

Project Proposal

Semester 2, 2014

* Charlotte Paterson 1243532
* Gaoxin Huang 1136424
* Yue Li 1251124

# Team details

## Current Team Members

James Williams *Software Development & Computer Science*

jimiwillz@gmail.com

+64 21 253 1706

Takuma Sato *Software Development & Computer Science*

mambo69vn@gmail.com

+64 22 169 5634

Xiangyu Chen *Software Development*

cxyyouyou@gmail.com

+64 21 0277 8858

## New Team Members

Charlotte Paterson *Software Development & Computer Science*

pdy2956@aut.ac.nz

+64 21 154 7735

Gaoxin Huang *Software Development*

gfx0590@aut.ac.nz

+64 21 0231 7756

Yue Li *Software Development*

pbq6275@aut.ac.nz

+64 21 0817 9058

## Supervisor

Anne Philpott *Senior Lecturer*

anne.philpott@aut.ac.nz School of Computing and Mathematical Sciences

(09) 921-9999 ext 5216

## Clients

Helen Varley Jamieson *Artist and Original UpStage Project Coordinator/Founder*

Germany/Europe

helen@creative\_catalyst.com

Vicki Smith *Artist, Educator and Original UpStage Founder*

Wellington

vicki@UpStage.org.nz

# Team Roles

|  |  |
| --- | --- |
| **Role Name and Description** | **Assigned To** |
| **Problem Solver** – Coordinates and solves group problems. When a problem arises, if no group decision can be made the Problem Solver’s decision is final. | James |
| **Software Developer** – Performs all the programming tasks. | All |
| **Progress Tracker** – Reviews progress made by the team on a weekly basis and are responsible for making fortnightly blog posts on the developer blog. | Takuma and Yue |
| **Communication Manager** – Ensures communication within and outside of the team is upheld to high standards. Ensures communications with the clients and supervisor on behalf of the team are carried out in a timely and professional manner. | James and Charlotte |
| **Server Administrator / Technical Manager –** Responsible for server maintenance and setup. Ensures the server is fully functional. Responsible for ensuring that code is packaged up and deploying it to the appropriate environments or on the appropriate systems. | James and Gaoxin |
| **Quality Assurance Advisor** – **Ensures code is tested and any artifacts such as word documents and designs are examined and their quality ensured to be of a high standard.** | Takuma and Yue |
| **Schedule Coordinator** – **Ensures plans and schedules are made and followed by the team. Co-ordinates the planning so to avoid planning work when members are unavailable etc.** | Xiangyu and Charlotte |
| **Software Architect – Designs Software at high-level and regulates the technical standards of chosen tools and platforms** | Yue |

# Table of Contents

[Team details 2](#_Toc396174805)

[Current Team Members 2](#_Toc396174806)

[New Team Members 2](#_Toc396174807)

[Supervisor 2](#_Toc396174808)

[Clients 2](#_Toc396174809)

[Team Roles 3](#_Toc396174810)

[Table of Contents 4](#_Toc396174811)

[Terms of References 6](#_Toc396174812)

[Rationale 6](#_Toc396174813)

[Scopes and Objectives 7](#_Toc396174814)

[Research on possible architecture designs for New UpStage 7](#_Toc396174815)

[Current Version 3 maintenance 8](#_Toc396174816)

[New version of UpStage initiation 9](#_Toc396174817)

[Technical/Project Approach 10](#_Toc396174818)

[Extreme Programming 10](#_Toc396174819)

[Scrum 11](#_Toc396174820)

[Test-Driven Development (TDD) 11](#_Toc396174821)

[Pair-programming 11](#_Toc396174822)

[Kanban 12](#_Toc396174823)

[Design Sciences 12](#_Toc396174824)

[Project Plan 13](#_Toc396174825)

[Semester 2, 2014 13](#_Toc396174826)

[Semester 1, 2015 15](#_Toc396174827)

[Quality Assuarance 16](#_Toc396174828)

[Risk Management 16](#_Toc396174829)

[Issue Log 17](#_Toc396174830)

[Change Management 17](#_Toc396174831)

