

# SW Verification

201211387 하헌규

201211337 김재현

201211354 성현구

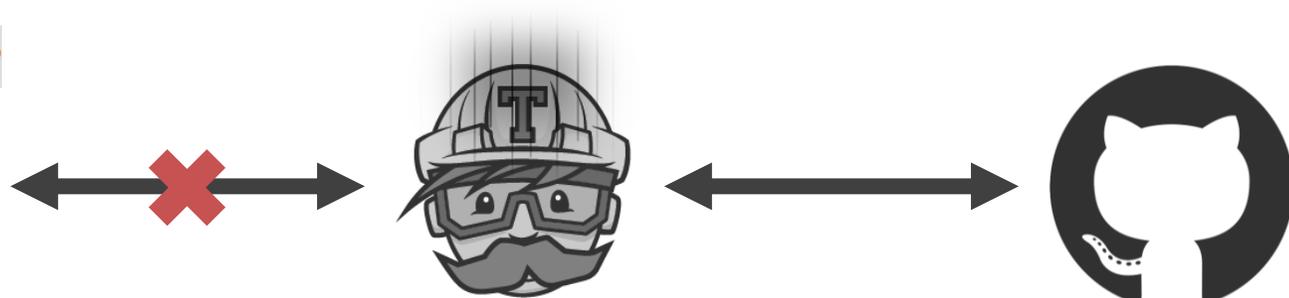
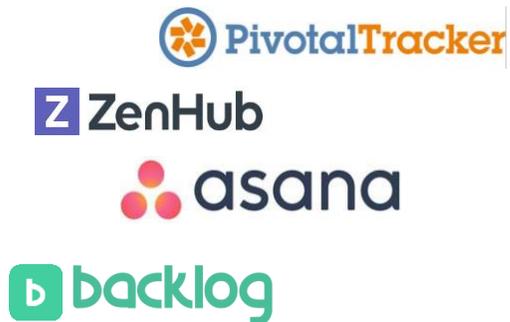
201211369 이병찬

- CTIP
  - IntelliJ / Gradle / Github / Travis CI / Serenity / SonarCloud / TSL / TestRail / CodeScroll
  - 장단점 및 개선안
- V&V
  - System Test / Static Analysis
  - 개선안
- Summary

- 개발하는 입장이 아니라서 직접적인 사용 無
- 개발팀을 지켜보니 이클립스와 큰 차이 無



- Travis에서 서버를 제공하는 것이 장점이자 단점
- yml 파일에 대한 참고문서가 제공되지만 불친절하다
- Github와 연동하기 쉽다 → Github 말고 아무것도 안 붙는다

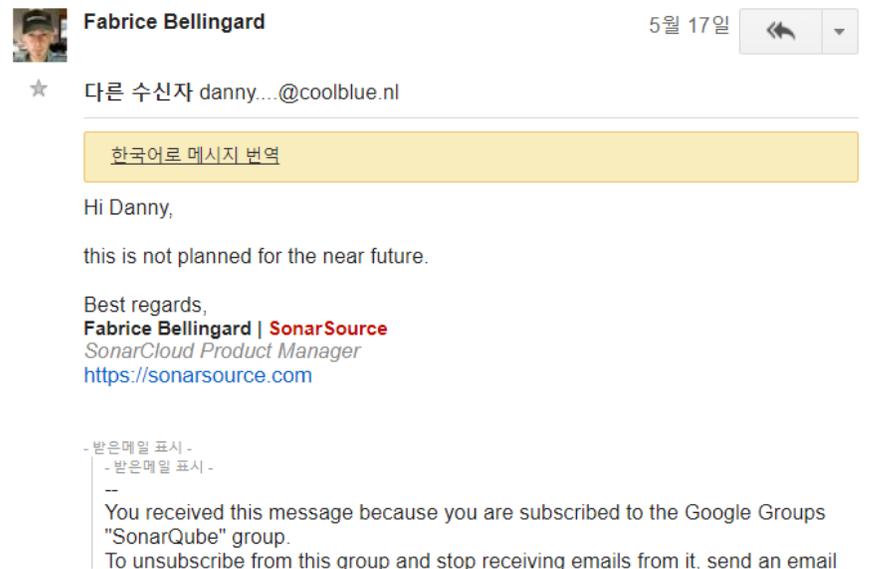


- Github가 없었다면 Travis CI를 포기하고 Jenkins를 사용하게 되었을 듯
- 대부분의 도구와 연동되어 Github 중심의 CTIP 구축

