

## แนวทางการนำแบบจำลองวุฒิความสามารถด้านซอฟต์แวร์มาใช้ในองค์กร ครรชิต มาลัยวงศ์

ปัจจุบันนี้คุณภาพซอฟต์แวร์มีความสำคัญอย่างยิ่งต่อองค์กรทุกแห่งทั้งนี้เพราะการใช้คอมพิวเตอร์และซอฟต์แวร์ในการปฏิบัติและการบริหารงานได้กลายเป็นเรื่องจำเป็นและหลีกเลี่ยงไม่ได้ไปแล้ว ผู้เกี่ยวข้องกับการจัดซื้อจัดหาซอฟต์แวร์มาใช้จึงจำเป็นต้องสนใจในเรื่องนี้อย่างจริงจังและต่อเนื่อง ในอดีต การพัฒนาซอฟต์แวร์ส่วนใหญ่ไม่ได้ใช้ระบบหรือวิธีการที่เป็นมาตรฐาน ไม่มีการควบคุมคุณภาพที่เหมาะสม ส่งผลให้เกิดผลร้ายต่อเศรษฐกิจและความก้าวหน้าขององค์กรต่าง ๆ อย่างมาก อาทิ

- ซอฟต์แวร์ที่ผิดพลาดทำให้เกิดปัญหาร้ายแรงต่อการดำเนินงานขององค์กร หรือบางครั้งเป็นอันตรายต่อชีวิต อาทิ ซอฟต์แวร์กำหนดตำแหน่งพิกัดทางภูมิศาสตร์ในเครื่องบินโดยสารอาจทำงานผิดพลาดจนทำให้เครื่องบินหลงทางเข้าไปในดินแดนของประเทศอื่นและอาจเกิดความเข้าใจผิดจนถูกยิงตกได้ หรือ ซอฟต์แวร์สำหรับควบคุมอุปกรณ์ในห้องไอซียูอาจทำงานผิดและทำให้ผู้ป่วยถึงแก่ความตายได้
- การจัดซื้อซอฟต์แวร์สำเร็จที่ทำงานไม่ได้ตามความต้องการอาจทำให้ต้องสูญเสียงบประมาณเป็นจำนวนมาก และอาจทำให้บริษัทที่ซื้อซอฟต์แวร์นั้นมาใช้ประสบปัญหาด้านการเงินได้
- ซอฟต์แวร์ที่ไม่ได้รับการทดสอบอย่างเหมาะสมอาจมีช่องโหว่ให้ผู้อื่นบ่อนทำลายด้วยการส่ง ไวรัสคอมพิวเตอร์เข้ามารบกวน ลบแฟ้มข้อมูลสำคัญ หรือสร้างความสับสนให้แก่ผู้ใช้ได้

ด้วยเหตุนี้เองก่อนที่บริษัทหรือหน่วยงานต่าง ๆ จะจัดหาหรือจัดซื้อซอฟต์แวร์มาใช้จึงมีความจำเป็นที่จะต้องศึกษาตรวจสอบซอฟต์แวร์ว่ามีคุณภาพหรือไม่ หรือในกรณีที่ต้องการว่าจ้างบริษัทซอฟต์แวร์มาพัฒนาซอฟต์แวร์ให้ ก็ต้องตรวจสอบว่าบริษัทนั้นใช้กระบวนการพัฒนาซอฟต์แวร์ที่มีคุณภาพหรือไม่ อีกนัยหนึ่งก็คือตรวจสอบวุฒิภาวะในการพัฒนาซอฟต์แวร์ของบริษัทนั้นก่อนนั่นเอง

บทความนี้อธิบายแนวทางที่หน่วยงานหรือบริษัทผู้ผลิตซอฟต์แวร์ควรใช้ในการพัฒนาระดับวุฒิภาวะของการผลิตซอฟต์แวร์ให้สูงขึ้นตามแบบจำลอง CMM (Capability Maturity Model) โดยจะเน้นเพียงวุฒิภาวะระดับสองซึ่งเป็นขั้นต้นสำหรับการก้าวไปสู่ความเป็นหน่วยงานหรือบริษัทที่มีศักยภาพและความสามารถในการพัฒนาซอฟต์แวร์ที่มีคุณภาพและน่าเชื่อถือ

แนะนำแบบจำลอง CMM

แบบจำลองสำหรับวัดระดับวุฒิภาวะความสามารถในการพัฒนาซอฟต์แวร์ CMM นี้มีจุดเริ่มต้นมาจากการที่ผู้ใช้ซอฟต์แวร์จำนวนมากในอดีตต้องประสบปัญหาเกี่ยวกับซอฟต์แวร์ที่ไม่มีคุณภาพดังกล่าวมาแล้วข้างต้น หนึ่งในผู้ที่ได้รับผลกระทบมากที่สุดคือกระทรวงกลาโหม สหรัฐอเมริกา ซอฟต์แวร์ที่ทำงานผิดพลาดหลายชุดได้ทำให้โครงการทางทหารและทางด้านอวกาศประสบความล้มเหลว ดังนั้นกระทรวงกลาโหม สหรัฐอเมริกา จึงได้เชิญชวนสถาบันการศึกษาที่สนใจงานวิจัยด้านซอฟต์แวร์ให้ยื่นข้อเสนอเพื่อจัดตั้งสถาบันค้นคว้าวิจัยทางด้านวิศวกรรมซอฟต์แวร์ขึ้น ผลที่สุดท้าย กระทรวงฯ ได้เลือกให้มหาวิทยาลัยคาร์เนกี เมลลอน เป็นผู้รับผิดชอบก่อตั้ง สถาบันวิศวกรรมซอฟต์แวร์ (Software Engineering Institute หรือ SEI) ขึ้น โดยกระทรวงกลาโหมเป็นผู้อุดหนุนด้านงบประมาณการดำเนินงานทั้งหมด

