

공개 SW를 이용한 자동화 테스트의 시작

STA테스팅컨설팅 ● 자동화솔루션사업팀 ● 박문석 팀장



WIKIPEDIA
The Free Encyclopedia

In [software testing](#), **test automation** is the use of special [software](#) (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes.^[1] Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually.



다양한 영역의 자동화 도구



다양한 영역의 자동화 도구

도구 유형		의미
인프라 도구	요구사항 관리	각각의 요구사항에 대하여 식별 가능한 ID를 부여하고 변경이력과 추적성이 유지될 수 있도록 관리해 주는 도구
	형상 관리	소스 코드에 대한 Branch 생성, Merge 등의 변경 관리와 다양한 산출물의 버전 및 이력관리를 할 수 있는 도구
	테스트 관리	테스트 케이스 설계, 테스트 계획 수립, 테스트 케이스 할당, 실행 결과 입력, 리포트 작성 등 테스트 전반을 지원하는 관리 도구
	결함 관리	테스트 도중에 발견 된 결함을 등록하여 식별 가능한 ID를 부여하여 추적성과 수정 과정에 대한 이력관리를 할 수 있는 도구 형상 관리 도구와 연동하여 결함에 대한 수정 코드의 추적성을 확보할 수 있도록 해주며 결함 뿐만 아니라 일반적인 워크아이템으로도 이용
	빌드 / 배포	소스 코드를 빌드하여 바이너리를 생성하고 이를 배포해주는 도구
정적 분석 도구	코딩 룰	개발 코딩 표준 준수 여부 검토 및 예방 차원의 코드 결함 검출
	코드 복잡도	소스코드 내에서 실행 경로의 수를 측정하여 복잡도 판단(순환복잡도)
	잠재적 결함	수집된 오류 형태를 기반으로 잠재적 오류 가능성 여부를 판단
	상관 관계 분석	Reverse Engineering을 기반으로 하여 내부 Class 간의 참조형태의 분석하여 코드의 유지 보수성을 향상 시키기 위한 도구
	동료 검토 도구	소스 코드 및 다양한 산출물을 함께 검토함으로써 문제점 조기 발견

