Group 11 - Project Plan

**Table of Contents**

1. Project introduction. 2

Project outline. 2

Project schedule. 2

2. Conflict resolution plan. 2

Potential conflicts and resolutions. 2

 Withdrawal from the team by a team member. 2

 Disagreement between two members over a project matter. 2

 Personal issue between two or more members of the team 3

 Personal issue between the project manager and another member of the team. 3

 Repeated absence from meetings by a team member. 3

 Work from a member judged too poor compared to requirements. 3

 Conflicts arising from peer assement. 3

3. Phase plan. 3

Project phases. 3

 Analysis. 3

 Design. 3

 Programming. 3

 Testing. 4

 Presentation. 4

PERT Network. 4

 PERT Network. 4

4. Organisation plan. 4

Staff organisation. 4

 Team Members. 4

 Organisation of the project filespace: 4

Analysis team responsibilities. 4

 Overall Responsibilities: 4

 Project Phases: 4

Design team responsibilities. 5

 Overall Responsibilities: 5

 Project Phases 5

Programming team responsibilities. 5

 Overall Responsibilities: 5

 Project Phases 5

Quality Assurance team responsibilities. 5

 Overall Responsibilities: 5

 Project Phases 6

5. Peer assessment plan. 6

# Project introduction.

## Project outline.

After learning all sorts of things about Computer Science during two years, including tools, working and applications, we are now going to study one of its more widely spread form of utilisation, Software Engineering, as it is in real situations. For this, we are going to carry out a project from the scratch, trying to do it in a way as similar as possible to the way a software development team would do it, basing our work on what we have learned from the Software Engineering course.

This project is a development for a game designer referred to as the “customer”, of a strategy game based on a series of specifications gathered in a written document by this customer. The idea is to allow players to manage ant colonies and improve their performances as such by having them create designs of ant brains represented by a simple form of final-state machine. This involves the modelisation of the ants provided with the ant brains as well as the world they evolve in. We will then enable all of these elements to be merged into proper simulations and compute results and statistics as to which of the colonies was the more efficient. Though players can have their ants execute several kind of actions, this is done by using the food accumulation task as a performance indicator. The architecture of the game process itself, the choice of the way it should be played and built is then entirely up to us. We can choose :

* + - the type of visualisation – text, 2D, 3D,
    - the type of software – applet, full application,
    - the machine it is to be played on – console, PC,
    - the gaming features – scoring, tournaments.
    - Etc.

Our software house, composed of eight engineers is divided into several software teams – Analysis Team, Design Team, Quality-Assurance Team and Programming Team – each with a specific set of tasks and subtasks assigned. Just like in a real software development project, our work will not only be restricted to some programming, it will also have to reflect the full process of an efficient software engineering, including the planning, the analysis, the design, the documentation and the testing of the final product, commiting to organisation, procedures and standards described in a quality manual.

## Project schedule.

To be done.

This section should give an overall schedule of the important milestones of the project. It should be a few paragraphs long, and a more detailed schedule should be given in the phase plan.

# Conflict resolution plan.

## Potential conflicts and resolutions.

### **Withdrawal from the team by a team member.**

In this case, a meeting will be held between the remaining team members, with the focus around which individual is most suited to take on the extra work that has been left by the departing team member. This will be dependant upon the suitability and competency of each individual in the role.

Preferably, the person taking on the role will have done so voluntarily. If not, then a vote will be passed to decide who will take on the role. If no one gains a majority vote, then the project manager will make a final decision.

### **Disagreement between two members over a project matter.**

If this occurs, then those in disagreement need to present the issue to the project manager initially. If it’s a minor decision then the project manager will decide. If a bigger problem, such a design decision or something that will affect other sub teams work, then the project manager will present the problem to the other members at a weekly meeting. A vote will be passed, and if no conclusive result comes as result, then the project manager will make a final decision.

### **Personal issue between two or more members of the team**

If this occurs, then the problem needs to be raised with the project manager. This will remain confidential, and the project manager will act as mediator and resolve the issue. It is important those involved act in a professional manner, and slandering to other team members is not acceptable. If it is impossible for the team members to work together, then the issue will be raised with Martin Berger, and any further decisions made by him.

### **Personal issue between the project manager and another member of the team.**

If this occurs, another member of the team will act as mediator whilst the problem is resolved. It is vital that during this process that the team member does not ignore the project managers instructions during this time, and equally that the project manager does not discriminate against the other team member.