สถาบัน SEI ได้ริเริ่มพัฒนาแบบจำลอง CMM ขึ้นเพื่อใช้วัดระดับวุฒิภาวะความสามารถในด้านต่าง ๆ ของหน่วยงานที่เกี่ยวข้องกับซอฟต์แวร์ อาทิ แบบจำลอง CMM ทางด้านบุคลากร แบบจำลอง CMM ทางด้านการจัดซื้อ แบบจำลอง CMM ทางด้านการพัฒนาซอฟต์แวร์ สำหรับในที่นี้จะนำแนวคิดเฉพาะแบบจำลองทางด้านการพัฒนาซอฟต์แวร์เท่านั้นมาอธิบาย อย่างไรก็ตามแบบจำลองอื่น ๆ ล้วนมีแนวคิดที่มีลักษณะเดียวกันหมด

แนวคิดหลักของแบบจำลอง CMM ทางด้านการพัฒนาซอฟต์แวร์ก็คือ วุฒิภาวะความสามารถในการพัฒนาซอฟต์แวร์ของหน่วยงานหรือบริษัทอาจแบ่งออกได้เป็นห้าระดับ เรียงกันไปจากระดับที่มีวุฒิภาวะต่ำสุดไปถึงระดับสูงสุด ดังนี้

1. ระดับเริ่มต้น (Initial Level) เป็นระดับที่หมายถึงหน่วยงานหรือบริษัทซอฟต์แวร์ทั่วไปที่พัฒนาซอฟต์แวร์กันอย่างไม่เป็นระบบ ไม่ได้ใช้หลักการจัดการโครงการมาใช้ ผลงานโดยทั่วไปมีคุณภาพไม่ค่อยดีนัก หน่วยงานและบริษัทเหล่านี้ต้องพึ่งพาอาศัยบุคลากรที่มีความสามารถสูง จึงจะพัฒนาซอฟต์แวร์สำเร็จ ถ้าหากบุคลากรเหล่านั้นไม่ปฏิบัติงานด้วยแล้ว หน่วยงานและบริษัทก็จะไม่สามารถพัฒนาซอฟต์แวร์ได้
2. ระดับทำซ้ำได้ (Repeatable Level) เป็นระดับที่หมายถึงหน่วยงานหรือบริษัทซอฟต์แวร์ที่นำหลักการบริหารจัดการโครงการมาใช้ในการพัฒนาซอฟต์แวร์ การพัฒนามีขั้นตอนที่กำหนดไว้อย่างเหมาะสมจากประสบการณ์ของหัวหน้าโครงการแต่ละคน มีการวางแผนงานอย่างรอบคอบ มีการประสานสัมพันธ์กับลูกค้าผู้เป็นเจ้าของงานอย่างเหมาะสม และมีการประกันคุณภาพซอฟต์แวร์ หน่วยงานและบริษัทที่มีวุฒิภาวะที่ระดับนี้จะสามารถพัฒนาซอฟต์แวร์ที่มีลักษณะเนื้อหาแบบเดียวกันให้ประสบผลสำเร็จได้อย่างต่อเนื่องซ้ำหลายหน

3. ระดับมีการนิยาม (Defined Level) เป็นระดับที่หมายถึงหน่วยงานหรือบริษัทซอฟต์แวร์ที่อยู่ในระดับสองแล้วได้เริ่มนำวิธีปฏิบัติงานที่ดีและเหมาะสมของโครงการต่าง ๆ มาพิจารณากลับกรองให้เป็นวิธีปฏิบัติของหน่วยงานเอง มีการกำหนดแนวทางให้หัวหน้าโครงการนำแนวปฏิบัติของหน่วยงานไปปรับใช้กับโครงการซอฟต์แวร์ต่าง ๆ ได้อย่างเหมาะสม นั่นก็คือหน่วยงานและบริษัทเริ่มมีระเบียบวิธีการปฏิบัติงานเป็นมาตรฐานของตนเอง
4. ระดับมีการจัดการ (Managed Level) เป็นระดับที่หมายถึงหน่วยงานหรือบริษัทซอฟต์แวร์ที่อยู่ระดับสามแล้วได้เริ่มนำข้อมูลเกี่ยวกับการปฏิบัติงานในด้านต่าง ๆ เช่น เวลาที่ใช้ในการทำงานตามขั้นตอนต่าง ๆ จำนวนข้อบกพร่องที่พบในการเขียนโปรแกรม มาใช้ในการควบคุมทางสถิติโดยหลักการ Statistical Process Control
5. ระดับปรับปรุงให้เหมาะสมที่สุด (Optimizing Level) เป็นระดับที่หมายถึงหน่วยงานหรือบริษัทซอฟต์แวร์ที่อยู่ในระดับที่สี่แล้ว ได้นำหลักการจัดการคุณภาพมาใช้เพื่อป้องกันไม่ให้เกิดข้อบกพร่องในการปฏิบัติงาน และเพื่อให้หน่วยงานและบริษัทสามารถปรับเปลี่ยนตัวเองให้สอดคล้องกับการเปลี่ยนแปลงทางด้านเทคโนโลยีได้