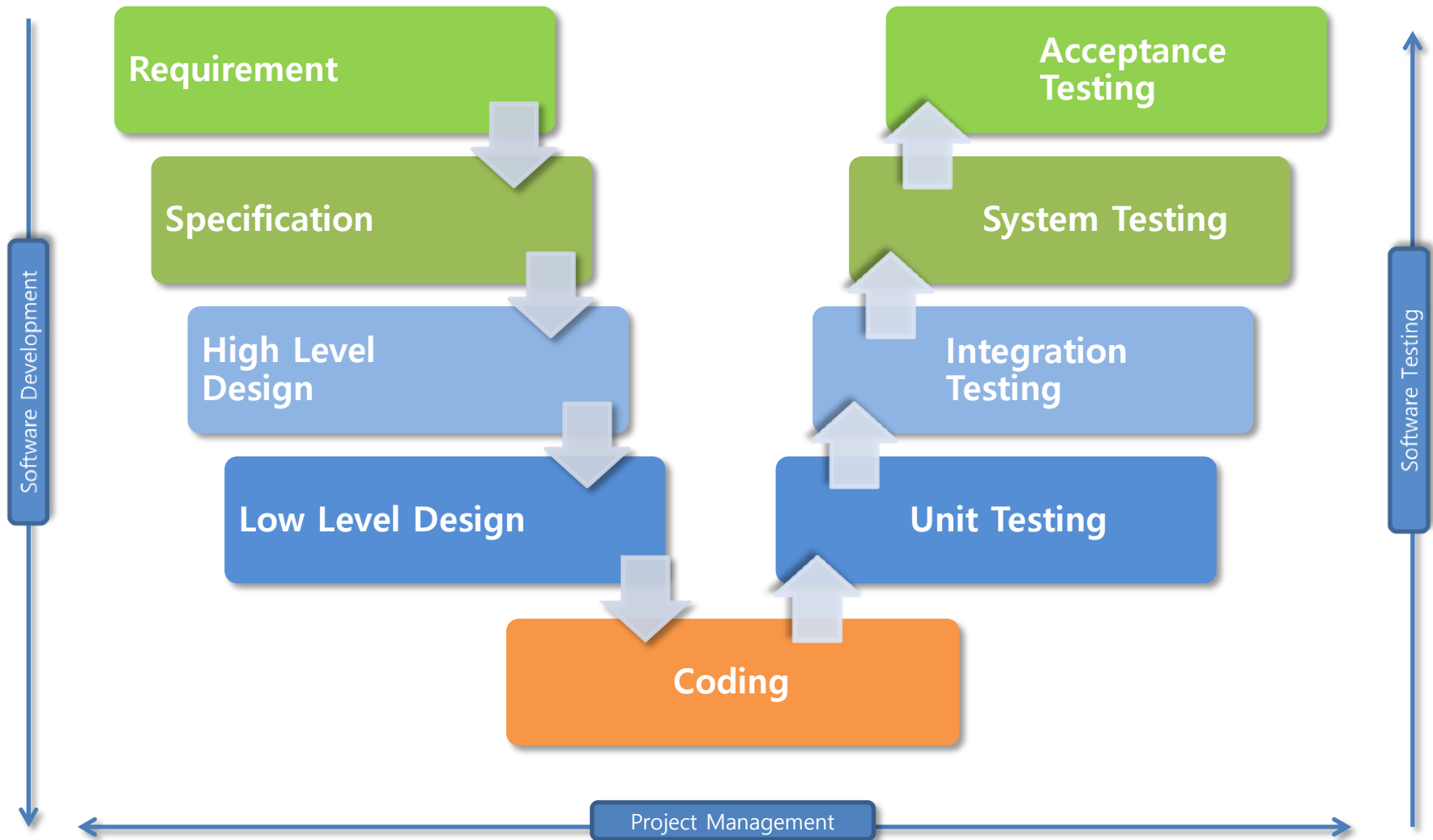


다양한 영역의 자동화 도구

도구 유형	의미
단위 테스트 도구	단위 모듈에 대한 테스트 케이스를 구현하고 실행하는 도구
단위 테스트 설계 도구	단위 테스트의 커버리지를 높일 수 있도록 테스트 케이스를 자동으로 생성해주는 도구
커버리지 측정 도구	구문, 결정, 조건, 조건/결정, 변경조건/결정, 다중조건 등의 다양한 커버리지 측정
화면 기반의 실행 테스트 도구	화면상의 오브젝트 인식 또는 이미지 비교를 기반으로 하여 실행 자동화 테스트 수행하는 도구
전문 기반의 실행 테스트 도구	시스템 간 사전 정의된 전문을 기반으로 하여 실행 자동화 테스트를 하거나 가상화를 통하여 테스트를 지원하는 도구
보안 테스트 도구	일반적인 위험과 취약점을 테스트 해주는 도구
API 테스트 도구	웹 서비스 또는 라이브러리에 구현된 API 테스트를 지원하는 도구
성능 테스트 도구	다수의 접속자에 대한 운영 시스템의 성능을 측정하거나 어플리케이션의 메모리 누수, 함수 별 실행 시간 등을 측정해 주는 도구
성능 모니터링 도구	운영 시스템의 성능을 실시간으로 모니터링 해주는 도구
결함 자동 리포트 도구	Mobile Device 소스코드에 import 하여 결함 발견 시 device 상에서 바로 리포트 할 수 있게 해주고 Crash Log를 자동으로 서버로 등록해 주는 도구



