



เพิ่มคุณภาพกระบวนการพัฒนาซอฟต์แวร์ระดับบุคคลด้วยเทคนิคหมวกความคิดหกใบ

An Increase Quality of Personal Software Process Development with the Six Thinking Hats Technique

ทกอลชัย อุตตรานหิ ปร.ด. (Takolchai Uttranahi, Ph.D.)¹

บทคัดย่อ

สารสนเทศเป็นเครื่องมือสำคัญขององค์กรในการปฏิบัติงานให้บรรลุเป้าหมายทางธุรกิจโดยมีซอฟต์แวร์เป็นส่วนประกอบหลัก ดังนั้น การพัฒนาซอฟต์แวร์ให้มีคุณภาพจึงเป็นเรื่องสำคัญ การพัฒนาซอฟต์แวร์ทำได้ที่ทั้งในระดับองค์กร ระดับทีม และระดับบุคคล กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคลเป็นแนวทางการปฏิบัติที่ดีในการพัฒนาซอฟต์แวร์ให้มีคุณภาพ ประกอบด้วยขั้นตอนและกิจกรรมย่อยที่มีการใช้ความคิดสำหรับการปฏิบัติที่แตกต่างกันมาก ทำให้นักพัฒนาเกิดความสับสน ส่งผลต่อคุณภาพของซอฟต์แวร์ การใช้เทคนิคหมวกความคิดหกใบ ซึ่งเป็นเครื่องมือที่มีประสิทธิภาพในการแบ่งแยกความคิดแต่ละแบบออกจากกันอย่างเป็นระบบ จะช่วยแก้ปัญหาดังกล่าวได้ โดยนักพัฒนาใช้หมวกความคิดหกใบที่สอดคล้องกับขั้นตอนการพัฒนาที่กำลังดำเนินการเป็นกรอบในการทำงานสำหรับขั้นตอนนั้น ทำให้พัฒนาได้เร็วและผลงานที่ได้มีข้อผิดพลาดน้อย

คำสำคัญ: การพัฒนาซอฟต์แวร์ระดับบุคคล หมวกความคิดหกใบ คุณภาพซอฟต์แวร์

Abstract

Information system which composed of software as a major component is an important tool for organizations in order to achieve their business goals. Thus, the development of software quality is an important issue. Software can be developed at organizational, team and personal levels. The personal software process development is the best practices in developing the software quality. This needs different thoughts and various steps and procedures which can confuse developer and affect software quality. The Six Thinking Hats technique, the powerful tool which can systematically facilitate the different type of thinking will be able to resolve this problem. The developers use a hat which consistent to each procedure as a framework for development, resulting in faster development and less errors.

Keywords: Personal Software Process, Six Thinking Hats, Software Quality

¹ ผู้จัดการฝ่ายเทคโนโลยีระบบสารสนเทศ ศูนย์ข้อมูลสังหาริมทรัพย์ ธนาคารอาคารสงเคราะห์



บทนำ

ระบบสารสนเทศมีบทบาทสำคัญต่อการบริหารจัดการและพัฒนาองค์กรสมัยใหม่ โดยประกอบด้วยส่วนประกอบสำคัญ 7 ส่วน (จารึก ชุกติติกุล, 2553) คือ 1) ฮาร์ดแวร์ (hardware) 2) ซอฟต์แวร์ (software) 3) ข้อมูล (data) 4) สารสนเทศ (information) 5) บุคลากร (peopleware) 6) เครือข่าย (network) และ 7) ขั้นตอนวิธี (Procedures) ทั้งนี้ ซอฟต์แวร์จัดเป็นส่วนประกอบสำคัญของระบบสารสนเทศเนื่องจากเป็นส่วนที่เชื่อมโยงส่วนประกอบอื่นๆ ให้ทำงานร่วมกันเพื่อเกิดสัมฤทธิ์ผลตามวัตถุประสงค์ของการออกแบบระบบ ซอฟต์แวร์ควบคุมการทำงานของฮาร์ดแวร์ในการนำเข้าข้อมูลมาเพื่อประมวลผลเป็นสารสนเทศตามขั้นตอนวิธีที่กำหนด และแสดงผลออกทางจอภาพหรือเครื่องพิมพ์ รวมถึงกระจายสารสนเทศไปสู่ผู้ใช้ในระดับต่างๆ ผ่านทางเครือข่ายคอมพิวเตอร์ นอกจากนี้ซอฟต์แวร์ยังทำหน้าที่ติดต่อกับผู้ใช้ผ่านทางส่วนต่อประสานผู้ใช้ของระบบอีกด้วย

การพัฒนาซอฟต์แวร์จึงสำคัญยิ่งต่อการพัฒนาระบบสารสนเทศ และกระบวนการพัฒนาซอฟต์แวร์ที่ไม่มีมาตรฐานจะทำให้งานที่ผลิตออกมาไม่มีคุณภาพส่งผลให้ระบบสารสนเทศที่ไม่มีคุณภาพตามไปด้วย

การพัฒนาซอฟต์แวร์ทำได้ทั้งในระดับองค์กร ระดับทีม และระดับบุคคล การพัฒนาซอฟต์แวร์ในระดับองค์กรและระดับทีม ใช้นักพัฒนาจำนวนมาก มีการแบ่งงานในความรับผิดชอบออกเป็นส่วนตามความถนัด เช่น นักวิเคราะห์ความต้องการ นักออกแบบและพัฒนาหน้าจอ นักเขียนรหัสคำสั่ง นักทดสอบโปรแกรม ฯลฯ แต่การพัฒนาซอฟต์แวร์ระดับบุคคลใช้นักพัฒนาเพียงคนเดียวเป็นผู้ดำเนินกิจกรรมและขั้นตอนที่หลากหลายทั้งหมดจนกระทั่งพัฒนาขึ้นเป็นซอฟต์แวร์สมบูรณ์ทำให้เกิดความสับสนในการแบ่งแยกความคิดในการทำงานที่แตกต่างกัน เช่น การเขียนรหัสคำสั่งใช้ความคิดเกี่ยวกับตรรกะการออกแบบหน้าจอใช้ความคิดเกี่ยวกับการออกแบบส่วนต่อประสานผู้ใช้ ความสวยงามและความสะดวก เป็นต้น ส่งผลต่อคุณภาพของผลงานในแต่ละกิจกรรมและคุณภาพโดยรวมของซอฟต์แวร์

การนำทฤษฎีหมวดความคิดหกใบซึ่งเป็นเครื่องมือในการแบ่งแยกความคิดแต่ละแบบออกจากกันอย่างเป็นระบบมาประยุกต์ใช้ให้สอดคล้องกับกิจกรรมในแต่ละขั้นตอนของการพัฒนาช่วยสนับสนุนให้ซอฟต์แวร์มีคุณภาพเพิ่มขึ้น เนื่องจากลดความสับสนในบทบาทหน้าที่ของนักพัฒนาในขั้นตอนที่แตกต่างกัน ทำให้นักพัฒนามุ่งความสนใจและใช้ความคิดในประเด็นที่กำลังดำเนินการได้อย่างมีประสิทธิภาพ โดยคุณภาพวัดได้จากระยะเวลาในการส่งมอบงาน จำนวนข้อผิดพลาดในโปรแกรมและความพึงพอใจของผู้ใช้

