EWI3615TU – CS Minor Projects BigData and AI (5EC)

Authors Bart Gerritsen
Datum Nov 2019
Version 2
Status FInal

Course overview

The CS Minor Projects EWI3620TU (Gaming Projects) or EWI3615TU (Big Data & Artificial Intelligence Projects) are part of the Computer Science Minor. This course overview is about the latter projects only. In these projects, five student teams collectively study, define, work out, implement and document an assignment on their own, thereby demonstrating knowledge and skills learned in the Computer Science Minor. Each student contributes work corresponding to 5EC points (140h), amounting to some 700h of work per project.

Projects are scheduled throughout Q2. Project organization and task assignments are up to the student teams. The team organizes and documents the project, i.e., the development process as well as the deliverables, such that project progress can be duly assessed by the course instructors. To that end, a 30 minute weekly progress meeting will be scheduled with each team. Estimated number of project teams in this course: 13-17 organized in 2 clusters of 8-10 groups.

Apart from being documented in the documentation set, deliverables will also be presented and demonstrated in an oral midterm and in a final presentation of the project, therein meriting key process steps and design choices made by the team. A critical review of the project results and team performance self-assessment is an integral element in the discussion.

Learning objectives

1. Work out a data science (big data) project assignment in a team effort
2. Define and stage the assignment in a development plan, and implement and follow that plan
3. Apply big data processing techniques to a real world problem in a real world context
4. Apply machine learning/ artificial intelligence techniques in a real world problem
5. Design and implement a data processing pipeline in a real world context
6. Document project realization in an adequate and complete project documentation set
7. Review and evaluate team and individual performances in the project

Course agenda

Week 2.1	Course outlining and overview (Instructor) Group formation and instruction process (Instructors, groups) Project topic elicitation (Instructors, groups)
Week 2.2	Development plan draft; discussion, submission, approval, finalization
Week 2.3	Design plan, methods, tools and activities
Week 2.4	Peer review 1 Design activities, design validation

Week 2.5	Implementation
Week 2.6	Implementation, midterm presentation Peer review 2
	Christmas Leave
	Christmas Leave
Week 2.7	Implementation wrap-up, Testing
Week 2.8	Testing, Evaluation
Week 2.9	Final Presentations and project round-up

Content

Big data / machine learning & artificial intelligence-oriented data science projects, in which either:

- A research goal is being formulated, worked out, investigated and reported upon
- A research method is being designed, implemented and verified
- A tool/utility/application/infrastructure is designed, implemented and validated
- A data set (pipeline) is being composed or extracted, contained, cleansed, validated, documented and prepared for further research projects
- A data classification system is designed/discerned, documented and validated

Any combination of these could also be fitting, of course. Student teams may come up with their own proposal plus project objectives, and submit that for provisional approval to the Instructors. Final approval will be granted, once the development plan has been submitted.

Planning and organization

Teams plan and organize their own project activities. Once a development plan has been compiled, it will be presented to the instructors, for approval. Upon approval, the team can take up working on the project according to the plan. Relevant amendments to or diverges from the development plan will be brought to the table during weekly progress meetings, as soon as possible.

Attendance and time spending

Project teams are free to plan and spend their project time. The only mandatory part is the weekly progress meeting, for which a time table will be issued by the instructors. Teams may appoint a project leader, as seen fit. Teams are collectively responsible for project results and project progress.

Project development

A project development plan, as a minimum, describes the following items and stages:

- Project objectives, context, requirements and constraints
- Project approach, resources, priorities, results expected
- Design objectives, design strategy, critical features, risk factors, ‘to-avoids’
- Design documentation set layout, with configuration management
- Design validation
- Implementation planning,
- Testing approach, test planning, validation reporting
- Evaluation

Progress evaluation will focus on these development stages in connection to the time table, and failure risks identified. Configuration management seeks to manage changes (version control) against a baseline.

Documentation set

The documentation set is aligned with the Project Development Plan:

- Development plan
- Design plan, design document set, design validation document
- Implementation plan, version management plan
- Code base, code documentation set
- Testing and validation plan
- Test results, validation reports
- Evaluation document

Documents shall be brief and concise, and reflect choices, their motivation, and results. Documents may be integrated resulting in a few or even a single document, as seen fit, as long as the above topics have been adequately covered in it. There is no value in including irrelevant or elsewhere documented background materials. Documents do reflect different views possibly on different aspects of the development and the design. Different views shall collectively compose a all-treated (holistic) view on the project, the process and the products developed.

Teams may decide to postpone issuing documents to a later stage, when they have better insights in the design or solution details, but at their own risk. Before commencing with for instance the implementation, a sufficiently mature and agreed Implementation plan must have been issued and approved.

Documents may be on paper, electronic, or embedded in a CASE-tool, as long as they are feely accessible for assessment. For educational qualification purposes, a (copy of the) documentation set must eventually be handed in for assessment and grading.

Code resides on Gitlab, along with its documentation. The same holds for testing and validation reports. For educational qualification purposes, a (copy of the) code and documentation set must eventually be handed in for assessment and grading.

Assessment

Each group will receive a group score, based on the following rubric:

Development plan (approach, organization, quality and completeness)	GROUP	20%
Design plan (approach, organization, quality and completeness)	GROUP	20%
Implementation quality (achievement, risk management, quality assurance)	GROUP	10%
Code and testing quality	GROUP	10%
Evaluation completeness and quality (lessons learned)	GROUP	20%
Midterm presentation (presentation, clarity, maturity, completeness, questions)	INDIVIDUAL	10%
Final presentation (presentation, clarity,	INDIVIDUAL	10%

maturity, completeness, questions)		
		100%
Bonus for extraordinary individual contributions	INDIVIDUAL	+1
Malus for exceptionally poor individual performances	INDIVIDUAL	-1

A grade cannot be higher than 10. In the end, group grades are administered for individual team members in OSIRIS, but only if sufficient participation and contribution has taken place and could be assessed. Peer reviews are an integral part of the projects, to assess the involvement and contribution of individual team members. In all cases, Instructors decide, if applicable, commonly after consulting other team members.

Teams review, self-assess and evaluate their achievements. In case of (partial) failures, teams are expected to identify root causes (chain of failures) and indicate how they would do differently the next time to remedy. A well-evaluated failure can thus be turned into a lesson-learned, and considered in the grading.