

테스트 툴 자료조사

- 테스트 관리 툴 -
 - 1) Zephyr
 - 2) TestRail
 - 3) Qase
 - 4) Testpad
 - 5) TestLink
- 웹 테스트 자동화 툴 -
 - 6) LambdaTest
 - 7) TestComplete
 - 8) Selenium
 - 9) HP UFT (aka QTP)
- 웹 API 테스트 툴 -
 - 10) Tricentis
 - 11) SOAP UI
- 결함 추적 관리 툴 -
 - 12) BugZilla
 - 13) JIRA
- 브라우저 호환성 툴-
 - 14) Ghostlab
 - 15) Sauce Labs
 - 16) Browser-Stack
- 성능 테스트 툴 -
 - 17) LoadUI Pro
 - 18) Apache Jmeter
- 요구사항 관리 툴 -
 - 19) Case Complete
- 테스트데이터 준비툴-
 - 20) ARX
- 정적 분석 도구 -
 - 21) PMD

1) Zephyr

Zephyr은 모든 규모의 애자일 팀을위한 엔드 투 엔드 솔루션을 제공하는 최고의 웹 판매 도구입니다.

더 나은 소프트웨어를 출시하는 데 필요한 유연성, 가시성 및 통찰력 확보

Zephyr의 주요 기능

- . JIRA, Confluence, Jenkins, Bamboo 등과의 원 클릭 통합
- . 클라우드, 서버 및 데이터 센터 배포 옵션
- . 고급 분석 및 DevOps 대시 보드
- . 연간 약정 불필요

웹 테스트 자동화 툴

8) Selenium

Selenium은 오픈 소스 웹 자동화 도구이며 현재 시장에서 가장 널리 사용되고 널리 사용되는 도구입니다.

Windows, Mac 및 Linux와 같은 여러 OS와 Firefox, Chrome, IE 및 헤드리스 브라우저와 같은 브라우저에서 자동화 할 수 있습니다.

Selenium 테스트 스크립트는 Java, C #, Python, Ruby, PHP, Perl 및 JavaScript와 같은 프로그래밍 언어로 작성 될 수 있습니다.

Selenium은 브라우저 추가 기능인 Selenium IDE와 함께 레코드 및 재생 기능을 제공합니다.

강력한 Selenium WebDriver는보다 복잡한 고급 자동화 스크립트를 작성하도록 도와줍니다.

웹 API 테스트 툴

10) Tricentis

Tricentis는 Agile 및 DevOps와 보조를 맞추기 위해

테스트를 가속화하는 Continuous Testing 플랫폼을 제공합니다.

Tricentis Tosca의 주요 API 테스트 기능은 다음과 같습니다.

- . HTTP JMS, AMQP, Rabbit MQ, TIBCO EMS, SOAP, REST,
- . IBM MQ, NET TCP 등 다양한 프로토콜을 지원합니다.
- . 애자일 및 DevOps주기에 통합
- . 스크립트 기반의 유지 관리가 용이 한 모델 기반 테스트 자동화를 사용합니다.
- . API 테스트를 모바일, 크로스 브라우저, 패키지 앱 등에서 사용할 수 있으므로 엔드 투 엔드 테스트가 가능합니다.

Tricentis의 400 명 이상의 고객은 ExxonMobil, HBO, Whole Foods, Toyota, Allianz, BMW, Starbucks, Deutsche Bank, Lexmark, Orange, A & E, Vantiv, Vodafone, Telstra 및 UBS와 같은 상위 500 개 브랜드의 글로벌 이름을 포함합니다.

브라우저 호환성 테스트 툴

16) Browser-Stack

브라우저 스택을 사용하면 데스크톱 및 모바일 브라우저에서

웹 기반 브라우저 테스트를 수행 할 수 있습니다.

클라우드 기반이므로 설치가 필요하지 않으며

사전 설치된 개발자 도구는 빠른 브라우저 간 테스트 및 디버깅에 유용합니다.

브라우저 스택을 사용하면 프록시, 방화벽 및

Active Directory를 지원하는 포괄적 인 테스트 환경을 설정할 수 있습니다.

그것은 오페라 모바일, 안드로이드, 윈도우 (XP, 7 및 8), iOS,

OSX 스노우 레오파드, 사자 및 산 사자 등을 지원합니다.

브라우저 스택을 사용하면 페이지를 원격으로 테스트 할 수 있습니다.

