

การใช้เจเนกินส์เพื่อจัดการคุณภาพซอฟต์แวร์

USING JENKINS TO MANAGE QUALITY OF SOFTWARE



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

การใช้เจนกินส์เพื่อจัดการคุณภาพซอฟต์แวร์

USING JENKINS TO MANAGE QUALITY OF SOFTWARE



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิชาวิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

USING JENKINS TO MANAGE QUALITY OF SOFTWARE



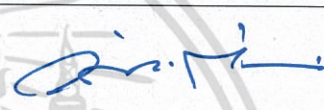
A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา การใช้เจเนกินส์เพื่อจัดการคุณภาพซอฟต์แวร์
 Using Jenkins to Manage Quality of Software
 ชื่อนักศึกษา นางสาวพิชาดา เลิศประเสริฐกิจ รหัสนักศึกษา 56050330
 ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
 ภาควิชา วิทยาการคอมพิวเตอร์
 ปีการศึกษา 2559
 อาจารย์ที่ปรึกษา ดร.ไพรัตน์ ธรเจริญศรี

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
อ.วิสันต์ ตั้งวงษ์เจริญ ประธานกรรมการ	
ดร.ไพรัตน์ ธรเจริญศรี กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การใช้เงินกินส์เพื่อจัดการคุณภาพซอฟต์แวร์
ชื่อนักศึกษา	นางสาวพิชาดา เลิศประเสริฐกิจ รหัสนักศึกษา 56050330
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	ดร.ไพรัตน์ ธรเจริญศรี

บทคัดย่อ

การใช้เงินกินส์เพื่อจัดการคุณภาพของซอฟต์แวร์ คือ กระบวนการจัดการและพัฒนาซอฟต์แวร์การแลกเปลี่ยนเงินตราระหว่างประเทศของ บริษัทรอยเตอร์ ซอฟต์แวร์ (ประเทศไทย) จำกัด (มหาชน) ให้เป็นระบบอัตโนมัติ โดยได้นำเทคโนโลยีและเครื่องมือเงินกินส์มาประยุกต์ในการควบคุมการทำงานตั้งแต่หลังจากผู้พัฒนาเขียนโปรแกรมเสร็จสิ้นตลอดจนไปถึงการส่งมอบซอฟต์แวร์ ซึ่งเครื่องมือและกระบวนการจะช่วยลดเวลาและข้อผิดพลาดต่างๆ ในการทำงาน ในการดำเนินงานระบบเดิมนั้น ผู้พัฒนาต้องการตั้งค่าและติดตามผลการดำเนินงานด้วยตัวเอง ซึ่งทำให้เกิดปัญหาในการทำงาน เช่น การคอมไพล์โปรแกรมผิดรุ่น ความล่าช้าของการทำงานเนื่องจากระบบไม่สามารถแจ้งข้อผิดพลาดของการทำงานได้ทันที ระบบดำเนินงานแบบใหม่นี้จะเรียกใช้งานผ่าน Web Browser และกระบวนการจัดการช่วยให้เกิดความสะดวกและถูกต้องในการใช้งานโดยใช้หลักการของการบูรณาการต่อเนื่อง (Continuous Integration) และเครื่องมือเงินกินส์ และภาษาที่ใช้ในการพัฒนาคือ Shell Script และ Vb Script

คำสำคัญ การบูรณาการอย่างต่อเนื่อง คอมไพล์ เงินกินส์

Title	Using Jenkins to Manage Quality of Software
Students	Miss Pichada Lerdpraserdkit Student ID 56050330
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2016
Advisor	Dr. Pairat Thorncharoensri

Abstract

Using Jenkins to manage quality of software is the process of managing and developing for International Electronic Trading software at Reuters Software (Thailand) Limited. The management uses Jenkins to manage the automatic operation, from the completion of software to delivery software, which reduces the time and errors in the work. In original process, users have to manually configure and retrieve the result. There are also some issues such as human errors because users have to manually check version of software for compilation, delay of work because the original process cannot warn the errors to user immediately. Finally, the new process had been developed to be more convenient and accurate to use, and using the principle of Continuous Integration and Jenkins tools, by writing more script for Jenkins to work with the International Electronic Trading software.

Keyword: Continuous Instigation, Compile, Jenkins

กิตติกรรมประกาศ

สหกิจเล่มนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลือและความกรุณาอย่างยิ่งจากดร.ไพรัตน์ ธรเจริญศรี อาจารย์ที่ปรึกษาสหกิจ ที่ได้ให้คำปรึกษาอย่างใกล้ชิดและเสนอแนะแนวทางการแก้ไขปัญหา รวมทั้งตรวจแก้ปัญหาพิเศษฉบับนี้ให้มีความสมบูรณ์เพิ่มขึ้น ผู้จัดทำสหกิจจึงใคร่ขอขอบพระคุณท่านเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณอาจารย์สาขาวิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ได้ให้วิชาความรู้และให้คำปรึกษาทั้งในภาคทฤษฎีและภาคปฏิบัติมาตลอด จนกระทั่งสหกิจนี้สัมฤทธิ์ผลได้ด้วยดีทุกประการ

ขอขอบคุณบริษัททอมสัน รอยเตอร์ (ประเทศไทย) ที่ให้โอกาสเข้าไปทำสหกิจศึกษา และขอขอบคุณทุกๆ ท่านในทีม Electronic Trading ที่ได้คำปรึกษาทั้งในภาคทฤษฎีและภาคปฏิบัติในการทำงานจริง ผลักดันให้ปัญหาพิเศษนี้สำเร็จลุล่วงไปด้วยดี

ท้ายที่สุดนี้ ผู้จัดทำสหกิจของกราบขอบพระคุณบิดา มารดาที่ได้ให้การสนับสนุนด้านทุนการศึกษา ให้คำปรึกษาและคอยเป็นกำลังใจที่สำคัญ ผู้จัดทำสหกิจจึงใคร่ขอขอบคุณทุกท่านเป็นอย่างสูงไว้ ณ ที่นี้

พิชาดา เลิศประเสริฐกิจ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 เครื่องมือและภาษาที่ใช้.....	2
1.5.1 ฮาร์ดแวร์.....	2
1.5.2 ซอฟต์แวร์.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 การบูรณาการอย่างต่อเนื่อง (Continuous Integration).....	3
2.1.1 ความหมายของการบูรณาการอย่างต่อเนื่อง.....	3
2.1.2 ขั้นตอนการทำงานของแนวคิด Continuous Integration.....	5
2.2 เครื่องมือเจนกินส์.....	6
2.2.1 ความหมายของเชลล์.....	6
2.2.2 ส่วนเพิ่มขยายของเจนกินส์ (Jenkins).....	6
2.2.2.1 SMC Sync Configuration Plugin.....	6
2.2.2.2 Publish Over SSH Plugin.....	7
2.2.2.3 Execute Shell Plugin.....	7
2.2.2.4 Email-ext Plugin.....	8
2.3 เชลล์ (Shell).....	9
2.3.1 ความหมายของเชลล์.....	9
2.3.2 ประเภทชุดคำสั่งของเชลล์.....	10
2.3.3 คำสั่งพื้นฐานของเชลล์.....	11
2.3.1 การเขียนเชลล์สคริปต์.....	14
2.4 ระบบการควบคุมรุ่นของซอฟต์แวร์.....	14
2.4.1 ความหมายของระบบควบคุมรุ่นของซอฟต์แวร์.....	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.2 ประเภทของระบบควบคุมรุ่นของซอฟต์แวร์	15
2.4.3 ความหมายของกิต	17
2.5 วีบีสคริปต์ (VbScript)	18
2.5.1 ความหมายของวีบีสคริปต์	18
2.5.2 ชนิดข้อมูลของวีบีสคริปต์.....	18
2.6 HP Application Life cycle Management.....	18
2.6.1 ความหมายของ HP Application Life Cycle Management	18
2.7 กระบวนการพัฒนาระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม	19
2.8 กระบวนการคอมไพล์ระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม	20
2.9 กระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม.....	22
2.10 การแสดงผลการทดสอบระบบควบคุมคุณภาพซอฟต์แวร์	23
2.11 การแสดงผล Release Notes แบบเดิม.....	23
บทที่ 3 วิธีการดำเนินงานวิจัย	26
3.1 การพัฒนาการคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ.....	30
3.1.1 ขั้นตอนการวางแผนและการเตรียมการ.....	28
3.1.2 ขั้นตอนการวิเคราะห์ระบบ	29
3.1.3 ขั้นตอนการออกแบบระบบ	29
3.1.4 ขั้นตอนการพัฒนาระบบ	31
3.2 การพัฒนาการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ	32
3.2.1 ขั้นตอนการวางแผนและการเตรียมการ.....	32
3.2.2 ขั้นตอนการวิเคราะห์ระบบ	32
3.2.3 ขั้นตอนการออกแบบระบบ	33
3.2.4 ขั้นตอนการพัฒนาระบบ	36
3.3 การปรับปรุงการแสดงผลการควบคุมคุณภาพของซอฟต์แวร์	36
3.4 การปรับปรุงการแสดงผล Release Notes ของซอฟต์แวร์	37
บทที่ 4 ผลการวิจัยและการอภิปรายผล	40
4.1 โครงสร้างของระบบ.....	40

สารบัญ (ต่อ)

	หน้า
4.2 ผลการดำเนินงาน.....	41
4.2.1 ผลการดำเนินงานการพัฒนาการคอมพิวเตอร์ระบบจำลองการ แลกเปลี่ยนเงินตราระหว่างประเทศ.....	41
4.2.2 ผลการดำเนินงานการพัฒนาการอ็อปเทรตระบบจำลองการ แลกเปลี่ยนเงินตราระหว่างประเทศ.....	42
4.2.3 ผลการดำเนินงานการปรับปรุงการแสดงผลการควบคุมคุณภาพของ ซอฟต์แวร์.....	47
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	49
5.1 สรุปผลการดำเนินงาน.....	49
5.2 สรุปผลการทดสอบ.....	49
5.3 ข้อเสนอแนะของระบบ.....	49
เอกสารอ้างอิง.....	52
ภาคผนวก.....	54
ภาคผนวก ก คู่มือการใช้งานระบบอัตโนมัติการคอมพิวเตอร์ระบบจำลองการแลกเปลี่ยนเงินตรา ระหว่างประเทศ.....	55
ภาคผนวก ข คู่มือการใช้งานระบบอัตโนมัติการอ็อปเทรตระบบจำลองการแลกเปลี่ยนเงินตรา ระหว่างประเทศ.....	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 เปรียบเทียบการใช้คำสั่งแบบมีเงื่อนไข ระหว่างบอร์นเชลล์ (ซ้าย) และซีเชลล์ (ขวา).....	10
2.2 เปรียบเทียบการตั้งค่าตัวแปรระหว่างบอร์นเชลล์ (ซ้าย) และซีเชลล์ (ขวา).....	10
2.3 แสดงสิทธิการเข้าถึงแฟ้มข้อมูล.....	12
3.1 ตารางเปรียบเทียบการจัดการการคอมไพล์.....	31
3.1 ตารางเปรียบเทียบการจัดการการอัปเดตซอฟต์แวร์	35
3.3 อธิบายยูสเคส Generated report of test (html, text file).....	36



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงการทำงานของ Continuous Integration	4
2.2 แสดงขั้นตอนการทำงานของ Continuous Integration	5
2.3 หน้าจอแสดงเครื่องมือเจ็นกินส์	6
2.4 ส่วนเพิ่มขยาย SCM Sync Configuration Plugin	6
2.5 ส่วนเพิ่มขยาย Publish Over SSH Plugin	7
2.6 ส่วนเพิ่มขยาย Execute Shell	8
2.7 อีเมลล์แสดงผลส่วนเพิ่มขยาย Email-ext Plugin	8
2.8 หน้าจอแสดงโปรแกรม Command Prompt ใน Windows	9
2.9 หน้าจอแสดงโปรแกรม Terminal ใน Unix	9
2.10 การใช้คำสั่ง ls	11
2.11 การใช้คำสั่ง cp	11
2.12 แสดงกลุ่มของการเข้าถึงแฟ้มข้อมูล	12
2.13 การใช้คำสั่ง chmod แบบตัวอักษร	13
2.14 การใช้คำสั่ง chmod แบบตัวเลขฐานแปด	13
2.15 รูปแบบการใช้คำสั่ง kill	13
2.16 รูปแบบการใช้คำสั่ง mv	14
2.17 รูปแบบการใช้คำสั่ง man	14
2.18 ระบบ Version Control แบบ local	15
2.19 ระบบควบคุมรุ่นของซอฟต์แวร์แบบรวมศูนย์.....	16
2.20 ระบบควบคุมรุ่นของซอฟต์แวร์แบบกระจายศูนย์	17
2.21 การเก็บข้อมูลแบบภาพถ่ายของกิต	18
2.22 โปรแกรม HP Quality Center (QC)	19
2.23 แผนภาพ Sequence Diagram แสดงการทำงานของระบบเดิม	20
2.24 แผนภาพ Sequence Diagram แสดงการทำงานของคอมไพเลอร์ระบบจำลอง การแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม	21
2.25 แผนภาพ Sequence Diagram แสดงการทำงานของอ็อปเทรระบบจำลอง การแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม	22
2.26 แผนภาพยูสเคสของการแสดงผลการทดสอบระบบควบคุมคุณภาพซอฟต์แวร์	23
2.27 แผนภาพ Sequence Diagram ของการทำงานของ Release Notes โดย SVN	26
3.1 โครงสร้างของระบบอัตโนมัติเพื่อควบคุมคุณภาพซอฟต์แวร์	28
3.2 แผนภาพ Sequence Diagram แสดงการทำงานของระบบใหม่	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภาพ Sequence Diagram แสดงการทำงานของสคริปต์การคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศแบบใหม่	29
3.4 แผนภาพกิจกรรมของการพัฒนาการคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศ	30
3.5 แผนภาพ Sequence Diagram แสดงการทำงานของสคริปต์การอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศแบบใหม่	33
3.6 แผนภาพกิจกรรมของสคริปต์เพิ่มเติมจากส่วนขยายของเจเนอรัลสคริปต์การพัฒนาการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศ	34
3.7 แผนภาพยูสเคสของระบบจำลองควบคุมคุณภาพซอฟต์แวร์ เมื่อมีการแก้ไขยูสเคส Generated report of test	36
3.8 แผนภาพ Sequence Diagram ของการแสดงผลลัพธ์ของระบบการควบคุมคุณภาพของซอฟต์แวร์	37
3.9 แผนภาพ Sequence Diagram ของการทำงานของ Release Notes โดย GIT.....	39
4.1 โครงสร้างของระบบอัตโนมัติเพื่อควบคุมคุณภาพซอฟต์แวร์.....	40
4.2 หน้าจอเมนูหลักในการแสดงการ Build Package ในเวอร์ชันต่างๆ.....	41
4.3 หน้าจอแสดงรายละเอียดของงานในเวอร์ชันที่เลือก.....	42
4.4 หน้าจอแสดงการตั้งค่า.....	42
4.5 หน้าจอแสดง Console ของการคอมไพล์.....	43
4.6 อีเมลแจ้งเตือนการทำงาน.....	43
4.7 หน้าจอเมนูหลักในการแสดงการอัปเดตระบบในเวอร์ชันต่างๆ.....	44
4.8 หน้าจอแสดงรายละเอียดแต่ละงานของการอัปเดตระบบ.....	44
4.9 หน้าจอแสดงการตั้งค่า.....	45
4.10 หน้าจอแสดง Console	46
4.11 อีเมลแสดงผลการทำงานของการอัปเดตระบบว่าสำเร็จ.....	46
4.12 อีเมลแสดงผลการทำงานของการอัปเดตระบบว่าไม่สำเร็จ.....	47
4.13 หน้าจอแสดงผลของการควบคุมคุณภาพของซอฟต์แวร์ ผ่านทางหน้าจอ Console ของเครื่องมือ Jenkins.....	48
4.14 ผลลัพธ์ของการควบคุมคุณภาพของซอฟต์แวร์ที่แสดงในอีเมล	49

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเรื่องเวลาและความถูกต้องถือเป็นสิ่งสำคัญในสังคมปัจจุบันเป็นอย่างมาก ยิ่งการทำงานซ้ำซ้อนทำให้เสียเวลาเป็นอย่างมาก องค์กรต่างๆ จึงมีการปรับเปลี่ยนการทำงานจากการใช้แรงงานมนุษย์เป็นการทำงานที่ควบคุมโดยคอมพิวเตอร์แทน ดังนั้นระบบอัตโนมัติจึงเป็นหลักสำคัญในการทำงาน เพราะช่วยลดเวลาในการทำงานและเพิ่มประสิทธิภาพของงาน จึงเป็นสาเหตุที่ว่าบริษัทหลายแห่งได้มีการเริ่มมีการใช้ระบบการทำงานแบบอัตโนมัติเพิ่มมากขึ้นโดยเฉพาะบริษัทที่พัฒนาซอฟต์แวร์ จึงทำให้ระบบการทำงานแบบอัตโนมัติเข้ามามีบทบาทมากยิ่งขึ้น โดยระบบอัตโนมัติคือ ระบบที่สามารถทำงานได้ด้วยตัวเองตามโปรแกรมที่ตั้งไว้ได้ เพราะสามารถเพิ่มความสะดวกรวดเร็วและลดงานที่ต้องทำซ้ำๆ ให้กับผู้พัฒนา ทั้งยังช่วยเพิ่มความถูกต้องและความเสถียรภาพของระบบงาน อีกทั้งยังช่วยลดความผิดพลาดของการทำงานของมนุษย์ได้มากขึ้นอีกด้วย

บริษัทรอยเตอร์ส ซอฟต์แวร์ (ประเทศไทย) จำกัด (มหาชน) เป็นบริษัทพัฒนาซอฟต์แวร์ทางด้านการเงินและการลงทุน สำหรับผลิตภัณฑ์ของบริษัทที่ให้ข้อมูลเกี่ยวกับอัตราแลกเปลี่ยนเงินตราระหว่างประเทศ ดังนั้นการให้บริการข้อมูลและความเร็วจึงเป็นปัจจัยหลักในการทำงาน เพราะฉะนั้นระบบการทำงานแบบอัตโนมัติจึงเป็นส่วนส่งเสริมปัจจัยดังกล่าว โดยทำให้การทำงานนั้นมีประสิทธิภาพมากขึ้น โดยระบบการทำงานแบบเดิมนั้นต้องใช้ทรัพยากรและแรงงานมนุษย์ในการทำงานเป็นส่วนมาก ทำให้เกิดการสูญเสียเวลาในการทำงานและอาจเกิดความผิดพลาดในการทำงานอีกด้วย

เนื่องด้วยการทำงานขององค์กรนั้นมีการเปลี่ยนแปลงและพัฒนาขึ้นทุกวัน ดังนั้นการทำงานโดยใช้มนุษย์ทำงานในทุกกระบวนการทำงานจึงทำให้เกิดการล่าช้า แต่ไม่ตอบสนองต่อความต้องการเปลี่ยนแปลงที่รวดเร็วของเทคโนโลยีในปัจจุบัน โดยระบบการทำงานใหม่ที่ใช้หลักการของระบบอัตโนมัติมาช่วยในการพัฒนา โดยจะตั้งค่าโปรแกรมควบคุมการทำงาน และแจ้งผลการการทำงานให้ผู้ใช้ทราบได้ทันที โดยจุดประสงค์ของการพัฒนาระบบอัตโนมัติเพื่อควบคุมคุณภาพซอฟต์แวร์นั้นจะช่วยเพิ่มความสะดวกให้กับผู้ใช้งานมากยิ่งขึ้น โดยจะลดเวลาในการทำงาน ลดความผิดพลาดที่เกิดจากผู้ใช้ได้ และสามารถติดตามขั้นตอนการทำงานได้ทุกขั้นตอน เพราะระบบจะทำการเก็บข้อมูลการทำงานของระบบเอาไว้เสมอ ทำให้สามารถตรวจสอบการทำงานเมื่อเวลาเกิดความผิดพลาดของการทำงานได้

จากปัญหาที่พบข้างต้น ผู้จัดทำได้เห็นถึงความสำคัญในการจัดการปัญหาข้างต้น จึงได้จัดทำ การพัฒนาระบบการทำงานแบบอัตโนมัติโดยใช้เครื่องมือเจเนกินส์ จากแนวคิดการบูรณาการอย่างต่อเนื่อง เพื่อลดปัญหาต่างๆ ที่กล่าวมาข้างต้น โดยได้ทำการเขียนสคริปต์เพิ่มเติมเพื่อเพิ่มการทำงานของเครื่องมือเจเนกินส์ให้สามารถครอบคลุมการทำงานทั้งหมดของระบบงานได้ เช่น การคอมไพล์ซอฟต์แวร์ การอัปเดตซอฟต์แวร์ เป็นต้น

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อพัฒนากระบวนการการทำงานและการประสานงานของทีมพัฒนาโปรแกรม
- 2) เพื่อพัฒนาการควบคุมคุณภาพของซอฟต์แวร์โดยประยุกต์ใช้กับระบบอัตโนมัติ
- 3) เพื่อลดความผิดพลาดของมนุษย์ (Human error) ในการลงมือปฏิบัติงาน
- 4) เพื่อลดขั้นตอนในการปฏิบัติงานในการควบคุมคุณภาพของซอฟต์แวร์
- 5) เพื่อให้ผู้ใช้งานกระบวนการทำงานสามารถนำกระบวนการควบคุมคุณภาพของซอฟต์แวร์ไปใช้กับซอฟต์แวร์ตัวอื่นๆ ได้

1.3 ขอบเขตของงานวิจัย

- 1) พัฒนาการควบคุมคุณภาพของซอฟต์แวร์เพื่อให้ผู้ใช้สามารถแสดงผลของการทดสอบของซอฟต์แวร์ผ่านหน้าจอของเจเนอรัลและอีเมลล์ของผู้ใช้งานได้ทันที
- 2) พัฒนาการควบคุมคุณภาพของซอฟต์แวร์เพื่อให้สามารถแสดงรายละเอียดและผลของการคอมไพล์ และอัปเดตผ่านหน้าจอของเจเนอรัลและอีเมลล์ของผู้ใช้งานได้ทันที
- 3) พัฒนาการควบคุมคุณภาพของซอฟต์แวร์เพื่อให้ผู้ใช้สามารถตั้งค่าและแก้ไขเครื่องมือเจเนอรัลให้เหมาะสมกับซอฟต์แวร์ได้
- 4) พัฒนาการควบคุมคุณภาพของซอฟต์แวร์เพื่อให้ผู้ใช้สามารถค้นหาเวอร์ชันของซอฟต์แวร์ที่ต้องการจะจัดการได้ผ่านหน้าจอเจเนอรัล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) พัฒนาการควบคุมคุณภาพของซอฟต์แวร์เพื่อช่วยลดขั้นตอนในการดำเนินงานและจัดการ ลดความผิดพลาดในการจัดการ
- 2) พัฒนาการควบคุมคุณภาพของซอฟต์แวร์เพื่อช่วยผู้ใช้สามารถทราบผลของการคอมไพล์ อัปเดต และทดสอบของซอฟต์แวร์ได้ทันที

1.5 เครื่องมือและภาษาที่ใช้

1.5.1 ฮาร์ดแวร์

- คอมพิวเตอร์พกพา ยี่ห้อ DELL รุ่น U6430
 - หน่วยประมวลผลกลาง Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz. 2.69Ghz
 - หน่วยความจำหลัก(RAM) 4.0GB

1.5.2 ซอฟต์แวร์

- Jenkins เครื่องมือในหลักการบูรณาการอย่างต่อเนื่อง (Continuous Integration)
- โปรแกรม Notepad++ ใช้ในการเขียนสคริปต์
- โปรแกรม SuperPuTTY ใช้ในเชื่อมต่อจากคอมพิวเตอร์ของตัวเองไปยังเครื่องเซิร์ฟเวอร์ Remote Terminal (telnet)
- โปรแกรม Source tree ใช้สำหรับจัดการกิต (GIT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและเทคโนโลยีที่เกี่ยวข้องกับการใช้เจเนกินส์เพื่อจัดการคุณภาพของซอฟต์แวร์ โดยผู้จัดทำได้ศึกษาเทคโนโลยีที่เกี่ยวข้องกับการทำวิจัยครั้งนี้ โดยสามารถแบ่งกลุ่มได้ 4 กลุ่ม ดังนี้

- กลุ่มที่ 1 แนวคิดการบูรณาการอย่างต่อเนื่อง เป็นแนวคิดที่เกี่ยวกับการทำงานเป็นกลุ่ม อยู่ในการทำงานแบบ Agile โดยการทำงานคือการที่คนในทีมการทำงานนั้นทำการอัปเดตโค้ดของซอฟต์แวร์บ่อยๆ และต้องทำงานกันอย่างต่อเนื่องโดยจะแสดงเนื้อหาในหัวข้อที่ 2.1

- กลุ่มที่ 2 ภาษาที่ใช้ในการพัฒนาและระบบควบคุมรุ่น เนื่องจากการพัฒนานั้นมีการทำงานกับเซิร์ฟเวอร์ระบบปฏิบัติการลินุกซ์ จึงทำให้ใช้ภาษาเชลล์ ซึ่งเป็นภาษาที่ใช้ติดต่อกับเครื่องเซิร์ฟเวอร์ลินุกซ์ ในหัวข้อที่ 2.3 และในหัวข้อที่ 2.4 จะกล่าวถึงภาษาที่ใช้กับโปรแกรมทดสอบระบบ HP - Quality Center (QC) นั่นคือภาษาวีบสคริปต์ (VbScript) โดยจะกล่าวถึงในหัวข้อที่ 2.5 และในระบบควบคุมรุ่นนั้นคือ จะคอยตรวจสอบว่าในแต่ละครั้งโค้ดมีอะไรเปลี่ยนแปลงไปบ้าง แล้วเก็บความเปลี่ยนแปลงนั้นไว้ และยังสามารถกู้คืนมาได้ อธิบายในหัวข้อที่ 2.4

- กลุ่มที่ 3 เครื่องมือที่ใช้ในการพัฒนา โดยเครื่องมือที่ใช้ในการพัฒนานั้นจะแบ่งเป็น 2 เครื่องมือ คือเครื่องมือที่ใช้สนับสนุนการแนวคิดการบูรณาการอย่างต่อเนื่อง โดยจะแสดงขั้นตอนการทำงานต่างๆ และสามารถตั้งค่าการทำงานล่วงหน้าได้ อีกเครื่องมือคือเครื่องมือที่ใช้ในการทดสอบระบบ คือ HP Quality Center (QC) เป็นเครื่องมือที่ทดสอบคุณภาพของซอฟต์แวร์โดยอธิบายไว้ในหัวข้อที่ 2.6

- กลุ่มที่ 4 กระบวนการทำงานแบบเดิม โดยกระบวนการพัฒนาระบบแลกเปลี่ยนเงินตราระหว่างประเทศแบบเดิมนั้นจะทำงานตั้งแต่การคอมไพล์ การอัปเดตและการทดสอบโดยจะแสดงให้เห็นในหัวข้อที่ 2.7 และสามารถย่อยกระบวนการทำงานได้ดังนี้ กระบวนการคอมไพล์ระบบแลกเปลี่ยนเงินตราระหว่างประเทศแบบเดิม โดยเป็นกระบวนการทำงานตั้งแต่หลังจากการเขียนโค้ดจนถึงการคอมไพล์โค้ดให้ได้ซึ่งซอฟต์แวร์ออกมา สามารถดูได้ในหัวข้อที่ 2.8 และกระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราระหว่างประเทศแบบเดิมนั้นจะทำงานหลังหัวข้อที่ 2.8 เมื่อได้ซอฟต์แวร์รุ่นใหม่ก็จะทำการแทนที่ซอฟต์แวร์รุ่นเดิมอธิบายในหัวข้อที่ 2.9 และจะอธิบายถึง การแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์ โดยเป็นการเขียนสคริปต์เพิ่มเติม ในหัวข้อที่ 2.10 และในหัวข้อที่ 2.11 นั้นจะแสดงผลของการทำงานของ Release Notes แบบเดิมซึ่งทางผู้จัดทำทำการปรับปรุง

2.1 การบูรณาการอย่างต่อเนื่อง (Continuous Integration)

