Product Vision

Team: goto fail;

Alex Geenen
Martijn Janssen
Mark van de Ruit
Bart de Jonge
Menno Oudshoorn

Table of Contents

Introduction 3

Who is the target customer? 4

What is the current situation for the customer 5

Which customer needs will the product address? 6

Director/Script writer 6

Camera operator 6

Score caller 6

Which product attributes are crucial to satisfy the selected needs? 7

Script creator 7

Tools for the camera operators 7

Tools for the score caller 8

Tool for the director 8

How does the product compare against existing products? 9

What is the target timeframe and budget to develop and launch the product? 10

References 11

Introduction

Our project focusses on creating software for companies that make music video productions in large teams. We will be focussing on three major parts: A script creator, automatic camera preset recalling and a clear overview of the timeline for all people involved during the recording.

In this document, we will give our product vision. This will be done through answering five fundamental questions:

1. Who is going to buy the product?
2. Which customer needs will the product address?
3. Which product attributes are crucial to satisfy the selected needs, and therefore to the success of the product?
4. How does the product compare against existing products?
5. What is the target timeframe and budget to develop and launch the product?

In the following sections, we will answer these questions one by one, and give you a clear overview of what we think our product is going to be, and why it will be great to use.

Who is the target customer?

The target customer is PolyCast [1] and companies that make music video productions in large teams in general. These customers have in common that a large amount of people are working on the same project: directors, camera operators, music announcers, etc. The communication between all these people is a very difficult process that in most cases still happens on paper. Another problem these companies have is that a camera operator can at most control about 3 cameras since he must manually recall a preset/ bring the camera in position and adjust the frame. All of this can be done automatically, and this is where our product wants to shine.

What is the current situation for the customer

During a recording, there is a lot going on. There are multiple cameras used which are operated by several camera operators. Before this, a script is created. This script is planed out by the director before the recording, and the camera operators will follow this script during the recording. However, nothing always goes fully as expected, so during the recording, the director can also make manual changes by telling the camera operators what to do. This can be a pretty stressful procedure.

The company is using new cameras that can be controlled through IP commands, which opens up a lot of possibilities to improve and automate the workflow at PolyCast, and possibly other similar companies.

The main problem at PolyCast at this moment is that the workflow still is a very manual procedure. It involves paper scripts, remote operation of cameras with joysticks, and, as already said before, a good dosis of chaos.

The script creation process is still very manual. Everything is still put in one timeline, and there is no support for collision detection or separate camera timelines. Collision detection has to be done manually be the director during or after creating the script, and this can take up quite some time. Also, the director might miss something which can create trouble during live recording

The people at PolyCast have recently started using a simple app that stores camera presets. However, this still has to be done manually, which means a camera operator can only operate a limited amount of cameras, and has to focus a lot on what is coming next rather than focus on the present moment and try to make the most beautiful shots. An app that automatically recalls presets would be very helpful here.

Finally, during recordings there is a score caller who calls out at which point we are and what shots are coming up. This can become quite chaotic, especially if the director is shouting in between as well. It would be much better if this can become digital. Then, highlighting the shots for certain camera operators is also possible, which makes it much clearer for a camera operator what they have to focus on.

Which customer needs will the product address?

There are different people that are going to use our product, and they all need different features. We'll discuss some of the people that will use the product and what they expect the product to have.

Director/Script writer

Before a recording takes place, a lot of work is already done by the director. He thinks about what kind of shots he wants to take, and how he wants to picture the concert, for example. To put this on paper, he creates a script. At the moment, this still happens completely manual, and it takes a lot of time. Therefore, what the director wants at this point is something that speeds up the process of script creation. What could also help the director to create better scripts is some visualization, for example of the layout of the stage, or highlighted parts in the script if there is some kind of camera conflict.

During the recording the director must be able to view the script and all its property in a simple and elegant way and he might even want to make some live changes to the script. Because the director has many things to do during a recording this must be presented in an intuitive interface that doesn't take too much attention to gather data from.

Camera operator

What the camera operator mostly needs is time. Currently, a camera operator has to do a lot of things at the same time. He has to listen to the score caller and/or the director to keep track of what's happening and what's coming next. Meanwhile, he has to manually adjust the current camera, and has to set ready the next camera. The cameras are operated through IP commands. [2] Presets are already used, but they have to be manually loaded, and this is where the camera operator wants something of our product. An automatically scrolling timeline with the score, and the shots that go with it, could greatly enhance the precision of the camera operator, because he will not have to listen and act at the same time anymore. He might not always hear or understand the caller directly, so having a user-friendly layout with a rolling timeline is something the camera operator would want. Something else that he desires is automatic preset recalling. If our product can do the preset recalling for him, the camera operator doesn't have to spend any time on this, and can put his full focus on what is happening at the moment, creating some beautiful shots.

Score caller

During the recording, the score caller calls out where we are at the moment, and what shots are coming up. His needs align with that of the camera operator. He could also use the automatically scrolling timeline. This gives him a good overview of the present situation. He can then still call out the scores and/or shots to provide a backup for the timeline, to make sure that there are no misunderstandings with the camera operators.

Which product attributes are crucial to satisfy the selected needs?

There are a lot of attributes that are of utmost importance in order to satisfy the needs of our customers. These attributes will not all be packed into one program, which is why we'll discuss them in their relevant groups.

