

การกำหนดตารางการทดสอบซอฟต์แวร์แบบอัตโนมัติ ให้ Rational Functional Tester โดยใช้ Microsoft Excel

เมฆินทร์ วรรณศาสตร์

คณะวิทยาการและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีมหานคร
maeklong@facebook.com

ศุภกร กังพิศดาร

คณะวิทยาการและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีมหานคร
supakorn@mut.ac.th

บทคัดย่อ

การทดสอบซอฟต์แวร์ (Software Testing) ถือเป็นขั้นตอนที่สำคัญในการพัฒนาซอฟต์แวร์ที่ช่วยเพิ่มคุณภาพของซอฟต์แวร์ แม้ว่าในปัจจุบันจะมีเครื่องมือที่ช่วยในการทดสอบแบบอัตโนมัติอยู่มากมาย แต่เครื่องมือเหล่านั้นก็ยังคงขาดคุณสมบัติด้านการใช้งานที่จำเป็น นั่นคือการจัดตารางการทดสอบ นอกจากนี้ผลการทดสอบที่ได้ยังต้องใช้โปรแกรมเฉพาะในการเปิดอ่าน ทำให้ไม่สะดวกต่อการใช้งาน งานวิจัยฉบับนี้มุ่งเน้นการพัฒนาในด้านการจัดตารางการทดสอบ และการบันทึกผลการทดสอบในรูปแบบที่ง่ายต่อการนำไปใช้งานต่อไป โดยทำการพัฒนาเครื่องมือสำหรับการทดสอบที่ชื่อว่า Rational Functional Tester ให้สามารถทำการทดสอบตามตารางการทดสอบจากแฟ้มข้อมูลที่เก็บกรณีทดสอบ ซึ่งแฟ้มดังกล่าวอยู่เป็นแฟ้มประเภท Microsoft Excel ซึ่งใช้เก็บกรณีทดสอบกันโดยแพร่หลาย รวมถึงบันทึกผลการทดสอบลงในแฟ้มข้อมูลดังกล่าวเพื่อนำไปออกรายงานได้อย่างสะดวก

Abstract

Software Testing is considered as one of the most crucial processes in software development. Although a number of tools have been proposed to facilitate software testing, they still lack necessary features including test scheduling. Moreover, the results given from the test must be open by specific program. In this paper, we propose "Rational Functional Tester" as an automated test tool that is able to deal with scheduling and generating result logs. Furthermore, the result logs can be read through the popular Microsoft Excel program.

คำสำคัญ

การทดสอบซอฟต์แวร์, การทดสอบซอฟต์แวร์แบบอัตโนมัติ, กรณีทดสอบ, ผลการทดสอบ, ตารางการทดสอบ

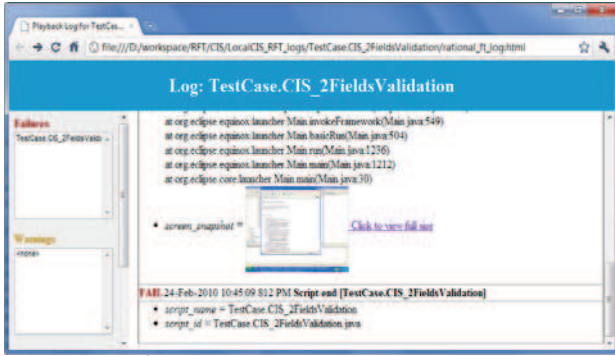
1. บทนำ

การทดสอบซอฟต์แวร์แบบอัตโนมัติ (Automated Software Testing หรือ Test Automation) คือ การใช้งานเครื่องมือซึ่งอยู่