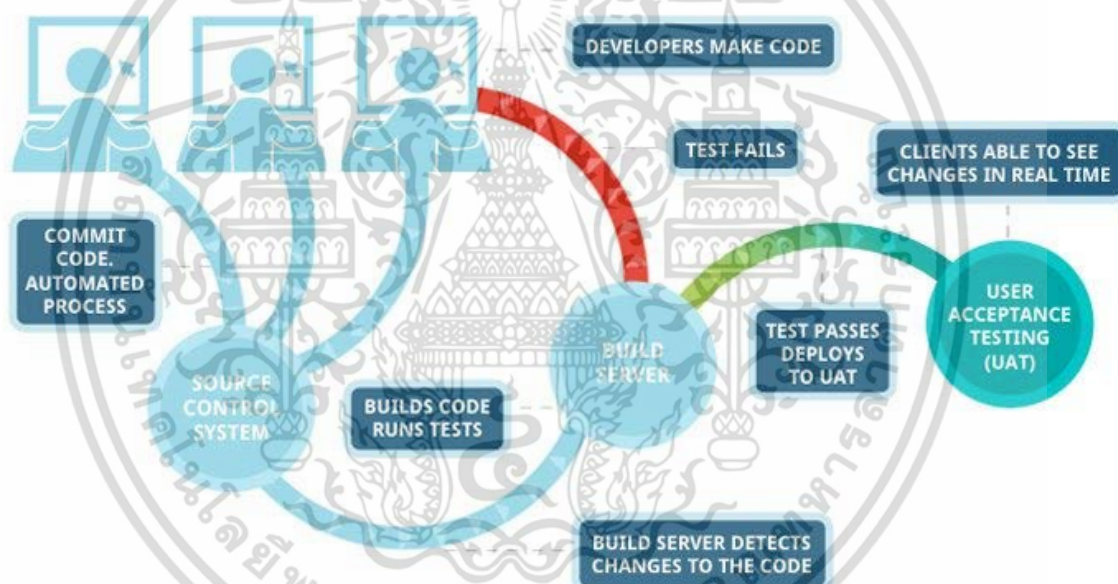
2.1.1 ความหมายของการบูรณาการอย่างต่อเนื่อง [1]

การบูรณาการอย่างต่อเนื่อง ชื่อภาษาอังกฤษ คือ Continuous Integration หมายถึง หลักการปฏิบัติของการทำงานเป็นทีม โดยเป็นส่วนหนึ่งของวิศวกรรมซอฟต์แวร์ มีการพัฒนาซอฟต์แวร์เป็นส่วนหนึ่งของเอจาย (Agile) ก่อตั้งโดย Grady Booch ในปีคริสต์ศักราชที่ 1991 ถึงแม้ว่า Booch จะไม่สนับสนุนการอินทิเกรชัน (Integration หมายถึงการ เอาโค้ดที่ผู้พัฒนา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ขึ้นด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Developer) เข้าสู่ซอร์สโค้ด (Source code) ส่วนกลางอย่างต่อเนื่อง เพื่อที่ระบบกลางจะสร้างแพ็คเกจให้ลูกค้าได้อย่างต่อเนื่อง) หลายๆ ครั้งต่อวัน การบูรณาการอย่างต่อเนื่องนั้นเป็นส่วนที่สนับสนุนเอ็กซ์ตรีมโพรแกรมมิ่ง (Extreme Programming) คือสามารถให้อินทิเกรชันได้มากกว่า 1 ครั้งต่อวัน หรืออาจจะถึง 10 ครั้งต่อวัน

เป้าหมายของ Continuous Integration นั้นคือ การป้องกันปัญหาของการส่งมอบโค้ดไปยังส่วนกลาง และยังช่วยในเรื่องการปรับปรุงคุณภาพของซอฟต์แวร์ ลดการใช้เวลาในการส่งมอบซอฟต์แวร์ให้กับลูกค้า การบูรณาการอย่างต่อเนื่องนั้นเกิดขึ้นได้เพราะได้รับแรงบันดาลใจจากหลักการเอ็กซ์ตรีมโพรแกรมมิ่ง กล่าวได้ว่า เมื่อทางทีมผู้พัฒนาได้มีการส่งมอบโค้ดไปยังส่วนกลางมากขึ้นจนทำให้ซอร์สโค้ดส่วนกลางมีการเปลี่ยนแปลงมากจากของเดิมมาก อาจจะทำให้เกิดอินทิเกรชัน เฮล (Integration hell) ได้ และอาจจะเกิดการขัดแย้งกันในการทำงานของซอฟต์แวร์ ทำให้ทางผู้พัฒนาต้องแก้ไขปัญหา และอาจจะใช้เวลานานในการแก้ไขปัญหา ดังนั้นหลังการบูรณาการอย่างต่อเนื่องนั้นมาช่วยลดการเกิดอินทิเกรชัน เฮล [2]



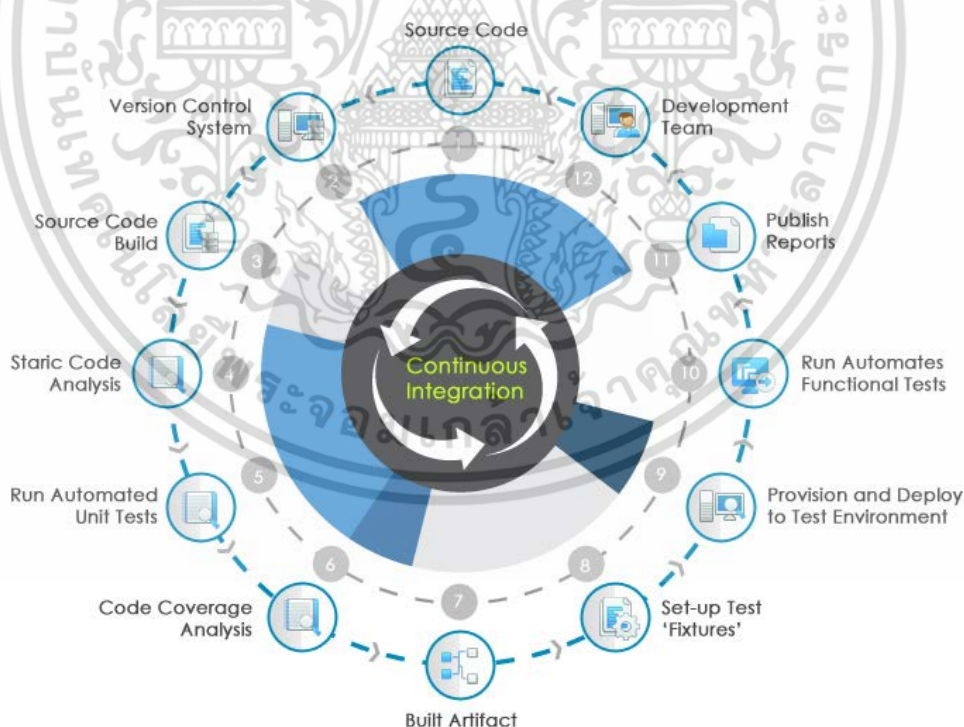
รูปที่ 2.1 แสดงการทำงานของ Continuous Integration

ในการพัฒนานั้น จะเริ่มกันตั้งแต่การอินทิเกรชัน กันเมื่อมีการเปลี่ยนแปลงซอร์สโค้ดที่ Repository กลาง ระบบจะทำการตรวจสอบโค้ด หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ ตั้งแต่การคอมไพล์ (Compile) จนไปถึงการทดสอบ (Testing) สิ่งที่สำคัญในการทำงานคือ ต้องได้ผลการทำงาน (Feedback) ที่รวดเร็วในการอินทิเกรชัน เพราะจะทำให้มองเห็นปัญหาที่เกิดขึ้นของระบบ ทำให้ช่วยลดเวลาในการค้นหาปัญหาและที่สำคัญช่วยลดเวลาของภาพรวมของระบบการพัฒนาได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ขั้นตอนการทำงานของแนวคิด Continuous Integration [3]

1. ผู้พัฒนาแต่ละคนจะต้องสามารถ Build¹ ทดสอบ และ Run² ระบบที่เครื่องของตัวเองได้
2. เมื่อทำการพัฒนาฟีเจอร์ (Feature) เสร็จ ต้องมีการทดสอบอัตโนมัติเสมอ
3. ทำการดึงซอสโค้ดล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่
4. ถ้ามีการเปลี่ยนแปลงให้ทำการผสาน (Merge) ที่เครื่องของผู้พัฒนาก่อน จากนั้นจึง build, ทดสอบ และ run อีกรอบ
5. หากเกิดการขัดแย้งหรือปัญหาให้แก้ไขที่เครื่องของผู้พัฒนาก่อนเสมอ
6. เมื่อทำการทดสอบทุกอย่างผ่านหมดในเครื่องของผู้พัฒนา ให้ส่งการเปลี่ยนแปลงไปที่ Repository กลาง
7. เมื่อซอสโค้ด (Repository) กลาง มีการเปลี่ยนแปลง ระบบจะทำการ Build, ทดสอบ และ run อีกรอบ
8. ถ้าหากเกิดความผิดพลาด ระบบจะส่งการแจ้งเตือนไปยังทีมที่พัฒนาทันที เพื่อให้เกิดการแก้ไข
9. เมื่อระบบทำงานเรียบร้อยจะส่งผลลัพธ์ไปยังส่วนการทำงานอื่นต่อไป เช่นการทดสอบระบบ การติดตั้งระบบ เป็นต้น



รูปที่ 2.2 แสดงขั้นตอนการทำงานของ Continuous Integration [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 เครื่องมือเจกินส์ (Jenkins)

2.2.1 ความหมายของเจกินส์ [4]

เจกินส์ (Jenkins) คือ เครื่องมือของ Continuous Integration ที่เขียนโดยภาษาจาวา (JAVA) เป็นโอเพนซอร์ส (Open Source) สำหรับการทำงานโดยอัตโนมัติของการบีวเซิร์ฟเวอร์ (Build Server) ทำให้การพัฒนาการซอฟต์แวร์ทำได้เร็วมากยิ่งขึ้น เจกินส์เป็นเครื่องมือที่ใช้จัดการและควบคุมวัฏจักรการกระบวนการการพัฒนาทั้งหมด ตั้งแต่การ build สร้างเอกสาร การทดสอบ การส่งมอบแพ็คเกจ เป็นต้น



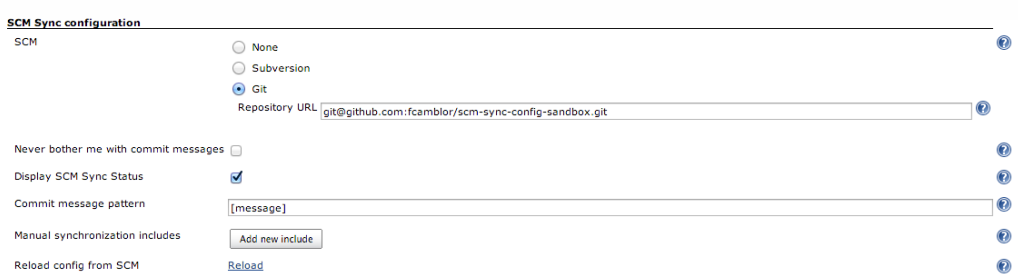
รูปที่ 2.3 หน้าต่างจอแสดงเครื่องมือเจกินส์

2.2.2 ส่วนเพิ่มขยายของเจกินส์ (Plugin)

โดยปกติแล้วเจกินส์จะมีความสามารถในการติดตั้งส่วนเพิ่มขยายเพิ่ม เพื่อช่วยในการทำงานนั้นมีการครอบคลุมระบบมากยิ่งขึ้น โดยส่วนเพิ่มขยายของเจกินส์สามารถติดตั้งได้โดยการดาวน์โหลดที่หน้าเว็บไซต์ โดยส่วนเพิ่มขยายนั้นได้แก่

2.2.2.1 SCM Sync Configuration Plugin

เป็นส่วนเพิ่มขยายที่ใช้ในการดึงเอา Source code จากระบบควบคุมรุ่นมาไว้ที่เซิร์ฟเวอร์ เพื่อรอการทำงานขั้นต่อไป



รูปที่ 2.4 ส่วนเพิ่มขยาย SCM Sync Configuration Plugin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.2 Publish Over SSH Plugin

เป็นส่วนขยายเพิ่มเติมเพื่อการเชื่อมต่อ Network Protocol ที่สามารถแลกเปลี่ยนข้อมูลโดยช่องทางที่ปลอดภัย (Secure Channel) ระหว่างอุปกรณ์เครือข่ายสองตัว ใช้ Linux หรือ Unix เป็นระบบปฏิบัติการพื้นฐานในการเข้าถึงบัญชีผู้ใช้ (Shell Accounts)

The screenshot shows the configuration interface for the Publish Over SSH Plugin. It includes a section for 'Hudson SSH Key' with input fields for 'Passphrase', 'Path to key', and a text area for the 'Key' (containing a long RSA private key). Below this is the 'SSH Servers' section with fields for 'Name' (bap@cech), 'Hostname' (cech), 'Username' (bap), and 'Remote Directory'. A 'Success' message is centered on the page. Action buttons include 'Add', 'Save', 'Advanced...', 'Test Configuration', and 'Delete'. The footer shows 'Page generated: 04-Feb-2011 15:08:34' and 'Jenkins ver. 1.396'.

รูปที่ 2.5 ส่วนเพิ่มขยาย Publish Over SSH Plugin

จากรูปที่ 2.5 อธิบายได้ว่า

- Key คือ RSA private key ซึ่งนำมาจาก ~/.ssh/
- Hostname คือ IP หรือชื่อเครื่อง SSH server
- Remote directory คือที่อยู่บน SSH Server

2.2.2.3 Execute Shell Plugin

เป็นส่วนเพิ่มขยายเพื่อดำเนินการเกี่ยวกับ Shell บนเครื่อง Linux เช่นการป้อนคำสั่งต่างๆ ของ Shell สามารถดำเนินการกับเชลล์สคริปต์ และสามารถเขียนเชลล์สคริปต์เพื่อดำเนินการต่างๆ ในระบบงานได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Build

Execute shell

```
Command # Prepare the zone files
mkdir -p "$WORKSPACE"/chroot/zones
authdns-gen-zones "$WORKSPACE"/templates "$WORKSPACE"/chroot/zones

# lint them
gdnssd -d "$WORKSPACE"
```

See [the list of available environment variables](#)

Delete

รูปที่ 2.6 ส่วนเพิ่มขยาย Execute Shell

จากรูปที่ 2.6 จะเห็นว่าสามารถป้อนคำสั่งของเชลล์ลงไปได้โดยตรงเลย และยังสามารถดำเนินการกับเชลล์สคริปต์ได้ด้วย

2.2.2.4 Email-ext Plugin

เป็นส่วนเพิ่มขยายที่ใช้สำหรับส่ง Feedback ของการทำงานของระบบกลับมาที่ผู้ใช้ โดยเป็นการส่งผ่านทางอีเมล

Message build.log (311 B)

BUILD SUCCESS

Build URL http://147.128.21.40:8080/job/yannik_test/50/
 Project: yannik_test
 Date of build: Thu, 27 Mar 2014 13:17:05 +0800
 Build duration: 62 ms

CHANGES
 No Changes

CONSOLE OUTPUT
 Started by user anonymous
 Building in workspace /var/lib/jenkins/jobs/yannik_test/workspace
 No emails were triggered.
 [workspace] \$ /bin/sh -xe /tmp/hudson1224512390593303847.sh
 + pwd
 /var/lib/jenkins/jobs/yannik_test/workspace
 + python -V
 Python 2.7.3
 Email was triggered for: Success
 Sending email for trigger: Success

รูปที่ 2.7 อีเมลแสดงผลของส่วนขยาย Email-ext Plugin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

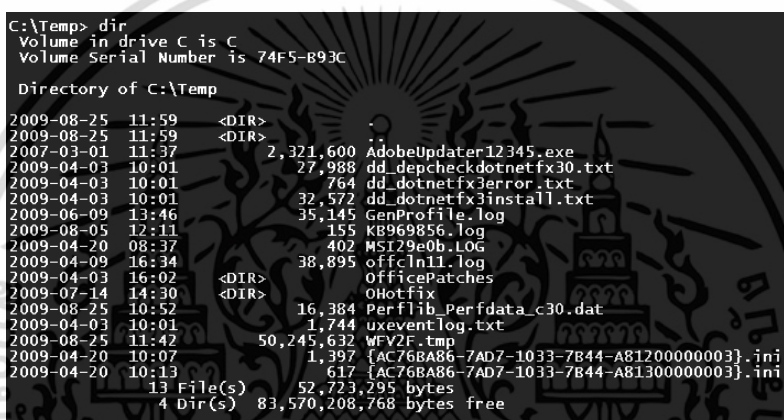
2.3 เชลล์ (Shell)

2.3.1 ความหมายของเชลล์ [5]

เชลล์ (Shell) คือ ส่วนประสานงานโต้ตอบระหว่างผู้ใช้กับระบบปฏิบัติการ (Operating System) ทำให้ที่แปลงคำสั่งจากผู้ใช้ให้ระบบปฏิบัติการเข้าใจ โปรแกรมเชลล์นั้นยังมีคุณสมบัติของภาษาของโปรแกรมอีกด้วย ดังนั้นทำให้สามารถใช้งานในรูปแบบของโปรแกรมโดยเอา รูปแบบของคำสั่งมารวมกัน เรียกว่าเชลล์สคริปต์ (Shell script) ซึ่งเชลล์สามารถแบ่งได้เป็น 2 ประเภท คือ

1. ประเภทชุดคำสั่ง (Command line interface, CLI)

เป็นส่วนประสานงานกับผู้ใช้ที่อนุญาตให้ป้อนรูปแบบคำสั่งที่เป็นตัวหนังสือ ส่งการลงไปด้วยตนเองเพื่อให้คอมพิวเตอร์ทำงานตามที่ต้องการที่ละบรรทัดคำสั่ง

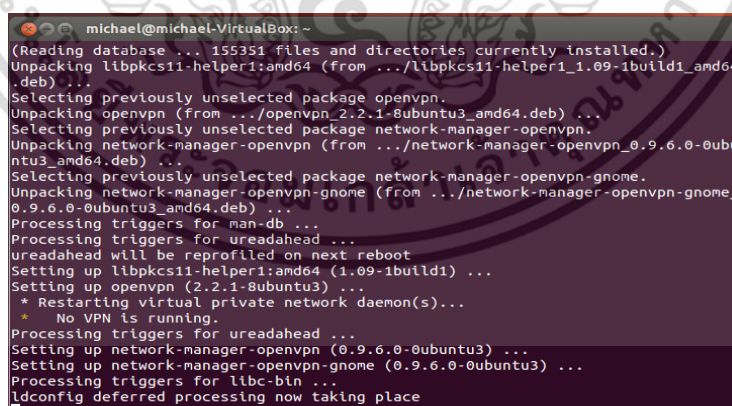


```
C:\Temp> dir
Volume in drive C is C
Volume Serial Number is 74F5-B93C

Directory of C:\Temp

2009-08-25 11:59 <DIR>          .
2009-08-25 11:59 <DIR>          ..
2007-03-01 11:37             2,321,600 AdobeUpdater12345.exe
2009-04-03 10:01             27,988 dd_depcheckdotnetfx30.txt
2009-04-03 10:01              764 dd_dotnetfx3error.txt
2009-04-03 10:01             32,572 dd_dotnetfx3install.txt
2009-06-09 13:46             35,145 GenProfile.log
2009-08-05 12:11              155 KB969856.log
2009-04-20 08:37              402 MSI29e0b.LOG
2009-04-09 16:34             38,895 officIn11.log
2009-04-03 16:02 <DIR>          OfficePatches
2009-07-14 14:30 <DIR>          OHotfix
2009-08-25 10:52             16,384 Perflib_Perfdata_c30.dat
2009-04-03 10:01              1,744 uxeventlog.txt
2009-08-25 11:42             50,245,632 WfY2F.tmp
2009-04-20 10:07              1,397 {AC76BA86-7AD7-1033-7B44-A81200000003}.ini
2009-04-20 10:13              617 {AC76BA86-7AD7-1033-7B44-A81300000003}.ini
                13 File(s)      52,723,295 bytes
                4 Dir(s)      83,570,208,768 bytes free
```

รูปที่ 2.8 หน้าจอแสดงโปรแกรม Command Prompt ใน Windows



```
michael@michael-VirtualBox: ~
(Reading database ... 155351 files and directories currently installed.)
Unpacking libpkcs11-helper1:amd64 (from .../libpkcs11-helper1_1.09-1build1_amd64.deb) ...
Selecting previously unselected package openvpn.
Unpacking openvpn (from .../openvpn_2.2.1-8ubuntu3_amd64.deb) ...
Selecting previously unselected package network-manager-openvpn.
Unpacking network-manager-openvpn (from .../network-manager-openvpn_0.9.6.0-0ubuntu3_amd64.deb) ...
Selecting previously unselected package network-manager-openvpn-gnome.
Unpacking network-manager-openvpn-gnome (from .../network-manager-openvpn-gnome_0.9.6.0-0ubuntu3_amd64.deb) ...
Processing triggers for man-db ...
Processing triggers for ureadahead ...
ureadahead will be reprofiled on next reboot
Setting up libpkcs11-helper1:amd64 (1.09-1build1) ...
Setting up openvpn (2.2.1-8ubuntu3) ...
* Restarting virtual private network daemon(s)...
* No VPN is running.
Processing triggers for ureadahead ...
Setting up network-manager-openvpn (0.9.6.0-0ubuntu3) ...
Setting up network-manager-openvpn-gnome (0.9.6.0-0ubuntu3) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
```

รูปที่ 2.9 หน้าจอแสดงโปรแกรม Terminal ใน Unix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ประเภทกราฟิก (Graphic user interface. GUI)

ระบบการติดต่อสื่อสารระหว่างผู้ใช้งานกับเครื่องคอมพิวเตอร์โดยผ่านทางภาพหรือแผนภูมิที่เข้าใจได้ง่ายแทนการพิมพ์คำสั่งโดยตรง เพื่อให้คอมพิวเตอร์ทำงานตามที่ต้องการ เช่น การใช้เมาส์กดเลือกไอคอน (Icon) หรือปุ่มคำสั่งที่ต้องการ หรือการเลือกคำสั่งตามรายการที่อยู่ในเมนูประเภทต่างๆ ซึ่งซอฟต์แวร์เตรียมไว้ให้แล้ว

2.3.2 ประเภทชุดคำสั่งของเชลล์ [6]

1. บอร์นเชลล์ (Bourne shell, sh)

เป็นโปรแกรมเชลล์มาตรฐานในการติดต่อกับผู้ใช้ผ่านทางบรรทัดคำสั่ง บนระบบปฏิบัติการยูนิกซ์ เวอร์ชัน 7 โปรแกรมนี้แทนที่ทอมสันเชลล์ (Thomson shell) ในยูนิกซ์รุ่นก่อนหน้านั้นซึ่งใช้ชื่อโปรแกรมเอสเอส (sh) เหมือนกัน ปัจจุบันบอร์นเชลล์ยังเป็นเชลล์พื้นฐานที่มีในยูนิกซ์ทุกตัว และยังสามารถย้ายเชลล์สคริปต์ไปยังยูนิกซ์ระบบอื่นได้โดยไม่ต้องแก้ไข ตัวโปรแกรมมักติดตั้งอยู่ที่ /bin/sh ในระบบยูนิกซ์ส่วนใหญ่

2. ซีเชลล์ (C shell, csh)

เป็นเชลล์ที่พัฒนาขึ้นมาหลังจากบอร์นเชลล์ที่มีรูปแบบคำสั่งและไวยากรณ์เหมือนกับภาษาซี มีฟังก์ชันการทำงานที่สะดวก และยังสามารถควบคุมการไหลของข้อมูลได้ดีกว่าบอร์นเชลล์ ตัวโปรแกรมมักติดตั้งอยู่ที่ /bin/csh

ตารางที่ 2.1 เปรียบเทียบการใช้คำสั่งแบบมีเงื่อนไข ระหว่างบอร์นเชลล์ (ซ้าย) และซีเชลล์ (ขวา)

บอร์นเชลล์	ซีเชลล์
<pre>#!/bin/sh if [\$days -gt 365] then echo This is over a year. fi</pre>	<pre>#!/bin/csh if (\$days > 365) then echo This is over a year. endif</pre>

ตารางที่ 2.2 เปรียบเทียบการตั้งค่าตัวแปรระหว่างบอร์นเชลล์ (ซ้าย) และซีเชลล์ (ขวา)

บอร์นเชลล์	ซีเชลล์
<pre>#!/bin/sh i=2 j=1 while [\$j -le 10] do echo '2 **' \$j = \$i i=`expr \$i '*' 2` j=`expr \$j + 1` done</pre>	<pre>#!/bin/csh set i = 2 set j = 1 while (\$j <= 10) echo '2 **' \$j = \$i @ i *= 2 @ j++ end</pre>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คอร์นเชลล์ (Korn shell, ksh)

เป็นเชลล์ที่พัฒนามาจากต้นแบบของบอร์นเชลล์และซีเชลล์ สามารถทำงานในฟังก์ชันของบอร์นเชลล์ได้ทุกอย่าง ทำให้การเขียนสคริปต์นั้นทำได้ง่ายและรัดกุมยิ่งขึ้น สามารถนำคำสั่งที่ใช้งานไปแล้วกลับมาทำใหม่ได้ ถือว่าคอร์นเชลล์เป็นการรวบรวมเอาข้อดีของบอร์นเชลล์และซีเชลล์มาไว้ด้วยกัน แต่คอร์นเชลล์ไม่ได้มีในยูนิกซ์ทุกตัว

4. แบช (Bourne again shell, bash)

เป็นเชลล์ยูนิกซ์ที่ถูกพัฒนามาจากการเอาบอร์นเชลล์นำมาพัฒนาใหม่ โดยยูนิกซ์ (Unix) ส่วนใหญ่ใช้แบชเป็นเชลล์มาตรฐาน รูปแบบของคำสั่งของแบชครอบคลุมคำสั่งทั้งหมดของบอร์นเชลล์ ดังนั้นสคริปต์ที่เขียนด้วยบอร์นเชลล์ส่วนใหญ่จึงสามารถรันในแบชได้โดยไม่ต้องมาการแก้ไข จะมียกเว้นก็เช่น สคริปต์ที่เรียกใช้ตัวแปรพิเศษในบอร์นเชลล์ หรือใช้คำสั่งภายในของบอร์นเชลล์ รูปแบบคำสั่งในแบชยังได้รับแนวความคิดจากคอร์นเชลล์และซีเชลล์ เวลาใช้เป็นเชลล์ทางบรรทัดคำสั่ง แบชจะเติมชื่อโปรแกรม ชื่อไฟล์ ชื่อตัวแปร ให้ครบโดยอัตโนมัติ เมื่อผู้ใช้กดปุ่มแท็บ (TAB)

2.3.3 คำสั่งพื้นฐานของเชลล์ [7]

- คำสั่ง ls คือ คำสั่งที่ใช้สำหรับแสดงแฟ้มข้อมูล ย่อมาจากคำว่า “list” ตัวเลือก (Option) ที่มีใช้กันใน ls คือ
 - -l จะแสดงผลไฟล์แบบ Long Format ซึ่งจะแสดงการอนุญาตเข้าใช้งานของแฟ้มข้อมูลด้วย
 - -a จะแสดงแฟ้มข้อมูลทั้งหมด

ls [option] [file]

รูปที่ 2.10 การใช้คำสั่ง ls

- คำสั่ง cd คือ คำสั่งที่ใช้สับเปลี่ยนสารบบ (Change directory) ของระบบยูนิกซ์และลินุกซ์
- คำสั่ง cp คือ เป็นคำสั่งที่ใช้สำหรับสำเนาแฟ้มข้อมูล เหมือนกับคัดลอก (Copy)

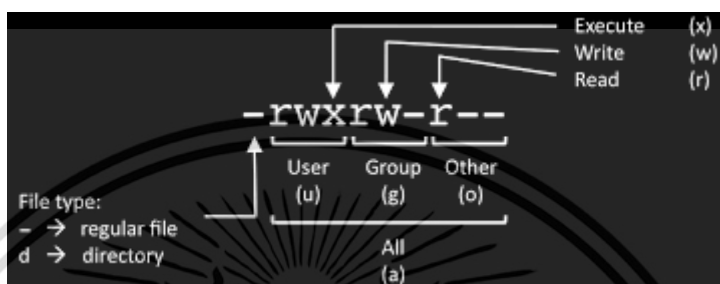
cp [source] [target]

