Projects

- ## What Roles can IS and CS students play?

The general rule is that everyone should be contributing roughly equally throughout all stages of the project. Everyone in the team should have a general understanding of all current activities but you do want to take advantage of the specialist knowledge and skills of each team member.

During the sprints computer science students can obviously contribute to programming and other technical tasks, but can also be involved in system and acceptance testing and planning for the next sprint.

Information systems students can contribute to the development and implementation of system and acceptance tests. The acceptance tests are technical implementations of the acceptance criteria for your stories. Ideally these tests should be automated. Information Systems students can take on a leading role in liaising with your client team and preparing for the next sprint. There will be time in the workshops at the start of every sprint to do sprint planning but there is some preparatory work that can be done to streamline the process. Information systems students can use their modelling skills from IAB201 to lead the design of the data storage mechanism used in your project.

Depending on your backgrounds either computer science or information systems students can design the user experience or how the front-end of your system will work.

All students who have reached this far in your degree should have passed IFB104, or its equivalent. This means that all students are able to contribute to some aspects of the programming. If your programming experience is limited to IFB104 then your programming contribution to the project will at best be limited to building simple components or simple user interface screens.

- ## What can Information Systems students do during the development sprints of the project?

In Scrum sprinting, the design, build and testing is done simultaneously to produce a shippable product after each sprint. From this point of view some possible activities for Information Systems students are:

Prototyping and developing the user experience. This could involve creating mock ups of different options for parts of the user interface and showing them to your client team and other students/friends. You can document the feedback and describe your rationale for the final version of the user experience. This can be done informally if you don't have any Human-Computer Interaction experience or more formally if you do have some HCI experience. You can use various web prototyping tools to create examples of the user interfaces.

Refining the acceptance criteria into full user acceptance tests. This involves creating detailed test scripts that can be run either manually or automatically to test every aspect of each story delivered. You could also create a fully automated continuous integration environment that builds and tests the system.

Doing some user acceptance testing with your client to ensure that the completed stories meet their expectations.

Getting feedback from a broader set of potential users for your product (e.g. friend/family/other students). Get a variety of people to use your completed stories and get their feedback on issues or concerns they have with the system.

Clarifying details of stories in the current sprint with your client. This involves fleshing out the detail of what the story is to do and considering all of the optional activities that are part of a story. Towards the end of a sprint you can also be doing some planning with your client regarding the stories to go into the next sprint.

Creating models of the data that needs to be stored or managed by your system and then designing, implementing and testing your storage mechanisms. This may involve some prototyping and testing of alternative storage technologies.

Creating business models for the potential environment(s) in which your product could be used. You can then use these business models to inform both the technical design of the system and the user experience.

- ## Which development tools our team should use?

You are free to choose any appropriate tool that is suitable for the skills within your team. The key word is "appropriate". If you look at the unit outline for IFB299 you will see that one of the key learning outcomes is that you can deliver a small sophisticated computing system. The expectation is that you are in the process of becoming IT professionals who will be able to work in complex environments.

IFB299 provides the following generic guidance in terms of technology selection.

- Choose a complete **full-stack** web framework such as
    - PHP - CakePHP
    - Python - Django
    - Ruby - Ruby on Rails **(Strongly recommended if your team has minimum or no programming experience)**
    - Java - Play
    - C# - ASP.Net
- Mature framework should have a comprehensive end-to-end tutorial, for example Ruby on Rails Tutorial. If you don't feel like using Rails, have a look at the table of contents and find a similar tutorial for your preferred language
- Choose the deployment solution before committing to the framework, preferably Platform As A Service (PAAS)
    - Heroku
    - AWS Elastic Beanstalk
    - Azure
- Get skeleton app submitted into version control and deployed ASAP

Content management systems like Wordpress or Joomla are not appropriate as they constrain the website to fit into their structure. Simple application construction tools like Filemaker or Dreamweaver are not appropriate for developing the full application, as they work by constraining development options to simple structures. It would be possible to use tools like Filemaker or Dreamweaver to create the front-end of your application but you would need to develop the logic that drives your application on your own. A danger of tools like Filemaker and Dreamweaver is that they constrain you to work within their framework. You will find it difficult to create a robust, maintainable and extensible application with proprietary tools. They are famous from creating very poorly designed structures with no regard to proper architectural design. (See the week 5 lecture for details about architectural design and different standard software architectures.) Part of the quality of your project will be assessed based on your choice of software architecture and the design of the components in your system.

There is a reason why the vast majority of sophisticated applications that you use are developed in PHP, Java or .NET. They provide the flexibility to deliver systems that meet changing customer demands and the ability to integrate with other systems.