

소프트웨어 V&V

CTIP - Final Presentation

201511243 김동언

201511246 김상재

201511262 박우진

201711356 천세진

Index

1. 한줄평

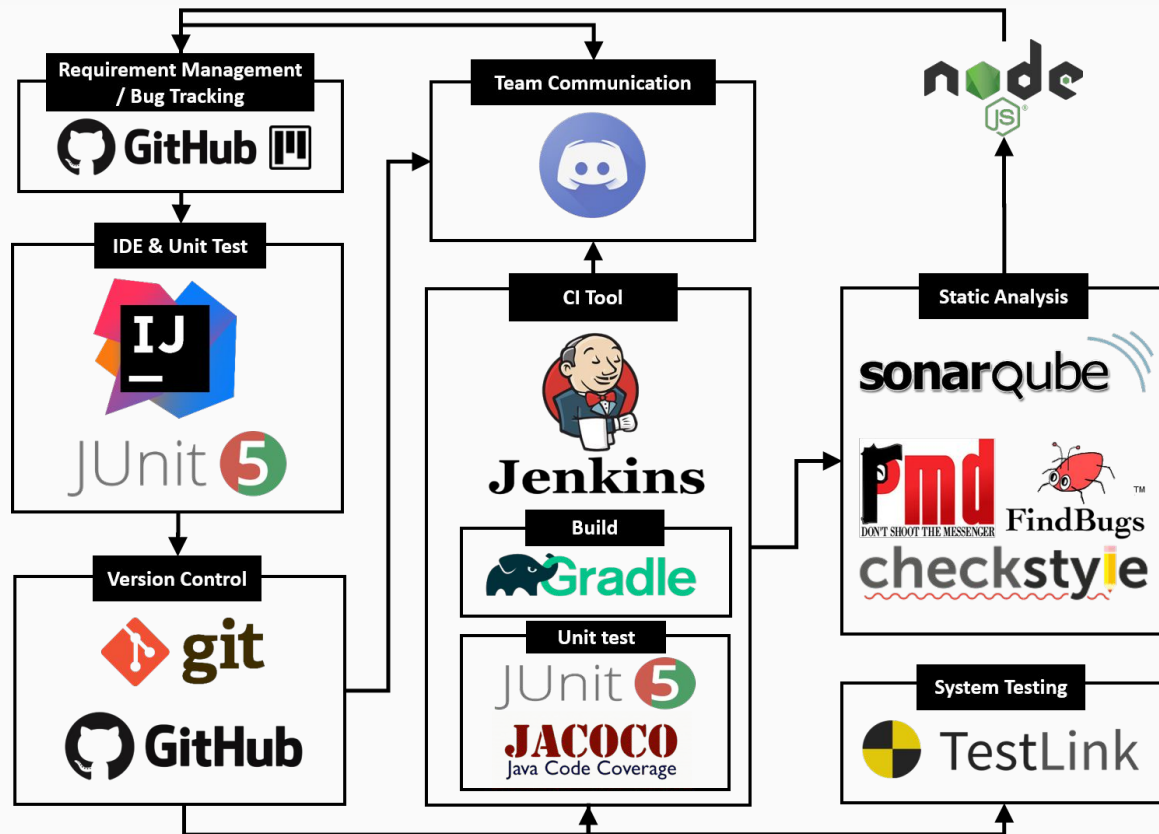
2. 전체 CTIP, SQA 활동
 - a. Overall Structure
 - b. A7
 - c. B6
 - d. B7
 - e. Improvement

3. 소감



결과가 좋던 좋지 않던 우선 소통이 되어야...

전체 CTIP, SQA 활동 - Overall Structure



전체 CTIP, SQA 활동 - A7

2020년 6월 25일

김동현 <A7> 2020.06.20.
@정연수 (A7) [팀장] @이도진 (A7) @박성현 (A7) 한 분야라도 답변 해주세요

2020년 6월 25일

김성재 <CTIP 관리자> 2020.06.09.
@SMA_A7 공지합니다. 코드 작성하신게 있으면 가능한 빠르게 CTIP 환경(KUSVV)에 push 해주시길 바랍니다. 방법을 모를시에는 @sVV 로 연락주시기 바랍니다.

김동현 <A7> 2020.06.09.
@SMA_A7 홈페이지에 PPT를 포함한 어떠한 파일도 올라와있지 않음으로 검증이 불가능합니다. 오늘 밤 열두시까지 코드와 실행파일을 올리시면 그 코드를 참고하여 검증하도록 하겠습니다.

2020년 6월 19일

김동현 <A7> 2020.06.10.
@SMA_A7 자정이 지났으므로 홈페이지에 올라온 상태 그대로 진행하도록 하겠습니다.