รูปที่ 2.11 การใช้คำสั่ง cp

- คำสั่ง cat คือ ใช้สำหรับดูข้อมูลในแฟ้มข้อมูล และแสดงผลออกทางหน้าจอ (Standard Output) มาจากคำว่า “Concatenate”

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อผู้ญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่ง `chmod` คือ ใช้เปลี่ยนสิทธิของแฟ้มข้อมูล ได้แก่ เจ้าของ (Owner) คนในกลุ่ม (Group) คนอื่น (Other) โดยแต่ละกลุ่มสามารถแบ่งกำหนดสิทธิได้ 3 ประเภทดังนี้
 - read (r)
 - write (w)
 - execute (x)



รูปที่ 2.12 แสดงกลุ่มของการเข้าถึงแฟ้มข้อมูล

สิทธิการเข้าถึงแฟ้มข้อมูลบนยูนิกซ์สามารถแสดงด้วยตัวอักษร หรือ เลขฐานแปด (Octal digits) ตามตารางที่ 2.1

ตารางที่ 2.3 แสดงสิทธิการเข้าถึงแฟ้มข้อมูล

Octal	Symbol	Permission
0	---	No Permissions
1	--x	Execute
2	-w-	Write
3	-wx	Write and Execute
4	r--	Read
5	r-x	Read and Execute
6	rw-	Read and Write
7	rwx	Read, Write and Permissions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งานคำสั่ง `chmod` นั้นสามารถใช้งานได้ 2 แบบคือ การใช้แบบตัวอักษร (Symbols) และ การใช้งานแบบตัวเลขฐานแปด (Octal values)

- การใช้คำสั่งแบบตัวอักษร นั้นมีดังนี้
 - u หมายถึง ผู้เป็นเจ้าของแฟ้ม (User)
 - g หมายถึง ผู้เป็นเจ้าของแฟ้ม (Group)
 - o หมายถึง บุคคลอื่นๆ
 - a หมายถึง ทุกๆ กลุ่ม
 - r หมายถึง สิทธิในการอ่าน
 - x หมายถึง สิทธิในการเขียนและแก้ไข
 - w หมายถึง สิทธิในการเรียกใช้งานหรือค้นหา (ในกรณีของสารบบ)

`chmod [option] [symbols] [file]`

รูปที่ 2.13 การใช้คำสั่ง `chmod` แบบตัวอักษร

- การใช้งานแบบตัวเลขฐานแปด เป็นการจัดเรียงตัวเองออกเป็น 3 กลุ่ม โดยจะให้สิทธิเข้าถึงแฟ้มข้อมูลตารางที่ 2.1

`chmod [option] [octal values] [file]`

รูปที่ 2.14 การใช้คำสั่ง `chmod` แบบตัวเลขฐานแปด

- คำสั่ง `df` คือ คำสั่งที่ใช้ในการตรวจสอบการใช้พื้นที่ของฮาร์ดดิสก์ (Hardisk)
- คำสั่ง `pwd` คือ คำสั่งที่ใช้แสดงสารบบ (Directory) ปัจจุบัน มาจากคำว่า `print work directory`
- คำสั่ง `kill` เป็นคำสั่งที่ขจัดการเลิกโปรเซส (Process) ของระบบ

`kill [option] [ID process]`

รูปที่ 2.15 รูปแบบการใช้งานคำสั่ง `kill`

- คำสั่ง `mkdir` คือ คำสั่งที่ใช้สำหรับสร้างสารบบ มาจากคำว่า `make directory`
- คำสั่ง `mv` คือ คำสั่งที่ใช้ย้ายแฟ้มข้อมูลและสารบบ รวมถึงการเปลี่ยนชื่อด้วย มาจากคำว่า `move`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mv [source] [target]
```

รูปที่ 2.16 รูปแบบการใช้คำสั่ง mv

- คำสั่ง man คือ คำสั่งที่ใช้แสดงและอธิบายการใช้คำสั่งต่างๆ ของระบบยูนิกซ์

