IFB299: Frequently Asked Questions

IFB299 has very large enrolments that result in many student queries. These Frequently Asked Questions (FAQs) are based on actual queries that IFB299 had received previously from students. Please refer to this document first in case you have a query. Let the tutor know if you have any unanswered queries. If you are still not satisfied, please send an email or contact the unit coordinator.

Queries are grouped in the order 1. Assessments, 2. Project, 3. Workshop, 4. Lectures 5. Team work, and 6. Unit

Disclaimer: In this document, some of the queries/comments/feedback were received from students via email, the Pulse and Insight surveys. These queries are reproduced as it is, i.e., without any attempts to improve grammar or spellings.

1. Assessments

- What is the Assessment Structure of IFB299?

The following is the assessment structure. The due dates explanation provided below the table.

Assessment Item Type	**Assessment Element Name**	**Team or Individual**	**Due Date (Explanation provided below)**
Project (50%)			
	User Stories, Release and Sprint 1 planning (10%)	Team	Midnight of your week 5 workshop day
	Release 1 (10%)	Team	Before midnight of your Week 9 workshop day
	Release 2 (10%)	Team	Before midnight of your Week 13 workshop day
	Personal Portfolio (30%)	Individual	Before midnight of Week 13
Development Activities (30%)			
	Reflective Journal (10%)	Individual	Throughout semester
	Tutor & Peer Observations (10%)	Individual	Midnight of your week 13 workshop day
Peer Review (20%)			
	Peer Review 1 (10%)	Team	Midnight of third day after your Week 7 workshop day
	Peer Review 2 (10%)	Team	Midnight of third day after your Week 11 workshop day

Due Dates Explanations

- What does “Midnight of your week 5 workshop day” etc. mean?

For example, if your workshop is conducted on Mondays then it will mean 11:59 PM of Week 5 Monday. And so on.

IFB299: Frequently Asked Questions

- What does “Midnight of third day after your Week 7 workshop day” etc. mean?

If your Workshop Day is	Midnight of third day after your Week 7 workshop day will be
On Monday	11:59 PM of Thursday of the same week
On Tuesday	11:59 PM of Friday of the same week
On Wednesday	11:59 PM of Monday of the following week
On Thursday	11:59 PM of Tuesday of the following week
On Friday	11:59 PM of Wednesday of the following week

- What does “Throughout semester” due date for Reflective Journal means?

  You will submit 3 entries in your Reflective Journal as follow:
  - First five weeks
  - Release 1 Activities (Sprint 1, Sprint 2, Demo, Peer Review etc.), Week 5 to Week 9
  - Release 2 Activities (Sprint 3, Sprint 4, Demo, Peer Review etc.), Week 9 to Week 13

- How come there are so many different due dates?

  IFB299 uses Agile Scrum method for project management. In Scrum, the project work is managed using weekly Scrum sprints. Your workshop day is used as the beginning of a sprint week from Week 5 onwards. All of your assessment tasks are surrounding your sprint cycles, and that’s why due dates are also organised accordingly.

- Complex assessment structure making it difficult to get a good grade. Not good for IS students

  First of all, IFB299 produced one of the best results in the entire faculty. Large number of IS students as well as CS students did well.

  Entire assessment structure is developed based on the university guidelines and reviewed to be clear, fair and equitable for team and individual work. Students will appreciate the manner in which these assessments items are designed as they become familiar with Agile SCRUM. The skills developed in IFB299 will help students to understand how information systems (software applications) are designed and developed in real life situation.

  Confusions on the assessment arise earlier due to students’ unfamiliarity of such structure. However, please do not hesitate to seek clarification from the tutor/unit coordinator if required.

- How do I/team submit our assessment work?

  Your team must use a Github/Bitbucket repository to upload/submit your team and your project work continuously throughout the semester. That is why you are required to share your team repository with your tutor.
  You must also submit your/team assessment work via blackboard. For example, for the first assessment item “User Stories, Release and Sprint 1 planning”, which is due in week 5, you will also submit an informative page indicating that your assessment work is available in your repository with repository details.

- Why can’t I use only Github/Bitbucket for submitting my assessment work?

From Semester 2, 2016, it is the university requirement to use Blackboard submission. Blackboard also facilitates an efficient process for tutors to provide the marks/criteria sheets and feedback to students in Blackboard grade centre.

- What is personal portfolio?

IFB299: Frequently Asked Questions

Personal portfolio is an individual assessment item. In this assessment item, you will submit your contribution to the team work by listing and describing the artefacts produced by you. The artefacts that you list as your own should exist in the repository with date/time stamps, and must be visible to all the team members.

- What is an artefact?

The following are some of the examples of the artefacts. Do not divide these artefacts in the line of CS and IS. All students are expected to work on each part of the project development during the semester, and free to submit any of the artefacts as part of their personal portfolio.