2020년 6월 18일

정연수 (A7) [팀장] 2020.06.18.
제가 깃허브 링크 접속에 404에러가 뜨는데 혹시 무슨 작업을 해야 하는 지 알려주실 수 있나요?

김동현 <A7> 2020.06.18.
Issue Tracking projects에 있는 테스트 결과는 기본적으로 V&V 2nd cycle 보고서에 있는데요, BruteForce Testing 결과는 Issue 내용에 쓰여있구요, Category Testing 결과는 180개에 대해 코멘트로 우선적으로 발생 원인을 작성해 드렸습니다. 작성해 된 모드의 개수 수정이 불가능 한 것과 Worldtime 모드 구현이 되지 않아서 테스트가 불가능한 경우이므로 기능 구현이 우선 되어야 합니다. Pairwise 결과는 Category Testing에 포함되는 Testcase이므로 Category-partitioning Testcase를 해결하시면 자동으로 해결되는 부분입니다

김동현 <A7> 2020.06.18.
BruteForce TC에 대해 Refuse처리 하시는 타당한 이유를 코멘트로 작성하시고 OOPT 3rd cycle 보고서에 작성하시면 됩니다. @이도진 @SMA_A7
BruteForce TC에 대해서도 어떤 Testcase인지 내용보강 하였습니다

#331 opened by Biccon
2nd Cycle Todo

Pairwise/Test30
#330 opened by Biccon
2nd Cycle Todo

Pairwise/Test29
#329 opened by Biccon
2nd Cycle Todo

BruteForce/Test5-3
#339 opened by Biccon
2nd Cycle Accept Todo

BruteForce/Test5-1
#337 opened by Biccon
2nd Cycle Accept Todo

BruteForce/Test3-2
#335 opened by Biccon
2nd Cycle Accept Todo

BruteForce/Test2-1
#334 opened by Biccon
2nd Cycle Accept Todo

BruteForce/Test1-3
#333 opened by Biccon
2nd Cycle Accept Todo

BruteForce/Test1-2
#332 opened by Biccon
2nd Cycle Accept Todo

seia1030 added Accept Refuse and removed Accept Todo labels 4 days ago

Biccon added the 2nd Cycle label 3 days ago

dojin2 commented 21 hours ago

DynamicDisplay는 시간문제상 구현을 하지 못했습니다.

☆ SMA_A7 Failed

Last analysis: June 17, 2020, 12:25 AM

142 Bugs | 101 Vulnerabilities | 0 Hotspots Reviewed | 576 Code Smells | 0.0% Coverage

- 각각 Iteration마다 Notification 및 소통 시도 하였으나 소통이 되지 않음
- 기능 구현 및 Issue 해결이 되지 않았음
- Unit testcase가 없어 Coverage가 0%로 마무리 되었음

전체 CTIP, SQA 활동 - B6

KUSVV / SMA_B6 | Private
generated from KUSVV/SMA_INIT

Static Analysis Results
Updated 16 hours ago

Issue Tracking
Updated 3 days ago

Summary Card:
☆ SMA_B6 Passed
Last analysis: June 21, 2020, 11:43 PM

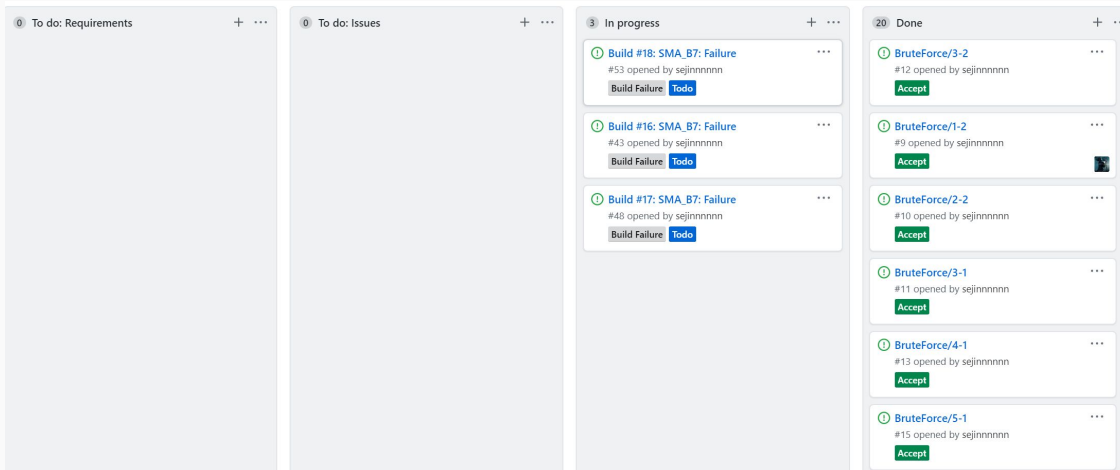
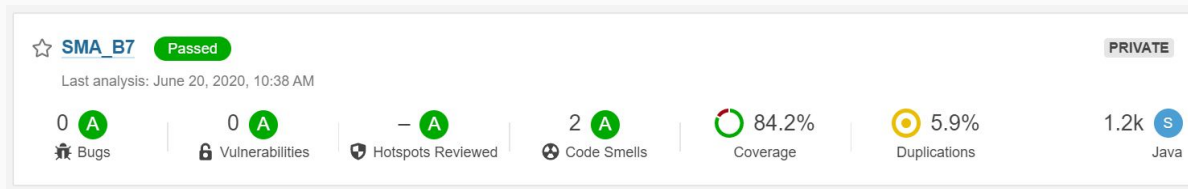
17 C Bugs | 7 B Vulnerabilities | - A Hotspots Reviewed | 50 A Code Smells | 78.1% Coverage | 0.0% Duplications | 1.8k S Java

