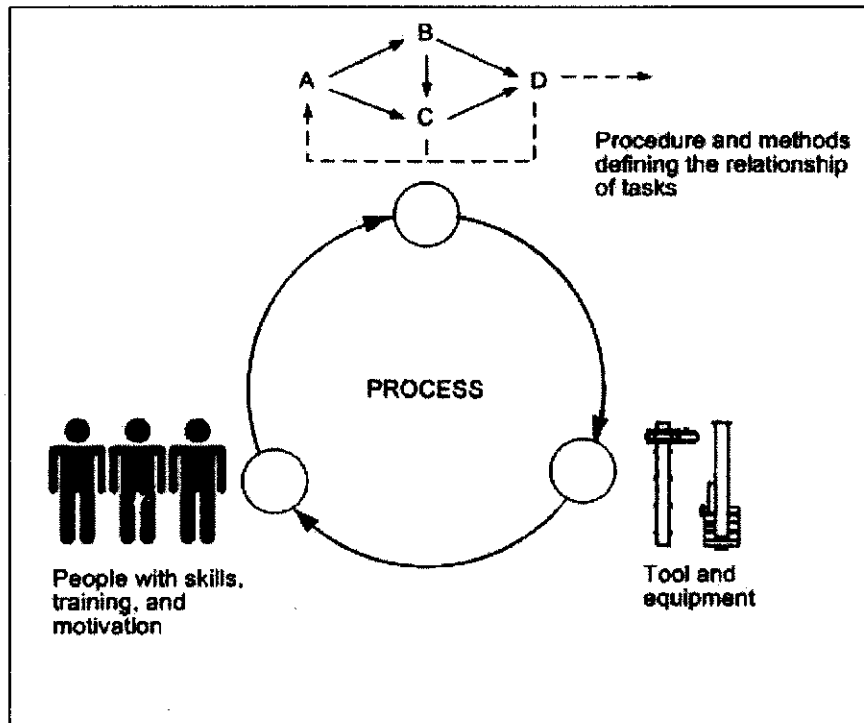
ประสิทธิภาพของกระบวนการซอฟต์แวร์ (Software process performance) นั้นแสดงถึงผลลัพธ์ที่เกิดขึ้นจริงที่บรรลุโดยการดำเนินการตามกระบวนการซอฟต์แวร์

ดังนั้นประสิทธิภาพของกระบวนการซอฟต์แวร์จะมุ่งเน้นในเรื่องของผลที่บรรลุได้ ในขณะที่ความสามารถของกระบวนการซอฟต์แวร์ จะมุ่งเน้นไปที่ผลที่สามารถคาดหวังได้

ความสมบูรณ์ของกระบวนการซอฟต์แวร์ (Software process maturity) หมายถึงความสามารถในการเติบโตในด้านประสิทธิภาพ หรือการเพิ่มประสิทธิภาพในด้านกระบวนการซอฟต์แวร์ ซึ่งต้องมีการสนับสนุนด้านการจัดการ ต้องอาศัยความสนใจอย่างสม่ำเสมอใน

CT 484 บทที่ 45

ระยะยาว ต้องมีโครงสร้างภายในและวัฒนธรรมองค์กรที่จะสนับสนุนวิธีการ แนวทางการปฏิบัติ และขั้นตอนของการที่จะเป็นองค์กรที่สมบูรณ์



รูปภาพที่ 1: Process: holding the pieces together

## Capability Maturity Model (CMM)

SW- CMM (Capability Maturity Model for Software) เป็นแบบจำลองด้านคุณภาพที่เข้บอว่าองค์กรมีระดับความสามารถในการพัฒนาซอฟต์แวร์มากน้อยขนาดไหน เพื่อใช้เป็นแนวทางในการประเมินและปรับปรุงกระบวนการพัฒนาซอฟต์แวร์

CMM เป็นแนวทางหนึ่งในการปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ในองค์กร (Software Process Improvement) ซึ่งถูกคิดค้นซึ่งคิดค้นขึ้นโดยสถาบันวิศวกรรมซอฟต์แวร์ (Software Engineering Institute หรือ SEI) แห่งมหาวิทยาลัยคาร์เนกีเมลลอน ประเทศสหรัฐ

อเมริกา ซึ่งเป็นมหาวิทยาลัยที่ได้รับการยอมรับในระดับสากลว่ามีชื่อเสียงด้านวิศวกรรมคอมพิวเตอร์ระดับโลก CMM ได้อธิบายถึงหลักการ และหลักปฏิบัติในกระบวนการพัฒนาซอฟต์แวร์ และได้ถูกพัฒนาขึ้นมาเพื่อที่จะช่วยให้องค์กรที่พัฒนาซอฟต์แวร์ ได้พัฒนากระบวนการพัฒนาซอฟต์แวร์ขององค์กร จากองค์กรที่มีการพัฒนาซอฟต์แวร์แบบไม่มีระบบระเบียบ ให้เป็นองค์กรที่มีการพัฒนาซอฟต์แวร์แบบมีระบบระเบียบตามมาตรฐานที่ยอมรับ