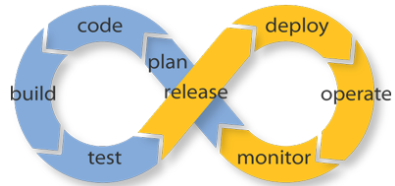
프로젝트 단계별 자동화 도구



프로젝트 단계별 자동화 도구



SW 테스트에 있어 자동화 도구의 필요성



새로운 개발 방법론 도입



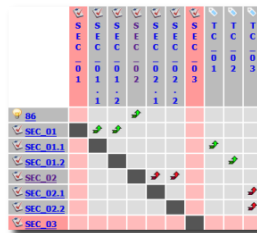
프로젝트의 비용절감 요구



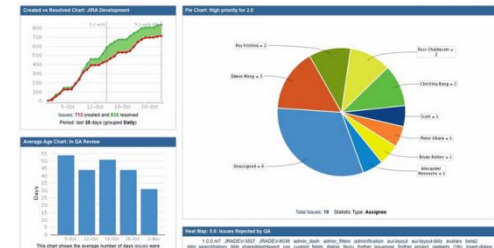
Time To Market의 중요성



분업화된 개발



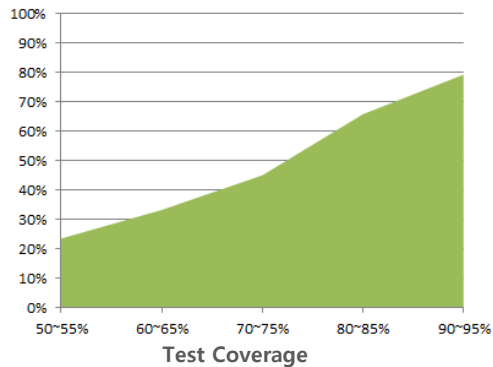
산출간의 추적성 확보



프로젝트의 진행사항과 예측

효과적인 자동화 접근 방법

Fault detection effectiveness



**Automated
Regression Test**

반복적인 테스트의 자동화

테스트 시점 당길 수 있는 자동화

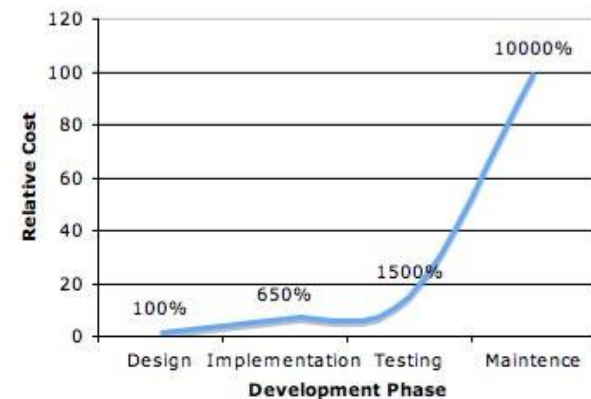
코드 기반의 자동화

인프라 기반의 자동화



- Requirement Management
- Configuration Management
- Build & Distribute
- Test Management
- Defect Management

Cost to correct bugs



자동화 도구 도입 시 오픈 소스 도구의 장단점

장점:

- 도입 및 확장 시 라이선스 비용이 들어가지 않음
- 현재 상황에 맞는 도구를 빠르게 도입하여 사용할 수 있음
- 프로그래밍 능력이 있는 경우, 스스로 유지 보수 및 추가 기능 개발 가능
- 더욱 좋은 제품이 나올 경우 새롭게 도입하는데 부담이 적음
- 많은 정보들이 인터넷 상에 공개되어 있음

단점:

- 문제 발생시 기술지원이 되지 않음
- 새로운 버전에 대한 안전성 문제가 발생할 수 있음
- 개발자의 상황에 따라 추가 버전의 개발이 중단 될 수 있음
- 특정 영역에서는 도구에 의한 결과물이 신뢰성을 인정 받지 못하는 경우가 있을 수 있음
- 동일한 기능의 상용 도구 보다 기능 범위가 작을 수 있음

