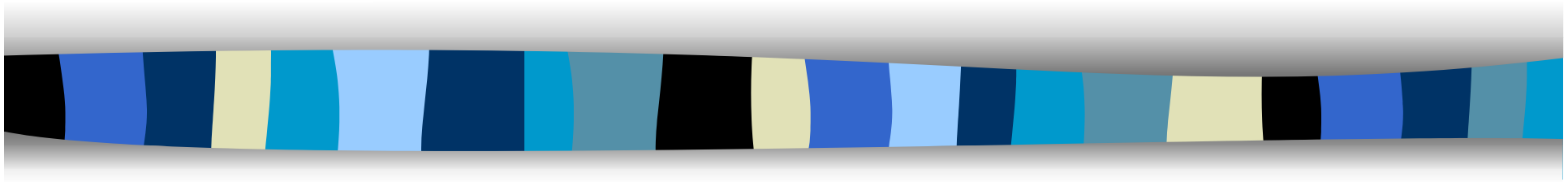
การบริหารโครงการซอฟต์แวร์และCMM



ดร. ครรชิต มาลัยวงศ์

ศูนย์บริการสารสนเทศทางเทคโนโลยี

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

11 กุมภาพันธ์ 2543

เนื้อหาคำบรรยาย

- สถานภาพงานพัฒนาซอฟต์แวร์ไทย
- การบริหารงานโครงการซอฟต์แวร์
- ภาพรวมของ CMM
- การรับรองระดับ CMM
- ไทยกับ CMM
- สรุป

โลกของซอฟต์แวร์

- โลกยุคปัจจุบันถูกควบคุมด้วยซอฟต์แวร์นานาประเภท
- ซอฟต์แวร์ระบบ
- ภาษาคอมพิวเตอร์
- ซอฟต์แวร์ประยุกต์
- ซอฟต์แวร์ระบบฝังตัว
- ซอฟต์แวร์เหล่านี้คือจุดเริ่มต้นของความเสี่ยง

วิวัฒนาการของซอฟต์แวร์

การพัฒนาซอฟต์แวร์ผ่านกระบวนการเรียนรู้มามาก

- ใครใคร่เขียนเขียน
- ใครใคร่คิดคิด
- สับสนและวิกฤติ
- แสวงหาโครงสร้าง
- พัฒนาแนวทางและกระบวนการ

จุดวิกฤติในงานซอฟต์แวร์

การพัฒนาซอฟต์แวร์มีประเด็นที่เป็นจุดวิกฤติหลายด้าน อาทิ

- คุณภาพและความน่าเชื่อถือ
- สมรรถนะและความยืดหยุ่น
- ความสร้างสรรค์และการเปิดเผย
- โครงสร้างและการปรับเปลี่ยน
- ผลลัพธ์และกระบวนการ