1. วิธีการพัฒนาซอฟต์แวร์

วิธีที่ใช้ในการพัฒนาซอฟต์แวร์แบ่งเป็นวิธีใหญ่ๆ ได้ 2 วิธี คือ 1. วิธีแบบดั้งเดิม (Traditional Methodology) โดยมีแบบจำลองน้ำตก (Waterfall Model) เป็นกระบวนการหลัก ประกอบด้วยขั้นตอนการทำงานเรียงเป็นลำดับต่อเนื่องกันไป 5 ขั้นตอน (Satzinger et al., 2011) ได้แก่ 1) ขั้นการวางแผน (Planning) 2) ขั้นการวิเคราะห์ (Analysis) 3) ขั้นการออกแบบ (Design) 4) ขั้นการพัฒนา (Implementation) และ 5) ขั้นการใช้งาน (Deployment) และ 2. วิธีแบบคล่องตัว (Agile Methodology) ซึ่งเป็นวิธีสมัยใหม่ที่เน้นการติดต่อสื่อสาร และผลผลิตที่เป็นชิ้นงานของซอฟต์แวร์ มากกว่าความสมบูรณ์ของเอกสาร โดยมีคุณลักษณะเด่น (Davis, 2012) คือ คล่องตัว ปรับเปลี่ยนได้ ให้ผลงานรวดเร็ว และใช้ทรัพยากรคุ้มค่า ทำให้เป็นวิธีพัฒนาซอฟต์แวร์ที่ได้รับความนิยมในปัจจุบัน

ข้อแตกต่างระหว่างวิธีแบบดั้งเดิมและวิธีแบบคล่องตัว คือ กระบวนการพัฒนาของวิธีแบบดั้งเดิมเป็นแบบเรียงลำดับ (Sequential order) มีเอกสารประกอบการทำงานทุกขั้นตอน และเอกสารที่ใช้อ้างอิงการทำงานจะไม่มีการแก้ไขหรือทบทวนในระหว่างการพัฒนา (Nir, 2014) ขณะที่กระบวนการพัฒนาของวิธีแบบคล่องตัวเป็นแบบวนรอบ (Cyclical order)



โดยซอฟต์แวร์จะถูกพัฒนาขึ้นให้สามารถทำงานได้เป็นส่วนๆ เพิ่มขึ้นทีละส่วน (Iterative and Incremental Development) พร้อมเอกสารประกอบสำหรับส่วนนั้นๆ โดยส่วนที่ยังไม่ถูกผลิตก็จะเก็บความต้องการเป็นรายการสะสม (Features list) เพื่อรอผลิตในรอบต่อไป (Davis, 2012; Subramaniam & Hunt, 2005)

2. การควบคุมคุณภาพการพัฒนาซอฟต์แวร์

การพัฒนาซอฟต์แวร์ แบ่งได้เป็น 3 ระดับ คือ 1) ระดับองค์กร หมายถึง การพัฒนาซอฟต์แวร์ขนาดใหญ่ที่มีความซับซ้อนโดยองค์กรหรือหน่วยงานที่ทำหน้าที่พัฒนาซอฟต์แวร์โดยเฉพาะ มีการจัดการโครงการ มีทีมงานพัฒนา มีการแบ่งหน้าที่และขอบเขตความรับผิดชอบชัดเจน มีการจัดทำเอกสารประกอบในแต่ละขั้นตอน การทำงาน โดยมาตรฐานที่ใช้ควบคุมคุณภาพกระบวนการพัฒนาซอฟต์แวร์ขององค์กร (Paulk et al., 1993) คือ แบบจำลองวุฒิภาวะและความสามารถ (Capability Maturity Model: CMM) 2) ระดับทีม หมายถึง การพัฒนาซอฟต์แวร์ที่พัฒนาโดยนักพัฒนาตั้งแต่ 2-5 คนขึ้นไปจนถึงมากกว่า 20 คน มีการแบ่งบทบาทหน้าที่ของแต่ละคนอย่างชัดเจน โดยมีเป้าหมายผลิตซอฟต์แวร์เดียวกัน และความสำเร็จของการพัฒนานั้นขึ้นอยู่กับผลงานของทุกคน โดยมาตรฐานที่ใช้ควบคุมคุณภาพกระบวนการพัฒนาซอฟต์แวร์ของทีม (Humphrey, 2000) คือ กระบวนการพัฒนาซอฟต์แวร์ระดับทีม (Team Software Process: TSP) และ 3) ระดับบุคคล หมายถึง การพัฒนาซอฟต์แวร์ที่รับผิดชอบโดยนักพัฒนาซอฟต์แวร์เพียงคนเดียว ส่วนใหญ่เป็นการพัฒนาด้วยบุคลากรภายในองค์กรที่ต้องการใช้ซอฟต์แวร์นั้นและเป็นซอฟต์แวร์ขนาดเล็ก โดยมีมาตรฐานที่ใช้ควบคุมคุณภาพ (Humphrey, 2000) คือ กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล (Personal Software Process: PSP)

ทั้งนี้ มาตรฐานที่ใช้ในการควบคุมคุณภาพกระบวนการพัฒนาซอฟต์แวร์ทุกระดับ พัฒนาขึ้นโดยสถาบันวิศวกรรมซอฟต์แวร์ (Software Engineering Institute: SEI) แห่งมหาวิทยาลัยคาน์egieเมลลอน (Carnegie Mellon)

3. มาตรฐานการพัฒนาซอฟต์แวร์ระดับบุคคล

การพัฒนาซอฟต์แวร์ระดับบุคคลในอดีตไม่มีการกำหนดมาตรฐานเพื่อควบคุมคุณภาพการพัฒนาทำให้คุณภาพของซอฟต์แวร์ขึ้นอยู่กับคุณภาพของบุคคล หมายถึง ทักษะ การฝึกอบรม และความรู้ความสามารถของนักพัฒนา (Subramaniam & Hunt, 2005) รวมไปถึงความสนใจศึกษาค้นคว้าเพื่อแสวงหาวิธีการใหม่ๆ ในการทำงานอยู่เสมอ ต่อมาสถาบันวิศวกรรมซอฟต์แวร์ แห่งมหาวิทยาลัยคาน์egieเมลลอน ซึ่งประสบความสำเร็จในการพัฒนามาตรฐานกระบวนการพัฒนาซอฟต์แวร์ระดับองค์กรและระดับทีมมาแล้ว จึงได้พัฒนามาตรฐานการพัฒนาซอฟต์แวร์ในระดับบุคคลขึ้น เรียกว่า กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล (Personal Software Process: PSP) แบ่งเป็น 4 ระดับ โดยนำแนวคิดจากการจัดระดับในแบบจำลองวุฒิภาวะและความสามารถ (Capability Maturity Model: CMM) ของมาตรฐานการพัฒนาซอฟต์แวร์ระดับองค์กร มาประยุกต์ใช้ ดังนี้

ระดับ 0 (PSP0) คือ กระบวนการส่วนบุคคลพื้นฐาน (Baseline Personal Process) หมายถึง การวัดและบันทึกค่าพื้นฐานของกระบวนการ ได้แก่ เวลา และข้อบกพร่องโดยนักพัฒนาเพื่อเปรียบเทียบผลการพัฒนาแต่ละครั้ง ระดับนี้จะมีการกำหนดขั้นตอนการพัฒนา ซอฟต์แวร์ไว้ 6 ขั้นตอน คือ 1) ขั้นการวางแผน (Planning) 2) ขั้นการออกแบบ (Design) 3) ขั้นการเขียนรหัสคำสั่ง (Code) 4) ขั้นการรวมรหัส (Compile) 5) ขั้นการทดสอบ (Test) และ 6) ขั้นการชันสูตร (Post mortem)