ในรูปของโปรแกรมคอมพิวเตอร์ในการทำการทดสอบแทนการทดสอบด้วยคน (Manual Testing) เป็นการช่วยเพิ่มความถูกต้องแม่นยำของการทดสอบ รวมถึงช่วยลดระยะเวลาและค่าใช้จ่ายในการพัฒนาของโครงการพัฒนาซอฟต์แวร์ได้อย่างมาก ซึ่งเครื่องมือที่ช่วยในการทดสอบแบบอัตโนมัตินั้นมีอยู่หลายเจ้าด้วยกัน เช่น โปรแกรม Rational Functional Tester (RFT) ของ IBM, Quick Test Pro (QTP) ของ HP เป็นต้น เครื่องมือเหล่านี้จะกระทำการทดสอบซอฟต์แวร์ (หรือรัน) ตามสคริปต์ที่ถูกสร้างขึ้นมาจากการบันทึกขั้นตอนการใช้งานซอฟต์แวร์ที่ต้องการทดสอบด้วยตัวเครื่องมือโดยอัตโนมัติ หรือรันตามสคริปต์ที่เขียนขึ้นด้วยภาษาที่เครื่องมือนั้นรองรับ [2]

โดยทั่วไปการรันสคริปต์นั้นจะกระทำโดยการเปิดเครื่องมือทดสอบขึ้นมาแล้วสั่งรันสคริปต์ทีละสคริปต์ ซึ่งโดยส่วนใหญ่แล้วแต่ละสคริปต์จะแทนหนึ่งกรณีทดสอบ (Test Case) หากต้องสั่งรันสคริปต์ทีละสคริปต์ในโครงการทดสอบซอฟต์แวร์ที่มีขนาดใหญ่ขึ้นจะทำให้เป็นการเสียเวลาและค่าใช้จ่ายมาก

สำหรับผลการทดสอบที่ได้จากเครื่องมือทดสอบนั้น มักจะอยู่ในรูปแบบที่ไม่สามารถนำไปใช้งานได้ทันที สืบเนื่องจากการรันสคริปต์หนึ่งสคริปต์นั้นแทนการทดสอบหนึ่งกรณีทดสอบ ผลการทดสอบของแต่ละกรณีทดสอบจึงถูกสร้างขึ้นมาแยกกันตามลำดับการสั่งรัน และผลการทดสอบดังกล่าวมีข้อมูลมากเกินไปจนความจำเป็น ผู้ทำการทดสอบจะต้องทำการรวบรวมผลเอง เพื่อออกรายงานในภาพรวมสำหรับใช้ในการตัดสินใจเกี่ยวกับการบริหารโครงการต่อไป ซึ่งเป็นการสิ้นเปลืองเวลาและค่าใช้จ่ายอีกเช่นกัน



รูปที่ 1 แสดงผลการทดสอบจากโปรแกรม RFT

2. งานและทฤษฎีที่เกี่ยวข้อง

ที่ผ่านมา มีนักวิจัยทำการศึกษเกี่ยวกับการใช้งานเครื่องมือทดสอบแบบอัตโนมัติ [2, 4, 6-8] ซึ่งส่วนใหญ่จะกล่าวถึงประโยชน์ของการใช้เครื่องมือสำหรับทดสอบว่าเป็นวิธีที่มีประสิทธิภาพมาก เพราะสามารถนำสคริปต์ที่บันทึก (Record) กลับมาใช้ใหม่ (Playback) ได้ง่าย ยืดหยุ่น และเป็นการลดค่าใช้จ่าย แต่อย่างไรก็ตามงานวิจัยดังกล่าว ไม่ได้กล่าวถึงการทำให้เครื่องมือที่ใช้งานได้อย่างรวดเร็ว และมีประสิทธิภาพมากขึ้นในทางปฏิบัติ

ผลที่ได้จากการทำการทดสอบ Nawalikit et al. ได้เสนอ SBTAR [1] สำหรับการเพิ่มประสิทธิภาพของการออกผลการทดสอบ โดยเปลี่ยนรูปแบบของผลการทดสอบจากรูปแบบที่เครื่องมือทดสอบนั้นสร้างขึ้น และต้องใช้เครื่องมือดังกล่าวเท่านั้นในการเปิด ให้มาอยู่ในรูปแบบของแฟ้มข้อมูลประเภท Microsoft Word ซึ่งใช้งานกันอย่างแพร่หลาย แต่อย่างไรก็ตามผลจากการทดสอบยังคงแยกแฟ้มข้อมูลกันตามกรณีทดสอบ ซึ่งถือเป็นการยากในการนำมาสร้างรายงานสรุปผลการทดสอบแบบภาพรวม

