

開放源碼測試工具網站之分析比較

林昱安 林盈達

國立交通大學資訊工程系

E-mail: shower0621.cs00g@nctu.edu.tw, ydlin@cs.nctu.edu.tw

September 30, 2011

摘要

產品測試為管理產品品質的重要手段，為了測試產品特定的性質，測試人員需要一套特定的測試工具幫助他們達成目的。本文特別針對開放源碼測試工具網站 opensourcetesting.org 測試工具項目中數量較多的三大部分：web UI 測試、網路服務效能測試及測試用例與缺陷管理系統，挑選較廣為人知的數個測試工具，比較其功能，以方便測試人員根據需求挑選適合的測試工具。Web UI 測試工具中 Canoo Webtest 適於對簡單頁面進行快速的測試，Selenium 利於腳本設計，提供多種腳本語言支援，Sahi 則是操作容易適合一般人員；網路服務效能測試工具中 Apache Jmeter 以樹狀結構的形式表現測試腳本，有較佳的擴展性，OpenSTA 使用 windows 平台易於安裝及使用，Grinder 將行程分成三種階層利於更細緻地控制單一行程；測試用例管理工具的部分 TestLink 有很好的整合性，可以跟多種缺陷管理系統搭配；Testopia 為 Bugzilla 的專屬套件，可以跟 Bugzilla 有更佳的契合度；缺陷管理系統 mantis 功能簡單介面人性化，適合小型開發團隊或開放給客戶回報缺陷；Bugzilla 為歷史悠久的軟體，擁有許多強大的功能，適合大型開發團隊使用。

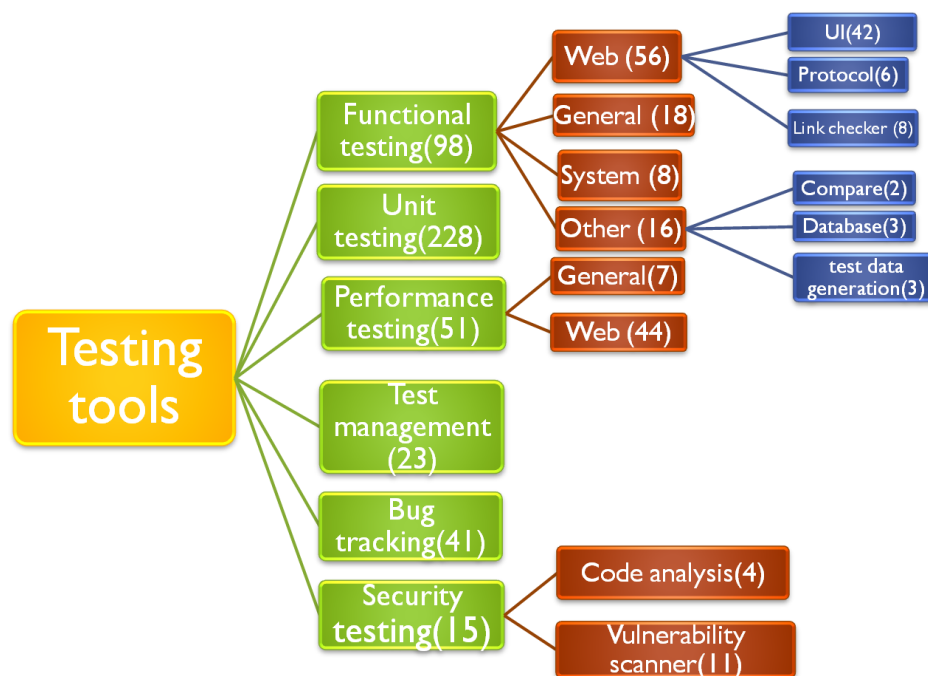
關鍵字：開放源碼、測試工具、[opensourcetesting](http://opensourcetesting.org)、web UI 測試、網路服務性能測試、測試用例與缺陷管理

1. 簡介

產品的生產過程中，測試是不可缺少的一環，產品測試的好壞關係到產品的質量，好的測試除了要制定一套完整嚴謹的測試程序規範外，還要針對產品的特性，使用適合的測試工具。市面上有許多測試工具，但專業的商用測試軟體大多所費不貲，對於預算有限的公司開放源碼測試工具也許是個不錯的選擇。這裡我們要介紹一個開放源碼測試工具網站 opensourcetesting.org[1]，這個網站包含了相當多不同功能的測試工具。在以下各節我們會先介紹這個網站的架構，由於這個網站中有太多的測試工具，接著我們會針對 Web UI、效能、測試用例與缺陷追蹤管理三大部分中較熱門的測試工具作分析比較，方便讀者了解各個工具的特性，將來再挑選這方面的測試工具時，可以根據測試需求挑選到適合的測試工具。