[Supervisor Meeting 17](#_Toc396174832)

[Developer Blog 17](#_Toc396174833)

[Mid-Project Review 17](#_Toc396174834)

[Pair-Programming 17](#_Toc396174835)

[Test-Driven Development (TDD) 18](#_Toc396174836)

[Code Standards 18](#_Toc396174837)

[Skills and Knowledge involved 18](#_Toc396174838)

[Personal Skills 18](#_Toc396174839)

[Professional Skills 20](#_Toc396174840)

[Technical Skills 21](#_Toc396174841)

[Project Deliverables 21](#_Toc396174842)

[COMMUNICATION 22](#_Toc396174843)

[Client 22](#_Toc396174844)

[Supervisor 22](#_Toc396174845)

[Team 22](#_Toc396174846)

[Cost 23](#_Toc396174847)

[Disclaimer 24](#_Toc396174848)

[References 25](#_Toc396174849)

[Revision History 26](#_Toc396174850)

# Terms of References

Artists, schools, universities and community groups have experienced the creative online concept of UpStage in its ten-year life span. It is an open source web-based application that allows users to use multiple avatars and communicate with others that are on their shared ‘stage’ in real time. Users can communicate over the stage as either a player with full access to audio clips, images called props, backdrops and drawing tools, or as an audience member who can use the chat box. UpStage has been created in a way so that users have no need to download any software to access the system ensuring a wider variety of access for users.

# Rationale

Recently we were able to experience a walkthrough with our clients. The public was also invited to experience this short tutorial on the functionalities of UpStage. This showed a full spectrum of the user interface as well as emphasised the fact that UpStage has existing users. Therefore the UpStage team is required to maintain the current system and ensure that any problems are attended to, to allow users the continuing benefits of the UpStage experience. As the system was only recently updated to version 3.0, most bugs from the last semester have been attended to. However the existing code has many flaws and is very difficult to work with. The system proves difficult to test and requires constant patching, which doesn’t improve the state of the system. As this is an ongoing project it is very important that continuing developers are able to add to the system, which requires clean code that adheres to coding standards. However the current code does not achieve this, with incomprehensible naming conventions it is difficult to read and also to test. So a re-evaluation of the current development software, is decided among clients, and developers the best option to build a new product and ensure its continued success in the future.

Also due to the increasing commonality of smart phones and mobile devices it is important to look at the web based system in new light. Our clients want to continue the online experience with more focus towards a mobile platform. Therefore while trying to find a solution to replace the current product and choose a better development platform we have the opportunity to include research towards a mobile styled platform. This will allow us to both meet the client’s new requirements, as well as prepare the system for future developers, and ultimately extend the lifetime of the UpStage system.

# Scopes and Objectives

## Research on possible technologies for New UpStage

The new team will consistently work with the existing team in order to choose an architecture/technology for redesigning new UpStage product. By researching a wide range of technologies, our knowledge base will expand and there will be sufficient resource for new UpStage product. This will ensure our capability to undertake the requirements and skills for implementing the new design.

By the end of semester 2 2014, we will have a set of technical approaches for new UpStage, which include:

* Replacement for outdated technologies such as ActionScript, Twisted and Flash (not sure Twisted as I found the latest Twisted update was in May 2014)
* Proper testing framework
* IDE development
* Coding Standard

## Current Version 3 maintenance

As the new members of the Upstage team, we will need to obtain knowledge and experience from the current software as we will shortly carry on the regular maintenance of UpStage Version 3 from the other team. At this stage, Version 3 is stable and was released in 2014 for celebrating its tenth year birthday event. Since the Version 3 was released, bugs and issues remained while players interact with each other on the stage. The current and previous UpStage teams had put much effort into bug fixing. Most issues were resolved. As the state of UpStage is an ongoing project, there will be unforeseen bugs and change requests from our client. The current version maintenance remains an important item in our project objectives. The up-to-date report of bugs shows there are 59 issues are waiting to be fixed.

