UpStage 2014

Motivation and Expectations for UpStage Redesign

Clients

Vicki Smith

Helen Varley Jamieson

Supervisor

Anne Philpott

Team Members

James Williams - 1120763

Xiangyu Chen - 1255801

Takuma Sato - 1259130

Charlotte Paterson - 1243532

Yue Li - 1251124

Gaoxin Huang - 1136424

Last modified: 20/10/2014
Version Number: 5.4

Contents

1. Introduction

The UpStage software had its 10th birthday in January this year. The two AUT team members at the time and the team supervisor attended the celebrations in Wellington with the two clients Vicki and Helen. A meeting was held to discuss the current state of UpStage and the need for a redesign. It was agreed that the current version of UpStage had reached its limit as a software as a result of continuous development. It had become too hard to work with, particularly in efforts to find and fix bugs. The future of UpStage lay in a complete redesign, it was decided.

The aim of this document is to identify and record the motivations behind developing new UpStage, as well as to clarify the expected features of the replacing software. As developing new UpStage will be a long-term project, this document also aims be used as a reference for the future team members to gain direction of development as well as to aid the ability to make decisions when necessary. The first section of this document describes the background and stake-holders of UpStage. The next section 2 version 3 which illustrates the motivation to have a new software. The following section describes known functional and non-functional requirements from UpStage version 3, and expectations for the new UpStage.

2. Current State of UpStage Development

2.1 Upstage Community

2.1.1 User Diversity

Our clients have found that UpStage is used by community arts groups, art makers of all disciplines, students from primary to tertiary, researchers interested in networked performance, online art, digital theatre, and teachers along with their students. The UpStage community also has diversity in term of age and includes all from young children to seniors.

2.1.2 Use of UpStage

The community primarily uses UpStage as a medium for constructing and sharing stories in real time. These are referred to as “cyber performances”, they are a networked performance accessible via a browser and in some instances with a physical venue as well, connecting actors and audience through live performance. UpStage is set up to be somewhat similar to traditional theatre. There are performers who use the stage, and audience members who only view. UpStage is also used for less theatre-like functions such as meetings, workshops and training as well as improvised jam sessions between musicians. It has also been used for online parties.

2.1.3 User Needs

Most users have desktop or laptop computers. However, there are a growing number of students in schools that are encouraged and even forced to use tablets or iPads, as well as consumers making the switch to the attractive and powerful portable devices. There have been problems where users have tried accessing the stage from a portable device even though the software does not support portable devices, as they were unaware it was unsupported.

2.1.4 UpStage Community Size

Currently there are 389 users on the contact list who are subscribed to latest news and event notifications. According to our clients there are about 20 regular and active users, but when events are held there are far more people accessing the software. There are 32 people on the developer list, however most are not active.

2.1.5 Developers in the Community

There are other developers aside from the AUT development team who contribute to the UpStage program. There is also a team currently at AUT devoted to creating a refreshed website for UpStage. Notable developers in the community are (with last names omitted): Martin, Paul (infrastructure rather than development), Gloria, and Douglas who created it but has not been active in years. There have also been a few people who claimed they have worked on the code yet have never actually submitted it to the project. There are also past AUT UpStage Team students who sometimes carry on contributing to the project.

2.2 UpStage Version 3

2.2.1 Current State of the Source Code

The source code for the project has continuously grown over the last 10 years and has been worked on by many different developers. The source code of UpStage has now become a large overly complex. With the current state of the source code, it is difficult and takes significant amount of time for any developers to reach a level of understanding of the code to contribute to the project. The source code is currently written in mixture of languages which are, Python, Javascript and Actionscript as well as some HTML.

Some of the original code does not follow standard naming conventions. We heard from previous members encountering and having to deal with (i.e. safely remove) entities such as “Pumpkin”, “Reap Zombies” and there are still multiple classes called “Thing” instead of descriptive and more comprehensive names. The state of the legacy code makes the task of extending the application’s functionality daunting to new team members. It also makes fixing or finding bugs a frustrating task for a developer.