2. 開放源碼測試工具網站

Opensourcetesting.org 是 Mark Aberdour 於 2003 年三月建立，當初只有 50 個測試工具，現在已經有超過 360 種測試工具，還在持續增加中。除了提供測試工具的資訊，也提供簡單的上傳功能，還有專屬的論壇提供技術支援。



圖一：數字表示項目中的工具數量(時間為 2011.08)

圖一為網站架構，由圖中可看出網站測試工具主要分為六大類：功能測試 (functional testing)、單元測試(Unit testing)、性能測試(performance testing)、測試管理(test management)、缺陷追蹤管理(bug tracking)、安全測試(security testing)。

- 功能測試：測試產品的特定功能是否如預期般正確，如：測試 UI 介面功能，測試資料庫功能等，其中又以 web UI 測試工具的數量最多。
- 單元測試：開發時對隔離的獨立單元進行測試，屬於白箱測試。網站提供相當多種語言測試工具包括 C/C++、java、Perl、PHP、SQL、HTML、XML…等。
- 性能測試：主要用於尋找產品的瓶頸，此類工具通常也可用於負載測試 (load testing) 及壓力測試(stress testing)[2]；負載測試要測試產品的耐力，檢查產品在最大工作量下是否還是正常運作；壓力測試則是讓產品超過負擔，測試產品的復原能力。網站除了提供通用性的工具，一部分為專門測試網路服務性能的工具。
- 測試管理：測試用例是針對產品要進行的測試動作的描述，其中包括測試目標、測試環境、測試方法、輸入數據、預期結果等等。測試用例讓測試過程更加嚴謹、更易於管理，可以用最少的時間、人力找到缺陷完成測

試。

- 缺陷追蹤管理：確保缺陷是否被解決相當重要，缺陷追蹤管理工具可以讓管理者清楚的知道每個缺陷的情況，通常會與測試用例管理工具整合。
- 安全測試：主要針對產品所有可能的漏洞，靜態原始碼分析工具利用白箱測試找出原始碼中潛在的邏輯或程序問題；漏洞掃描工具則是黑箱測試，藉由模擬駭客攻擊的手法檢查漏洞。

以下針對此網站主要三大部分：功能測試的 web UI、網路服務效能測試、測試用例及缺陷追蹤管理，挑選較多人使用的工具比較分析。

3. Web UI 測試工具：Selenium、Sahi、Canoo Webtest

當一個 web 應用程式開發完成，需要很多的時間及人力執行測試，由於太過沒有效率，出現眾多能將 UI 測試自動化的測試工具，如 Selenium[3]、Sahi[4]、Canoo Webtest[5]、WatiN、Watir 等。其中又以 Selenium、Sahi、Canoo Webtest 較受歡迎，以下為三種測試工具的比較表[6]。

name	recorder	language support	speed	simulated way	report
Selenium	FireFox	Java,Ruby, C#,Python, Perl,PHP	slow	JavaScript-enabled browser	bad
Sahi	all browsers	Sahi script, Java,Ruby	slow	JavaScript-enabled browser	good
Canoo Webtest	FireFox (no update)	XML, Groovy	fast	HtmlUnit	good

表 1-1 web UI 測試工具比較

name	DOM	JS support	wait for page load & AJAX	editor support	user friendly
Selenium	XPath	better	no	very good	good
Sahi	Sahi API	better	yes	bad	good
Canoo Webtest	XPath	bad	no	good	bad

表 1-2 web UI 測試工具比較

由表 1-1, 表 1-2 中可以得知以下資訊：

- 錄製功能：Selenium 及 Sahi 都有提供錄製腳本的功能，提升測試的便利性。Selenium 錄製程式為 Firefox 附加元件，只能利用 Firefox 錄製；Sahi 擷取瀏覽器與伺服器之間的互動，支援在所有瀏覽器上的錄製功能，Canoo Webtest 雖然也有錄製工具，但版本太舊，新版的 Firefox 並不支