```
man [command]
```

รูปที่ 2.17 รูปแบบการใช้งานคำสั่ง man

- คำสั่ง ssh คือ โปรแกรมสำหรับเข้าสู่ระบบและคำสั่งที่เครื่องปลายทางได้ โดยไม่จำเป็นต้องไปใช้งานที่หน้าจอคอนโซลของเครื่อง จุดประสงค์หลักของโปรแกรมคือ ทำหน้าที่แทนโปรแกรมประเภท rlogin, rsh หรือ telnet โดยจะมีการเข้ารหัสข้อมูล (encrypted) เพื่อความปลอดภัยของข้อมูลที่ส่งระหว่าง SSH Client และ SSH Server

2.3.4 การเขียนเชลล์สคริปต์

เชลล์สคริปต์ คือ การนำคำสั่งของลินุกซ์มาเรียงต่อกันตามให้ทำงานตามที่ต้องการ โดยอาจจะมีการเพิ่มเติมฟังก์ชันขึ้นมา หรือโดยสามารถเขียนในโปรแกรมเขียนโปรแกรม (Editor) ได้ โดยในลินุกซ์จะมีโปรแกรมเขียนภาษา ที่เป็นพื้นฐานให้หลายตัว เช่น วิโอ (vi), นาโน (nano), จีอีดิท (gedit) เป็นต้น

การเขียนเชลล์สคริปต์จะต้องเขียนตามชนิดของเชลล์ เพราะเชลล์แต่ละชนิดมีไวยากรณ์ที่แตกต่างกัน หากเขียนเชลล์สคริปต์ไม่ตรงตามเชลล์ อาจจะทำให้ตัวสคริปต์ที่เขียนใช้งานไม่ได้

2.4 ระบบการควบคุมรุ่นของซอฟต์แวร์และกิต

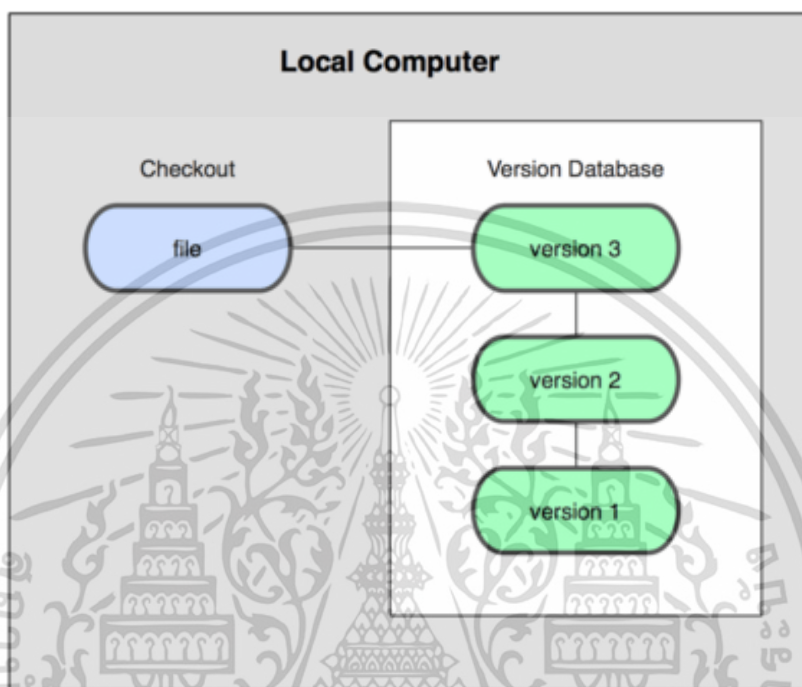
2.4.1 ความหมายของระบบควบคุมรุ่นของซอฟต์แวร์ [8]

ระบบควบคุมรุ่นของซอฟต์แวร์ (Version control system) คือ เป็นระบบที่จัดเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่ง หรือหลายไฟล์ เพื่อที่คุณสามารถเรียกดูในรุ่นหนึ่งกลับมาดูได้เสมอ เราสามารถใช้ ระบบควบคุมรุ่นของซอฟต์แวร์กับไฟล์ชนิดไหนก็ได้ ระบบควบคุมรุ่นของซอฟต์แวร์สามารถเปรียบเทียบการแก้ไขไฟล์ที่เกิดขึ้นในอดีต สามารถดูได้ว่าใครเป็นคนแก้ไขไฟล์แก้ไขเมื่อไร แก้ไขอะไรไปบ้าง เป็นต้น และยังสามารถกู้ไฟล์ที่ลบไปแล้วได้อย่างง่ายดาย

2.4.2 ประเภทของระบบการควบคุมรุ่นซอฟต์แวร์ [9]

1. ระบบการควบคุมรุ่นของซอฟต์แวร์แบบ local

เป็นการคัดลอกไฟล์ที่ต้องการจะเก็บไว้ในไดเรกทอรีใหม่ อาจจะตั้งชื่อไดเรกทอรีด้วยวันที่ วิธีนี้เป็นวิธีที่แพร่หลาย เพราะสามารถทำได้ง่ายแต่ในขณะเดียวกันเป็นวิธีที่เกิดขึ้นผิดพลาดได้ง่ายเหมือนกัน



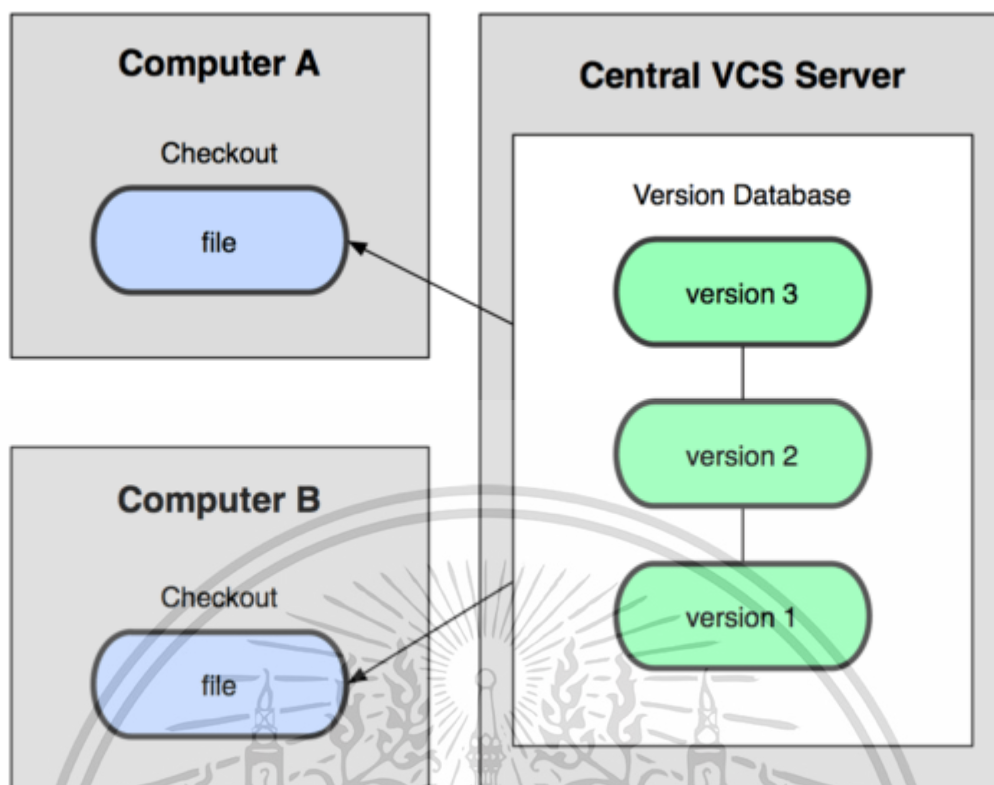
รูปที่ 2.18 ระบบ Version control แบบ local

2. ระบบควบคุมรุ่นของซอฟต์แวร์แบบรวมศูนย์

ระบบถูกพัฒนาเนื่องจากนักพัฒนาต้องการพัฒนาร่วมกับนักพัฒนาคนอื่น จึงเกิดระบบนี้ขึ้นมา เรียกว่า Centralized Version Control System (CVCSs) หรือระบบ Version Control System แบบรวมศูนย์กลาง ซอฟต์แวร์ที่ใช้ระบบนี้ได้แก่ CVS, Subversion และ Perforce ซึ่งมีเซิร์ฟเวอร์กลางที่เก็บไฟล์ทั้งหมดเอาไว้ที่เดียวกัน เพื่อให้ผู้ใช้หลายๆ คนสามารถต่อเข้ามาเพื่อดึงข้อมูลหรือไฟล์จากศูนย์กลางออกไปแก้ไขที่เครื่องของตัวเองได้ ระบบนี้ถูกนำมาใช้หลายปี

ระบบแบบรวมศูนย์รวมมีข้อได้เปรียบกว่าระบบ local ในหลายๆ ด้าน เช่น ทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจกต์กำลังทำอะไรอยู่ สามารถจัดการข้อมูลได้ที่ศูนย์กลางที่เดียว ทำให้จัดการข้อมูลได้ง่ายกว่า แต่ระบบนี้มีจุดอ่อนตรงที่การรวมศูนย์กลาง ทำให้เป็นจุดเดียวที่เซิร์ฟเวอร์หมายความว่าถ้าเซิร์ฟเวอร์นั้นเกิดปัญหา ก็จะไม่มีความสามารถทำงานร่วมกันได้ หรือบันทึกการเปลี่ยนแปลงงานที่กำลังทำอยู่ไปที่เซิร์ฟเวอร์ได้เลย หรือถ้าหน่วยความจำสำรองของเครื่องเซิร์ฟเวอร์นั้นเกิดปัญหาและไม่มีการสำรองข้อมูลไว้ หมายความว่าข้อมูลทั้งหมดที่อยู่บนเครื่องเซิร์ฟเวอร์จะหายไปทั้งหมด เหลือเพียงงานที่อยู่บนเครื่องแต่ละเครื่องเท่านั้น

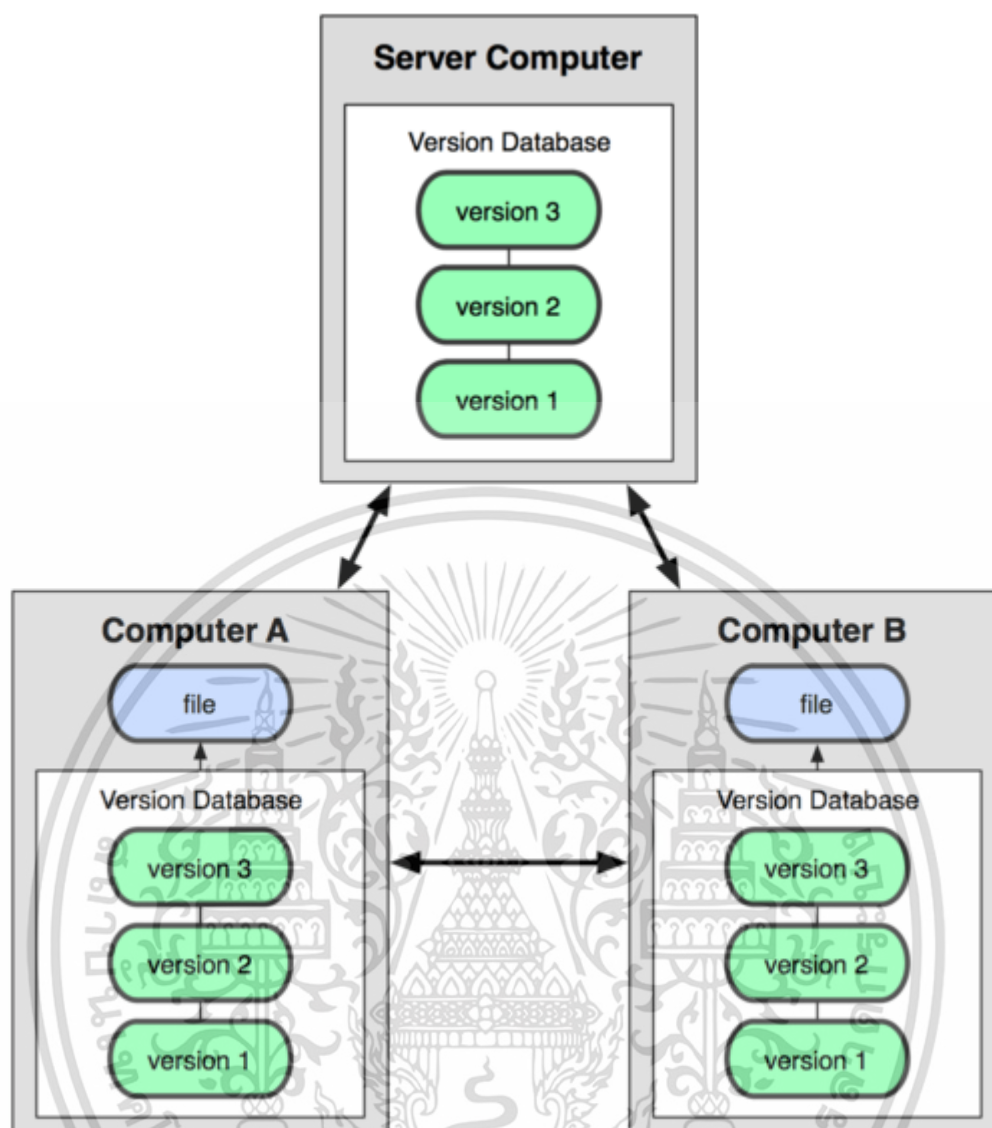
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 ระบบควบคุมรุ่นของซอฟต์แวร์ แบบรวมศูนย์

3. ระบบควบคุมรุ่นของซอฟต์แวร์ แบบกระจายศูนย์

ชื่อเต็มของระบบนี้คือ Distributed version control system (DVCSs) ซอฟต์แวร์ที่ใช้ระบบนี้คือ Git, Mercurial, Bazaar หรือ Darcs) ในระบบงานแบบนี้แต่ละคนไม่ได้คัดลอกงานล่าสุดเท่านั้น แต่ได้คัดลอก repository มาเลย หมายความว่า ถึงแม้ว่าเซิร์ฟเวอร์จะเสียหรือมีปัญหา ไคลแอนต์ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository ของเครื่องไคลแอนต์ยังสามารถคัดลอกกลับไปทีเครื่องเซิร์ฟเวอร์เพื่อกู้ข้อมูลได้อีกด้วย



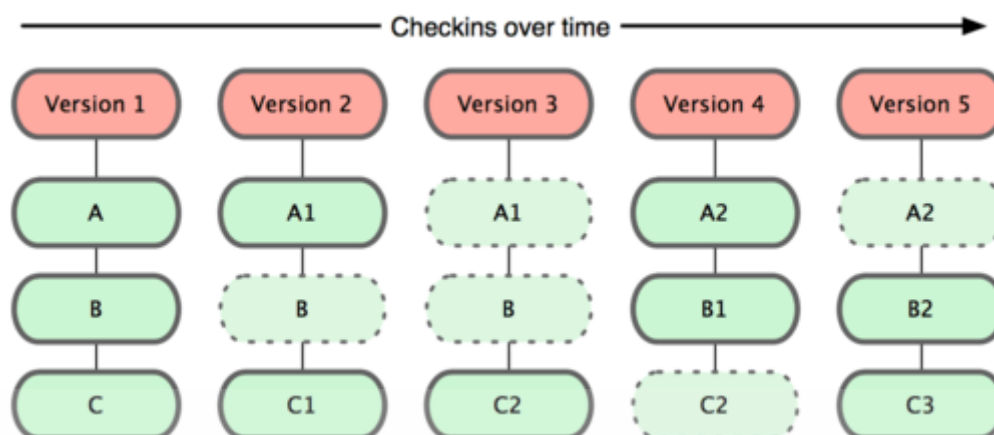
รูปที่ 2.20 ระบบควบคุมรุ่นของซอฟต์แวร์ แบบกระจายศูนย์

2.4.3 ความหมายของกิต [10]

กิต (GIT) คือ เป็นซอฟต์แวร์ที่ใช้ในระบบควบคุมรุ่นของซอฟต์แวร์ (Version control system) โดยเริ่มแรกถูกพัฒนามาใช้สำหรับลินุกซ์ โดยลินุส โตร์วัลดส์ ไดรอกทอร์ของกิตทุกตัวมีการเก็บประวัติทั้งหมด ปัจจุบันดูแลโดย ฮุนิโอ ฮามาโน กิตเป็นซอฟต์แวร์ฟรีและแจกจ่ายในลักษณะของ GPL (GNU General Public License)

การทำงานของกิต จะมองข้อมูลเป็นเหมือนภาพถ่าย (Snapshot) ของระบบไฟล์ขนาดเล็กๆ ทุกครั้งที่มีการบันทึกสถานะภาพของโปรเจก (Commit) ลงในกิต กิตจะทำการถ่ายภาพของไฟล์ทั้งหมดในตอนนั้นและบันทึกการเปลี่ยนแปลง เพื่อให้การจัดเก็บมีประสิทธิภาพ ถ้าไฟล์ไม่ได้มีการเปลี่ยนแปลง กิตจะไม่บันทึกไฟล์นั้นอีกครั้ง เพียงแต่ทำการเชื่อมโยงไปยังไฟล์เดิมที่เคยถูกบันทึกเอาไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 การเก็บข้อมูลแบบภาพถ่ายของกิต

2.5 วิบีสคริปต์ (VBScript)

2.5.1 ความหมายของวิบีสคริปต์ [11]

VBScript คือ ภาษาแบบ Script นั่นก็คือ เป็นภาษาโปรแกรมแบบหนึ่งแต่จะมีขนาด และ โครงสร้างที่เล็กกว่าโปรแกรมภาษาอื่น ซึ่งเป็นภาษาที่ใช้ในการเขียนโปรแกรมทางอินเทอร์เน็ต ถูก พัฒนาโดยบริษัทไมโครซอฟท์ สามารถที่จะทำให้โฮมเพจที่เขียนด้วยภาษา HTML ซึ่งเป็นข้อมูลแบบ คงที่ให้เป็นโฮมเพจที่ข้อมูลเปลี่ยนแปลงตลอดเวลาตามผู้ใช้ต้องการได้ ภาษา VBScript เป็นภาษาที่ถูก พัฒนามาจากภาษา Visual Basic ดังนั้นจึงโครงสร้างคล้ายกับภาษา Visual Basic โดยภาษา VBScript นี้ สามารถใช้ร่วมกับคำสั่งต่าง ๆ ของภาษา HTML เพื่อให้เว็บเพจทำงานได้มี ประสิทธิภาพมากขึ้น

2.5.2 ชนิดข้อมูลของวิบีสคริปต์

VBScript มีชนิดข้อมูลเพียงตัวเดียวเรียกว่า Variant ซึ่งเป็นชนิดข้อมูลแบบพิเศษที่สามารถ บรรจุชนิดข้อมูลที่แตกต่างกันได้หลายชนิด ขึ้นอยู่กับว่าจะใช้ชนิดข้อมูลนี้อย่างไร ชนิดข้อมูลแบบ Variant สามารถเป็นได้ทั้งตัวเลขและสายอักขระ โดยที่ชนิดข้อมูล Variant จะประพฤติตัวเหมือน ตัวเลขเมื่อใช้ร่วมกับข้อมูลตัวเลข และประพฤติตัวเหมือนเป็นสายอักขระเมื่อใช้ร่วมกับข้อมูลสาย อักขระ โดยเมื่อต้องการทำให้ตัวเลขมีชนิดข้อมูลเป็นแบบอักขระทำได้โดยล้อมรอบตัวเลขนั้นด้วย เครื่องหมาย “ ”

2.6 HP Quality Center (QC)

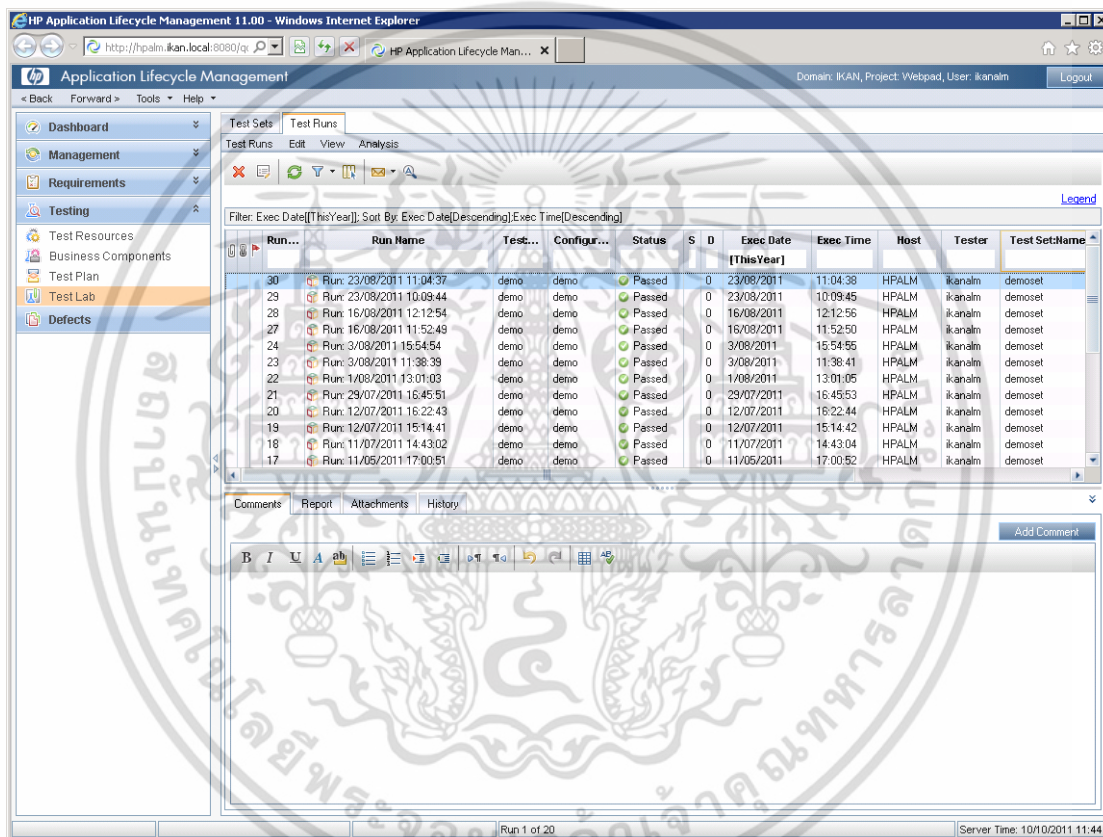
2.6.1 ความหมายของ HP Quality Center (QC) [12]

เป็นระบบที่รองรับบทบาทหน้าที่ด้านต่างๆ ของบุคลากรในองค์กร โดยผนวกรวมเครื่องมือ สำคัญๆ ที่ใช้ในการพัฒนาทั้งหมดเข้าไว้ด้วยกัน เพื่อสนับสนุนผู้มีส่วนเกี่ยวข้องให้สามารถนำ แอปพลิเคชันมาใช้งานได้อย่างเต็มที่ ทั้งยังช่วยพัฒนากระบวนการดำเนินงานภายในและข้ามทีมงาน พัฒนาแอปพลิเคชันให้เป็นแบบอัตโนมัติ ตลอดจนถึงดูแลและควบคุมวงจรการพัฒนาแอปพลิเคชันให้มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความรวดเร็วและมีประสิทธิภาพสูงสุด ครอบคลุมตั้งแต่การจัดการตามความต้องการใช้งานจนถึงการควบคุมคุณภาพ ประสิทธิภาพการทำงาน และการส่งผลิตภัณฑ์ใหม่ออกสู่ตลาด

สำหรับควบคุมคุณภาพของระบบจะประกอบไปด้วยหลาย Module แต่ที่นิยมนำมาใช้งานในเรื่องการ Test คือ Requirements, Test Plan และ Defect Manager เราสามารถที่จะนำ Requirement และ Test Case ใส่เข้าไปในระบบ เมื่อเราทำการทดสอบตาม Test case แล้ว ให้มาบันทึก Defect ไว้ใน Quality Center หลังจากนั้น Quality Center จะทำการส่งอีเมลแจ้งเตือนไปยังผู้ที่เกี่ยวข้องต่างๆ ให้รับทราบว่า มี defect เกิดขึ้น เมื่อทำการแก้ไข defect และ re-test เสร็จเรียบร้อยแล้ว ให้ทำการ close defect ด้วย จาก Tool ตัวนี้จะสามารถ Track Status ของการทดสอบและสามารถที่จะทราบได้ว่าจะมีการทำ Test Case ครอบคลุมทุก Requirement แล้วหรือยัง



รูปที่ 2.22 โปรแกรม HP Quality Center (QC)

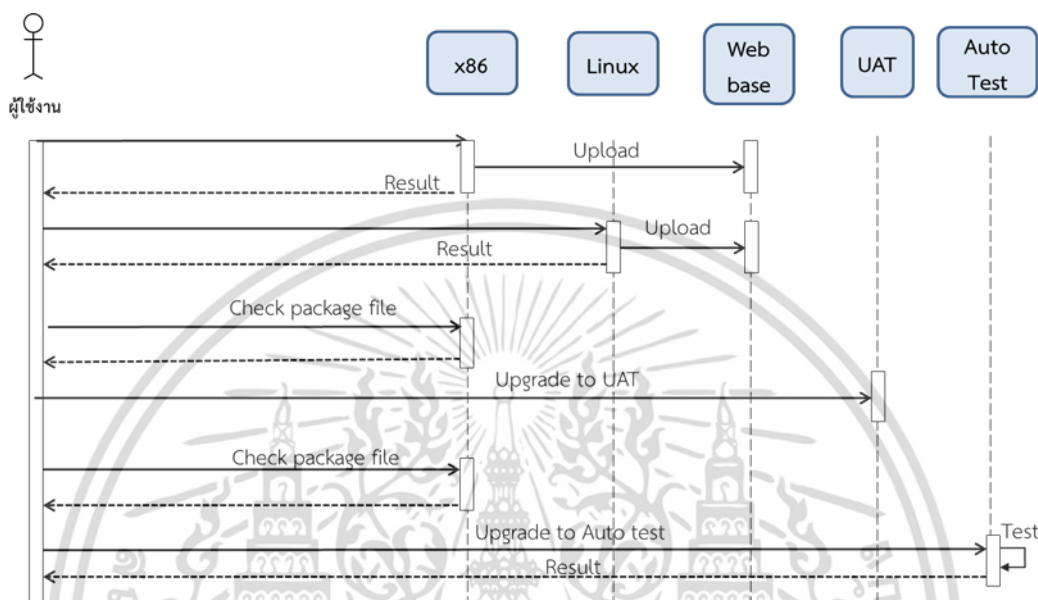
2.7 กระบวนการพัฒนาระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม

ในการจัดการทำงานเพื่อควบคุมคุณภาพของซอฟต์แวร์นั้น ระบบนี้เป็นระบบที่ทำงานต่อจากกระบวนการพัฒนาซอฟต์แวร์ นั่นคือหลังจากการเขียนโปรแกรมแล้ว โดยทางระบบเดิมเป็นการทำ โดยผู้ใช้ใช้งานทั้งหมด เป็นระบบที่ใช้สนับสนุนกระบวนการการพัฒนาซอฟต์แวร์ โดยซอฟต์แวร์ที่ทำการพัฒนานั้นคือ ระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ (Electronic Trading) โดยกระบวนการที่ทางผู้จัดทำได้ทำการทำขึ้นคือการเปลี่ยนกระบวนการทำงานของการพัฒนาระบบการจำลองการแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม ให้เป็นกระบวนการการทำงานเป็น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้เพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นใบเขียวระบบเช่นนี้ไม่ควรนำ
 ไม่ว่าการมีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตโนมัติ โดยใช้ทฤษฎีในหัวข้อที่ 2.1 คือการบูรณาการอย่างต่อเนื่อง (Continuous Integration) และใช้เครื่องมือเจกันส์ในทฤษฎีที่ 2.2

ในกระบวนการทำงานแบบเดิมนั้น การพัฒนาซอฟต์แวร์ในขั้นตอนต่างๆ ต้องอาศัยมนุษย์ในการทำงาน ตั้งแต่การคอมไพล์ระบบ การติดตั้งระบบ การทดสอบระบบ โดยกระบวนการเหล่านี้สามารถทดแทนได้ด้วยระบบอัตโนมัติ โดยระบบการทำงานแบบเดิมนั้นทำงานดังรูปที่ 2.23



รูปที่ 2.23 แผนภาพ Sequence Diagram แสดงการทำงานของระบบเดิม

จากรูปที่ 2.23 อธิบายได้ว่าผู้ใช้งานต้องกระทำงานทุกอย่างด้วยตัวเองทั้งหมด โดยอธิบายระบบได้ดังนี้

1. X86 คือ เครื่องเซิร์ฟเวอร์ที่ทำการคอมไพล์ระบบ ซึ่งเป็นระบบปฏิบัติการ Windows
2. Linux คือ เครื่องเซิร์ฟเวอร์ที่ทำการคอมไพล์ระบบ ซึ่งเป็นระบบปฏิบัติการ Linux
3. Web Base คือ เครื่องเซิร์ฟเวอร์ที่ไว้สำหรับเก็บระบบซึ่งสามารถเรียกใช้ผ่าน Web Browser
4. UAT คือ เครื่องเซิร์ฟเวอร์ที่มีสภาพแวดล้อมที่เหมือนหรือใกล้เคียงมากที่สุด กับสภาพแวดล้อมของการใช้งาน เพื่อให้การทดสอบใกล้เคียงกับการทำงานจริงมากที่สุดและได้ผลผลการทดสอบที่น่าเชื่อถือ ในที่นี้ให้เป็นเครื่องผู้ใช้งาน
5. Auto Test คือ เครื่องเซิร์ฟเวอร์ที่ใช้สำหรับทดสอบเกี่ยวกับ Automate Test

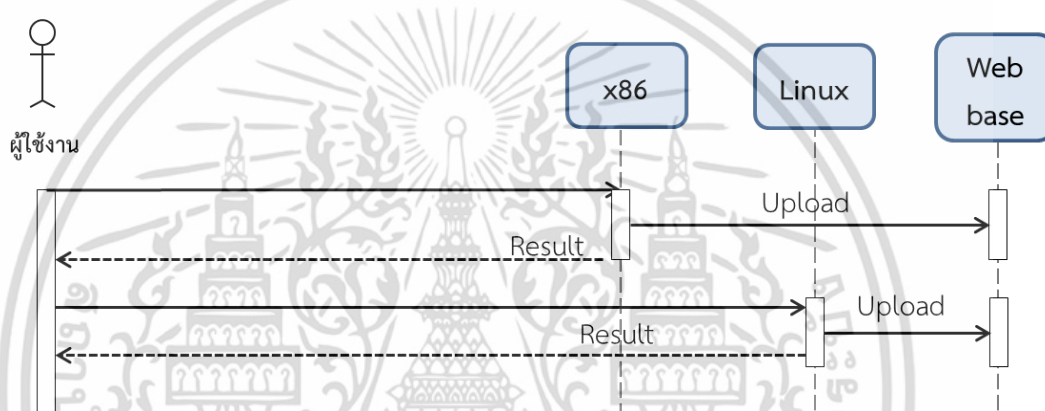
2.8 กระบวนการคอมไพล์ระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม

กระบวนการคอมไพล์ระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิมนั้น หากจะมีการเริ่มคอมไพล์ระบบ ต้องใช้ทรัพยากรมนุษย์ในการทำงานเสมอ โดยมีขั้นตอนในการทำงานดังนี้

- 1) ในแต่ละสัปดาห์ต้องกำหนดว่าจะมีการ Build package ในวันที่ไหน ซึ่งจะกำหนดโดยผู้ใช้โดยการประชุมประจำสัปดาห์

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) หากผู้พัฒนาต้องการที่จะ Build Package ทางผู้พัฒนานั้นจะต้องเข้าสู่ระบบด้วยบัญชีผู้ใช้ที่ทางบริษัทกำหนดให้ โดยต้องทำการเข้าสู่ระบบที่เครื่องเซิร์ฟเวอร์ทั้ง 2 เครื่อง ได้แก่ เครื่องเซิร์ฟเวอร์ x86 และเครื่องเซิร์ฟเวอร์ลินุกซ์ โดยเครื่องทั้งสองเครื่องนั้นอยู่ที่เมืองนอตทิงแฮม ประเทศอังกฤษ
- 3) ค้นหาที่อยู่ของโพลเดอร์ที่จะทำการ Build Package โดยโปรแกรม Telnet
- 4) พิมพ์คำสั่งในการ Build Package ในเครื่องเซิร์ฟเวอร์ x86 เป็นเครื่องแรก
- 5) จากนั้นทำการเข้าไปที่เครื่องเซิร์ฟเวอร์ลินุกซ์ เพื่อทำการเตรียมขั้นตอนต่างๆ ที่กล่าวมาข้างต้น
- 6) เมื่อเครื่องเซิร์ฟเวอร์ x86 ทำการ Build Package สำเร็จแล้ว จึงทำการพิมพ์คำสั่งเพื่อทำการ Build Package บนเครื่องเซิร์ฟเวอร์ลินุกซ์



รูปที่ 2.24 แผนภาพ Sequence Diagram แสดงการทำงานของคอมพิวเตอร์ระบบการจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศแบบเดิม

จะเห็นได้ว่าการทำงานในระบบงานเดิมนั้นมีการทำงานโดยใช้ทรัพยากรมนุษย์ในการทำงานเป็นหลักในการทำงาน จึงทำให้เกิดปัญหาต่างๆ ดังนี้

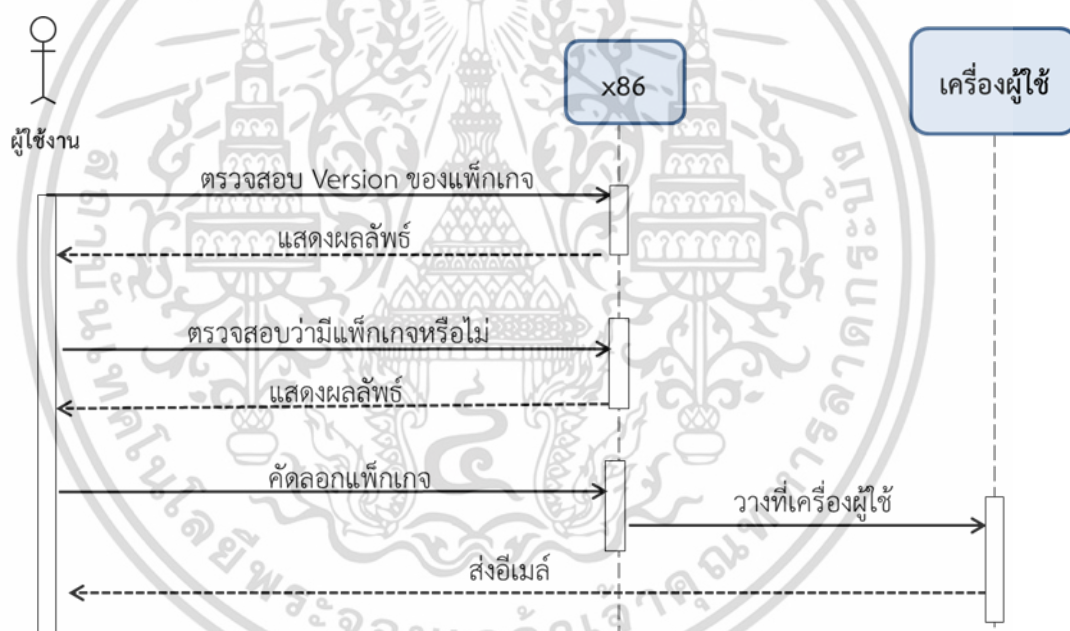
- 1) ผู้พัฒนาไม่สามารถจำคำสั่งในการทำงานได้ ทำให้ต้องเสียเวลาในการค้นหา
- 2) หากไม่มีการกำหนดว่าในหลายๆ สัปดาห์ว่าจะทำการ Build Package วันอะไร อาจจะเกิดการชนกันของ Package ได้
- 3) ในการทำงานนั้น จะต้องมีการลงชื่อเข้าใช้ระบบด้วย
- 4) หากต้องการที่จะ Build Package เครื่องเซิร์ฟเวอร์ลินุกซ์ ต้องทำเครื่องเซิร์ฟเวอร์ x86 ให้เสร็จก่อน จึงจะสามารถที่จะทำงานได้
- 5) การตั้งเวลาให้ระบบทำงานอัตโนมัติตอนกลางคืนนั้นทำได้ยาก เพราะต้องตั้งค่าในครอนแท็บ (Crontab)
- 6) หากเริ่มทำการทำงานของระบบ ต้องเปิดเครื่องคอมพิวเตอร์ที่ทำงานทิ้งเอาไว้ เพราะหากปิดเครื่องนั้น จะหยุดการทำงานของ การ Build Package ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 กระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม

กระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิมนั้นมีขั้นตอนที่มากมาย และยุ่งยาก ดังนี้

- 1) ผู้พัฒนาต้องใช้โปรแกรม Putty ในรีโมตเซิร์ฟเวอร์เข้าสู่เครื่องเซิร์ฟเวอร์x86
- 2) ผู้พัฒนาต้องใช้ชุดคำสั่งในการหาที่อยู่ของแพ็กเกจของระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ
- 3) ผู้พัฒนาต้องตรวจสอบว่ามีแพ็กเกจของระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศหรือไม่
- 4) ผู้พัฒนาต้องตรวจสอบว่าแพ็กเกจของระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศเป็นรุ่นล่าสุดหรือไม่
- 5) ผู้พัฒนาต้องคัดลอกแพ็กเกจของระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศรุ่นล่าสุดจากเครื่องเซิร์ฟเวอร์x86 มายังเครื่องของผู้พัฒนา



รูปที่ 2.25 แผนภาพ Sequence Diagram แสดงการทำงานของกรอัปเดตระบบการจำลองการแลกเปลี่ยนเงินตราต่างประเทศแบบเดิม

จะเห็นได้ว่าการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศนั้นมีขั้นตอนที่ยุ่งยากซับซ้อน จึงเกิดปัญหาดังนี้

- 1) ต้องพิมพ์ชุดคำสั่งในการอัปเดตระบบทุกขั้นตอน โดยการพิมพ์ชุดคำสั่งนั้นต้องทำโดยผู้พัฒนาระบบ ทำให้มีโอกาสในการผิดพลาดสูง เช่นการพิมพ์ชุดคำสั่ง การดูแลรุ่นของระบบจำลองผิด
- 2) ผู้พัฒนาระบบไม่อยากจะอัปเดตระบบ เพราะว่ามีกระบวนการในการอัปเดตระบบที่ยุ่งยาก

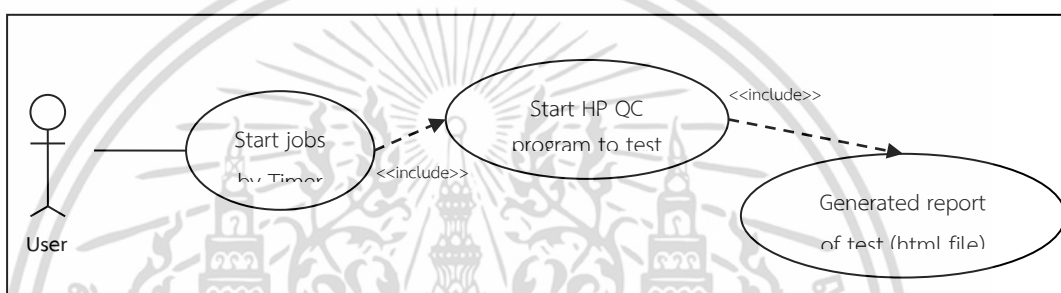
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) ถ้าหากอยู่ในระหว่างการอัปเดตระบบอยู่ แล้วมีการปิดเครื่องคอมพิวเตอร์นั้น การอัปเดตระบบก็จะสิ้นสุดลง

2.10 การแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์

โดยการทดสอบนั้นจะมีเครื่องมือ (Tool) ที่ใช้ในการทดสอบระบบอยู่แล้ว นั่นคือ HP Quality Center (QC) โดยการทดสอบนั้นจะทำงานบนเครื่องเซิร์ฟเวอร์ที่ไว้สำหรับการทดสอบ โดยเฉพาะ และผลที่ได้จากการทดสอบนั้นจะอยู่ในเครื่องมือที่ใช้ในการทดสอบที่เซิร์ฟเวอร์ที่ใช้ทดสอบ

- ยูสเคสของแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์



รูปที่ 2.26 แผนภาพยูสเคสของแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์