CMM เป็นแนวทางปฏิบัติ ให้องค์กรพัฒนาซอฟต์แวร์สามารถเลือกแนวทางในการปรับปรุงกระบวนการขององค์กร โดยเลือกได้จากการพิจารณาคุณภาพของกระบวนการที่มีอยู่ และกำหนดกิจกรรม หรือการทำงานที่จำเป็นต่อคุณภาพของซอฟต์แวร์ และการปรับปรุงกระบวนการขององค์กร ซึ่งการจำกัดกลุ่มของกิจกรรม และการทำงานเพื่อให้ได้เป้าหมายที่กำหนดไว้ ทำให้องค์กรสามารถปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ภายในองค์กรได้อย่างมาก และสามารถทำการปรับปรุงต่อเนื่องจนสามารถได้รับกระบวนการพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพ

หลักการของ CMM ก็คือ ความสำเร็จในการทำงานใด ๆ ในอนาคตของบริษัทหรือหน่วยงาน ขึ้นอยู่กับระดับวุฒิภาวะความสามารถในการทำงานของบริษัทหรือหน่วยงานนั้น ในทำนองเดียวกัน วุฒิภาวะความสามารถของบริษัทหรือหน่วยงานนั้น ก็ขึ้นอยู่กับผลการทำงานในอดีตของบริษัทหรือหน่วยงานนั้น

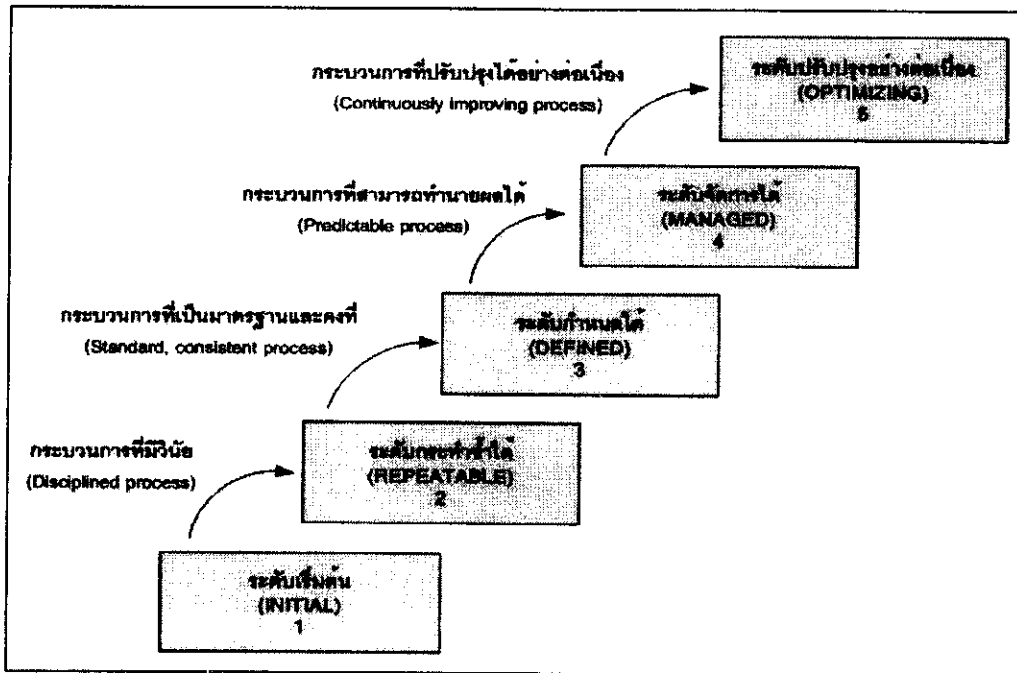
## ระดับของ CMM

CMM ได้กำหนดโครงสร้าง (Framework) สำหรับองค์กรระดับต่างๆ ตามความสามารถ (Capability) และวุฒิภาวะ (Maturity) ของการพัฒนาซอฟต์แวร์ ออกเป็น 5 ระดับ ตามความสามารถ และวุฒิภาวะจากน้อยไปมาก ดังรูปภาพที่ 2 ลูกศรที่แสดงในผังนี้ แสดงถึงประเภทความสามารถของกระบวนการที่เกิดขึ้นในแต่ละชั้นของโครงสร้างวุฒิภาวะ (Maturity Framework) ในองค์กร