援，必須搭配其他額外的錄製工具。

- b. 測試UI的方式：Selenium與Sahi以代理的方式控制真實瀏覽器做測試，而Canoo Webtest則是利用HtmlUnit去模擬瀏覽器，而非使用真正的瀏覽器。這兩種方式的差異造成Selenium、Sahi執行速度較慢但對JavaScript的支援較佳，且較接近使用者真實情況；而Canoo Webtest執行速度快但對JavaScript的支援較差。Sahi和Selenium運作方式雖然相似，但還是有些許不同。Sahi會接收伺服器的回應再做下個指令，對Ajax[7]還有等待頁面讀取的處理較佳，Selenium只會單方面的將腳本指令下給瀏覽器，不理會伺服器的回應，需要測試人員自行處理。
- c. 腳本編寫及使用語言：Selenium擁有最多種的語言支援還有提供腳本語言轉換的功能，與Canoo Webtest都使用廣為人知的語言，可由測試者自行編寫，而Sahi主要使用自定義的API，較不適合編寫。但在文件物件模型(DOM)處理上，Sahi不像selenium及Canoo Webtest使用傳統Xpath查詢物件，而是使用較直覺的方式，自行定義API如_near、_under來抓取物件，優點在於當介面改變時，腳本的更動不大，但是較不精確。

綜合以上資訊，如果測試目標為單純的頁面，較少JavaScript及AJAX，Canoo Webtest會是好的選擇，可以節省時間；但如果是比較複雜的頁面，Selenium及Sahi較為適合，若測試者程式能力不強時，Sahi又比Selenium更加適合。

4. 效能測試工具：Apache Jmeter、OpenSTA、Grinder

網路服務要能提供大量用戶穩定的品質，為了測試網路服務的負載及效能，測試者需要可以模擬使用者連線並進行一些動作、測量回應時間的工具，這類型的工具有Apache Jmeter[8]、OpenSTA[9]、Grinder[10]等。

name	prerequisite	recorder	script language	protocol support	report	distributed test setting
Apache Jmeter	OS independent	yes	N/A	http,https,SOAP,JBDC,LDAP,JMS,POP3,IMAP	detailed	good
OpenSTA	windows 2000,NT4,XP	yes	Script Control Language	http,https	simple	good
Grinder	OS independent	yes	Jython	http,https,SOAP,JBDC,JMS,POP3	simple	bad (script for each)

表二 效能測試工具比較表

由表二[11]可以得知以下訊息：

- a. 作業平台：Apache Jmeter及Grinder是純Java應用程式可以跨平台，而OpenSTA限定windows平台。
- b. 腳本語言：Apache Jmeter為圖形介面，透過樹狀結構建立測試計畫，

提供靈活的擴展性。OpenSTA 則是使用自行定義的語言 SCL，一種類似 VB 的語言；而 Grinder 則是使用較為人知的 Jython，以上兩者較能對行程進行更深入的操作。

- c. 分散測試：Apache Jmeter 及 OpenSTA 皆使用主從結構，由一台主機主控，測試計畫位於主控的機器，修改只需在主控端進行。而 Grinder 則是將行程分成三種身分 console、agent、worker。console 為主控只有一個，負責提供腳本給其他機器，agent 則是每台機器都有，負責產生 worker 行程，worker 負責模擬使用者進行腳本的動作。若測試計畫須要修改每台機器的行程數量時，就必須修改每台機器的 agent 設定檔，較為麻煩。

雖然以上三個測試工具都沒有辦法像一些商業測試軟體如 LoadRunner 等工具能提供更詳細的數據，更深入的追蹤，但仍然相當不錯。就上手容易度，OpenSTA 最適合剛接觸者使用，但 Apache Jmeter 在熟悉他的腳本形態後，也很容易上手；Grinder 及 OpenSTA 都有提供腳本編輯的功能，可以讓測試者對每個行程進行更細緻的操作，但 OpenSTA 稍遜於 Grinder，除了只支援 windows 平台的問題以及協定只支援 http 外，最大的問題是上次更新時間為 2007 年，已經很久沒有更新了。