จากรูปที่ 2.26 ระบบการควบคุมคุณภาพของซอฟต์แวร์มีผู้กระทำคือ ผู้ใช้งาน (User) และระบบมีหน้าที่หลักในการทำงานทั้งหมดดังนี้

- 1) ยูสเคส Start Job รับผิดชอบเกี่ยวกับการ Remote Desktop เข้าไปที่เครื่องเซิร์ฟเวอร์ ที่ทำการควบคุมคุณภาพของซอฟต์แวร์
- 2) ยูสเคส Start HP QC program to test รับผิดชอบเกี่ยวกับการเปิดโปรแกรม HP Quality Center (QC) และทำการทดสอบการควบคุมคุณภาพของซอฟต์แวร์
- 3) ยูสเคส Generated report of test (html file) รับผิดชอบเกี่ยวกับการสร้างผลลัพธ์ของการทดสอบให้อยู่ในรูปแบบของ HTML ไฟล์ เพื่อสามารถเปิดแสดงได้ในหน้าเว็บไซต์

โดยรายงานผลลัพธ์ของผลการทดสอบควบคุมคุณภาพของซอฟต์แวร์จะอยู่ในรูปของ HTML ไฟล์ ซึ่งสามารถเปิดได้เฉพาะที่หน้าเว็บเบราว์เซอร์เท่านั้น จึงทำให้การที่จะดูผลลัพธ์ของการทดสอบนั้นมีความยุ่งยากไม่สะดวกสำหรับผู้ใช้งาน

2.11 การแสดงผล Release Notes แบบเดิม

1) โครงสร้างของการแสดง Release notes ของซอฟต์แวร์

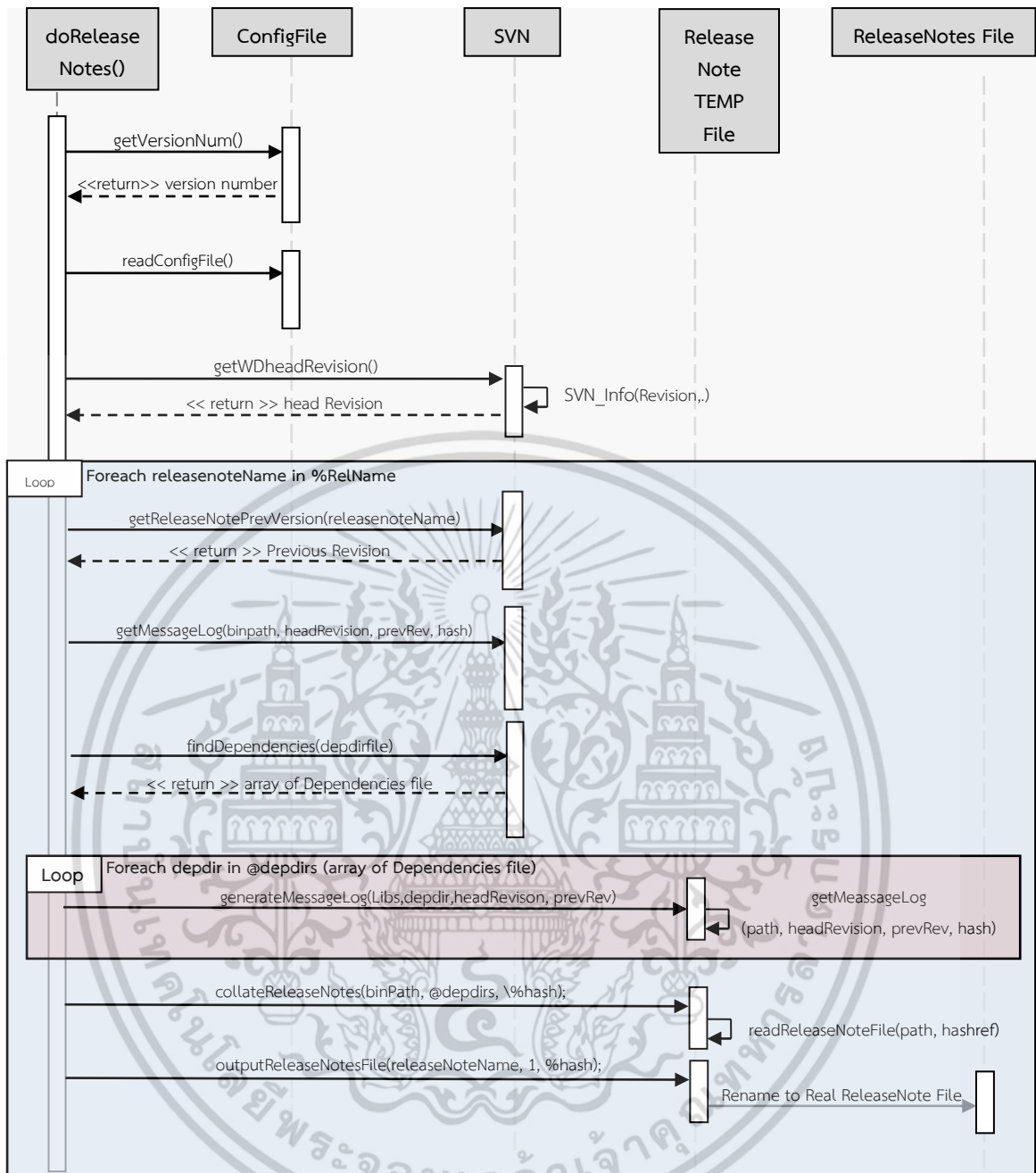
โครงสร้างของการปรับปรุงการสร้าง Release notes โดยเดิมที่การสร้าง Release notes จะเริ่มหลังจากมีการคอมไพล์โค้ด และการสร้าง release notes นั้นจะทำอยู่ภายใต้การควบคุมรุ่น เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาก็เท่านั้น เมื่อนักพัฒนาไปใช้ประโยชน์ในครั้งถัดมาว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Version Control) ซึ่งก็คือ SVN (Subversion) โดยการคอมไพล์และสร้างไฟล์ Release notes นั้น จะมีสคริปต์ที่ใช้ในเป็นคำสั่งในการทำงาน โดยการทำงานของสคริปต์นั้นจะมีการทำงานอยู่ทั้งหมด 3 Modules คือ

- Build เป็นส่วนที่เกี่ยวกับการคอมไพล์โค้ดและทอรวมเป็นแพ็คเกจ
- Upload เป็นส่วนในการอัปโหลดไฟล์และสร้าง Release notes
- Test เป็นส่วนในการทดสอบโค้ด

โดยแต่ละฟังก์ชันทำงานดังนี้

- getVersionNum() คือ ฟังก์ชันการดึงเลขรุ่นปัจจุบันของซอฟต์แวร์
- SVN_Info() คือ ฟังก์ชันการเรียกการแก้ไขครั้งล่าสุดจากการเปลี่ยนแปลงของโค้ด
- getMessageLog() คือ ฟังก์ชันที่ใช้ค้นรายละเอียดของการเปลี่ยนแปลงของโค้ด ทั้งผู้แก้ไข รายละเอียดโค้ดที่เปลี่ยนแปลง จากชุดคำสั่งของ Subversion
- findDependencies() คือ ฟังก์ชันที่อ่านค่าจากไฟล์ที่เก็บรายชื่อโปรแกรมที่เกี่ยวข้องกับการเปลี่ยนแปลงของโค้ด
- generateMessageLog() คือ ฟังก์ชันที่ใช้ในการสร้างไฟล์ Release Notes Temp ขึ้นมาเก็บ ข้อมูลจากฟังก์ชัน getMessageLog()
- outputReleaseNotFile() คือ ฟังก์ชันในการเขียนรายละเอียดจาก ฟังก์ชัน generateMessageLog() ต่อจาก Release Notes รุ่นเก่าๆ



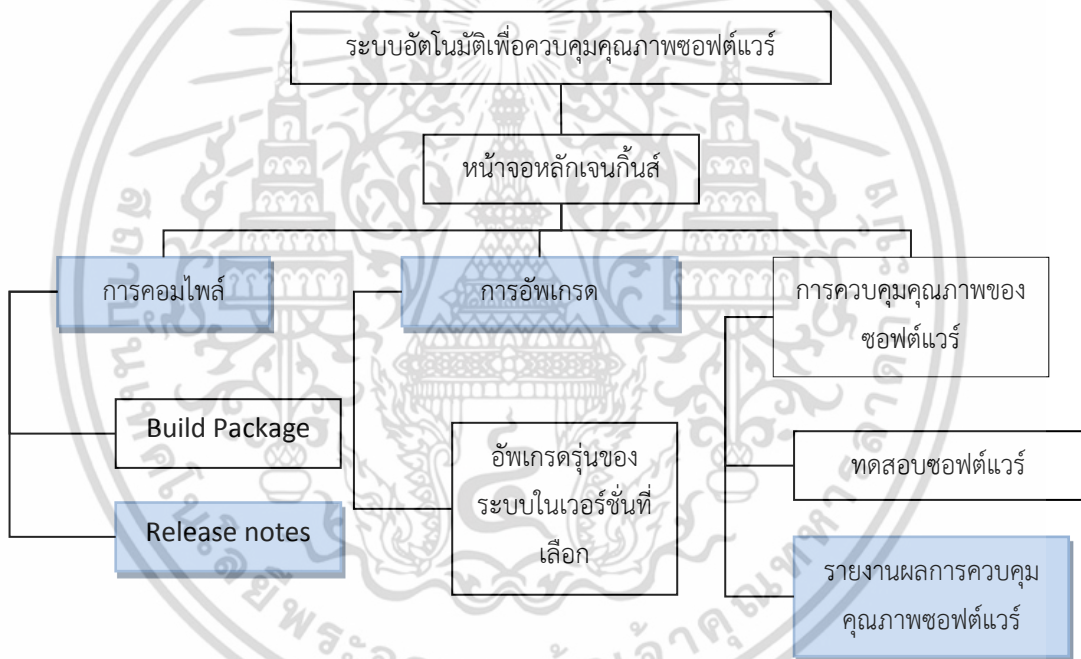
รูปที่ 2.27 แผนภาพ Sequence Diagram ของการทำงานของ Release Notes โดย SVN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 วิธีการดำเนินงานวิจัย

ในบทนี้จะแสดงถึงวิธีการและขั้นตอนของการจัดการระบบการพัฒนาการแลกเปลี่ยนเงินตราต่างประเทศแบบเดิมในหัวข้อ 2.7 ให้เป็นระบบการพัฒนาแบบอัตโนมัติโดย การจัดการนี้เป็นกระบวนการทำงานที่ต่อจากการพัฒนาซอฟต์แวร์ นั่นคือหลังจากเขียนโปรแกรมเสร็จแล้ว โดยจะใช้ทฤษฎีในหัวข้อที่ 2.1 คือการบูรณาการอย่างต่อเนื่อง (Continuous Integration) และใช้เครื่องมือเงินกันส์ในทฤษฎีที่ 2.2

โดยในการจัดการกระบวนการทำงานแบบระบบอัตโนมัตินี้ จะช่วยลดความผิดพลาดที่เกิดจากการกระทำของมนุษย์ ช่วยลดการเสียเวลาในกระบวนการทำงาน และยังช่วยเพิ่มประสิทธิภาพการทำงานให้กับนักพัฒนาอีกด้วย โดยมีโครงสร้างของระบบดังนี้



รูปที่ 3.1 โครงสร้างของระบบอัตโนมัติเพื่อควบคุมคุณภาพของซอฟต์แวร์

จากรูปที่ 3.1 นั้นสามารถแบ่งกระบวนการทำงานได้ออกเป็น 4 กลุ่มด้วยกันดังนี้

- กลุ่มที่ 1 การคอมไพล์ หรือกระบวนการคอมไพล์ระบบแลกเปลี่ยนเงินตราต่างประเทศ คือกระบวนการทำงานหลังจากการเขียนโปรแกรมและการ check in โค้ดของโปรแกรมเข้าสู่ระบบควบคุมรุ่นเสร็จสิ้นแล้ว โดยเครื่องมือเงินกันส์จะทำการจัดการเรื่องการคอมไพล์โค้ดของซอฟต์แวร์เพื่อให้ได้ซอฟต์แวร์ออกมา โดยจะจัดการอัตโนมัติตามที่ผู้จัดทำทำการตั้งค่าและเขียนสคริปต์เพิ่มเติมไว้ โดยจะอธิบายในหัวข้อที่ 3.1

- กลุ่มที่ 2 การอัปเดต หรือกระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศ เป็นกระบวนการทำงานที่ต่อเนื่องมาจากหัวข้อที่ 3.1 โดยจะทำการอัปเดตซอฟต์แวร์รุ่นใหม่ไป

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญตเห็นใบเซอร์ไอเซนด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

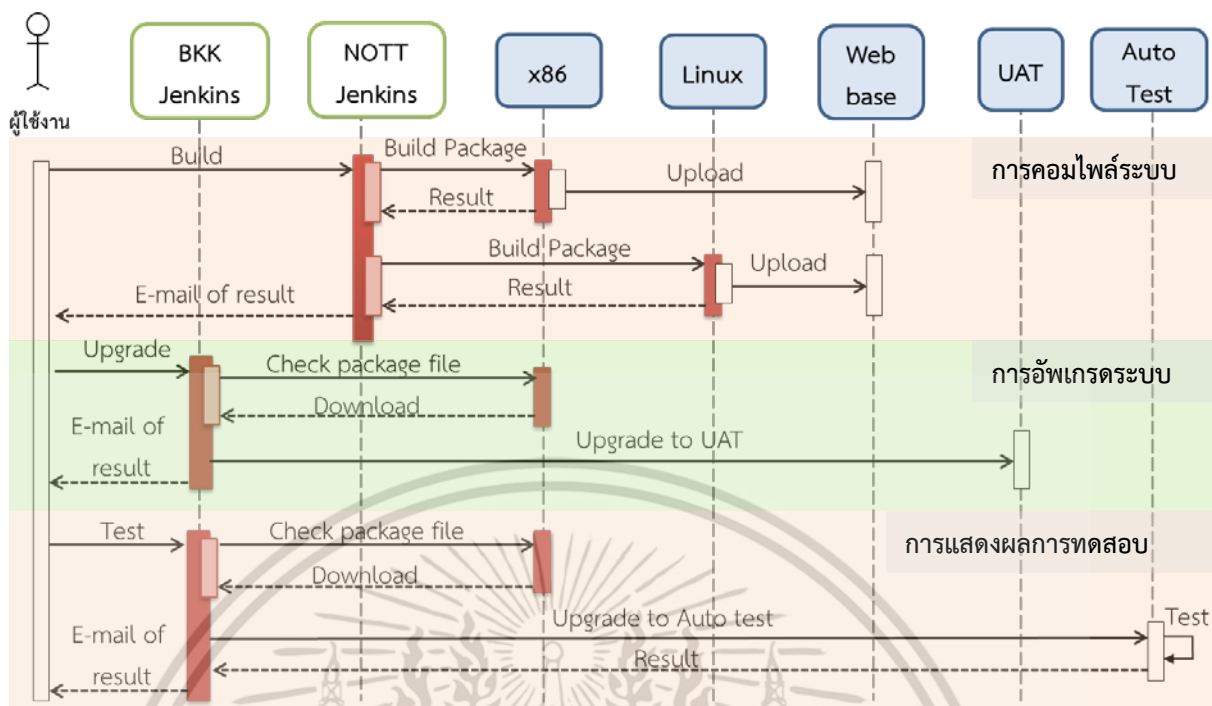
แทนที่ซอฟต์แวร์รุ่นเก่าโดยใช้เครื่องมือเจเกินส์เป็นเครื่องมือตัวกลางในการทำงาน โดยทางผู้จัดทำได้ทำการเขียนสคริปต์เพิ่มเติมเพื่อตรวจสอบเงื่อนไขในการอัปเดตซอฟต์แวร์ อธิบายในหัวข้อที่ 3.2

- กลุ่มที่ 3 รายงานผลการควบคุมคุณภาพซอฟต์แวร์ หรือการแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์ เป็นกระบวนการปรับปรุงการแสดงผลของการทดสอบโดยผู้จัดทำได้ทำการปรับปรุงสคริปต์การแสดงผลการทดสอบให้แสดงผ่าน Text file ได้และสามารถแสดงผลถึงผู้ใช้งานได้ทันทีหลังจากการทดสอบสำเร็จ โดยอธิบายในหัวข้อที่ 3.3

-กลุ่มที่ 4 Release Notes หรือการปรับปรุงการแสดงผล Release notes ของซอฟต์แวร์ เนื่องจากทางบริษัทได้มีการเปลี่ยนระบบควบคุมรุ่นจาก Subversion เป็น GIT ทำให้เกิดปัญหาในการแสดงผลของ Release Notes โดยทางผู้จัดทำจึงได้ทำการปรับปรุงสคริปต์ที่ใช้ในการแสดงผล ซึ่งอธิบายในหัวข้อที่ 3.4

ในการวิเคราะห์ระบบงานใหม่ ได้มีการเอาแนวทางวิธีการทำงานแบบอัตโนมัติมาช่วยในการวิเคราะห์ระบบงาน เพื่อให้ได้ระบบงานที่มีความเสถียรมากยิ่งขึ้น โดยได้ปรับปรุงขั้นตอนให้มีการใช้งานที่สะดวกมากยิ่งขึ้นโดยใช้เครื่องมือเจเกินส์ที่กล่าวไปในทฤษฎีที่ 2.2 โดยจะแสดงให้เห็นในแผนภาพดังรูปที่ 3.2 โดยจะแสดงถึงการทำงานระบบงานใหม่ ซึ่งมีองค์ประกอบต่างๆ คือ

1. BKK Jenkins คือ เครื่องมือเจเกินส์ที่ดำเนินงานอยู่ที่ประเทศไทยโดยจะทำงานแทนผู้ใช้งาน
2. NOTT Jenkins คือ เครื่องมือเจเกินส์ที่ดำเนินงานอยู่ที่ประเทศอังกฤษโดยจะทำงานแทนผู้ใช้งาน
3. X86 คือ เครื่องเซิร์ฟเวอร์ที่เป็นระบบปฏิบัติการวินโดวส์ โดยเป็นเครื่องที่ใช้สำหรับคอมไพล์โค้ด
4. Linux คือ เครื่องเซิร์ฟเวอร์ที่เป็นระบบปฏิบัติการลินุกซ์ โดยเป็นเครื่องที่ใช้สำหรับคอมไพล์โค้ด
5. Web base คือ เครื่องเซิร์ฟเวอร์ที่ไว้สำหรับเก็บซอฟต์แวร์ซึ่งสามารถเรียกใช้ผ่าน Web browser
6. UAT คือ เครื่องเซิร์ฟเวอร์ที่มีสภาพแวดล้อมที่เหมือนหรือใกล้เคียงมากที่สุด กับสภาพแวดล้อมของการใช้งาน เพื่อให้การทดสอบใกล้เคียงกับการทำงานจริงมากที่สุด และได้ผลการทดสอบที่น่าเชื่อถือ ในขั้นนี้ให้เป็นเครื่องผู้ใช้งาน
7. Auto Test คือ เครื่องเซิร์ฟเวอร์ที่ใช้สำหรับทดสอบเกี่ยวกับ Automate Test



รูปที่ 3.2 แผนภาพ Sequence Diagram แสดงการทำงานของระบบใหม่

3.1 การพัฒนากระบวนการคอมไพล์ระบบจำลองแลกเปลี่ยนเงินตราต่างประเทศ

โดยในกระบวนการการคอมไพล์ระบบจำลองแลกเปลี่ยนเงินตราต่างประเทศในหัวข้อที่ 2.8 นั้น เป็นการทำงานในกระบวนการเดิม ผู้จัดทำได้ทำการพัฒนาโดยใช้เครื่องมือเจเนกินส์ ในทฤษฎีที่ 2.2 และแนวคิดการบูรณาการอย่างต่อเนื่องในทฤษฎีที่ 2.1 และทำการทำการเขียนสคริปต์เพิ่มเติมจากส่วนขยายของเจเนกินส์ในทฤษฎีที่ 2.2.2.2 และ 2.2.2.3 เพื่อให้ครอบคลุมการทำงานของระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศให้เป็นระบบอัตโนมัติขึ้น เพื่อลดการเสียเวลาในการทำงาน และทำให้สามารถดำเนินงานเป็นไปได้อย่างราบรื่น และสามารถได้แฟ้มเอกสารที่สามารถใช้งานได้ทันที

3.1.1 ขั้นตอนการพัฒนาและการวางแผน

ในการเขียนสคริปต์เพิ่มเติมจากส่วนขยายของเจเนกินส์เพื่อคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศนี้ ได้มีการวางแผนและเตรียมการดังนี้

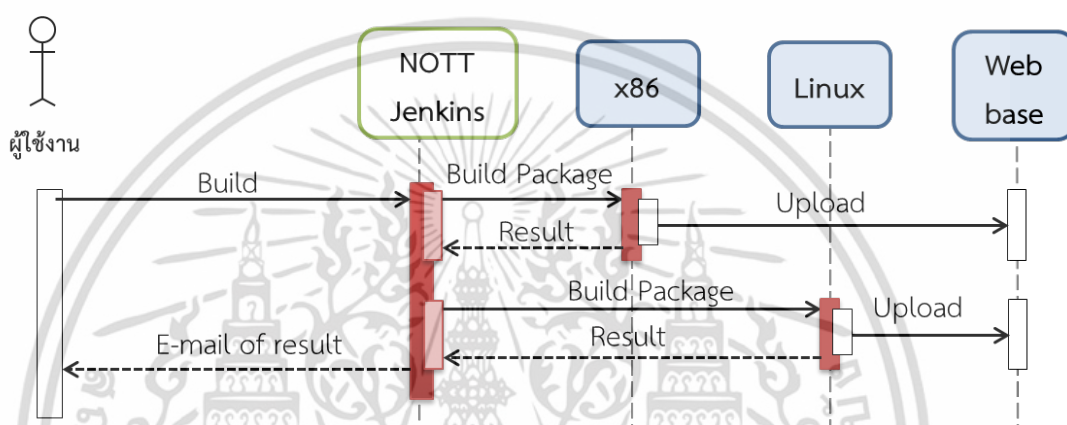
1. ศึกษาความเป็นไปได้ในการทำงานและเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับการจัดทำโครงการในครั้งนี้
2. ศึกษาข้อมูลเกี่ยวกับภาษาเชลล์สคริปต์ในการจัดทำโครงการนี้ ที่เลือกภาษานี้มาใช้งาน เพราะว่า มีความเห็นว่าจะต้องทำงานบนระบบปฏิบัติการลินุกซ์
3. ศึกษาขั้นตอนการทำงานของระบบเดิมของการคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ เพื่อนำไปพัฒนาระบบงานใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ขั้นตอนการวิเคราะห์ระบบ

- วิเคราะห์ระบบงานใหม่

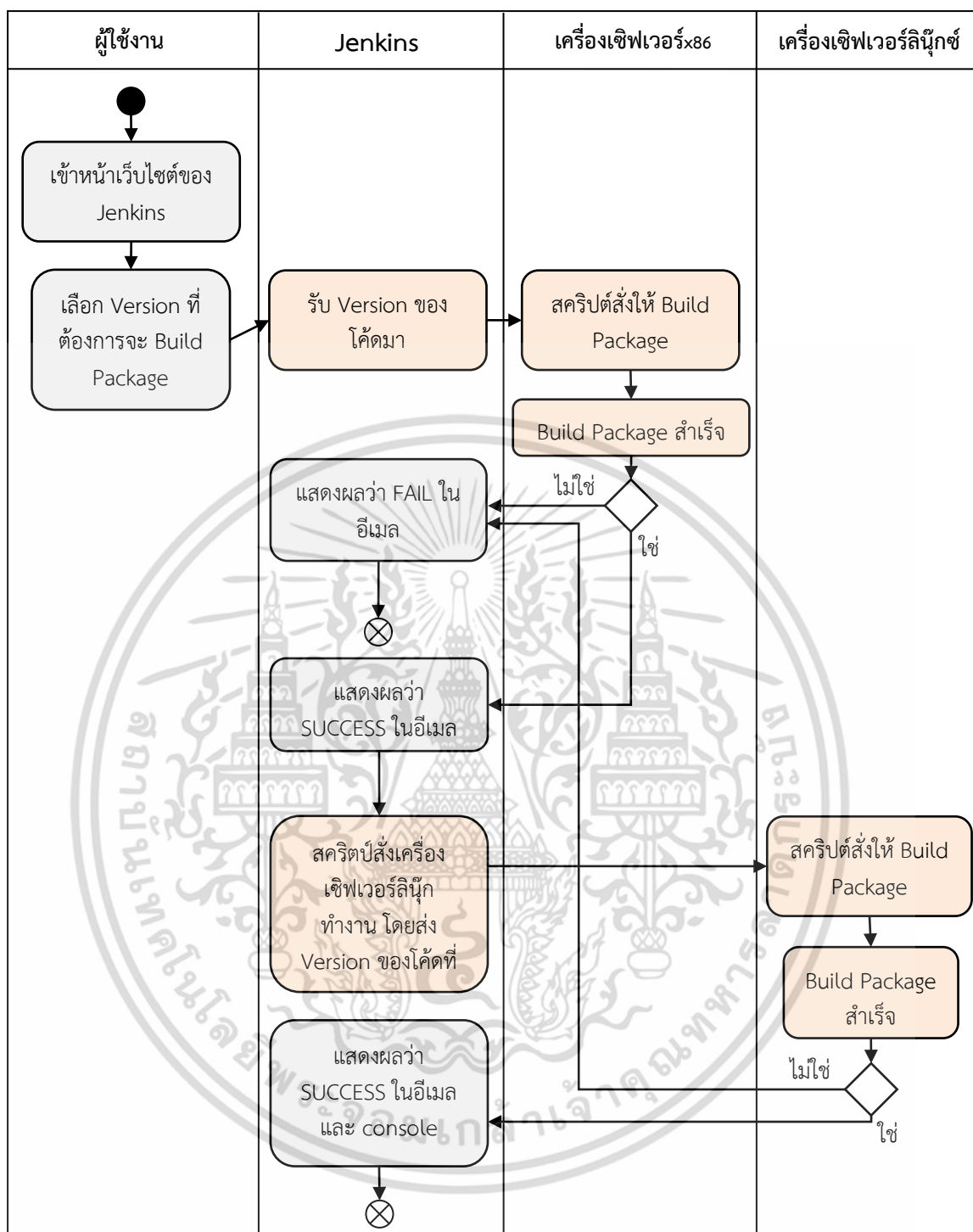
โดยระบบงานใหม่ ทางผู้จัดทำได้นำทฤษฎีที่ 2.1 มาเป็นแนวคิดในการดำเนินการระบบงานใหม่ โดยใช้เครื่องมือเจกินส์เป็นเครื่องมือหลังในการการทำงาน โดยทางผู้จัดทำได้ทำการเขียนสคริปต์เพิ่มเติมในส่วนของส่วนเพิ่มขยายของเครื่องมือเจกินส์ในทฤษฎีที่ 2.2.2.2 โดยสคริปต์ที่ทางผู้จัดทำได้เขียนเพิ่มเติมขึ้นมาเพื่อทำการช่วยในการคอมไพล์ของระบบ โดยเริ่มทำงานจากผู้ใช้ได้ทำการดำเนินการกับเครื่องมือเจกินส์ โดยสคริปต์จะรับเลข Version จากเครื่องมือเจกินส์มา แล้วเริ่มกระบวนการคอมไพล์ซอฟต์แวร์ หากทำงานสำเร็จหรือไม่สำเร็จจะทำการแจ้งข้อความบอกผู้ใช้ทางอีเมลล์



รูปที่ 3.3 แผนภาพ Sequence Diagram แสดงการทำงานของคอมไพล์ระบบการจำลองการแลกเปลี่ยนเงินตราต่างประเทศแบบใหม่

3.1.3 ขั้นตอนการออกแบบระบบ

3.1.3.1 แผนภาพกิจกรรม ในการพัฒนาการคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ โดยใช้เครื่องมือเจกินส์และการเขียนสคริปต์เพิ่มเติม จะแสดงขั้นตอนการทำงานต่างๆ ในการทำงานในระบบตามขั้นตอนในรูปที่ 3.4



รูปที่ 3.4 แผนภาพกิจกรรมของการพัฒนาการคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศ (กล่องสีเทา คือการทำงานของเครื่องมือเจเนกินส์ และ กล่องสีส้ม คือส่วนการทำงานของสคริปต์ส่วนเพิ่มขยาย)

จากรูปที่ 3.4 แผนภาพกิจกรรมของการเขียนสคริปต์เพิ่มเติมเพื่อพัฒนาการคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศนั้น เริ่มจากผู้ใช้งานได้ทำการเข้าสู่เว็บไซต์ของเจเนกินส์ และเลือกเวอร์ชัน ที่ต้องการจะให้ทำงาน จากนั้นเครื่องเจเนกินส์จะทำการเรียกสคริปต์ที่เขียนเพิ่มเติม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาทำงานเพื่อคอมไพล์ระบบ เมื่อการทำงานนั้นสิ้นสุด หากได้ผลลัพธ์ที่สำเร็จจะทำการแสดงผลออกทางอีเมลและส่งการทำงานต่อไปยังงานต่อไป ซึ่งคือการเรียกการทำงานของเครื่องเซิร์ฟเวอร์ลินุกซ์ และทำงานตามนั้นเป็นไปตามขั้นตอนข้างต้น หากไม่สำเร็จจะทำการแสดงผลทางอีเมลและจะจบการทำงานทันที โดยมีการเขียนสคริปต์เพิ่มเติมดังนี้

- Jenkins คือ ส่วนที่ได้เขียนสคริปต์เพิ่มเติมจากส่วนเพิ่มขยายของเจเนกินส์
- เครื่องเซิร์ฟเวอร์ x86 คือ ส่วนที่ได้เขียนสคริปต์ในส่วนของการคอมไพล์ระบบในระบบปฏิบัติการวินโดวส์
- เครื่องเซิร์ฟเวอร์ลินุกซ์ คือ ส่วนที่ได้เขียนสคริปต์ในส่วนของการคอมไพล์ระบบในระบบปฏิบัติการลินุกซ์

3.1.3.2 การเปรียบเทียบการคอมไพล์ซอฟต์แวร์

ในส่วนนี้จะแสดงความแตกต่างระหว่างการจัดการแบบเดิมและการจัดการแบบใหม่ โดยใช้เครื่องมือเจเนกินส์ แสดงดังตารางที่ 3.1

ตารางที่ 3.1 ตารางเปรียบเทียบการจัดการการคอมไพล์ซอฟต์แวร์

กระบวนการทำงาน	การจัดการแบบเดิม	การจัดการแบบใหม่
1) หน้าจอเมนูหลัก	- ไม่มีหน้าจอเมนูหลัก - หากต้องการดูเวอร์ชันของซอฟต์แวร์ต้องทำการเรียกใช้ชุดคำสั่ง (Command Line)	- แสดงหน้า Web browser ของเจเนกินส์ - แสดงเวอร์ชันของซอฟต์แวร์ผ่านหน้า Web browser
2) แสดงรายละเอียดงาน	- ไม่มีการแสดงรายละเอียดของซอฟต์แวร์	- แสดงรายละเอียดของซอฟต์แวร์ผ่านหน้า Web browser ของเจเนกินส์
3) การตั้งค่า	- ต้องตั้งค่าผ่านชุดคำสั่ง (Command line)	- สามารถตั้งค่าผ่านหน้า web browser ของเจเนกินส์
4) แสดงผลการทดสอบ	- ผลการคอมไพล์จะถูกเก็บไว้ในไฟล์ Log ต้องทำการใช้ชุดคำสั่งในการเรียกดู	- แจ้งผลการคอมไพล์ผ่านทางอีเมลและ Web browser ของเจเนกินส์

3.1.4 ขั้นตอนการพัฒนา

จากกระบวนการข้างต้นที่กล่าวมาทำให้เห็นถึงขั้นตอนต่างๆ ในการเขียนสคริปต์เพิ่มเติมเพื่อพัฒนาการคอมไพล์ระบบจำลองการแลกเปลี่ยนอัตราเงินระหว่างประเทศ ซึ่งการคอมไพล์ระบบนั้นต้องใช้เครื่องมือในของกระบวนการบูรณาการอย่างต่อเนื่อง คือ Jenkins และต้องใช้ภาษาเชลล์สคริปต์ในการเขียนสคริปต์เพื่อเรียกใช้คำสั่งในการ Build Package อีกด้วย

3.2 การพัฒนากระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศ

ในการกระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศแบบเดิมที่กล่าวไว้ในในหัวข้อที่ 2.9 นั้น ทางผู้จัดทำทำการพัฒนากระบวนการให้เป็นกระบวนการทางอัตโนมัติโดยใช้เครื่องมือเงินกิ้นส์ในทฤษฎีที่ 2.2 ในการเขียนสคริปต์เพิ่มเติมจากส่วนเพิ่มขยายของเงินกิ้นส์ในทฤษฎีที่ 2.2.2.3 เพื่ออัปเดตระบบการจำลองการแลกเปลี่ยนเงินตราต่างประเทศให้เป็นระบบอัตโนมัตินั้น เพื่อความสะดวกของผู้พัฒนาระบบในการอัปเดตระบบ อีกทั้งยังช่วยลดเวลาในการทำงาน และสามารถใช้งานระบบได้ทันที

3.2.1 ขั้นตอนการวางแผนและเตรียมการ

ในการดำเนินการจัดทำโครงการการเขียนสคริปต์เพิ่มเติมจากส่วนเพิ่มขยายของเงินกิ้นส์เพื่อคอมไพล์ระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศนี้ ได้มีการวางแผนและเตรียมการดังนี้

- 1) ศึกษาความเป็นไปได้ของการพัฒนาและเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับการจัดทำโครงการในครั้งนี้
- 2) ศึกษาข้อมูลเกี่ยวกับภาษาเซลล์สคริปต์ในการจัดทำโครงการนี้ ที่เลือกภาษานี้มาใช้งาน เพราะว่า มีความเห็นว่าจะต้องทำงานบนระบบปฏิบัติการลินุกซ์
- 3) ศึกษาขั้นตอนการทำงานของระบบเดิมของกระบวนการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ เพื่อนำไปพัฒนาระบบงานใหม่

3.2.2 ขั้นตอนการวิเคราะห์ระบบ

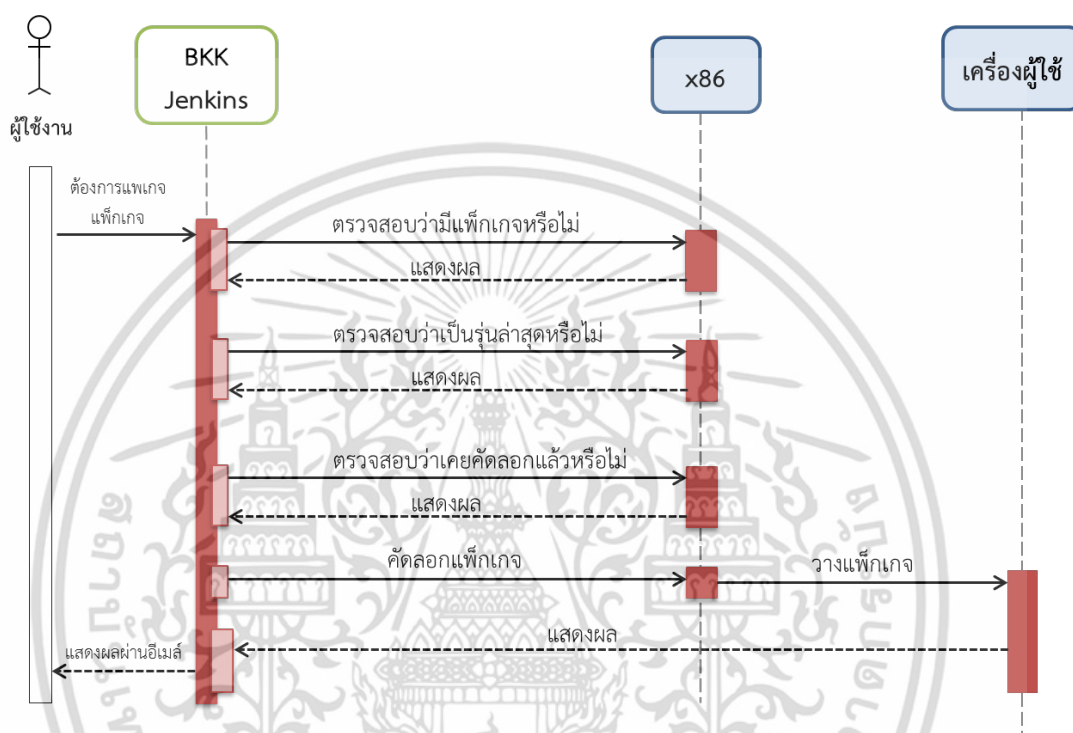
วิเคราะห์ระบบงานใหม่

การเขียนสคริปต์เพิ่มเติมจากส่วนเพิ่มขยายของเงินกิ้นส์ ในทฤษฎีที่ 2.2.2.2 และ 2.2.2.3 เพื่อพัฒนาระบบการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศให้เป็นระบบอัตโนมัติ นั้นทางผู้จัดทำได้เขียนขั้นตอนต่างๆ ในการพัฒนาระบบใหม่เป็น Flow Chart

ในการวิเคราะห์ระบบงานใหม่ ได้เอาแนวคิดของการบูรณาการอย่างต่อเนื่องมาเป็นตัวช่วยในพัฒนาระบบ โดยใช้แนวคิดของการบูรณาการอย่างต่อเนื่อง ในทฤษฎีที่ 2.1 และใช้เครื่องมือเงินกิ้นส์ในทฤษฎีที่ 2.2.2.3 มาเพื่อพัฒนากระบวนการแบบใหม่ โดยสคริปต์ที่เขียนเพิ่มเติมของการทำงานของเครื่องมือเงินกิ้นส์ได้ทำงานตามขั้นตอนดังนี้

- 1) รับเลข Version ของแพ็คเกจมาจากเครื่องมือเงินกิ้นส์
 - 2) ทำการตรวจสอบว่าเลข Version ของแพ็คเกจที่ได้รับมานั้นมีอยู่ในเครื่องเซิร์ฟเวอร์หรือไม่
 - 3) ทำการตรวจสอบว่าแพ็คเกจที่ต้องการนั้นเป็นรุ่นปัจจุบันหรือไม่
 - 4) ทำการตรวจสอบว่าแพ็คเกจที่ต้องการนั้นเคยทำการคัดลอกไว้ที่เครื่องเซิร์ฟเวอร์ของเงินกิ้นส์แล้วหรือไม่
 - 5) คัดลอกแพ็คเกจที่ต้องการไปที่เครื่องของผู้ใช้