### ระดับที่ 1 ระดับเริ่มต้น (The Initial Level)

ระดับเริ่มต้นนี้ เป็นองค์กรที่เริ่มพัฒนาซอฟต์แวร์ซึ่งไม่มีความแน่นอนในการพัฒนาและบำรุงรักษาซอฟต์แวร์ ไม่สามารถควบคุมกระบวนการพัฒนาให้เป็นขั้นตอนตามที่กำหนดไว้ ซึ่งซอฟต์แวร์ถูกพัฒนาขึ้นตามความต้องการของผู้พัฒนาซึ่งความสำเร็จขึ้นอยู่กับประสบการณ์และ

ความสามารถของหัวหน้าโครงการและผู้พัฒนา การทำงานจะไม่ทำตามแผนการที่วางไว้ หรือ ไม่มีการวางแผนการพัฒนา และทดสอบซอฟต์แวร์ ความรู้ที่เกิดขึ้นในโครงการ จะถูกเก็บไว้ที่ตัว หัวหน้าโครงการ หรือทีมนักพัฒนาที่มีความสามารถในโครงการนั้นๆ แต่เมื่อบุคคลเหล่านั้นลา ออกจากโครงการไป ความรู้ความสามารถก็ติดตามบุคคลเหล่านั้นไปด้วย โครงการต่างๆ มักจะ ใช้งบประมาณเกินที่กำหนดไว้ รวมทั้งโครงการเสร็จล่าช้ากว่ากำหนด



รูปภาพที่ 2: ระดับวุฒิภาวะของกระบวนการพัฒนาซอฟต์แวร์

ความสำเร็จขององค์กรระดับที่ 1 ขึ้นอยู่กับความสามารถ และประสบการณ์ของ บุคลากรในองค์กร ดังนั้นในระดับวุฒิภาวะที่ 1 ความสามารถเกิดจากความลักษณะเฉพาะของ ตัวบุคคล ไม่ใช่ขององค์กร

### ระดับที่ 2 ระดับกระทำซ้ำได้ (The Repeatable Level)

ระดับที่ 2 นี้ จะมีการกำหนดนโยบายสำหรับการบริหารโครงการซอฟต์แวร์ และมีกระบวนการใน การพัฒนานโยบายเหล่านั้นด้วย การวางแผน และการจัดการโครงการใหม่ จะใช้ประสบการณ์ที่

เคยเกิดขึ้นในโครงการที่มีลักษณะคล้ายคลึงกัน ความสามารถของกระบวนการ เกิดขึ้นจากการกำหนดระเบียบ กระบวนการพื้นฐานในโครงการ ตามลักษณะของโครงการนั้นๆ กระบวนการที่มีประสิทธิภาพสามารถแบ่งแยกได้ว่า กระบวนการใดที่ต้องทำ กระบวนการใดที่ต้องมีเอกสารอ้างอิง กระบวนการใดต้องมีการวัดผล และกระบวนการใดต้องมีการปรับปรุง

องค์กรระดับวุฒิภาวะที่ 2 มีการควบคุมการจัดการซอฟต์แวร์เบื้องต้น สามารถกำหนดข้อตกลงในโครงการได้จากประสบการณ์จากโครงการที่ผ่านมา และอยู่บนพื้นฐานของความต้องการในโครงการปัจจุบัน ผู้จัดการโครงการสามารถติดตามต้นทุน ระยะเวลา และหน้าที่ที่เกิดขึ้นในการพัฒนาซอฟต์แวร์ได้ มีการควบคุมการพัฒนาซอฟต์แวร์ให้เป็นไปตามความต้องการของซอฟต์แวร์ และกำหนดให้มีผลิตภัณฑ์ที่เกิดจากการทำงาน (Work Product) มีการกำหนดมาตรฐานของโครงการพัฒนาซอฟต์แวร์ และองค์กรสามารถยืนยันได้ว่า ทำตามกระบวนการที่วางไว้

ความสามารถในกระบวนการซอฟต์แวร์ขององค์กรระดับวุฒิภาวะที่ 2 สามารถสรุปได้ว่า เป็นระดับที่มีระเบียบวินัย เนื่องจากมีการวางแผน และติดตามผลในโครงการพัฒนาซอฟต์แวร์ที่แน่นอน และง่ายต่อการทำให้ประสบผลสำเร็จได้อีก กระบวนการต่างๆ ในโครงการจะดำเนินไปตามการควบคุมระบบจัดการโครงการที่มีประสิทธิภาพ เป็นไปตามแผนที่วางไว้ โดยมีพื้นฐานจากผลสำเร็จที่เกิดขึ้นในโครงการที่ผ่านมา