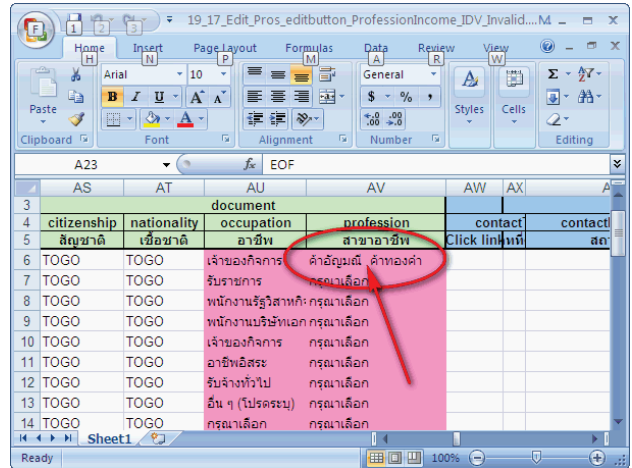
3. รายละเอียดการพัฒนา

3.1 ภาพรวมของระบบ

IBM Rational Functional Tester (หรือ RFT) เป็นซอฟต์แวร์สำหรับการทดสอบการทำงานของซอฟต์แวร์แบบอัตโนมัติ โดยการทำงานเริ่มจากใช้ RFT ในการบันทึกขั้นตอนการทำงานตามกรณีทดสอบ และสร้างสคริปต์จากการบันทึกขั้นตอนทดสอบดังกล่าวออกมาเป็นโค้ดภาษา Java Scripting [10] ซึ่งสามารถทำการแก้ไขสคริปต์ดังกล่าวเพื่อความเหมาะสมมากขึ้นได้ เช่น การเปลี่ยนข้อมูลสำหรับทำการทดสอบจาก

ข้อมูลที่ใช้ขณะบันทึกสคริปต์ไปเป็นข้อมูลสำหรับทดสอบ (Test Data) ชุดอื่นๆ หรือแม้กระทั่งการเพิ่มเงื่อนไขเส้นทางในการทดสอบให้ครอบคลุมตามกรณีทดสอบที่มีอยู่ เป็นต้น

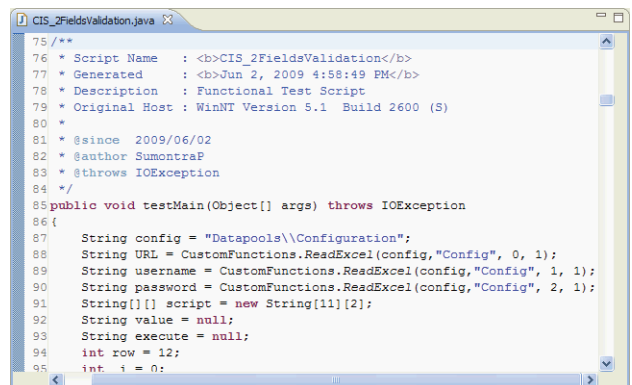
โดยส่วนใหญ่การเตรียมข้อมูลสำหรับทำการทดสอบ เช่น ข้อมูลที่ใช้ในการกรอกเข้าไปฟอร์มของโปรแกรมที่จะทำการทดสอบนั้น มักจะถูกเตรียมไว้ในแฟ้มข้อมูลตารางงาน เช่น Microsoft Excel (หรือ Excel) ดังตัวอย่างในรูปที่ 2