ระดับ 1 (PSP1) คือ การบริหารโครงการส่วนบุคคล (Personal Project Management) หมายถึง การนำเทคนิคการบริหารโครงการมาใช้ในกระบวนการโดยนักพัฒนา ได้แก่ การวางแผน การจัดตารางทำงาน การประมาณขนาดโปรแกรม การรายงาน เป็นต้น



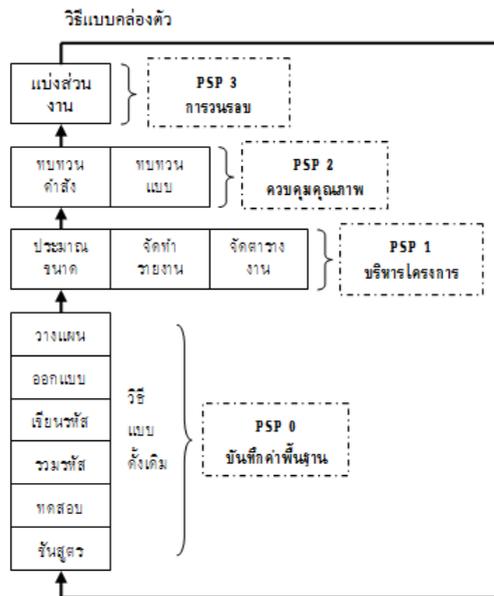
ระดับ 2 (PSP2) คือ การจัดการคุณภาพส่วนบุคคล (Personal Quality Management) คือ การจัดการคุณภาพกระบวนการในระดับบุคคลโดยนักพัฒนา ประกอบด้วย กิจกรรมการทบทวนรหัสคำสั่ง (Code Review) หมายถึง การตรวจสอบความถูกต้องของการทำงานของซอฟต์แวร์ตามรหัสคำสั่งที่เขียน และกิจกรรมการทบทวนการออกแบบ (Design Review) หมายถึง การตรวจสอบความถูกต้องของการออกแบบซอฟต์แวร์ในด้านคุณสมบัติสำคัญ 4 ด้าน คือ 1) คุณสมบัติเกี่ยวกับการปฏิบัติการ (Operational Specifications) 2) คุณสมบัติเกี่ยวกับหน้าที่งาน (Functional Specifications) 3) คุณสมบัติเกี่ยวกับสถานะ (State Specifications) และ 4) คุณสมบัติเกี่ยวกับตรรกะ (Logic Specifications)

ระดับ 3 (PSP3) คือ กระบวนการวนรอบส่วนบุคคล (Cyclic Personal Process) คือ การพัฒนาเป็นวงรอบของนักพัฒนาด้วยการแบ่งซอฟต์แวร์ออกเป็นส่วนๆ ตามลำดับความสำคัญของการพัฒนา และพัฒนาส่วนที่เหลือในวงรอบถัดไป ด้วยเทคนิคนี้ทำให้นักพัฒนาสามารถพัฒนาซอฟต์แวร์ที่มีขนาดใหญ่ได้

เห็นว่ากระบวนการพัฒนาซอฟต์แวร์ระดับบุคคลไม่ได้เป็นมาตรฐานเช่นเดียวกับแบบจำลองวุฒิภาวะและความสามารถ แต่เป็นแนวทางปฏิบัติหรือข้อเสนอแนะการดำเนินการแต่ละขั้นสำหรับกระบวนการพัฒนาเท่านั้น การวัดคุณภาพของนักพัฒนาวัดจากข้อมูลพื้นฐานที่ทำการบันทึกเอาไว้ เช่น เวลาที่ใช้ จำนวนข้อบกพร่อง และขนาดของซอฟต์แวร์ โดยนำข้อมูลเหล่านี้มาใช้เป็นสถิติเพื่อการวางแผนการพัฒนาในอนาคตหรือใช้ติดตามความก้าวหน้าของงานที่กำลังทำอยู่ในปัจจุบัน

ข้อดีของการพัฒนาซอฟต์แวร์ระดับบุคคล (Grutter & Ferber, 2002) คือ 1) ช่วยเพิ่มประมาณขนาดซอฟต์แวร์รวมถึงเวลาและแรงงานที่ใช้ได้แม่นยำขึ้น และ 2) ช่วยเพิ่มคุณภาพซอฟต์แวร์ ในด้านความหนาแน่นข้อผิดพลาด (Defect Density) ที่ลดลง และเพิ่มคุณภาพกระบวนการ ในด้านการป้องกันข้อผิดพลาดของซอฟต์แวร์ เนื่องจากสามารถค้นพบข้อบกพร่องก่อนการรวมคำสั่งและขณะทำการทดสอบได้เพิ่มขึ้น

ข้อด้อยของการพัฒนาซอฟต์แวร์ระดับบุคคล (Couprie, 1998) คือ 1) ไม่ได้ให้ความสำคัญต่อขั้นตอนการรวบรวมความต้องการซึ่งเป็นสิ่งสำคัญในการสร้างซอฟต์แวร์ 2) ขั้นตอนการทำงานที่มากขึ้น การวัดและบันทึกค่าต่างๆ เป็นการเพิ่มภาระและความสับสนแก่นักพัฒนา และ 3) นักพัฒนามีเครื่องมือช่วยน้อยในกระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล แสดงให้เห็นดังภาพที่ 1



ภาพที่ 1 กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล



4. หมวกความคิดหกใบ

หมวกความคิดหกใบ (Six Thinking Hats) เป็นทฤษฎีคุณภาพเชิงจิตวิทยาของเอ็ดเวิร์ด เดอ โบโน (Edward, 1999) โดยมีแนวคิดที่ว่าหากคนเราพยายามทำหลายเรื่องพร้อม ๆ กันในเวลาเดียวกันมากเกินไปจะทำให้เกิดความคิดสับสน ส่งผลให้สิ่งที่คิดหรืองานที่ทำไม่มีประสิทธิภาพเท่าที่ควร หรือไม่ประสบความสำเร็จ จึงจำเป็นต้องแยกความคิดแต่ละแบบออกจากกัน เปรียบเหมือนกับการพิมพ์ภาพสีของเครื่องพิมพ์ที่ทำการพิมพ์ที่ละสีจนครบ สุดท้ายก็จะได้ภาพสีที่สมบูรณ์ ดังนั้น จึงกำหนดสัญลักษณ์แทนการคิดในเรื่องที่แตกต่างกันขึ้นเป็นหมวกหกใบที่มีสีแตกต่างกันสำหรับความคิดที่แตกต่างกันหกแบบของคน ได้แก่ สีขาว สีแดง สีดำ สีเหลือง สีเขียว และสีฟ้า มีความหมาย ดังนี้

หมวกสีขาว (White Hat) หมายถึง ความคิดที่เกี่ยวกับข้อเท็จจริง (Facts) ข้อมูลที่เป็นความจริง และสารสนเทศ (Information) ที่เกี่ยวข้อง

หมวกสีแดง (Red Hat) หมายถึง ความคิดที่เกี่ยวกับอารมณ์ (Emotion) และความรู้สึก (Feeling) ปฏิกริยาที่สนองตอบต่อความรู้สึกที่เกิดขึ้นในใจหรือสัญชาตญาณ

หมวกสีฟ้า (Blue Hat) หมายถึง ความคิดที่เกี่ยวกับการควบคุม (Control) และตรวจสอบ (Monitor) การจัดการ (Managing) หรือการสรุปเรื่องราวทั้งหมด (Summary and Conclusion)