### **ระดับที่ 3 ระดับกำหนดได้ (The Defined Level)**

ระดับที่ 3 เป็นกระบวนการมาตรฐานสำหรับการพัฒนา และการบำรุงรักษาซอฟต์แวร์ภายในองค์กร รวมทั้งกระบวนการจัดการ และวิศวกรรมซอฟต์แวร์ จะถูกบันทึกและรวบรวมไว้เป็นลายลักษณ์อักษร ซึ่งจะถูกนำมาใช้ และเปลี่ยนแปลงตามความเหมาะสม เพื่อที่จะช่วยผู้จัดการโครงการ และผู้พัฒนาทำงานได้อย่างมีประสิทธิภาพ มีการนำโครงการฝึกอบรมให้กับบุคลากรในองค์กร เพื่อมั่นใจได้ว่า พนักงาน และผู้จัดการมีความรู้ ความสามารถตามที่กำหนด เพื่อทำงานที่ได้รับมอบหมายได้อย่างมีประสิทธิภาพ

ความสามารถของกระบวนการซอฟต์แวร์ขององค์กรในระดับวุฒิภาวะที่ 3 สามารถสรุปได้ว่า เป็นระดับที่มีมาตรฐาน และมั่นคง เพราะทั้งวิศวกรรมซอฟต์แวร์ และกิจกรรมเกี่ยวกับการบริหารนั้นคงที่ และสามารถทำซ้ำได้ มีการควบคุมต้นทุน ตารางการทำงาน และหน้าที่ และมี

การติดตามควบคุมคุณภาพซอฟต์แวร์ ความสามารถของกระบวนการนี้ อยู่บนพื้นฐานของกิจกรรม บทบาท และความรับผิดชอบในกระบวนการที่ถูกกำหนดไว้ในองค์กร และใช้ร่วมกัน

#### **ระดับที่ 4 ระดับจัดการได้ (The Managed Level)**

ระดับที่ 4 องค์กรได้กำหนดจุดมุ่งหมายด้านปริมาณเชิงคุณภาพสำหรับผลิตภัณฑ์ซอฟต์แวร์ และกระบวนการที่เกิดขึ้นในองค์กร มีการวัดผลผลิตภัณฑ์ และคุณภาพของกิจกรรมในกระบวนการซอฟต์แวร์ที่สำคัญในโครงการทั้งหมด โดยการวัดผลขององค์กรจะกำหนดมาตรฐานวัดขึ้นเพื่อใช้ในการประเมินกระบวนการและผลิตภัณฑ์ซอฟต์แวร์ของโครงการ เพื่อควบคุมผลิตภัณฑ์และกระบวนการโดยจำกัดความแตกต่างที่เกิดขึ้นในกระบวนการให้มีปริมาณที่ยอมรับได้ ซึ่งสามารถจัดการความเสี่ยงที่เกิดขึ้นจากการเรียนรู้เพื่อนำไปใช้สำหรับโครงการใหม่ๆ ได้อย่างมีประสิทธิภาพ

ความสามารถของกระบวนการซอฟต์แวร์ขององค์กรในระดับวุฒิภาวะที่ 4 สามารถสรุปได้ว่า เป็นระดับที่สามารถพยากรณ์และควบคุมปริมาณได้ เพราะกระบวนการมีการวัด และกระทำในขอบเขตที่สามารถตรวจวัดได้ ทำให้สามารถคาดเดาแนวโน้มของกระบวนการ และปริมาณที่มีคุณภาพภายในขอบเขตที่กำหนดไว้ได้อย่างมีประสิทธิภาพ ทำให้เมื่อเกิดเหตุการณ์ที่ไม่คาดคิดขึ้นเช่นปริมาณผลิตภัณฑ์หรือกระบวนการเกินขอบเขตที่วางไว้ องค์กรจะสามารถแก้ไขสถานการณ์ได้อย่างทันท่วงทีและสามารถหาทางเลือกที่เหมาะสมถูกต้อง ผลิตภัณฑ์ซอฟต์แวร์จะมีคุณภาพสูงตามที่คาดการณ์ไว้ได้

#### **ระดับที่ 5 ระดับปรับปรุงอย่างต่อเนื่อง (The Optimizing Level)**

ระดับที่ 5 ทั้งองค์กรจะมุ่งไปที่การปรับปรุงกระบวนการอย่างต่อเนื่อง องค์กรทราบความสำคัญของจุดอ่อน และจุดแข็งของกระบวนการ โดยมีจุดมุ่งหมายเพื่อป้องกันข้อผิดพลาดที่อาจเกิดขึ้น ข้อมูลที่มีประสิทธิผลของกระบวนการซอฟต์แวร์ถูกนำไปใช้ในการวิเคราะห์ต้นทุนกับผลประโยชน์ที่เกิดขึ้นจากการใช้เทคโนโลยีใหม่ และมีแนวโน้มที่จะเปลี่ยนกระบวนการซอฟต์แวร์ขององค์กร การสร้างแนวทางปฏิบัติด้านวิศวกรรมซอฟต์แวร์ที่ดีที่สุดได้ถูกกำหนดไว้ และถูกถ่ายทอดไปทั่วทั้งองค์กร