성능 테스트 툴

18) Apache Jmeter

JMeter는 다양한 서비스의 성능을 분석하고 측정하기 위한 성능 및로드 테스트에 사용할 수 있는 오픈 소스 도구입니다. 이 도구는 주로 웹 및 웹 서비스 응용 프로그램에 사용됩니다.

기능:

이 툴은로드 테스트를 위한 최첨단 인프라를 요구하지 않으며 단일 컨트롤러로 관리되는 여러로드 인젝터를 지원합니다

요구사항 관리 툴

19) Case Complete

Case Complete는 애자일 요구 사항 관리 도구입니다.

적은 노력으로 웹 테스트 프로젝트의 요구 사항과 사례를 캡처 할 수 있습니다.

주요 특징 및 기능 :

- . 보관 요구 사항
- . 정의되지 않았거나 누락되었거나 정의 된 요구 사항 식별
- . 요구 사항의 추적 성
- . 테스트 관리 도구와 인터페이스
- . 요구 사항 범위

테스트 데이터 준비 툴

20) ARX

테스트 데이터 익명화는 은행과 같이 민감한 데이터가 포함 된 웹 사이트를 테스트 할 때 중요합니다.

ARX는 강력한 익명화 도구입니다.

주요 특징 및 기능 :

- . 파일 또는 데이터베이스에서 선택된 데이터 레코드 추출
- . 데이터 익명화
- . 임의의 데이터로 채워진 새 레코드 작성
- . 템플릿에서 많은 수의 유사한 레코드 만들기
- . 오픈 소스입니다

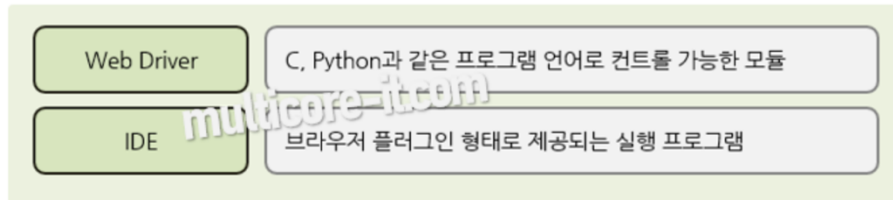
출처:

<http://blog.naver.com/PostView.nhn?blogId=genycho&logNo=221970592665&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=false&from=postView> [구랭구랭이님 네이버 블로그]

셀레니움에 대해 더 알아보자!

웹 테스트 자동화 도구, 셀레니움

셀레니움은 프로그램을 통해 사람과 같이 브라우저를 제어할 수 있는 프로그램이다. 셀레니움은 웹 드라이버와 IDE 형식으로 제공되는데 웹 드라이버(Web Driver)는 C, 파이썬, 루비와 같은 프로그램 언어와 API 수준의 인터페이스를 제공하는 모듈이다. 웹 드라이버를 사용하면 프로그래밍을 통해 웹 스크롤러와 테스트 자동화 도구를 개발할 수 있다.



셀레니움 구성

셀레니움 IDE

브라우저 플러그인 형태로 제공되는 실행프로그램.

간단한 클릭 뿐만 아니라 마우스를 이동하는 것과 로그인까지 사람이 하는 모든 동작을 기록해서 저장하고 나중에 똑같이 재현할 수 있다.

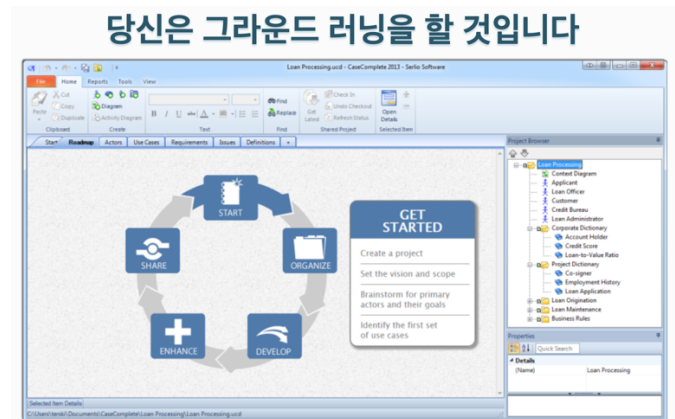
Record & Play 방식의 자동화 테스트가 구현 가능하다.