Script creator

The keyword for the script creator is user-friendliness. The director must be able to easily add, delete, and move camera shots, and be informed of possible problems directly. This gives us the following list of features that we think must be in our script creator:

- An easy and fast interface to add a camera block. This should contain enough information to be useful to the director, and later, the camera operators, but should contain little enough information so that the director does not have to fill out a lot of fields to create a camera block. Optional information fields are very important here.
- Visual aid to detect problems in the current timeline. Think of a red camera block when it is used too soon after another camera block for the same camera.
- Multiple timelines, one for each camera. If there are multiple timelines, it is much easier for the director to see how much time he has between cameras, if there are cameras that are never used, etc.
- Automatic merging of the camera timelines to one general timeline. In order to make the separate timelines work in practice, our program must be able to merge the camera timelines into one general timeline that can be used for the actual recording.
- Easy storing and loading of projects. This means being able to easily select a file for importing and exporting the project. Also this importing or exporting must be fast and the exported project can not be too large.
- Easy editing of the timelines. This means moving around camera blocks by drag and drop, assistance in aligning two camera shots of different camera's (that is, make sure one camera starts at exactly the same time the other camera stops). Also, editing descriptions or titles by just clicking on them.
- The ability to create constraints for a final composition of camera shots, think about what cameras to use and how long each shot should be, and let the separate camera blocks/timelines be generated automatically.

Tools for the camera operators

These contain a UI for the automatically scrolling timeline, and the automatic loading of presets. Some important features are:

- A clear overview of the timeline, with all the useful information shown without the camera operator having to click on anything.
- Shots of cameras that are operated by this camera operator are highlighted
- Automatic preset recall which is done as soon as possible. That is, as soon as the camera that should be set ready is not live anymore. This gives the camera operator the as much time as possible to prepare for what's coming and adjust the frame if necessary.
- Some kind of queue with previews of the camera's which have been set ready. In this way, the camera operator gets visual input of what he is about to record, and can already think about possible adjustments.
- A manual override button, which stops automatic presetting and lets the camera operator take full control. The camera operator should also be able to revert to guided mode.

Tools for the score caller

The score caller should keep the overview at all times. Therefore, the UI is the most important part here. Some features are:

- A clear overview of the timeline, with all the useful information shown.
- Easy interface for switching between two modes: tick mode and automatic mode. In tick mode the score caller presses a button for each count to advance in the score. In automatic mode the count will be called automatically via a predefined length of each count in seconds.
- Possibility to take over manual control. At this point, the camera operators will see on their view that the score caller has taken over manual control, which means they will have to listen to what he says, instead of relying on the predefined setup. This can also be reverted to automatic mode

Tool for the director

- A clear overview of the timeline, with all the useful information shown.
- Possibility to edit data in the timeline on the fly, for example the description of the shot. Warnings should be given if this happens too close to the actual shot.

How does the product compare against existing products?

Seeing as our main focus lies on improving the current workflow at PolyCast, our final product should have some unique features that the systems that are currently used do not have. In this section, we will highlight some of these features and see how they compare to the current situation

The first major change lies in the automatic recalling of presets. The software that is currently used completely relies on manual control. This means that the camera operator has to select the correct preset and load it manually. Our software will do this automatically, which gives the camera operator much more time to focus on other things, or to operator more cameras.

Another point on which our product will improve the current situation is the creation of scripts. Especially the fact that you can have multiple timelines, and that collisions are automatically detected, make the director's life much easier. The overall overview will be much clearer, and editing the timeline will be much smoother than before. Also, separate camera timelines can be automatically generated from a set of constraints about cameras and shot length that the director can input. This gives the director a better environment to create a perfect timeline that shows everything the director wants the public to see.

Finally, the current workflow during the actual recording will also be improved by our product. This is due to the clear overview of the timeline that the camera operators, the score caller and the director all see. Also, they all have their own features that they need. For example, the director can edit the timeline on the fly, and the score caller can change the speed of the timeline. These changes will then be propagated to every other team member, making sure that everyone is always up to date. This is a large improvement to the current situation, in which everything is just called out. This is prone to error, someone might misunderstand something or miss something at all. Our product will improve this.

An existing product that focuses on productivity automation for the broadcast industry is Just Macros [3]. Our product will be different in the sense that it will be much more integrated with script creation. Just Macros focuses more on automated control of camera equipment. We will focus less on actual control, and more on just presets. We do this because there is already solid equipment for moving cameras present.

What is the target timeframe and budget to develop and launch the product?

We are aiming to release the final product in 9 weeks. This gives us enough time to make sure that all the wanted features are implemented and tested thoroughly, but still gives us a release date as early as possible, so that the product can be taken into use by our target audience as fast as possible. This is important to us because we believe that our product will really make the lives of those people easier.

We are working in an agile development structure. This means that we aim to always have a working version. We will release a new working version every week, with improvements and additions with respect to the previous version.

Looking at budget, we don't think budget will really be an issue. We don't see any upcoming large purchases that we will need to make, and if there are any small expenses this will not be a problem.

References

[1] http://www.polycast.nl

[2] http://pro-av.panasonic.net/en/sales_o/broch_pdf/aw-he130.pdf

[3] https://secure.justmacros.tv/