หมวกสีเขียว (Green Hat) หมายถึง ความคิดที่เกี่ยวกับการสร้างสรรค์ (Creativity) สิ่งใหม่ที่ไม่เคยมีมาก่อน การศึกษา ค้นคว้า สำรวจสิ่งใหม่ ๆ (Exploration) การกระตุ้น (Provocation) การมุ่งไปข้างหน้า (Movement) มากกว่าการตัดสินใจ (Judgment)

หมวกสีเหลือง (Yellow Hat) หมายถึง ความคิดเชิงบวก ปฏิกริยาตอบสนองในทางบวก (Positive) ธรรมชาติที่ระบุบุคลิกที่เป็นผลประโยชน์ (Benefits) และข้อตกลงที่ลงรอยกัน (Harmony)

หมวกสีดำ (Black Hat) หมายถึง ความคิดที่เกี่ยวกับความผิดพลาด ธรรมชาติที่ระบุถึงเหตุผลที่ควรระมัดระวัง (Caution) การดูแล (Careful) ความรอบคอบ (Conservation) หรือสิ่งที่ป็นด้านลบ (Negative)

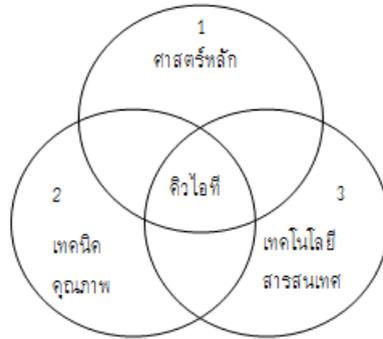
แนวคิดหมวกความคิดหกใบได้รับการยอมรับในฐานะที่เป็นเครื่องมือเพิ่มทักษะในการคิดเรื่องที่มีความสำคัญ การทำงานร่วมกัน การสื่อสาร และความคิดสร้างสรรค์ (The Opportunity Thinker, 2014) ในทางธุรกิจ พบว่าเป็นเครื่องมือที่ช่วยให้การประชุมมีประสิทธิภาพ ใช้เวลาน้อยลง และช่วยให้ตัดสินใจได้ดีขึ้น โดยหลายบริษัทนำมาประยุกต์ใช้และประสบความสำเร็จ เช่น บริษัท 3M ใช้ในการพัฒนาผลิตภัณฑ์ใหม่ (Curtin, 2014) บริษัทผลิตอาหารใช้ในการทำงานร่วมกันของทีมวิจัยและพัฒนาผลิตภัณฑ์ (Hormel Foods Corporation, 2010) และอื่น ๆ อีกหลายบริษัท (Stennes, 2004) นอกจากนี้ยังมีงานวิจัยที่เกี่ยวข้อง เช่น การเปรียบเทียบผลสัมฤทธิ์ทางการเรียนวิชาการปฏิบัติการฐานข้อมูลปีการศึกษา 1/2555 ของมหาวิทยาลัยราชภัฏสวนดุสิต ที่เรียนด้วยเทคนิคหมวกความคิดหกใบและการเรียนแบบปกติ (ปริศนา มัชฌิมา, 2556) พบว่าผลสัมฤทธิ์ทางการเรียนด้วยเทคนิคหมวกความคิดหกใบกับการเรียนแบบปกติแตกต่างกันอย่างมีนัยสำคัญ และนักศึกษามีคะแนนเฉลี่ยมากกว่ากลุ่มที่เรียนแบบปกติ และผลการวิจัยการนำแนวคิดหมวกความคิดหกใบไปใช้ในชั้นเรียนพยาบาลประจำห้องผ่าตัด พบว่า ช่วยพัฒนาความคิดสร้างสรรค์และความสามารถในการคิดเรื่องที่มีความสำคัญ (Critical Thinking) ส่งผลให้เกิดการพัฒนาทักษะในการดูแลผู้ป่วยดีขึ้น (Karadağ et al., 2009)

5. เทคโนโลยีสารสนเทศคุณภาพ

เทคโนโลยีสารสนเทศคุณภาพเรียกย่อว่าคิวไอที (Quality Information Technology: QIT) เป็นทฤษฎีคุณภาพกระบวนการที่ (จารึก ชุกติติกุล, 2553) พัฒนาขึ้นโดยมีหลักการสำคัญ คือ การใช้เทคโนโลยีสารสนเทศหรือไอทีบูรณาการร่วมกับเทคนิคคุณภาพของศาสตร์หลักของกระบวนการนั้นเพื่อยกระดับคุณภาพของกระบวนการให้สูงขึ้น ดังแสดงในภาพที่ 2



แนวคิดและหลักการดังกล่าวได้มีผู้นำไปวิจัยและประสบความสำเร็จในการประยุกต์ใช้ในระบบงานที่แตกต่างกัน เช่น นำไปใช้ด้านการเกษตร (ชนิษฐา กุลนาวิน, 2557) ใช้ด้านบริหารองค์กร (ชนพร ธาวนพงษ์, 2557) และใช้ด้านการดูแลผู้ป่วย (วารภรณ์ อ่ำขวัญยืน, 2556) เป็นต้น

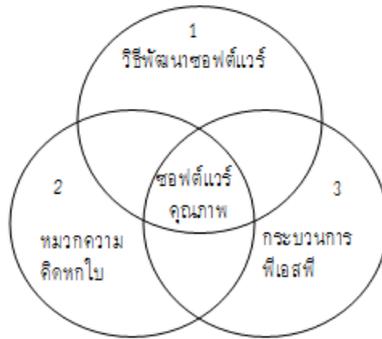


ภาพที่ 2 การบูรณาการศาสตร์หลัก เทคนิคคุณภาพ และเทคโนโลยีสารสนเทศตามทฤษฎีคิวไอที

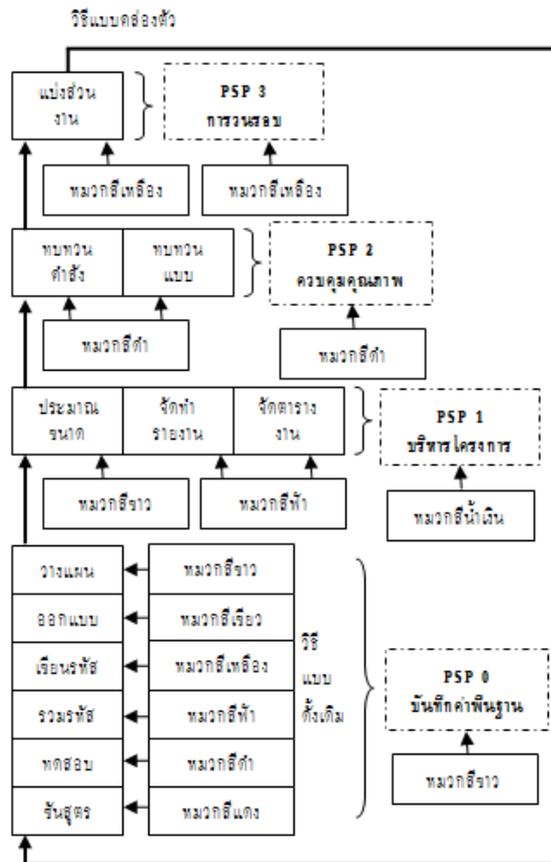
6. การเพิ่มคุณภาพการพัฒนาซอฟต์แวร์ระดับบุคคลด้วยหมวกความคิดหกใบ

การพัฒนาซอฟต์แวร์ระดับบุคคลใช้กระบวนการพัฒนาซอฟต์แวร์ในระดับบุคคลหรือพีเอสพี (Personal Software Process: PSP) เป็นแนวทางในการพัฒนาตามข้อกำหนดของสถาบันวิศวกรรมซอฟต์แวร์ที่ระบุขั้นตอนการพัฒนาไว้ 6 ขั้นตอน (PSP Level 0) ซึ่งพบว่าเป็นการใช้วิธีพัฒนาในลักษณะเดียวกับวิธีแบบดั้งเดิม คือ แบบจำลองน้ำตก และใช้การบริหารโครงการ ในการประมาณขนาดซอฟต์แวร์ จัดทำรายงาน และจัดตารางงาน (PSP Level 1) โดยระเบียบวิธีควบคุมคุณภาพ ใช้เทคนิคการทบทวนรหัสคำสั่งและการทบทวนการออกแบบ (PSP Level 2) ทั้งนี้ สามารถแบ่งส่วนการพัฒนาซอฟต์แวร์เป็นส่วนๆ เพื่อการพัฒนาเป็นวงรอบ (PSP Level 3) ซึ่งพบว่าเป็นลักษณะของการพัฒนาด้วยวิธีสมัยใหม่ คือ วิธีแบบคล่องตัว ดังนั้น การพัฒนาซอฟต์แวร์ในระดับบุคคลจึงเป็นกระบวนการที่บูรณาการวิธีการพัฒนาซอฟต์แวร์ทั้งวิธีแบบดั้งเดิมและวิธีแบบคล่องตัวเข้าด้วยกันได้อย่างเหมาะสม สอดคล้องกับพฤติกรรมการทำงานในสภาพแวดล้อมจริงของนักพัฒนาระดับบุคคลที่ไม่สามารถพัฒนาซอฟต์แวร์ตั้งแต่ต้นจนเสร็จสมบูรณ์ในคราวเดียวจากข้อกำหนดความต้องการเริ่มแรกของผู้ใช้ โดยจะต้องมีการปรับปรุงแก้ไขซอฟต์แวร์ที่พัฒนาในระหว่างทางตามความต้องการของผู้ใช้งานที่เปลี่ยนแปลงหรือเพิ่มขึ้น

เนื่องจากการพัฒนาซอฟต์แวร์ระดับบุคคลที่นักพัฒนาต้องดำเนินการในกิจกรรมและขั้นตอนที่หลากหลาย ตั้งแต่ต้นจนกระทั่งพัฒนาเป็นซอฟต์แวร์สมบูรณ์เพียงคนเดียว ทำให้เกิดความสับสนในการแบ่งแยกความคิดและการรวมสมมติ ในงานแต่ละขั้นตอนซึ่งมีความแตกต่างกัน เช่น การวัดและบันทึกค่าพื้นฐาน เป็นการใช้ความคิดในเรื่องตัวเลขและข้อมูล การบริหารโครงการ เป็นการใช้ความคิดในเรื่องที่เกี่ยวกับการควบคุม คุณภาพซอฟต์แวร์ เป็นการใช้ความคิดในเรื่องที่เกี่ยวกับความผิดพลาด และการแบ่งส่วนเพื่อวงรอบ เป็นการใช้ความคิดในเรื่องที่เกี่ยวกับประโยชน์ใช้สอยของส่วนต่างๆ ของซอฟต์แวร์ที่สามารถแยกพัฒนาที่ละส่วนได้ เป็นต้น การนำทฤษฎีหมวกความคิดหกใบมาประยุกต์จึงสอดคล้องกับกระบวนการทำงานของการพัฒนาซอฟต์แวร์ระดับบุคคล และสามารถสนับสนุนให้มีคุณภาพเพิ่มขึ้นได้ เนื่องจากช่วยลดความสับสนในบทบาทหน้าที่ของนักพัฒนาในแต่ละขั้นตอนที่แตกต่างกัน ทำให้นักพัฒนามุ่งความสนใจและใช้ความคิดในประเด็นที่กำลังดำเนินการได้ถูกต้อง โดยได้ประยุกต์ใช้ด้วยทฤษฎีคิวไอที ซึ่งสามารถสรุปแนวคิดดังภาพที่ 3 และแสดงรายละเอียดดังภาพที่ 4



ภาพที่ 3 การบูรณาการหมวกความคิดหกใบกับวิธีพัฒนาซอฟต์แวร์และกระบวนการพีเอสทีโดยใช้ทฤษฎีคิวไอที



ภาพที่ 4 รายละเอียดการบูรณาการหมวกความคิดหกใบกับขั้นตอนและกิจกรรมในกระบวนการพีเอสที

จากภาพที่ 3 สามารถอธิบายเปรียบเทียบกับการบูรณาการตามทฤษฎีคิวไอทีในภาพที่ 2 ได้ ดังนี้
 วงกลมที่ 1 ศาสตร์หลัก คือ วิธีพัฒนาซอฟต์แวร์ มี 2 วิธี คือ วิธีแบบดั้งเดิม และวิธีแบบคล้องตัว
 วงกลมที่ 2 เทคนิคคุณภาพ คือ หมวกความคิดหกใบ
 วงกลมที่ 3 เทคโนโลยีสารสนเทศ คือ กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคลหรือพีเอสที



จากภาพที่ 4 อธิบายการบูรณาการหมวดความคิดหกใบ ได้ดังนี้

ระดับ 0 (PSP 0) การบันทึกค่าพื้นฐาน ใช้การคิดแบบหมวดสี่ขา โดยนักพัฒนาคำนึงถึงข้อมูลและตัวเลขที่ต้องการวัดเป็นหลัก เช่น จำนวนข้อผิดพลาด

สำหรับกระบวนการย่อยในระดับ 0 คือ ขั้นตอนการพัฒนาซอฟต์แวร์ซึ่งใช้วิธีแบบจำลองน้ำตก มีการประยุกต์ใช้หมวดความคิดหกใบ ดังนี้

1. ขั้นการวางแผน ใช้หมวดความคิดสี่ขา เนื่องจากเป็นการวางแผนโดยนักพัฒนาจะพิจารณาข้อเท็จจริงของสถานการณ์ของหน่วยงาน ข้อมูลความต้องการของผู้ใช้ เป้าหมายการพัฒนา ตารางเวลาการทำงาน และทรัพยากรที่ใช้เป็นต้น เพื่อวางแผนการทำงาน

2. ขั้นการออกแบบ ใช้หมวดความคิดสี่ขา เนื่องจากเป็นการออกแบบสิ่งใหม่ จำเป็นต้องใช้ความคิดสร้างสรรค์เพื่อค้นหารูปแบบการทำงาน ขั้นตอนวิธี ที่เหมาะสม เพื่อให้ได้ซอฟต์แวร์ที่ทำงานได้ถูกต้อง รวดเร็ว นำใช้ มีส่วนต่อประสานที่มีคุณภาพ คงทน สามารถสร้างความพึงพอใจให้แก่ผู้ใช้ เป็นต้น

3. ขั้นการเขียนรหัส ใช้หมวดความคิดสี่เหลี่ยม โดยนักพัฒนาคำนึงถึงอรรถประโยชน์จากการทำงานของรหัสคำสั่งที่เขียนเป็นหลัก เพื่อให้ได้รหัสคำสั่งเฉพาะที่จำเป็นและถูกต้อง (Clean Code) เป็นการใช้ความคิดที่เกี่ยวกับตรรกะและเหตุผล

4. ขั้นการรวมรหัส ใช้หมวดความคิดสี่เหลี่ยม เนื่องจากเป็นการจัดการรวมรหัสคำสั่งที่เขียนขึ้นเพื่อแปลงเป็นซอฟต์แวร์ โดยนักพัฒนาต้องควบคุมการทำงานของเครื่องมือที่ใช้ในการรวมรหัสและตรวจสอบเมื่อพบข้อผิดพลาดในการทำงานของการรวมรหัสและดำเนินการแก้ไข

5. ขั้นการทดสอบ ใช้หมวดความคิดสี่เหลี่ยม เนื่องจากเป็นการใช้ความคิดที่เกี่ยวกับข้อผิดพลาด ตรรกะที่ใช้งานเหตุผลของข้อผิดพลาด โดยนักพัฒนาจะต้องมีความระมัดระวัง รอบคอบ และดูแลการทดสอบให้เป็นไปอย่างเหมาะสม

6. ขั้นการขั้นสูตร ใช้หมวดความคิดสี่เหลี่ยม เนื่องจากเป็นการประเมินซอฟต์แวร์ที่พัฒนาสำเร็จเป็นผลิตภัณฑ์แล้วและนำไปใช้งานโดยนักพัฒนาจะใช้ความคิดที่เกี่ยวกับความรู้สึก และปฏิกิริยาที่ตอบสนองต่อการใช้งานซอฟต์แวร์ อาร์มณ และประสบการณ์การใช้งาน ความรู้สึกพึงพอใจ เป็นต้น

ระดับ 1 (PSP 1) การบริหารโครงการ ใช้หมวดความคิดสี่เหลี่ยม โดยนักพัฒนาคำนึงถึงการบริหารจัดการโครงการพัฒนาซอฟต์แวร์ให้ประสบความสำเร็จ และควบคุมให้เป็นไปตามแผนงานที่วางไว้ ภายใต้ระยะเวลา งบประมาณ และทรัพยากรที่ได้รับจัดสรร

สำหรับกระบวนการย่อยในระดับ 1 คือ การประมาณขนาด การจัดทำรายงาน และการจัดตารางงาน มีการประยุกต์ใช้หมวดความคิดหกใบ ดังนี้

1. การประมาณขนาด ใช้หมวดความคิดสี่ขา เนื่องจากเป็นการใช้ความคิดที่เกี่ยวกับการคำนวณเพื่อหาขนาดของซอฟต์แวร์ โดยสามารถคำนวณจากจำนวนบรรทัดของรหัสคำสั่ง (Line of Code: LOC) หรือจากการวิเคราะห์ฟังก์ชันการทำงานของซอฟต์แวร์ (Function Point: FP)

2. การจัดทำรายงาน ใช้หมวดความคิดสี่เหลี่ยม เนื่องจากเป็นการใช้ความคิดที่เกี่ยวกับการตรวจสอบการจัดการ และการสรุปเรื่องราวทั้งหมด

3. การจัดตารางงาน ใช้หมวดความคิดสี่เหลี่ยม เนื่องจากเป็นการใช้ความคิดที่เกี่ยวกับการบริหารจัดการเวลาที่ใช้ในการพัฒนาซอฟต์แวร์ของนักพัฒนาให้เหมาะสม สอดคล้องกับแผนงาน ภาระงาน และสภาพแวดล้อมที่เป็นจริง

ระดับ 2 (PSP 2) การควบคุมคุณภาพ ใช้หมวกความคิดสี่ตา โดยนักพัฒนาคำนึงการค้นหาข้อผิดพลาดจากการดำเนินการพัฒนาซอฟต์แวร์ ข้อดีของการออกแบบ สิ่งที่เป็นต้นฉบับ เป็นต้น

สำหรับกระบวนการย่อยในระดับ 2 คือ การทบทวนรหัสคำสั่ง และการทบทวนการออกแบบ มีการประยุกต์ใช้หมวกความคิดหกใบ ดังนี้

1. การทบทวนรหัสคำสั่ง ใช้หมวกความคิดสี่ตา เนื่องจากการคิดในสิ่งที่ต้นฉบับของรหัสคำสั่งที่เขียนเพื่อการปรับปรุงให้ดีขึ้น เช่น ข้อบกพร่อง ช่องโหว่ทางด้านความปลอดภัย ทำงานซ้ำ เป็นต้น

2. การทบทวนการออกแบบ ใช้หมวกความคิดสี่ตา เช่นกัน เนื่องจากการคิดในสิ่งที่ต้นฉบับของการออกแบบ เช่น ขั้นตอนวิธี (Algorithm) ที่ไม่เหมาะสม สิ้นเปลืองทรัพยากรระบบ (System Resources) ส่วนต่อประสานผู้ใช้ (User Interface) ที่ไม่ดี เป็นต้น

ระดับ 3 (PSP 3) การวนรอบ ใช้หมวกความคิดสี่เหลี่ยม โดยนักพัฒนาใช้ความคิดทางด้านบวก คำนึงถึงประโยชน์ใช้สอย หรือหน้าที่งานส่วนที่ผู้ใช้ต้องการในการแบ่งส่วนซอฟต์แวร์ออกเป็น ส่วน ๆ และจัดลำดับความสำคัญในการพัฒนา ก่อน-หลัง เพื่อพัฒนาและส่งมอบงานให้ผู้ใช้สามารถใช้งานซอฟต์แวร์ส่วนที่ต้องการใช้งานก่อนได้โดยเร็ว และทยอยพัฒนาซอฟต์แวร์ส่วนที่เหลือเพิ่มเติมในลักษณะเป็นวงรอบของการพัฒนาจนกระทั่งได้ซอฟต์แวร์ที่สมบูรณ์แบบทั้งระบบ

สำหรับกระบวนการย่อยในระดับ 3 คือ การแบ่งส่วนงาน ใช้หมวกความคิดสี่เหลี่ยมเช่นกัน

7. วิธีการดำเนินการ

แนวคิดในการเพิ่มคุณภาพการพัฒนาซอฟต์แวร์ระดับบุคคลด้วยการบูรณาการเทคนิคหมวกความคิดหกใบตามทฤษฎีคิวไอที สามารถนำมาสร้างเป็นขั้นตอนการดำเนินการเพื่อการปฏิบัติและวัดผลได้ มีขั้นตอน ดังนี้

1. นักพัฒนาเข้ารับการอบรมหลักสูตรการพัฒนาซอฟต์แวร์ระดับบุคคล เพื่อให้มีความรู้ความเข้าใจในหลักการ แนวทาง และวิธีปฏิบัติที่ถูกต้อง

2. นักพัฒนาทำการพัฒนาโดยใช้กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล และทำการบันทึกค่าพื้นฐาน

3. นักพัฒนาเข้ารับการอบรมเทคนิคหมวกความคิดหกใบ เพื่อให้มีความรู้ความเข้าใจในหลักการ แนวคิด และวิธีปฏิบัติอย่างแท้จริง และฝึกการคิดแบบหมวกความคิดหกใบ เพื่อเสริมสร้างทักษะในการใช้งาน

4. จัดทำสัญลักษณ์แทนหมวกความคิดหกใบ โดยให้สัญลักษณ์แทนหมวกแต่ละใบมีสีของหมวกความคิดใบนั้นเป็นส่วนประกอบที่เห็นได้เด่นชัด เช่น หมวกความคิดสี่เหลี่ยมใช้สีเหลือง เป็นต้น

5. นักพัฒนาทำการพัฒนาซอฟต์แวร์โดยใช้กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคลซึ่งได้บูรณาการหมวกความคิดหกใบ และทำการบันทึกค่าพื้นฐาน

6. นำสัญลักษณ์แทนหมวกความคิดติดตั้งบนโต๊ะทำงานหรือบริเวณที่เห็นได้ชัดเจนเพื่อเตือนให้ทราบว่าเป็นการดำเนินการที่ต้องใช้ความคิดแบบใดในการทำงานขั้นตอนนั้น ๆ และเปลี่ยนสัญลักษณ์เมื่อเปลี่ยนขั้นตอน

7. นักพัฒนาเปรียบเทียบค่าพื้นฐานที่บันทึกไว้ของการพัฒนาแบบปกติกับแบบบูรณาการด้วยเทคนิคหมวกความคิดหกใบ

8. การวัดผล

การวัดผลสัมฤทธิ์ของการเพิ่มคุณภาพการพัฒนาซอฟต์แวร์ระดับบุคคลด้วยการบูรณาการเทคนิคหมวกความคิดหกใบ ทำได้ด้วยการเปรียบเทียบค่าพื้นฐานที่ทำการบันทึกไว้ก่อนและหลังการบูรณาการ โดยมีปัจจัยด้านคุณภาพที่ใช้ในการพิจารณา ได้แก่ จำนวนข้อบกพร่อง เวลาในการส่งมอบผลงาน ขนาดของซอฟต์แวร์ ประสิทธิภาพของซอฟต์แวร์



และความพึงพอใจของผู้ใช้ เป็นต้น ทั้งนี้ ผลที่ได้รับ คือ นักพัฒนาสามารถใช้งานได้เร็วขึ้น และผลงานมีข้อผิดพลาดน้อยลง เครื่องมือที่ใช้ ได้แก่ สมุดบันทึกสำหรับการเปรียบเทียบค่าพื้นฐาน โดยนักพัฒนาเป็นผู้ใช้เครื่องมือ และ แบบรายงานความผิดพลาด (Faults Report) แบบร้องขอการเปลี่ยนแปลงโปรแกรม (Request for Change: RFC) โดยผู้ใช้งานและนักพัฒนาเป็นผู้ใช้เครื่องมือร่วมกัน

สรุป

การพัฒนาซอฟต์แวร์ระดับบุคคลเป็นกระบวนการที่มีคุณภาพของผลงานขึ้นอยู่กับคุณสมบัติส่วนบุคคลของนักพัฒนาเป็นหลัก ภายหลังจากได้มีการกำหนดหลักการ แนวทางและวิธีการปฏิบัติขึ้นเป็นกระบวนการมาตรฐานเพื่อช่วยควบคุมคุณภาพของการพัฒนา เรียกว่า กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล อย่างไรก็ตามกระบวนการดังกล่าวประกอบด้วยขั้นตอนที่ใช้ความคิดในการดำเนินการที่แตกต่างกัน ทำให้นักพัฒนาเกิดความสับสน การนำวิธีคุณภาพหมวกความคิดหกใบมาใช้บูรณาการตามทฤษฎีคิวไอที่ช่วยแก้ไขปัญหาดังกล่าวและยกระดับคุณภาพของกระบวนการให้เพิ่มสูงขึ้นได้ โดยสามารถสรุปแนวทางการประยุกต์ใช้กับขั้นตอนหลักของกระบวนการได้ ดังนี้

กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล ระดับ 0 (PSP0) เป็นขั้นตอนการบันทึกค่าพื้นฐาน ประยุกต์ใช้หมวกสีเขียว ซึ่งเป็นการใช้ความคิดวิจารณ์ในการพิจารณาดำเนินการเกี่ยวกับข้อเท็จจริงต่างๆ ซึ่งมีผลการวิจัย (วิลสัน คัลลิ่งสัน, 2556; กัสมา สิทธิกุล, 2547) ระบุว่าเทคนิคหมวกความคิดหกใบช่วยเพิ่มความสามารถในการคิดอย่างมีวิจารณญาณให้สูงขึ้น

กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล ระดับ 1 (PSP1) เป็นขั้นตอนการบริหารจัดการโครงการ ประยุกต์ใช้หมวกสีฟ้า ซึ่งเป็นการใช้ความคิดเกี่ยวกับการควบคุมเพื่อให้ได้ผลลัพธ์ตามวัตถุประสงค์โครงการ โดยผลการวิจัย (ชาโรน คัลลิ่งสัน, 2550) พบว่า เทคนิคหมวกความคิดหกใบช่วยให้ผู้ใช้มีสัมฤทธิ์ผลสูงขึ้น

กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล ระดับ 2 (PSP2) เป็นขั้นตอนการควบคุมคุณภาพ ประยุกต์ใช้หมวกสีดำ ซึ่งเป็นการใช้ความคิดเกี่ยวกับข้อผิดพลาดและการแก้ไข โดย (เปียณี จิตรเจริญ, 2543) ทำการวิจัยพบว่าเทคนิคหมวกความคิดหกใบทำให้ความสามารถในการคิดแก้ปัญหาสูงขึ้น

กระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล ระดับ 3 (PSP3) เป็นขั้นตอนการตรวจสอบของการพัฒนาซอฟต์แวร์ประยุกต์ใช้หมวกสีเหลือง ซึ่งเป็นการใช้ความคิดเกี่ยวกับการวิเคราะห์หน้าที่งานส่วนที่ผู้ต้องการเพื่อแบ่งซอฟต์แวร์เป็นส่วนและจัดความสำคัญก่อนหลังในการพัฒนา โดย (นิลวรรณ เจตวรณัญญ, 2549) ทำการวิจัยพบว่าเทคนิคหมวกความคิดหกใบทำให้ความสามารถในด้านการคิดวิเคราะห์สูงขึ้น

นอกจากนี้ยังมีงานวิจัยที่ระบุว่าหมวกความคิดหกใบทำให้ความคิดสร้างสรรค์สูงขึ้น (ประยูกรัตน์ ไทยธานี, 2541) และทักษะการทำงานทางศิลปะสูงขึ้น (ปริยาภรณ์ โพธิ์บัณฑิต, 2544) สอดคล้องกับการประยุกต์ใช้หมวกสีเขียวในขั้นตอนการออกแบบของการพัฒนาซอฟต์แวร์

ข้อเสนอแนะ

1. นำวิธีการเพิ่มคุณภาพการพัฒนาซอฟต์แวร์ระดับบุคคลด้วยการบูรณาการเทคนิคหมวกความคิดหกใบตามทฤษฎีคิวไอที่ไปทำการวิจัยในสภาพแวดล้อมการทำงานจริงของนักพัฒนาเพื่อวัดและประเมินผล ซึ่งจะนำไปสู่ผลสรุปของการวิจัยต่อไป
2. นำวิธีการเพิ่มคุณภาพการพัฒนาซอฟต์แวร์ระดับบุคคลด้วยการบูรณาการเทคนิคหมวกความคิดหกใบตามทฤษฎีคิวไอที่ไปใช้งานจริง โดยปฏิบัติตามขั้นตอนในหัวข้อวิธีดำเนินการดังกล่าวมาแล้ว ทั้งนี้ มีข้อควรพิจารณาในการนำไปใช้งาน คือ ในการเปรียบเทียบค่าพื้นฐานที่บันทึกไว้ของการพัฒนาแบบปกติกับแบบบูรณาการด้วยเทคนิคหมวกความคิดหกใบ ซอฟต์แวร์ที่ใช้เพื่อการเปรียบเทียบควรมีขนาดใกล้เคียงกัน ใช้เทคนิค ภาษา และเครื่องมือชนิดเดียวกันในการพัฒนา



รายการอ้างอิง

กัสม่า ลิทธิกุล. (2547). **ผลของการฝึกคิดแบบหมวกหกใบที่มีต่อการคิดอย่างมีวิจารณญาณของนักเรียนชั้นประถมศึกษาปีที่ 2 โรงเรียนสุเหร่าบ้านดอน เขตวัฒนา กรุงเทพมหานคร.** สารนิพนธ์การศึกษามหาบัณฑิต คณะศึกษาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ.

