# ISYS2102 Software Engineering 2

## Assignment Details

### *Aims of the project*

The project aims to provide students with the opportunity to manage themselves, their work and time and to gain insight into difficulties of team-based software development.

The following skills will be used:

* Requirements analysis
* UI design
* Prioritization of requirements and time-boxing (and cutting features to make a deliverable product)
* Team and conflict management
* Professional software development practices (testing, repeatable builds, documentation, source control etc)
* Design using appropriate patterns
* Programming to a deadline

### *Deliverables*

Each team must deliver the following:

* A technical report detailing the design of their solution, the difficulties faced during development and how the students overcame them, and which software tools were used and the reason why these tools were chosen.
* A source control url of the source code for their solution (this should include unit tests).
* A report of three design patterns (chosen at random in week 1) describing the patterns and when they should be used (reference to the solution is ok if appropriate).

Each team must also demonstrate a working product in Week 11 of the semester.

### *Cutting Features*

The specification given is detailed and contains many features that would be desirable for a management game of this type. Each team is responsible for deciding which of the features in the specification are required and which can be cut to save time.

The technical report should have a section describing which features were cut along with a justification. ***Delivering a working, tested product with 50% of the features is worth more marks than delivering a non-working product with 100% of the features.***

### *Changing or Adding Features*

Some teams may be tempted to add features to the game. This is encouraged, if there is a justification for adding the feature. ***Any added feature must be documented and justified in the technical report. Be careful not to work on additional features if your core game is buggy!***

Teams may want to swap a feature in the specification for a different, but similar feature. This is also encouraged, ***if the change is justified and documented. Changing the entire game from a development management simulation to a first-person shooter will not be allowed however!***

### *Part A - Marking guide*

Part A of the assignment is the delivery of the design patterns report and the demonstration of the functionality defined for the first sprint.

* Each team must create a product backlog, composed of two sprint backlogs. The contents of the first sprint backlog will be used to judge how well the team is performing when comparing the backlog to the functionality demonstrated in the sprint demo (week 6).

The sprint demo will be assessed according to these criteria:

|  |  |
| --- | --- |
| ***Grade*** | ***Required*** |
| PA | * All team members attend * 2-3 features demonstrated * Team can describe their design using a whiteboard (basic design only) |
| CR | * As above+: * 4-7 features demonstrated * Each team member can describe a feature in detail * Team can draw a detailed design on a whiteboard to describe the how a feature is implemented |
| DI | * As above+: * All features in the sprint backlog are completed or changed\* (changing a feature requires justification) * Each team member demonstrates knowledge of how the complete system works |
| HD | * As above+: * All features in sprint backlog are implemented to high standard (UI is professional, workflow is sensible) * Team can describe in detail the design decisions made and can describe the advantages and dis-advantages of the chosen design (team is expected to use whiteboard to show alternative designs) |

The design patterns report will be assessed according to these criteria:

|  |  |
| --- | --- |
| ***Grade*** | ***Required*** |
| PA | * Submitted on time * Report is logically laid out * Contains descriptions of the 3 allocated design patterns * Diagrams of patterns are correct |
| CR | * As above+: * Advantages and disadvantages of using each pattern * Simple examples of where the patterns could be used |
| DI | * As above+: * Good examples of where the patterns could be used * Alternative patterns to the allocated patterns explored and presented * No spelling mistakes or grammar mistakes |
| HD | * As above+: * Excellent examples of where the patterns could be used * Alternative patterns presented and justification of when choosing an alternative would be a better design decision * Alternative methods for achieving the same goal as the patterns presented and advantages and dis-advantages of the alternative methods explored in depth |

### *Part B - Marking guide*

Part B of the assignment is the delivery of a technical report concentrating on:

* Design of the devfortress game (including relevant diagrams, UI mockups etc).
* Difficulties encountered and overcome.
* Applicability of the design patterns taught in the course.
* Applicability of Scrum project management to the development of a class project.
* Changes made to the specification and justification for the changes.
* Additional constraints not listed in the specification.
* Technology stack and tools chosen and justification.

Each team must also provide a URL to the source control system containing the finished code of the game. ***If the lecturer cannot build the game and run it on linux using just this URL and any README or INSTALL instructions provided in the source control, then the team will have difficulty attaining high marks.***

Each team must demonstrate the finished game in the final sprint demo. No team is expected to create a complete game that corresponds 1:1 with the specification, however ***every feature missing or changed in the game that was in the product backlog must be justified in the technical report.***

|  |  |
| --- | --- |
| ***Grade*** | ***Required*** |
| PA | * Project report * Turn based game engine * Simple GUI * Game rules for applying skills to projects * Turns implemented correctly * Playable game (names, projects etc can be hardcoded) * Tests provided |
| CR | * As above +: * New developers are generated each turn * New projects are generated each turn * Game rules for applying the effects of food/drink and beer * More functional UI * Save game functionality implemented * Design of code is reasonable and logical * Tests check boundary conditions, and exceptional cases * Build script provided |
| DI | * As PA & CR +: * Skill advancement implemented at the end of each project * Special events are implemented * Project income and developer salaries implemented * Polished GUI showing profit/loss by project, summary screen of all developers and their current skills * High score table implemented |
| HD | * AS PA, CR & DI +: * Simple graphics implemented (photo’s or cartoons for developers, icons for pizza, coffee, beer etc) * Graphs and charts implemented where appropriate * Different difficulty levels implemented * Project bonuses implemented * Project related hardware and software costs implemented (servers, license fees etc) * Developer happiness implemented * Students add more special events * Students produce exceptional report * Design is extremely clear and logical |