Notification: ✓ #28 | 2020. 6. 21 오후 11:41

- Unit Test 7개 실패 개선되어 0개로 100% Success
- Coverage 수치도 개선
- 최종 Build Success

Overview	
On new code	
Coverage	81.9%
Lines to Cover	118
Uncovered Lines	15
Line Coverage	87.3%
Conditions to Cover	48
Uncovered Conditions	15
Condition Coverage	68.8%

전체 CTIP, SQA 활동 - B7



- 코드 커버리지 값이 매우 낮았지만, 그 부분을 거의 완벽하게 해결하였음.
- Code Smell 영역에서도 150개의 issue를 단 2개로 줄였음.

- System Testing Result

- A7
 - Category: 180개 -> 180개
 - Pairwise: 31개 -> 31개
 - BruteForce: 9개 -> 2개
- B6
 - Category: 0개 -> 0개
 - Pairwise: 0개 -> 0개
 - BruteForce: 1개 -> 1개
- B7
 - Category: 8개 -> 0개
 - Pairwise: 0개 -> 0개
 - BruteForce: 8개 -> 0개

- Static Analysis Result

- A7
 - Bug: 121개 -> 142개
 - Vulnerability: 39개 -> 101개
 - Coverage: 0% -> 0%
- B6
 - Bug: 20개 -> 17개
 - Vulnerability: 8개 -> 7개
 - Coverage: 53.0% -> 78.1%
- B7
 - Bug: 5개 -> 0개
 - Vulnerability: 0개 -> 0개
 - Coverage: 26.2% -> 84.2%

김동언 (A7): 작년 객체지향프로그래밍 수업을 들을 때에는 개발자의 입장에서 만드는 방법을 배웠다면, 이번 수업에서는 검증자 관점에서 개발의 Quality를 높이기 위해 신경을 써야하는지를 배울 수 있었다.

보통 개인 프로젝트를 진행할 때 CI/CD 프로세스를 적용하지 않고 테스트도 따로 진행하지 않는데, 이번 수업에서 CI/CD 환경 구축을 해본 것이 가장 큰 경험이었고 **테스팅의 중요성에 대해 알게 된 사실**이 가장 앞으로 개발 생활에 있어서 긍정적인 변화를 만들 수 있을 것 같다.

V&V 과정에서 가장 힘들었던 점은 **객체지향 프로그래밍 팀과 소통을 할 수가 없어** 요구사항에 대한 고지가 불가능하였고 이전 단계의 스펙과 다른점, 명확하지 않은점이 너무 많아서 하나의 프로그램으로 귀결되지 않아 스펙 검증 단계에서 너무 많은 시간이 소요되어 **상대적으로 다른 프로세스에 사용할 시간이 부족**하였다. 객체팀에서 작업하지 않아서 검증할게 별로 없을거라고 생각한 것은 큰 오산이었고, 문서화가 제대로 되지 않은 부분을 검증하는 것이 더 어려웠던 것 같다.

개발하는 입장보다는 시간이 적게 소모된다고 생각되는 것은 사실이나 Due date를 지나 업데이트 되는 스펙과 코드를 이해하고 반영하여 검증을 수행하려다 보니 이전에 진행중이던 스펙 리뷰에서 변화가 조금씩 생겼던 것 같고 이러한 사소한 변화가 **검증팀에게 부담을 더 가중하는 느낌**이 있었고, 객체지향방법팀은 피드백을 받고 잘/잘못에 대해 알게되는 프로세스가 있는데, 검증팀이 **SQA로서 검증하는 역할의 잘/잘못을 지적받지 못**했던 실습 상황에서 잘못된 점을 수용하고 고칠 수 있는 기회가 적지 않았나 싶었다. 검증팀이 검증을 잘 수행하고 있는지 피드백이 생기면 더 많은것을 배울 수 있을것 같다.