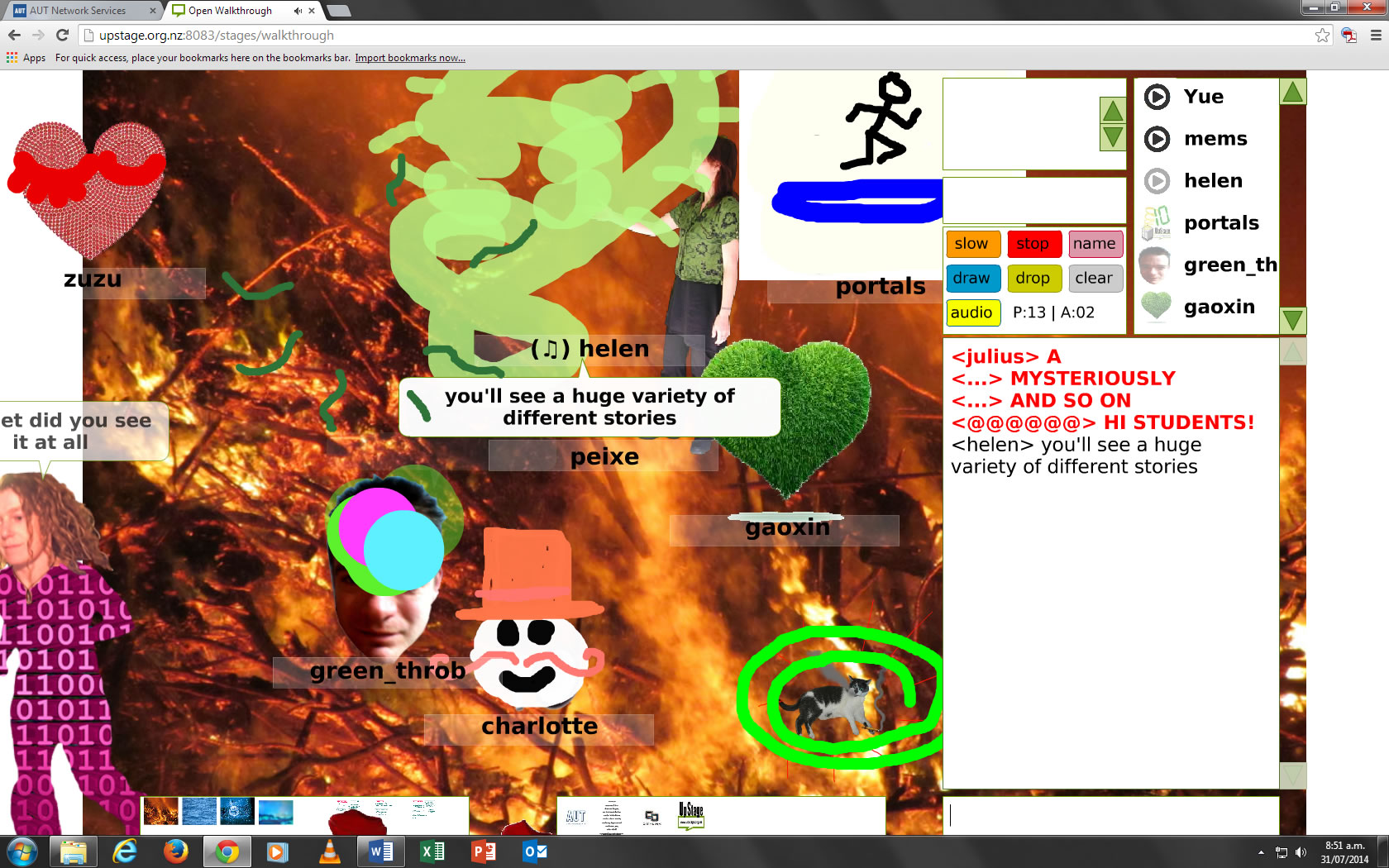


FIGURE 1. Chat board only shows one line of regular chatting message

From our first walkthrough with Helen and Vicki, we found the audio tools seemed to not stop as expected and messages disappeared from Chat board when players used certain commands. As a result, the current Version 3 maintenance will remain its priority through the project.