สถาบัน SEI ได้สังเคราะห์วุฒิภาวะความสามารถในการพัฒนาซอฟต์แวร์ทั้งห้าระดับนี้จากการศึกษาวิธีการและผลการพัฒนาซอฟต์แวร์ของหน่วยงานและบริษัทซอฟต์แวร์จำนวนมาก โดยนักวิจัยของสถาบันได้วิเคราะห์รายละเอียดขั้นตอนการพัฒนาซอฟต์แวร์ รวมทั้งกิจกรรมต่างๆ ในแต่ละขั้นตอนแล้วกำหนดขึ้นเป็นกลุ่มกิจกรรมสำคัญ หรือ Key Process Area (KPA) รวมทั้งสิ้น 18 กลุ่ม ในแต่ละกลุ่มได้กำหนดหัวข้อกิจกรรมสำคัญ หรือ Key Practice ที่ต้องดำเนินการ รวมทั้งสิ้น 316 กิจกรรม

หน่วยงานและบริษัทซอฟต์แวร์สามารถประเมินตัวเองว่ามีวุฒิภาวะความสามารถในการพัฒนาซอฟต์แวร์อยู่ที่ระดับใดได้โดยเข้าสู่กระบวนการประเมินที่เรียกว่า CMM Based Assessment/Internal Process Improvement การประเมินนี้มีวัตถุประสงค์เพื่อตรวจสอบว่าหน่วยงานหรือบริษัทมีจุดอ่อนที่ต้องแก้ไขในจุดใดบ้าง อีกนัยหนึ่งก็คือการตรวจสอบว่าหน่วยงานหรือบริษัทได้ดำเนินงานตามกลุ่มกิจกรรมสำคัญที่กำหนดไว้ในแบบจำลอง CMM ครบถ้วนหรือไม่ หากไม่ครบก็จะต้องหาทางปรับปรุงดำเนินงานตามกิจกรรมสำคัญนั้นให้ครบ เพราะกิจกรรมสำคัญเหล่านั้นได้รับการพิสูจน์แล้วว่ามีความจำเป็นสำหรับการพัฒนาซอฟต์แวร์ที่มีคุณภาพ

พิจารณาในอีกแง่มุมหนึ่งคือจากฝ่ายผู้ต้องการว่าจ้างหรือใช้บริการการพัฒนาซอฟต์แวร์ของหน่วยงานหรือบริษัทนั้น หากทราบว่าหน่วยงานหรือบริษัทนั้นมีวุฒิภาวะความสามารถอยู่

ระดับใด ก็จะสามรถคาดการณ์ได้ว่าหน่วยงานหรือบริษัทนั้นจะมีผลการดำเนินงานที่มีคุณภาพมากน้อยเพียงใดด้วย ความรู้นี้จะใช้ในการตัดสินใจได้ง่ายขึ้นว่าสมควรว่าจ้างหน่วยงานหรือบริษัทนั้นหรือไม่

## การริเริ่มนำแบบจำลอง CMM มาใช้

ความคิดที่จะนำแบบจำลอง CMM มาใช้เพื่อเป็นแนวทางในการปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ให้มีวุฒิภาวะที่สูงขึ้นนั้นจำเป็นต้องได้รับการสนับสนุนจากผู้บริหารของหน่วยงานเป็นอันดับแรก การสร้างวุฒิภาวะนั้นไม่เหมือนกับการซื้อรถยนต์มาใช้ เพราะแม้ว่ารถยนต์จะมีราคาสูงแต่การใช้งานก็ไม่ต้องเตรียมตัวมากนัก อีกทั้งการมีรถยนต์ก็ไม่ได้กระทบต่อวิถีปฏิบัติงานปกติของหน่วยงานแต่อย่างใด ในกรณีของ CMM นั้นการใช้ต้องมีการเปลี่ยนแปลงนโยบายต้องปรับกระบวนการทำงาน ต้องสร้างวัฒนธรรมองค์กรขึ้นมาใหม่ ดังนั้นจึงมีผลกระทบต่อหน่วยงานมาก หากผู้บริหารไม่เห็นพ้องด้วยกับความคิดนี้แล้ว การนำ CMM มาใช้ก็จะประสบกับการต่อต้านได้ง่าย และการใช้ก็จะไม่ประสบผลสำเร็จอย่างที่ต้องการ

เมื่อผู้บริหารเห็นด้วย และยินดีสนับสนุนในด้านการลงทุน รวมทั้งจัดสรรงบประมาณให้ดำเนินการแล้ว ผู้อำนวยการศูนย์คอมพิวเตอร์ซึ่งในที่นี้จะสมมุติว่าทำหน้าที่ดูแลงานพัฒนาซอฟต์แวร์ของหน่วยงาน ก็สมควรตั้งเป็นโครงการ CMM ขึ้นและพิจารณาแต่งตั้งผู้ที่เหมาะสมให้ทำหน้าที่รับผิดชอบเป็นหัวหน้าโครงการ

ผู้เป็นหัวหน้าโครงการ CMM ควรจะมีความรู้หรือได้รับการอบรมทางด้าน CMM มาก่อนแล้ว หากยังไม่ได้รับการอบรมอย่างเป็นทางการ หัวหน้าโครงการก็ต้องพิจารณาคัดเลือกพนักงานที่มีประสบการณ์ในการพัฒนาซอฟต์แวร์เข้ารับการฝึกอบรม Introduction to CMM พร้อมกับตนด้วย แต่ถ้าหากเคยผ่านการฝึกอบรมมาแล้วก็ให้ส่งเฉพาะพนักงานที่คัดเลือกไว้จำนวนผู้ที่ควรส่งเข้าฝึกอบรมควรจะประมาณ 5 คน และควรจะเป็นผู้ที่เกี่ยวข้องกับกิจกรรมต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ พนักงานเหล่านี้จะเป็นผู้ร่วมในโครงการนำ CMM มาใช้ต่อไป