- จะเห็นได้ว่า ระบบงานใหม่ที่ทำนั้นมีขั้นตอนการทำงานที่สะดวกและรวดเร็วกว่าการทำงานแบบระบบเก่า ในหัวข้อที่ 2.9 ประโยชน์ของการนำระบบอัตโนมัติมาใช้

- 1) เพิ่มประสิทธิภาพในการทำงาน และช่วยลดเวลาในการทำงานของผู้พัฒนา ระบบ
- 2) เพื่อเพิ่มความสะดวกในการทำงาน และช่วยเพิ่มความรวดเร็วในการทำงานให้มากยิ่งขึ้น
- 3) ลดความผิดพลาดในการทำงาน



รูปที่ 3.5 แผนภาพ Sequence Diagram แสดงการทำงานของสคริปต์การอัปเดตระบบการจำลอง การแลกเปลี่ยนเงินตราต่างประเทศแบบใหม่

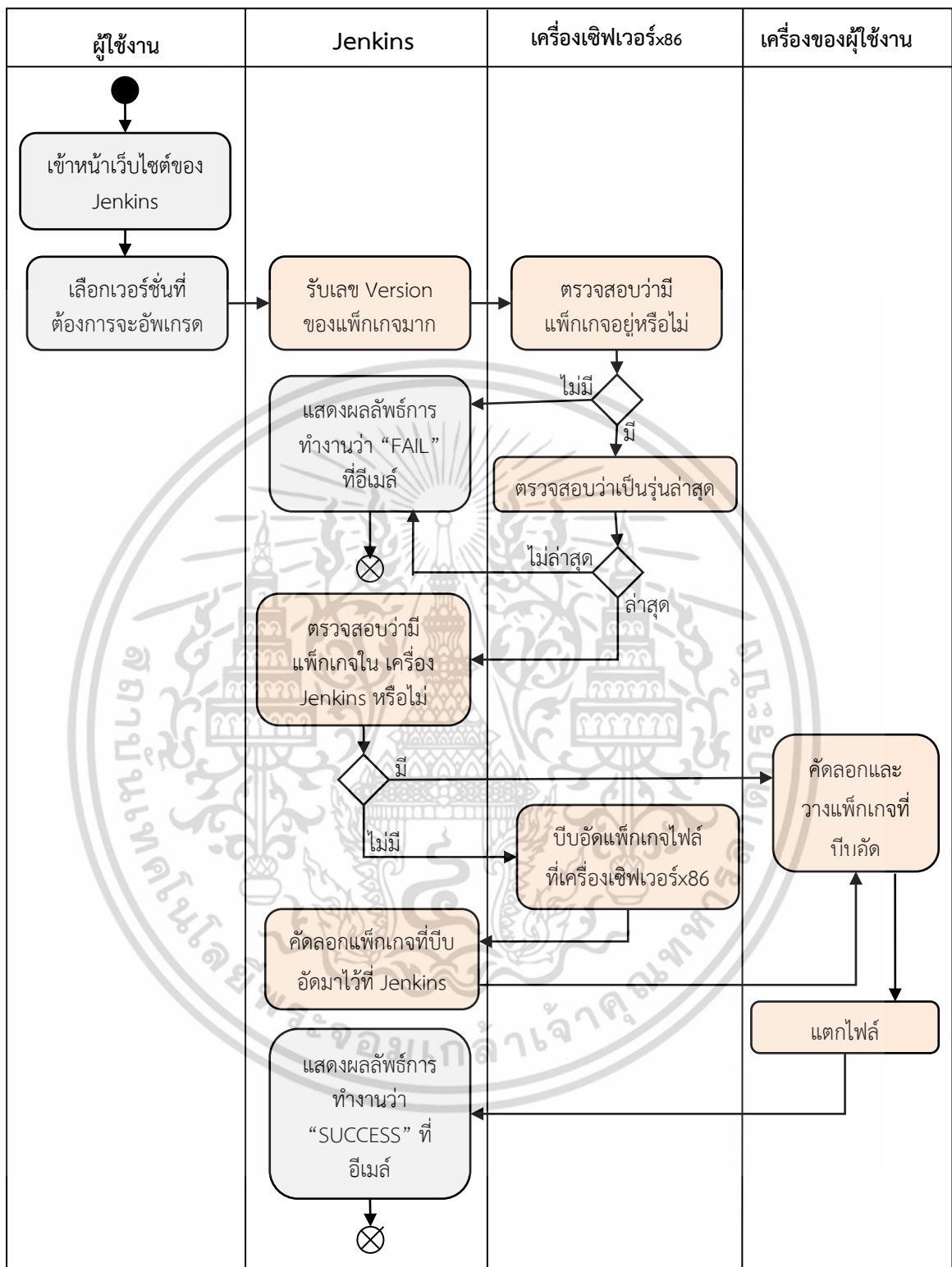
โดยจากรูปที่ 3.8 จะเห็นได้ว่า กระบวนการทำงานแบบใหม่นั้นจะใช้เครื่องมือเจเนกินส์และสคริปต์ที่เขียนเพิ่มเติมเพื่อตรวจสอบเงื่อนไข มาช่วยการทำงานแทนผู้ใช้โดยสามารถเปรียบเทียบได้จากรูปภาพที่ 2.25

3.2.3 ขั้นตอนการออกแบบระบบ

- แผนภาพกิจกรรม (Activity Diagram) ของการพัฒนาการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศ โดยรวมเครื่องมือเจเนกินส์ และสคริปต์เพิ่มเติมส่วนขยายของเจเนกินส์ โดยสคริปต์ของการทำงานมีฟังก์ชันต่างๆ ดังนี้

- การตรวจสอบการมีอยู่ของแพ็คเกจไฟล์
- การตรวจสอบแพ็คเกจไฟล์ว่าเป็นรุ่นปัจจุบัน
- การตรวจสอบว่ามีแพ็คเกจไฟล์ในเครื่องเจเนกินส์
- การบีบอัดแพ็คเกจไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ การคัดลอกแพ็คเกจไฟล์ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แผนภาพกิจกรรมของสคริปต์เพิ่มเติมจากส่วนเพิ่มขยายของเจเนกินส์พัฒนาการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศ (กล่องสีเทา คือการทำงานของเครื่องมือเจกินส์ และ กล่องสีส้ม คือส่วนของการทำงานของสคริปต์ส่วนเพิ่มขยาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.6 แผนภาพกิจกรรมของกระบวนการเพื่อพัฒนาการอัปเดตระบบจำลอง แลกเปลี่ยนเงินตราต่างประเทศ โดยสคริปต์จะทำการตรวจสอบเครื่องเซิร์ฟเวอร์ x86 ว่ามีแพ็กเกจ เวอร์ชันที่ต้องการอยู่หรือไม่ หากมีจะทำการตรวจสอบต่อว่าแพ็กเกจที่ต้องการนั้นเป็นรุ่นล่าสุดหรือไม่ หากใช่จะทำการตรวจสอบที่เครื่องเจเนกินส์ว่ามีแพ็กเกจรุ่นล่าสุดหรือไม่ หากมีจะทำการ คัดลอกแพ็กเกจรุ่นล่าสุดมาไว้ที่เครื่องของผู้ใช้งาน จากนั้นคัดลอกมาไว้ที่เครื่องของผู้ใช้งาน เมื่อ คัดลอกแพ็กเกจล่าสุดมาไว้ที่เครื่องของผู้ใช้งานได้แล้ว จะทำการติดตั้งที่เครื่องของผู้ใช้งาน เมื่อสิ้นสุด การทำงานจะแสดงผลการทำงานโดยส่งผลลัพธ์ผ่านทางอีเมล โดยจะทำการเขียนสคริปต์เพิ่มเติมดังนี้

- Jenkins คือ ส่วนที่ได้เขียนสคริปต์เพิ่มเติมจากส่วนเพิ่มขยายของเจเนกินส์ ได้แก่ การตรวจสอบเวอร์ชัน การคัดลอกแพ็กเกจ
- เครื่องเซิร์ฟเวอร์ x86 คือ ส่วนที่ได้เขียนสคริปต์ในส่วนของเจเนกินส์ ได้แก่ การบีบอัด และแตกไฟล์
- เครื่องของผู้ใช้งาน คือ ส่วนที่คัดลอกแพ็กเกจจากเซิร์ฟเวอร์วางไว้ที่เครื่องผู้ใช้งาน

3.2.3.1 การเปรียบเทียบการอัปเดตซอฟต์แวร์

ในส่วนนี้จะแสดงความแตกต่างระหว่างการจัดการแบบเดิมและการจัดการ แบบใหม่โดยใช้เครื่องมือเจเนกินส์ แสดงดังตารางที่ 3.2

ตารางที่ 3.2 ตารางเปรียบเทียบการจัดการการอัปเดตซอฟต์แวร์

กระบวนการทำงาน	การจัดการแบบเดิม	การจัดการแบบใหม่
1) หน้าจอเมนูหลัก	- ไม่มีหน้าจอเมนูหลัก - หากต้องการดูเวอร์ชันของซอฟต์แวร์ต้องทำการเรียกใช้ชุดคำสั่ง (Command Line)	- แสดงหน้า Web browser ของเจเนกินส์ - แสดงเวอร์ชันของซอฟต์แวร์ผ่านหน้า Web browser
2) แสดงรายละเอียดงาน	- ไม่มีการแสดงรายละเอียดของซอฟต์แวร์	- แสดงรายละเอียดของซอฟต์แวร์ผ่านหน้า Web browser ของเจเนกินส์
3) การตั้งค่า	- ต้องตั้งค่าผ่านชุดคำสั่ง (Command line)	- สามารถตั้งค่าผ่านหน้า web browser ของเจเนกินส์ - สามารถตรวจสอบเงื่อนไขก่อนการอัปเดตได้ผ่านทางเจเนกินส์
4) แสดงผลการทดสอบ	- ไม่มีรายงานผลของการอัปเดต	- แจ้งผลการอัปเดตผ่านทางอีเมลและ Web browser ของเจเนกินส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

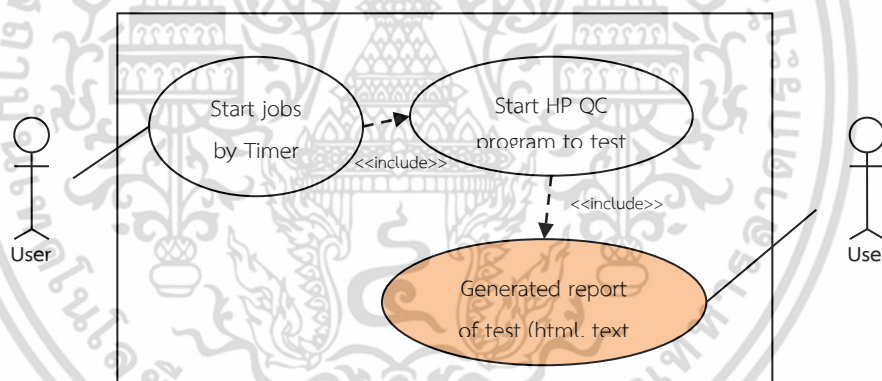
3.2.4 ขั้นตอนการพัฒนาระบบ

จากกระบวนการการทำงานระบบงานข้างต้นที่กล่าวมาทำให้เห็นถึง ขั้นตอนต่างๆ ในการเขียนสคริปต์เพิ่มเติมเพื่ออัปเดตระบบจำลองการแลกเปลี่ยนอัตราเงินระหว่างประเทศ ซึ่งการอัปเดตระบบนั้นต้องใช้เครื่องมือในของกระบวนการบูรณาการอย่างต่อเนื่อง คือ Jenkins เพื่อในการช่วยให้การทำงานเป็นระบบอัตโนมัติ เพื่อช่วยลดความซับซ้อนและเวลาในการทำงาน เนื่องจากระบบเครื่องเซิร์ฟเวอร์นั้นเป็นระบบปฏิบัติการโซลาริส (Solaris) ทำให้ต้องเขียนชุดคำสั่งในภาษา Shell script ในการทำตามขั้นตอนต่างๆ

3.3 การปรับปรุงการแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์

โดยการแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์แบบเดิม ในหัวข้อที่ 2.10 นั้นผู้ใช้สามารถเข้าถึงผลการทดสอบได้ยาก ผู้จัดทำจึงทำการปรับปรุงการแสดงผลของการทดสอบ โดยทำการเขียนสคริปต์เพิ่มเติม เพื่อทำการแสดงผลในรูปแบบข้อความถึงผู้ใช้โดยตรงได้ผ่านทางอีเมล

โดยทำการแก้ฟังก์ชันในยูสเคส Generated report of test โดยจะทำการสร้างไฟล์ HTML และ Text File เพื่อส่งกลับมาแสดงผลในผู้ใช้ผ่านทางอีเมล โดยเขียนเป็น Use case ดังรูปที่ 3.9



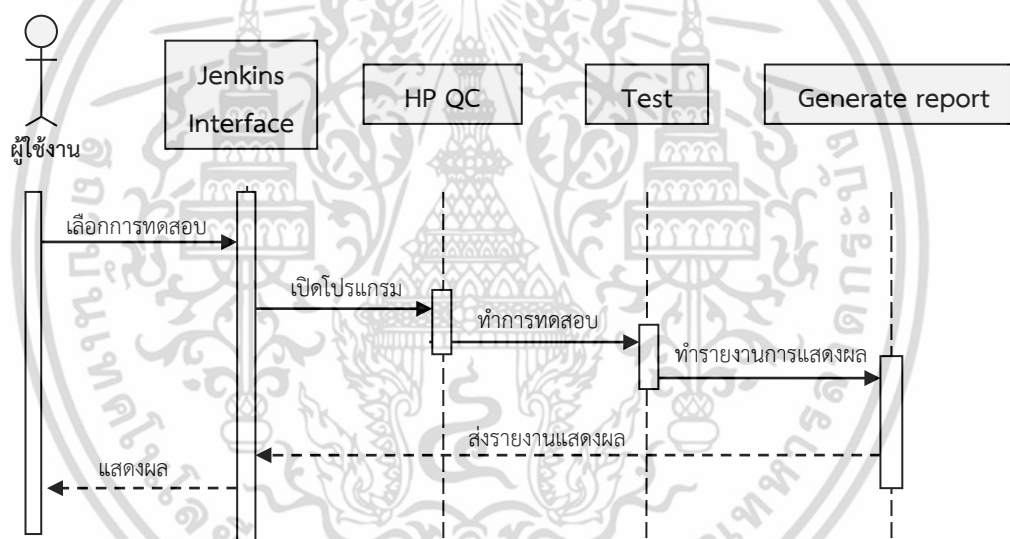
รูปที่ 3.7 แผนภาพยูสเคสของระบบการควบคุมคุณภาพซอฟต์แวร์ เมื่อมีการแก้ไขยูสเคส Generated report of test

จากรูปที่ 3.7 ยูสเคส generated report of test (html, text file) ออกแบบโดยให้ถูกเรียกใช้ทุกครั้งเมื่อยูสเคส Start HP QC program to test ถูกเรียกโดยยูสเคส Start Job ที่ถูกเรียกด้วยยูสเคสผู้ใช้งาน ผ่านการกดปุ่ม Build Now ของเครื่องมือเจเนกิ้นส์ ดังนั้น ยูสเคส generated report of test (html, text file) มีความสัมพันธ์แบบเรียกใช้งาน (Include) กับยูสเคส Start HP ALM program to test

ตารางที่ 3.3 อธิบายยูสเคส Generated report of test (html, text file)

Use Case Name :	Generated report of test (html, text file)
Scenario :	โปรแกรมทำการสร้างรายงานผลลัพธ์
Trigger Events :	โปรแกรมต้องทำงานเสร็จสิ้นและทำการสร้างรายงานผลลัพธ์
Primary Actor :	ผู้ใช้งาน
Preconditions :	โปรแกรมทำการทดสอบเสร็จสิ้นตามเทสเคส
Postconditions :	โปรแกรมสร้างรายงานผลลัพธ์ของการทดสอบ
Flow of Events :	โปรแกรมทดสอบเสร็จและสร้างรายงานและส่งให้ User ผ่านทางอีเมล และ console ของเจเนกินส์เสร็จสิ้น

จากขั้นตอนในตารางที่ 3.3 สามารถเขียนแผนภาพแสดงลำดับเหตุการณ์ของยูสเคส Generated Report of test (html, text file) แสดงดังรูปที่ 3.8



รูปที่ 3.8 แผนภาพ Sequence Diagram ของการแสดงผลของระบบการควบคุมคุณภาพของซอฟต์แวร์

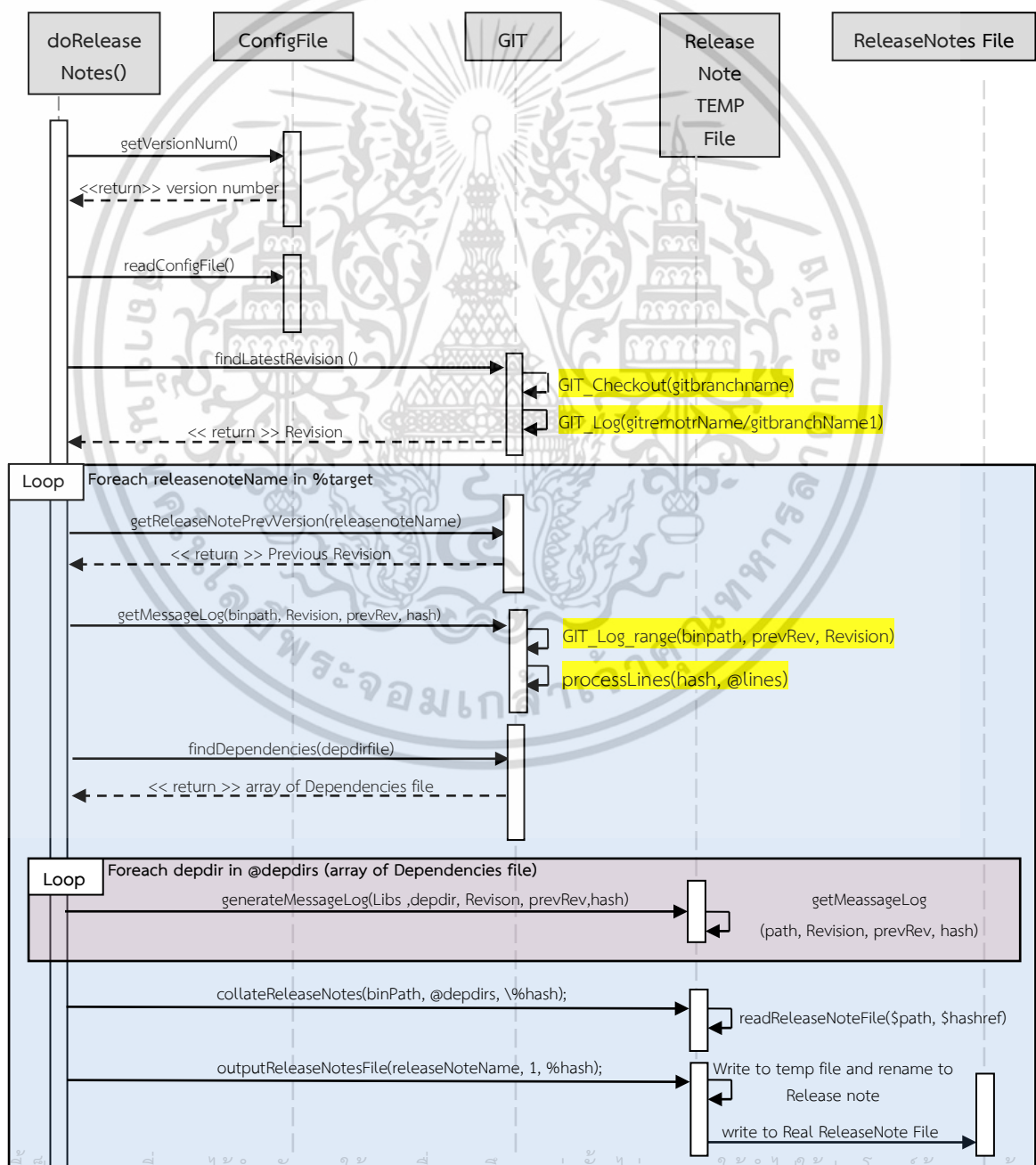
จากรูปที่ 3.8 แสดงให้เห็นถึงลำดับการทำงานของระบบการแสดงผลของการควบคุมคุณภาพซอฟต์แวร์ โดยเริ่มจากผู้ใช้งานได้กดปุ่ม “Build Now” ที่เครื่องมือเจเนกินส์ จากนั้นจะทำการไปที่เครื่องเซิร์ฟเวอร์ที่ทำการทดสอบ และทำการเปิดโปรแกรม HP Quality Center ขึ้นมา แล้วทำการดำเนินการทดสอบการควบคุมคุณภาพของซอฟต์แวร์ เมื่อทำการทดสอบเสร็จ สคริปต์จะทำการสร้างรายงานผลลัพธ์ของการทดสอบออกมาในรูปแบบของไฟล์ HTML และไฟล์ข้อความ เพื่อทำการแสดงผลให้ผู้ใช้งาน และทำการส่งกลับไปที่เครื่องมือเจเนกินส์ และทำการส่งอีเมลผลการทดสอบโดยใช้ส่วนเพิ่มขยายของเจเนกินส์ในทฤษฎีที่ 2.2.2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การปรับปรุงการแสดงผล Release notes ของซอฟต์แวร์

โดย Release Notes คือ สิ่งที่เราอธิบายการเปลี่ยนแปลงหรือแก้ไขต่างๆ ของซอฟต์แวร์ โดยข้อมูลนั้นต้องเป็นประโยชน์ต่อผู้ใช้งาน เมื่อทางบริษัทได้ทำการเปลี่ยนแปลงระบบควบคุมรุ่นจาก Subversion มาเป็น GIT ทำให้เกิดปัญหาในการสร้าง Release Note ขึ้นมา โดยไม่สามารถที่จะบันทึกการเปลี่ยนแปลงได้เมื่อได้ผลิตภัณฑ์ออกมา

หลังจากการวิเคราะห์ระบบการแสดงผลของ Release note นั้นในหัวข้อที่ 2.11 นั้น ทำให้ได้เข้าใจการทำงานของระบบมากยิ่งขึ้น โดยทางผู้จัดทำได้มีการปรับปรุงแก้ไขสคริปต์โดยการเปลี่ยนคำสั่งต่างๆในการเรียกข้อมูลจากระบบควบคุมรุ่น Subversion ให้เป็นการใช้คำสั่งของระบบควบคุมรุ่น GIT แทน โดยโครงสร้างของการทำงานของสคริปต์ยังคงเทียบเคียงกับสคริปต์เดิมอยู่ โดยสามารถเขียนเป็น Sequence Diagram ได้ดังรูปที่ 3.11



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นๆ
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 แผนภาพ Sequence Diagram ของการทำงานของ Release Notes โดย GIT

โดยได้มีการเพิ่มเติมและแก้ไขฟังก์ชันจากหัวข้อที่ 2.11 ดังนี้

- doReleaseNotes() คือ ฟังก์ชันในการสร้าง Release Note โดยค้นหาจากไฟล์ที่เก็บ Dependencies ของโค้ด
- getMessageLog() คือ ฟังก์ชันที่ทำการค้นหารายละเอียดการเปลี่ยนแปลงของโค้ดแล้วทำการเขียนลงไฟล์ Release note ของจริง
- doCommitReleaseNotes() คือ ฟังก์ชันเพื่อการ Commit ไฟล์ Release note ไปที่ Repository
- GIT_Checkout() คือ ฟังก์ชันการเรียกการแก้ไขครั้งล่าสุดจากการเปลี่ยนแปลงของโค้ด
- GIT_Log() คือ ฟังก์ชันการดึงรายละเอียดของการเปลี่ยนแปลงโค้ดของซอฟต์แวร์
- GIT_Log_range() คือ ฟังก์ชันการดึงรายละเอียดของการเปลี่ยนแปลงโค้ดโดยสามารถกำหนดช่วงของเลขรุ่นของการแก้ไขได้
- processLines() คือ ฟังก์ชันในการจัดบรรทัด และตัดคำที่ได้จาก ฟังก์ชัน GIT_Log()

บทที่ 4

ผลการดำเนินงานและการอภิปรายผล

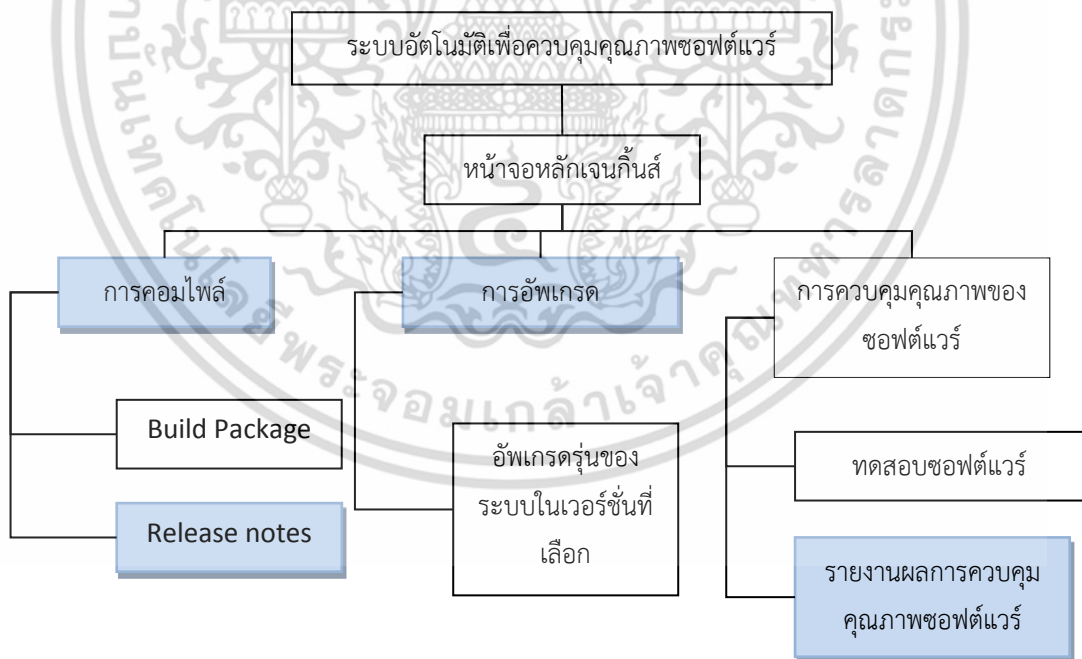
ในเนื้อหาบทที่ 4 จะกล่าวถึงการแสดงผลการดำเนินงานของการใช้เจเนนกินส์เพื่อจัดการคุณภาพของซอฟต์แวร์ ซึ่งการพัฒนาการออกแบบระบบนั้น ได้ถูกพัฒนาตามทีออกแบบไว้ในบทที่ 3 ซึ่งพัฒนาโดยเน้นการใช้ภาษา shell script และเครื่องมือเจเนนกินส์โดยจะทำการแสดงผลลัพธ์ของกระบวนการการทำงานระบบใหม่ และมีการเปรียบเทียบการทำงานของระบบการทำงาน สามารถแบ่งเป็นสองหัวข้อใหญ่ๆ ได้ดังนี้

หัวข้อที่ 4.1 ระบบอัตโนมัติเพื่อควบคุมคุณภาพซอฟต์แวร์นั้นประกอบไปด้วยโครงสร้างของระบบงาน

หัวข้อที่ 4.2 จะอธิบายถึงผลการดำเนินงานและวิธีการดำเนินงานจัดการเบื้องต้นของการใช้เจเนนกินส์เพื่อจัดการคุณภาพของซอฟต์แวร์ ซึ่งหัวข้อนี้ประกอบไปด้วยการแสดงผลการดำเนินงานการคอมไพล์ การอัปเดต การแสดงผลการทดสอบ และการแสดง Release Notes

4.1 โครงสร้างของระบบ

ระบบอัตโนมัติเพื่อควบคุมคุณภาพซอฟต์แวร์นั้นประกอบไปด้วยโครงสร้างของระบบงาน ดังรูปที่ 4.1



รูปที่ 4.1 โครงสร้างของระบบอัตโนมัติเพื่อควบคุมคุณภาพของซอฟต์แวร์

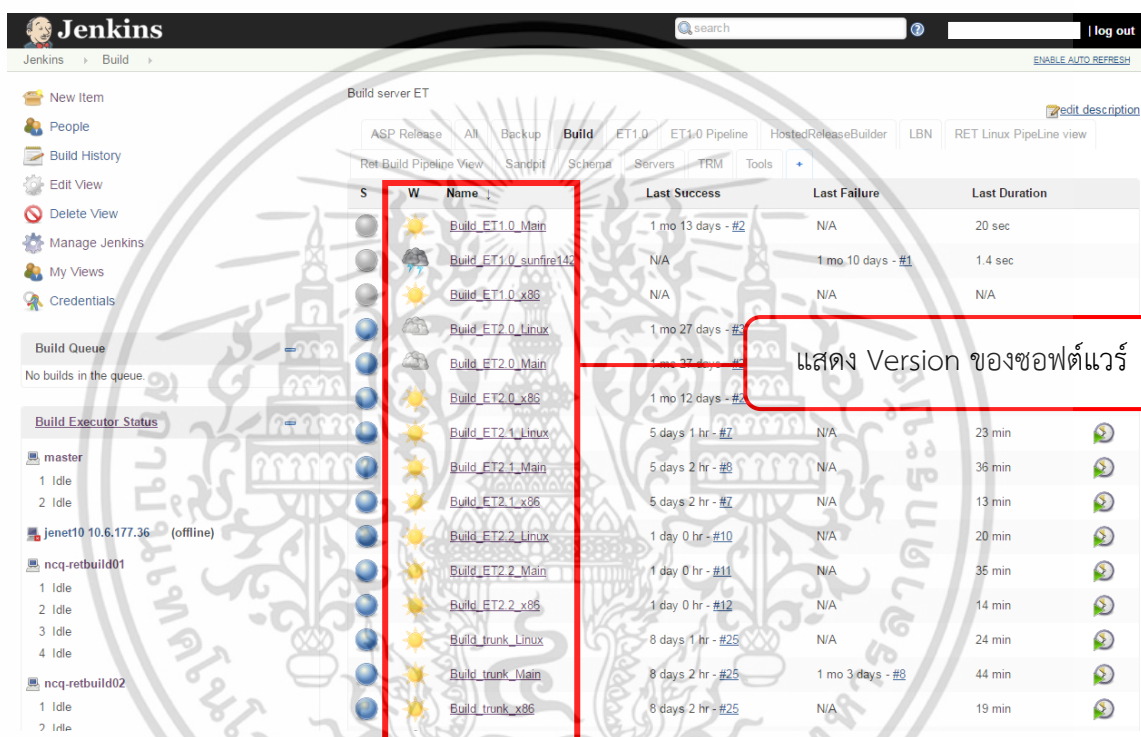
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการดำเนินงาน

4.2.1 ผลการดำเนินงานการพัฒนากระบวนการคอมไพล์ระบบจำลองแลกเปลี่ยนเงินตรา ระหว่างประเทศ

1) หน้าจอเมนูหลัก

เมื่อเข้าสู่ระบบเรียบร้อยแล้ว ก็จะแสดงหน้าจอเมนูหลัก ซึ่งจะแสดงงาน (Jobs) ต่างๆ ที่เกี่ยวข้องกับการ Build Package ในแต่ละเวอร์ชัน ดังรูปที่ 4.2



รูปที่ 4.2 หน้าจอเมนูหลักในการแสดงการ Build Package ในเวอร์ชันต่างๆ

2) หน้าจอแสดงรายละเอียดงาน (job) ของการ Build Package

เป็นหน้าจอที่แสดงรายละเอียดต่างๆ และเป็นหน้าที่สามารถสั่งเริ่มการทำงานของการ Build Package ได้ โดยคลิกที่ปุ่ม “Build Now” ดังรูปที่ 4.3

รูปที่ 4.3 หน้าจอแสดงรายละเอียดของงานในเวอร์ชันนั้น

3) หน้าจอแสดงการตั้งค่า

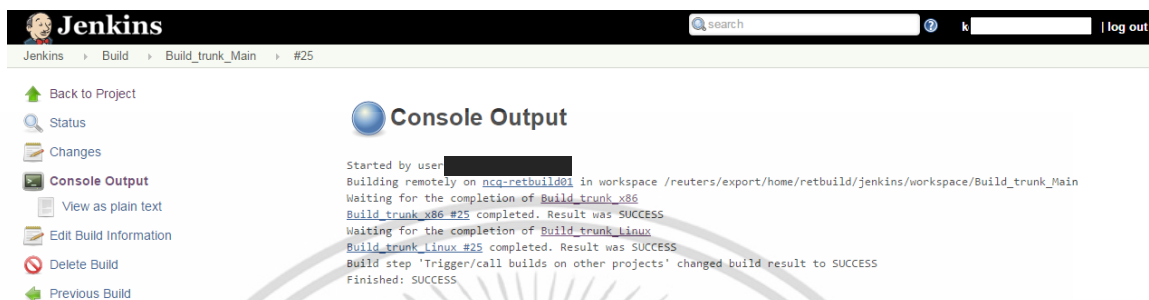
เป็นหน้าจอที่ไว้สำหรับการตั้งค่าของงานในแต่ละเวอร์ชัน โดยจะทำการตั้งค่าเพื่อเรียกใช้งานสคริปต์ในการ Build Package ของระบบ และตั้งค่าต่างๆ เกี่ยวกับการ Build Package ของระบบ อีกทั้งการส่งอีเมลแจ้งเตือนการทำงานด้วย ดังรูปที่ 4.4

รูปที่ 4.4 หน้าจอแสดงการตั้งค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) หน้าจอแสดง Console

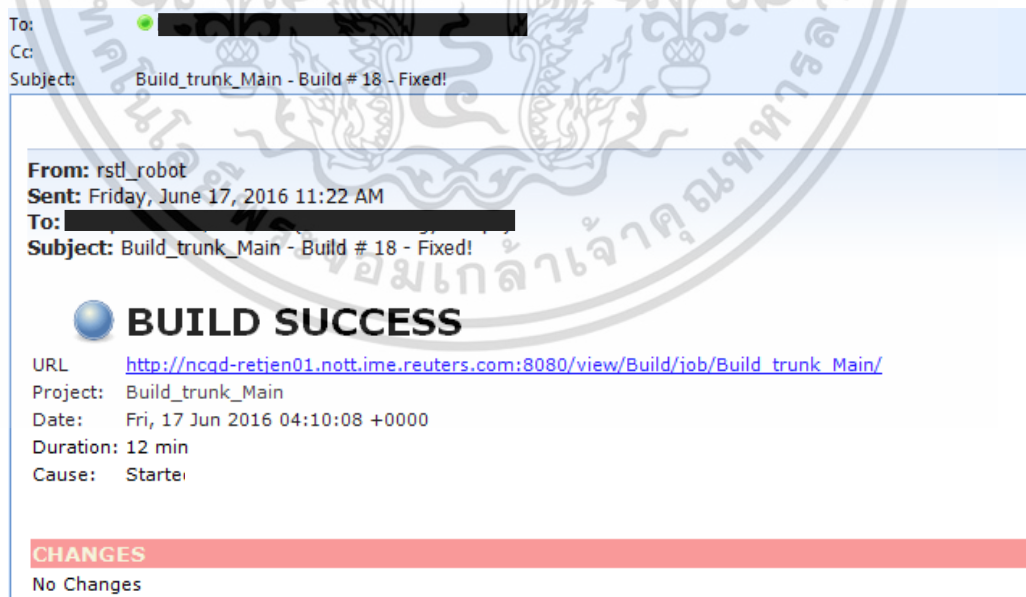
หน้าจอ Console เป็นหน้าจอที่ไว้แสดงการทำงานของ การ Build Package ตั้งเริ่มการทำงานจบจบการทำงาน โดยจะแสดงรายละเอียดการคอมไพล์ และจะแสดงผลว่าการคอมไพล์หรือไม่ โดยการทำงานของ ดังรูปที่ 4.5



รูปที่ 4.5 หน้าจอแสดง Console

5) อีเมลแจ้งเตือนผลการทำงาน

จะเป็นการแจ้งเตือนผลการให้ผู้ใช้งานทำงานว่า Success หรือว่า Failure และแสดงรายละเอียดการทำงานผ่านทางอีเมล โดยจะแสดงออกมาในดังรูปที่ 4.6



รูปที่ 4.6 อีเมลแจ้งเตือนผลการทำงาน

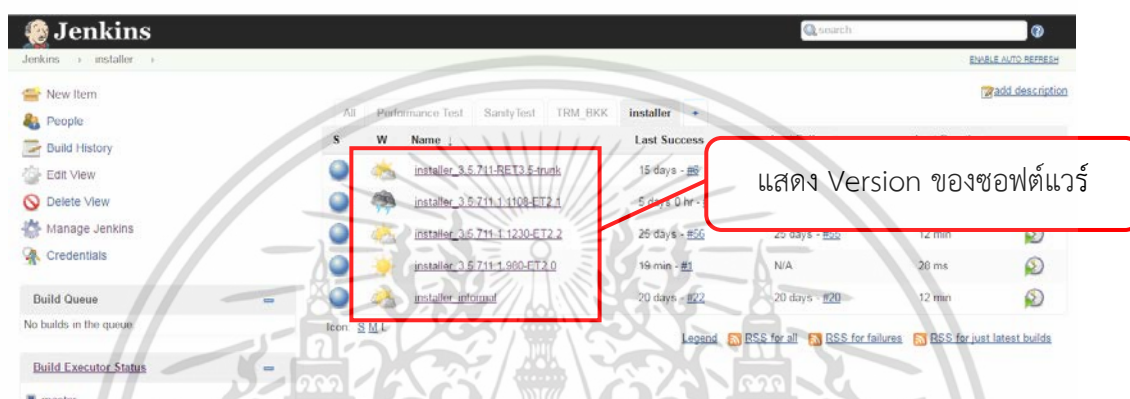
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการดำเนินงานการพัฒนากระบวนการอัปเดตระบบจำลองแลกเปลี่ยนเงินตรา ระหว่างประเทศ

จากการทำงานทั้งหมดของการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตรา ระหว่างประเทศนั้นสามารถอธิบายหลักการทำงานของระบบได้ดังนี้

1) หน้าจอเมนูหลัก

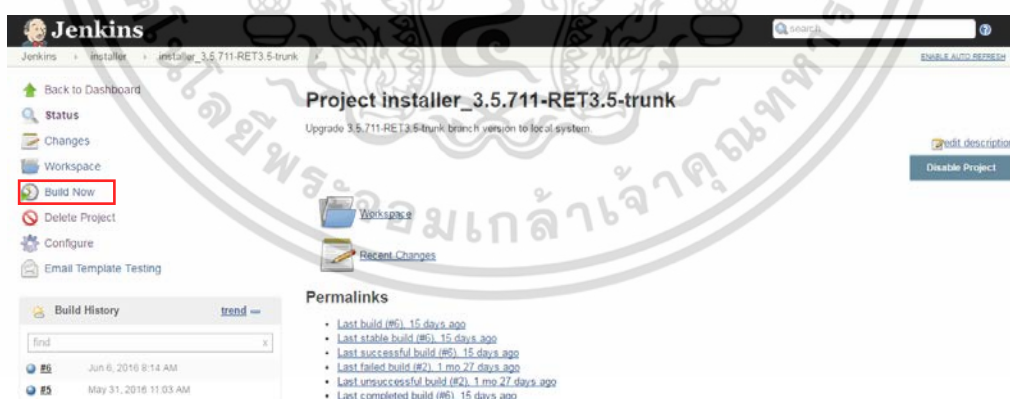
เมื่อเข้าสู่ระบบเรียบร้อยแล้ว ก็จะแสดงหน้าจอเมนูหลัก ซึ่งจะแสดงงาน (Jobs) ต่างๆ ที่เกี่ยวข้องกับกระบวนการอัปเดตระบบในแต่ละเวอร์ชัน ดังรูปที่ 4.7



รูปที่ 4.7 หน้าจอเมนูหลักในการแสดงการอัปเดตระบบในแต่ละเวอร์ชันต่างๆ

2) หน้าจอแสดงรายละเอียดแต่ละงาน (Jobs) ของการอัปเดตระบบ

เป็นหน้าจอที่แสดงรายละเอียดต่างๆ และเป็นหน้าที่สามารถสั่งเริ่มการทำงานของ การ Build Package ได้ โดยคลิกที่ปุ่ม “Build Now” ดังรูปที่ 4.8

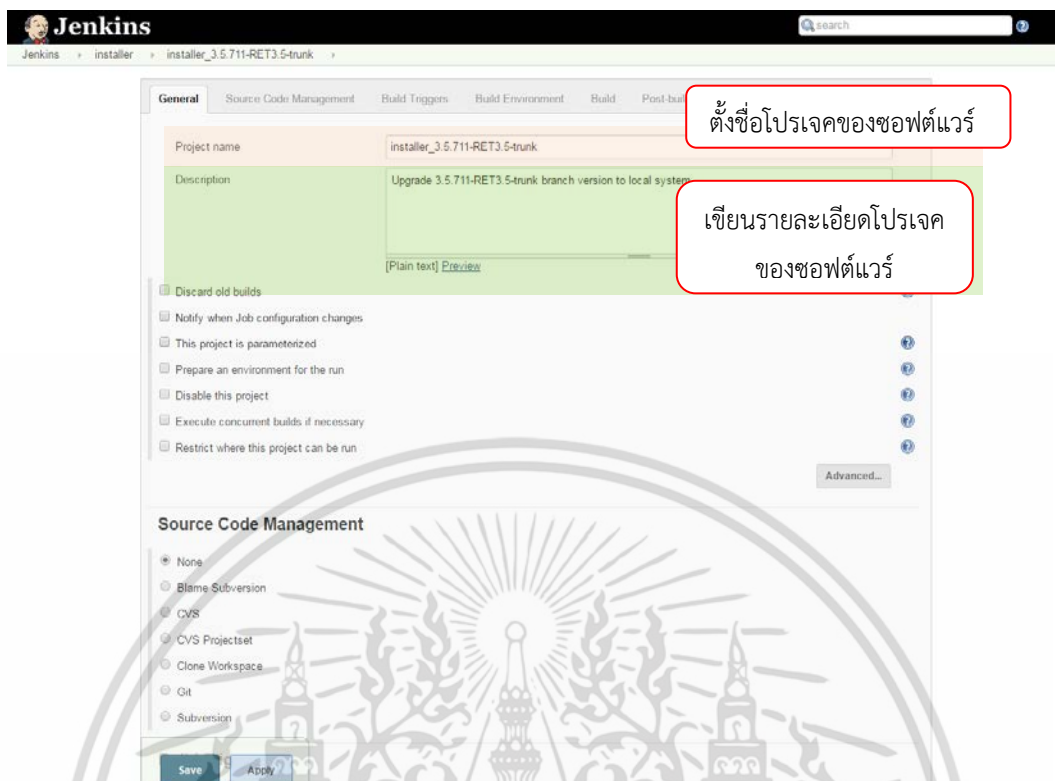


รูปที่ 4.8 หน้าจอแสดงรายละเอียดแต่ละงานของการอัปเดตระบบ

3) หน้าจอแสดงการตั้งค่า

เป็นหน้าจอที่ไว้สำหรับการตั้งค่าของงานในแต่ละเวอร์ชัน โดยจะทำการตั้งค่าเพื่อเรียกใช้งานสคริปต์ในการอัปเดตระบบ และตั้งค่าต่างๆ เกี่ยวกับการอัปเดตระบบ อีกทั้งการส่งอีเมลแจ้งเตือนการทำงานด้วย ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

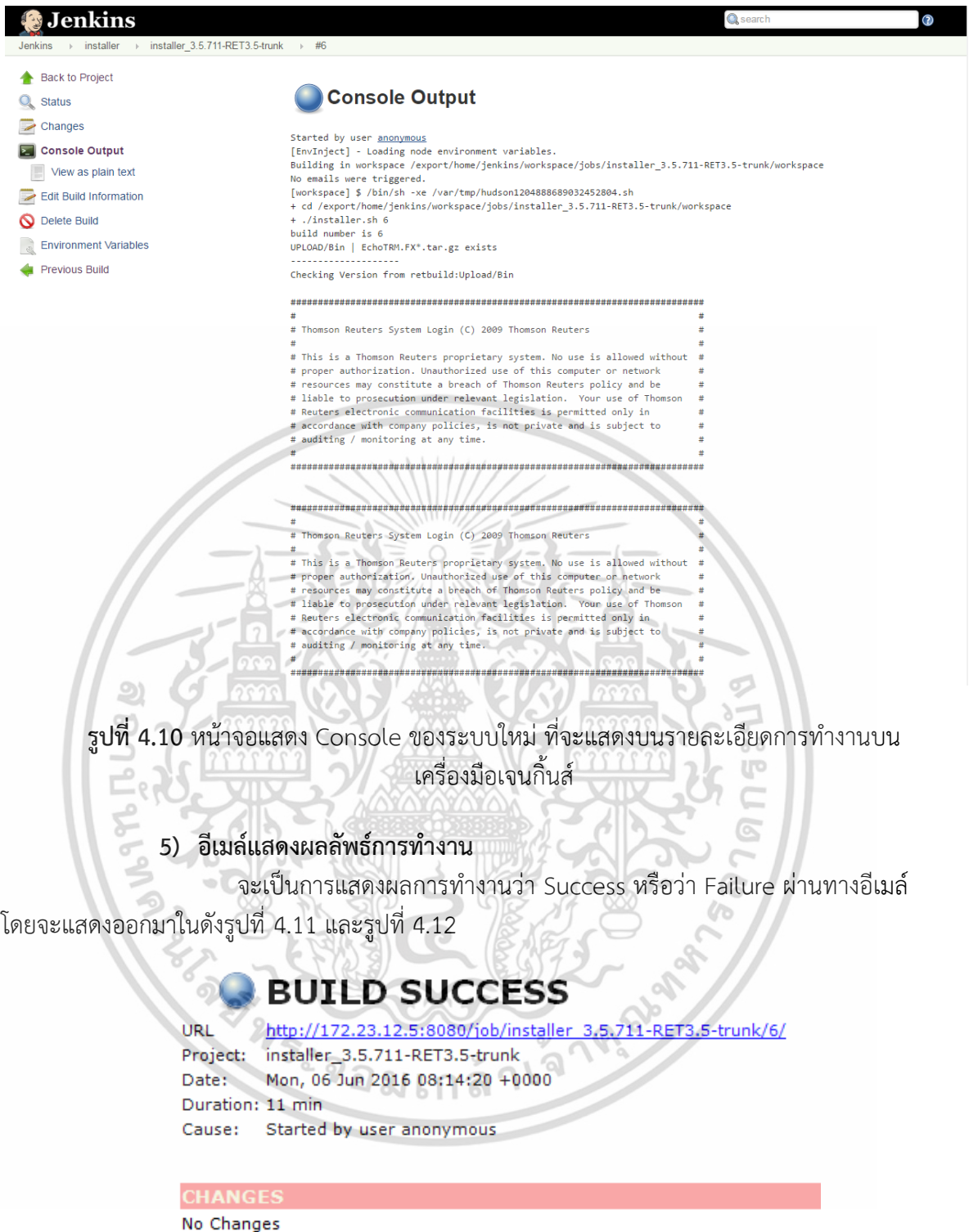


รูปที่ 4.9 หน้าจอแสดงการตั้งค่า

4) หน้าจอแสดง Console

หน้าจอ Console เป็นหน้าจอที่ไว้แสดงการทำงานของการทำงานของการอัปเดตระบบ ตั้งเริ่มการทำงานจนจบการทำงาน โดยจะแสดงรายละเอียดการอัปเดตระบบ และจะแสดงผลว่าการอัปเดตระบบสำเร็จหรือไม่ โดยการทำงานของ ดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Jenkins search

Jenkins > installer > installer_3.5.711-RET3.5-trunk > #6

[Back to Project](#)
[Status](#)
[Changes](#)
Console Output
[View as plain text](#)
[Edit Build Information](#)
[Delete Build](#)
[Environment Variables](#)
[Previous Build](#)

Console Output

Started by user [anonymous](#)
 [EnvInject] - Loading node environment variables.
 Building in workspace /export/home/jenkins/workspace/jobs/installer_3.5.711-RET3.5-trunk/workspace
 No emails were triggered.
 [workspace] \$ /bin/sh -xe /var/tmp/hudson1204888689032452804.sh
 + cd /export/home/jenkins/workspace/jobs/installer_3.5.711-RET3.5-trunk/workspace
 + ./installer.sh 6
 build number is 6
 UPLOAD/Bin | EchoTRM.FX*.tar.gz exists

 Checking Version from retbuild:Upload/Bin