## New version of UpStage initiation

In the second half of our project, we will start implementing a new software for UpStage. We will also expect team changing (3 of current team members will graduate and 2 to 3 new team members will join in). We will spend time with new team members on the software implementation and make sure our work is correctly passed on to the new team members.

By the end of our project, new UpStage will be under development. According to current team’s motivational document, the new product will have the most functionality as current Version 3 product (Appendix A).   
The improvement of new UpStage will have:

A better flexibility on extending current code.

A testable development environment.

# Technical/Project Approach

## Extreme Programming

Extreme Programming is a software development methodology which is a kind of Agile Methodology. There are five core values in Extreme Programming and they are communication, simplicity, feedback, courage and respect. In fact, the main purpose of XP is to improve project quality and responsiveness to improve customer requirements. Comparing with traditional project, Extreme Programming lists the project practice and focuses on the points to try doing these practice best. Extreme Programming ignores others which do not list in the list. Kent Beck (1999) points out that XP can bring a lot of benefits to small team .As for us, we are a small team and there are a lot of methodologies in XP worked for small team, such as pair-programming. Besides, from our realistic situation the team members will be changed annually. Thus we should pay the attention in most important things in this limited time and try not to leave uncompleted tasks for next team members. In this way, the characters of XP is suitable for our project.

## Scrum

Scrum is an agile framework for completing projects. Scrum is useful to transform project management in projects. By using Scrum, the project would be more Agile, to react more quickly and respond more exactly to the change (Paulk, 2013). There are three ceremonies in Scrum (sprint planning, daily Scrum meeting and sprint review). The ceremonies contribute to improve the quality of the project. Besides, product backlog and sprint backlog benefit to promote the plan in project and to control project. Therefore, Scrum would enable our project to be more manageable.

## Test-Driven Development (TDD)

Test-Driven development is a common agile software development technique which is the Methodology of test-first development. In fact, it is growing in many development contexts associate with Extreme Programming (Ambler, 2008). As for the situation of our project, we have a current version while we are preparing for the new project. In this case, we can try TDD in new project based on current version, which enables us to design the new version on the right way.

## Pair-programming

Pair programming means code is designed by two programmers working together. One programmer manages the workstation and thinks about the detail of code. Another programmer is focused on the overall situation and is reviewing the code that is coded by the first programmer. After minute to hour periods, they swap their roles to develop project (Arthur, 2002). Based on our condition, we have to work on school computers which are dedicated to Upstage, so pair-programming enables us to work together and improve quality assurance of project.

## Kanban

Kanban is a new methodology for controlling a process of software development in a high-efficient way. Indeed, it is based on a simple principle which is agreeing on a limit to work-in-process (David Peterson, 2009). In our project, it enables the team to deal with the processes in project stage by stage, which can help the team limit items of work-in-process in order to improve efficiency in project.

## Design Sciences

During semester 2, we will firstly identify the difficulties and problems of current project so that we can improve the quality of the new product. Current software development methods are not suitable for the project to achieve our goal. We looked into Design Science and found its research process can be used for researching a new approach of UpStage product.

In their article, Offermann, P., Levina, O., [Schönherr](http://dl.acm.org/author_page.cfm?id=81375596365&coll=DL&dl=ACM&trk=0&cfid=400186133&cftoken=91518230), M., & Bub, U. (2009) suggested a research process that improves the quality of Information System research. The processes involve in Problem Identification, Solution Design and Evaluation. We think such method can be used in our research on coming up the new software design for UpStage.

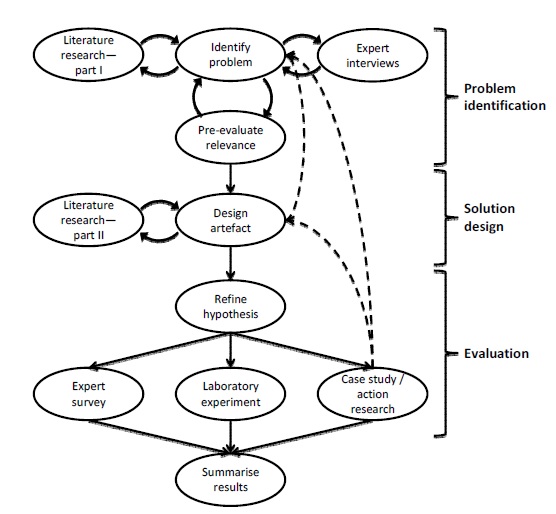


FIGURE 2. Offermann et al. (2009)