ทีมโครงการพัฒนาซอฟต์แวร์ในองค์กรระดับที่ 5 สามารถวิเคราะห์ข้อผิดพลาด เพื่อหาสาเหตุที่เกิดขึ้น กระบวนการซอฟต์แวร์ถูกประเมินเพื่อป้องกันข้อผิดพลาดที่ทราบอยู่ ไม่ให้เกิดขึ้นอีก และบทเรียนที่ได้รับสามารถนำไปใช้ในโครงการอื่นได้

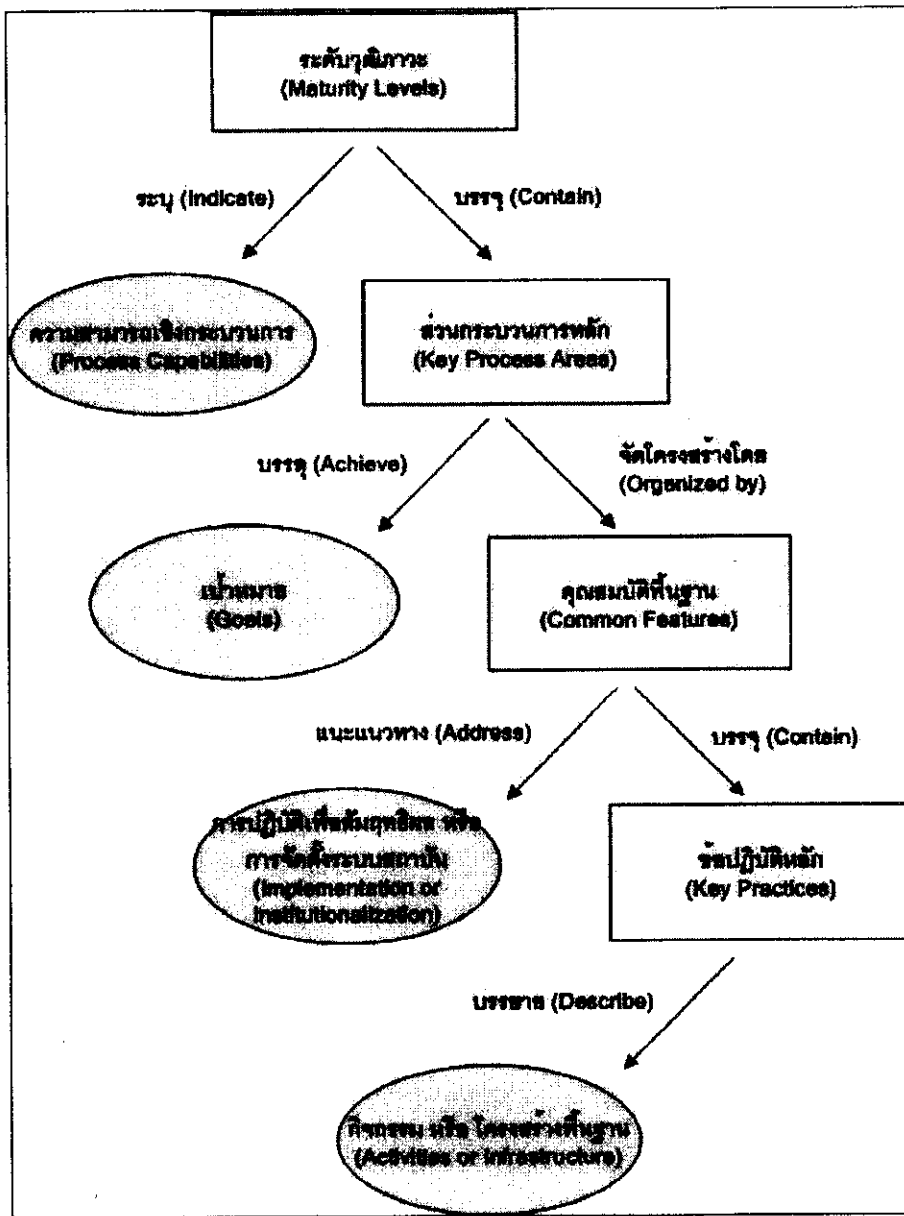
ความสามารถของกระบวนการซอฟต์แวร์ขององค์กรในระดับวุฒิภาวะที่ 5 สามารถสรุปได้ว่า มีการปรับปรุงอย่างต่อเนื่อง เพราะองค์กรในระดับที่ 5 ต้องปรับปรุงขอบเขตความสามารถของกระบวนการซอฟต์แวร์อย่างต่อเนื่อง การปรับปรุงนี้ สามารถทำได้ทั้งการเพิ่มความสามารถให้กับกระบวนการที่มีอยู่ หรือโดยการนำเทคโนโลยี หรือวิธีการใหม่มาใช้ ซึ่งการปรับปรุงเทคโนโลยี และกระบวนการ จะถูกวางแผน และจัดการ เป็นเสมือนกิจกรรมพื้นฐานทางธุรกิจ