### **Repeated absence from meetings by a team member.**

In the case of a team member missing two consecutive meetings without telling any other team member, then the project manager will get in contact asking for an explanation. If this happens repeatedly, then the project manager will notify Martin Berger of the repeated absence, and this will be taken into consideration in the peer assessment.

### **Work from a member judged too poor compared to requirements.**

Should a member not withdraw but still provide unsatisfying work given other’s expectations, the matter would immediately have to be sorted out. This would initially be discussed between the manager and the person involved, where a possible agreement can be made, before putting it to the group, so to keep embarrassment to a minimum. Some issues can be due to personal problems getting in the way, but if this is just the consequence of some sort of laziness or lack of involvment in the project, it should then to be treated like a repeated absence. Also, if it is just that the member in question doesn’t have the necessary skills to efficiently accomplish the tasks he/she has been assigned and one or more other members then have to be chosen to help depending on available free time and similarity of tasks in the project. If a meeting is not held due to lack of involvement then the decision will be discussed by the group as per point 1.

### **Conflicts arising from peer assement.**

To be done.

# Phase plan.

* Project phases. This section should describe the phases of the project. Each phase should be broken down into tasks and subtasks. The allocation of tasks to staff should be described. A work breakdown structure for the project should be given.

## Project phases.

### **Analysis.**

* 1. Analysis modelling
  2. Setup change control system
  3. Requirements specification document \*milestone Monday 12 March, Spring Week 10
  4. Acceptance criteria document \*milestone Monday 12 March, Spring Week 10

### **Design.**

* 1. High level design
  2. Detailed design
  3. High level design document \*milestone Monday 12 March, Spring Week 10
  4. Detailed design document \*milestone Tuesday 1 May, Summer Week 3

### **Programming.**

* 1. Coding
  2. Refactoring
  3. Integration
  4. Optimisation
  5. Deliver source code \*milestone Monday 11 June, Summer Week 9

### **Testing.**

* 1. White box testing
  2. Black box testing
  3. Test specification document \*milestone Monday 11 June, Summer Week 9

### **Presentation.**

* 1. User Documentation \*milestone Monday 11 June, Summer Week 9
  2. Peer assessment \*milestone Monday 11 June, Summer Week 9
  3. Prepare presentation \*milestone Wednesday 13 June, Summer Week 9

## PERT Network.

### **PERT Network.**

To be done.

# Organisation plan.

## Staff organisation.

### **Team Members.**

* + - Andreas Nicholaou Analysis Team & Quality-Assurance Team
    - Mark Purser Analysis Team & Quality-Assurance Team
    - Eleanor Shakeshaft Analysis Team & Quality-Assurance Team
    - Ben Watt Analysis Team & Quality-Assurance Team
    - Oliver McCarthy Design Team
    - Mark Merriman Design Team
    - Robert Johnson Programming Team & Project Leader
    - Joanne Robert Programming Team

*Note: The Analysis Team and the Quality-Assurance Team have been merged to relieve pressure on these teams during the exam period.*

### **Organisation of the project filespace:**

* + - Version Control:

To be done.

* + - Software Forum: To be used as a discussion space, and PDFs of all Minutes and Activity Logs will be uploaded there by the Team Recorder.
    - Dropbox: To be used to hold documents which each member has access to and can edit. All versions should be then uploaded to [version controll].

## Analysis team responsibilities.

### **Overall Responsibilities:**

The Analysis team must interrogate the customer’s specifications and requirements, scrutinise the data and present the principal facts and tasks that the software engineering team must adhere to when developing and producing the game. This will be the basis for the creation of the software design and the set of requirements that will be validated when the software is done.

During the development the team will take meeting minutes, make sure documents are done appropriately and produce a team website where they will upload project deliverables.

Once the program is finished they must present the customer with documentation which will explain how they can install and use the program, within the licensing agreement.

### **Project Phases:**

The project website will store all the deliverables submitted during the project and will be completed by 1st March 2012. It will contain a list of group members, and any further information deemed necesaary to keep on the site. It will be made by Eleanor Shakeshaft and Andreas Nicholaou and is hosted at <http://dl.dropbox.com/u/12957105/Website/home.html>

Analysis modelling is the key componment which needs to be completed before the design team can start, this means that it should be completed before or sometime early in the Easter holidays to hand over. It will hold all the customer requirements and the requirements specification which will have details from the goals of the game or the objectives to the security requirements.