# Project Plan

## Semester 2, 2014

In the first semester, we will research over a wide range of technologies for a new software design and choose suitable architecture for new UpStage in the following semester. In semester 2, we will follow processes from Design Sciences, which we found suitable for the future UpStage development. By doing this, we will have adequate knowledge to choose suitable technologies for new UpStage development in the next semester. During semester 2 2014, we will also keep current UpStage V3 running in order and fix bugs if possible.

|  |  |  |
| --- | --- | --- |
| **UpStage Semester 2 2014** | | |
|  | V3 Maintenance | Design New Software |
| Week 1 |  |  |
| Week 2 |  |  |
| Week 3 | Regular Maintenance and Developer Blog Update |  |
| Week 4 |  | Gathering Current Issues of UpStage and Define research objects |
| Week 5 | Regular Maintenance and Developer Blog Update | Research on Java equivalent of Twisted, ActionScript and Flash plugin, Ajax? |
| Week 6 |  | Develop Prototypes for the researched technologies |
| Week 7 | Regular Maintenance and Developer Blog Update |
| Week 8 |  | Evaluation and feasibility on Ubuntu server |
| Week 9 | Regular Maintenance and Developer Blog Update | Evaluation & Summary |
| Week 10 |  | Further research on the corresponding topics |
| Week 11 | Regular Maintenance and Developer Blog Update | Develop Prototype |
| Week 12 |  |
| Week 13 | Regular Maintenance and Developer Blog Update | Evaluation and feasibility study |
| Week 14 |  | Summary |
| Week 15 | Regular Maintenance and Developer Blog Update | Plan the development strategy |

## Semester 1, 2015

As there uncertainty remains in the future development of UpStage, we would like to assume that once the new version of UpStage is evaluated and approved by our supervisor, we will start implementing the new software over next semester and build up the major functionality for the new product. We will also need to give training to our new team members so that we can pass our knowledge and work to the continuing team.

|  |  |  |  |
| --- | --- | --- | --- |
| **UpStage Semester 1 2015** | | | |
|  | V3 Maintenance | New UpStage Implementation | School Activities |
| Week 1 | Regular Maintenance and Developer Blog Update | Assign User Stories & Sprint 1 Starts |  |
| Week 2 |  | Sprint 1 Implementation |  |
| Week 3 | Regular Maintenance and Developer Blog Update | Sprint 1 Implementation | Mid Project Review, |
| Week 4 |  | Sprint 1 Ends, Product Backlog & Sprint Review, Sprint 2 starts |  |
| Week 5 | Regular Maintenance and Developer Blog Update | Sprint2 Implementation |  |
| Week 6 |  | Sprint2 Implementation |  |
| Week 7 | Regular Maintenance and Developer Blog Update | Sprint2 Implementation |  |
| Week 8 |  | Sprint 2 Ends, Sprint 2 Review & Product Backlog, Sprint 3 Starts |  |
| Week 9 | Regular Maintenance and Developer Blog Update | Sprint 3 Implementation | Client's feedback |
| Week 10 |  | Sprint 3 Implementation |  |
| Week 11 | Regular Maintenance and Developer Blog Update | Sprint 3 Implementation |  |
| Week 12 |  | Sprint 3 End, Product Backlog & Sprint 4 Starts |  |
| Week 13 | Regular Maintenance and Developer Blog Update | Sprint 4 implementation & Project Summary | Poster, Reflective Reports, Portfolio and Final Project Presentation. Supervisor feedback |
| Week 14 |  |
| Week 15 | Regular Maintenance and Developer Blog Update | Sprint 4 implementation & Passing development roles |  |

# Quality Assuarance

Due to the uncertainty of future development, we are aware of several concerns. Quality Assurance is vital for the project success. We will spend time on research in order to produce an executable product plan for the new promising UpStage. As the consequence will be vital to the future project execution, we prepared such methods to assure our progress as scheduled.