## โครงสร้างภายในของระดับการพัฒนา

ในแต่ละระดับของการพัฒนา เราจะแบ่งออกเป็นส่วนต่างๆ ขอบเขตของแต่ละระดับการพัฒนา ดังรูปภาพที่ 3

### KPA (Key Process Area)

นอกเหนือจากระดับวุฒิภาวะแล้ว ในแต่ละระดับวุฒิภาวะจะมี KPA ที่บ่งบอกว่า แต่ละองค์กรควรมีจุดมุ่งหมายในการปรับปรุงกระบวนการซอฟต์แวร์ขององค์กรอย่างไร KPA จะระบุสิ่งที่ต้องกระทำ เพื่อให้ได้บรรลุถึงระดับวุฒิภาวะนั้น ซึ่งแต่ละ KPA จะระบุส่วนประกอบของกิจกรรมที่เกี่ยวข้องกัน เมื่อได้มีการปฏิบัติสะสมกัน จะทำให้บรรลุจุดมุ่งหมายสำคัญในการเพิ่มความสามารถเชิงกระบวนการ KPA จะถูกกำหนดไว้ในแต่ละระดับวุฒิภาวะ ดังรูปภาพที่ 4 แสดงความก้าวหน้าในการบรรลุจุดประสงค์ขององค์กรในแต่ละระดับ ซึ่งแต่ละ KPA อาจแตกต่างกันไปตามโครงการที่มีพื้นฐานหรือสภาพแวดล้อมที่ต่างกัน อย่างไรก็ตามองค์กรต้องทำให้ได้จุดประสงค์ทั้งหมดของ KPA นอกจากนี้องค์กรในระดับที่สูงต้องรักษามาตรฐานของ KPA ในระดับที่ต่ำกว่าไว้ได้ เช่น ถ้าองค์กรอยู่ในระดับที่ 3 องค์กรนั้นจะต้องมีทุก KPA ที่กำหนดในระดับที่ 2 และ 3



รูปภาพที่ 3: โครงสร้างภายในของระดับ

KPA ในระดับที่ 2 มุ่งประเด็นไปที่การกำหนดการควบคุมการจัดการ โครงการขั้นพื้นฐานของโครงการพัฒนาซอฟต์แวร์ KPA ในระดับนี้ มีดังต่อไปนี้



- การจัดการความต้องการ (Requirements Management) มีวัตถุประสงค์เพื่อสร้างความเข้าใจร่วมกันระหว่างลูกค้าและผู้พัฒนาซอฟต์แวร์ โดยผู้พัฒนาซอฟต์แวร์จะรวบรวมความต้องการของลูกค้าสร้างเอกสารเพื่อใช้เป็นข้อตกลงร่วมกัน
- การวางแผนโครงการซอฟต์แวร์ (Software Project Planning) มีวัตถุประสงค์เพื่อสร้างแผนงานพัฒนาซอฟต์แวร์ที่น่าเชื่อถือ สำหรับการปฏิบัติงานและจัดการโครงการซอฟต์แวร์
- การติดตามโครงการพัฒนาซอฟต์แวร์และเฝ้ามอง (Software Project Tracking and Oversight) มีวัตถุประสงค์เพื่อช่วยให้ฝ่ายบริหาร สามารถมองเห็นความก้าวหน้าของโครงการ สามารถกำหนดการปฏิบัติงาน หรือแก้ไขปรับปรุง เพื่อเพิ่มประสิทธิภาพ หากความก้าวหน้าของโครงการเบี่ยงเบนไปจากแผนงานที่ได้กำหนดไว้ซึ่งก่อให้เกิดผลกระทบต่อการพัฒนาซอฟต์แวร์
- การจัดการผู้รับช่วงสัญญาซอฟต์แวร์ (Software Subcontract Management) มีวัตถุประสงค์เพื่อเลือกผู้รับจ้างงานซอฟต์แวร์ที่มีคุณสมบัติเหมาะสม และสามารถจัดการเกี่ยวกับผู้รับเหมาช่วงการพัฒนาซอฟต์แวร์ได้อย่างมีประสิทธิภาพ
- การประกันคุณภาพซอฟต์แวร์ (Software Quality Assurance) มีวัตถุประสงค์เพื่อตรวจสอบและประกันคุณภาพของการพัฒนาซอฟต์แวร์ เป็นไปตามมาตรฐานที่กำหนดไว้
- การจัดการส่วนประกอบของซอฟต์แวร์ (Software Configuration Management) มีวัตถุประสงค์เพื่อสร้างและรักษาความถูกต้องเป็นอันหนึ่งอันกันของผลิตภัณฑ์ (Product) จากโครงการซอฟต์แวร์ ตลอดช่วงอายุของซอฟต์แวร์ในโครงการนั้น

