



System Testing & Static Analysis

2021 Software V&V

Team #7

김승환 김유진 이현우 전도현



Index

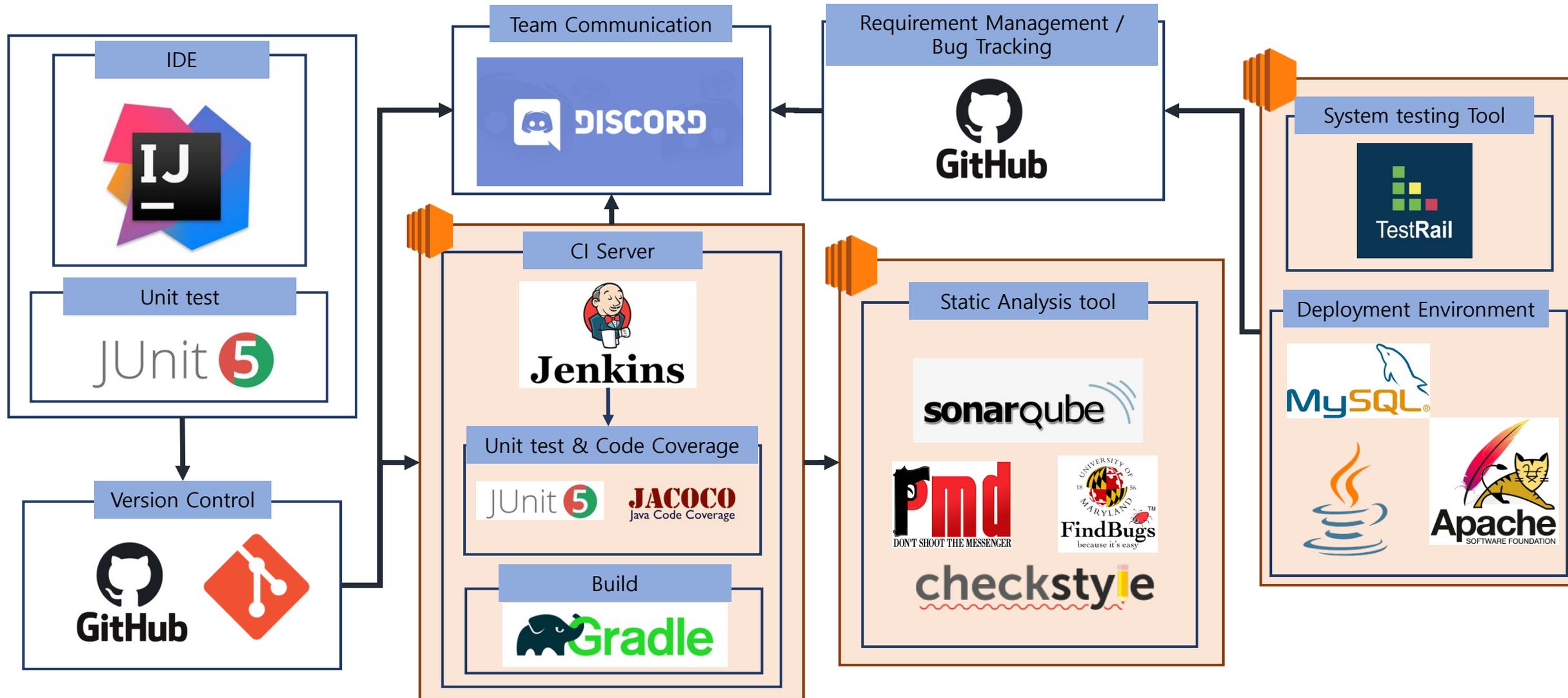
- 1. 한줄평
- 2. CTIP Overall
- 3. V&V 수행 결과
- 4. 개인별 V&V 소감
- 5. V&V를 잘 하기 위해 필요한 조건은?

감정의 변화

1. 한줄평



2. CTIP - OVERALL



3. V&V 수행 결과

Test Runs & Results

Open

- Brute Force v2**
By KUSVV on 2021-06-02 | [Edit](#) | 83%
50 Passed, 0 Blocked, 0 Untested, 0 Retest and 10 Failed.
- Category Partition v2**
By DoHyeon J. on 2021-05-31 | [Edit](#) | 100%
38 Passed, 0 Blocked, 0 Untested, 0 Retest and 2 Failed.