## Risk Management

Risk Management will be presented in documents. We will record all the concerns from either team or the client, so that all the concerns can be discussed and possibly resolved during meetings.

## Issue Log

Issue Log will be used for recording any issue we cannot resolve. The log will be discussed during supervisor meeting or client meeting regarding the type of the issues.

## Change Management

Change Management is for how changes to be managed. As UpStage is an ongoing project, how to manage changes will be frequently encountered. There are two changes that we will encounter frequently.

## Supervisor Meeting

Supervisor Meeting is a way for the team to keep the communication with supervisor. The meeting will also help us encountering difficulties that the team cannot resolve.

## Developer Blog

Developer Blog is used as a way to reach UpStage community. The development team shall receive feedback from the community and clients. It is also a good way for getting feedback from users.

## Mid-Project Review

Mid-Project Review will discuss the achievement in the first half of our project. It will also sum up what project goes well and

## Pair-Programming

Pair-Programming is a software development practice that there are a driver and an observer. Indeed, the observer reviews and checks the code while the driver is typing in the code. It enable programming to reduce the risks in project. In this way, pair-programming brings a lot of benefit to the quality of the code.

## Test-Driven Development (TDD)

Test-driven development is a methodology of test-first development. In this case, we are preparing for the new project. Using TDD in new project based on current version to ensure the new version will be on the right way, which enables the project to be in high-quality level.

## Code Standards

UpStage is an open-source project and many programmers will code it. Coding standards are to make ensure readability and maintainability of the source code. In order for other member to understand and maintain the code. In this way, we can revise and continue the existing version of UpStage’s coding standards document, which promotes the quality of project.

# Skills and Knowledge involved

### Personal Skills

**Comprehending**

To understand the existing work. As for new team members, we will understand the current situation of the project within the first month, which contributes to joining the project work. Indeed, we are able to read previous materials to improve its efficiency.

**Learning**

To learn new programming language and knowledge. Each team member will learn the programming language which may be used in this project.

**Communication Skills**

To build a good communication environment in the project. In order for the project to move forwards, each team member will need good communication with the clients, supervisor, and other team members.

**Problem Solving**

Each of team member will be able to deal with problems and face challenges. Indeed, team members will try to handle with the problems by themselves first in order to improve their ability for problem solving.

**Self-management**

Each of team member should manage time, promote their work efficiency, understand how to be responsible for assigned tasks and improve the quality of their tasks.

**Teamwork**

To learn how to work as a team. Each of team member must be able to work together effectively, hold good meetings, retrospective meetings, and online discussions. Any activities performed independently will be reported to the team.

**Attitude**

All team members will have a positive attitude to work in team environment and treat other members with respect.

### Professional Skills

**Project Tracking**

All work and time used in the project must be recorded and documented. Such as meeting minutes, change log, issue log, and so on. The documents will be as references and evidences in the future. For example, the documents can give a reference to us, when we face issues and implement new project.

**Quality Assurance**

To make sure the product with high quality, following the project requirements and testing is to be done before the release.

**Research**

To hunt for more information about our project. Before making a plan, we should do research for ensure which plan or technical is suitable for our project.

**Risk Management**

To have a good plan to deal with risks, when we face problems. Before we start to design the project, the team should have risk management to control the risks.

**Time Management**

The team members must have a professional schedule and plan for project. In order that the assigned tasks can be completed on time and reach the goals of project.

**Project Management**

To manage the project appropriately, the new team members will be adapted to team norms, forming and following a high-quality plan and track progress. It enables the team to improve quality of standard on our work.