KPA ในระดับวุฒิภาวะที่ 3 ได้มุ่งประเด็นไปทั้งองค์กร โดยการสร้างโครงสร้างพื้นฐานภายในองค์กร เพื่อส่งเสริมประสิทธิภาพของงานวิศวกรรมซอฟต์แวร์และการจัดการกระบวนการภายในโครงการต่าง ๆ KPA ในระดับนี้ คือ

- การจัดการกระบวนการเชิงปริมาณ (Quantitative Process Management) มีวัตถุประสงค์เพื่อควบคุมการดำเนินงานตามกระบวนการของโครงการซอฟต์แวร์ ด้วยการพิจารณาค่าที่วัดได้ในเชิงปริมาณ
- การจัดการคุณภาพซอฟต์แวร์ (Software Quality Management) มีวัตถุประสงค์เพื่อพัฒนาคุณภาพของผลิตภัณฑ์ซอฟต์แวร์จากโครงการในเชิงปริมาณ และเพื่อการบรรลุเป้าหมายทางคุณภาพที่กำหนดไว้

**KPAในระดับที่ 5** ครอบคลุมเนื้อหาทั้งในระดับโครงการ และระดับองค์กรกำหนดขึ้นเพื่อปรับปรุงกระบวนการซอฟต์แวร์อย่างต่อเนื่องและสามารถวัดค่าได้ โดยKPA ในระดับนี้คือ

- การป้องกันข้อบกพร่อง (Defect Prevention) มีวัตถุประสงค์เพื่อค้นหา และชี้ให้เห็นสาเหตุของข้อบกพร่องต่าง ๆ และป้องกันความผิดพลาดเหล่านั้นไม่ให้เกิดขึ้นอีก
- การจัดการการเปลี่ยนแปลงเทคโนโลยี (Technology Change Management) มีวัตถุประสงค์เพื่อชี้ให้เห็นประโยชน์จากเทคโนโลยีใหม่ ๆ ตัวอย่าง เช่น เครื่องมือ วิธีการ และกระบวนการต่าง ๆ รวมถึงการนำเทคโนโลยีเหล่านั้นเข้ามาใช้ในองค์กรได้อย่างเหมาะสม สำหรับในโลกที่มีการเปลี่ยนแปลงตลอดเวลา
- การจัดการการเปลี่ยนแปลงกระบวนการ (Process Change Management) มีวัตถุประสงค์เพื่อปรับปรุงกระบวนการซอฟต์แวร์ที่ใช้อยู่ในองค์กรอย่างต่อเนื่อง เพื่อประโยชน์ในการปรับปรุงคุณภาพของซอฟต์แวร์ รวมทั้งการเพิ่มความสามารถในการผลิต และลดระยะเวลาของการพัฒนาผลิตภัณฑ์ลง

ในแต่ละระดับวุฒิภาวะประกอบด้วย KPA (Key Process Area) ในแต่ละ KPA จะมีองค์ประกอบ 5 ส่วนเรียกว่า คุณสมบัติพื้นฐาน (Common Features) ซึ่งคุณสมบัติพื้นฐานนี้จะระบุแนวทางปฏิบัติ (Key Practices) หรือ ข้อปฏิบัติสำคัญหลายข้อให้ปฏิบัติตาม และเนื้อหาในข้อปฏิบัติเหล่านี้จะแนะนำกิจกรรม และโครงสร้างพื้นฐาน ที่ต้องทำหรือจัดให้มีภายในองค์กร เพื่อให้ได้บรรลุถึงเป้าหมาย (Goals) ของ KPA

## คุณสมบัติพื้นฐาน (Common Features)

คุณสมบัติพื้นฐาน คือ กลุ่มของข้อปฏิบัติที่จัดไว้เป็นหมวดหมู่ จำแนกตามประโยชน์ของข้อปฏิบัติ เพื่อช่วยให้ง่ายต่อการเข้าใจ ตลอดจนใช้เพื่อบ่งบอกว่า องค์กรสามารถสร้างและรักษา KPA ได้อย่างมีประสิทธิภาพหรือไม่ คุณสมบัติพื้นฐานแบ่งได้เป็น 5 ส่วน ได้แก่