셀레니움 장점

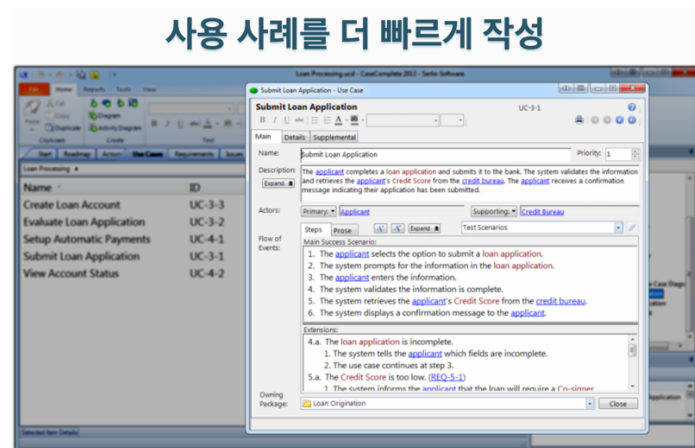
통합 테스트 시점에 실행한 내역을 그대로 저장하고 있다가 오류 수정 후 테스트의 참여 없이 자동화 테스트를 수행할 수 있다. 고비용의 테스트 작업을 최소화하여 저비용으로 고품질의 소프트웨어를 만들 수 있다.

출처 : <https://multicore-it.com/64> [깊고넓은지식창고]

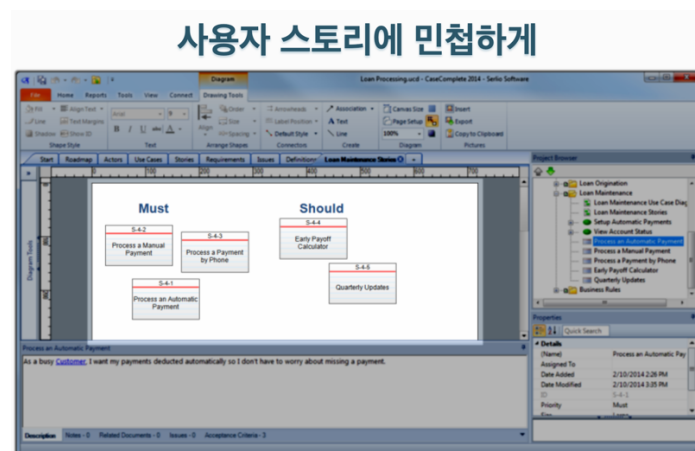
요구사항 관리 툴, Case Complete에 대해 더 알아보자!



로드맵은 다양한 요구 사항을 수집하기 위한 일관된 접근 방식을 안내합니다.

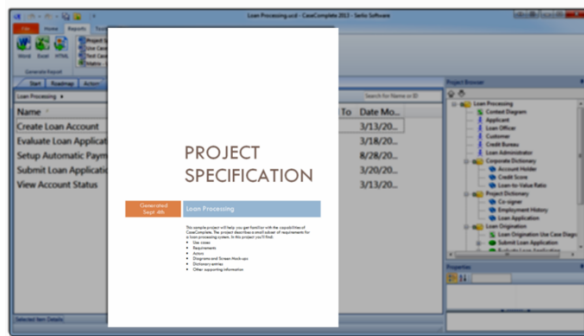


매끄럽고 직관적인 인터페이스로 사용 사례를 쉽게 작성할 수 있습니다.



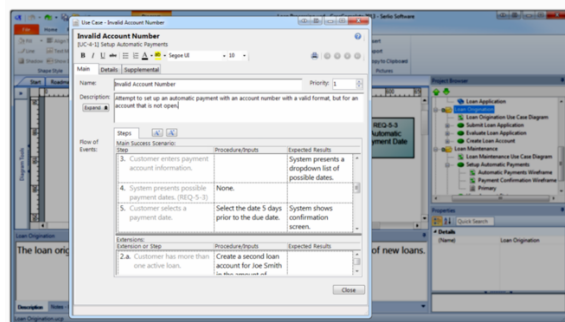
당신의 스프린트를 구성하고 계획하기 위해 유저 스토리를 만드세요.

다시 문서를 작성하지 마십시오



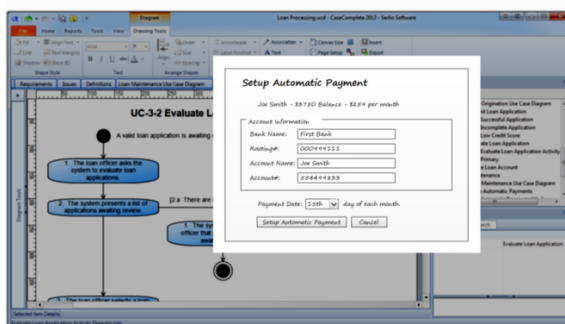
요구사항에서 자동으로 문서를 생성합니다.

