บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันธุรกิจซอฟต์แวร์ได้เจริญเติบโตอย่างรวดเร็ว ทำให้เกิดจำนวนนักพัฒนาซอฟต์แวร์ขึ้นมาอย่างมากมาย จนทำให้คุณภาพของนักพัฒนาซอฟต์แวร์ไม่มีมาตรฐาน ทำให้งานที่ผลิตออกมาไม่มีคุณภาพ และไม่มีโปรแกรมหรือระบบที่ช่วยพัฒนาตัวนักพัฒนาซอฟต์แวร์เอง ในอดีตคุณภาพและความชำนาญของนักพัฒนาซอฟต์แวร์จะขึ้นอยู่กับประสบการณ์ทำงานของแต่ละบุคคล รวมไปถึงการศึกษาหาความรู้ใหม่ ๆ ให้กับตนเอง

จากปัญหาดังกล่าว จึงมีแนวคิดในการแก้ไขปัญหาด้วยทฤษฎี กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล หรือ Personal Software Process (PSP) เพื่อสร้างนักพัฒนาซอฟต์แวร์ให้มีวินัย ทักษะ และความรู้ในด้านกระบวนการพัฒนาซอฟต์แวร์ที่มีคุณภาพในระดับบุคคล ซึ่งเป็นมาตรฐานที่พัฒนาโดยสถาบัน Software Engineering Institute (SEI) ถือเป็นยุทธศาสตร์ที่ใช้ในการปักธงธุรกิจซอฟต์แวร์ไทยเข้าสู่ระดับโลก

Personal Software Process (PSP) คือ แบบจำลองการปรับปรุงกระบวนการซอฟต์แวร์ระดับบุคคล พัฒนาโดย Watts Humphrey และ Software Engineering Institute (SEI) โดยวัตถุประสงค์ของ PSP ก็คือต้องการฝึกวิศวกรซอฟต์แวร์ให้รับผิดชอบต่อการมีระเบียบวินัยในกระบวนการซอฟต์แวร์

PSP ถูกออกแบบมาเพื่อช่วยเหลือวิศวกรให้สามารถผลิตซอฟต์แวร์ที่มีคุณภาพสูง โดย PSP จะช่วยในด้านการประมาณ การวางแผน และการพัฒนาระบบซอฟต์แวร์ คล้ายกันกับ Capability Maturity Model (CMM) ของ SEI ที่อยู่บนหลักการของการปรับปรุงกระบวนการ แต่ในขณะที่ CMM มุ่งเน้นการพัฒนาความสามารถในระดับองค์กร, PSP จะมุ่งเน้นการพัฒนาคุณภาพที่ตัวของวิศวกรเป็นรายบุคคล

PSP ปกติจะเป็นหลักสูตรซึ่งใช้ในการเรียนการสอน เช่น ในมหาวิทยาลัย หรือฝึกอบรมเป็นหมู่คณะ โดย PSP จะเสนอขั้นตอนที่ถูกกำหนดมาแล้วอย่างดี มีรูปแบบ และมาตรฐานเพื่อปรับปรุงกระบวนการการพัฒนาซอฟต์แวร์เป็นรายบุคคล

ในกระบวนการของ PSP เป็นการเก็บข้อมูลโดยการวัดที่ผลผลิต และกระบวนการพัฒนาของผู้พัฒนาซอฟต์แวร์เป็นรายบุคคล และใช้การวัดเหล่านี้ในการขับเคลื่อนการเปลี่ยนแปลงพฤติกรรมการพัฒนาซอฟต์แวร์ของนักพัฒนา โดยวัตถุประสงค์หลักของการปรับปรุงกระบวนการ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การพัฒนาซอฟต์แวร์เป็นรายบุคคลนี้ คือ การลดข้อบกพร่อง (Defect) เพิ่มทักษะในการวางแผนและการประมาณค่า (Planning and Estimation) ซอฟต์แวร์