การที่จำเป็นต้องเข้ารับการฝึกอบรมก่อนนั้นเป็นเงื่อนไขสำคัญที่ทางสถาบัน SEI ได้กำหนดขึ้น ทั้งนี้เพื่อให้แน่ใจว่าผู้ที่ให้นำ CMM ไปใช้ต้องรู้ปรัชญาและแนวทางของ CMM จริง

ผู้ที่จะเป็นวิทยากรสอนหลักสูตร CMM ได้นั้นจะต้องได้รับอนุญาตจากสถาบัน SEI และถ้าหากต้องการเปิดสอนก็ต้องได้รับอนุญาตจากเขตอุตสาหกรรมซอฟต์แวร์ในสังกัดของสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช) อีกด้วยเพราะ สวทช ได้รับอนุญาตจากสถาบัน SEI ให้เป็นตัวแทนส่งเสริมการนำแบบจำลอง CMM มาใช้ในประเทศไทย ปัจจุบันนี้มีผู้ได้รับอนุญาตให้ดำเนินการฝึกอบรมเรื่อง CMM จาก SEI รวม 5 คนด้วยกัน

เมื่อหัวหน้าโครงการและผู้ร่วมทีมผ่านการฝึกอบรมด้าน CMM เรียบร้อยแล้ว ก็พร้อมที่จะดำเนินงานตามขั้นตอนต่อไป เริ่มต้นด้วยการเตรียมการ

### การเตรียมการ

การนำแบบจำลอง CMM มาใช้นั้นจำเป็นจะต้องมีคู่มือที่สำคัญ คือ หนังสือชื่อ The Capability Maturity Model: Guidelines for Improving the Software Process ซึ่งมีรายละเอียดของกิจกรรมสำคัญต่าง ๆ ที่จะต้องดำเนินการในแต่ละกลุ่มกิจกรรมหรือ KPA เอาไว้ครบถ้วน หากไม่มีหนังสือเล่มนี้ การนำ CMM มาใช้ก็ไม่ประสบความสำเร็จ

การพัฒนาซอฟต์แวร์อย่างมีวุฒิภาวะตามหลัก CMM นั้น ถ้าพิจารณาตามเนื้อหาของสาระของ CMM แล้ว ก็ประกอบด้วยการสร้างให้เกิดปัจจัยสำคัญต่อไปนี้

- นโยบายด้าน CMM ของผู้บริหาร
- วิธีปฏิบัติงานตามหลักการ CMM
- มาตรฐานต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์
- แบบฟอร์มต่าง ๆ สำหรับใช้ในการปฏิบัติงาน
- การฝึกอบรมเพื่อให้ผู้ปฏิบัติงานรู้แนวทางปฏิบัติที่ถูกต้อง
- การเลือกโครงการทดลองเพื่อศึกษาผลการใช้ CMM

งานอีกอย่างหนึ่งที่จำเป็นมากสำหรับโครงการนำ CMM มาใช้ ก็คือการวางแผนให้เห็นชัดว่าจะต้องมีกิจกรรมอะไรบ้าง และจะต้องทำอะไรเมื่อใด อย่างไรก็ตามในที่นี้จะไม่อธิบายการวางแผนสำหรับโครงการนี้ เพราะหัวหน้าโครงการจะเริ่มเห็นภาพชัดเจนขึ้นจนนำไปวางแผนได้เองเมื่อรู้ว่าจะต้องทำอะไรบ้างแล้ว

ก่อนอื่น ทีมงาน CMM ต้องตกลงกับผู้บริหารก่อนว่า ต้องการสร้างและประเมินวุฒิภาวะระดับใด หน่วยงานหรือบริษัทที่เริ่มต้นนำแบบจำลอง CMM มาใช้ไม่ควรรีบด่วนกำหนดระดับวุฒิภาวะที่ต้องการสูงถึงระดับ 4 หรือ 5 ในครั้งแรก อาจจะทำแค่ระดับ 2 หรือ 3 ก็ได้ ทั้งนี้ขึ้นอยู่กับว่าหน่วยงานนั้นเคยมีประสบการณ์กับการทำงานที่เป็นระบบและเคยใช้หลักการจัดการโครงการในการพัฒนาซอฟต์แวร์มานานมากน้อยแค่ไหน หากยังไม่เคยมาก่อนก็ควรเริ่มที่ระดับ 2 หากเคยมาแล้วก็อาจตั้งเป้าไว้ที่ระดับ 3 เลย

### การกำหนดนโยบาย

CMM จะเกิดขึ้นในหน่วยงานได้ก็ต่อเมื่อทุกคนปฏิบัติตาม และยอมรับว่าการทำงานตามหลักการ CMM เป็นวิถีชีวิตของตน ในเรื่องนี้ CMM ใช้คำว่า Institutionalization สำหรับระบุ

กิจกรรมที่จะสร้างวัฒนธรรมองค์กรดังกล่าว วัฒนธรรมองค์กรนี้เกิดได้ด้วยการกำหนดเป็นนโยบายบ้าง สร้างมาตรฐานกำกับบ้าง หรือ กำหนดขั้นตอนการดำเนินงานให้ทุกคนเดินตามบ้าง การประเมินว่าหน่วยงานซอฟต์แวร์มีวุฒิภาวะความสามารถ CMM ระดับใดนั้นปกติเริ่มต้นด้วยการพิจารณาว่าหน่วยงานมีนโยบายกำกับในด้านต่าง ๆ ครบถ้วนหรือไม่ หากไม่มี หรือมีไม่ครบนั้นก็หมายความว่าหน่วยงานมีวุฒิภาวะเพียงแค่อันดับแรกเท่านั้น