테스트 케이스로 검증



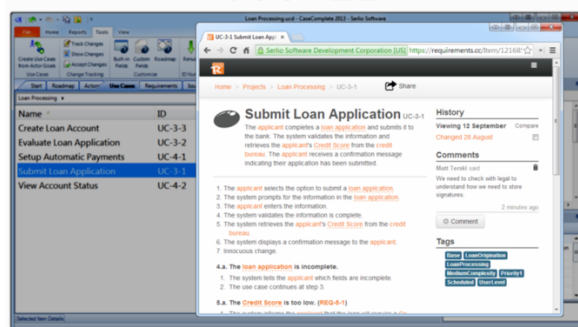
요구사항을 검증 하기 위해 테스트 케이스를 추가하십시오. 테스트 관리 도구와 통합하거나 테스트 케이스 문서를 생성합니다.

다이아그램으로 명확히



Use Case에서 다이어그램을 생성합니다.

즉각적인 협업

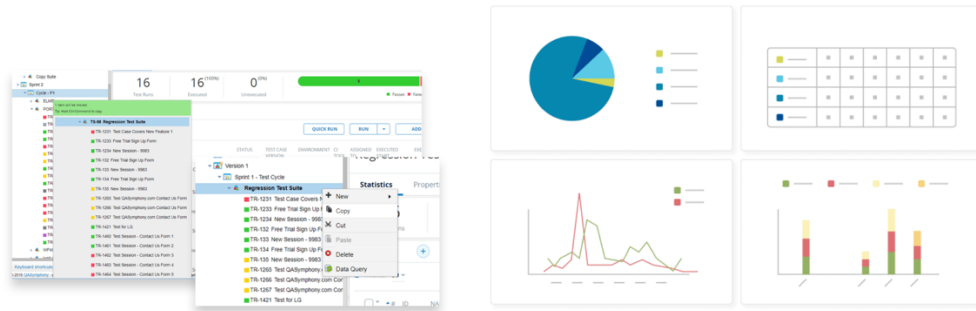


온라인으로 공동작업 하십시오. 이해 관계자가 즉시 검토하고 댓글을 달 수 있습니다.

출처 : Case Complete 공식 홈페이지

웹 API 테스트 툴, Tricentis에 대해 알아보자!!

우리의 접근 방식은 **완전히 자동화되고 완전히 코드가 없으며 AI에 의해 지능적으로 구동됩니다**. 민첩한 개발과 복잡한 엔터프라이즈 앱을 모두 해결하여 소프트웨어 제공 속도를 높이고 비용을 절감하며 품질을 개선합니다. 이제 테스트는 며칠 또는 몇 주가 아닌 몇 분 또는 몇 시간이 걸리며 훨씬 더 비용 효율적이고 매우 정확하여 엔터프라이즈 위험 범위를 증가시킵니다. 이러한 종류의 엔터프라이즈 자동화는 엄청난 효율성과 속도 향상을 가져옵니다. 그리고 디지털 혁신의 힘은 매우 분명해졌습니다.

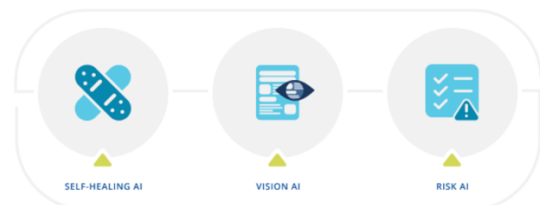
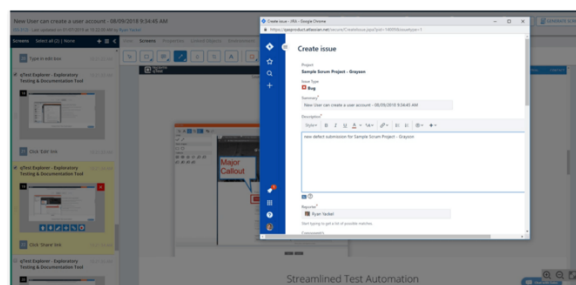


모든 테스트 단계 지원

공통 정보 모델을 사용하면 개발자, 테스터 및 비즈니스 사용자가 단위에서 성능에 이르기까지 모든 테스트 단계에 대해 공통 테스트 케이스, 계획, 데이터 및 아티팩트 세트에 대해 협업 할 수 있습니다.