PSP ถูกออกแบบบนหลักการของการวางแผน และคุณภาพ

- คุณภาพของระบบซอฟต์แวร์ถูกพิจารณาจากคุณภาพขององค์ประกอบของโปรแกรม (Components)
- คุณภาพขององค์ประกอบของโปรแกรมถูกกำหนดโดยบุคคลผู้พัฒนา
- คุณภาพขององค์ประกอบของโปรแกรมถูกกำหนดโดยกระบวนการที่พัฒนา
- หัวใจของคุณภาพก็คือทักษะ การยอมรับ ความรับผิดชอบของผู้พัฒนา
- วิศวกรทุกคนต่างกัน ซึ่งเป็นผลอย่างมาก วิศวกรจึงต้องวางแผนงานของพวกเขาและแผนที่วางไว้ต้องอยู่บนข้อมูลการทำงานส่วนบุคคล
- เพื่อให้สมรรถนะในการทำงานเป็นไปอย่างสม่ำเสมอ วิศวกรควรใช้กระบวนการที่ถูกกำหนด นิยามไว้อย่างดี และสามารถวัดได้
- เพื่อที่จะผลิตผลิตภัณฑ์ที่มีคุณภาพ วิศวกรจะต้องรู้สึกรับผิดชอบต่อคุณภาพของผลิตภัณฑ์ของพวกเขาเอง ผลิตภัณฑ์ที่มีคุณภาพชั้นเยี่ยมจะไม่ถูกสร้างโดยความผิดพลาด วิศวกรจะต้องใช้ความพยายามอย่างหนักในการรักษาคุณภาพของงาน
- ค่าใช้จ่ายจะลดลงหากตรวจพบข้อบกพร่องที่จะต้องแก้ไขแต่เนิ่น ๆ
- ประสิทธิภาพการทำงานจะมากขึ้นหากป้องกันข้อบกพร่องมากกว่าการค้นหาแล้วแก้ไข
- ทางที่ดีที่สุดก็คือ ทางที่เร็วและประหยัดที่สุดในการทำงาน

เพื่อให้งานของวิศวกรรมซอฟต์แวร์เป็นไปในทางที่ถูกวิศวกรต้องวางแผนงานก่อนที่จะรับทำสัญญา หรือเริ่มการทำงาน และจะต้องกำหนดกระบวนการไว้เป็นแผนงาน เพื่อให้เข้าใจถึงสมรรถภาพในการทำงานเป็นรายบุคคล วิศวกรต้องทำการวัดเวลาที่ใช้ในแต่ละขั้นตอนของงาน จำนวนข้อบกพร่องที่สร้างขึ้น และที่กำจัดออก ขนาดของซอฟต์แวร์ที่สร้าง เพื่อให้คุณภาพของผลิตภัณฑ์เป็นไปอย่างสม่ำเสมอ วิศวกรต้องวางแผน วัด และติดตามคุณภาพของผลิตภัณฑ์ และจะต้องมุ่งเน้นที่คุณภาพตั้งแต่เริ่มโครงการ ในที่สุดแล้ว ต้องมีการวิเคราะห์ผลของงานแต่ละชิ้น

โปรแกรมประยุกต์เว็บได้รับความนิยมกันอย่างแพร่หลายและสามารถเข้าถึงได้ง่าย มีความสามารถในการอัพเดท และดูแลโดยไม่ต้องแจกจ่าย และติดตั้งซอฟต์แวร์บนเครื่องผู้ใช้ ซึ่งสะดวกกับผู้ใช้งานและง่ายต่อการควบคุมและบำรุงรักษาซอฟต์แวร์ของผู้ผลิต

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ทางผู้วิจัยจึงได้มีแนวความคิด คิดค้นรูปแบบในการพัฒนาคุณภาพของวิศวกรซอฟต์แวร์ในรูปแบบใหม่ซึ่งสามารถเปรียบเทียบข้อดีข้อเสียกับระบบเดิมดังตารางที่ 1.1