กระบวนการซอฟต์แวร์

กระบวนการ หมายถึง ลำดับของขั้นตอนต่าง ๆ ในการดำเนินงานให้บรรลุเป้าหมาย

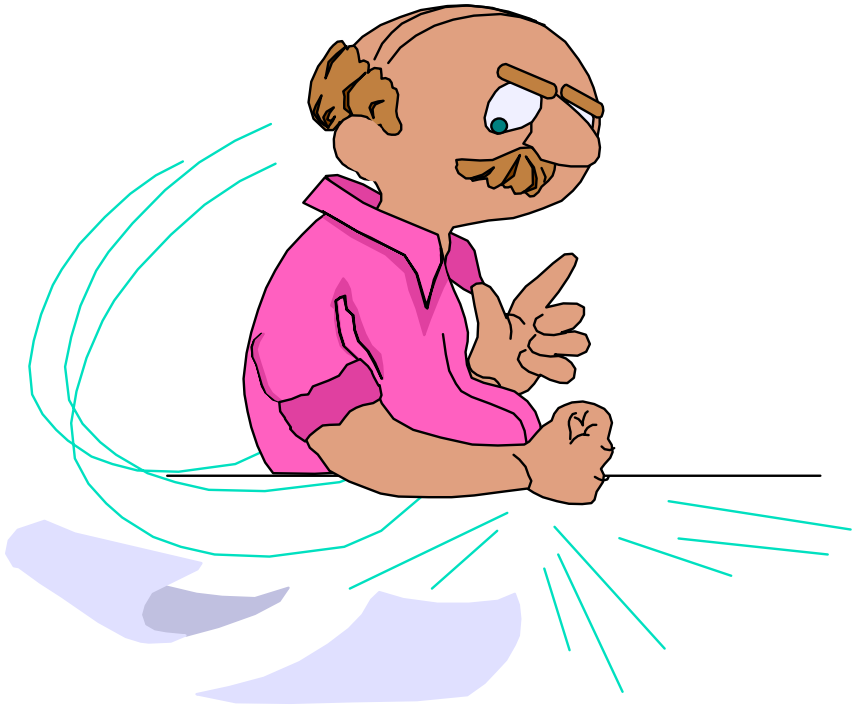
กระบวนการซอฟต์แวร์ หมายถึงกลุ่มของกิจกรรม วิธีการ วิธีการปฏิบัต และการเปลี่ยนแปลงที่ใช้ในการพัฒนาและบำรุงรักษา ซอฟต์แวร์ ตลอดจนผลิตภัณฑ์ที่เกี่ยวเนื่อง

กระบวนการซอฟต์แวร์ประกอบด้วย คน วิธีการ และเครื่องมือ

คุณภาพซอฟต์แวร์ไทย

- อยู่ในตัวซอฟต์แวร์เอง
  - Interface
  - Processing
  - Results
  - Robustness
- อยู่ในกระบวนการสร้างซอฟต์แวร์
- และอยู่ในกระบวนการบำรุงรักษาซอฟต์แวร์

การบริหารงานโครงการซอฟต์แวร์



การบริหารงานโครงการ

- โครงการ หมายถึง งานที่มีวัตถุประสงค์แน่ชัด มีจุดเริ่มต้น และ จุดสิ้นสุดที่ชัดเจน มีทรัพยากรที่ชัดเจน
- การบริหารงานโครงการมักจะมอบหมายให้มีผู้จัดการโครงการ ทำหน้าที่ควบคุมทรัพยากร และ จัดสรรงานต่าง ๆ ให้แก่ผู้ร่วมทีม พร้อมกับดูแลให้งานดำเนินไปตามกำหนดการที่วางไว้

ซอฟต์แวร์ก็เป็นงานโครงการ

- การพัฒนาซอฟต์แวร์จัดเป็นงานโครงการได้
- นั่นคือมี เนื้องานและวัตถุประสงค์ชัดเจน มี ทรัพยากร งบประมาณ และกำหนดเวลาชัดเจน
- การบำรุงรักษา และ แก้ไขซอฟต์แวร์ไม่จัดว่าเป็นงานโครงการ เพราะไม่มีกำหนดเวลาสิ้นสุดที่ชัดเจน

หลักการบริหารโครงการซอฟต์แวร์

- ต้องเข้าใจตัวแปรของโครงการ
  - ปริมาณงานที่ต้องพัฒนา
  - กำหนดเวลาดำเนินการ
  - กำลังคนที่ต้องใช้
  - งบประมาณ
  - วิธีการและเครื่องมือ

หลักการบริหารโครงการซอฟต์แวร์

- ต้องมีทีมงานที่มีความสามารถ ทั้งตัวผู้บริหารโครงการ ผู้ร่วมทีม และผู้ประสานงานฝ่ายผู้ใช้
- ต้องใช้วิธีการพัฒนาที่เป็นมาตรฐาน
- ต้องบันทึกเหตุการณ์ต่าง ๆ ระหว่างการดำเนินการไว้เป็นสถิติเพื่อปรับปรุงในอนาคต
- ต้องมีการควบคุมและประเมินโครงการตลอดเวลา