ด้วยเหตุนี้จึงจำเป็นที่จะต้องกำหนดนโยบายให้ครบถ้วนในทุกกลุ่มกิจกรรม KPA ที่ต้องการประเมิน แนวทางก็คือให้ทีมงานโครงการ CMM ศึกษาแต่ละ KPA ว่าต้องกำหนดนโยบายอะไรบ้าง ใครควรเป็นผู้กำหนด เนื้อหานโยบายคืออะไร จากนั้นก็ให้ร่างนโยบายขึ้นให้สอดคล้องกับที่แนะนำไว้ในหนังสือ

### กำหนดตัวบุคคลผู้ทำหน้าที่ประกันคุณภาพซอฟต์แวร์

งานด้านการประกันคุณภาพซอฟต์แวร์ (Software Quality Assurance หรือ SQA) เป็นงานพื้นฐานสำคัญของ CMM และ SQA นี้ต้องดูแลกิจกรรมต่าง ๆ ในทุก KPA งาน SQA นี้หากกล่าวอย่างโดยย่อก็คือการเป็นหูเป็นตาให้ผู้บริหารระดับสูง รวมทั้ง หัวหน้าโครงการพัฒนาซอฟต์แวร์ด้วย นั่นคือช่วยในการตรวจสอบว่าทีมงานโครงการได้ดำเนินงานต่าง ๆ ตามกิจกรรมที่กำหนดไว้หรือไม่ จากนั้นก็รายงานให้ผู้บริหารและหัวหน้าโครงการทราบเป็นระยะ ๆ หรือเมื่อเกิดเหตุการณ์ผิดปกติหรือที่ควรสนใจเป็นพิเศษขึ้น

งาน SQA นั้นเริ่มต้นที่ CMM ระดับสอง ดังนั้นจึงควรกำหนดตัวบุคคลผู้ที่จะทำหน้าที่ประกันคุณภาพ หรือ SQA ก่อน วิธีการคือ ให้เลือกนักวิเคราะห์ระบบอาวุโสที่มีความรู้และประสบการณ์ในการสร้างซอฟต์แวร์มาสักคนหนึ่งในระยะแรก บุคคลผู้นี้ควรจะต้องได้รับการฝึกอบรมด้าน CMM มาด้วย เมื่อเลือกได้แล้วให้สอบถามว่าเรามีความรู้หรือเคยผ่านการฝึกอบรมวิธีการประกันคุณภาพซอฟต์แวร์มาบ้างหรือไม่ หากยังไม่มีก็จำเป็นต้องส่งไปฝึกอบรมในหลักสูตรประกันคุณภาพซอฟต์แวร์เพื่อให้สามารถทำงานนี้ได้

เมื่อได้ตัวคนมาแล้ว ก็มอบหมายให้เขาศึกษากระบวนการประกันคุณภาพที่ทางสถาบัน SEI กำหนดเป็นกลุ่มกิจกรรม KPA อย่างละเอียด และให้คิดสร้างแนวทางปฏิบัติงานในโครงการพัฒนาซอฟต์แวร์ว่าจะดำเนินการอย่างไรจึงจะตรวจสอบการทำงานของทีมงานพัฒนาซอฟต์แวร์ว่าเป็นไปตามข้อกำหนดกิจกรรมของ CMM จริง

โดยที่ผู้ทำหน้าที่ SQA ควรจะรายงานขึ้นตรงต่อผู้บริหารระดับสูงซึ่งในที่นี่ได้สมมุติว่าเป็นหัวหน้าศูนย์คอมพิวเตอร์ ผู้ทำงาน SQA จึงไม่ควรมีสายงานขึ้นตรงต่อหัวหน้าโครงการ เพราะจะทำให้ไม่สามารถรายงานการดำเนินงานได้โดยอิสระ

สำหรับจำนวนผู้ทำหน้าที่ เป็น SQA นั้น บริษัทหรือหน่วยงานอาจจะต้องพิจารณาอีกครั้ง หากเห็นว่าจะทดลองนำ CMM มาใช้ในโครงการพัฒนาซอฟต์แวร์มากกว่าหนึ่งโครงการก็อาจจะ ต้องกำหนดให้มี SQA หลายคน ในบางบริษัทมีการมอบหมายงาน SQA ไขว้กันระหว่างหัวหน้า โครงการกับ SQA นั่นคือ หัวหน้าโครงการแรกไปเป็น SQA ในโครงการที่สอง และหัวหน้าโครงการที่สองก็มาเป็น SQA ในโครงการแรก บางครั้งผู้บริหารระดับสูงอาจจะทำหน้าที่เป็น SQA เอง ก็ได้

### กำหนดตัวบุคคลทำหน้าที่บรรณารักษ์

กลุ่มกิจกรรม KPA ในระดับสองที่สำคัญอีกกลุ่มหนึ่งก็คือการจัดการ (Software Configuration Management) งานนี้เกี่ยวกับการจัดเก็บผลงานซอฟต์แวร์และผลงานอื่นที่เกี่ยวข้องแต่ละชุดหรือแต่ละรุ่นที่ได้จัดทำขึ้นเอาไว้เป็นหมวดหมู่เพื่อให้สามารถค้นคืนมาใช้งาน หรือนำมาพิจารณาแก้ไขเปลี่ยนแปลงได้สะดวก