ตารางที่ 1.1 แสดงการเปรียบเทียบข้อดีข้อเสียระหว่างระบบเดิมกับระบบใหม่

คุณสมบัติ	ระบบเดิม (Microsoft Access)	ระบบใหม่ (Web Application)
การเรียกใช้งาน	ต้องมี Microsoft Access	ใช้งานได้ทันที
การใช้งาน	ยากและมีความซับซ้อนไม่เหมาะสำหรับผู้ใช้งานทั่วไป	ใช้งานและทำความเข้าใจได้ง่ายไม่ซับซ้อน
คู่มือในการใช้งาน(ภาษาไทย)	ไม่มี	มี
การสนับสนุน	ไม่เยอะเท่าที่ควรรู้จักแค่ในวงแคบ เช่น มหาวิทยาลัย หรือ องค์กรพัฒนาซอฟต์แวร์ต่างๆ	เป็น Web Application สามารถเข้าถึงได้ง่าย
ช่องทางในการแจ้งปัญหา	ไม่มี	มี
การใช้งานระดับองค์กร	ไม่รองรับ	รองรับ
ระบบรายงาน	ชัดเจน	ชัดเจนและเห็นภาพยิ่งขึ้น
ระบบสืบค้นข้อมูล	ไม่มี	มี
การ Update Version	มีปัญหา	ไม่มีปัญหา

1.2 วัตถุประสงค์ของการศึกษา

ในการศึกษาค้นคว้างานวิจัยในครั้งนี้ ผู้ค้นคว้าได้มุ่งหวังที่จะศึกษาเกี่ยวกับการพัฒนาคุณภาพของวิศวกรซอฟต์แวร์ โดยใช้โปรแกรมประยุกต์เว็บมาเป็นเครื่องมือในการพัฒนาเพื่อช่วยในส่วนของการทำให้เข้าถึงข้อมูลได้ง่ายของผู้ใช้ ซึ่งมีวัตถุประสงค์ในการศึกษาดังต่อไปนี้

1.2.1 เพื่อส่งเสริมพัฒนาในการพัฒนาคุณภาพตนเองแก่วิศวกรซอฟต์แวร์

1.2.2 เพื่อลดข้อบกพร่อง (Defect) เพิ่มทักษะในการวางแผนและประมาณค่า (Planning and Estimation) ของวิศวกรซอฟต์แวร์

1.2.3 วิศวกรซอฟต์แวร์สามารถเรียนรู้ได้ด้วยตนเองเองผ่านระบบพัฒนาคุณภาพของวิศวกรซอฟต์แวร์โดยกระบวนการซอฟต์แวร์ระดับบุคคลบนโปรแกรมประยุกต์เว็บ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

1.3 ประโยชน์ที่รับได้รับจากการศึกษาเชิงทฤษฎีและเชิงประยุกต์

การศึกษาค้นคว้างานวิจัยระบบพัฒนาคุณภาพของวิศวกรซอฟต์แวร์โดยกระบวนการซอฟต์แวร์ระดับบุคคลบนโปรแกรมประยุกต์เว็บ ผู้ค้นคว้าได้พิจารณาถึงประโยชน์ที่จะได้รับจากการศึกษาในเชิงทฤษฎีและเชิงประยุกต์ใช้ในครั้งนี้ คือ

1.3.1 วิศวกรซอฟต์แวร์มีความรู้ความเข้าใจในทฤษฎีกระบวนการซอฟต์แวร์ระดับบุคคลเพิ่มมากขึ้น

1.3.2 ทฤษฎีเป็นที่แพร่หลายออกไปในวงกว้าง

1.3.3 การประมาณขนาดของซอฟต์แวร์แม่นยำขึ้น

1.3.4 การประมาณเวลาที่ใช้ในการพัฒนาแม่นยำขึ้น

1.3.5 นำข้อมูลหรือผลสรุปที่ได้มาพัฒนาตนเองเพื่อให้สามารถสร้างงานที่มีคุณภาพ

1.3.6 คุณภาพของกระบวนการและจำนวนของข้อบกพร่องที่ถูกค้นพบก่อนการคอมไพล์โปรแกรมเพิ่มขึ้น

1.3.7 คุณภาพของผลิตภัณฑ์ สามารถค้นพบข้อบกพร่องขณะทำ Unit Test พัฒนามากขึ้น

1.3.8 การเพิ่มผลิตภาพส่วนบุคคล (Personal Productivity) โดยดูจากค่า LOC/HOUR

1.3.9 สามารถพัฒนากระบวนการทำงานให้มีคุณภาพได้ด้วยตนเอง

1.4 ขอบเขตของการศึกษาวิจัย

การพัฒนาระบบพัฒนาคุณภาพของวิศวกรซอฟต์แวร์โดยกระบวนการซอฟต์แวร์ระดับบุคคลบนโปรแกรมประยุกต์เว็บ ได้แบ่งขอบเขตของงานออกเป็น 3 ส่วนหลักดังนี้