2.2.2 Accessibility and Portability

An important factor driving change is accessibility and portability. The definition of portability has changed since the first version of UpStage was released 10 years ago. By today’s standards to be accessible and portable the software needs to be available for mobile devices such as Android and iOS as well as sustaining the accessibility from desktops. To achieve this, utilisation of obsolete technology such as Flash which is not supported on mobile devices, would need to be replaced by modern technology. The other option for making UpStage available for mobile devices is developing applications for each device which would run on them natively. Although downloading apps on a mobile device has become the norm, the clients in charge of the UpStage project are adamant that users should not need to download anything before they can access and use UpStage.

2.2.3 Dependency on Obsolete Technology

Flash is becoming an obsolete technology as more browsers and operating system distributers are announcing its intention to stop supporting in the future version. UpStage heavily relies on Flash technology to steam videos, playing sounds, and handling user interaction. First part of researching technology to replace Flash, will be to analyse the current source code thoroughly to understand what we are trying to replace.

2.3 Software Development

2.3.1 Development Tools

The team is currently using Sublime Text Editor, a somewhat basic text editor, to make changes to the source code as the structure of the code (i.e. multiple languages and client server model) does not easily allow for use of an intelligent Integrated Development Environment (IDE). Using an IDE increases productivity by supplying the developer with helpful shortcuts for formatting code, and organising imports and dependencies.

2.3.2 Testing

As well as this, there is the fact that there have never been any unit tests written for this program in its lifetime. This means there is no automated testing functionality at all, and unit testing, regression testing, and debugging is all carried out manually by having to just run and use the application. The lack of automated testing means more time spent by the team testing the program as appose to having a facility in place to run a test suite before every update. Absence of automated testing contributes to the fact that the software is hard to test which allows more bugs to remain hidden in the program. We have concluded that the time required to write unit test for UpStage version 3 may not be worthwhile as there will be less updates, meaning less time spent on manual regression tests. On the other hand, investing time on testing suit for new UpStage during the process of re-designing is much viable and does not have as much overhead.

2.3.3 Development Process

As Martin, the creator of Upstage, mentioned on the UpStage developer log, developing on Upstage is similar to reverse engineering where in terms of iterative process, the development can be broken down to the following:

1. Code analysis and debugging
2. Modifying the code base
3. Testing the modification

With the current development process, to add new functionality or to debug, most of the time is spent analysing the code to search where to modify the code. Moreover, having no automated testing as mentioned in *section 2.3.2 Testing* creates a bottleneck for productivity. Ideally, when adding new functionality to an existing software, less time should be spent on code analysis and testing. To reduce the overhead, the new UpStage should incorporate well know design-pattern or practices such as de-coupling.

2.3.4 Resource

There are no funds for creating the new software as far as we know. All human resources are supplied by the open source community, subscribers of UpStage IRC channel and AUT students/staff. Martin Eisenbarth, a developer who previously helped AUT students with development work, will likely not be able to aid the project with the new UpStage development.

3. Goals for New Project

The goal for the new project is to provide software that offers the same features and functionality as the current version, but with the underlying code being of a much higher quality.

The code needs to:

- Be maintainable and easy for new developers to begin working on the project and to add new features to the existing code.
- Strictly follow coding standards and practices (documenting code and naming conventions).
- Be written modularly (not just one massive code file with one massive method etc.).
- Be loosely coupled (so it’s easy to extend functionality and to change single parts of the code without affecting other parts).
- Be well documented, with useful comments among code.
- Follow carefully selected design pattern(s), architecture, languages and tools.
- Be made with modern and well supported technologies that aren’t in danger of becoming obsolete anytime soon (selected to the best ability of our current foresight).
- Utilise an automated testing suite (e.g. use a suitable IDE for the technology and tools chosen for us to work with).

This will all make it much easier to add and extend functionality to the program, identify and remove bugs, and easy for new students to step in and add value to.

4. Requirements

This section provides a summary of requirements for the new UpStage from our client. On top of these requirements, it is understood that the new upstage should inherit all functionality of UpStage version 3. Full requirements from UpStage version 3 is attached in *Appendix*.