### Technical Skills

* Python 2
* HTML and XHTML
* JavaScript
* ActionScript 2
* CSS
* Linux/Debian OS
* Terminal and Shell
* GitHub (for code)
* Subversion (for documentation)

# Project Deliverables

Following is a complete list of all project deliverables:

|  |  |
| --- | --- |
| Deliverable | Description |
| Project Proposal | An initial plan for the team throughout the project. It presents the goals that we would like to achieve for UpStage Project |
| Project Portfolio | All the documentation through semester 2, 2014 to semester 1, 2015 |
| UpStage V3 Patching | Improved bugs and changes requested by the client |
| New UpStage Product Initiation | The foundation of new UpStage |

# COMMUNICATION

## Client

Due to our clients reside in different locations, it is difficult to manage physical meetings through our project. The current methods for meeting our clients are:

* Emails
* UpStage Activities (walkthroughs, Developer Blog update)
* Physical Meetings with Vicki

## Supervisor

We have set up regular meetings with Anne on 3:30 pm Thursdays in semester 1.

* Emails
* Supervisor Meetings

## Team

We created a new Facebook group page to keep our communication channel open at all time. The communication will improve the team dynamics so as for the team performance.

* Facebook Group Page
* Team Meetings
* Emails
* Google Drive
* Other activities with our supervisor or client

# Cost

UpStage is an open source software; therefore, there is no cost for resources required in this project. We have included the hours that we will spend on the project over the two semesters.

|  |  |  |
| --- | --- | --- |
| Name | Hour Planned | Per Week |
| Charlotte Paterson | 300 | 10 |
| Gaoxin Huang | 300 | 10 |
| Yue Li | 300 | 10 |

# Disclaimer

**Auckland University of Technology**

**Bachelor of Computer & Information Sciences**

**Research & Development Project**

**Clients should note the general basis upon which the Auckland University of Technology undertakes its student projects on behalf of external sponsors:**

*While all due care and diligence will be expected to be taken by the students, (acting in software development, research or other IT professional capacities), and the Auckland University of Technology, and student efforts will be supervised by experienced AUT lecturers, it must be recognised that these projects are undertaken in the course of student instruction. There is therefore no guarantee that students will succeed in their efforts.*

*This inherently means that the client assumes a degree of risk. This is part of an arrangement, which is intended to be of mutual benefit. On completion of the project it is hoped that the client will receive a professionally documented and soundly constructed working software application, some part thereof, or other appropriate set of IT artefacts, while the students are exposed to live external environments and problems, in a realistic project and customer context.*

*In consequence of the above, the students, acting in their assigned professional capacities and the Auckland University of Technology, disclaim responsibility and offer no warranty in respect of the “technology solution” or services delivered, (e.g. a “software application” and its associated documentation),both in relation to their use and results from their use.*

# References

Ambler, S. W. (2008). Scaling test-driven development. *Dr.Dobb's Journal, 33*(2), 71-73. Retrieved from ProQuest database.

Beck, K. (1999). Embracing change with extreme programming.[*Computer,*](http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=2) *32*(10), doi: [10.1109/2.796139](http://dx.doi.org/10.1109/2.796139).

English, A. (2002). Extreme programming. *Network World, 19*(4), 60. Retrieved from ProQuest database.

Kollanus, S. (2010) ICMM—a maturity model for software inspections. *Journal of Software Maintenance and Evolution: Research and Practice, 23*(5), 327–341. doi:10.1002/smr.456.

Offermann, P., Levina, O., [Schönherr](http://dl.acm.org/author_page.cfm?id=81375596365&coll=DL&dl=ACM&trk=0&cfid=400186133&cftoken=91518230), M., & Bub, U. (2009) Outline of a design science research processs. In P.A. [Vaishanvi](http://www.google.co.nz/search?tbo=p&tbm=bks&q=inauthor:%22Vijay+Vaishanvi%22&source=gbs_metadata_r&cad=2) (Ed.), *DESRIST '09: Design Science Research in Information Systems and Technologies, 7*. NY, USA: ACM. doi: [10.1145/1555619.1555629](http://dx.doi.org/10.1145/1555619.1555629)

Paulk, M. C. (2013). A scrum adoption survey. *Software Quality Professional, 15*(2), 27-34. Retrieved from ProQuest Central