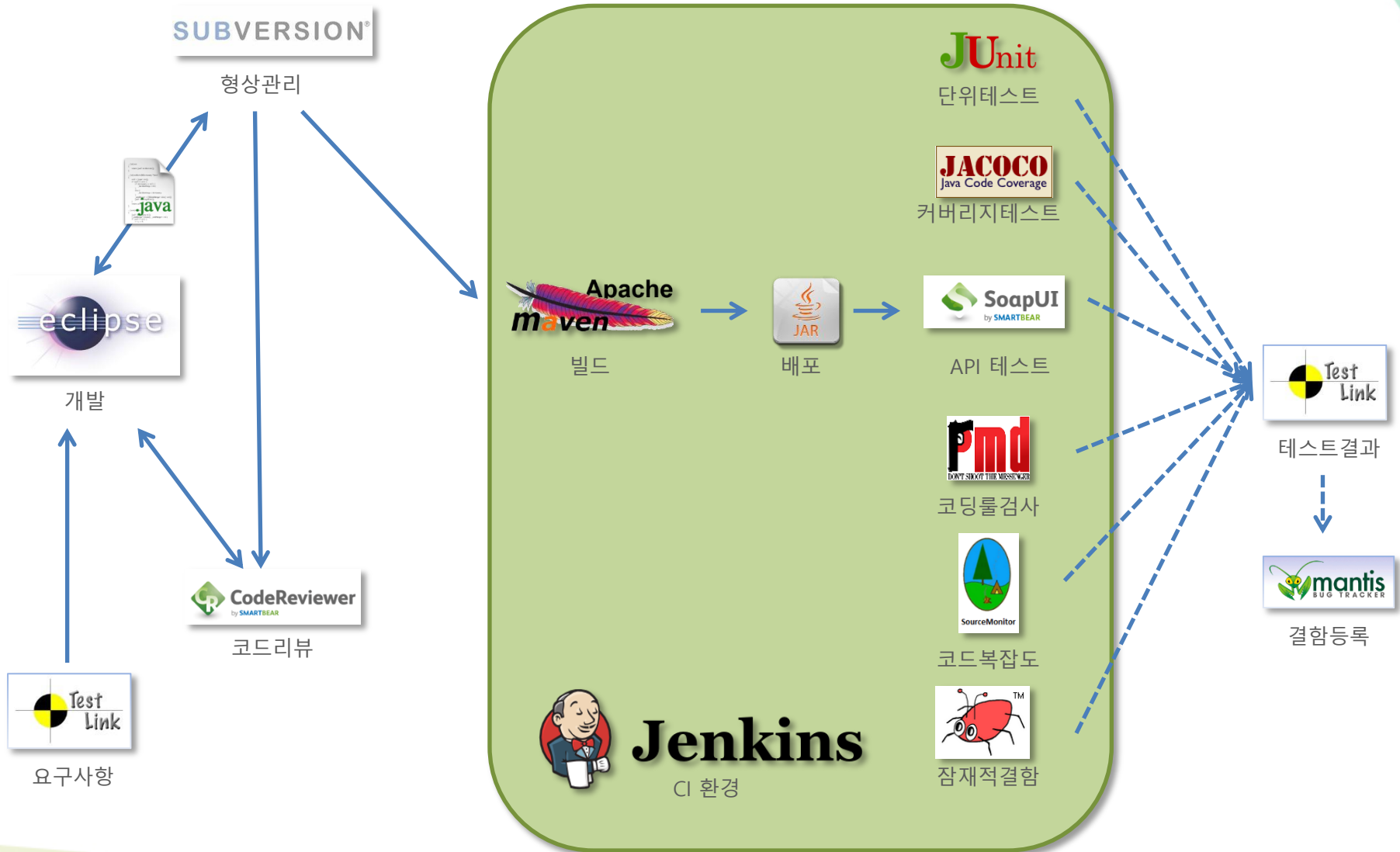


오픈 소스를 프로젝트 환경 구성 예

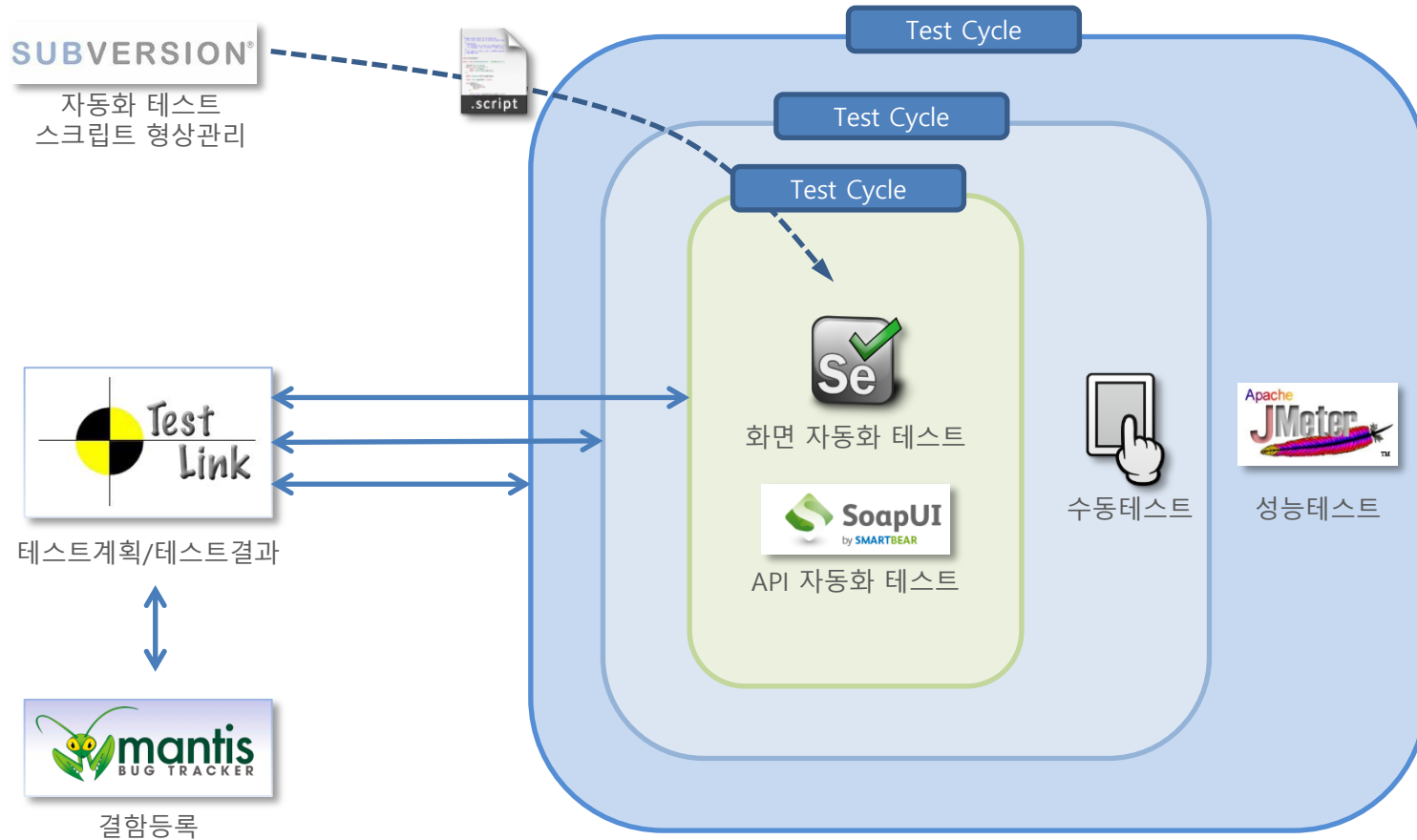
JAVA 기반의 Web Service 개발



오픈 소스를 이용한 프로젝트 환경 구축 예



오픈 소스를 이용한 프로젝트 환경 구축 예



참고자료

- TEST LINK
<http://www.testlink.org/>
- MANTIS BT
<https://www.mantisbt.org/>
- ECLIPSE
<https://eclipse.org/>
- JUNIT
<http://junit.org/>
- SUBVERSION
<https://subversion.apache.org/>
- JENKINS
<https://jenkins-ci.org/>
- MAVEN
<https://maven.apache.org/>
- CODE REVIEWER
<https://codereviewer.org/>
- JACOCO
<http://www.eclemma.org/jacoco/>
- PMD
<http://pmd.sourceforge.net/>
- SOURCE MONITOR
<http://www.campwoodsw.com/sourcemonitor.html>
- FINDBUGS
<http://findbugs.sourceforge.net/>
- SELENIUM
<http://www.seleniumhq.org/>
- SOAPUI
<http://www.soapui.org/>
- JMETER
<http://jmeter.apache.org/>



Thank You