รูปที่ 2 แสดงข้อมูลสำหรับทดสอบใน Microsoft Excel

3.2 การออกแบบและพัฒนาระบบ

ในการออกแบบระบบนี้ ผู้วิจัยกำหนดให้สคริปต์เริ่มต้นของการทดสอบไปการอ่านแฟ้มข้อมูลคอนฟิกูเรชัน (Configuration) ก่อน ดังรูปที่ 3 ซึ่งแฟ้มดังกล่าวถูกจัดเก็บอยู่ในแฟ้มข้อมูล Excel ดังรูปที่ 4 จึงจำเป็นต้องใช้ไลบรารี (Library) สำหรับภาษา Java เพื่อการอ่านและเขียนข้อมูลลงในแฟ้มข้อมูล Excel ซึ่งได้แก่ Java Excel API [12], Jacob API [13] และ POI library [14]



รูปที่ 3 แสดงสคริปต์เริ่มต้นในโปรแกรม RFT

	A	B	C
1	URL	http://maeklong.org.7001/cis/	
2	Username	maykin.W	
3	Password	password1	
4			
5	PrtSc	0	
6	Doc	0	
7	All Script	0	
8			
9	Script	Execute	
10	Search	0	
11	Add Customer	0	
12	Edit Customer	1	
13	Add Prospect	0	
14	Edit Prospect	1	
15	Change P to C	0	
16	Change Cust type	1	
17	Relation	0	
18	Add Single Account	1	
19	Add Co Account	0	
20	Edit Account	0	
21	EOF		

รูปที่ 4 แสดงข้อมูลในแฟ้มคอนฟิกูเรชัน

ภายในแฟ้มคอนฟิกูเรชัน จะกำหนดคอลัมน์และแถวที่ใช้กรอกชื่อกรณีทดสอบ โดยมีอีกหนึ่งคอลัมน์ไว้ใช้กรอกค่า เช่น 0 หรือ 1 เพื่อใช้เป็นเงื่อนไขในการตัดสินใจว่าจะทดสอบกรณีดังกล่าวหรือไม่ โดยกำหนดไว้ว่าถ้าไม่มีกรณีทดสอบแล้วให้กรอกข้อมูลเซลล์ถัดไปเป็นค่าบางอย่าง ที่ใช้บ่งบอกว่าไม่มีกรณีทดสอบอีกแล้ว เช่น EOF เป็นต้น ดังแสดงในรูปที่ 4

โดยวิธีนี้จะช่วยให้ไม่ต้องเสียเวลาเปิดสคริปต์ขึ้นมาสั่งรันทีละสคริปต์อีกต่อไป เพื่อเป็นการประหยัดเวลา และค่าใช้จ่ายในการบริหารโครงการได้ รวมถึงสามารถเลือกได้ว่าจะทำการทดสอบกรณีทดสอบใด และละเว้นกรณีทดสอบใดได้อีกด้วย

หลังจากอ่านค่าจากเซลล์ในคอลัมน์ที่บ่งบอกว่า ต้องทำการทดสอบกรณีใดแล้ว ตัวสคริปต์เริ่มต้นจะทำการอ้างอิงไปยังสคริปต์ถัดไปซึ่งใช้อ้างอิงว่าแถวใดจากข้างต้น จับคู่กับสคริปต์ของกรณีทดสอบใด ซึ่งมีการจัดเก็บข้อมูลสำหรับการทดสอบไว้ในแฟ้มคอนฟิกูเรชันดังกล่าวด้วย โดยเก็บแยกตารางงาน (Worksheet) ไว้ ทำให้มีการเปิดแฟ้มข้อมูลเพียงแค่แฟ้มเดียวคือแฟ้มคอนฟิกูเรชันเท่านั้น

ในการอ่านข้อมูลจากแฟ้มข้อมูลสำหรับการทดสอบ ด้วยความสามารถของไลบรารีที่เพิ่มเข้ามาในตอนต้น สามารถทำการสกัดข้อมูลจากเซลล์ต่างๆ ออกมา และนำไปกรอกในฟอร์มของโปรแกรมที่ต้องการทำงานการทดสอบได้โดยง่าย ซึ่งการเก็บข้อมูลจะเก็บไว้ในลักษณะ หนึ่งกรณีทดสอบต่อหนึ่งแถว ข้อมูลในแฟ้มข้อมูลสำหรับการทดสอบ เมื่ออ่านข้อมูลไปใช้งานจนหมดทั้งแถวแล้ว ถือว่าหมดข้อมูลที่จะทำการกรอกแล้ว ซึ่งแต่ละกรณีทดสอบจะมีการกรอกข้อมูลที่ไม่เหมือนกัน ขึ้นอยู่กับวัตถุประสงค์ของกรณีทดสอบนั้นๆ ทำให้สามารถละเว้น