Completed

- 2021년 6월 3일 목요일
BF v1.1
- 2021년 6월 1일 화요일
Brute Force v1.1
- 2021년 5월 31일 월요일
BF v1.0
- Category Partition

SMA_Project master

Quality Gate **Failed**

- Reliability Rating **C** (is worse than B)
- Maintainability Rating **C** (is worse than B)
- 36.3% Duplicated Lines (%) (is greater than 10.0%)
- 75.0% Condition Coverage (is less than 80.0%)
- 79.90% Line Coverage (is less than 80.0%)
- 1 **C** Bugs
- 1 **B** Vulnerabilities
- 10 Security Hotspots
- 1 **B** New Vulnerabilities
- 10 New Security Hotspots
- 8d **C** Debt
- 172 Code Smells
- 8d **C** New Debt
- 168 New Code Smells
- 78.9% Coverage
- 323 Unit Tests
- 78.9% Coverage on 313 New Lines to Cover
- 36.3% Duplications
- 28 Duplicated Blocks
- 31.8% Duplications on 925 New Lines

Dashboard > SMA_Project

Project SMA_Project

SonarQube Quality Gate **Failed**

Code Coverage Trend

line covered (green), line missed (red)

- build, (#42), 2 days 1 hr 전
- stable build, (#42), 2 days 1 hr 전
- successful build, (#42), 2 days 1 hr 전
- failed build, (#31), 7 days 20 hr 전
- unsuccessful build, (#31), 7 days 20 hr 전
- completed build, (#42), 2 days 1 hr 전

3. V&V 수행 결과

main SMA_Project / FOR_START.md

yuz413 Update FOR_START.md

2 contributors

117 lines (65 sloc) 4 KB

[소검] 개발 시작 준비

main SMA_Project / ISSUE_TRACKING.md

IMRaccoon Issue Tracking Docs

1 contributor

80 lines (58 sloc) 3.07 KB

[소검] 이슈 지정 및 처리 방법

main SMA_Project / README.md

IMRaccoon Readme Update

1 contributor

83 lines (54 sloc) 2.89 KB

[소검] CTIP 개발 환경

jeffkimsh commented 6 days ago

- Typical Course of Events
확인 버튼 함수가 없으므로 추가

jeffkimsh added this to the Spec Review (2nd cycle) milestone 6 days ago

jeffkimsh changed the title [SR-3130]-2031_18-01 [SR 3131] 2031_18 - 01 6 days ago

jeffkimsh added this to To do in Spec Review 6 days ago

Kev9171 moved this from To do to In progress in Spec Review 4 days ago

Kev9171 commented 4 days ago

확인 버튼 함수가 무엇을 의미하는지 잘 모르겠습니다.
Use case 레벨에서 함수를 언급하는 이유도 모르겠구요

jeffkimsh changed the title [SR-3131]-2031_18-01 [SR 3131] 2033_18 - 01 4 days ago

jeffkimsh commented 4 days ago

이슈 등록할 때 실수가 있었네요 2033 sequence diagram에 해당하는 내용입니다.

ksy0009 commented 4 days ago

잔액확인하는 건 카드리더기(혹은 api)에서 수행하는 부분이고 자판기쪽에선 결제 결과(성공,실패)만 받는 거라 생각했으며, 카드리더기의 구현은 이 과정에서 원하는 방향이 아니라 생각해 구현하지 않았습니다.
해서 잔액확인에 대한 절차가 없으며 카드 임시 제거, 카드 한도 초과, 잔고부족, 기한 초과 등으로 인한 결제 실패는 확률을 통해 이루어집니다.

IMRaccoon commented 4 days ago

저희가 계속 결제 관련되어서 계속 이슈를 만드는 데에는 결제에 대한 확률 계산 방식의 문제입니다.
잔액을 확인하는 것은 카드리더기가 카드를 인식하고, 외부의 API를 사용하여 해당 카드의 유효성 및 잔액 등을 확인하는 것은 알고 있습니다.
하지만 결제 시에 유효수의 값에 따라 결제가 유동적으로 진행되기 때문에 결제 시에 잔액을 확인하는 과정은 필요하다는 것은 알고 계실 겁니다.
다만 90%로 결제가 성공하는 경우, 결제가 성공했는지 실패했는지 실패했는지에 대한 일관성이 부족합니다.

결제가 가능한 카드일 경우에도 결제가 실패하거나, 결제가 불가능한 카드일 경우에도 결제가 성공할 수 있는 프로그램의 정확성에 대한 문제입니다.
해당 상황의 경우 가장 좋은 방법은 카드의 잔액에 대한 로직을 만들거보다는 카드를 넣는 버튼을 여러가지로 분류하여 구현하고 문서화하는 방식이 존재합니다.
이럴 경우, 잔액이 부족한 경우, 도용된 카드의 경우 등등 다양한 경우에 대해 각각 대처가 가능합니다.
테스트 프로그램이 아니라 실제 문서를 통해 구현하고자 하는 목적이 있는 프로그램에서는 확률이 들어갈 수 없습니다.

