

## WEDNESDAY Aug 4 Sprint 2 Retrospective

In general the team's performance in sprint 2 was much improved compared to sprint 1. There are still various issues regarding following ADRs and time management but we are starting to understand the process much better (although more time would go a long way). Improvements to the current implementations in terms of testing and completeness, as well as GUI, would be a big help to delivering a complete and polished product.

Additionally, there was still a big integration at the end of the sprint (instead of Continuous Integration throughout) for the majority of the group members. Travis, Coveralls, Automated deployment etc. all worked well and, in conjunction with, a consistent and unified code base to work off of meant **when** code was integrated, it went smoothly.

### JAMES:

#### What went well?

- The teams, as a whole, has learnt and adapted to the AGILE process
  - Pull requests and descriptions are up to standard
  - Merge conflicts are minor and easy to resolve
  - The current base is, mostly, easy to work off of and make changes to
- Testing has been implemented for previous and current implementations which has increased our code coverage overall. While the percentage has only increased slightly there has been a large increase in the number of files and Lines Of Code (LOC) which means this increase is more impact than at first glance
- Various bug fixes and improvements during the sprint were delivered, aligning with CID and AGILE principles
- The app now has a 'flow' to it allowing the user to access the various functions as the implementations are (mostly) all linked correctly

#### What went wrong?

- Team members are still violating ADRs, specifically regarding the MVVM architecture
- Time management has been an issue as the stories delivered are incomplete or lacking polish (reviews need to be more in-depth, which is difficult with time constraints)
- CID is not being followed as we had one big release/integration at the end of the sprint, instead of code being continuously merged throughout
- Team members are not descriptive enough in their Pull Requests stating what is and isn't present in their code in terms of the story being delivered
- Managing of time, possibly due to not having thin enough slices for our stories
- The delivering on time as the sprint was completed 3 Days past the due date

### **What could be improved?**

- Testing could be improved in terms of implementation and separation
  - The live database is being used to test
  - Functions are not mocked so, only, a very broad range of functions are test (need to get down to the specifics)
  - Very little client side testing
- Communication and attendance is very hit or miss, although some team members are experiencing technical difficulties e.g. load shedding, internet connection, poorly speced equipment
- Reading and familiarity with ADRs and various documentation for the libraries and systems used
  - Closing keywords for linking issues
  - Bootstrap usage
  - PR Review Style guide ADR
  - Jest

### **SINO**

#### **What went well?**

- I was able to deliver the complex part of my story
- Incrementing on tests was nice to see, coverage went up
- The team's performance was much improved in sprint 1, especially considering time constraints

#### **What went wrong?**

- Team members could not deliver on all of their assigned tasks/stories - the slices were possibly too large

#### **What could be improved?**

- Being strict with TDD approach (as outlined in the ADR)

James: This will require people's current implementations to be up to date and working i.e. bug fixes

- The number of pull requests per sprint should be increased, spreading out the merges

## **STEPHEN**

### **What went well?**

- Time to integrate and make pull requests was much shorter - team is improving, learning and adapting
- Testing increased which bodes well for the next sprint
- Enhancements were made to the existing implementations which improved the final product

### **What went wrong?**

- Thinks, in general, it was a good sprint

### **What could be improved?**

- The user interface could be improved with a consistent theme and styling throughout the app
- More tests could be done to increase our coverage of the code

## **SINAZO**

### **What went well?**

- I was able to deliver part of my sprint
- Fixing bugs from the previous sprint
- Increasing the numbers of tests and improving code coverage

### **What went wrong?**

- Integration was not very smooth as CID was not followed correctly (a large part of the team only merged code before the sprint end) - time constraints
- Bugs/Missing links in the code - time constraints

### **What could be improved?**

- Improve the testing of different layers and separated functions - use mocking correctly
- Finish stories earlier (maybe try create thinner slices) so that the team can integrate continuously rather than one big integration

# **LUNGELO**

## **What went well?**

- We worked well together and assisted each other

## **What went wrong?**

- The Sprint was too short for the workload assigned

## **What could be improved?**

- Time management and breaking down of functionality