

CTIP 소감

- **장점:** github을 통해 팀원과의 협업 개발 환경을 구축해준 점이 좋았습니다. 또한 github에 push를 할 때마다 github Action을 통해 자동으로 분석되도록 구축이 되어 있는 부분이 좋았습니다. 분석 역시 굉장히 정확하고 세부 부분도 잘 묘사되어 있어서 편했습니다. 또한 통계적 자료만 보여주는 것이 아니라 정확히 어디 어느 부분에서 문제가 생겼는지 잘 알려주어 작업을 하는데 빠른 interaction이 가능해 좋았습니다. jira에서 Test Details 부분이 무엇을 넣고 실패했는지 보여서 좋았습니다.
- **단점:** github에 push를 할 때마다 github action이 돌아가는데 팀원들간 서로 역할이 나눠져 있어 누군가가 일부 분만 올렸을 때 fail이 나는 부분이 아쉽습니다. 보고서에 굉장히 잘 정리되어 있어서 보는 데는 불편함이 없었으나 이슈의 경우, Test PASS된 이슈들도 올라와 있어 아쉬웠던 것 같습니다. Jira를 사용하는데 있어서 활용이 살짝 떨어진 감이 있었던 것 같습니다. 일단 진행중인 작업, 보류, 완료된 작업 분간이 잘 가지 않아 어떤 이슈들이 있고 아직 해결되지 않은 이슈들이 무엇이 있는지 분류하는 작업이 조금 힘들었습니다

UP 개발 소감

1) 류창현

- **장점:** 미리 앞에서 특정 SW의 기능 부분, 클래스 및 함수 부분들을 정의하고 넘어가 구현 단계 당시에 생각할 부분들이 적어져서 편했던 것 같습니다. 또한 순차적으로 문서를 만들다 보니 처음부터 구현을 하는 것에 비해 단계별로 올라가다 보니 생각이 더 잘 정리되는 듯합니다.
- **단점:** UP 기반의 단점보다는 저의 능력 부족한 부분입니다. 앞에서 문서 작업을 할 때 OOA, OOD등에서 명확하게 짚고 넘어가지 못 했거나 생각을 잘못된 부분들이 있는 경우, 뒤의 OOI 단계에서 도리어 앞의 문서보다 더 필요한 부분들이 많아 도리어 개발에 어려움을 겪는 부분이 있었습니다. 해당 문서 작성 능력이 충분하다면 괄찬을 듯 싶습니다.
- **개선 사항:** 수업에 대한 이야기인데 각 문서마다 설명이나 피드백 등이 더 자세했으면 좋았을 것 같습니다.

2) 강현수

- **장점:** 계획이 비교적 자세해서 구현이 쉬웠습니다.
- **단점:** 계획의 과정이 자세한데 기간은 짧아서 제대로 계획하기 힘들었으며 계획단계에서 매우 바빴습니다.
- **개선 사항:** 계획부분을 조금 덜 자세하게 다룬다면 구현단계와 계획단계의 시간분배가 더 좋아질 것 같습니다. 물론 현업에서 사용할 때는 현재 형태가 좋겠지만 수업을 듣는 사람들은 학생이라서 할 게 많다는 것을 고려한다면 말입니다

3) 송주한

- **장점:** UP의 개념인 프로그램을 여러 번 반복하여 만들어서 장점을 모은다는 것이 마음에 들었습니다. 개발하면서 CTIP를 받는 것은 처음인데 생각보다 자세하고 가독성이 좋아서 개발을 하는데 있어 도움을 받은 것이 좋았습니다.
- **단점:** 저의 부족한점이지만 제한 시간안에 개발을 못하면 자꾸 뒤로 밀려서 전체적으로 프로그램을 만족하는 퀄리티로 못 만들었다는 점에서 UP기반은 확실히 실력이 중요하다고 느낌을 받았습니다. 제한시간 안에 정확하게 만들 수 있는 자신감이 있으면 UP기반이 유용하다고 생각합니다.
- **개선 사항:** MSG부분은 다들 학교수업으로 배우지 않아서 다들 독학으로 공부하고 만드는 것이 시간이 많이 들

은 것에 대해 기본적인 자료를 제공하면 더 좋았을 것 같습니다.

4) 남승협

- **장점:** 지금까지 해왔던 간단한 응용프로그래밍에서는 배우지 못했던 대형 프로젝트의 개발 단계와 여러 스킬들을 배우고 또 응용을 해서 실습을 해볼 수 있다는 것이 큰 장점이라 생각합니다. 간단한 레벨에서의 코딩이라 전반적인 설계작업에 대한 프로세스를 거치지 않고 머리에서 생각나는 대로 코드에 옮겼지만 문서로 정리하여 client와 developer간의 interaction 그리고 개발팀과 테스트 검증팀의 communication 등 실무에서 필요한 스킬과 더불어 팀단위의 작업을 효율적으로 분배하고 성취해가는 과정을 배울 수 있었습니다.
- **단점:** 프로젝트를 진행하며 OOA, OOD 문서를 정확하게 해석하는 단계에서는 본인의 능력이 약간 부족하다고 느꼈습니다. 이러한 문서 해석에 대한 부족한 점을 채우기 위해서 다음 수업부터는 문서제작 이후에 팀끼리 바뀌어서 구현을 하는 방식을 도입해 문서해석능력을 키우는 것이 좋을 거 같습니다.
- **개선 사항:** 구현 단계에 있어서 메시지 프로토콜과 같은 한번도 다뤄보지 않았던 기술을 구현하는데 어려움이 있었습니다. 전공책을 찾아가며 해결은 하였지만 overhead가 큰 spinning을 사용하는 등 성능까지 고려하지는 못한 모델이라고 생각합니다. 이러한 부분에 있어서 OOAD를 온전히 제 것으로 만들기 위해 어느정도 NFR과 Quality를 고려하는 방법에 대해서 배워볼 수 있는 부분이 있었으면 합니다.
-