หลักการบริหารโครงการซอฟต์แวร์

- ต้องตั้งเป้าหมายในการพัฒนาซอฟต์แวร์ที่มีคุณภาพ
  - ตรงกับความต้องการของผู้ใช้
  - มีความยืดหยุ่น ดูแล แก้ไข และ ปรับปรุงได้ง่าย
  - มีประสิทธิภาพในการทำงาน
  - มีความถูกต้องน่าเชื่อถือ
  - ใช้ง่าย

หลักการบริหารโครงการซอฟต์แวร์

- ต้องวางแผน
- กำหนด Work Breakdown ของโครงการ
- มอบหมายให้ลูกทีมทำงานตามภาระงานที่ตนเองมีประสบการณ์
- ควบคุมการทำงานอย่างใกล้ชิด พร้อมช่วยแก้ปัญหาและ หาทางแก้ไขหากมีปัญหา

กระบวนการซอฟต์แวร์

- การพัฒนาซอฟต์แวร์ เป็นกระบวนการที่เรียกว่า Software Process
- ผลของการพัฒนาเป็นผลิตภัณฑ์ Software product
- ทั้ง Process และ Product จะต้องมีคุณภาพ
- อีกนัยหนึ่ง กระบวนการจะต้องมีวุฒิภาวะ

กระบวนการที่ยังไม่บรรลุวุฒิภาวะ

- การทำงานเป็นไปแบบต่างคนต่างคิด
- ไม่ได้กำหนดเป็นแนวทางแน่ชัด
- ขึ้นอยู่กับประสบการณ์ของคนพัฒนาซอฟต์แวร์
- ติดตามความก้าวหน้าและคุณภาพได้ยาก
- อาจต้องยอมลดฟังก์ชันและคุณภาพเพื่อพัฒนาให้ตรงกำหนด
- เสี่ยงที่จะใช้เทคโนโลยีที่ก้าวหน้า
- ค่าบำรุงรักษาสูงมาก
- คาดคะเนคุณภาพได้ยาก

(CMM)

กระบวนการที่เจริญก้าวหน้าสูงสุด

- สอดคล้องกับเนื้องานที่ทำจริง ๆ
- เป็นกระบวนการที่กำหนดขั้นตอนอย่างชัดเจน มีการบันทึกการทำงานเป็นเอกสาร/เพื่อให้สามารถแก้ไขปรับปรุงผลงานได้อย่างต่อเนื่อง
- ทำให้ฝ่ายบริหารและฝ่ายอื่น ๆ สนับสนุนได้ชัดเจน
- สามารถควบคุมการดำเนินงานได้
- ใช้เครื่องมือวัดผลและกระบวนการอย่างสร้างสรรค์
- ใช้เทคโนโลยีอย่างมีวินัย

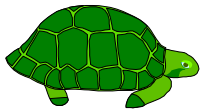
ประโยชน์ของการมีกระบวนการที่บรรลุวุฒิภาวะ

- ช่วยให้สามารถพิจารณาเจาะลงไปยังจุดที่เป็นสาเหตุของปัญหาได้ง่ายขึ้น
- ช่วยให้เจ้าหน้าที่พัฒนาศักยภาพของตนเองได้อย่างมีประสิทธิผล
- ช่วยปรับปรุงผลงานต่าง ๆ ได้อย่างมีประสิทธิภาพและยั่งยืน
- ช่วยให้นำเทคโนโลยีที่เหมาะสม เทคนิค และเครื่องมือมาใช้อย่างมีประสิทธิภาพได้เพิ่มขึ้น

โครงการพัฒนาซอฟต์แวร์กับ CMM

- CMM ไม่ได้เป็นเทคนิคหรือเครื่องมือในการทำโครงการซอฟต์แวร์
- แต่การใช้เทคนิคการควบคุมโครงการพัฒนาซอฟต์แวร์ เป็น ก้าวสำคัญในการยกตัวเองขึ้นมาสู่ระดับที่สองของ CMM ซึ่งเรียกว่า ระดับ Repeatable