ข้อมูลในบางเซลล์ได้ โดยกำหนดในสคริปต์ว่าเมื่ออ่านข้อมูลจากเซลล์แล้วเป็นค่าว่างถือว่าไม่ต้องกรอกข้อมูล และได้กำหนด EOF ไว้ เพื่อบอกสคริปต์ว่า ข้อมูลที่ต้องการใช้ทำการทดสอบหมดลงแล้ว

เมื่อทำการทดสอบจนเสร็จกรณีทดสอบนั้นแล้ว ผลลัพธ์ที่ผู้ทดสอบต้องการคือผลการทดสอบว่าผ่าน (Passed) คือโปรแกรมที่ต้องการทดสอบนั้น ทำงานได้ถูกต้องตามที่คาดหวัง (Expected Result) หรือไม่ผ่าน (Failed) กรณีที่โปรแกรมทำงานไม่ถูกต้องตามที่คาดหวัง และด้วยความสามารถของ RFT เองที่สามารถอ่านค่าจากคอมพิวเตอร์ของซอฟต์แวร์ที่ถูกทดสอบได้ เช่น ฟอร์ม ตาราง ปุ่ม หรือหน้าต่างข้อความเตือนต่างๆ ซึ่งสามารถนำค่าดังกล่าวมาเปรียบเทียบกับผลที่ควรจะเป็นได้โดยสคริปต์เองเลย และสามารถบันทึกผลการทดสอบ (Test Result) ลงในแฟ้มที่เป็นข้อมูลสำหรับการทดสอบได้เลย โดยมีคอลัมน์ที่เตรียมไว้สำหรับเก็บผลการทดสอบ

ผลจากการทดสอบ ซึ่งถูกจัดเก็บรวมไว้ในแฟ้มเดียวกันกับแฟ้มข้อมูลสำหรับการทดสอบนี้ ทำให้สามารถนำไปใช้งานต่อไป เช่น การวิเคราะห์หาสาเหตุการทำงานของโปรแกรมได้สะดวก รวดเร็ว

3.3 ข้อจำกัดของระบบ

เนื่องจากงานวิจัยนี้ทำการพัฒนาด้วยโปรแกรม RFT และ Excel ทำให้การนำไปใช้งานและพัฒนาต่อยอดจึงถูกจำกัดอยู่บนระบบที่มี RFT และ Excel ติดตั้งอยู่ พร้อมกันนั้นจำเป็นต้องมี Java Runtime Environment (หรือ JRE) [15] เพื่อสนับสนุนการทำงานของ RFT รวมถึงไลบรารีที่จำเป็นดังหัวข้อ 3.2

4. การทดสอบการใช้งาน

ในการทดสอบผู้วิจัยทำการทดสอบกับเว็บแอปพลิเคชันที่ทาง IBM เตรียมไว้สำหรับการทดลองใช้งาน RFT เนื่องจากเป็นเว็บแอปพลิเคชันที่มีคอมพิวเตอร์ครบถ้วน เพื่อให้ได้ผลการทดสอบที่ครบถ้วนสำหรับกรณีทดสอบที่จำเป็นโดยทั่วไป