ขอเรียกผู้ทำหน้าที่นี้ว่าบรรณารักษ์ (Librarian) และ เรียกงานนี้ว่า SCM

หัวหน้าโครงการ CMM จะต้องพิจารณาหาตัวผู้ที่เหมาะสมจะทำหน้าที่บรรณารักษ์ตั้งแต่แรก คนผู้นี้ไม่จำเป็นจะต้องเป็นนักวิเคราะห์ระบบหรือนักโปรแกรม และไม่จำเป็นจะต้อง ผ่านการฝึกอบรม CMM มาก่อน ขอแต่ให้เป็นคนที่สนใจในรายละเอียด เข้าใจกระบวนการจัด เอกสาร การกำหนดหมายเลขเอกสารอย่างมีหลักเกณฑ์และเป็นระบบ และที่สำคัญคือใช้ ซอฟต์แวร์สำหรับการทำ Configuration Management ของระบบซอฟต์แวร์ได้

ไม่ว่าบรรณารักษ์จะมีความรู้ด้านเทคนิคของการเขียนโปรแกรม หรือ การพัฒนา ซอฟต์แวร์หรือไม่ก็ตาม หัวหน้าโครงการ CMM ก็จำเป็นจะต้องตรวจสอบความรู้พื้นฐานด้าน SCM ของบุคคลที่คัดเลือกไว้และส่งไปเข้ารับการศึกษาความรู้ด้าน CMM อย่างเป็นทางการ เพราะการมีความรู้ถือเป็นเงื่อนไขสำคัญในการทำงานแบบ CMM หากหัวหน้าโครงการ CMM มอบหมายให้ทำหน้าที่นี้แต่ไม่เคยเรียนรู้เรื่องนี้มาก่อน ในทางปฏิบัติแล้วจะต้องถือว่าไม่มีความรู้ที่จะทำงานนี้ และจะถือว่าหน่วยงานไม่มีวุฒิภาวะ

ในการทำงานขั้นแรกทางด้าน SCM นั้นหัวหน้าโครงการ CMM จะต้องช่วยกำหนดว่าจะ ต้องเก็บเอกสาร หรือ สิ่งอื่นใดไว้เป็นระบบ configuration บ้าง เราเรียกสิ่งที่กำหนดนี้ว่าเป็น Configuration Item หรือ CI นอกจากนั้น CMM ยังระบุให้แยกกันระหว่าง Software Products กับ Software Work Products ด้วย

Software Products หมายถึงตัวซอฟต์แวร์ และ สิ่งที่จะส่งมอบให้แก่ลูกค้า เช่น คู่มือต่าง ๆ และอาจจะรวมถึง ชุดข้อมูลทดสอบด้วย

Software Work Products หมายถึง Software Products และ เอกสารตลอดจนสิ่งอื่น ๆ ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ เช่น ข้อกำหนดความต้องการของลูกค้า รายงานการประชุม เอกสารต่าง ๆ ของลูกค้าที่นำมาศึกษา ฯลฯ

เมื่อแต่งตั้งผู้เป็นบรรณาธิการ ตลอดจนจัดให้ไปรับการฝึกอบรมเสร็จเรียบร้อยแล้ว หัวหน้าโครงการ CMM ก็ต้องมอบหมายให้บรรณาธิการเริ่มวางระบบเอกสาร และ ระบบ configuration ของซอฟต์แวร์เอาไว้เพื่อให้พร้อมที่จะนำไปใช้งานในโครงการจริง

### กำหนดกิจกรรมต่าง ๆ

เมื่อได้ตัวบุคคลที่จำเป็นสำหรับงาน CMM แล้ว หัวหน้าโครงการก็ต้องเริ่มกำหนดวิธีการทำงานกิจกรรมต่าง ๆ (Key practice) ในส่วนที่เกี่ยวกับโครงการพัฒนาซอฟต์แวร์ขึ้น วิธีการก็คือให้พิจารณากิจกรรมต่าง ๆ แล้วกำหนดวิธีการดำเนินงานตามกิจกรรมเหล่านั้นให้ละเอียด ที่ต้องกำหนดเองก็เพราะว่าหนังสือคู่มือที่อ้างถึงนั้นระบุแต่เพียงว่าต้องทำกิจกรรมอะไร แต่ไม่บอกว่าจะต้องทำอย่างไร สถาบัน SEI เปิดโอกาสให้หน่วยงานและบริษัทพิจารณาเองว่าจะดำเนินการตามกิจกรรมนั้นอย่างไร

ยกตัวอย่างเช่น ในกิจกรรม AC1 ของ KPA หัวข้อ การจัดการความต้องการ (Requirement Management) ระบุว่า

AC1 กลุ่มวิศวกรรวมซอฟต์แวร์ทบทวนข้อกำหนดความต้องการที่ได้รับมอบหมายก่อนที่จะนำมารวบรวมไว้ในโครงการซอฟต์แวร์

นั่นหมายความว่าทีมงานโครงการ CMM จะต้องกำหนดกิจกรรมนี้เอาไว้อย่างชัดเจนว่าหลังจากได้รับข้อกำหนดความต้องการแล้วกลุ่มวิศวกรรวมซอฟต์แวร์จะต้องพิจารณารายละเอียดที่ได้รับว่าข้อกำหนดเหล่านั้นชัดเจนหรือไม่ มีรายละเอียดครบถ้วนหรือไม่ จะทำได้หรือไม่ มีส่วนใดบ้างที่จะเป็นปัญหาหรือไม่ หากมีก็ต้องเจรจาต่อรองกับผู้ระบุข้อกำหนดความต้องการเหล่านั้นเสียก่อนที่จะนำมาระบุเป็นข้อกำหนดความต้องการของโครงการ

ทีมงานโครงการ CMM จะต้องตรวจสอบกิจกรรมเหล่านี้ทั้งหมด นำมาเขียนเป็นรายละเอียดที่ต้องทำให้ครบถ้วน ต่อจากนั้นจึงนำกิจกรรมเหล่านี้มาเรียงเรียงเสียใหม่ตามลำดับเวลาที่สมควรจะทำกิจกรรมนั้นในโครงการ