박우진 (B6): 이번 소프트웨어 V&V 수업을 들으면서 개발과정을 진행하며 검증하는 방법과 그 과정에 대해서 배웠다. 하나의 프로젝트를 완성하기 위한 과정을 배우는 **소프트웨어 공학계열 수업을 들으면서 지식과 경험**이 생기고 있다. 뿐만아니라 평소에 관심을 가지고 있지 않던 ISO26262, DO-178C 에 경험해본것도 좋은 경험이었다.

평소 혼자서 개발을 하거나 소규모 프로젝트를 진행할 때, CI/CD 에 대해서 용어만 들어만 봤었지 제대로 해본적도 없었다. 이번 프로젝트 실습을 위해 개발 환경을 구축하면서 도구들을 붙이고 어떻게 조금더 관리하기 쉽게 고민하면서 **개발 환경을 만드는 경험을 해봐서 가장 의미**있었다. 뿐만아니라 이론에서 배운 내용을 가지고 직접 Test case 도 만들어보고 그에 맞는 Test Execution 도 하고 그 결과도 만들어 개발팀에서 전달하는 등의 프로세스를 경험해서 의미있었다.

이번 프로젝트를 진행하면서 힘들었던 점은 우리가 생각했던 Spec 의 정도가 달랐고, 그것에 대해 전달받지 못해 **Spec Review 를 하는데 시간을 많이 썼고** 다른 프로세스에 많은 시간을 못쓴거 같아 아쉬웠다. 어느정도의 소통은 되었지만 정말 기본적인 권한 부여, 문서 주고 받는 정도의 소통밖에 되지 못해 이점도 아쉬웠다. 뿐만아니라 **V&V 팀을 검증하거나 빠른 피드백을 전달을 못받은 점이 아쉬웠다.** 우리의 팀의 경우도 Test Case 를 뒤늦게 피드백을 받아서 다시 Generate 했었는데, V&V팀에게도 빠른 피드백이 전달되어 개선되었으면 좋을 것 같다.

천세진 (B7): 객체지향개발방법론 수업을 듣지 않아, **모든 것이 낯설었습니다.** 수업에서 들었던 이론보다 CTIP 이용 시 되게 거리감이 있는 느낌이었습니다. 그러나, 그것 뿐만이 아니었습니다. **객체 팀과의 소통 및 Report 작성**도 쉬운 일이 아니라는 것을 깨달았습니다. 처음이라 무엇을 어떻게 기준을 잡고 가야 하는지도 몰랐습니다.

그러나 이 SQA라는 역할이 왜 필요한지 느끼게 되었습니다. 개발자의 입장에서만 역할을 수행해왔던 저는, 상당히 놓치는 부분이 많구나 라는 것을 깨달았습니다. 개발 시 우리가 흔히 겪는 문제들에 대해서 **SQA를 통해서 많은 부분이 개선되어지고, 또 좋은 코드를 만들어낼 수 있게 발전시킨다고 생각했습니다.**

그러나, 그 길은 매우 험했습니다. 처음 보고서를 읽을 때엔 한 눈에 그 팀의 프로그램이 어떤 기능을 하고, 테스트는 어떤 방식으로 진행해야 하는지, 그리고 그것이 객체팀이 원래 의도했던 바였는지 등 너무나 많은 것이 프로젝트 중의 저를 혼란스럽게 하였습니다. 또한 객체팀과의 소통이 원활하게 이루어지지 않다 보니 이것들이 다 너무나도 힘들었습니다. 그러나 이것은 값진 경험으로 남겨질 것 같습니다.

쉽게 할 수 없는 경험을 대학 수업에서 하게 되어 매우 인상적이었고, 또한 오랫동안 기억에 남을 것 같습니다.

김상재 (Overall): 소프트웨어 검증 환경의 구축이 상당히 어렵다는걸 느낄 수 있었습니다. 아무것도 모르는 상태에서 환경을 구축하기란 쉽지 않았고 어떤 안정적인 서버 및 환경을 구축하느냐에 대해서 많은 어려움을 겪었지만 결과적으로는 SQA 입장에서나 개발자 입장에서나 원활하게 사용할 수 있어서 뿌듯함을 느꼈습니다.

막상 SQA가 되보니 **개발자들의 작성한 문서의 의도를 파악하고** 실제 개발 결과물을 확인하여 **테스팅 하는 과정을 수립하는 것도 무척이나 어려웠었고** 테스팅한 내용이 옳은 내용이고 옳은 결과인지 확인하는 것이 쉽지 않았었지만 **값진 경험**을 했다고 생각합니다.

감사합니다