CMM เครื่องมือวัดระดับวุฒิภาวะ



CMM คืออะไร

- การนำกระบวนการตัดสินใจและหลักการปรับปรุงคุณภาพมาใช้กับการพัฒนาและการบำรุงรักษาซอฟต์แวร์
- เป็นแนวทางสำหรับให้บริษัทซอฟต์แวร์ใช้
- เป็นแบบจำลองสำหรับปรับปรุงองค์กร
- เป็นโครงสร้างพื้นฐานสำหรับใช้ประเมินการทำงานของบริษัทซอฟต์แวร์ได้อย่างมั่นใจ

CMM คืออะไร

- Capability Maturity Model
- เป็นแบบจำลองสำหรับวัดว่าหน่วยงานที่ทำหน้าที่พัฒนาซอฟต์แวร์นั้น มีความสามารถและได้บรรลุวุฒิภาวะในการทำงานมากน้อยเพียงใด
- ใช้ในการตรวจสอบคุณภาพของตัวเองเป็นหลัก แต่สามารถใช้สร้างความมั่นใจให้แก่ผู้อื่นได้
- พัฒนาและเผยแพร่โดย Software Engg Institute

CMM ไม่ได้ครอบคลุมอะไรบ้าง

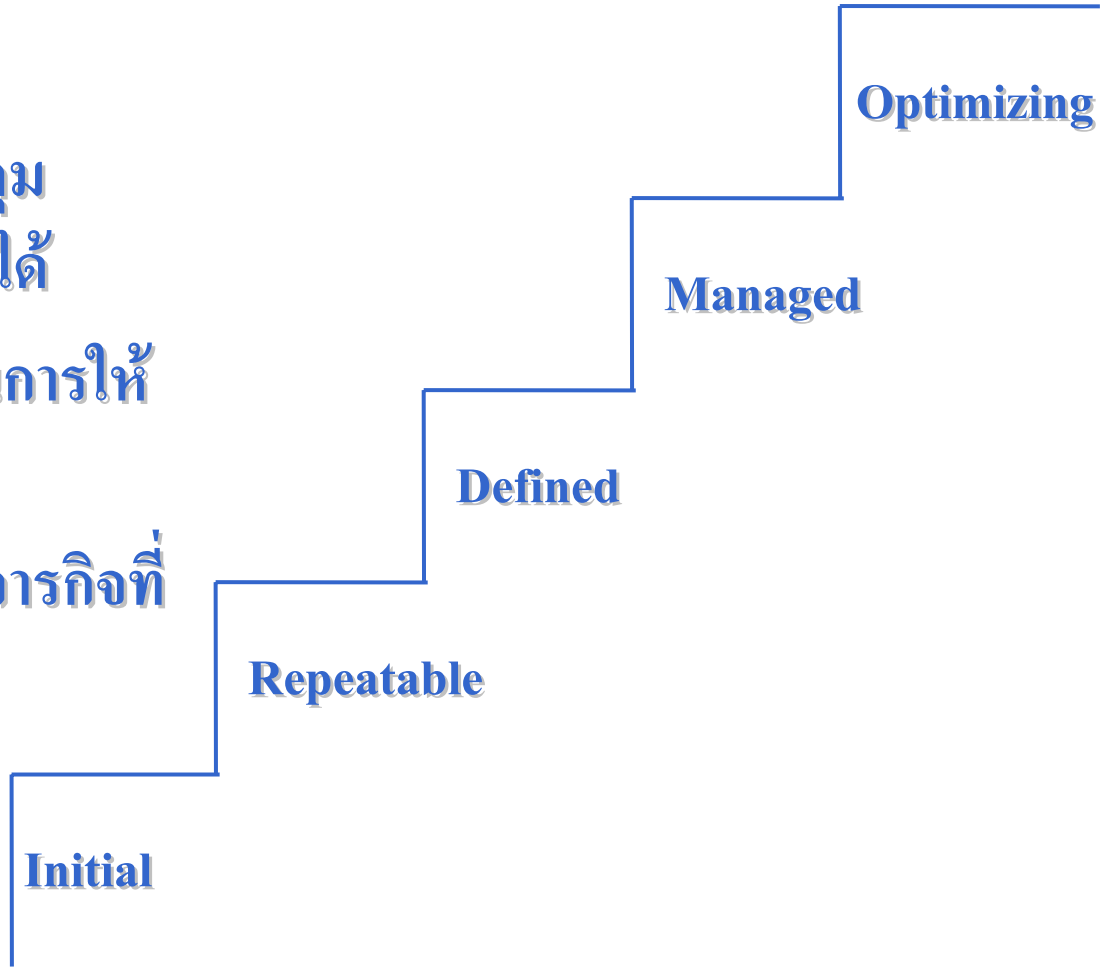
- CMM ไม่ได้ครอบคลุมถึงประเด็นทั้งหมดทางกระบวนการซอฟต์แวร์ และการปรับปรุงคุณภาพ
- ประเด็นที่เกี่ยวข้องเพียงบางส่วน หรือโดยอ้อม คือ
  - เครื่องมือ วิธีการ และเทคโนโลยี
  - ทีมงานและกระบวนการทำงาน
  - วิศวกรรมระบบและการตลาด
  - ทรัพยากรมนุษย์
  - พฤติกรรมองค์กร