김유진 2021.05.25.
어떠한 결과가 나오는지
저희가 알 수 없어서,
어떤 식의 결과가 나와야하는지 의견이 조율이 되었다면

이지영 2021.05.25.
제가 잘못오해한거같아서

김유진 2021.05.25.
그 부분을 알려주시면 그 스펙에 맞게 테스트 하겠습니다
네네

이지영 2021.05.25.
저희가 팝업으로 띄우는 창중에
잘못된 입력입니다 와같은 창이 있어서
오해한거같습니다

김유진 2021.05.25.
아! 넵넵

이지영 2021.05.25.
다시 처음부터 확인해보겠습니다

김유진 2021.05.25.
넵 알겠습니다

전도현 2021.05.25.
해당 alert에대해서는 저희도 파악한 상태에서 테스트 진행하셨다고 생각하시면 됩니다

이지영 2021.05.25.
함수들중에 기능이 같은 함수들을 꼭 다른이름으로 선언해야되나요?

허준호 2021.05.25.
도현님 질문드릴게 있습니다. 1.usecase 11번과 use case 3번이 동일하게 other dvm과 통신하는데 11번의 usecase만 hidden으로 변경하라고 요청하셨는데 어떻게 좋을까요 ??
2.요청사항중 pre requisites에서 전원을 유지하라는것과 삭제하라고 한것이 혼재되어있는데 대부분은 삭제할까요 유지할까요??

전도현 2021.05.25.
다른 파일이나 클래스에서도 동일한 기능을 가지고 있다면 무관합니다만, 대체적으로 2, 3번 넘게 쓰이는 경우와 같은 이름으로 선언하는 것이 좋습니다

4. 개인별 V&V 소감 - 김승환

Quality Experts 수행 소감

- 한 학기 동안 스펙 리뷰, 시스템 테스트, 정적 분석을 하였지만 계속해서 만족하지 못한 퀄리티가 나왔습니다. V&V 팀원들이 요구하는 것들을 제대로 수행하지 않아서 시간도 지체되었고 검증을 제대로 하지 못하였습니다. 이를 통해 결과물을 내는 것도 중요하지만 **소통도 중요하다**는 것을 알게 되었습니다. 그리고 검증한 것들을 **개발팀에게 납득이 가게끔 설득하는 것도 Quality Expert의 역할** 중 하나이고 힘든 작업이라는 것을 알게 되었습니다.

3학년 개발팀에게 해 주고 싶은 얘기

- 처음 planning 단계부터 좀 더 체계적으로 문서를 작성하였다면 모순된 설명이나 잘못된 설명으로 인해 개발에 차질이 가지 않았을 것 같습니다. 문서 리뷰를 하고 수정을 해도 계속 연쇄적으로 꼬여 모순된 부분이 없어지지 않았고 퀄리티는 좋아지지 않았습니다. 그런 스펙으로 개발을 하다보니 다양한 테스트를 해도 잘못된 결과가 계속 나왔던 것 같습니다. 첫 단추부터 planning을 잘했다라면 좀 더 좋았을 것 같은데 아쉽습니다.

4. 개인별 V&V 소감 - 김유진

Quality Experts 수행 소감

- 처음 팀원들과 CTIP 환경을 구축하기 위해 툴 종류를 알아보고 비교해보며 세팅을 진행했을 때 이제까지 알지 못한 새로운 부분들을 알게 되어 좋았습니다. 이후 스펙 리뷰를 진행하며 각 문서마다 내용이 다르다는 걸 알았을 때, 그리고 테스트를 진행하는 데에 있어 **정확한 범위, 구체적인 설명 등이 주어지지 않아 어떤 기준에 맞춰 테스트를 진행해야 하는 지** 등의 3학년 개발팀과 협업하며 몇 가지 어려움이 따랐습니다. 긍정적으로 생각하면 검증 입장에 있으면서 개발할 때 어떤 부분이 필요한 지도 알아가고 **소통의 중요성을 깨달을 수 있었던** 시간이었던 것 같습니다.
- 이처럼 이제까지 개발 입장이었다면, 이 수업을 들으며 **검증 관점에 다가가 보는 뜻 깊은** 시간이었습니다.