What did you do that made a meaningful contribution to the teamwork in producing the final solution	**Evidence of that work (artefact)**	**How did it fit in the team work/project work?**
Requirements provided by the client were ambitious and lacked enough descriptions. I held meeting with the clients and produced a better version	Date and Time of the meetings, Submit the new requirement document that you have produced	Team has clear understanding of the requirements which resulted in good user stories
Studies the estimation techniques for the project	Provide a details on the estimation methods	Improved project estimation, work breakdown, and deadlines
Performed the project ROI analysis	Provide the list of ROI methods that were considered and final ROI analysis	Improved client's understanding and potential benefits of the project
Conducted a thorough Root Cause Analysis and Sprint planning activities)	Provide details of the Root Cause Analysis methods (i.e. Fishbone Diagrams, Perato Analysis etc.). Details of the sprint planning	Improve project scope definition and product backlog. Improved project management process.
Refined the acceptance criteria of the following … stories into full user tests	List of user acceptance tests	Reduced revisions and rework, improved client/user acceptance.
Designed User Interfaces incorporating the user requirements	Provide the UI designs	Improved system usability and acceptance by the client
Study the client's business processes and modelled improved business processes using BPMN 2.0 process modelling language	Details of the process models	Accurate illustration of the business processes helped improved system design
Team was not sure how to write a business letter for the peer review. I did the research in this regard to make recommendations	Produce the evidence of your research and actual recommendations you made.	Business letter submitted reflecting your work
Helped team to draft the business letter	Submit the business letter with all the track changes showing contributions made by the team members and highlighting your contribution	Business letter submitted reflecting your work
Used scenario and storyboarding to explore and explain client interaction with the web site	Submit the scenarios used and corresponding storyboards. Explain how your approach improved the final design of the product	Include the description of the final product where your work is used by the team
Used Data flow diagrams to explain overview of the web application	Submit the web page information that shows the input and output of the data from the web pages.	

Developed SQL queries to generate reports	List the SQL queries and report design features	
Created mock ups of different options for parts of the user interface	Submit the mock ups together with all the feedback documents received from your team members/clients	final version of the user interface screen used in the project
Use the … web prototyping tool to create examples of the user interface	List the user interface examples	final version of the user interface screen used in the project
My team didn't possess enough programming skills, so I learned the …. Programming language by following the online tutorials/books etc.	List the tutorial work that you did.	The actual programs that you developed and used by the team
Use of flow charts to develop program logic	Submit flow charts that used to developed program logic	Actual program listing(s)
Did the survey/study of the various programming framework and recommended the client to use the … programming (framework)	List the survey result/study documents, and how did you convince the client to use the programming language (framework)?	Client decided to use the programming language(framework) for the project
Conducted a thorough analysis of modelling paradigms and recommended to the team	Provide details of the analysis	Improved system architecture designs
Helped developed/refined data model/part of the data model as part of the final data model.	List the working (rough work) of the data model that you develop/refined.	Final data model used in the project
Helped developed/refined the final database design	List the working (rough work) of the database design	The final database design
Developed the test data for the database tables based on the client feedback	Show the test data that you produced	Show the listing of the table with test data through SQL command/other mechanism
Developed SQL queries to generate reports	List the SQL queries and report design features	
Implementation of a module - a vertical slice of the site, which includes creation, update and deletion operations (CRUD)	Submit actual example such as a module that consist of a model, view and a controller, if the site follows MVC architecture	
Produced the automated build and deployment script	Submit these scripts	

2. 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.

IFB299: Frequently Asked Questions

- 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.

- The subject would benefit greatly from being allowed to select any “app” idea to develop rather than a limited selection. Either that or do not give groups choice on what THEY want to develop. From the real world experience, the clients should select the project (of the ones on offer) they want to developed from them and the develop group can build it to their specifications.

This comment puts too much emphasize on the actual product that is being developed but this is not the only aim of IFB299. We do want each team to come up with a great product but still the IFB299 focus is actually on the process of development of the product as well.

In real world you are paid to do the job but this is not the case in the university. Though, we do try to resemble the real world scenario as much as possible.

You will have more freedom, and wide varieties of the projects available in third year capstone project unit but still you will develop a product that you select.

3. Workshops

- Is workshop attendance compulsory?

The workshops in IFB299 are a very important component of the learning process. The activities in the workshops are based around the assessment for IFB299. Consequently it is very important that you attend all workshops.

- The client/developer meetings are joke. How is one team that is doing the work required to get good marks when the client team is not doing their work?

IFB299 allows each team to select a project that they feel comfortable with. So as far as IFB299 is concerned, this is not true because you as developer will get good marks even if your client team may not work as per your team expectations provided you have developed the project keeping in mind the agile development and submit all the required assessments meeting the criteria.

As part of a development team, you are learning agile project management technique (Scrum) to develop a complete solution for client needs. From this point of view, the client and developer team meetings are organised in the workshops.

The “Peer Review” assessment (20%) is based on each team’s client role. If your client team didn’t do good job, unfortunately, they will lose marks.

- One of the pillars of Agile was deliverables over project documentation but all we have done in first five weeks of workshops is creating project documentation.