4.1 สภาพแวดล้อมในการทดสอบ

ในการทดสอบการใช้งาน ได้ทำการทดสอบด้วย

- Microsoft Windows XP
- Java Runtime Environment v.6 updates 23

- IBM Rational Functional Tester v.8
- Microsoft Excel 2003

4.2 ผลการทดสอบและการวิจารณ์ผล

จากผลการทดสอบพบว่ารูปแบบที่นำเสนอสามารถลดภาระงานของผู้ทดสอบซอฟต์แวร์กรณีทำการทดสอบซ้ำภายหลัง (Regression Testing) [16] ได้ดี ดังตารางที่ 1 ซึ่งช่วยให้สามารถลดค่าใช้จ่ายเกี่ยวกับกำลังคนได้อีกด้วย

ตารางที่ 1. แสดงการเปรียบเทียบผลลัพธ์ระหว่างก่อนและหลังใช้ระบบที่นำเสนอ

กิจกรรม	ก่อนใช้ระบบฯ	หลังใช้ระบบฯ
การสร้างสคริปต์	30 วัน	31 วัน
การเตรียมข้อมูลทดสอบ	2 วัน	2 วัน
การรันสคริปต์รอบที่ 1	20 วัน	10 วัน
การรันสคริปต์รอบที่ 2	19 วัน	10 วัน
การรันสคริปต์รอบที่ 3	19 วัน	10 วัน

แม้ว่าในระบบที่นำเสนอนี้การทำสคริปต์เพื่อทำการทดสอบยังคงเสียเวลามากในช่วงแรกก็ตาม แต่ก็ถือได้ว่าเป็นปกติของการทำการทดสอบแบบอัตโนมัติอยู่แล้ว ที่ต้องเสียเวลามากในการเตรียมสคริปต์ให้สามารถทำงานได้อย่างยืดหยุ่น ถูกต้องแม่นยำ และครอบคลุมทุกกรณีทดสอบตามที่ได้วางแผนการทดสอบเอาไว้

การจัดการเรื่องผลการทดสอบซอฟต์แวร์ทำได้ง่ายขึ้น เพราะผลการทดสอบและกรณีทดสอบถูกจัดเก็บอยู่ด้วยกันในแฟ้มเดียวทำให้ง่ายในการวิเคราะห์หาสาเหตุในกรณีที่ทำการทดสอบแล้วผลลัพธ์ของการทดสอบออกมาเป็นไม่ผ่าน อีกทั้งยังสามารถตรวจสอบเพิ่มเติมในส่วนของผลการทดสอบที่สร้างขึ้นจากเครื่องมือทดสอบเองได้ด้วย เพราะในการพัฒนาระบบที่นำเสนอนี้ได้คงไว้ซึ่งผลการทดสอบที่เครื่องมือทดสอบสร้างขึ้นมาเองเอาไว้ด้วย

5. บทสรุป

ในการเพิ่มคุณภาพให้กับซอฟต์แวร์จำเป็นต้องมีการทดสอบเพื่อการรับรองคุณภาพของซอฟต์แวร์ดังกล่าว กระบวนการทดสอบซอฟต์แวร์ถือได้ว่าเป็นกระบวนการที่สิ้นเปลืองเวลา และสร้างความเบื่อหน่ายแก่ผู้ที่มีหน้าที่ในการทำการ

ทดสอบ จึงได้มีการพัฒนาซอฟต์แวร์ที่ใช้สำหรับหุ่นแรงในการทำการทดสอบซอฟต์แวร์ อย่างไรก็ตามซอฟต์แวร์ดังกล่าวก็ยังไม่สามารถจัดตารางการสั่งรันสคริปต์ทดสอบได้ รวมถึงไม่สามารถออกผลลัพธ์ของการทดสอบที่นำไปใช้ประโยชน์ในการตัดสินใจในภาพรวมได้

งานวิจัยนี้ได้เสนอวิธีในการแก้ปัญหาดังกล่าวโดยการใช้งานแฟ้มข้อมูลชนิดตารางงาน ในการจัดตารางการรันสคริปต์ทดสอบ และเก็บผลการทดสอบแต่ละกรณีทดสอบไว้ด้วยกัน เพื่อลดเวลาและค่าใช้จ่ายในการสั่งรันสคริปต์ รวมถึงผลการทดสอบที่ได้นำไปใช้ประโยชน์ในภาพรวมได้ง่าย