Capability และ Performance

- Process Capability พิสัยของผลลัพธ์ที่ต้องการ และทำได้โดยใช้กระบวนการที่กำหนดขึ้นในระดับองค์การเป็นดัชนีสำหรับคาดคะเนผลการดำเนินงานโครงการในอนาคต
- Process performance การวัดผลลัพธ์จริง ๆ ที่ได้จากการดำเนินงานตามกระบวนการที่กำหนด ปกติมักจะหมายถึงโครงการหนึ่ง ๆ ในองค์การ

ระดับของวุฒิภาวะ

5. เน้นในด้านการปรับปรุงกระบวนการ
4. สามารถวัดผลและควบคุมกระบวนการซอฟต์แวร์ได้
3. สามารถจำแนกกระบวนการให้เข้าใจได้ง่าย
2. โครงการสามารถทำซ้ำภารกิจที่มีการควบคุมอย่างดี
1. ไม่สามารถคาดคะเนกระบวนการได้ และการทำงานก็ยัง/ไม่มีการควบคุมที่ดี



วิวัฒนาการของ Process Capability

ระดับ	ลักษณะกระบวนการ
5. Optimizing	ปรับปรุงกระบวนการซอฟต์แวร์ทั้งองค์การ และ ทำอย่างต่อเนื่อง
4. Managed	ควบคุมผลผลิตและกระบวนการด้วยเทคนิคทางสถิติจำนวน
3. Defined	กำหนดกระบวนการและรวมวิศวกรรมซอฟต์แวร์กับกระบวนการจัดการ
2. Repeatable	ใช้ระบบจัดการโครงการ สามารถทำให้เกิดผลสำเร็จแบบเดียวกันได้
1. Initial	กระบวนการไม่มีรูปแบบทางการและคาดเดาไม่ได้

ระดับ Initial

- ผลงานขึ้นอยู่กับความเก่งกล้าสามารถของบุคลากรที่พัฒนาซอฟต์แวร์ของหน่วยงาน
- ผลงานอาจมีคุณภาพสูงและเยี่ยมยอดได้ตราบเท่าที่หน่วยงานยังสามารถจ้างคนเก่งเอาไว้
- ทำนายผลงานไม่ได้ว่าจะมีคุณภาพดีหรือไม่
- ปัญหาสำคัญที่หน่วยงานซอฟต์แวร์ประสบอยู่คือปัญหาด้านการจัดการ ไม่ใช่ปัญหาด้านเทคนิค
- หน่วยงานยังไม่มี Key Process Area