4.1 General Requirements

4.1.1 Existing features of UpStage

●*cross-platform & cross-browser (anyone can access)*
●*no log-in or download required for audience (link to access a stage)*
●*no download required for players (server hosted)*

4.1.2 New features wish-list

●*access from any internet connected device (truly cross platform & browser)*
●*API modular & customisable - e.g. can choose to have chat window or not, can create different "skins"*
●*customisable bandwidth options - from as low bandwidth as possible through to high if you don't care about audience access (eg working with universities or high bandwidth institutions)*

4.2 On-Stage Requirements

4.2.1 Existing features

●*avatars that can speak & move, animated or still, shared by all players*
●*text2speech audio for avatar voices*
●*ability to play pre-recorded audio*
●*live audio-visual web cam streams*
●*drawing in layers - able to manipulate and move*
●*chat window for players & audience*
●*player/audience counter visible to all*
●*applause button for audience*
●*ability to "whisper" to other players*

4.2.2 New features wish-list

●*add media on the fly from anywhere online (e.g. YouTube, Flickr, etc)*
●*give audience permissions (eg guest avatar cameo appearance)*
●*improved real-time communication between players during a show*
●*- ability to open a feature that allows 'back stage' conversation for prompts etc*
●*pixel-movement of avatars with arrows (for really precise placement & movement)*
●*improved webcam operation*
●*- adjust size on stage*
●*- add tools to manipulate on stage (webcam specific palette)*
●*data input sensors (things like gps location, arduino inputs)*
●*audio consistency*
●*- when people enter a show after it's started, they hear pre-recorded audio at whatever point it's at for everyone else (currently they just don't hear the audio if they enter after it's been started, it doesn't play at all)*
●*recording: way to record a show & play it back as it was (e.g. record & replay the log, rather*

than make a screen recording)

- *NB: this will have to have a way for the webcam avs to be recorded if used other aspects can be pulled from the log*
- *– needs to be a feature turned on so as not to clog the server with recordings*
- *- needs to also record at a minimum file size possible without losing too much quality (perhaps settings are possible*
- *- allow file to be saved elsewhere*
- *on-stage tool to adjust chat window font size*
- *- players able to set for everyone (globally)*
- *- audience able to adjust just for themselves*
- *on-stage sign or text message:*
- *- ability to leave a text message on the stage for random visitors*
- *audience tool palette*
- *- add to the options*
- *- NB: we need to investigate more what & how we want to offer them*
- *more & better avatar voices*
- *- a way for players to add their own voices*
- *- explore other txt2speech options to integrate*

4.3 Wardrobe Requirements

4.3.1 Existing features

- *multiple players collaborate in real-time performances for online audiences*
- *players create own stage, upload own media*
- *media can be shared across multiple stages simultaneously*
- *information about players (email address, first & last log-in)*
- *browse and view media*

4.3.2 New features wish-list

- *a way to schedule & advertise shows (push to calendar on website)*
- *way to communicate asynchronously with & between players e.g. via email or messages in a player's profile*
- *ability to set permissions for each media item - e.g. free for anyone to use, use only with permission, not to be used by anyone else (as well as cc options)*

5. Research Areas for New Project

The most immediate stages planned for the development of new UpStage involves investigating the technology used in the current software, and researching potential solutions or replacements of suitable technologies.

Once we reach an understanding of how the current features are utilized we will be able to create a basis to compare possible solutions, in attempts to reduce any faults in the current software. We have already started researching three areas of the potential technology solutions. These areas are Flash, Text To Speech and Video/Webcam streaming, which have been split between the team.

Each pair will research their assigned area and provide a proof of concept on the most viable solution. This will ensure the new options are functional and can meet our clients' requirements. Other areas of new UpStage that are planned for future research are as follows.

- Potential Technology Solutions
- Programming languages
- Design patterns
- Open source community
- IDEs
- Automated testing suites
- Versioning control software
- Research methodology (followed by development methodology at a later stage)

6. Proposed Plan for Next Phase

The team will follow the plan as below to investigate possible solutions for the new project as well as familiarising new members with the existing code in parallel.