1.4.1 ขอบเขตโดยรวม

- สามารถใช้งานตามทฤษฎี Personal Software Process ได้ตั้งแต่ PSP0 ถึง PSP3
- ใช้หลักสถิติในการวิเคราะห์, วัดผลและคำนวณผล
- ใช้โปรแกรมภาษาพีเอชพี (PHP) ในการพัฒนา
- ใช้มายเอสคิวแอล (MySQL) ในการจัดการกับฐานข้อมูล
- ใช้ ISO 29110 VSE เป็นมาตรฐานในการพัฒนาซอฟต์แวร์
- กลุ่มเป้าหมายคือ วิศวกรซอฟต์แวร์ที่ต้องการพัฒนาตนเองทุกคน

1.4.2 ขอบเขตในส่วนประมวลผลข้อมูล

- ระบบคำนวณข้อมูลที่ได้จากการเก็บข้อมูลกระบวนการทำงานของผู้ใช้งาน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- ระบบวิเคราะห์และวัดผลผู้ใช้งานว่าควรเพิ่มทักษะในด้านใดเพื่อไปพัฒนาทักษะในด้านนั้นซึ่งจะทำให้การทำงานเป็นไปอย่างมีคุณภาพ

1.4.3 ขอบเขตในส่วนติดต่อประสานกับผู้ใช้งาน

- ระบบเก็บข้อมูลรายละเอียดในการเก็บสถิติในการทำงานตามทฤษฎี PSP
- ระบบเก็บข้อมูลสถิติในการใช้งาน
- ระบบสืบค้นสามารถสืบค้นจากคำหลัก (Keyword) หรือ สืบค้นจากหัวข้อ (Topic) ที่เกี่ยวข้องกับการทำงานได้
- ระบบสามารถสืบค้นข้อมูลการทำงานตามทฤษฎี PSP ในแต่ละหัวข้อ
- ระบบสามารถสืบค้นข้อมูลผลลัพธ์ใน PSP ได้
- ระบบสามารถแสดงผลข้อมูลผลลัพธ์ใน PSP ว่าวิศวกรซอฟต์แวร์ควรเพิ่มทักษะในด้านใดเพื่อให้ วิศวกรซอฟต์แวร์ทุกคนมีสมรรถนะในการทำงานที่เป็นไปอย่างมีคุณภาพได้
- ระบบสามารถแสดงรายงานวิเคราะห์เปรียบเทียบข้อมูลประสิทธิภาพการแก้ปัญหาได้
- ระบบสามารถแสดงรายงานวิเคราะห์เปรียบเทียบข้อมูลระยะเวลาในการวางแผนและประมาณค่าการแก้ปัญหาได้
- ระบบสามารถแสดงรายงานวิเคราะห์เปรียบเทียบข้อมูลจำนวนข้อบกพร่องได้
- ระบบสามารถแสดงรายงานข้อมูลการทำงานได้
- ระบบสามารถแสดงรายงานข้อมูลโครงการได้
- ระบบสามารถแสดงรายงานข้อมูลผู้ใช้งานได้