พันธะสัญญาเพื่อการปฏิบัติ (Commitment to Perform)	พันธะสัญญาเพื่อการปฏิบัติ คือ การกระทำต่างๆ ที่องค์กรต้องปฏิบัติ เพื่อแน่ใจว่า กระบวนการถูกสร้างขึ้นและรักษาไว้ได้อย่างต่อเนื่อง ข้อปฏิบัติในกลุ่มนี้ จึงเกี่ยวข้องกับการกำหนดนโยบาย และการสนับสนุนจากผู้บริหารระดับสูงในองค์กร
ความสามารถเพื่อการปฏิบัติ (Ability to Perform)	ความสามารถเพื่อการปฏิบัติ คือ เงื่อนไขพื้นฐานต่างๆ ที่โครงการ หรือ องค์กรต้องมี เพื่อปฏิบัติตามกระบวนการซอฟต์แวร์ได้อย่างสมบูรณ์ ข้อปฏิบัติในกลุ่มนี้ จึงเกี่ยวข้องกับทรัพยากรของโครงการ และองค์กร รวมถึงการฝึกอบรม
กิจกรรมที่ต้องปฏิบัติ (Activities Performed)	กิจกรรมที่ต้องปฏิบัติ คือ กิจกรรม บทบาท และขั้นตอนต่างๆ ที่จำเป็นในการทำให้บรรลุ KPA คุณสมบัติพื้นฐานกลุ่มนี้ จึงจะเกี่ยวข้องกับการสร้างแผน หรือกำหนดแนวทางขั้นต้น วิธีปฏิบัติงาน การติดตามความก้าวหน้าของงาน และการแก้ไขข้อผิดพลาดที่เกิดขึ้น
การวัด และการวิเคราะห์ (Measurement and Analysis)	การวัดและการวิเคราะห์ คือ ข้อปฏิบัติต่างๆ ที่กล่าวถึงการวัดค่าที่จำเป็นสำหรับการพิจารณาสถานะที่เกี่ยวข้องกับกระบวนการ ค่าที่วัดได้จากการวัดค่าเหล่านี้ ถูกใช้เพื่อควบคุม และปรับปรุงกระบวนการ
การตรวจสอบ (Verifying Implementation)	การตรวจสอบ คือ ขั้นตอนต่างๆ ที่สร้างความมั่นใจว่า กิจกรรมต่างๆ ถูกปฏิบัติได้อย่างสอดคล้องกับกระบวนการ ข้อปฏิบัติในกลุ่มนี้ มักประกอบด้วย การทวนสอบ และการตรวจสอบ โดยฝ่ายจัดการ และฝ่ายประกันคุณภาพซอฟต์แวร์

## ข้อปฏิบัติหลัก (Key Practices)

แต่ละ KPA ได้อธิบายนิยามของข้อปฏิบัติที่พึงปฏิบัติเพื่อให้ได้บรรลุเป้าหมายที่วางไว้ ข้อปฏิบัติหลัก ได้อธิบายถึงโครงสร้างพื้นฐาน และกิจกรรมที่ต้องปฏิบัติ โดยเน้นถึงการนำไปใช้

คุณสมบัติพื้นฐานแต่ละกลุ่มจะประกอบไปด้วยข้อปฏิบัติสำคัญ ซึ่งบรรยายถึงกิจกรรมที่ต้องกระทำ รวมทั้งโครงสร้างพื้นฐานต่างๆ ที่องค์กรและโครงการต้องมี ข้อปฏิบัติสำคัญเหล่านี้จะบอกว่า “อะไร” บ้างที่ต้องทำ แต่จะไม่ได้เจาะจงว่าจะต้องกระทำ “อย่างไร” เพื่อให้องค์กรได้สร้างทางเลือกเพื่อปฏิบัติเอง โดยคำนึงถึงเป้าหมายตาม KPA ในระดับที่เป็นประโยชน์ต่อองค์กรได้

CMM เป็นแบบจำลองที่ใช้ในการปรับปรุงตัวกระบวนการที่ใช้ในพัฒนาซอฟต์แวร์ เพื่อพัฒนากระบวนการพัฒนาซอฟต์แวร์ที่ดีเพื่อให้ได้ซอฟต์แวร์ที่มีคุณภาพ ซึ่งส่งผลให้เกิดข้อผิดพลาดต่างๆ น้อยลง องค์กรพัฒนาซอฟต์แวร์ที่มีความเติบโตในระดับสูงย่อมสร้างความมั่นใจให้กับลูกค้าและโอกาสที่จะพัฒนาผลิตภัณฑ์ที่มีคุณภาพสูงกว่าองค์กรที่ยังไม่เติบโตหรือเติบโตอยู่ในระดับที่ต่ำกว่า ดังนั้น CMM จึงเป็นแนวทางในการปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ โดยผู้บริหารองค์กรมีข้อมูลเกี่ยวกับประสิทธิภาพขององค์กร และสามารถประมาณต้นทุนในการผลิต ประมาณเวลา ได้อย่างใกล้เคียงกับความเป็นจริง และสามารถจัดการกับงานหรือกิจกรรมต่างๆ ได้อย่างเป็นระบบ