Week 6	#225	Replacement for Flash Plugin	Define the feature of Flash plugin
	#207	Text2speech technologies	Find available technology
	#200	Video streaming technology	Analyse the role and functions of VideoHack in UpStage
Week 7	#225	Replacement for Flash Plugin	Start looking for technologies to replace Flash
	#207	Text2speech technologies	Find available technology
	#200	Video streaming technology	Produce a report on feasible video stream technology
Break	#207	Replacement for Flash Plugin	Start looking for technologies to replace Flash
	#199	Text2speech technologies	Find available technology
	#198	Video streaming technology	Produce a report on feasible video stream technology
Break	#199	Replacement for Flash Plugin	Start looking for technologies to replace Flash
	#198	Text2speech technologies	Find available technology
	#87	Video streaming technology	Produce a report on feasible video stream technology
Week 8	#198	Replacement for Flash Plugin	Start looking for technologies to replace Flash
	#195	Text2speech technologies	Find available technology
	#231,219,216	Video streaming technology	Produce a report on feasible video stream technology
Week 9	#231,219,216	Replacement for Flash Plugin	Start looking for technologies to replace Flash
	#151	Text2speech technologies	Choosing suitable technology
	#195	Video streaming technology	Select a feasible video stream technology
Week 10	#231,219,216	Replacement for Flash Plugin	Start looking for technologies to replace Flash
	#151	Text2speech technologies	Choosing suitable technology
	#210	Video streaming technology	Select a feasible video stream technology
Week 11		Replacement for Flash Plugin	Create proof of concept for a replacement of flash
	#195	Text2speech technologies	Create proof of concept for text to speech without flash
		Video streaming tecnology	Create proof of concept for video streaming technology
Week 12		Replacement for Flash Plugin	Finish proof of concept for a replacement of flash
		Text2speech technologies	Finish proof of concept for text to speech without flash
		Video streaming technology	Finish proof of concept for video streaming technology replacement

Week 13	Regression Test	Poster, Portfolio, Reflective Reports Team Development Manual, Project Plan, Proofs of Concepts for Clients Updated Acceptance Test Sheet

7. Appendix

Functionality of UpStage Version 3

Outside System User

- Can register to be a Player
- Sign in with the registered account

Audience - same as Outside System User plus:

- Can access a link to find user's local time in the homepage
- Can enter a stage
- Can view a performance
- Can send text messages in the stage

Player - same as Audience plus:

- Can log out
- Can be assigned to a stage or stages
- 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

Maker - same as Player plus:

- 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 - limited to 10MB
  - 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

Unlimited Maker - same as Maker plus:

- Has no limit for file size when uploading media

Admin - same as Maker plus:

- 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 details of all existing player accounts
  - Update the changes
  - Delete the player's account
  - Can modify a user account's password
  - Can modify a user account's email
  - Can modify a user account type
- 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

Creator - same as Admin plus:

- Can create other Creator user accounts

8. Version History

Date	Version	Description	Modified By
20/05/2014	1	Created main structure of document. Added known status of Upstage from proposal document (2014_Sem1)	Takuma
22/05/2014	2	Added headings for important sections and bullet points which must be elaborated	Takuma
23/05/2014	2.1	Added more heading such as technology and language used. Bullet points should be elaborated	Takuma
30/5/2014	2.2	Added existing functionality/mechanism	Xiangyu
31/05/14	3	Combined with James' work on the rationale and motivation, and proof read.	James
29/07/14	4	Edited document following Anne's feedback	James
12/08/14	4.1	Added introduction Added sub-heading for "Current code" section	Takuma
		Modified the "list of the main limitations of current UpStage"	Xiangyu
26/08/14	5	Modified and added to User Base section using data from communications with clients	James
		Proof read and added the most updated goingoing plan	Takuma
26/08/14	5.2	Added Design to Redesign UpStage section	James
26/08/14	5.3	Corrected user account roles	James
20/10/14	5.4	Fixed the presentation (font, header orders, layouts, content page)	Takuma

		Added details to introduction Added new section “Dependecy on obsolete technology” Categorised the list of functionality Updated to the newest plan	
20/10/14	5.5	Modified introduction grammar. Added initial steps for new research areas.	Charlotte