การเขียนรายละเอียดกิจกรรมเหล่านี้เป็นงานที่ค่อนข้างจะน่าเบื่อและต้องใช้เวลามาก แต่ถ้าหากไม่ทำหัวหน้าโครงการ CMM ก็จะไม่ทราบว่าจะต้องทำอะไรบ้างและอาจจะลืมกิจกรรมบางอย่างไปโดยไม่ทันรู้ตัว

## จัดเตรียมเครื่องมือและแบบฟอร์มเอกสาร

การนำ CMM มาใช้นั้นจำเป็นต้องอาศัยเครื่องมือหลายอย่าง โดยเฉพาะคือ ซอฟต์แวร์ สำหรับการจัดการโครงการ ซอฟต์แวร์สำหรับงาน Software Configuration Management ซอฟต์แวร์สำหรับการทำฐานข้อมูลเอกสาร ฯลฯ เครื่องมือเหล่านี้หัวหน้าโครงการ CMM จะต้องพิจารณาเลือกใช้ตามความจำเป็นและเหมาะสม หน่วยงานที่จะใช้ CMM ไม่จำเป็นจะต้องใช้ เครื่องมือที่มีราคาแพงมาก ถ้าหากรู้จักดัดแปลงเครื่องมือให้ทำงานตามกิจกรรมต่าง ๆ ได้ครบถ้วน ก็น่าพอใจแล้ว

บริษัทซอฟต์แวร์บางแห่งที่นำแนวคิด CMM มาใช้ได้สร้างระบบสารสนเทศ CMM ขึ้นใช้ ภายใน ระบบสารสนเทศนี้ประกอบด้วยฟังก์ชันสำคัญดังต่อไปนี้

- ข้อมูลพื้นฐานเกี่ยวกับโครงการทุกโครงการในหน่วยงาน วันเริ่มต้นโครงการ วันที่ คาดหมายว่าจะเสร็จสิ้น ชื่อลูกค้า ชื่อผู้ประสานงาน ฯลฯ
- ข้อมูลเกี่ยวกับพนักงานที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ เช่น พื้นฐานความรู้ การ ฝึกอบรมที่เคยได้รับ โครงการที่เคยร่วมงาน โครงการที่กำลังเกี่ยวข้องอยู่ เวลาที่จะ เสร็จสิ้นหน้าที่ในโครงการปัจจุบัน ฯลฯ
- ข้อมูลเกี่ยวกับการดำเนินงานโครงการแต่ละโครงการอย่างละเอียด วัตถุประสงค์ของ โครงการ แผนงานโครงการ ข้อกำหนดความต้องการ กิจกรรมที่ทำเสร็จสิ้นแล้ว ผล การดำเนินงานโครงการ ฯลฯ งานในส่วนนี้ก็เหมือนกับระบบจัดการโครงการโดยทั่ว ไปนั่นเอง
- ข้อมูลสรุปของโครงการที่ทำสำเร็จแล้ว รวมทั้งตัววัดต่าง ๆ เช่น ขนาดของซอฟต์แวร์ ระยะเวลาที่ใช้ในโครงการ กำลังงานที่ใช้ จำนวนข้อกำหนดความต้องการที่ระบุใน ระยะเวลาแรก และที่มีการเปลี่ยนแปลงไป จำนวนข้อบกพร่องที่ตรวจพบในระหว่างการพัฒนาซอฟต์แวร์ ฯลฯ
- ฐานข้อมูลเกี่ยวกับลูกค้า ผู้ประสานงาน รายละเอียดต่าง ๆ เกี่ยวกับลูกค้าที่ควรทราบ
- ฐานข้อมูลสำหรับจัดเก็บ นโยบาย ระเบียบปฏิบัติ และ มาตรฐานต่าง ๆ ของหน่วย งาน
- ฐานข้อมูลสำหรับจัดเก็บแบบฟอร์มเอกสารต่าง ๆ ที่จะใช้ในโครงการ
- ระบบจัดการข้อมูลเอกสารต่าง ๆ ที่เก็บไว้ในฐานข้อมูลข้างต้น

ข้อที่นำคิดอย่างหนึ่งก็คือการพัฒนาซอฟต์แวร์ตามระบบวุฒิภาวะ CMM นั้นมีผู้ตำหนิว่า ต้องจัดเตรียมเอกสารค่อนข้างมาก เรื่องนี้เป็นจริงส่วนเดียว เพราะการจัดการโครงการพัฒนา ซอฟต์แวร์ที่ถูกต่อนั้นจำเป็นต้องใช้เอกสารมากอยู่แล้ว วิธีการ CMM เพียงแต่กำหนดให้เห็นชัด ว่ามีเอกสารอะไรบ้างที่จำเป็นในส่วนที่เกี่ยวกับงานโครงการ และในส่วนที่เกี่ยวกับตัวหน่วยงานเอง

เอกสารที่เกี่ยวข้องกับตัวหน่วยงานเองนั้นไม่ต้องจัดทำบ่อยครั้ง เพียงแต่ทำครั้งแรกให้เสร็จสมบูรณ์เท่านั้นก็พอแล้ว นั่นก็คือนโยบาย ระเบียบ และมาตรฐานต่าง ๆ ที่เคยกล่าวมาแล้วนั่นเอง