분석 및 통찰력 통합

보고 및 데이터웨어 하우스 통합을 통해 한 곳에서 모든 테스트 활동에 대한 포괄적인 가시성과 통찰력을 얻으십시오.

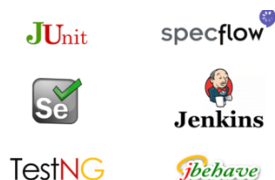


프로덕션 배포를 다시 요구 사항으로 추적

동급 최강의 Jira 및 DevOps 에코 시스템 통합을 통해 프로덕션 배포를 비즈니스 및 기능 요구 사항으로 거슬러 올라갑니다.

더 스마트하고 빠른 테스트를 위해 ML 적용

Vision AI, 자가 치유 AI 및 Risk AI와 같은 고급 ML 기술을 통해 팀은 더 스마트하고 빠르게 테스트하고 애플리케이션 변경으로 인한 비즈니스 위험을 줄일 수 있습니다.



지속적인 테스트를 CI / CD 도구 체인과 통합

Tricentis 플랫폼은 광범위한 오픈 소스 및 상용 기술을 지원하고 조정합니다.

출처 : Tricentis 공식 홈페이지

브라우저 호환성 테스트 툴, BrowserStack 에 대해 알아보자!

BrowserStack 이란 무엇입니까?

브라우저 스택은 클라우드 기반 테스트 도구입니다. 이 도구를 사용하여 개발자와 테스터는 다양한 웹 브라우저에서 다양한 웹 및 모바일 애플리케이션의 브라우저 간 테스트를 수행 할 수 있습니다.

왜 BrowserStack 인가?

실시간으로 다양한 버전의 여러 운영 체제를 사용하는 여러 브라우저에서 애플리케이션을 테스트하기가 어렵습니다.

다른 운영 체제를 설치하기 위해 많은 컴퓨터 기계를 구입하는 데 많은 비용이 듭니다. 오늘날 시장에서 구할 수있는 모든 안드로이드 및 iOS 모바일 장치를 구입하는 것은 훨씬 더 어렵습니다. 이러한 이유로 모든 장치를 구입하고 설정하는 것은 실용적이지 않습니다.

따라서 운영 체제, 브라우저 및 다른 버전 간 전환의 복잡성을 피하기 위해 BrowserStack 을 사용합니다.

모바일 애플리케이션 테스트와 관련하여 현재 사용 가능한 모든 모바일 장치를 구매하지 않아도 됩니다. 브라우저 스택은 매우 유연하고 확장 가능합니다. Browser Stack 의 도움으로 언제 어디서나 테스트 할 수 있습니다.

브라우저 스택을 원격 랩으로 사용할 수 있으며이를 실제 데스크톱 브라우저로 사용할 수 있습니다. 무료 자바 스크립트 단위 테스트를 제공하고 750 개의 구성을 지원합니다. 브라우저 스택을 사용하기위한 설정이 필요하지 않습니다. 브라우저 스택 URL 과 로그인 자격 증명을 사용하여 모든 독립 컴퓨터에서 직접 사용할 수 있습니다.

BrowserStack 의 기능

서로 다른 운영 체제에서 서로 다른 브라우저로 브라우저 간 테스트.

모바일에서 네이티브 앱 테스트.

하이브리드 애플리케이션 테스트.

웹 및 모바일 애플리케이션을 자동화합니다.

실제 장치 테스트에 대한 BrowserStack 테스트 비교

브라우저 스택 테스트	실제 장치 테스트
경제적 : 다른 장치와 브라우저를 전환하여 단일 장치에서 응용 프로그램을 테스트 할 수 있습니다.	경제적이지 않음 : 테스트를 위해 시장에서 사용 가능한 모든 장치를 구입해야 합니다.
매우 유연함 : 단일 플랫폼에서 브라우저 간 테스트 및 모바일 애플리케이션 테스트를 수행 할 수 있습니다.	어디를 가든 여러 장치를 휴대해야 합니다.
OS 라이선스를 구매할 필요가 없습니다.	OS 라이선스를 구입해야 합니다.
빠른 : OS 및 브라우저를로드 할 때까지 기다릴 필요가 없습니다.	느린 : OS 및 브라우저를로드하려면 기다려야 합니다.
가상 머신을 사용할 필요가 없습니다.	가상 머신을 사용해야 합니다.