```

#####
#
# Thomson Reuters System Login (C) 2009 Thomson Reuters
#
# This is a Thomson Reuters proprietary system. No use is allowed without
# proper authorization. Unauthorized use of this computer or network
# resources may constitute a breach of Thomson Reuters policy and be
# liable to prosecution under relevant legislation. Your use of Thomson
# Reuters electronic communication facilities is permitted only in
# accordance with company policies, is not private and is subject to
# auditing / monitoring at any time.
#
#####
#
# Thomson Reuters System Login (C) 2009 Thomson Reuters
#
# This is a Thomson Reuters proprietary system. No use is allowed without
# proper authorization. Unauthorized use of this computer or network
# resources may constitute a breach of Thomson Reuters policy and be
# liable to prosecution under relevant legislation. Your use of Thomson
# Reuters electronic communication facilities is permitted only in
# accordance with company policies, is not private and is subject to
# auditing / monitoring at any time.
#
#####

```

BUILD SUCCESS

URL: http://172.23.12.5:8080/job/installer_3.5.711-RET3.5-trunk/6/
 Project: installer_3.5.711-RET3.5-trunk
 Date: Mon, 06 Jun 2016 08:14:20 +0000
 Duration: 11 min
 Cause: Started by user anonymous

CHANGES

No Changes

รูปที่ 4.10 หน้าจอแสดง Console ของระบบใหม่ ที่จะแสดงบนรายละเอียดการทำงานบน เครื่องมือเจเนกินส์

5) อีเมลล์แสดงผลการทำงาน

จะเป็นการแสดงผลการทำงานว่า Success หรือว่า Failure ผ่านทางอีเมลล์ โดยจะแสดงออกมาในดังรูปที่ 4.11 และรูปที่ 4.12

รูปที่ 4.11 อีเมลล์แสดงผลการทำงานของการอัปเดตระบบว่าสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BUILD FAILURE

URL http://172.23.12.5:8080/job/installer_3.5.711.1.1108-ET2.1/17/
 Project: installer_3.5.711.1.1108-ET2.1
 Date: Fri, 17 Jun 2016 04:05:57 +0000
 Duration: 3.2 sec
 Cause: Started by user anonymous

CHANGES

No Changes

CONSOLE OUTPUT