1.5 นิยามคำศัพท์

1. **วิศวกรรมซอฟต์แวร์ (Software Engineering)** หมายถึง เป็นศาสตร์เกี่ยวกับวิศวกรรมด้านซอฟต์แวร์ มีเนื้อหาเกี่ยวข้องกับการใช้กระบวนการทางวิศวกรรมในการดูแลการผลิต ตั้งแต่การเริ่มเก็บความต้องการ การตั้งเป้าหมายของระบบ การออกแบบ กระบวนการพัฒนา การตรวจสอบ การประเมินผล การติดตามโครงการ การประเมินต้นทุน การรักษาความปลอดภัย ไปจนถึงการคิดราคาซอฟต์แวร์เป็นต้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2. **กระบวนการซอฟต์แวร์ระดับบุคคล (Personal Software Process)** หมายถึง กระบวนการที่นำมาปรับปรุงกระบวนการการพัฒนาซอฟต์แวร์ของวิศวกรซอฟต์แวร์ให้มีคุณภาพมากขึ้น
3. **Software Engineering Institute (SEI)** หมายถึง หน่วยงานที่มีหน้าที่พัฒนาเกี่ยวกับมาตรฐานการพัฒนาซอร์ฟแวร์
4. **Capability Maturity Model (CMM)** หมายถึง แบบจำลองวุฒิภาวะความสามารถ มีชื่อเรียกอีกอย่างหนึ่งว่า SW-CMM (The Capability Maturity Model for Software) ถือกำเนิดจาก Software Engineering Institute (SEI) ของมหาวิทยาลัยคาร์เนกีเมลลอน (Carnegie Mellon University) เป็นมาตรฐานที่กำหนดขึ้นเพื่อวัดความเชื่อมั่นและคุณภาพของกระบวนการพัฒนา ซอฟต์แวร์ของบริษัทพัฒนาซอฟต์แวร์ (Software House)
5. **โปรแกรมประยุกต์เว็บ (Web Application)** หมายถึง โปรแกรมประยุกต์ที่เข้าถึงด้วยโปรแกรมค้นดูเว็บผ่านเครือข่ายคอมพิวเตอร์อย่างอินเทอร์เน็ตหรืออินทราเน็ต
6. **กระบวนการ PSP0 ถึง PSP3** หมายถึง ลำดับขั้นตอนในกระบวนการซอฟต์แวร์ระดับบุคคล
7. **ISO 29110 VSE** หมายถึง แนวคิดยุคใหม่ของ ISO ที่จะเน้นการเติบโตของอุตสาหกรรมขนาดกลางและเล็กรวมทั้งผู้ประกอบการใหม่ที่เข้ามาให้มีโอกาสในการแข่งขันตามแนวทางการพัฒนา Outsourcing ซึ่งในอดีตที่ผ่านมามาตรฐานวิศวกรรมซอฟต์แวร์ได้ถูกทำให้เป็นเรื่องที่เข้าใจยากและมีความสลับซับซ้อนยุ่งยากในการปฏิบัติตาม ประกอบกับมาตรฐานซอฟต์แวร์ระดับสากลที่มีอยู่ในปัจจุบันจะหมาะสมกับการปฏิบัติงานขององค์กรขนาดใหญ่ ISO 29110 จึงถูกพัฒนาด้วยแนวคิดพื้นฐานเพื่อสนับสนุนองค์กรขนาดเล็กให้มีโอกาสในการปรับปรุงกระบวนการและรับรองคุณภาพในระดับสากล
8. **การคอมไพล์โปรแกรม (Compile)** หมายถึง ผลลัพธ์ของการแปลโปรแกรม (คอมไพล์) โดยทั่วไป ที่เรียกว่า ออบเจกต์โค้ด จะประกอบด้วยภาษาเครื่อง (Machine Code) ที่เต็มไปด้วยข้อมูลเกี่ยวกับชื่อและสถานที่ของแต่ละจุด และการเรียกใช้วัตถุภายนอก (Link Object) (สำหรับฟังก์ชันที่ไม่ได้อยู่ใน อ็อบเจกต์) สำหรับเครื่องมือที่เราใช้รวม อ็อบเจกต์เข้าด้วยกัน จะเรียกว่าโปรแกรมเชื่อมโยงเพื่อที่ผลลัพธ์ที่ออกมาในขั้นสุดท้าย เป็นไฟล์ที่ผู้ใช้งานทั่วไปสามารถใช้งานได้สะดวก

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

9. **Unit Test** หมายถึง บางครั้งเรียกอีกอย่างว่า Module Testing เป็นการทดสอบโปรแกรมทีละโมดูล เพื่อหาข้อผิดพลาดที่จะเกิดขึ้นภายในการทำงานของแต่ละโมดูล
10. **LOC/HOUR** หมายถึง หน่วยวัดของ Productivity ที่นำมาใช้ในการประมาณขนาดของโปรแกรมและแรงงานที่ใช้พัฒนาโปรแกรม

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved