

ความสำคัญของการทดสอบซอฟต์แวร์ทางการแพทย์ก่อนการขึ้นใช้งานจริง

The importance of software testing Medical before actual use

นันทน์ภัท จินดา^{1*} และ วิไลลักษณ์ อมรมนเทียร²

Nunnapat Jinda^{1*} and Wilailuck Amornmontien²

บทคัดย่อ

การทดสอบซอฟต์แวร์ (Software testing) เป็นกระบวนการทดลองใช้ซอฟต์แวร์อย่างมีแนวทาง โดยใช้ความรู้ทางด้านเทคนิค เพื่อให้สามารถระบุหรือค้นหาความผิดพลาด (Error) ของซอฟต์แวร์ที่อาจจะซ่อนอยู่ให้ปรากฏออกมา และสามารถระบุถึงแนวทางของการเกิดปัญหา พร้อมสมมติฐานของความผิดพลาดที่อาจจะเกิดขึ้นได้

ซึ่งในทางด้านการพัฒนาระบบนั้น การทดสอบซอฟต์แวร์ เป็นส่วนหนึ่งในการยืนยันได้ว่าระบบที่ได้รับการออกแบบหรือจัดทำขึ้นมานั้น มีประสิทธิภาพและสามารถคาดหวังผลตอบรับที่ดีได้ หากผู้ใช้งานมีความต้องการที่จะนำซอฟต์แวร์มาเป็นตัวช่วยในการทำงาน เพื่อลดภาระงาน ลดกำลังคน ประหยัดเวลา หรือต้องการความแม่นยำ ความถูกต้องของข้อมูล การจ้างหรือการซื้อซอฟต์แวร์ มาใช้งานนับว่าเป็นอีกหนึ่งวิธีการที่สามารถตอบสนองความต้องการของผู้ใช้งานได้ แต่การที่จะนำระบบหรือซอฟต์แวร์ มาใช้งานนั้น จะต้องวางแผนการทำงาน การวางกำลังคน การกำหนดขอบเขตของการทำงาน รวมไปถึงจุดประสงค์ที่ชัดเจนของผู้ใช้งานเอง รายละเอียดเหล่านี้จำเป็นต้องมีความละเอียดรอบคอบและมีแบบแผนการทำงานที่รัดกุม เมื่อมีการเตรียมการแล้วและมีทีมหรือผู้รับผิดชอบในส่วนต่าง ๆ แล้ว ก็เป็นการเข้าสู่กระบวนการทำงานในขั้นตอนต่าง ๆ แต่ในที่นี้จะระบุในส่วนของการทดสอบซอฟต์แวร์ โดยหลักการแล้วก่อนการนำระบบไปใช้งานจะแบ่งการทดสอบเป็นหลายส่วน แต่แต่ละการทดสอบจะมีเป้าหมายในการทดสอบต่างกัน

ในเอกสารเล่มนี้เป็นการกล่าวถึง การทดสอบในส่วนของ System Test หรือ จะเรียกว่า User Acceptance Test (UAT) กระบวนการในขั้นตอนนี้เป็นส่วนที่จะต้องให้ผู้ใช้งานมาร่วมทำการทดสอบ และตัดสินใจว่าระบบดังกล่าวจะสามารถนำไปใช้งานจริงได้หรือไม่ และข้อมูลในการทดสอบนั้นจะต้องคำนึงถึงความปลอดภัยทางด้านข้อมูลที่จะนำมาทดสอบ (ระบบบริหารความมั่นคงปลอดภัยของสารสนเทศ) เนื่องจากเอกสารฉบับนี้จะเป็นการกล่าวถึงการทดสอบซอฟต์แวร์เกี่ยวกับระบบโรงพยาบาล เพราะฉะนั้นข้อมูลที่จะนำมาทดสอบจำเป็นต้องใช้ข้อมูลผู้ป่วย รวมถึงรายละเอียดอื่นๆ ที่เกี่ยวข้องที่ต้องนำมาใช้ในการทดสอบของระบบนั้น ๆ

คำสำคัญ: การทดสอบซอฟต์แวร์; ระบบบริหารความมั่นคงปลอดภัยของสารสนเทศ; ข้อกำหนดมาตรฐาน ISO/IEC 27001:2013

¹ สำนักงานคณบดี คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล

¹ Dean's Office Faculty of Medicine Siriraj Hospital Mahidol University

* Corresponding Author : e-mail : nunnapat.jin@mahidol.edu

Abstract

Software testing is the process of using software with a guideline. It is using technical knowledge In order to be able to identify or search for software errors that may be hidden to appear and able to specify the path of the problem Along with the assumption of errors that may occur. Come to be a helper. Which in the area of system development, software testing as part of confirming that System designed or Created. Effective. And. able to expect good feedback. If users need to bring the software to work.

To reduce the workload reduce Manpower Save time or need precision data accuracy Hiring or purchasing software to use is another method that can meet the needs. If user but the will bring the system or Software to use must plan the work manpower boundary the work including purpose it clear of the user himself these details must have prudence and have a work plan that concise once it has been prepared and have a team or responsible person in different parts. It is a step into the work process. But here will specify in the software test section in principle before implementing the system will divide the test into many parts each experiment has different testing goals.

such as unit test, integration test, system test in this document, it is mentioned testing for system or will be called user acceptance test (UAT) the process in this step Is the part that requires users to participate in the test and decide that system will can be put into practice or not? and Test data must consider safety of Data to be tested (Information Security Management System) due to this document will be mention software testing About the hospital system therefore Data to be tested required patient information including Other details related that must bring used in testing of that system

Keywords: Software testing; Information Security Management System; ISO/IEC 27001:2013

บทนำ

การพัฒนาซอฟต์แวร์ (Software Development) มีความสำคัญและจำเป็นอย่างมากในดำเนินธุรกิจต่าง ๆ ในปัจจุบันและในอนาคต สถานประกอบการไม่ว่าจะเป็นบริษัท ห้างร้าน โรงพยาบาล โรงเรียน โรงแรม และอื่น ๆ อีกมากมาย ใช้ซอฟต์แวร์ในการบริหารงานและการดำเนินกิจกรรมต่าง ๆ ขององค์กร อาจเปรียบได้ว่าซอฟต์แวร์เป็นส่วนหนึ่งในการดำเนินชีวิตก็เป็นได้ ซึ่งซอฟต์แวร์ที่ถูกพัฒนาขึ้นมาขึ้นเพื่อมาตอบสนองความ

ต้องการในการดำเนินธุรกิจให้มีความสะดวก ง่าย และรวดเร็ว ส่งผลให้การประกอบกิจการขององค์กรเติบโต และมีความเจริญก้าวหน้าทางธุรกิจ แนวทางการพัฒนาและทดสอบซอฟต์แวร์ได้ถูกพัฒนาต่อยอดขึ้นมาทั้งในด้านแนวคิดต่าง ๆ หรือวิธีการทดสอบระบบซอฟต์แวร์และระบบที่เกี่ยวข้อง ก็ได้รับการสืบทอดหรือตกผลึกมากขึ้นจากรุ่นสู่รุ่นภายในองค์กรนั้น ๆ ดังนั้นในปัจจุบันการทดสอบซอฟต์แวร์จึงมีความละเอียดแม่นยำและมีประสิทธิภาพมากขึ้นเมื่อเทียบกับยุค

แรกๆ ไม่ว่าจะเป็นการศึกษาข้อมูลความเป็นไปได้ในการนำซอฟต์แวร์มาใช้งาน ขั้นตอนหรือเทคนิคในการทดสอบ การเก็บผลการทดสอบในส่วนที่ต้องการแก้ไขหรือปรับปรุงแล้วนำมาวางแผนแก้ไขจุดบกพร่องในตัวซอฟต์แวร์นั้นให้มีประสิทธิภาพและประโยชน์สูงสุดต่อองค์กร และในอนาคตซอฟต์แวร์ที่ได้ทดสอบไปแล้วนั้น จะสามารถทำประโยชน์ต่อองค์กร เพิ่มประสิทธิภาพและประสิทธิผลในการทำงาน อีกทั้งซอฟต์แวร์ที่ได้รับการพัฒนาสามารถที่จะรองรับเทคโนโลยีที่จะพัฒนาได้ในอนาคต

การทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริงเป็นกระบวนการที่มีความสำคัญอีกขั้นตอนหนึ่งที่จะนำซอฟต์แวร์หรือระบบนั้น ๆ ไปใช้งานภายในองค์กร ซึ่งการทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริงจะต้องอาศัยปัจจัยและความพร้อมต่าง ๆ ในการที่จะทดสอบกระบวนการทำงานให้สำเร็จลุล่วง ให้มีประสิทธิภาพ ปัจจัยต่าง ๆ นั้น ก็คือตัวซอฟต์แวร์เองว่าตรงต่อความต้องการหรือไม่ ความพร้อมและความสามารถรองรับความต้องการของผู้ใช้งาน ปัจจัยอีกอย่างหนึ่งก็คือ ความพร้อมทางด้านทรัพยากร ทั้งทางฮาร์ดแวร์ (Hardware), ฐานข้อมูล (Data Base) ความพร้อมในส่วนของผู้ใช้งานเอง และความพร้อมของทีมที่จะมาทำหน้าที่ในการทดสอบหรือเก็บผลการทดสอบ สิ่งเหล่านี้ล้วนมีผลต่อการทดสอบซอฟต์แวร์ หรือ การทำ UAT (User Acceptance Test) ทั้งสิ้น

การทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริง มีประสิทธิภาพ และสามารถนำไปใช้งานจริงได้โดยไม่ส่งผลกระทบต่อการใช้งานจริงขององค์กร ดังนั้นองค์กรควรจัดให้มีการทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริงเพื่อใช้ในการตรวจสอบการทำงานของฟังก์ชันต่าง ๆ ของซอฟต์แวร์นั้น ๆ ก่อนการขึ้นใช้งานจริงจะทำให้ซอฟต์แวร์ที่องค์กรพัฒนาขึ้นมาสามารถนำไปใช้งานได้ตรงตามวัตถุประสงค์ สอดคล้องกับกระบวนการทำงาน และเกิดประโยชน์สูงสุดต่อองค์กร ในที่นี้จะกล่าวถึงการทดสอบซอฟต์แวร์เกี่ยวกับระบบโรงพยาบาล ดังนั้นการทดสอบอาจ

หมายถึงการต้องเอาข้อมูลที่เกี่ยวข้องกับผู้ป่วยเข้ามาใช้ในการทดสอบระบบด้วย อาทิเช่น หมายเลขผู้ป่วย ชื่อ-นามสกุล และรายละเอียดอื่น ๆ ดังนั้น ในการทดสอบจำเป็นจะต้องคำนึงถึงความปลอดภัยของข้อมูลผู้ป่วยเป็นส่วนสำคัญ ทั้งยังต้องมีการปกป้องข้อมูลของผู้ป่วยที่อาจจะเชื่อมโยงไปยังตัวผู้ป่วยได้ทั้งทางตรงและทางอ้อม เพื่อให้สอดคล้องกับระบบบริหารความมั่นคงปลอดภัยของสารสนเทศ (ข้อกำหนดมาตรฐาน ISO/IEC 27001:2013) (ปริญญา เสรีพงศ์, 2562)

การทดสอบซอฟต์แวร์ คืออะไร

การทดสอบซอฟต์แวร์ เป็นกระบวนการที่จะช่วยให้ซอฟต์แวร์ที่พัฒนาขึ้นมีความถูกต้อง, ความสมบูรณ์, มีความปลอดภัย, และมีคุณภาพที่ดี ซึ่งการทดสอบซอฟต์แวร์มีแนวทางในการดำเนินการโดยใช้ความรู้ทางด้านเทคนิค เพื่อให้สามารถระบุหรือค้นหาความผิดพลาด (Error) ของซอฟต์แวร์ที่อาจจะซ่อนอยู่ให้ปรากฏออกมา และสามารถระบุถึงแนวทางของการเกิดปัญหา พร้อมสมมติฐานของความผิดพลาดที่อาจจะเกิดขึ้นได้ เพื่อให้สามารถดำเนินการประเมิน ปรับปรุงคุณภาพของซอฟต์แวร์ และทำการแก้ไขความผิดพลาดนั้น (naparat.h, 2562) ในการทดสอบควรทดสอบโดยอ้างอิงตามการใช้งานในปัจจุบันของระบบนั้น ๆ ต้องคำนึงถึงเหตุการณ์ที่เกิดขึ้นในปัจจุบันเช่น ผู้ป่วยพบแพทย์ แพทย์สั่งจ่ายยา ปรากฏว่ายาที่แพทย์สั่งจ่ายนั้นไม่สามารถเบิกได้ตามสิทธิของผู้ป่วย ทำให้ต้องมีการเอาใบยานั้นมาแก้ไข ส่วนขั้นตอนการแก้ไขนั้นก็ขึ้นอยู่กับว่ากระบวนการจ่ายยา หรือสั่งยาของแพทย์ไปถึงขั้นตอนไหน จึงจะสามารถทดสอบได้ครบถ้วนที่สุด

ขั้นตอนการทดสอบซอฟต์แวร์

ขั้นตอนการทดสอบซอฟต์แวร์ส่วนใหญ่ขึ้นอยู่กับระเบียบวิธีการทำงานของแต่ละองค์กร หรือการปฏิบัติงานของทีมงานในแต่ละแห่ง โดยขั้นตอนการ

ทดสอบซอฟต์แวร์ไม่มีรูปแบบตายตัวแน่นอน แต่สามารถระบุขั้นตอนหลักๆ ได้ 6 ขั้นตอน ดังนี้

- การวิเคราะห์ความต้องการ (วิเคราะห์ถึงกระบวนการทำงานในปัจจุบัน ปัญหาที่เกิดขึ้น และหาวิธีแก้ไขพร้อมทั้งเสนอกระบวนการทำงานที่ดีขึ้นให้ผู้ใช้งาน)
- การจัดทำแผนงานการทดสอบ (แผนการทดสอบถูกจัดทำขึ้นกระบวนการทำงานและระบบที่ถูกพัฒนาขึ้นโดยอ้างอิงจากความต้องการที่ผ่านการวิเคราะห์มาแล้ว)
- การจัดทำแนวทางการทดสอบ (กระบวนการหรือแนวทางหรือลำดับในการทดสอบ)
- การทดสอบจริง (การทดสอบโดยอิงกระบวนการทำงานจริง)
- การรายงานผลการทดสอบ (สรุปผลการทดสอบจากสถานการณ์ที่ทดสอบ)
- การทดสอบผลจากการแก้ไข (หากมีการแก้ไขต้องมีการทดสอบอีกครั้ง)

ขั้นตอนเหล่านี้มีความสำคัญในการดำรงอยู่ของซอฟต์แวร์ เนื่องจากต้นทุนในการแก้ไข ปรับปรุง มักจะสูงมาก สิ่งสำคัญสำหรับความสำเร็จของโครงการพัฒนาซอฟต์แวร์ใด ๆ คือ การระบุปัญหาได้อย่างชัดเจนและเฉพาะเจาะจง และ พิจารณาอย่างรอบคอบว่าได้ใช้เวลาในขั้นตอนเหล่านี้ได้อย่างเหมาะสม (ณัฐดนัย วังประธรรม, 2562)

แนวทางในการทดสอบ

การทดสอบซอฟต์แวร์เป็นกระบวนการที่จัดทำขึ้นเพื่อตรวจสอบการทำงานของซอฟต์แวร์ที่พัฒนาขึ้นให้มีความถูกต้อง, ความสมบูรณ์, มีความปลอดภัย, และมีคุณภาพที่ดี แนวทางในการทดสอบซอฟต์แวร์โดยมีแนวทางในการทดสอบ ดังนี้

- การทดสอบความน่าเชื่อถือ (Reliability)
- ประสิทธิภาพ (Efficiency)
- ความสามารถในการเคลื่อนย้าย (Portability)
- ความสามารถในการดูแลรักษา (Maintainability)
- ความสามารถเข้ากันได้ (Compatibility)
- สามารถใช้ได้ง่ายเข้าใจได้ง่าย (Usability)

กระบวนการทดสอบซอฟต์แวร์มีอะไรบ้าง

1. การทดสอบการทำงานของซอฟต์แวร์ในแต่ละหน่วย (Unit test) เป็นการทดสอบในระดับการเรียกใช้งานฟังก์ชัน (Function call) เพื่อเป็นการยืนยันการทำงานระดับย่อยที่สุดว่าทำงานได้ถูกต้อง เป็นการทดสอบโดยโปรแกรมเมอร์ผู้เขียนโค้ด โดยการ Unit test แต่ละระบบจะขึ้นอยู่กับการแบ่งช่วงการทดสอบ (Phase) เพื่อให้สะดวกหรือง่ายต่อการทดสอบแต่ละครั้ง เช่น การแบ่งการทดสอบเป็นฟังก์ชัน เป็นต้น บางหน่วยงานได้แบ่งทีมผู้ทดสอบ (Unit test) ออกจากโปรแกรมเมอร์ เพื่อลดภาระงานของผู้พัฒนา อีกทั้งยังให้ผู้ที่มีความสามารถเฉพาะทางเข้ามาทดสอบเพื่อให้เกิดข้อผิดพลาดน้อยและเป็นกลางที่สุด

2. การทดสอบการเชื่อมต่อส่วนย่อย ๆ (Integration test) เป็นการทดสอบซอฟต์แวร์ที่ถูกพัฒนาขึ้นว่าสามารถทำงานร่วมกันได้อย่างถูกต้อง ซึ่งจะทดสอบการทำงานร่วมกันระหว่าง Component หรือ Module ที่นำมาประกอบกัน ให้ได้ซอฟต์แวร์ที่สมบูรณ์ การทดสอบลักษณะนี้จำเป็นต้องจัดทีมที่มีความรู้ทางด้านทดสอบส่วนเชื่อมต่อโดยเฉพาะ ซึ่งอาจเป็นทีมเดียวกับทีมที่ทดสอบ Unit test

3. การทดสอบการเชื่อมต่อหรือติดต่อสื่อสารกันระหว่างซอฟต์แวร์ (System test) เป็นการทดสอบการเชื่อมต่อหรือติดต่อสื่อสารกันระหว่างซอฟต์แวร์หรือระบบอื่น ที่จริงแล้วการทดสอบการเชื่อมต่อระหว่างซอฟต์แวร์ (System test) มีความสำคัญมากในกรณีระบบที่พัฒนาขึ้นจำเป็นต้องใช้ข้อมูล หรือส่งต่อข้อมูลระหว่างระบบ เช่น มีการพัฒนาระบบการเงินขึ้นมาใหม่ ในระบบการการเงินเองจะมีส่วนของ สิทธิ ค่าใช้จ่าย ค่าอุปกรณ์ และอื่น ๆ ในส่วนที่ต้องเชื่อมต่อเพิ่มคือ ระบบที่เก็บข้อมูลผู้ป่วย ระบบที่เก็บข้อมูลบุคลากรที่เกี่ยวข้อง ระบบยา ระบบหัตถการต่าง ๆ และสิ่งที่เชื่อมต่อระหว่างระบบเหล่านี้กับระบบที่พัฒนาขึ้น คือวิธีการเชื่อมต่อ ข้อมูลที่เชื่อมต่อระหว่างกัน ระยะเวลาในการเชื่อมต่อ เป็นต้น

4. การทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริง (User Acceptance Test) เป็นการทดสอบโดยผู้ใช้ (End user) หรือโดยลูกค้า เพื่อดูว่าซอฟต์แวร์ทำงานถูกต้องตามต้องการหรือไม่ โดยการทดสอบลักษณะนี้จะละเอียดและมีความแม่นยำมากเนื่องจากผู้ใช้งานเข้ามาทดสอบด้วยตัวเอง แต่ในส่วนของระบบนั้นผู้พัฒนาหรือผู้มีส่วนร่วมในระบบจะมีความเข้าใจ ดังนั้นจึงจำเป็นต้องมีทีมที่เข้าร่วมการทดสอบ (User Acceptance Test Team) ไปพร้อมกับผู้ใช้งานเพื่อแนะนำวิธีการใช้งาน รวมถึงบันทึกข้อผิดพลาดที่เกิดขึ้นระหว่างการทดสอบ

5. การทดสอบการใช้งานซอฟต์แวร์โดยผู้เชี่ยวชาญส่วนติดต่อผู้ใช้ (Usability test) เป็นการทดสอบการใช้งานซอฟต์แวร์โดยผู้เชี่ยวชาญส่วนติดต่อผู้ใช้ ว่าใช้งานได้ง่าย เข้าใจได้ง่าย หรือไม่ ทั้งนี้จากสถานะการที่เกิดขึ้นจริง ผู้ใช้งานอาจมีการเพิ่มความ ต้องการ หรือขอปรับในส่วนของความสะดวกในการใช้งานระบบ ผู้ที่เข้าร่วมการทดสอบ จะต้องประเมิน

สถานะการบางส่วนได้ หรือสามารถบันทึกทั้งในส่วน ของข้อผิดพลาดและ ความต้องการเพิ่มเติม เพื่อมาแจ้งให้ ทีมผู้พัฒนา และผู้ควบคุมดูแลระบบทราบ

การทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริงคืออะไร

การทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริง (User Acceptance Test: UAT) คือ กระบวนการของการทดสอบระบบโดยผู้ใช้งานก่อนใช้งานจริงเพื่อตรวจสอบว่าซอฟต์แวร์สามารถตอบสนองตามความต้องการของลูกค้า หรือผู้ใช้งาน (Requirement) ตรงตามขั้นตอนของการดำเนินการทางธุรกิจ (Business process) จริงในระดับที่ยอมรับได้ (Naznin, 2018) การยอมรับได้อาจหมายถึง ระบบสามารถทำงานได้โดยไม่มีปัญหาแก่การใช้งาน หรือทำให้งานที่ทำอยู่ล่าช้า และทดสอบที่สภาพแวดล้อมในการทำงานที่ใกล้เคียงกับระบบงานจริง (Production environment) มากที่สุด ซึ่งต้องมีการจัดทำเอกสารยืนยันยอมรับว่าระบบสามารถใช้งานได้จริง (UAT sign off) พร้อมนำไปใช้งานจริงได้ (ปริญญา ชวนะสุวรรณกุล, 2561)

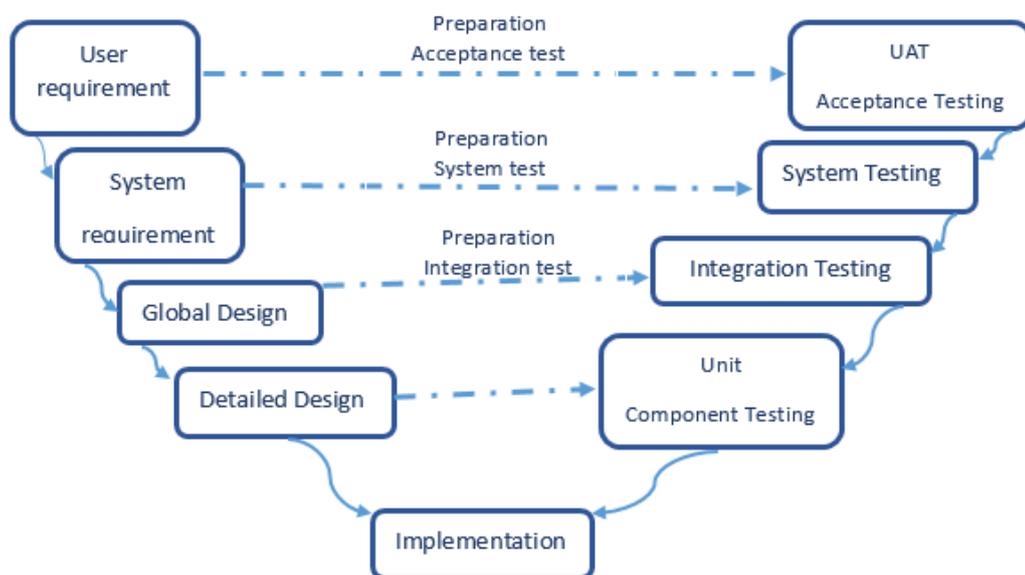
** ช่วง UAT เป็นช่วงที่ Defect จะเยอะ สาเหตุหลักเกิดจากความเข้าใจของ User, BA, SA และ DEV มีความเข้าใจที่ไม่ตรงกัน ต้องได้รับการประเมิน Defect หรือ ปัญหาที่เกิดขึ้นจะทำให้แก้ปัญหาได้ตรงจุด

ขั้นตอนการทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริง

ก่อนเริ่มขั้นตอนการทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริงนั้น ซอฟต์แวร์ที่ถูกพัฒนาขึ้นต้องผ่านการดำเนินการทดสอบขั้นตอนอื่น ๆ ก่อน เช่น Unit Testing, Integration Testing, System Testing และ Regression Testing และกระบวนการดังกล่าวต้องดำเนินการเสร็จสิ้นเรียบร้อยแล้ว จึงจะสามารถดำเนินการทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อน

ใช้งานจริงได้ ซึ่งจุดประสงค์หลักของการทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริงนั้น คือ การ

ทดสอบฟังก์ชัน (Function) การใช้งานจริง ในมุมมองของผู้ใช้งาน



รูปที่ 1 แสดงขั้นตอนของการทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์โดยผู้ใช้งานก่อนใช้งานจริง ประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

1. ทำแผนทดสอบระบบ UAT (User Acceptance Testing Planning) โดยจะต้องมีการวางแผนการทดสอบว่าจะมีวิธีการทดสอบอย่างไรให้สอดคล้องกับการใช้งานจริง เป็นขั้นตอนและครอบคลุมการทำงานทั้งหมด รวมถึงระบบที่ต้องการเชื่อมโยง (ถ้ามี) สามารถทำเป็นรายการเพื่อตรวจสอบ (Check list) จะสามารถตรวจสอบรายละเอียดการทำงานได้ถูกต้องและครบถ้วน
2. ออกแบบหัวข้อที่ใช้ในการทดสอบ (Test Cases) สำหรับทดสอบระบบ เป็นขั้นตอนที่จะระบุลงไปในการทดสอบว่าขั้นตอนไหนผู้ใช้งานต้องดำเนินการทดสอบอย่างไร อาจจะมีการสร้างแบบฟอร์ม (Template) เพื่อง่ายต่อการทดสอบ
3. ผู้ดำเนินการทดสอบ UAT ตาม Test Cases ว่ามีใครเกี่ยวข้องต่อการทดสอบบ้าง

4. ทดสอบระบบ (UAT) การทดสอบในส่วนนี้จะต้องดำเนินการภายใต้สภาพแวดล้อมที่เหมือนหรือใกล้เคียงสภาพแวดล้อมของการใช้งานจริง ทั้งในส่วนของตัวระบบและอุปกรณ์ที่เกี่ยวข้อง เช่น หากมีการพิมพ์ ต้องมีการติดตั้งอุปกรณ์เหล่านี้เข้ามาด้วย เพื่อให้ผลการทดสอบถูกต้องและน่าเชื่อถือ รวมถึงข้อมูลที่จะนำมาทดสอบจะต้องนำมาจากการสถานะการจริงที่ผู้ใช้งานดำเนินการอยู่ในปัจจุบัน และ ในขณะที่ทำการทดสอบจะต้องมีคนควบคุมการทดสอบ เพื่อให้ระหว่างทำการทดสอบเป็นไปตามเงื่อนไขในการทดสอบ หรือเป็นไปตาม Test Cases เพราะการทดสอบที่ใช้ผู้ใช้งานทดสอบนั้น เมื่อซอฟต์แวร์ไม่ตอบโต้หรือ มีหน้าจอการใช้งานที่ต้องปรับให้ใช้งานง่าย (User friendly) ผู้ใช้งานก็จะมีการปรึกษาเรื่องการออกแบบหน้าจอให้รองรับ เพราะฉะนั้น ผู้ควบคุมดูแลการทดสอบระบบจะต้องเป็นคนคอยควบคุม ให้เรื่องอื่น ๆ ที่อยู่นอกเหนือจากการทดสอบระบบ ให้สามารถไปตกลงกัน

หรือปรึกษากันในวาระอื่น ๆ เพื่อให้การทดสอบซอฟต์แวร์นั้น เสร็จสิ้นตามวันและเวลาตามที่ตั้งเป้าหมาย

5. จุดหรือบันทึกข้อผิดพลาด (Bug หรือ Defect) ที่พบในการทดสอบระบบ และประชุมเพื่อตรวจสอบข้อบกพร่อง (Review Defect) เพื่อสรุปสิ่งที่แก้ไข และกำหนดลำดับความสำคัญของ Bug หรือ Defect รวมถึงระยะเวลาที่ใช้ในการแก้ไข

6. ดำเนินการแก้ไขและทดสอบเพื่อยืนยันว่าระบบสามารถดำเนินการ UAT ได้

7. จัดทำเอกสารลงนามอนุมัติ (Sign off) เป็นเอกสารที่เป็นผลการทดสอบโดยผู้ที่ทำการทดสอบระบบจะต้องลงนามเพื่อยอมรับการทดสอบ โดยเอกสารฉบับนี้จะถูกจัดทำโดยทีมหรือ ผู้ได้รับมอบหมายรวมไปถึงคนคุมการทดสอบซึ่งภายในเอกสารหากมีความลับขององค์กร หรือมีข้อมูลที่เป็นความลับ จำเป็นจะต้องปกปิดข้อมูลดังกล่าว เช่น ข้อมูลผู้ป่วย ที่สามารถยืนยันหรือระบุตัวบุคคลนั้นได้

UAT Test Case ควรทำขึ้นมาจากอะไร

หัวข้อหรือกรณีที่ใช้ในการทดสอบระบบ (UAT Test Case) คือ ข้อมูลหรือเงื่อนไขที่ใช้ในการทดสอบ ทั้งนี้อาจรวมถึงข้อมูลตั้งต้น เหตุการณ์สมมติที่คาดว่าจะเกิดขึ้นจริงในการปฏิบัติงาน โดยเอาหน้าจอกการทำงานมาประยุกต์เข้าด้วยกัน ส่วนความละเอียดของข้อมูลที่ใช้ในการทดสอบอาจจะเป็นการจำลองสถานการณ์ทั้งในส่วนของเหตุการณ์ในการทำงานที่ปกติ คือ การทำงานที่สิ้นไหลตั้งแต่ต้นจนจบ และอาจจะจำลองการทำงานที่อาจจะปัญหาเกิดขึ้นในการ

ทำงานในปัจจุบัน ทั้งนี้ก็เพื่อผลการทดสอบที่สมบูรณ์และมีประสิทธิภาพ โดยคนที่จะทำ UAT Test Case จะเป็นผู้ใช้งาน เพราะผู้ใช้งานจะรู้การทำงานที่เป็นอยู่ได้เป็นอย่างดี สามารถมองเห็นภาพการทำงานได้ครอบคลุม หรืออาจจะเป็นการร่วมกันออกแบบระหว่างผู้ใช้งานกับผู้พัฒนาก็ได้ (Peham, 2018)

การเขียน UAT Test Case การเขียนนั้นไม่ได้มีลักษณะหรือรูปแบบตายตัว ขึ้นอยู่กับการใช้งานและลักษณะของหน้าจอกการทำงานของระบบ แต่สิ่งที่ต้องพึงระวังในการเขียน กรณีที่ใช้ในการทดสอบระบบ จะต้องเขียนการทำงานที่ครอบคลุมเริ่มตั้งแต่ต้นจนจบระบบเงื่อนไขให้ครบถ้วน รวมไปถึงระบบอื่น ๆ ที่มีการเชื่อมโยงเกี่ยวข้อง เช่น การทดสอบระบบการเงินผู้ป่วยนอกของโรงพยาบาลศิริราช มีระบบที่เชื่อมโยงกับระบบการเงิน ซึ่งกว่าข้อมูลหรือผู้ป่วยจะมาถึงการเงินจะต้องมีการ เปิด Visit ผู้ป่วยที่เวชระเบียน (MRS) และมีการส่งคำสั่งการรักษา (Order) สั่งหัตถการต่าง ๆ รวมไปถึงการส่งยา (Medical) โดยเงื่อนไขเหล่านี้จะต้องถูกระบุไปใน Test Case ด้วยเพื่อให้ทราบที่มาที่ไปของการไหลของข้อมูลที่ชัดเจน และขั้นตอนการแก้ไข หรือยกเลิก การรับชำระเงินตามเงื่อนไขต่าง ตามสิทธิการรักษาของผู้ป่วย ข้อมูลที่ใช้จึงจำเป็นต้องใช้ข้อมูลผู้ป่วยจริง เนื่องจากการทดสอบระบบเป็นระบบที่ใช้ทางการแพทย์ ดังนั้นข้อมูลจึงต้องต่อเนื่อง สามารถยืนยันสิทธิการรักษาได้ อีกทั้งข้อมูลอ้างอิงที่เกี่ยวข้องต้องเป็นของจริงทั้งหมด เพื่อที่จะสามารถตรวจสอบความถูกต้องได้อย่างครบถ้วน เนื่องจากระบบของโรงพยาบาลนั้นมีความเชื่อมโยงกันอย่างละเอียดซับซ้อน

1. คู่มือตาราง ONCO โดยใช้สิทธิ์เบิกจ่ายตรงกรมบัญชีกลาง (03) แพทย์สั่งจ่ายยา ONCO PA เบิกได้ตามสิทธิ์ ส่วนเกินสิทธิ์จ่ายเงินสด					
ลำดับ	หน่วยงาน	Scenario Brief	Expected Outcome	Pass	Fail Detail
1	เวชชเนื้องอก	ผู้ป่วย HN XXXXXXX สิทธิเบิกจ่ายตรงกรมบัญชีกลาง (16) เบิก Visit clinic xxxxx	Visit เลขที่.....		
2	แพทย์ตรวจแพทย์	สั่งยา -GLIVEC *** จำนวน 20 (GLITV400) -MABTHERA จำนวน 1 (MABH)			
3	พียงยา	คิดราคายาทั้งหมด -GLIVEC *** จำนวน 20 (GLITV400) (86840/81720/5120) -MABTHERA จำนวน 1 (MABH) (18463/17255/1214) สิทธิ 98975 บาท ส่วนลดค่ายา 6334 บาท	เลขที่จ่ายยา.....		
4	ผู้ป่วย	ระบบHN ที่หน้าจอเครื่องรับชำระเงินสดเงินฟรี			
5	ผู้ป่วย	ระบบประเภทการจ่าย "จ่ายเงินสด" และใส่ธนบัตรให้เกินกว่าจำนวนเงินที่จะต้องจ่าย ดูจำนวนเงินที่ทอน	จำนวนเงินที่ทอน ใบเสร็จรับเงินอ้างอิง (running_no เป็นของ S001)		
6	การเงิน	พิมพ์ใบแจ้งค่ารักษาพยาบาล , ใบเสร็จรับเงิน, ใบยกเว้นฯ ฉบับเต็ม ตรวจสอบรายละเอียด			
หน้าจอบันทึก	Config	พิมพ์ใบแจ้งค่ารักษาพยาบาล	พิมพ์ใบเสร็จรับเงิน	พิมพ์ใบยกเว้น หรือลดหย่อน	ใบแสดงการขอ คืนเงินค่า รักษาพยาบาล

รูปที่ 2 แสดงตัวอย่างเอกสาร User acceptance test (ระบบการเงินผู้ป่วยนอก)

การทดสอบระบบ (UAT) คืออะไร

การทดสอบระบบ (User Acceptance Test) เป็นกระบวนการที่นับว่ามีความสำคัญต่อการพัฒนาระบบ กระบวนการนี้เป็นการใช้ทดสอบการใช้งานของระบบ เพื่อให้มั่นใจว่าระบบ (Software) สามารถนำไปใช้งานได้จริง ทั้งนี้การทดสอบนั้นจะต้องเป็นไปตามเงื่อนไขที่ได้ระบุไว้ เช่น ความสมบูรณ์ของระบบ ตรงตามความต้องการของผู้ใช้งาน โดยเงื่อนไขเหล่านี้จะต้องเป็นไปตามข้อตกลงที่ได้ระบุไว้โดยผู้ที่เกี่ยวข้อง เพื่อให้การทดสอบเป็นไปตามเป้าหมายที่วางไว้ (Software Requirements) (ปริญญา ชวนะสุวรรณกุล, 2552)

โดยการทดสอบระบบนั้นจะเป็นการทดสอบโดยผู้ใช้งานจริง เพื่อให้มั่นใจว่าระบบนั้นสามารถใช้งานได้จริง ซึ่งจะแตกต่างกับการทดสอบในขั้นตอนอื่นที่อาจจะถูกทดสอบโดยผู้เขียนระบบหรือผู้ที่ทำหน้าที่ทดสอบระบบ (Tester) ทั้งนี้หากการทดสอบ ผู้ใช้งานได้ตัดสินใจแล้วว่าสามารถนำไปใช้งานได้ ก็จะเป็นขั้นตอนการทำเอกสารเพื่อลงนามอนุมัติจากผู้ใช้งาน (UAT Sign Off) (Rice, 2019) เอกสารฉบับนี้จะต้องได้รับการตรวจสอบจากผู้ตรวจสอบก่อนนำเอกสารไป

ให้ผู้ใช้งานเซ็นยอมรับ จะต้องได้รับการปกปิดข้อมูลผู้ป่วย HN, ชื่อ-นามสกุล, เลขบัตรประชาชน หรือรายละเอียดต่าง ๆ ที่ใช้ในทดสอบระบบ (UAT) แต่ถ้าหากระบบยังไม่สมบูรณ์หรือมีส่วนที่ต้องปรับปรุงและแก้ไข ก็จะไปปรับปรุงและเข้าสู่กระบวนการทดสอบระบบ (UAT) อีกครั้ง

การทดสอบระบบ (User Acceptance Test) จำเป็นจะต้องได้รับการทดสอบในสภาพแวดล้อมที่เหมือนจริงหรือใกล้เคียงของการทำงานในด้านต่าง ๆ ทั้ง Hardware และ Software รวมทั้งข้อมูลที่ใช้ทดสอบจะต้องเป็นข้อมูลจริง โดยเอาข้อมูลจริงมาลงในระบบทดสอบ (Server UAT) เพื่อให้ผลการทดสอบน่าเชื่อถือ (Radic, 2018)

ความสำคัญของ UAT

การทดสอบระบบ (UAT) เป็นสิ่งสำคัญลำดับต้น ๆ ในการที่จะนำระบบ (Software) ที่ได้รับการพัฒนาขึ้นมาใช้งาน เนื่องจากการทดสอบจะช่วยตรวจสอบการทำงานของระบบนั้นว่าทำงานได้อย่างถูกต้อง ครบถ้วน เป็นไปตามความต้องการของผู้ใช้งาน

(User Requirements) หรือไม่ (ปริญา ชวนะสุวรรณกุล, 2561) และสามารถที่จะตรวจสอบความผิดปกติที่เกิดขึ้นจากตัวระบบ (Defect) หรืออุปกรณ์ที่เกี่ยวข้องในการใช้งาน กระบวนการทดสอบซอฟต์แวร์ (UAT) จึงเป็นส่วนที่สำคัญที่จะช่วยสร้างความเชื่อมั่นให้กับผู้ใช้งานได้ และสามารถลดความผิดพลาดที่จะเกิดขึ้นกับระบบหลังจากขึ้นใช้งานไปแล้ว

การทดสอบระบบนั้นไม่ได้มีแค่ขั้นตอนการทดสอบระบบ (UAT) เพียงขั้นตอนเดียว ดังที่กล่าวมาข้างต้นนั้น การ UAT เป็นการทดสอบโดยผู้ใช้งานจริง โดยจำลองสถานการณ์จริง ข้อมูลจริง ขั้นตอนการดำเนินงานจริง เพื่อให้ประสิทธิภาพของการทดสอบนั้นสูงสุด แต่ก่อนที่จะมาถึงการทดสอบโดยผู้ใช้งานนั้น จะต้องผ่านการทดสอบต่าง ๆ โดยผู้ที่เกี่ยวข้อง Unit Test, Integration Test โดยการทดสอบแต่ละแบบมีความแตกต่างและความสำคัญต่างกัน แต่การทดสอบที่จะเป็นการยืนยันถึงความพร้อมในการใช้งานคือการทดสอบโดยผู้ใช้งาน

ปัจจัยที่ส่งผลต่อความสำเร็จในการทำ UAT

ปัจจัยที่มีผลต่อความสำเร็จในการทำการทดสอบซอฟต์แวร์(UAT) นั้นมีอยู่หลายปัจจัยด้วยกัน หรืออาจจะแบ่งเป็น 2 ส่วนได้ดังนี้

ปัจจัยภายนอก เช่น ปัญหาที่อาจจะเกิดขึ้นจากสิ่งแวดล้อมภายนอกเช่น

1. ผู้ใช้งานระบบยังไม่มีความพร้อมในการทดสอบระบบ
2. ผู้ใช้งานระบบยังไม่มีความรู้หรือความชำนาญในส่วนงานนั้น หรือยังไม่เข้าใจต่อระบบที่จะทดสอบ
3. ความพกร่องที่เกิดจากอุปกรณ์ (Hardware /Network) หรืออุปกรณ์ต่อพ่วงต่าง ๆ
4. สถานที่ ที่ใช้เก็บข้อมูล หรือ Server จำเป็นจะต้องคำนวณปริมาณการใช้ข้อมูล หรือ Growth rate (อัตราการเจริญเติบโตของข้อมูล) เพื่อให้ Server มีพื้นที่เพียงพอต่อการใช้งาน

ปัจจัยภายใน เช่น ปัญหาที่เกิดตัวซอฟต์แวร์

1. ซอฟต์แวร์ที่นำมาทดสอบนั้นอาจจะยังไม่พร้อมที่จะทดสอบ ทำให้เกิดปัญหา (Bug, Defect)
2. ข้อมูลที่ใช้ในการทดสอบ ทั้งนี้อาจรวมถึงข้อมูลตั้งต้น เหตุการณ์สมมติ หรือความละเอียดของข้อมูลที่ใช้ในการทดสอบ (Scenario) และ ปริมาณของข้อมูล

อุปสรรคของการทำ UAT

หากจะยกตัวอย่างของอุปสรรคในการทำ UAT นั้นก็คงจะมีหลายปัจจัย เช่น

1. ผู้ใช้งานไม่พร้อมที่จะดำเนินการทดสอบ อาจจะเป็นปัจจัยจากเวลา หรือ ความพร้อมทางด้านข้อมูลของผู้ใช้งานเอง
2. ความพร้อมทางด้านตัวระบบ หรือ Software อาจจะทำให้เกิดจากระบบ มี Issue หรือ Error ทำให้ระบบไม่สามารถทดสอบตามแผนได้
3. ความพร้อมจากการทดสอบก่อนการ UAT คือขั้นตอนการทดสอบย่อย (Unit Test) หากการทดสอบย่อยยังไม่ผ่านก็จะไม่สามารถทดสอบในขั้นตอนต่อไปได้
4. ความพร้อมทางด้าน Hardware เช่น เครื่องพิมพ์ (Printer) มีความขัดข้อง อุปกรณ์ต่อพ่วงเสีย หรือยังไม่พร้อมใช้งาน
5. ความต้องการ (requirement) ของผู้ใช้งาน (User) ที่เพิ่มขึ้นหรือเปลี่ยนแปลง โดยการเปลี่ยนแปลงนี้จะทำให้ขั้นตอนการทำงานล่าช้ากว่าที่วางแผนไว้ แต่สามารถปรับการทำงานได้โดยการ เพิ่ม Test case ได้ในขณะที่กำลังทำการทดสอบ หรือจะเรียกได้ว่าทดสอบไปด้วยและสร้าง Test case ไปด้วย ทั้งนี้ขึ้นอยู่กับการบริหารจัดการของผู้ทดสอบ

ความท้าทายของการทำ UAT

การทดสอบระบบ(UAT) ย่อมมีความท้าทายที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับหลายปัจจัยด้วยกัน เช่น ซอฟต์แวร์ที่ทดสอบนั้น เป็นซอฟต์แวร์แบบไหน เป็น

ซอฟต์แวร์ระบบ (System Software) และซอฟต์แวร์ประยุกต์ (Application Software) และขึ้นอยู่กับทรัพยากรที่มีการใช้ในระบบ เช่น เครื่องพิมพ์ (Printer), เครื่องอ่านบาร์โค้ด (Barcode Reader), เครื่องรูดบัตรเครดิต เป็นต้น

หากถามถึงความเสียหายที่เกิดขึ้นจากตัวผู้เขียนเองแล้ว ตัวของผู้ใช้งานน่าจะเป็นสิ่งที่ทำมากกว่าสิ่งอื่นที่เกิดขึ้นในการทดสอบระบบ เพราะเป็นปัจจัยที่ควบคุมได้ยากมาก ทั้งเวลา ความต้องการความเข้าใจ ซึ่งอาจจะหมายถึงการให้ความร่วมมือในการทดสอบระบบด้วย แต่เหนือสิ่งอื่นใด การกำหนดขอบเขตและความต้องการให้ชัดเจน หรืออาจรวมไปถึงการสร้างตารางการปฏิบัติงานขึ้นมาเพื่อกำหนดระยะเวลาการดำเนินงานที่ชัดเจน กำหนดปัจจัยและสิ่งต่าง ๆ ที่เกี่ยวข้องในการทดสอบซอฟต์แวร์ (UAT) จะสามารถช่วยให้การปฏิบัติงานเป็นไปด้วยความราบรื่นหรือไม่ และสามารถที่จะตรวจสอบความผิดปกติที่เกิดขึ้นจากตัวระบบ หรืออุปกรณ์ที่เกี่ยวข้องในการใช้งาน กระบวนการทดสอบระบบ (UAT) จึงเป็นส่วนที่สำคัญในการที่จะนำระบบใดระบบหนึ่ง หรือซอฟต์แวร์ตัวใดขึ้นไปใช้งานจริง

ข้อดีของการทดสอบระบบโดยผู้ใช้งาน

ดังตัวอย่างที่ได้กล่าวไปข้างต้นนี้แล้วในหัวข้อของปัญหาที่จะเกิดขึ้นกับองค์กรหรือบริษัทหากไม่มีกระบวนการทดสอบซอฟต์แวร์ ในหัวข้อนี้จะยกตัวอย่างถึงเหตุผล หรือสิ่งที่เกิดขึ้นหากมีการทำ UAT มีการกำหนดทรัพยากรทั้งทาง ซอฟต์แวร์หรือฮาร์ดแวร์ มีการวางแผนอย่างเป็นระเบียบในการจัดการทดสอบระบบ รวมไปถึงเอกสารที่ใช้ในการทดสอบ จะสามารถทำให้เกิดประโยชน์ต่อองค์กรหรือบริษัทได้ เช่น

1. ทำให้เกิดปัญหาน้อยที่สุดในการขึ้นใช้งานจริง ทั้งนี้ในการทดสอบ UAT จะทำในกรณีจำลองกระบวนการทำงานจริง จึงเกิดความผิดพลาดน้อยที่สุดหรือไม่มีเลย

2. เป็นการลงทุนที่คุ้มค่า ทั้งทางด้านเวลาและงบประมาณ

3. สามารถจัดการกับทรัพยากรอย่างชาญฉลาด ไม่ก่อให้เกิดความเสียหายต่อทรัพยากร

4. สร้างความเชื่อถือที่ดีและสร้างชื่อเสียงให้กับองค์กร

ข้อเสียของการไม่ทดสอบระบบโดยผู้ใช้งาน

กรณีที่มีซอฟต์แวร์มา 1 ระบบ หากให้นักพัฒนาเขียนขึ้นมาเพื่อใช้ในองค์กร หรือจัดจ้างบริษัทภายนอกมาพัฒนาซอฟต์แวร์ให้กับองค์กรแล้ว ไม่มีการทดสอบซอฟต์แวร์ (UAT) เมื่อพัฒนาแล้วนำไปใช้ทันที ย่อมเกิดความเสียหายในด้านต่าง ๆ ได้ เช่น

1. การเสียหายต่อการจัดจ้าง จัดซื้ออุปกรณ์ต่าง ๆ ทำให้เกิดการสิ้นเปลืองทางด้านงบประมาณ

2. เสียหายในทางด้านเวลา ที่จะต้องนำซอฟต์แวร์นั้นกลับมาแก้ไขหรือพัฒนาใหม่

3. ความเสียหายต่อความเชื่อถือที่มีต่อองค์กร ไม่ว่าจะเป็นการจ้างหรือการพัฒนาขึ้นมาเอง

ดังนั้น การทดสอบซอฟต์แวร์ จึงมีความจำเป็นมาก ไม่ว่าจะเป็นซอฟต์แวร์ที่พัฒนาขึ้นมาเอง หรือซอฟต์แวร์ที่จัดจ้างให้องค์กรภายนอกมาพัฒนาให้เป็นซอฟต์แวร์ที่จัดจ้างยังต้องมีความเข้มข้นในการทดสอบ เพราะการทดสอบที่เข้มข้นจะเป็นการยืนยันถึงประสิทธิภาพ ความสมบูรณ์ของระบบ และทำให้ลดความยุ่งยากที่จะเกิดขึ้นในการดูแลรักษาระบบต่อจากบริษัทผู้พัฒนา

สรุป

การทดสอบซอฟต์แวร์ (UAT) ไม่สามารถรับรองได้ว่าจะไม่เกิดปัญหาในการใช้งานระบบ ทั้งนี้ การที่จะไม่เกิดปัญหาได้นั้นต้องขึ้นอยู่กับหลายปัจจัย ทั้งในส่วนของผู้ใช้งาน ผู้ที่สร้างระบบหรือซอฟต์แวร์ขึ้นมา แม้กระทั่งข้อมูลที่น่ามาใช้ทดสอบ ซึ่งบทความนี้สามารถใช้เป็นแนวทางในการทดสอบซอฟต์แวร์และ

ดำเนินการ การปฏิบัติงานให้มีประสิทธิภาพ ความรัดกุมในการปฏิบัติงาน กระบวนการทำ UAT จะนำมาซึ่งประสิทธิภาพ ของการใช้งาน คุณภาพของระบบ ความถูกต้องแม่นยำของข้อมูล สิ่งที่สำคัญในการทำ (UAT) คือ แผนการทดสอบ (UAT Planning) การกำหนดของเขตให้ชัดเจน และความเข้าใจของงานรวมทั้งผู้ใช้งาน ซึ่ง การทดสอบซอฟต์แวร์(UAT) สามารถลดความเสี่ยงในการที่จะเกิดปัญหาที่ระบบได้จริง และทำให้ผู้ใช้งานเองเกิดความมั่นใจในการใช้ระบบมากขึ้น

ข้อเสนอแนะ

การทำ UAT เป็นแค่ส่วนหนึ่งในระหว่างการพัฒนาหรือระบบหรือนำระบบไปใช้ ยังมีสิ่งอื่นที่ต้องคำนึงถึงอีกหลายหัวข้อ โดยกระบวนการทำ UAT นั้น เป็นกระบวนการที่ดำเนินการทดสอบก่อนการนำระบบนั้น ๆ ไปใช้ หากต้องการจะต้องการพัฒนาระบบขึ้นมาใช้งาน จำเป็นจะต้องมีส่วนประกอบอื่น ๆ เพิ่มเติม เช่น นักวิเคราะห์ระบบ (SA&BA) , คนเขียนโปรแกรม (Programmer) , ทีม Server , ทีม network , ทีม Data, ทีมรับเรื่อง (Help Desk) , ทีม Client เป็นต้น ซึ่งกว่าจะพัฒนาระบบหรือโปรแกรมมาได้แต่ละระบบ จำเป็นจะต้องมีบุคลากรหรือทีมต่าง ๆ เข้ามาเพื่อดำเนินการให้ระบบนั้น ๆ สามารถใช้งานได้เรียบร้อย และเกิดประสิทธิภาพสูงสุด นอกจากปัจจัยที่กล่าวมาข้างต้นแล้ว ปัจจัยที่ต้องคำนึงอีกอย่าง คือ ความมั่นคงปลอดภัยทางด้านข้อมูล (ข้อกำหนดมาตรฐาน ISO/IEC 27001:2013) ไม่ว่าจะเป็นระบบที่พัฒนาขึ้นมาเองหรือจัดจ้าง การปกปิดความลับทางด้านข้อมูลสำคัญมาก ซึ่งหากขาดความปลอดภัยทางด้านข้อมูล อาจจะทำให้ข้อมูลเกิดการรั่วไหลไปสู่ภายนอก ทำให้เกิดความเสียหายแก่องค์กร ทั้งในส่วนของความน่าเชื่อถือ รวมถึงคดีความ จึงทำให้ต้องมี การเซ็นสัญญาปกปิดความลับในระหว่างมีการพัฒนาระบบ

กิตติกรรมประกาศ

บทความทางวิชาการฉบับนี้สำเร็จและสมบูรณ์ได้จากการสนับสนุนจาก ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล ที่ได้เล็งเห็นความสำคัญและให้โอกาสในการจัดทำบทความทางวิชาการนี้ขึ้นมา เพื่อเป็นการพัฒนาองค์กรและพนักงานให้มีความรู้และเพิ่มประสิทธิภาพในการปฏิบัติหน้าที่ ทั้งยังให้คำแนะนำในการพัฒนาส่วนของการพัฒนาซอฟต์แวร์ โดยเล็งเห็นถึงความจำเป็นในการใช้งานจริง ผู้เขียนขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

เอกสารอ้างอิง

- ณัฐกฤตา โกลนาค. (2565, 17 กุมภาพันธ์). *การรักษาความมั่นคงปลอดภัยด้านสารสนเทศ สำหรับผู้ดูแลระบบ*.
<https://erp.mju.ac.th/acticleDetail.aspx?qid=549>.
- ณัฐดนัย วังประธรรม. (2562, 6 พฤษภาคม). *การเก็บความต้องการวิเคราะห์ปัญหา (Problem and Requirements Analysis) และแบบจำลอง WRSPM MODEL*.
<https://medium.com/@nutdnuy/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%81%E0%B9%87%E0%B8%9A%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%A7%E0%B8%B4%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B2%E0%B8%B0%E0%B8%AB%E0%B9%8C%E0%B8%9B%E0%B8%B1%E0%B8%8D%E0%B8%AB%E0%B8%B2-problem-and-requirements-analysis->

- %E0%B9%81%E0%B8%A5%E0%B8%B0-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%88%E0%B8%B3%E0%B8%A5%E0%B8%AD%E0%B8%87-wrsprm-model-c2adf4d0bf88.
- ปริญญา เสรีพงศ์.(2562, 14 ธันวาคม). ระบบเอกสารของ ISO 27001 (ISMS Documentation) ตอนที่1.
<http://www.club27001.com/2019/12/iso-27001-isms-documentation-1.html>.
- ปริญญา ชวนะสุวรรณกุล. (2552, 13 ตุลาคม). *What is UAT*. <https://welovebug.com/what-is-uat-c59f71d03c21>.
- ปริญญา ชวนะสุวรรณกุล. (2561, 10 กุมภาพันธ์). *What is User Acceptance Test*. <https://welovebug.com/what-is-user-acceptance-test%E0%B8%A1%E0%B8%B2%E0%B8%97%E0%B8%B3%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-%E0%B8%81%E0%B8%B1%E0%B8%9A-user-acceptance-test-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-5209ea43cd0a>.
- สมเกียรติ ป๋วยสูงเนิน. (2558, 10 สิงหาคม). *แนวทางการเขียน Acceptance test ที่ดี*. <http://www.somkiat.cc/better-way-to-create-acceptance-test/>.
- อุไร หัวทองไผ่. (2558). การทดสอบซอฟต์แวร์. *วารสารเกษมบัณฑิต*, 16(2), 142-153.
- naparat.h. (2562, 22 กุมภาพันธ์). *การทดสอบซอฟต์แวร์* (SoftwareTesting). <https://sysadmin.psu.ac.th/2019/02/22/softwaretesting-chapter1/>.
- Randall W. Rice. (2019, May 14). *What is User Acceptance testing*. <https://www.practitest.com/qa-learningcenter/best-practices/what-is-uat-testing/>.
- Tahmina Naznin. (2018, November 12). User Acceptance Testing [UAT]. <https://medium.com>. <https://medium.com/oceanize-geeks/user-acceptance-testing-uat-19e660c96bd6>.
- Thomas Peham. (2018, May). *Let's UAT: A practical test case illustrated on the example of Trello*. <https://usersnap.com/blog/user-acceptance-testing-example>.
- Zeljko Radic. (2018, April 29). *What Do You Need to Know about UAT testing*. <https://wpamelia.com/uat-testing/>