5. 測試用例及缺陷追蹤管理工具:TestLink+mantis、Testopia+Bugzilla

建立一套完備的測試用例及缺陷追蹤管理工具可以讓產品開發更有效率，開放源碼工具必須自行將測試用例工具及缺陷追蹤工具整合在一起，雖然不像已完整的整合商用管理工具方便，但多元的組合可以讓開放源碼管理工具更適合團隊的需求。以下介紹兩套不同的組合：TestLink[12]整合 mantis[13] 以及 Testopia[14]整合 Bugzilla[15]。由於測試用例工具差異不大，這裡會著重在缺陷追蹤管理工具的比較。

name	install difficulty	bug label	user management	search interface	report	use friendly
TestLink+mantis	easy	simple	limited	weak(filter)	simple but fast	good
Testopia+Bugzilla	complex in Windows	detailed	flexibility	strong (can save)	detailed, customize	bad

表三 測試用例與缺陷追蹤管理工具

由表三[16]可得出以下訊息：

- a. 安裝容易度：兩者都很容易，但 Bugzilla 於 windows 下安裝略為複雜。
- b. 功能性：mantis 具備一般常用的基本功能，對於客製化功能支援較少，若要變更須修改原始碼；Bugzilla 則可根據需求配置適當的設置，用戶管理也有很大的自由度。在搜尋資料庫中的缺陷的情況，Bugzilla 擁有

較強大的搜尋功能並能把搜尋條件存成連結，相當方便；而 mantis 搜尋方式是利用篩選(filter)的方式，不利過於複雜的搜尋。

- c. 介面：mantis 介面較人性化且支援多國語言；Bugzilla 介面樸實，較不易操作。

綜合以上資訊，測試用例管理工具中 TestLink 有較多元的整合性，可與 mantis、Bugzilla、JIRA 等多種缺陷管理工具搭配；而 Testopia 雖然只能與 Bugzilla 搭配，但它為 Bugzilla 的專屬套件，可以與 Bugzilla 有更完美的配合。mantis 屬於輕量級的缺陷管理工具，適合小型的開發團隊；而 Bugzilla 擁有較多、較強大的功能，較能提供客製化，適合大型開發團隊。由於 mantis 人性化的介面，若缺陷回報系統會提供給一般客戶使用，mantis 的人性化介面設計會是較好的選擇。

6. 結論

Opensourcetesting.org 除了提供各種開放源碼測試工具的資訊外，簡單的上傳程序和提供技術支援的論壇，建立了一個好的平台讓測試人員可以輕易取得適合的測試工具並給予新的開發測試工具一個能快速增加能見度的舞台。

雖然開放源碼測試工具不如商業測試工具能提供較完善的技術支援，但較多元的工具選擇以及可修改程式碼，讓開放源碼測試工具可以更貼近開發產品的測試需求。因為產品的差異性，沒有一個測試工具在某個領域特別強大，必須考慮產品的特性、測試人員的能力，唯有善用合適的測試工具，才能確實達成品質的測試。

7. 參考資料

- [1] opensourcetesting,
<http://www.opensourcetesting.org/>.
- [2] Agile testing - Performance vs. load vs. stress testing,
<http://agiletesting.blogspot.com/2005/02/performance-vs-load-vs-stress-testing.html>.
- [3] Selenium,
<http://seleniumhq.org/>.
- [4] Sahi,
<http://sahi.co.in/w/sahi>.
- [5] Canoo Webtest,
<http://webtest.canoo.com/webtest/manual/WebTestHome.html>.
- [6] M. Guillemot, "Webtest vs Selenium: Webtest wins 13-5", 2007
<http://mguillem.wordpress.com/2007/10/29/>.
- [7] A. Marchetto, F. Ricca, P. Tonella, "A case study-based comparison of web testing techniques applied to AJAX web applications", STTT10(6), p477-492, 2008.
- [8] Apache Jmeter,

- <http://jakarta.apache.org/jmeter/>.
- [9] OpenSTA,
<http://opensta.org/>.
- [10] Grinder,
<http://grinder.sourceforge.net/>.
- [11] B. Julien and D. Nicolas, "An overview of load test tools", 2003,
http://clif.ow2.org/load_tools_overview.pdf.
- [12] TestLink,
<http://testlink.sourceforge.net/docs/testLink.php>.
- [13] mantis,
<http://www.mantisbt.org/>.
- [14] Testopia,
<https://wiki.mozilla.org/Testopia>.
- [15] Bugzilla,
<http://www.bugzilla.org/>.
- [16] 通達人驛站 - 關於 Mantis-3/3 功能比較與個人觀點,
<http://www.prudentman.idv.tw/2009/01/mantis-33.html>.