출처 : BrowserStack Tutorial

https://ko.myservername.com/browserstack-tutorial#What_Is_BrowserStack

성능 테스트 툴, Apache Jmeter에 대해 더 알아보자!

1. Apache Jmeter 의 개요

Jmeter는 100% 순수 Java로 구현된 성능테스트 프로그램입니다.

2. Apache Jmeter의 기능 및 장단점

기능/장단점	설명
다양한 APP/Server/ Protocol지원	Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...) SOAP / REST Webservices FTP Database via JDBC LDAP Message-oriented middleware (MOM) via JMS Mail - SMTP(S), POP3(S) and IMAP(S) Native commands or shell scripts TCP Java Objects
Test IDE의 전체 기능	Recording, Building, Debugging
Command Line지원	Load Test를 command line으로 하여 Controller의 부하감소 및 대용량가능
Dynamic HTML report	전체 결과를 HTML report 로 저장
100% 순수 Java로 구현	Java의 Multi OS기능 지원
높은 확장성	Plugin Manager를 통한 다양한 Plugin 설치 Open Source로 소스코드를 통한 확장 Script기반 Sampler를 통한 다양한 기능 개발

출처 : <https://jangiloh.tistory.com/73> [파이어맨 이야기 티스토리]

JMeter 의 핵심기능은 샘플러(Sampler)이다. 샘플러가 성능 테스트를 위한 부하를 발생시키는 역학을 하며 각각은 샘플러에서 실행할 수 있는 특별한 프로토콜에 맞는 값을 설정할 수 있다. 그리고 샘플러의 실행 결과가 수집되어서 성능 테스트에 대한 결과를 분석할 수 있는데, 요청에 대한 성공/실패, 소요 시간 등에 대한 정보를 기록한다. JMeter 에서 제공하는 주요 샘플러는 다음과 같다.

Http Request

HTTP/HTTPS 프로토콜을 이용해서 성능 테스트를 수행할 때 사용 (JMeter 에서 가장 많이 사용되는 샘플러)

JDBC Request

데이터베이스의 특정 SQL 의 성능 테스트를 수행할때 사용

HTTP Request Sampler 를 이용해서 테스트를 수행한 후 특정 JDBC 에 대한 테스트가 필요할때 종종 사용.

JDBC 테스트를 위해서는 별도의 JDBC 드라이버를 JMeter 의 클래스 패스에 추가해야 한다.

출처: <https://12bme.tistory.com/503> [길은 가면, 뒤에 있다.]

QA

Quality Assurance Metrics

메트릭스가 없으면 우리는 무엇을 측정해야 하는지 모른다.

메트릭스는 QA를 할 때 당신이 얼마나 잘하고 있는지 보여준다.

Software Metrics는 세가지로 분류된다.

1. Product Metrics

Size, Complexity, Design Feature, Performance, Quality Level 등 제품에 대한 묘사

2. Process Metrics

소프트웨어 개발 및 유지 보수 활동을 개선하는데 사용할 수 있다.

3. Project Metrics

프로젝트의 특성과 실행 등을 설명한다. 소프트웨어 개발자 수, 소프트웨어 수명주기 동안의 인력 패턴, 비용, 일정 및 생산성

Software Quality Metrics

소프트웨어를 품질적인 측면에서 바라보는 메트릭스.

Product Metrics와 Process Metrics에 더 가깝다.

이 또한 세가지 카테고리로 나뉜다.

Product Quality Metrics

In-process Quality Metrics

Maintenance Quality Metrics

Product Quality Metrics은 다음과 같은 내용들이 포함된다.

Mean Time to Failure : 실패 사이들의 시간. 실패가 얼마나 자주 발생하는지 실패 주기를 의미한다.

Defect Density : 결함 밀도. 코드 길이, function point등으로 표현되는 소프트웨어의 크기에 비교해 결함을 측정한다. 주로 상업적인 소프트웨어 시스템에 사용된다.

Customer Problem : 고객이 제품을 사용할 때 발생하는 문제점을 측정한다.

Customer Satisfaction : 매우 만족, 만족, 보통, 불만족, 매우 불만족의 5척도로 고객 만족도 설문 조사를 시행한 데이터로 측정한다.