Finally the User Documentation will be finalised near the end of the project, this will reflect the product delivered, not the envisaged product designed at the beginning. It will comprise of 3 sections which need to be accessable via the team website, they are: the licensing agreement, installation guide and user manual. The installation guide is aimed at system admins who will intall the software. The user manual is aimed at the users of the software.

## Design team responsibilities.

### **Overall Responsibilities:**

The Design team’s responsibilities centre on the structure of the project by taking the information given by the analysis team and fleshing it out into a full design. This is split into two stages; the high-level design and the detailed level design.

### **Project Phases**

The high-level design involves taking the requirements specification and implementing it. The central part of this section is taking the classes and objects identified by the analysis team and expanding or adding to them, ultimately creating an architectural design which will be detailed with various diagrams in UML notation. This is also the stage where we decide on the coding style of the project, which will remain in place for the projects lifetime. Opportunities for concurrency are also identified, as well as local mechanisms within the system and the policies on how to handle them. Finally, a requirements cross-reference will be created to bridge the gap in-between the analysis model and the design model, detailing how the architecture has evolved from its inception.

The detailed level design is about taking the high-level design and detailing every aspect of it, taking the original architectural design and describing it in even more detail with UML diagrams. This includes detailed information on classes, object states, methods and the identification of abstractions and opportunities for inheritance. Changes that have been made since the high-level design will be identified and justified. The purpose of this part of the design is to provide a blueprint which will allow the programming team to implement it quickly and efficiently.

The design team consists of Oliver McCarthy and Mark Merriman, with the duties split into different areas, which will allow us to build the design separately before meshing them together as a final design.

## Programming team responsibilities.

### **Overall Responsibilities:**

The programming team are responsible for implementing producing the end software in line with the requirements of functionality and design made by the design team.

It is important that although the programming team has a freedom over how the final functionality and design of the software is implemented, it meets the requirements set out by the design team as accurately and efficiently as possible.

### **Project Phases**

There are numerous phases to the implementation of the design, which can be split into either high or low level models. The low level design is often dynamic and can change, so it is hard to predict the exact phases. However, the high level programming can be viewed as followed:

* Syntax Analyser for Ant Brain;
* Syntax Analyser for Ant World;
* Ant World generator;
* Game simulation including server-client functionality;
* Game visualisation;
* Game GUI; and
* Ant Brain for use in competition;

## Quality Assurance team responsibilities.

### **Overall Responsibilities:**

Like the other teams, the Quality Assurance team is responsible to their parts of the phase and organisation plans. The QA team is also responsible for ensuring that the other teams work to the specifications in the guidelines given and are responsible for taking the minutes of the deliverable reviews and any other group meetings.

The QA team will specify how the program should be tested, and write unit tests as the program is developed. Subsequently, the QA team will perform integration testing as the program gets larger, and finally validation testing to ensure that the program meets the specifications specified during the analysis phase.

To be added: details about Acceptanc Criteria.

### **Project Phases**

* + *Meeting minutes:*

To be done at every meeting.

* + *Program testing:*

To be done while the program is in development

# Peer assessment plan.

To be done.

This part of the project plan should describe how your group will evaluate the performance of the group members. You have complete freedom how to do this. One example of a peer assessement policy would be to agree that all members gave the same contribution in advance. Such a policy easy to implement, but might motivate some group members to do their fair share of the work.

Here are some suggestions how to evaluate group contributions (you are free to use or ignore them, and/or to make up your own).

* Relating positively to other group members: allowing other members to have a fair chance to contribute, responding constructively to each other's contributions.
* Being a teacher: explaining things to others, helping other group members to learn and grow.
* Being a social mediator: helping to detect and defuse conflicts in the group.
* Out-of-class attendance.
* Out-of-class participation.
* In-class attendance on group days.
* In-class attendance.
* Quality of work.
* Interest in the project.
* Overall level of participation.
* Understanding what is required.
* Suggesting ideas.
* Extracting something useful from the ideas.
* Performing routine tasks.
* Consolidating (e.g., drawing things together).
* Keeping the group going when things are rough.
* Sorting out problems.
* Initiative: Generating ideas for the activities and methods of solution,  Findings ideas from other sources.
* Commitment: doing a fair share of the work, meeting deadlines, attending meetings and being punctual.

I also suggest to make a note of how the group will keep scores, and what to do if a member of the group does not agree with their peer assessment. Conflicts arising from peer assessement need to be discussed in the Conflict Resolution plan.

Finally, I suggest that your peer assessment plan features a clause saying that all group members have to agree on (and sign off) the final peer assessment that is part of the last deliverable.