ระดับ Repeatable

- ความจำเป็นที่เห็นชัดคือต้องมีวิธีการจัดการโครงการซอฟต์แวร์ให้ประสบผลสำเร็จ
- หน่วยงานมีวิธีการจัดการโครงการซอฟต์แวร์ ซึ่งใช้เป็นหลักในการติดตามและบันทึกผลการทำงาน
- มีนโยบายองค์กรสำหรับเป็นแนวทางในการกำหนดวิธีการจัดการโครงการ
- สามารถทำงานแต่ละโครงการให้ประสบความสำเร็จได้เหมือนโครงการอื่น ๆ ที่เคยสำเร็จไปแล้ว (repeatable)

Key Process Area ในระดับ Repeatable

- Software Configuration Management
- Software Quality Assurance
- Software Subcontract Management
- Software Project Tracking and Oversight
- Software Project Planning
- Requirement Management

ระดับ Defined

- คุณภาพระดับนี้สร้างบนพื้นฐานของการจัดการโครงการซอฟต์แวร์
- การควบคุมกระบวนการจำเป็นต้องนิยาม บันทึกรายละเอียด และเข้าใจกระบวนการนั้นเป็นอย่างดี
- ผลลัพธ์ของภาระงานอย่างหนึ่งไหลอย่างราบรื่นเป็นอินพุตไปสู่อีกภาระงานหนึ่ง
- หน่วยงานมีกระบวนการที่ให้อำนาจบุคคลในการทำงาน

Key Press Areas ระดับ Defined

- Peer Reviews
- Intergroup Coordination
- Software Project Engineering
- Integrated Software Management
- Training Program
- Organization Process Definition
- Organization Process Focus

ระดับ Managed

- ใช้หลักการการควบคุมกระบวนการเชิงสถิติ ในการศึกษาว่าอะไรเป็นสาเหตุของความแปรปรวนของการทำงานในโครงการ
- **Key Process Areas คือ**
  - **Software Quality Management**
  - **Quantitative Process Management**

