Bachelor of Computer and Information Sciences

Contemporary Methods in Software Engineering
Semester 1, 2018

ASSIGNMENT 1B: Team Development

Contribution to final marks: 50%

1. Due dates:

Reflective Presentation: **in class week 11**
Electronic Portfolio of Artefacts and Evidence of Process (Blackboard Wiki): **5pm Friday Week 11**
Software Product and Source Code (Heroku and GitHub): **9am Monday Week 12**

NOTE: Each Team should submit ONE set of deliverables.

2. Team Effort

- This is a team assignment to be managed and submitted as a Team of 3-4 people.
- All Team members will receive the same mark for this assessment unless it is indicated in the Team Participation Form that contribution or participation was not equal.
- The lecturer may also negotiate an unequal assignment of marks under certain circumstances.

3. Purpose of Assignment

This assignment relates directly to the following course learning outcomes:

- *Describe the emerging challenges in current software development contexts*
- *Describe emerging trends in software development methods and explain the rationale for some of these*
- *Critically assess, compare and contrast the distinguishing features of a variety of software development approaches and methods*

4. Assessment Aims

- To research and gain deep understanding and insight of a new and emerging technology related to software development.
- To evaluate the potential future impact of this technology on software development.
- To understand the current challenges and limitations of the technology.

Secondary aims:

- practice researching
- evaluate the credibility of sources of information
- practice academic writing
- Practice using the online electronic databases of journals (e.g. IEEExpore, ACM, Web of Science, Scopus)

Your Product Brief (Product Owner Requirements)
The student advisers in our School have noticed that they spend a lot of time answering very similar questions from new and existing students about what papers they can and cannot take. The students often just need advice on co-requisites and prerequisites and which papers go together well to create a good program of study for different majors. They want options that make sense.

You have been asked to develop an AI-based ChatBot that can answer these student questions in a natural conversation style (written or voice). The hope is that this will free up the student advisers for the more valuable responsibilities they have. Students should be able to interact with the ChatBot online and on a mobile device.

We have funding until Friday 25th May 5pm, so must stop then.

The final product is not clear yet and part of your brief is to get clarity on the possibilities and get clear on what can be delivered in this time.

Your Software Development Practice Brief (Academic Requirements)
In your Blackboard Portfolio you should gather evidence of your software development process showing how you developed software iteratively, incrementally and collaboratively using an Agile way of working.

Create a team Wiki on Blackboard
- Create your team wiki portfolio by self-assigning yourselves into Blackboard Groups – these are your teams of three. Team members must all be in the same tutorial stream.
- Coordinate with your team members so that you all join the same team!
- Don't join a team without the approval of the other team members.

You should have evidence of your team process including:

Pre-sprint planning	
Scheduling and monitoring	What dates for sprints? [see below- this is fixed] How will we make progress transparent? What will be delivered every sprint? How will we split the work up? When and how will we re-plan? How will we make progress highly visible? How will we monitor progress in a sprint?
Development process	How will we split work up? How will we share code and tests? How will we integrate code What tools and technology stack will be used?
Quality assurance & testing,	What are the criteria for high quality for: Code The Product The development process The documentation The team What practices will we do to ensure quality is reached for each of these?
Risk	What are the main risks and how will these be managed?
Communications	How will the team regularly make sure they are all on the same page (shared understanding)? How will the team regularly make sure they are on the same page as the Product Owner? What information needs sharing when and with whom and in what form?
Team	How will the team get to know each other and keep together as a self-organizing team? How learn together? What roles and responsibilities will different team members have? How will you make collaborative decisions?

Initial design	What is the initial high-level architecture

Each Sprint	**Evidence of different practices and documentation related to:**
	Collaboration in Coding (e.g. GitHub activity, version control)
	Part of a product useful to the PO (source code and product)
	Quality assurance of code and product (e.g. standards, unit/acceptance testing and reviewing)
	Quality assurance of documentation (e.g. reviewing, version control)
	Continuous process improvement (e.g. video of retrospective meetings and list of actions to try in the next sprint)
	Team maintenance (e.g. examples of learning together and supporting each other, updated team agreement)
	Progress (e.g. screenshots of Trello or Asana)
	Re-planning (new sprint plans, update roadmap)
	Risk management (e.g. list of top 5 risks and how to mitigate against them-revisited every sprint)
	Sharing understanding with team and PO (e.g screen shots of communications, spring goals, definition of done, definition of ready)
	Getting feedback from PO on what is built (e.g. screen shots ofg communications, video of sprint review meetings, screenshots of modified product Backlog)
Handover	Product available to the client with appropriate details to use it.

ROADMAP
Preparation for Sprinting (Sprint 0)
Friday April 6th to Friday 27th April

Initial Product Understanding
Scan for alternative approaches to get a feel for what is important (EVIDENCE)
Decide on criteria for selecting an AI ChatBot framework (DELIVERABLE)

Initial Development planning and setup
Research and Select the technology stack to develop/test and deliver the ChatBot. (DELIVERABLE)
Set up your shared development environment and Tech Stack (EVIDENCE)
Design the high-level architecture of the ChatBot (DELIVERABLE)
Start Learn any new technology needed (EVIDENCE)

Initial Product Requirements
Develop an initial list of features and product backlog work items in the form of user stories. (EVIDENCE)
PO orders the product backlog. (EVIDENCE)
Decide (planning poker?) on what will be done by the end of sprint 1 and COMMIT to it. (EVIDENCE)

Initial Development Process planning and setup
Decide on and JUSTIFY (explain why) a development process to

- collaborate together
  - plan frequently
  - communicate important information
  - divide work up
  - integrate work
  - share work
  - make decisions
- assure quality,
- continuously improve, and
- manage risks.

(DELIVERABLE)
Develop initial team agreement together (DELIVERABLE)

Sprint 1

Monday 30th April to Friday 11th May (weeks 8,9)
Deliver initial ChatBot with limited functionality, proving the concept. Not deployed, some UI.

Sprint Reviews Monday and Tuesday 14th – 15th May (start of week 10)

Sprint 2
Monday 14th May to Friday 25th May (weeks 10,11)
Deliver final ChatBot deployed to Web and mobile

Final Handover Monday and Tuesday 28th -29th May (Week 12)

Marking Criteria
Product – 30%
Good quality code.
Good quality product.

Portfolio of Evidence of experimenting with practices and tools and improving – 70%
Covers all the areas specified for planning and sprinting.
Shows good team collaboration
Shows good practices and understanding
Shows good skills in tool use