Peterson, D. (2009). *What is Kanban?*

Retrieved from www.kanbanblog.com

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author/Modified by |
| 03/08/2014 | Version 0.1 | Initial draft | Charlotte, Gaoxin & Yue |
| 04/08/2014 | Version 0.2 | Refined proposal structure | Charlotte, Gaoxin & Yue |
| 05/08/2014 | Version 0.3 | Fixed formatting | Charlotte, Gaoxin & Yue |
| 07/08/2014 | Version 0.4 | Removed duplicate content | Yue |
| 09/08/2014 | Version 0.5 | Editing the content  Cost needs attention in the new version | Gaoxin & Yue |
| 11/08/2014 | Version 1.0 | Completed all the missing parts | Charlotte, Gaoxin & Yue |
| 18/08/2014 | Version 1.1 | Edited content requested by Anne | Yue |
| 20/08/2014 | Version 1.2 | Rewrite objectives | Yue |

Appendix

**UpStage’s Functionality**

**Audience**

* Can access a link to find user’s local time in the homepage
* Can enter a stage
* Can send text messages in the stage
* Can register to be a Player
* Sign in with the registered account

**Player**

* Has the same functionalities as audience
* Can log out
* Will have tool bar in the stage if the player is assigned to this stage
* Can put avatars in the stage
* Can move the avatars in the stage
* Can stop the avatars in the stage
* Can select a prop at the bottom right side of the window
* Can select a background at the bottom left side of the window
* Can clear the avatars in the stage
* Can play audio in the stage
* Can turn volume up/down for audio in the stage
* Can stop a single audio
* Can stop all audio
* Can rename their current avatar
* Can draw line with chosen color in the stage
* Can select the type of line to draw
* Can change the thickness of line to draw
* Can clear the line

**Creator**

* Can log out
* Create a new Stage
  + Enter the stage’s full name and short name as a url
* Edit a stage
  + Modifying the stage’s full name and short name for url
  + Modify the splash message
  + Can choose if the debug messages are printed on the backdrop
  + Can change the props and backdrops’ color
  + Can change the chat window’s color
* Upload media so that it can be used in stage
  + Choose the type of media for uploading: avatar, prop, backdrop, audio or video-avatar
  + Modify media’s name and tags
  + For avatar, a voice and be chosen
  + The chosen voice can be tested
  + The number of frames can be selected
  + A local file can be chosen to attach
  + Choose the stages to assign the media to.
  + For prop, the number of frames can be selected
  + Choose the stages to assign the media to
  + For backdrop, the number of frames can be selected
  + Choose the stages to assign the media to
  + For audio, choose the type of this audio: sound effect or music
  + A local file can be chosen to attach
  + Choose the stages to assign the media to
  + For video-avatar, choose from existing streams in the /media/video directory
  + Choose the stages to assign the media to
* View an uploaded media
  + Display specific media by choosing filters
  + Add more filter to display an eligible media
  + Can remove the filter
  + Can reset the searching filter and result
  + Search by media’s tags
  + Edit an uploaded media by modify name, voice and the stages to assign
  + Can display the detail of the media
  + Can save changes
  + Can delete the media
  + Delete the media even if in use
* Edit players:
  + Can modify their password
  + Can modify their email
  + Can modify their user type
* Enter a stage
  + Can send text message in the stage
  + Be assigned to a stage :
  + The message sent can be read out as a chosen voice
  + Has the tool bar(same toolbar as player’s) in the stage if the creator is assigned to this stage
* Can create a new player account
  + Can enter the username, password, confirm password and email address
  + Select the type of the new player account: player, maker, unlimited maker, admin or creator
  + Able to save
* Can edit existing player details
  + Can display the detail of all existing player accounts
  + Can select one of them to change email, password
  + Update the changes
  + Delete the player’s account
* A link to edit page mode
  + Can edit the homepage
  + Can edit the workshop
  + Can edit the player page
  + Can edit the stages page
  + Can edit the sign up page
  + Submit the changes
  + Reset the page to be default

**Maker**

* Has the same functionality as creator except they can’t create new players or edit.
* Has an upload limit for file sizes when uploading files (10MB)

**Unlimited Maker**

* Has the same functionality as maker
* Has no limit for file size when uploading media

**Admin**

* Has the same functionality as creator but can’t edit or create creators.
* Has an upload limit for file sizes when uploading files.