ขนิษฐา กุลนาวิน. (2557). **ระบบต้นแบบเทคโนโลยีสารสนเทศคุณภาพประยุกต์ใช้ชีกซ์ซิกมาเพื่อทำนายผลผลิตพีซีไรต์. คุชฎีนิพนธ์ปรัชญาดุษฎีบัณฑิต (เทคโนโลยีสารสนเทศคุณภาพ) คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี.**

จารึก ชูทิตกุล. (2553). เทคโนโลยีสารสนเทศคุณภาพ : ทฤษฎี วิธีวิจัย และการนำไปใช้. **คอมพิวเตอร์และเทคโนโลยีขั้นสูง, 11, 1-15.**

ชนพร ชานพวงษ์. (2557). **สารสนเทศคุณภาพบารันซ์สเกอร์การ์ดสนับสนุนการบริหารงบประมาณขององค์การบริหารส่วนตำบลจังหวัดสมุทรสาคร. คุชฎีนิพนธ์ปรัชญาดุษฎีบัณฑิต (เทคโนโลยีสารสนเทศคุณภาพ) คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี.**

ธารณี คลังสมบัติ. (2550). **การศึกษาผลการจัดการเรียนรู้ภาษาไทยของนักเรียนชั้นประถมศึกษาปีที่ 3 โดยเทคนิคหมวกความคิด 6 ใบ ของ Edward de Bono. วิทยานิพนธ์ครุศาสตรมหาบัณฑิต คณะครุศาสตร์ มหาวิทยาลัยราชภัฏนครราชสีมา.**

นิลวรรณ เจตวรัญญู. (2549). **การเปรียบเทียบความสามารถด้านการคิดวิเคราะห์ในวิชาภาษาไทยของนักเรียนชั้นมัธยมศึกษาปีที่ 3 ที่ได้รับการสอนโดยใช้เทคนิคการคิดแบบหมวกความคิดหกใบกับการสอนปกติ. สารนิพนธ์การศึกษามหาบัณฑิต คณะศึกษาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ.**

ประยุกต์ ไทยธานี. (2541). **ผลของการใช้การตั้งคำถามตามแนวคิดหมวกความคิดหกใบที่มีต่อการคิดอย่างมีวิจารณญาณของนักศึกษาครู. คณะครุศาสตร์ศาสตร์ มหาวิทยาลัยราชภัฏนครราชสีมา.**

ปรียาภรณ์ โพธิบัณฑิต. (2544). **ผลของการสอนวาดภาพระบายสีโดยใช้เทคนิคหมวกความคิดหกใบที่มีต่อผลสัมฤทธิ์ทางการเรียนวิชาศิลปะศึกษาของนักเรียนชั้นประถมศึกษาปีที่ 5. วิทยานิพนธ์ครุศาสตรมหาบัณฑิต คณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.**

ปริศนา มัชฌิมา. (2556). **การเปรียบเทียบผลสัมฤทธิ์ทางการเรียนวิชาการปฏิบัติการฐานข้อมูล ปีการศึกษา 1/2555 ของมหาวิทยาลัยราชภัฏสวนดุสิต ที่เรียนด้วยเทคนิคหมวกความคิดหกใบและการเรียนแบบปกติ. คณะมนุษยศาสตร์และสังคมศาสตร์ มหาวิทยาลัยราชภัฏสวนดุสิต.**

ปิยานี จิตรเจริญ. (2543). **ผลของการฝึกการคิดโดยใช้เทคนิคหมวกความคิดหกใบที่มีต่อความสามารถในการคิดแก้ปัญหาเชิงสร้างสรรค์ของนักเรียนชั้นประถมศึกษาปีที่ 5. วิทยานิพนธ์ครุศาสตรมหาบัณฑิต คณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.**

วารภรณ์ อำนวยอิน. (2556). **ตัวแบบสารสนเทศคุณภาพชีกซ์ซิกมาสำหรับบริการด้วยตนเองของผู้ป่วย. คุชฎีนิพนธ์ปรัชญาดุษฎีบัณฑิต (เทคโนโลยีสารสนเทศคุณภาพ) คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี.**

วีไลลักษณ์ สายสนธ์. (2556). **ผลการสอนโดยใช้เทคนิควิธีสอนแบบหมวกความคิดหกใบในวิชาการคิดเชิงวิจารณ์ญาณและสร้างสรรค์ที่มีผลต่อความคิดสร้างสรรค์ของนักศึกษาปริญญาบัณฑิต กรณีศึกษามหาวิทยาลัยธุรกิจบัณฑิต. มหาวิทยาลัยธุรกิจบัณฑิต.**

Coupric, D. (1998). **The Personal Software Process: A Good Start for the New Software Engineer.** Retrieved December 11, 2014, from http://www.issi.uned.es/AGDS/documentos/otros/resumen_PS.htm.



- Curtin, L. (2014). **3M used Six Thinking Hats to create products for new markets**. Retrieved December 11, 2014, from <http://www.debonoforschools.com/pdfs/3M-Six-Hats-Case-Study.pdf>.
- Davis, B. (2012). **Agile Practices for Waterfall Projects: Shifting Processes for Competitive Advantage**. Florida: J. Ross.
- Edward D. B. (1999). **Six Thinking Hats**. Massachusetts: Little Brown and Company.
- Grutter, G. & Ferber, S. (2002). The Personal Software Process in Practice: Experience in Two Cases over Five Years. In J. Kontio & R. Conrai (Eds.), **Proceedings of the 7th. International Conference on Software Quality** (pp.165-174). Helsinki, Finland, June 9–13, London: Springer.
- Hormel Foods Corporation. (2010). **2010 R&D Team Winner: Hormel Foods, Our 2010 R&D Team of the Year winner is rooted in the past, ahead of its time**. Retrieved December 11, 2014, from <http://www.foodprocessing.com/articles/2010/rd-hormel/?show=all>.
- Humphrey, W. S. (2000). **The Personal Software Process (PSP)**. Pennsylvania: Carnegie Mellon University.
- Karadağ et al. (2009). Using the 'six thinking hats' model of learning in a surgical nursing class: sharing the experience and student opinions. **Australian Journal of Advanced Nursing Research and Practice**, 26(3), 59-69.
- Nir, M. (2014). **Agile Product Owner Secrets**. n.p.: Sapir Consulting.
- Paulk et al. (1993). **Capability Maturity Model for Software: Version 1.1**. Pennsylvania: Carnegie Mellon University.
- Satzinger et al. (2011). **System Analysis and Design in a Changing World**. Massachusetts: Course Technology.
- Stennes, B. (2004). **Innovation: Case by Case - How the de Bono Thinking Systems Have Transformed Companies across the Globe**. Iowa: de Bono Thinking Systems.
- Subramaniam, V. & Hunt, A. (2005). **Practices of an Agile Developer: Working in the Real World**. n.p.: Pragmatic Bookshelf.
- The Opportunity Thinker. (2014). **Six Thinking Hats: A Tool to Strengthen Critical Thinking, Collaboration, Communication, and Creativity Skills**. Retrieved December 11, 2014, from http://www.debonoforschools.com/asp/six_hats.asp.