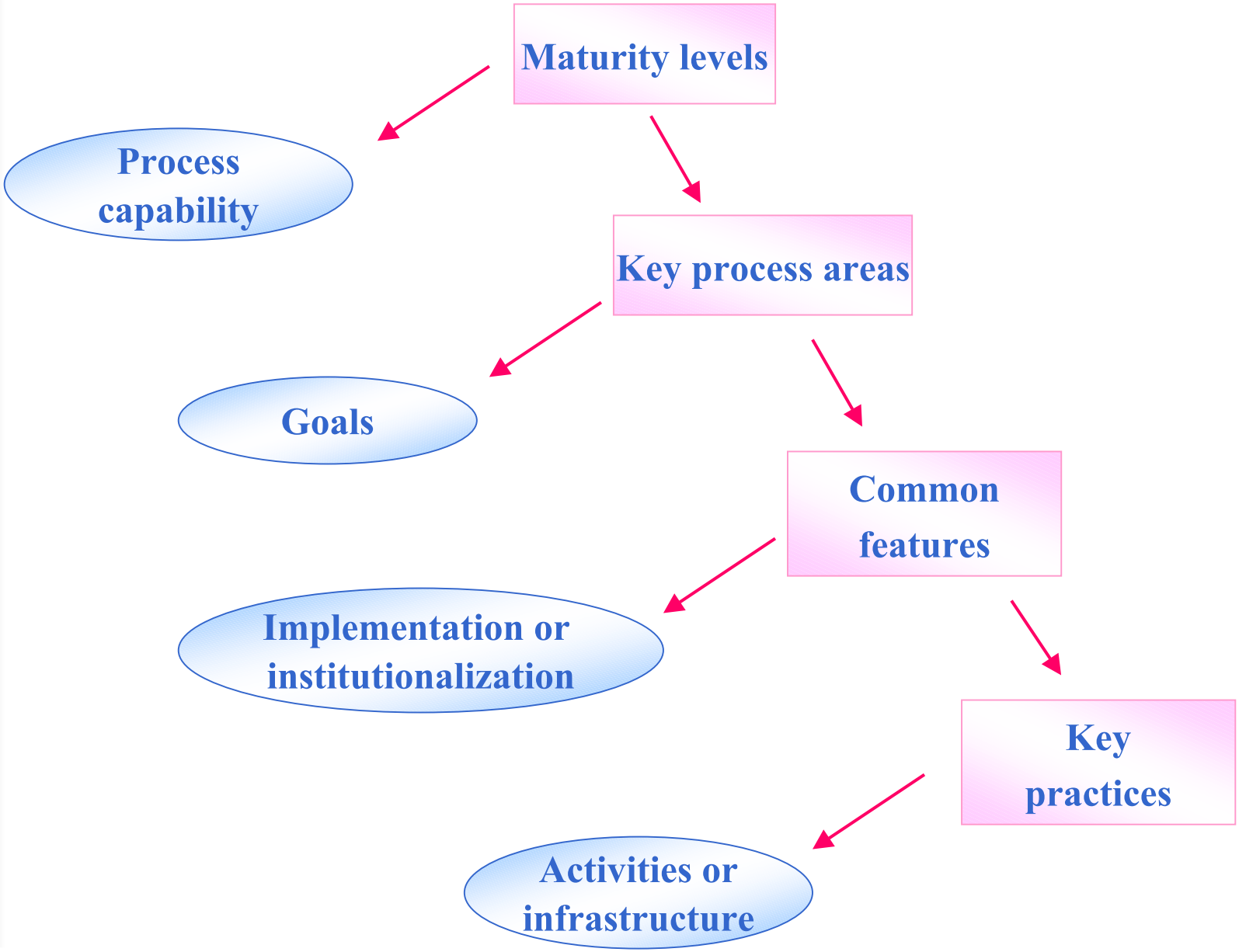
ระดับ Maturity Level

- จำแนกและกำจัดสาเหตุของผลงานที่ไม่ถึงระดับ
- พยายามปรับปรุงกระบวนการซอฟต์แวร์อย่างต่อเนื่อง
- **Key Process Areas :**
  - **Process Change Management**
  - **Technology Change Management**
  - **Defect Prevention**