This comment is right in saying that one of the pillars of Agile was deliverables over project documentation but wrong in saying that this is what students do in first five weeks. In first five weeks you developed i) user stories ii) product backlog and iii) release and sprint plans. These are scrum elements (not project deliverables) that you use to convince yourself (i.e. your team, your company etc.) to tell the client that what you will be able to develop in given time span (i.e. roughly thirteen weeks.) In week 13, the product that you demonstrate to your client (i.e. tutor) is the only product deliverable.

- How many releases and sprints are there in our workshops?

You will have two releases consisting of two fortnightly sprints each. Release 1 consists of two sprints namely Sprint 1 and Sprint 2 and Release 2 consists of two sprints namely Sprint 3 and Sprint 4.

The following table shows the weekly sprint duration:

Release 1	
Sprint 1, Week 1 **(Week 6)**	From Week 5 to Week 6, i.e. after your workshop day on Week 5 to before your workshop day on Week 6
Sprint 1, Week 2 **(Week 7)**	From Week 6 to Week 7, i.e. after your workshop day on Week 6 to before your workshop day on Week 7.
Sprint 2, Week 1 **(Week 8)**	From Week 7 to Week 8, i.e. after your workshop day on Week 7 to before your workshop day on Week 8.
Sprint 2, Week 2 **(Week 9)**	From Week 8 to Week 9, i.e. after your workshop day on Week 8 to before your workshop day on Week 9.
Release 2	
Sprint 3, Week 1 **(Week 10)**	From Week 9 to Week 10, i.e. after your workshop day on Week 9 to before your workshop day on Week 10.
Sprint 3, Week 2 **(Week 11)**	From Week 10 to Week 11, i.e. after your workshop day on Week 10 to before your workshop day on Week 11.
Sprint 4, Week 1 **(Week 12)**	From Week 11 to Week 13, i.e. after your workshop day on Week 11 to before your workshop day on Week 12.
Sprint 4, Week 2 **(Week 13)**	From Week 12 to Week 13, i.e. after your workshop day on Week 12 to before your workshop day on Week 13.

You will also plan a third release with Sprint 5 as a backlog. This sprint will have all the least important stories ("Would have" stories.)

- Can I form a team with students from different workshops?

No, this is not possible. As explained earlier, all the important learning as well as assessment activities occurs in workshops in presence of your team members.

4. Lectures

- There were many assumptions made by the lecturer/tutors…

The assumptions are made based on the prerequisite knowledge. For example, we assumed that CS students will know programming, while IS students will be aware of the programming concepts through their IFB104 study. IS students will know about the data modelling (database) while the CS students will be aware of the database through their prerequisite study.

JIRA and Selenium are introduced as self-learning tools. It is assumed that students will have ability to learn these tools through the use of these tools help facility and internet sources.

- **In terms of lecture content, the first half of the semester which focused on agile practices and such was very good. It was new, interesting and applicable. However the focus then moved to basic IT knowledge like system architecture and databases, which I believe any 2nd year IT student should already know off by heart.**

IFB299 receives students from different cohorts (CS, IS, Engineering etc.) who will have knowledge of some of the topics covered after Week 3 but not all. For example, CS student will know about the system architecture but will not know much about data modelling concepts etc. IS students will know about data modelling but will not have studied system architecture. Because students of different cohorts are required to work together it is important that IFB299 provide a common understanding of these concepts so that students can follow and participate in team work.

- **There is a whole lecture on Testing.**

Testing is important required component to develop software applications. In Scrum, Design, Build and Testing occurs simultaneously during a sprint. Many IT professionals build their career around testing activities of software development.

- The lecture times are awkward

Time tabling process is beyond the control of IFB299 unit as it is centrally controlled.

5. Teamwork

- I had members who easily did little to no work this whole semester and may still get a 5 or 6 for this subject. It frustrates me that QUT thinks this sort of group work is just like the real world. It is not. Teams consist of managers and people in charge that can pull under performers in line. As I am a fellow student, I do not have this power. A few slack members can pull the whole team down. Because of this I found myself working on this course more hours than any other subject.

IFB299 has many unit learning outcomes that are based on teamwork. Please refer to the unit outline for the further details. To avoid the situation that is mentioned in this query, IFB299 has come up with individual as well as team assessment work. If you are not happy with your team members' contribution, please let the tutor know immediately. Please keep all the written and other evidence so that the unit coordinator can deal with the situation fairly and amicably.

6. Unit

- I feel the unit is a bit slow, like we have to wait 4 weeks to start the project? Coming from computer science background where I am usually given coding in the first few weeks this is highly unusual.

IFB299 is **not a programming** unit but introduces you to new knowledge such as negotiation, conflict resolution, requirements management, agile development and

technologies specific to your solution choice, Each week of the unit study is very important from these points of view.

- None of us in our team know programming. I fear that I will fail the IFB299 unit.

Again, I repeat that IFB299 is not a programming unit. Your team may suffer in the programming part of Release 1 and Release 2. But this is also true for other teams consisting of good programmers if they do not have good user stories, release and sprint planning, user interface, database, testing etc.

In general an IFB299 team should do well if they overall project quality is high.

IFB299 do suggest using Ruby on Rails programming environments to teams who has limited programming experience. It is one of the easiest programming languages to learn as well.