ยอร์ช วินเตอร์ส วิทยากรจาก SEI ที่มาสอน CMM ให้กับเขตอุตสาหกรรมซอฟต์แวร์ไทย นั้นเคยกล่าวว่า ทางสถาบัน SEI กำลังพิจารณาอยู่ว่าบรรดาเอกสารทั้งหลายที่เคยกำหนดมาให้เขียน และให้ผู้เกี่ยวข้องลงลายมือชื่อ นั้นในอนาคตคงจะปรับเปลี่ยนรูปแบบไปเป็นเอกสารอิเล็กทรอนิกส์อย่างแน่นอน และผู้ใช้ CMM แต่ละรายคงจะต้องพิจารณาเองว่าจะสร้างเอกสารอิเล็กทรอนิกส์อย่างไร

### การเตรียมการตรวจสอบทบทวน

กิจกรรมจำนวนมากที่จะต้องทำในกระบวนการ CMM ก็คือการตรวจสอบทบทวนผลการดำเนินงานโครงการพัฒนาซอฟต์แวร์ว่าเป็นไปโดยเรียบร้อยหรือไม่ หากมีปัญหาผู้บริหารก็จะได้รับทราบโดยเร็วพลัน

หัวหน้าโครงการ CMM ต้องพิจารณาว่าต้องการจะให้ทีมงานพัฒนาซอฟต์แวร์ต้องตรวจสอบผลการดำเนินงานอย่างไร ตรวจสอบเมื่อใด และจะต้องส่งรายงานให้ใครบ้าง การพิจารณาเรื่องนี้ทำได้ไม่ยาก เพียงแต่ดูจากกิจกรรมในส่วนที่เกี่ยวข้องกับการวัดผลที่ระบุในหนังสือ และกิจกรรมส่วนที่เป็นการตรวจสอบการดำเนินการในแบบจำลอง CMM ก็จะเห็นรายละเอียดที่จะต้องจัดเตรียมการไว้ครบถ้วน

เมื่อเห็นภาพของงานที่เป็นความรับผิดชอบของผู้บริหารหน่วยงานแล้ว หัวหน้าโครงการ CMM ก็จะต้องแจ้งให้ผู้บริหารของหน่วยงานหรือบริษัททราบว่า มีหน้าที่และความรับผิดชอบที่จะต้องตรวจสอบทบทวนอะไรบ้าง หากงานส่วนนี้ผู้บริหารไม่เข้าใจ การปฏิบัติตาม CMM ก็จะไม่บรรลุเป้าหมายที่วางไว้

### บทส่งท้าย

การผลักดันให้หน่วยงานมีวุฒิภาวะความสามารถในการพัฒนาซอฟต์แวร์ ตามแนวแบบจำลอง CMM นั้นความจริงไม่ใช่เรื่องยาก หน่วยงานที่สนใจอาจศึกษาแนวทางทั้งหมดได้จากหนังสือคู่มืออยู่แล้ว อย่างไรก็ตามคู่มือที่ทางผู้เชี่ยวชาญของ SEI ได้สร้างขึ้นนั้นได้ลงรายละเอียดกิจกรรมที่ทำไว้ตามลำดับ KPA ที่นำมาอธิบายประกอบวุฒิภาวะแต่ละระดับ ทำให้ยากแก่การที่หน่วยงานจะนำมาใช้ได้โดยไม่ต้องคิดพิจารณา

บริษัทซอฟต์แวร์หลายแห่งที่สนใจนำ CMM มาใช้ล้วนต้องการคำตอบสำเร็จที่นำไปใช้ได้เลย นั่นคืออยากทราบว่าบริษัทอื่นเขาเขียนนโยบายกันอย่างไร มีแบบฟอร์มอะไรใช้บ้าง บริษัทที่ประเมินผ่านแล้วจะยินดีให้เอกสารมาสนใจศึกษาหรือไม่ ความสนใจเช่นนี้เป็นเรื่องธรรมดาเพราะ

ทุกคนย่อมอยากได้คำตอบสำเร็จที่ได้มาง่าย ๆ อย่างไรก็ตามบริษัทซอฟต์แวร์ทั้งหลายนั้นมิทั้งที่  
ต้องร่วมมือกันและต้องแข่งขันกัน การพัฒนาตัวเองตามแนวทาง CMM นั้นต้องลงทุนทั้งด้าน  
เวลา กำลังคน และกำลังทรัพย์ ดังนั้นจึงไม่มีบริษัทใดยินดีที่จะเอางานที่ตนเองค้นคว้าศึกษามา  
มอบให้ผู้อื่นนำไปใช้โดยไม่ได้อะไรตอบแทน

ข้อที่ควรตระหนักก็คือหน่วยงานและบริษัทซอฟต์แวร์ทั้งหลายล้วนแตกต่างกัน การ  
ทำงาน การบริหาร การกำหนดตลาดและลูกค้า ส่วนไม่เหมือนกัน ดังนั้นนโยบายจึงไม่ควร  
เหมือนกัน เหตุนี้เองหน่วยงานและบริษัทซอฟต์แวร์จึงควรกำหนดนโยบาย ระเบียบ และ มาตรฐาน  
ต่าง ๆ เป็นของตัวเอง ไม่ควรพยายามใช้แนวทางและแบบฟอร์มของผู้อื่น

การจัดทำระบบสารสนเทศสำหรับใช้ในการบริหารงานพัฒนาซอฟต์แวร์ตามแนวทาง  
CMM ก็เป็นวิธีการที่น่าสนใจ ในที่นี้ได้กล่าวถึงเรื่องนี้แต่พอสังเขปไปในตอนต้นแล้ว ผู้สนใจจะ  
สามารถนำไปพิจารณาออกแบบระบบของตนเองได้โดยไม่ยากนัก

#### หนังสืออ้างอิง

Paulk, M.C., C.V. Weber, B. Curtis, and M. Beth. Chrissis, *The Capability  
Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley, Reading  
Massachusetts, 1994.