**CITS3200 - Team J - Mentor Meeting 1 - 16/08/2018 15:05 to 16:20**

Time start: 3:05 pm (project mentoring 3:46 pm onwards) Time end: 4:20 pm

Attendees:

* Henry
* James
* Jason
* Josh
* Augustin

Agenda:

* How to go about with requirements analysis
* How to integrate separate components of the system?
* Strategies for test environments/test cases.

Notes:

* Interdependency between components, **form** connected to **database (**Avoid Scope creep?).
* Get requirements firm and **agreed**(with client)
  + high level of requirements needs lots of back and forth with client
  + analysis and requirement writing must be rephrased until unambiguous.
  + Avoid examples, be specific.
  + Agreed **upfront**, **testable components** should drive requirements writing.
  + Daniel’s current project required a “Design concept” which broke down a requirement with many components further into some more requirements. Always have “shalls”(the IVR system shall enable…), independent function in each requirement, segment them.
* We shouldn’t take no for an answer and be persistent about a sandpit environment.
* Well defined process of how software is integrated into actual practice (if taking the proof of concept route).

Some acronyms i didn’t quite get Daniel say…

CSIM - computer software integration model

CSCI? CSComponent? Computer Software Configuration Item

CSC: Computer Software Component

CSU: Computer Software Unit

MRI - master record index - hierarchical structure level 1 to 5

* How to show Client a storyboard so they can confirm you’ve understood and won’t waste valuable time doing hard backend for them to say: “oh but i meant this...” : Understand workflow, mandatory fields, ppt to show links leading to which links, visual stuff we can do without much work.
* How to come up with test cases?
  + Defined range of inputs?
  + Automate testing Capybara? <https://en.wikipedia.org/wiki/Capybara_(software)>
  + Mercury? [https://www.connective.com.au/mercury-software](https://www.connective.com.au/mercury-software/)
  + Quick runner & Quick test pro?

Mentor Meeting 1 Notes

16 August 2018

Questions from the email:

1. How often and under what circumstances to you decouple dependent tasks for immediate demonstrable progress at the expense of additional integration work to ‘connect’ the two decoupled systems after they’re complete?
2. Do you think our solution design (we will bring documentation along) is appropriate for the 9ish week timeframe? Do any difficulties we might have jump out to you?
3. What do you recommend in terms of CI/CD for this project? Do you think it would be worth the effort to automate deployments considering the client?
4. Have you previously done any integration work with Oracle RightNow? If so, how was the integration experience?
5. Is there any general advice you’d give us to help us achieve a better outcome on the project?
6. Have you used verification methods in any of your projects? What languages/tools/techniques did you use? How extensive is the use of such methods?

More questions?

A:

Get rid of examples in requirements. Make sure that the agreed upon requirements are very specific.

*What is the best way to define requirements with the client?*

Try to keep requirements independent (if you can).

Try to get access to sandbox / environments!!

Understand the process towards potential deployment. How to get the software integrated into their existing system.

MRI: Master record index. Hierarchical. Manage version numbers.

Storyboard. Wireframe. Mockups.

*Coming up with a comprehensive acceptance test?*

For fields which have a defined range of inputs, there are ways to automate the testing. Capybara.

Leave 20% of time for testing (generally). This is 20% on top of your average estimate for completion time.