5.1 แนวทางการพัฒนาต่อ

การพัฒนาการใช้งานร่วมกับแฟ้มข้อมูลชนิดตารางงานแบบอื่นๆ เช่น โปรแกรม Open Office [17] หรือ IBM Lotus Symphony [18] จะช่วยให้สามารถนำไปใช้งานได้แพร่หลายมากขึ้นเพราะเป็นโปรแกรมที่ไม่ต้องเสียค่าลิขสิทธิ์

6. กิตติกรรมประกาศ

ขอขอบคุณ คุณสุนตรา ปัญจรัตน์ และคุณกันต์กมล เจริมปลั่ง เพื่อนร่วมทีม TQIM บริษัท IBM Solutions Delivery จำกัด ที่ได้ให้ความช่วยเหลือทุกอย่างเมื่อพบกับปัญหาต่างๆ ขณะพัฒนาระบบ รวมถึงข้อมูลในเว็บไซต์ชุมชนนักพัฒนาของทั้ง IBM และ Microsoft ที่ช่วยให้เกิดแนวคิด แรงบันดาลใจ และองค์ความรู้ในการนำมาพัฒนาระบบนี้

7. เอกสารอ้างอิง

- [1] N. Nawalikit and P. Bhattarakosol, "SBTAR: An Enhancing Method for Automate Test Tools", World Academy of Science, Engineering and Technology 56 2009, pp. 754-758.
- [2] Z. Xiaochun, Z. Bo, L. Juefeng, G. Qiu, "A Test Automation Solution on GUI Functional Test", Eighth ACIS International Conference on Volume 3, Issue, July 30 2007-Aug. 1 2007, pp. 1124 – 1128.
- [3] Rational Functional Tester.
<http://www-01.ibm.com/software/awdtools/tester/functional/>
- [4] K.Karhu, T. Repo, O. Taipale, K. Smolander, "Empirical Observations on Software Testing Automation", Software Testing Verification and Validation, 2009. ICST apos;09.

International Conference on Volume , Issue , 1-4 April 2009,
pp. 201 - 209

- [5] J.B. Michael, B.J. Bossuyt, B.B. Snyder, "*Metrics for measuring the effectiveness of software-testing tools*", Software Reliability Engineering, 2002. ISSRE 2002. Proceedings. 13th International Symposium on Volume , Issue , 2002, pp. 117 – 128
- [6] C.Huang, H. Y. Chen, "*A Tool to Support Automated Testing for Web Application Scenario*", Systems, Man and Cybernetics, 2006. SMC apos;06. IEEE International Conference on Volume 3, Issue , 8-11 Oct. 2006, pp. 2179 – 2184
- [7] C.A. Bering, J.H. Covey, "*Software testing-concepts and approach*", Aerospace and Electronics Conference, 1991. NAECON 1991., Proceedings of the IEEE 1991 National Volume , Issue , 20-24 May 1991 pp. 750 - 756 vol.2
- [8] O. Taipale, K. Smolander, H. Kälviäinen, "*Cost Reduction and Quality Improvement in Software Testing*", in Software Quality Management Conference, Southampton, UK, 2006.
- [9] M. Fewster, "*Common Mistakes in Test Automation*", Grove Consultants 2001.
- [10] Oracle Technology, *Java Scripting Programmer's Guide*.http://download.oracle.com/javase/6/docs/technotes/guides/scripting/programmer_guide/
- [11] MSDN Community Center. <http://msdn.microsoft.com/en-us/aa497440>
- [12] Java Excel API. <http://jexcelapi.sourceforge.net/>
- [13] The JACOB Project. <http://danadler.com/jacob/>
- [14] Apache POI. <http://poi.apache.org/>
- [15] Java Runtime Environment.
<http://java.com/en/download/index.jsp>
- [16] Regression Testing.
http://en.wikipedia.org/wiki/Regression_testing
- [17] OpenOffice. <http://www.openoffice.org/>
- [18] IBM Lotus Symphony. <http://symphony.lotus.com/>