3학년 개발팀에게 해 주고 싶은 얘기

- 처음 문서를 보았을 때 팀원들이 똑같은 문서에 대해 각자 다르게 해석했습니다. 처음 보는 문서라 그런 것이겠지 싶었지만, 문서마다 다른 내용들에 점차 혼란이 왔습니다. 그렇게 문서에 한 번 스쳐가고 테스트를 진행했을 땐 스펙 대로 테스트를 검증한다는 느낌보단 테스트를 진행하며 개발과 문서에 있어 다른 부분 혹은 모호한 부분을 찾아서 알려주는 느낌이었습니다.
- 그만큼 객체지향팀도 많은 리뷰 양으로 힘들었을 것 같습니다.

4. 개인별 V&V 소감 - 이현우

Quality Experts 수행 소감

- 처음에는 CTIP를 세팅하는 것에서 제일 애를 먹을 것이라고 생각했지만 **3학년팀과의 소통에서 가장 힘들었습니다.** 그래도 팀원들과 함께 준비한 **개발환경에 코드를 올려 분석**하고 문서의 단어 하나하나 보면서 눈이 빠질 뻔하기도 했지만 3학년 객체지향개발방법론에 이어 소프트웨어 검증 수업을 들으니 확실히 **프로그램 하나 개발하는데 많은 노력이 필요하다는 것을 느낄 수 있는 경험**이었습니다.

3학년 개발팀에게 해 주고 싶은 얘기

- 분명 저도 작년에 들었던 수업인데 분명 팀원이 5명인데 이 정도 양의 문서 오류와 한 팀이 작업한 느낌이 아닌 소통이 안되는 2팀이 작업을 한 것을 저희가 다시 합치는 작업을 하는 느낌이라 너무 힘들었습니다. 문서를 기반으로 하나의 프로그램이 나와야 하는데 한 가지 주제를 가지고 문서 따로 프로그램 따로 만든 느낌이라 덕분에 저희 팀의 단합이 올라갔습니다.

4. 개인별 V&V 소감 - 전도현

Quality Experts 수행 소감

- CTIP 환경을 구축하면서 말로만 들어봤던, 또는 **처음 보는 틀들을 조사하고 설치, 연동하며 유기적으로 작동시키는** 경험은 새로웠습니다. 또 다른 팀과는 별개로 웹으로 진행되는 데에 더불어 중앙 서버의 역할을 하는 패킷 서버를 위한 DB, Tomcat 등 추가적인 환경들이 필요했고, 이러한 환경들을 구축하면서 실제 다양한 경험을 해볼 수 있었습니다.
- 소프트웨어 검증 수업을 들으면서 가장 힘들었던 것은 **3학년 팀과의 협업**이었습니다. CTIP 환경에 관하여 필요한 부분들은 모두 문서화 해주고, Spring4, Mysql5, Tomcat8 과 같은 예상치 못한 환경은 최대한 맞추어 주려 노력했습니다. 하지만 **웹으로 개발하는 데에 가지는 이점들을 전혀 활용하지 못했을** 뿐만 아니라, 따로 만들어 놓은 **중앙서버를 활용하여 통신하지만 분산 네트워크 환경처럼 구현한 것은 오히려 분석에 혼란을** 주었습니다.
- 또한 JavaScript에 실제 구현 내용이 들어가 있어, 어떻게 할 수 조차 없었습니다. 그나마 마무리 되었지만 만족스럽지 않았고, 힘들었을 뿐만 아니라 고통스러운 과정들이었습니다.

3학년 개발팀에게 해 주고 싶은 얘기

- IntelliJ를 설치하는 방법에 대한 링크를 보내주었을 때, 영어가 너무 어렵다는 말은 아직도 제게 충격으로 다가왔습니다. 지금까지 어떤 식으로 개발해 왔는지 잘 모르겠지만, 기본적으로 질문하기 전에 검색하는 습관을 들이는 것이 좋을 것 같습니다.
- 저희는 모두가 매 사이클마다 문서 화면을 공유하여 토의하면서 리뷰를 진행했고, 할 때마다 문서 내용에 대해 서로 다르게 받아들인 것들이 많았습니다. 저희가 잘못된 것일 수도 있지만, 최소한 읽는 사람이 서로 다른 의도로 생각하지 않는 문서가 되었으면 좋겠습니다.

5. V&V를 잘 하기 위해 필요한 조건?

객체지향팀 매칭 운
인내 심소통 설득력
꾸준함 이해력 일정 관리



Thank You.

TEAM #7

2021 Software V&V