Product vision

The First Order

Chris Berg 4216776
Martin Koster 4371011
Hung Nguyen 4232410
Christian aan de Wiel 4396286
Ruben Wiersma 4214250

28 apr 2016

Contents

1 Introduction

The Multimedia context project 2016 is an assignment for the company PolyCast. This company records classical concerts for orchestras (PolyCast, 2016a). There are many aspects to such a big event and every aspect has its struggles and problems. First, there is the question of what to record at a given moment. To answer this question, a script is written in advance. A problem that shows up here this is that you cannot use the same camera for two consecutive cues and that sometimes, it's hard to visualize the concert before it actually happens. Another aspect is to control the cameras during the performance. A problem in this aspect is, for example, a view from a camera which is different than expected. Finally, you have the editing after the performance to fine tune the audio and video. As you can see, there are a lot of places where things could go wrong or where work is done that can be done more efficiently, like a camera with a wrong view. Automation can help take away human error and do some repetitive or 'boring' work so that the people at Polycast can focus on creative tasks. Software can help visualize different scenarios for the polycast team and can put those into effect at the actual concert.

At this moment the company uses software to create scripts (filemaker), control the cameras, and edit their recordings, but none of these are incorporated in an overarching workflow where all their programs work together. The software to control the cameras does a decent job at controlling the cameras, but someone has to find the right cue at the right time by listening to the scorereader and the script is made in software that is not specifically made for writing scripts. We will propose a product that can serve as a thread throughout the entire workflow, starting with script creation.

The next section will explain what the vision is of our project and which problems we would like to solve. In section three, we explain how we want to accomplish this during our sprints. To do this optimally we have to set priorities which are stated in section four. Finally to check whether we have done a feature we must define what we define by done, this is done in section five.

2 The vision of our project

2.1 The target market

There are a couple of groups within the Polycast team who will use the application. There are people who are experts in the audio and visual aspects. These people have experience with computers and with editing applications (Polycast, 2016b). The directors have a lot of knowledge about the piece which is going to be played. They are less likely to have a lot of experience with computers but know about editing software like Final Cut Pro (personal notice, April 20th, 2016). The director will prepare a script of the concert beforehand. This script dictates a for a specific camera what instruments need to be covered with a given action. These two groups have to be taken into account when designing the application.

Because we only have seven weeks to develop an application, we will focus on one of these groups while creating the software with all the other groups in mind. Practically, this means that we will create our software in such a way that it can be easily extended with different views for different people. The user we will focus on for this project is the director. The director crafts a vision for his project and has to communicate this vision to his team (AHK, 2016). He is also involved with editing the concert after recording has wrapped.

2.2 Customer needs

Our first version of the product will address the director's needs in crafting a vision and communicating this to his team. Because the director has to visualize the whole concert beforehand in a scripted concert, our software will provide in the need to have a clear overview of the concert and the cameras and instruments that can be used.

2.3 Product attributes necessary for customer needs

In order to address these needs, we will need to make sure our software has a couple of attributes. The software has to be pleasant to use, reliable, easily understandable and has to meet the functionality requirements of the user (Gram and Cockton, 1996). This is captured in the word user-friendliness and intuitiveness. The functionality requirements of the user (personal notice, April 20th, 2016) are that they can visualize the concert beforehand, create a script in the software and then output

it to an A4 document where all actions are displayed in ordered fashion. These priorities will be listed in a MoSCoW list where requirements are ordered in Must, Should, Could and Won't sections (Clegg and Barker, 2004).

2.4 Existing products

In order to create intuitive software, but also to meet a need that has not been met yet, we have taken a look at software that is already out there. We can distinguish three types: scriptwriting software, scriptwriting software for the entire filmmaking workflow and editing software. Examples of scriptwriting software are Final Draft (Final Draft, 2016) and the open-source software Trelby (Trelby, 2016). These packages focus mainly on script creation and are not involved with the full workflow of production. These programs are also not made for script creation of concerts. Celtx is software that is created for the full workflow of video production (Celtx, 2016). They have software where everyone works from one document, the same kind of work we envision our software doing, but their software is also not created for musical productions. Finally, we have software suites like Premiere Pro (Adobe, 2016) and Final Cut Pro X (Apple, 2016). These programs are made to edit footage together and have no scripting functionality. We can use these programs to distill user interface design patterns that directors are familiar with. This can be done with all the software that was mentioned before.

2.5 Our vision on the project

To summarize our vision on the project: We are going to develop an application that offers support to the process of scheduling and scripting a recording of a classical concert. Creating the schedule is facilitated by a visualization of the concert hall with the positions of the cameras in it. The application will present the created schedule in different formats, fulfilling the needs of the people involved in the process of recording.

2.6 Our unique selling point

We are going to stand out from other planning tools by providing a visualization of the concert hall with the positions of the cameras in it. This visualization can, among others, be used to check whether a camera is available for filming during a certain queue. this interface provides an intuitive workflow that is a natural extension of the analogous way of adding cues to a score by hand.

2.7 Target timeframe and budget

The project has to be delivered on June 23rd (Hanjalic and Bacchelli, 2016). This means we have eight sprints of one week each where we have about 20 hours to work on the project. There is no financial budget available.

3 References

- Adobe (2016). Video editing software. Retrieval at 4 May, 2016 from http://www.adobe.com/products/premiere.html
- AHK (2016). Regie documentaire. Retrieval at 4 May, 2016 from http://www.ahk.nl/filmacademie/opleidingen/regie-documentaire/
- Apple (2016). Final Cut Pro X. Retrieved at 4 May, 2016 from http://www.apple.com/final-cut-pro/
- Hanjalic, A.; Bacchelli, A. (2016). Context Project Guidelines. Retrieved at 4 May 2016 from https://docs.google.com/document/d/1Xb4vrF9V78ge_vq4VLrl3YZ2Qq0daCLW2oid2IrGRfA
- Celtx (2016). Celtx - Free Scriptwriting & All-In-One Production Studios. Retrieved at 4 May, 2016 from https://www.celtx.com/index.html
- Clegg, D.; Barker, R. (2004). Case Method Fast-Track: A RAD Approach. Boston, MA: Addison-Wesley. ISBN 978-0-201-62432-8.
- Final Draft (2016). Official Final Draft — #1-Selling Screenwriting Software —Final Draft. Retrieval at 4 may, 2016 from http://www.finaldraft.com
- Gram, C.; Cockton, G. (1996). Design Principles for Interactive Software. London, UK: Chapman & Hall.
- PolyCast (2016a). PolyCast. Retrieval at 2 May, 2016 from http://www.polycast.nl/
- PolyCast (2016b). PolyCast Our Team. Retrieval at 4 May, 2016 from http://www.polycast.nl/medewerkers
- Trelby (2016). Trelby. Retrieval at 4 May, 2016 from http://www.trelby.org