```
Started by user anonymous
[EnvInject] - Loading node environment variables.
Building on master in workspace /export/home/jenkins/workspace/jc
No emails were triggered.
[workspace] $ /bin/sh -xe /var/tmp/hudson5283900860431141997.sh
+ cd /export/home/jenkins/workspace/jobs/installer_3.5.711.1.1108
+ ssh retbuild@10.6.185.206 hostname
```

รูปที่ 4.12 อีเมลแสดงผลการทำงานของการทำงานของการอัปเดตระบบว่าไม่สำเร็จ

4.2.3 ผลการดำเนินงานการปรับปรุงการแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์

จากการดำเนินงานทั้งหมดของปรับปรุงการแสดงผลการควบคุมคุณภาพของซอฟต์แวร์นั้นสามารถอธิบายผลงานการดำเนินงานของระบบได้ดังนี้

1) หน้าจอแสดงการทำงานของเครื่องมือ Jenkins

เป็นหน้าจอที่แสดงการทำงานและผลลัพธ์ของการทำงานของการควบคุมคุณภาพซอฟต์แวร์ โดยผลลัพธ์ของการทำงานจะแสดงออกมาที่หน้าจอนี้ ดังรูปที่ 4.13

[[Test Set Execution Summary]]

Server : <http://172.23.12.27/qcbin>
 Domain Name : RET_DOMAIN
 Project Name : RETAD
 TestSet : 20150129 - TRM_Replay_Set - Template_2015
 Test Set Folder : Root\Dev Sanity Test\ET2.0\Sanity Test\TRM Sanity\

 Blocked : 0
 Failed : 2
 N/A : 0
 No Run : 0
 Not Completed : 0
 Passed : 31

 ## Executed Tests ##
 #####

[STATUS]	TEST
[Passed]	[1]ResetSanityTest
[Passed]	[1]GetServerVersions
[Passed]	[1]Basic Pricing 1
[Passed]	[1]Basic Pricing 2
[Passed]	[1]Basic specific dates
[Passed]	[1]RoundFedSpot 1
[Passed]	[1]RoundFedSpot 2
[Passed]	[1]RoundFedSpot 3
[Passed]	[1]RoundFedSpot 4
[Passed]	[1]Round Spot for Fwd Pts 1
[Passed]	[1]Round Spot for Fwd Pts 2
[Passed]	[1]MinMax Spread
[Passed]	[1]R0 for Breakdown 1
[Passed]	[1]R0 for Breakdown 2
[Passed]	[1]Spot Tolerance
[Passed]	[1]FwdTolerance 1
[Passed]	[1]FwdTolerance 2
[Passed]	[1]Spot Throttle Staling
[Passed]	[1]InterpolatedStaling 1
[Passed]	[1]InterpolatedStaling 2
[Passed]	[1]Forward Throttle Staling
[Passed]	[1]Crossed Forward Min Movement
[Passed]	[1]Republishing
[Passed]	[1]CrossedSpotThrottling (Min Int. + R0)
[Passed]	[1]CrossedFwdThrottling (Direct spot)
[Passed]	[1]CrossedFwdThrottling (Crossed spot)
[Passed]	[1]DirectSpotMinMovement
[Passed]	[1]FwdMinSpotMovement

รูปที่ 4.13 หน้าจอแสดงผลของการควบคุมคุณภาพของซอฟต์แวร์ ผ่านทางหน้าจอ Console ของเครื่องมือ Jenkins

2) อีเมลล์แสดงผลลัพธ์ของการควบคุมคุณภาพของซอฟต์แวร์

จะเป็นการแสดงผลของการควบคุมคุณภาพของซอฟต์แวร์ โดยจะส่งผลลัพธ์ของการทดสอบมายังอีเมลล์ของผู้ใช้ระบบ ดังรูปที่ 4.14

Quality Center					
Test Set Execution Summary					
Server	http://172.23.12.27/qcbin				
Domain Name	RET_DOMAIN				
Project Name	RETAD				
TestSet	20150129 - TRM_Replay_Set - Template_2015				
Started	08/06/2016 05:02:10				
Finished	08/06/2016 07:35:25				
Duration	153 minutes 15 seconds				
Blocked	Failed	N/A	No Run	Not Completed	Passed
0	2	0	0	0	31
Test Set - 20150129 - TRM_Replay_Set - Template_2015					
Name:	20150129 - TRM_Replay_Set - Template_2015			Test Set ID	4
Test Set Folder	Root\Dev Sanity Test\ET2.0\Sanity Test\TRM Sanity\			Type	Default
Target Cycle				Baseline	
Executed Tests					
Test	Status	Responsible Tester	Tester	Exec Date	Exec Time
[1]ResetSanityTest	Passed		dev	08/06/2016	23:25:44
[1]GetServerVersions	Passed		dev	08/06/2016	23:26:08
[1]Basic Pricing 1	Passed		dev	08/06/2016	23:28:15
[1]Basic Pricing 2	Passed		dev	08/06/2016	23:30:23
[1]Basic specific dates	Passed		dev	08/06/2016	23:34:41
[1]RoundFedSpot 1	Passed		dev	08/06/2016	23:38:05
[1]RoundFedSpot 2	Passed		dev	08/06/2016	23:41:30
[1]RoundFedSpot 3	Passed		dev	08/06/2016	23:44:40
[1]RoundFedSpot 4	Passed		dev	08/06/2016	23:48:07
[1]Round Spot for Fwd Pts 1	Passed		dev	08/06/2016	23:49:54
[1]Round Spot for Fwd Pts 2	Passed		dev	08/06/2016	23:51:42
[1]MinMax Spread	Passed		dev	08/06/2016	23:53:20
[1]R0 for Breakdown 1	Passed		dev	08/06/2016	00:01:30

รูปที่ 4.14 ผลลัพธ์ของการควบคุมคุณภาพของซอฟต์แวร์ที่แสดงในอีเมล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

การใช้เงินกิ้นส์ ในหัวข้อที่ 2.2 เพื่อจัดการคุณภาพของซอฟต์แวร์นั้นได้นำหลักการของ Continuous Integration มาทำงานร่วมด้วย โดยจะเน้นในเรื่องการทำงานเป็นระบบอัตโนมัติ โดยจะมีการตรวจสอบ Source code หากมีการเปลี่ยนแปลง จะเริ่มการคอมไพล์และรวบรวมเป็นแพ็คเกจไฟล์ และนำมาทดสอบคุณภาพของซอฟต์แวร์ เพื่อให้ระบบนั้นพร้อมในการใช้งาน ซึ่งระบบที่ผู้จัดทำได้ทำการพัฒนาแล้ว จะสามารถทำงานได้อย่างอัตโนมัติ อีกทั้งยังแสดงรายละเอียดต่างๆ ในการทำงาน และผลลัพธ์ของการทำงาน ให้ผู้ใช้งานทราบได้โดยทันที ทำให้ลดความผิดพลาดที่จะเกิดขึ้นในการปฏิบัติงาน

หลักการในการทำงานของการใช้เงินกิ้นส์เพื่อคุณภาพของซอฟต์แวร์ แบ่งออกเป็น 4 ส่วนคือ

- 1) ส่วนของการพัฒนากระบวนการคอมไพล์ระบบจำลองแลกเปลี่ยนเงินตราระหว่างประเทศทำงานโดยการ เมื่อมีการเปลี่ยนแปลง Source code ที่ Repository กลาง ผู้ใช้งานจะเริ่มทำการเริ่มส่งการคอมไพล์และห่อรวมแพ็คเกจ
- 2) ส่วนของการพัฒนากระบวนการอัปเดตระบบแลกเปลี่ยนเงินตราต่างประเทศ ในส่วนนี้จะทำการตรวจสอบรุ่นของแพ็คเกจที่ได้มาจากการคอมไพล์ และอัปเดตระบบ
- 3) การปรับปรุงการแสดงผลการทดสอบระบบการควบคุมคุณภาพซอฟต์แวร์ในส่วนนี้จะเป็นการควบคุมประสิทธิภาพของซอฟต์แวร์โดยใช้เครื่องมือ HP quality center (QC) ช่วยในการทดสอบและแสดงผลการทดสอบอย่างทันที
- 4) ส่วนของการปรับปรุงการแสดงผล Release notes ของซอฟต์แวร์ ในส่วนนี้จะเป็นการเอกสารที่แสดงถึงความเปลี่ยนแปลงของระบบจำลองการแลกเปลี่ยนเงินตราต่างประเทศในแต่ละรุ่น

5.2 สรุปผลการทดสอบ

จากการพัฒนากระบวนการการควบคุมคุณภาพของซอฟต์แวร์นั้น ผู้ใช้งานได้ทำการใช้งานระบบผ่านทางเว็บเบราว์เซอร์และเป็นประโยชน์ในการช่วยเหลือผู้ใช้งานในการแสดงผลการคอมไพล์อัปเดตของซอฟต์แวร์ได้โดยทันที อีกทั้งยังลดขั้นตอนในการปฏิบัติงานในการผลิตซอฟต์แวร์และยังลดความเสี่ยงจากความผิดพลาดในการทำงานของผู้พัฒนาซอฟต์แวร์ได้

5.3 ข้อเสนอแนะของระบบ

เนื่องจากระบบนี้ได้ถูกพัฒนาขึ้นมาเพื่อเป็นการปรับปรุงกระบวนการต่างๆ ในการควบคุมคุณภาพของซอฟต์แวร์ ทางผู้พัฒนาเห็นว่าบางอย่างสามารถที่จะพัฒนาต่อไปได้ เพื่อความสะดวกในการใช้งานมากยิ่งขึ้น มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) การลงทะเบียนเพื่อการใช้งานระบบ หากสามารถเรียนรู้วิธีการสร้างการลงทะเบียนจากทีมงานต่างประเทศได้ ก็ไม่ต้องติดต่อทีมงานทางต่างประเทศเพื่อให้ลงทะเบียนให้
- 2) ด้านภาษาในพัฒนาระบบนั้นไม่สามารถเปลี่ยนแปลงหรือแก้ไขได้ เพราะจำเป็นต้องใช้เซิร์ฟเวอร์ลินุกซ์ในการทำงาน
- 3) สามารถเก็บสถิติจำนวนครั้งที่คอมไพล์สำเร็จของนักพัฒนาแล้วมาคำนวณโอกาสที่นักพัฒนาจะทำการคอมไพล์สำเร็จ ที่ไม่สามารถทำได้เพราะขาดการเก็บสถิติของการคอมไพล์ในขั้นตอนการพัฒนาซอฟต์แวร์
- 4) ออกแบบสคริปต์และการตั้งค่าเงินกินส์ให้ตอบสนองได้กับทุกซอฟต์แวร์ที่ต้องการใช้งาน ที่ไม่สามารถทำได้เพราะซอฟต์แวร์แต่ละตัวต้องการการตั้งค่าเฉพาะซอฟต์แวร์นั้นๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Anonymous. **Continuous Instigation**. [Online]. Available from : https://en.wikipedia.org/wiki/Continuous_integration
- [2] Sahill Patel. 2014. **Continuous Integration: How to Avoid 'Integration Hell'**. [Online]. Available from : <https://dzone.com/articles/continuous-integration-how-0>
- [3] Somkiat. 2014. **แนวปฏิบัติ 10 ข้อเรื่อง Continuous Integration**. [Online]. Available from : <http://www.somkiat.cc/continuous-integration-10-practice/>
- [4] Anonymous. **Jenkins (software)**. [Online]. Available from : [https://en.wikipedia.org/wiki/Jenkins_\(software\)](https://en.wikipedia.org/wiki/Jenkins_(software))
- [5] Anonymous. **Shell Script**. [Online]. Available from : https://en.wikipedia.org/wiki/Shell_script
- [6] Anonymous. **เชลล์โปรแกรมมิ่ง (Shell Programming)**. [Online]. Available from : http://teacher.en.rmutt.ac.th/nuchtiphong.o/index2.php?option=com_docman&task=doc_view&gid=14&Itemid=32
- [7] Jeremy Sanders. 2011. **Basic Shell Commands**. [Online]. Available from : https://www.xray.ast.cam.ac.uk/~jss/lecture/computing/notes/out/commands_basic/
- [8] Anonymous. **Version Control**. [Online]. Available from : https://en.wikipedia.org/wiki/Version_control
- [9] Scott Chacon, Ben Straub. **Git Pro: Getting Started – About Version Control**. [Online]. Available from : <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- [10] Scott Chacon, Ben Straub. **Git Pro: Getting Started - Git Basics**. [Online]. Available from : <https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>
- [11] Anonymous. **Vbscript Tutorial**. [Online]. Available from : <http://www.tutorialspoint.com/vbscript/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง(ต่อ)

- [12] Anonymous. HP Quality Center. [Online]. Available from :
https://en.wikipedia.org/wiki/HP_Quality_Center
- [13] Promlert. 2015. การ Build คืออะไร ? Build Tool คืออะไร? Gradle และ Ant คืออะไร และเกี่ยวข้อกันยังไง. [Online]. Available from :
<http://3bugs.com/2015/03/what-is-build-tool-gradle-ant/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



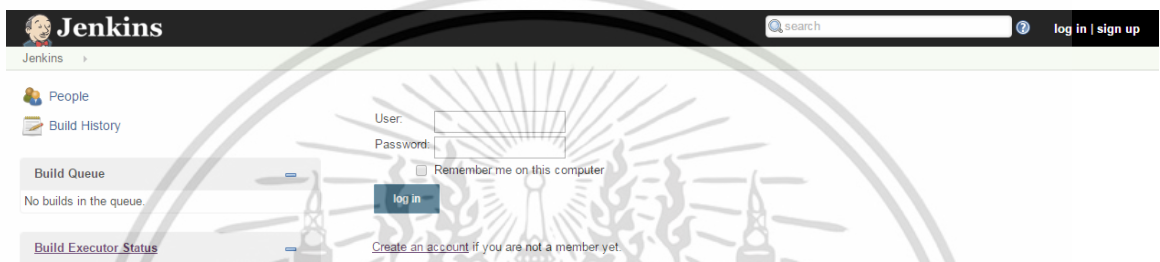
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คู่มือการใช้งานระบบอัตโนมัติการคอมไพล์ระบบจำลองการแลกเปลี่ยน เงินตราระหว่างประเทศ

ก.1 คู่มือการใช้งานระบบ

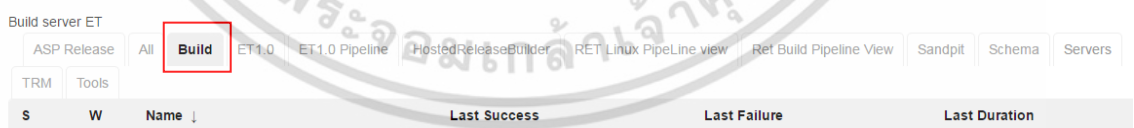
1) เมื่อผู้พัฒนาระบบได้ทำการปรับปรุงหรือเพิ่มเติมโค้ดเรียบร้อยแล้ว หากต้องการที่จะ Build Package นั้น ต้องเข้าไปที่หน้าเว็บไซต์เจเนกินส์แล้วทำการสู่ระบบ ดังรูปที่ ก.1



รูปที่ ก.1 หน้าจอเข้าสู่ระบบ

จากรูปที่ ก.1 ให้ผู้ใช้งานเข้าสู่ระบบโดยทำการป้อน “ชื่อผู้ใช้งาน” และ “รหัสผ่านของผู้ใช้งาน” จากนั้นกดปุ่ม “log in” เพื่อทำการเข้าสู่ระบบ

2) เมื่อผู้ใช้งานเข้าสู่ระบบเรียบร้อยแล้ว จะมีแท็บให้เลือก โดยให้เลือกไปที่แท็บ “Build” ดังรูปที่ ก.2



รูปที่ ก.2 แสดงแท็บการ Build

3) เมื่อผู้ใช้งานคลิกที่แท็บ “Build” แล้ว หน้าจอจะแสดงรายละเอียดของงาน (Jobs) ที่สามารถทำงานได้ โดยจะแบ่งไปในแต่ละเวอร์ชัน ดังรูปที่ ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

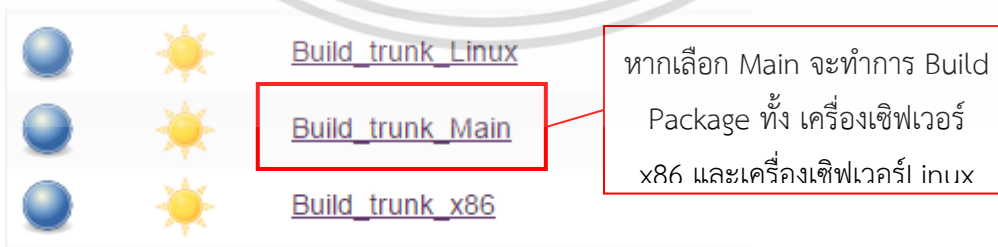
S	W	Name ↓	Last Success	Last Failure	Last Duration
☉	☀	Build_ET1.0_Main	1 mo 19 days - #2	N/A	20 sec
☉	☁	Build_ET1.0_sunfire142	N/A	1 mo 16 days - #1	1.4 sec
☉	☀	Build_ET1.0_x86	N/A	N/A	N/A
🌐	☁	Build_ET2.0_Linux	2 mo 3 days - #32	2 mo 6 days - #30	16 min
🌐	☁	Build_ET2.0_Main	2 mo 3 days - #21	2 mo 6 days - #18	0.39 sec
🌐	☀	Build_ET2.0_x86	1 mo 18 days - #25	2 mo 6 days - #17	16 min
🌐	☀	Build_ET2.1_Linux	11 days - #7	N/A	23 min
🌐	☀	Build_ET2.1_Main	11 days - #8	N/A	36 min
🌐	☀	Build_ET2.1_x86	11 days - #7	N/A	13 min
🌐	☀	Build_ET2.2_Linux	3 days 19 hr - #11	N/A	25 min
🌐	☀	Build_ET2.2_Main	3 days 20 hr - #12	N/A	42 min
🌐	☀	Build_ET2.2_x86	3 days 20 hr - #13	N/A	16 min
🌐	☀	Build_trunk_Linux	11 hr - #28	N/A	24 min
🌐	☀	Build_trunk_Main	11 hr - #29	1 mo 9 days - #8	44 min
🌐	☀	Build_trunk_x86	11 hr - #29	N/A	19 min

รูปที่ ก.3 หน้าจอแสดงรายละเอียดของงาน

คำอธิบาย

- S : แสดงผลการทำงานว่าสำเร็จ (สีฟ้า) หรือ ไม่สำเร็จ (สีแดง)
- W : แสดงค่าเฉลี่ยการทำงานสำเร็จหรือไม่สำเร็จโดยจะแสดงเป็นสภาพอากาศ
- Name : แสดงชื่อของงาน
- Last Success : แสดงการทำงานที่สำเร็จล่าสุดของงานนั้นๆ
- Last Failure : แสดงการทำงานที่ไม่สำเร็จล่าสุดของงานนั้นๆ
- Last Duration : แสดงเวลารวมในการทำงานล่าสุดของงานนั้นๆ

4) หากต้องการทำการ Build Package ของเวอร์ชันนั้นๆ ให้ทำการเลือกงาน ดังรูปที่ ก.4



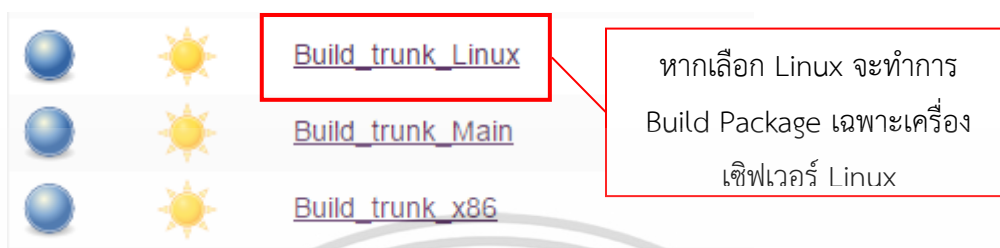
รูปที่ ก.4 แสดงการเลือกงานของการ Build Package (1)

จากรูปที่ ก.4 ชื่อของงานจะเป็น “Build_ [ชื่อของเวอร์ชัน]_ [ชื่อเครื่องเซิร์ฟเวอร์]” โดยชื่อเซิร์ฟเวอร์จะแบ่งด้วยกัน 3 แบบดังคำอธิบายด้านล่าง

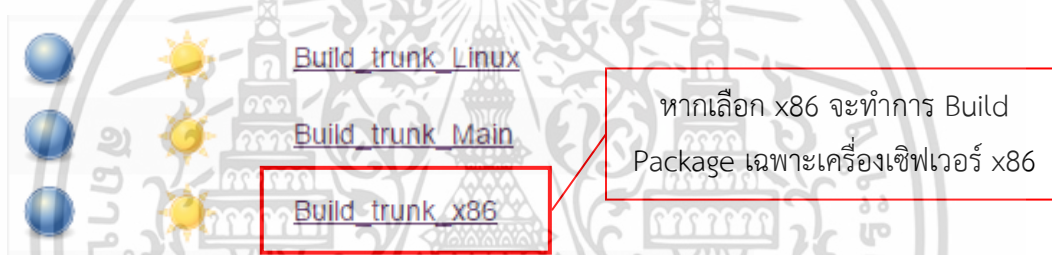
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบาย

- Main : สั่งทำงานทั้งเครื่องเซิร์ฟเวอร์ x86 และเครื่องเซิร์ฟเวอร์ Linux
- Linux : สั่งทำงานเพราะเครื่องเซิร์ฟเวอร์ Linux
- X86 : สั่งทำงานเฉพาะเครื่อง x86



รูปที่ ก.5 แสดงการเลือกงานของการ Build Package (2)



รูปที่ ก.6 แสดงการเลือกงานของการ Build Package (3)

5) เมื่อผู้ใช้งานทำการเลือกงานได้แล้ว หน้าจอจะแสดงรายละเอียดของงานนั้นๆ โดยจะแสดงรายละเอียดของงานว่าทำอะไรบ้าง ดังรูปที่ ก.7

Jenkins > Build > Build_trunk_Main >

ชื่อของงาน: Project Build_trunk_Main

รายละเอียดของ: build Trunk on x86 and Linux

Build Number	Time	User
#29	Jun 27, 2016 4:13 PM	[User Icon]
#28	Jun 27, 2016 2:54 PM	[User Icon]
#27	Jun 24, 2016 12:49 PM	[User Icon]
#26	Jun 22, 2016 10:27 AM	[User Icon]

ประวัติการทำงาน: Build_trunk_x86(blocking)

Other executed recently: Build_trunk_Linux(blocking)

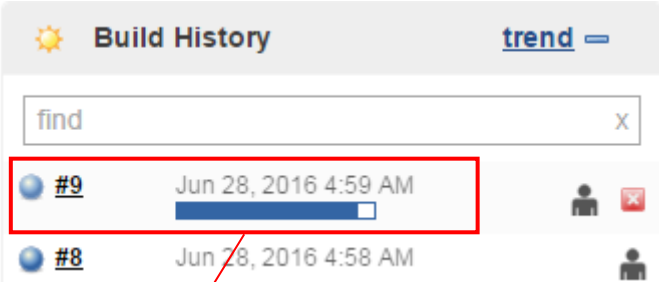
รูปที่ ก.7 หน้าจอแสดงรายละเอียดของงาน

6) เมื่อเข้าสู่หน้าจอแสดงรายละเอียดได้แล้ว หากผู้ใช้งานต้องการทำการ Build Package จะต้องทำการคลิกที่ปุ่ม “Build Now” แล้วจะเริ่มทำงาน ดังรูปที่ ก.8

เมื่อคลิกที่ปุ่ม “Build Now” จะทำการเริ่มกระบวนการการ Build Package ของเวอร์ชันนั้นๆ

รูปที่ ก.8 แสดงการเริ่มต้นการ Build Package

เมื่อผู้ใช้งานคลิกที่ปุ่ม “Build Now” เสร็จสิ้นแล้ว จะเริ่มทำการ Build Package โดยจะแสดงการทำงานในส่วนของ “Build History” ดังรูปที่ ก.9



Build History trend

find X

#9 Jun 28, 2016 4:59 AM

#8 Jun 28, 2016 4:58 AM

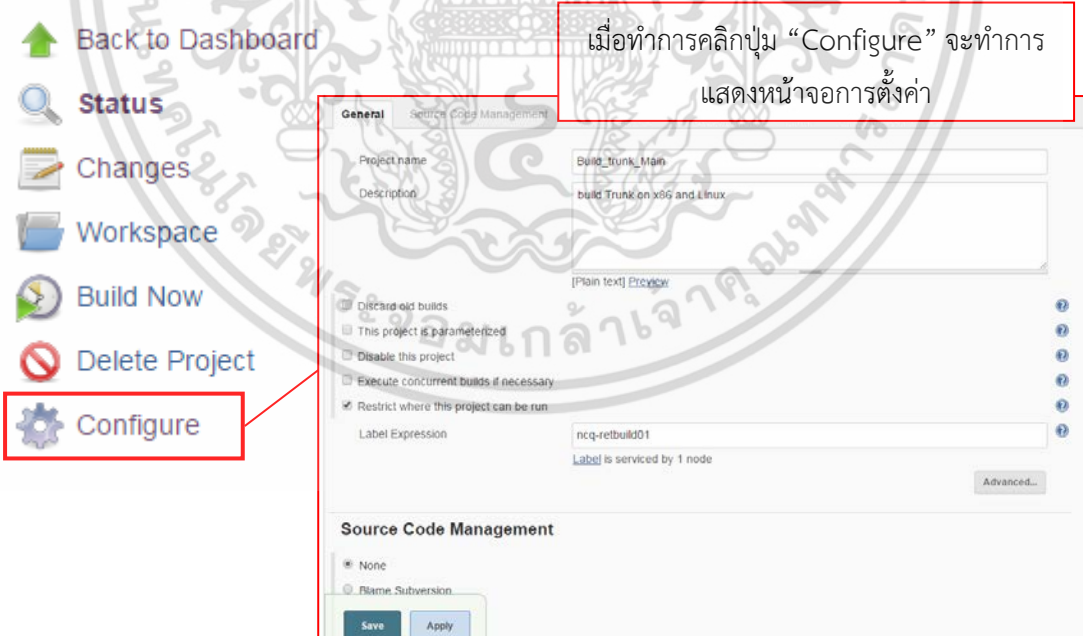
Console Output

Started by user [redacted]
 Building remotely on ncq-retbuild01 in workspace [redacted]
 Waiting for the completion of Build_trunk_x86
 Build_trunk_x86 #29 completed. Result was SUCCESS
 Waiting for the completion of Build_trunk_Linux
 Build_trunk_Linux #28 completed. Result was SUCCESS
 Build step 'Trigger/call builds on other projects' changed build result to SUCCESS
 Finished: SUCCESS

เมื่อทำการคลิกที่การทำงาน จะแสดง “Console Output” ที่แสดงการทำงานของการทำงานของการ Build Package

รูปที่ ก.9 แสดงการทำงานและผลลัพธ์ของการ Build Package

7) หากผู้ใช้งานต้องการที่จะแก้ไขการทำงานต่างๆ ในแต่ละงาน สามารถแก้ไขได้โดยคลิกปุ่ม “Configure” และจะแสดงต่างๆ ในการแก้ไขหรือตั้งค่าการทำงานต่างๆ ดังรูปที่ ก.10



Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

เมื่อทำการคลิกปุ่ม “Configure” จะทำการแสดงหน้าจอการตั้งค่า

General Source Code Management

Project name: Build_trunk_Main
 Description: build Trunk on x86 and Linux
 [Plain text] Etc...
 Discard old builds
 This project is parameterized
 Disable this project
 Execute concurrent builds if necessary
 Restrict where this project can be run
 Label Expression: ncq-retbuild01
 Label is serviced by 1 node
 Advanced...

Source Code Management

None
 Flame Subversion

Save Apply

รูปที่ ก.10 หน้าจอแสดงการตั้งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คู่มือการใช้งานระบบอัตโนมัติการอัปเดตระบบจำลองการแลกเปลี่ยนเงินตรา ระหว่างประเทศ

ข.1 คู่มือการใช้งานระบบ

1) เมื่อผู้ใช้งานต้องการที่จะอัปเดตระบบจำลองการแลกเปลี่ยนเงินตราระหว่างประเทศ ต้องเข้าที่เว็บไซต์เจเนกินส์ และต้องทำการคลิกแท็บ “installer” เพื่อทำการเข้าสู่หน้าจอที่แสดงรายการของงานต่างๆ ดังรูปที่ ข.1



รูปที่ ข.1 แสดงการเลือกแท็บ installer

2) เมื่อผู้ใช้งานคลิกแท็บ “installer” เสร็จสิ้นแล้ว จะเข้าสู่หน้าจอแสดงงานของการอัปเดตรุ่นของระบบโดยแยกไปตามเวอร์ชันของระบบ ดังรูปที่ ข.2

S	W	Name ↓	Last Success	Last Failure	Last Duration
		installer_3.5.711-RET3.5-trunk	4 hr 6 min - #10	2 mo 3 days - #2	15 min
		installer_3.5.711.1.1108-ET2.1	11 days - #18	11 days - #17	12 min
		installer_3.5.711.1.1230-ET2.2	3 days 21 hr - #57	1 mo 1 day - #55	13 min
		installer_3.5.711.1.980-ET2.0			
		installer_informal			
		Test_A			

S	W	Name ↓
		installer_3.5.711-RET3.5-trunk
		installer_3.5.711.1.1108-ET2.1
		installer_3.5.711.1.1230-ET2.2
		installer_3.5.711.1.980-ET2.0
		installer_informal
		Test_A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.2 หน้าจอแสดงงานของการอัปเดตระบบ

3) เมื่อผู้ใช้งานต้องการที่อัปเดตระบบ ต้องทำการเลือกเวอร์ชันที่ต้องการอัปเดต โดยชื่อของงานจะเป็นตัวกำหนด installer_[เวอร์ชันของระบบ] ดังรูปที่ ข.2 เมื่อผู้ใช้งานเลือกเวอร์ชันที่จะทำการอัปเดตได้แล้ว จะเข้าไปสู่หน้าจอแสดงรายละเอียดของการอัปเดตระบบในแต่ละเวอร์ชัน ดังรูปที่ ข.3

The screenshot shows the Jenkins interface for a project named "Project installer_3.5.711-RET3.5-trunk". The main content area displays the project name and a description: "Upgrade 3.5.711-RET3.5-trunk branch version to local system." Below this, there is a "Build History" table with the following data:

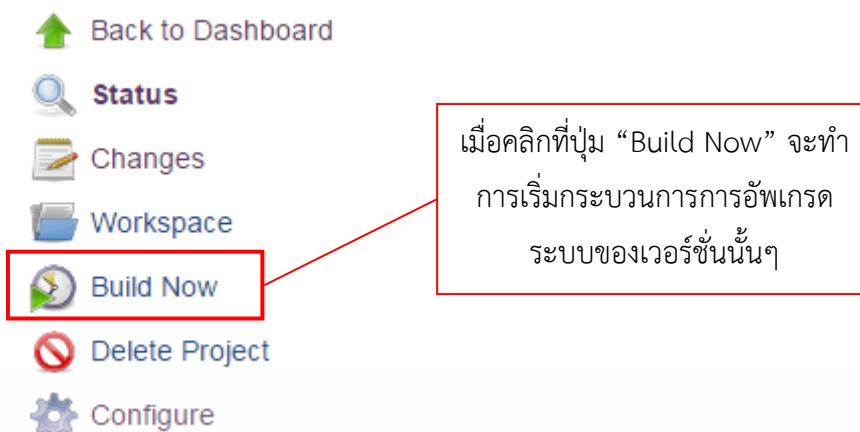
Build Number	Time
#10	Jun 28, 2016 3:20 AM
#9	Jun 27, 2016 4:03 AM
#6	Jun 6, 2016 8:14 AM

The "Permalinks" section provides links to various build-related information:

- ประวัติการทำงาน (Build History)
- Last successful build (#10), 4 hr 22 min ago
- Last failed build (#2), 2 mo 3 days ago
- Last unsuccessful build (#2), 2 mo 3 days ago
- Last completed build (#10), 4 hr 22 min ago

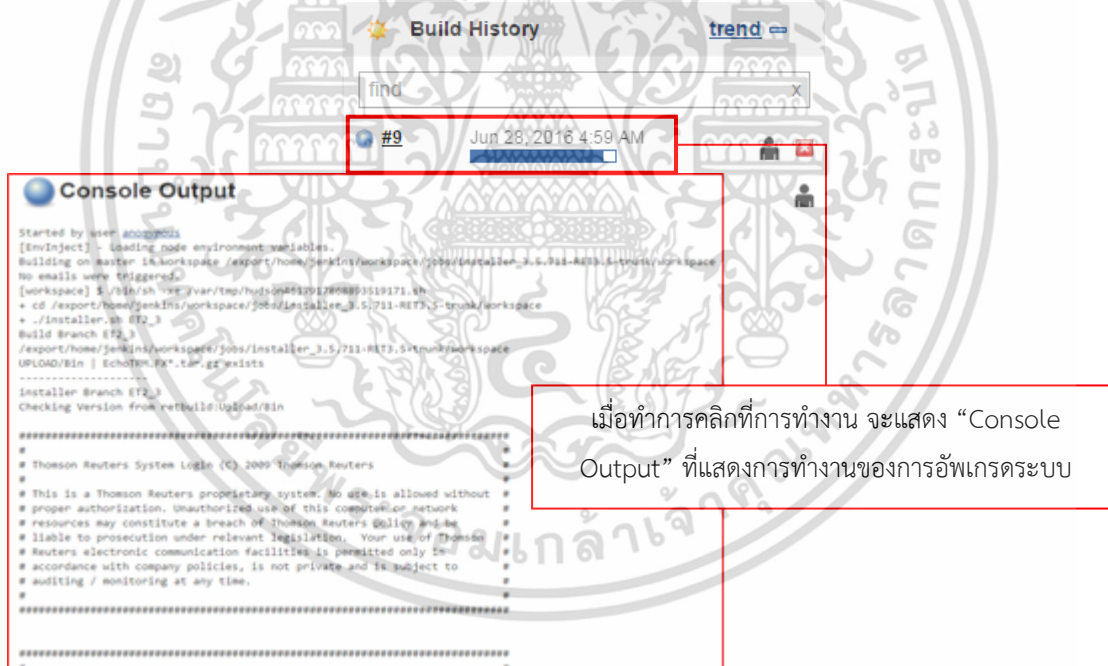
รูปที่ ข.3 หน้าจอแสดงรายละเอียดของงานในแต่ละเวอร์ชัน

4) เมื่อผู้ใช้งานต้องการที่จะอัปเดตระบบ ให้คลิกที่ปุ่ม "Build Now" ที่หน้าจอแสดงรายละเอียดของงานในแต่ละรุ่น ดังรูปที่ ข.4



รูปที่ ข.4 แสดงการเริ่มต้นการอัปเดตระบบ

เมื่อผู้ใช้งานคลิกปุ่ม “Build Now” แล้ว จะเริ่มทำการอัปเดตระบบ โดยจะแสดงการทำงานในส่วนของ “Build History” ในหน้าจอแสดงรายละเอียดของงาน ดังรูปที่ ข.5



รูปที่ ข.5 หน้าจอแสดงการทำงานและผลลัพธ์ของการอัปเดตระบบ

5) หากผู้ใช้งานต้องการที่จะแก้ไขการทำงานต่างๆ ในแต่ละงาน สามารถแก้ไขได้โดยคลิกปุ่ม “Configure” และจะแสดงต่างๆ ในการแก้ไขหรือตั้งค่าการทำงานต่างๆ ดังรูปที่ ข.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการคลิกปุ่ม “Configure” จะทำการ
แสดงหน้าจอการตั้งค่า

รูปที่ ข.6 หน้าจอแสดงการตั้งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้