การบริหารซอฟต์แวร์ต้องผ่านไปทีละระดับ

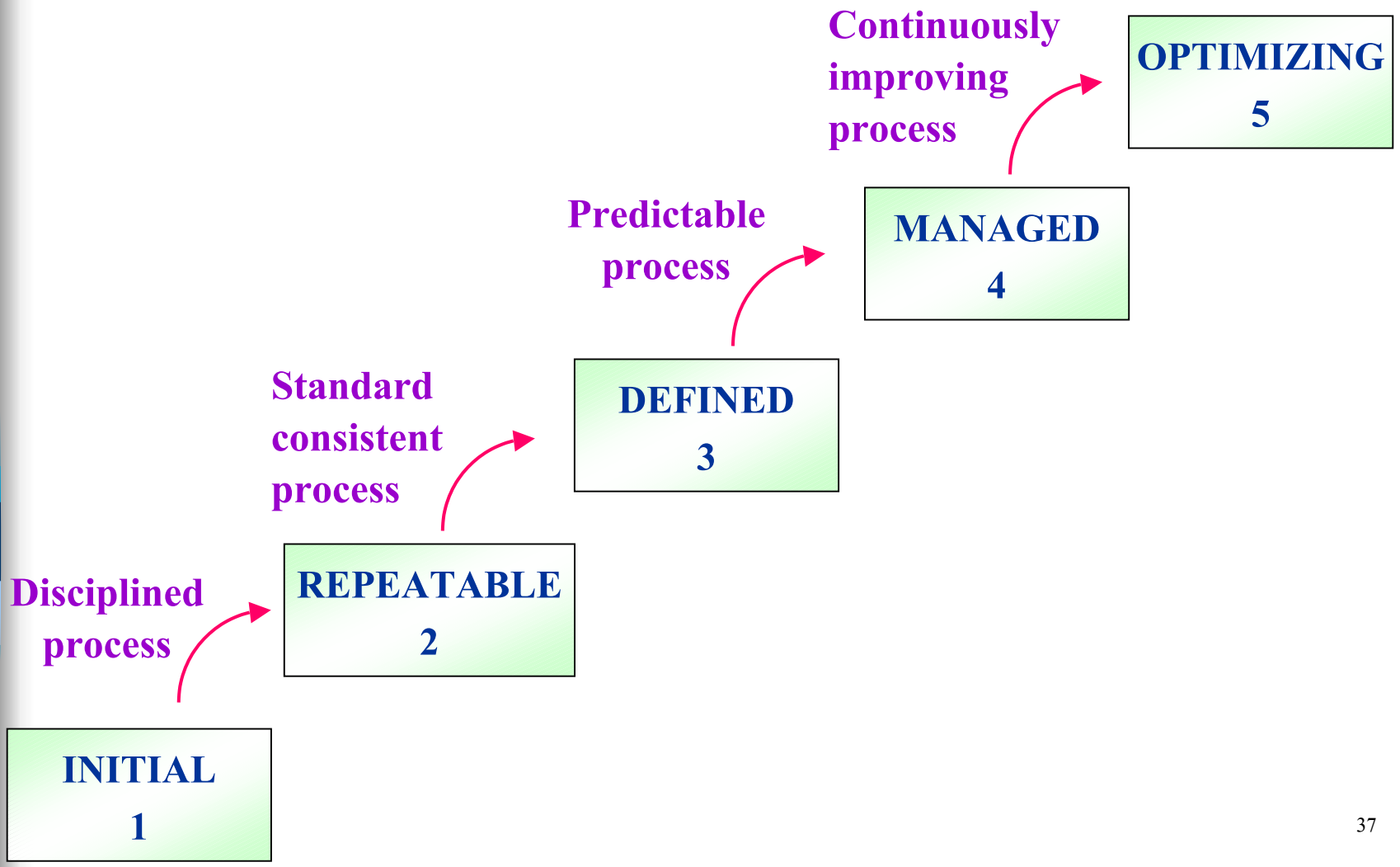
- หน่วยงานซอฟต์แวร์อาจทำกระบวนการที่อยู่ระดับสูงกว่าได้ แต่มักจะไม่ได้ผล
- ความสามารถในการทำงานแต่ละกระบวนการต้องสร้างขึ้นทีละระดับ
- คุณภาพแต่ละระดับเป็นพื้นฐานของระดับที่สูงกว่า
  - กระบวนการวิศวกรรมที่สำคัยอาจถูกละเลยเพราะขาดวินัยทางการจัดการ
  - การจัดผลอย่างละเอียดจะทำไม่ได้หากไม่นิยามกระบวนการให้ชัด
  - ผลของการปรับกระบวนการใหม่จะคลุมเครือหากกระบวนการไม่ชัดเจน

CMM Structure



Key process areas by maturity level

Five levels of software process maturity



สรุป

- CMM เน้นที่การจัดการซอฟต์แวร์
- กระบวนการจะชัดเจนหรือไม่ขึ้นอยู่กับวุฒิภาวะในการทำงานตามกระบวนการนั้น
- CMM เป็นแบบจำลอง 5 ระดับ และแต่ละระดับแตกออกเป็น Key Process Areas
- แต่ละระดับต้องอาศัยความสามารถในระดับที่ต่ำกว่า

บรรณานุกรม

- คำบรรยาย - หลักสูตรแนะนำ CMM ของมหาวิทยาลัยคาร์เนกีเมลลอน
- SEI, The Capability Maturity Model : Guidelines for Improving the Software Process, Addison Wesley, 1994.
- Caputo, Kim, CMM Implementation Guide, Addison Wesley, 1998.
- Hollenbach, Craig,et al, Combining Quality and Software Improvement, Communication of the ACM, June 1997, Vol 40., No.6 .