QA Metrics Examples

Number of tests in a certain time period : $\text{Number of tests run} / \text{Total time}$

Test design efficiency : $\text{Number of test designed} / \text{Total time}$

Test review efficiency : $\text{Number of test reviewed} / \text{Total time}$

Number of bugs per test : $\text{Total number of defects} / \text{Total number of tests}$

QA를 위한 질문

- 테스트를 얼마나 오래할 것인가?
- 테스트에 얼마나 많은 비용이 요구되는가?
- 버그의 심각성은 어느 수준인가?
- 얼마나 많은 버그가 해결되었는가?
- 각 버그들은 어떤 상태인가? (해결됨, 재발생, 발생가능성 있음)

- 이 소프트웨어는 얼마나 테스트 되었는가?
- 주어진 시간 내에 테스트가 완료될 수 있는가?
- 테스트의 양이 적당한가? 같은 단위 시간 동안 더 많은 테스트가 진행될 수 있는가?

Absolute QA Testing Metrics

이것들을 이용해 위에 있는 **QA Metrics Examples**에 있는 저 항목들을 계산해낼 수 있다.

- 총 테스트 케이스 수
- 통과한 테스트 케이스 수
- 실패한 테스트 케이스 수
- 차단된 테스트 케이스 수
- 확인된 버그 수
- 허용된 버그 수
- 거부된 버그 수
- 지연된 버그 수
- 치명적인 버그 수
- 결정된 테스트 수
- 실제 테스트 시간
- 출시 후 발견된 버그 수

원하는 결과를 도출 하는 QA 프로세스를 계획하고 실행 하려면 올바른 메트릭을 고정하고 정확하게 사용하는 것이 중요 합니다. Agile 프로세스의 QA 메트릭은 관리자가 각 스프린트에서 달성되는 가장 작은 목표에 세심한주의를 기울여야하기 때문에 특히 중요합니다. 세련되고 구체적인 메트릭은 테스터가 추적을 유지하고 정확히 어떤 숫자에 도달해야하는지 파악하는 데 도움이됩니다.

Refactoring

리팩토링이란?

외부 동작을 바꾸지 않으면서 내부 구조를 개선하는 방법.

코드가 작성된 후에 디자인을 개선하는 작업.

모든 것을 미리 생각하기 보다는 개발을 하면서 지속적으로 좋은 디자인을 찾는다.

메소드 내의 지역변수와 파라미터를 주의 깊게 볼 필요가 있다.

값이 수정되지 않는 변수는 파라미터로 넘길 수 있다.

값이 수정되는 변수는 주의가 필요하다. 변화되는 부분을 함수로 추출하여 리턴 값으로 돌려줄 수 있다.

사람이 이해하기 쉽도록 네이밍을 중요시한다.

클래스의 메소드는 클래스의 변수를 사용해야 한다.

임시 변수는 가능하면 제거하는 것이 좋다.

임시 변수를 하나의 메소드로 구현하여 리팩토링 하자.

리팩토링 하는 과정에서 알고리즘적으로 성능이 더 안좋아 질 수 있다. 하지만 어느 알고리즘 부분이 수정되어야 하는지 명확하게 보이게 되어 최적화하기 쉬운 형태로 변경되는 장점이 있다.

다른 객체의 속성을 기반으로 한 switch문은 좋지 않다. 그 부분을 함수로 추출하여 자신의 데이터를 사용하는 것으로 리팩토링 해야 한다.

메소드에서 두개 이상의 클래스 분수를 사용하고 있다면 변화하기 쉬운 쪽으로 메소드를 이동시켜 변화의 폭을 최대한 작게 하자.

기능이 한쪽으로 모여진 클래스를 추상화하여 상속관계로 표현할 수 있다.

테스트 -> 리팩토링 -> 테스트 -> 리팩토링 -> 테스트 -> ... 점진적인 개선

리팩토링의 목적

소프트웨어를 보다 이해하기 쉽고 수정하기 쉽도록 만드는 것이다.

겉으로 보이는 소프트웨어의 기능을 변경하지 않는다.

기능을 추가할 때는 기존 코드를 건드릴서는 안되고 단지 새로운 기능만 추가해야 한다.

리팩토링을 할 때는 기능을 추가해서는 안되고 단지 코드의 구조변경에만 신경써야 한다.

리팩토링을 왜하는가?

코드의 디자인을 유지하도록 도와준다.

중복을 제거함으로써 각각의 작업에 대한 코드가 오직 한 곳에만 있게 해준다.

소프트웨어의 디자인을 개선시킨다.

소프트웨어를 더 이해하기 쉽게 만든다.

버그를 찾도록 도와준다.

프로그램을 빨리 작성하도록 도와준다.

리팩토링을 언제하는가?