- Testlink보다 좋은 UI
- 결과와 Comment를 입력하면 원클릭으로 Github Issue Tracker에 등록이 되어 편하다

- Travis CI 때문에 사용
- 도구 사용이 매우 불편 (UNIT TEST에 코드를 추가하여 관리)

- Travis CI 때문에 사용
- SonarQube의 룰셋 이외에는 Import해서 사용 불가



◀ 개발자 측에서도 아직 고려사항이 아니라고 답변

- 플러그인 추가가 쉽다
- XML파일에 익숙하지 않은 개발자에게 더 유용하다(Groovy)

# CTIP - TSL Generator

- Hierarchy가 최대 2개까지만 가능한 점이 아쉽다
- Test Case의 개수만 알려주는 기능도 있으면 좋을 듯



- 3레벨 에러 검출까지 가능하나 이번 프로젝트에서는 확인하지 못했다
- 체감상 기존 도구와 큰 차이점이 없다고 생각

# CTIP - 개발팀 입장에서의 장단점

## SMA T2 :

- Travis CI에서 바로 확인 가능해 코드를 빨리 수정할 수 있다
- 정적분석 결과를 보면서 안 좋은 코드 부분도 알 수 있다
- Github 사용 시 브랜치 사용법을 몰라 아쉬웠다

# CTIP - 검증팀 입장에서의 장단점

## SV T2 :

- 개발팀의 진행상황을 대부분 알 수 있다
- 글로만 소통하기 때문에 해석 차이가 발생할 수 있다
- Travis는 최대한 안 쓰는 쪽이 좋다고 생각한다

<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> dohyeon (2)	Re: [Nineclown/The-Convenient-ATM] 메시지에 대한	6월 5일
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Jinho Jeong (5)	Re: [Nineclown/The-Convenient-ATM] Definition: 권	6월 5일
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> Jinho Jeong (4)	Re: [Nineclown/The-Convenient-ATM] 복권 : 주차 0	6월 5일
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> jaehyun739 (4)	[Nineclown/The-Convenient-ATM] Failed test: 해당	6월 5일

# CTIP - 개선안

- 개발팀이 사용할 IDE나 요구사항 관리 도구는 개발팀과 상의한 뒤 정하는 것을 추천



Xcode



# V&V - CPT 장단점

- CPT를 작성하면서 입력값을 정하기 때문에 테스트하는 데 걸리는 시간이 줄었다
- SMA팀의 문서가 꼼꼼하지 못하면 Representative Value를 정하는 것이 어렵다
- 기본 기능이 제대로 동작하지 않으면 연쇄적으로 모든 Test가 Fail된다

# V&V - SA 장단점

- 중간중간 에러가 아닌 것도 있어 별도의 판단이 필요하다
- Testing과 비교하여 룰을 정해서 돌려주는 작은 비용의 행동으로 문제점을 찾을 수 있다
- 잠재적인 위험을 알 수 있다

# V&V - 개선안

- 대부분의 조에서 Pairwise Test도 진행했는데 CPT처럼 실습 수업이 있으면 더욱 수월할 듯
- Static Analysis 단계에서 주제에 맞게 미리 룰셋을 정해 주면 좋을 듯
- 테스트를 수월하게 하기 위해 SMA팀에서 매뉴얼을 작성

# Summary

- 다른 SMA팀이 대부분 코드를 완성하지 못해 SV팀의 테스트에 어려움이 있었는데 SMA T2의 경우 코드를 제때 완성해 테스트하기 편했다
- 피드백도 빠르고 수정 요청사항을 대부분 반영했다
- CTIP 환경을 잘 이용해주었다

감사합니다 :)