기능을 추가할 때

버그를 수정해야 할 때

코드 검토를 할 때

어떤 작업을 하는데 계속 시간이 부족한 듯 느껴진다면 보통 리팩토링이 필요하다는 신호다

리팩토링 방법들

1. Extract Method

긴 메소드의 일부 코드를 새 메소드로 만드는 것

이해를 위해 주석을 이용해서 필요한 부분을 분리한다.

메소드명은 짧고 기능을 잘 설명할 수 있어야 한다.

일련의 과정을 block으로 관리한다.

긴 메소드의 코드를 맥락없이 block으로 나누는 것은 리팩토링이 아니다.

작은 기능, 일련의 과정 단위로 모듈화 하여 가독성을 위해 주석으로 분리하고, 명백히 나누어 짧고 분명한 이름의 새 메소드를 만든다.

2. Move Method

한 메소드가 정의된 클래스의 위치가 적절한가?

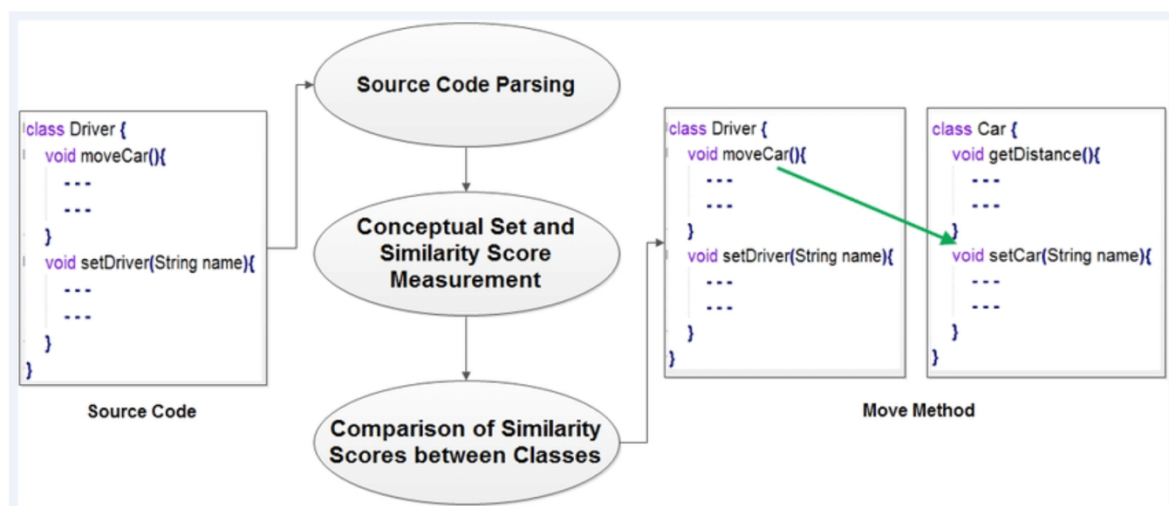
객체의 상태를 전혀 사용하지 않을 경우

다른 클래스에서 더 많이 사용될 경우

메소드의 정의 위치(클래스)를 변경하는 방법

기존 클래스에서는 위임호출로 유지하거나 완전히 제거를 고려한다.

예시) 출처 : 이끼의 생각 <https://ikkison.tistory.com/83>



3. 불필요한 임시 변수 제거

임시 변수는 대부분 메소드로 대체할 수 있다.

임시 변수의 사용은 코드의 길이를 증가시킬 수 있으므로 주의해야한다.

임시변수를 메소드화 하면 코드 길이 축소 및 재활용에 좋다.

임시 변수 제거는 Extract Method 리팩토링을 수월하게 해준다.

4. 분기 제거 Remove Control Flag

일종 분기처리들은 조건의 추가 및 삭제에 유연하지 못하다.
Control Flag 사용을 지양한다.

리팩토링을 하지 말아야 할 때

설계가 크게 변경된다면 리팩토링 보다는 재작성이 더 좋을 수 있다.
단, 대부분의 기능이 정상적으로 작동 해야한다.
마감시간이 다 되었을 경우. 고객과의 약속이 더 중요하다.

전략 패턴

소프트웨어 실행 중에 알고리즘을 선택할 수 있게 하는 소프트웨어 디자인 패턴

1. 특정한 계열의 알고리즘들을 정의한다.
2. 각 알고리즘을 캡슐화 한다.
3. 이 알고리즘들을 해당 계열 안에서 상호 교체가 가능하게 만든다.