

Multimedia Services : Product Vision

Group BeNine
TI2316 Context Project

Bryan van Wijk (bryanvanwijk, 4363329)
Dorian de Koning (tcmdekoning, 4348737)
Ege de Bruin (kedebruin, 4400240)
Jochem Lugtenburg (jlugtenburg, 4370805)
Naomi de Ridder (nderidder, 4383109)

May 4, 2016

Supervisor: Dr. Cynthia Liem
Software Aspect TA: Valentine Mairet
Context Aspect TA: Alessio Bazzica

Delft University of Technology
Faculty of EEMCS

Contents

1	Introduction	3
2	Target customer	4
3	Customer needs	5
3.1	Current work-flow	5
3.2	Proposed solution	5
4	Success of the product	7
5	Existing products	8
6	Timeframe and budget	9
	References	10
	Glossary	11

1 Introduction

This document contains the Product Vision of context project Team BeNine, it guides the team throughout the process, acting as a compass heading towards the final product. For this project, the team will work with a small, already existing idea. This idea, which has already been prototyped by the client, is a small preset-control app which is only capable of handling basic switching. We will further improve this 'notion' of a product from scratch, and optimize it, since it is inefficient and requires more man-power than it should need.

The client, PolyCast Productions B.V.¹ is a recording company, with it's primary focus on classical music productions. They use high quality recording devices, such as the Panasonic HE130-W which can be controlled using Internet Protocol commands. (*HE130-W/K HD Integrated Camera Manual*, n.d.).

In the upcoming sections the background of the project will be discussed. First, an overview of intended users is given followed by the addressed user needs and satisfied attributes. Second, the already existing products and unique selling points of the teams solution will be discussed. Finally, the time-frame and budget for development and launch of the final product will be discussed.

¹<http://www.http://www.polycast.nl>

2 Target customer

This section goes into detail about who the target customer is and who will eventually 'buy' the final product. While the main purpose of this project is educational, the client is free to use it for non-commercial purposes. If the client would want to use the final product for their production, this would have to be further discussed with the team.

The client, or in other words the target customer, that is addressed to above is PolyCast. PolyCast explicitly asked the help of team BeNine to optimize their recording work-flow because as it is now, the process of recording concerts is not quite optimal yet: it needs more man-power (and hours) to operate than it should need. The team will therefore optimize the recording work-flow for PolyCast to their needs. The needs of the customer will be further discussed in the next sections.

However, since the client works in the audio/video production branch, other possible target customers belong to the same branch. To be more precise, a target customer for the product is a recording company who wants to optimize their recording work-flow, using software managing their presets in a more efficient way. It is possible though that some slight adaptations have to be made if this product is used for other types of recordings than the classical music recordings of PolyCast. The reason for this is that the recording work-flow of another customer might be different and thus they might have different needs.

3 Customer needs

Since the team is working on a notion the client has had for a long time, clear ideas about the product to be created are available. The client shared these ideas during a seminar, which the team attended during the initial week of the project. From these ideas, BeNine selected optimizing the camera control work-flow.

3.1 Current work-flow

Currently, a script is written in a simple database or spreadsheet, instructing the camera operators on how to program their presets. Optimizing the camera control work-flow is done by eliminating the mundane task of pressing multiple buttons, into a single tap button on a touch interface. This, combined with another notion of making camera presets more intelligent (for instance by comparing them to a stored image of a certain composition, automatically adjusting the preset).

A possible scenario involves a recording session of a concert at the Concertgebouw. Before the concert, rehearsals take place. The client also takes part in these rehearsals by testing the script created by the director. Before rehearsing, the camera operator programs the required presets into the camera controller. This is a very intensive task, requiring a lot of buttons to be pressed. During the rehearsal, these presets have to be called, also requiring a lot of buttons to be pressed. After the rehearsal, the camera operator knows which presets to call. Since the scene changes because people move, he needs to adjust the presets while the concert is in progress before starting the recording. Due to the intensive task of calling a preset, the camera operator has only little time to adjust the presets.

While PolyCast had many other ideas to work on, the team is limited by time. This lead to the team selecting a subset of these ideas. Since the team is multidisciplinary in the sense of Multimedia, Embedded Systems and Data Science, the team will select ideas that span all disciplines.

3.2 Proposed solution

The team wants to implement several features, all linked to each other. The solution consists of two important parts. First the web interface on which the camera operator can select a camera to use, work with presets quickly (saving, loading, adjusting) and control the movements of the camera with a simple joystick. With a few sliders the camera operator can also change the the focus of the camera and zoom in or out. Every camera has a set of presets that can be selected by one click on them after selecting a camera. All of the presets are stored in a database which makes them persistent.

Second, besides quick selection of presets, the team proposes another solution to selecting presets. Together with a representative acting on behalf of PolyCast, the team came up with a solution involving 'tagging' the scene. A tag is considered to be a picture of the scene taken by a camera. By taking a picture, we collect the pan, tilt and zoom angles of the scene, so the camera can be adjusted easily on preset selection. The camera operator

collects these tags for each camera. In a later stage, even individual tags can be assigned keywords such as 'First violin' or 'Strings' to make quick selection of various scenes a possibility. If this later stage is implemented, a 'search-engine' for scenes can be created. The camera operator will then be able to search for keywords to find appropriate scenes.

Since the client cannot afford software failure during a recording, a stable, failure resistant program is required. To achieve this, Agile development is used as described by Black, van Veenendaal, and Graham (2012, p. 32). Also, the program must be persistent in the case of failure. Settings have to be stored in a database, so that in the unlikely event of a failure a quick reboot can be facilitated to reduce disruption of the work-flow to a minimum.

4 Success of the product

This section goes into detail about the attributes that are crucial to satisfy the user needs, and which also influence the success of the product. Words will be written about the requirement- and the development side of the product success.

The success of the product lies mostly in improving the work-flow, this is implemented in the form of intelligent (possibly computer vision driven) presets. An intelligent preset might for instance recall multiple camera positions. The client stated, that they would highly benefit from this.

Also influencing the success of the product is ease of use. With a simple web interface the production team can control more camera's per person than they could before. This is due to the fact that different buttons will enable the production team to do multiple tasks on the press of a single button.

Development poses the biggest threat to the product success. Bugs may occur, threatening the stability of the program. Since the client requires a program with a very low chance of failure, Test Driven Design (TDD) and Continuous Integration (CI) are used to improve both quality and productivity of the development process. Due to the fact that more features can be implemented saved by bug fixing, TDD increases productivity. Also the development process becomes more predictable, due to robust design and faster bug fixing as written by Kawadkar (2014). To support this, Miller (2008) published that CI improves productivity by reducing overhead and ensures that the quality of the code base remains equal to the quality of the product without using CI.

5 Existing products

In this section the already existing solutions to the problem will be discussed.

One of the existing software products managing camera presets is Just Macros². Just Macros is an expensive product, for commercial use. Just Macros does not have an intuitive tablet interface, requiring the software to be run on a computer.

Our unique selling point is the ability to have intelligent presets. While a solution to simple non-intelligent preset management is quite trivial, computer-vision related intelligent presets are not. For instance the SIFT Algorithm as described by Lowe (1999) may be used to detect a scene, which can then be adjusted to a similar scene. Due to the complexity, the team will use as less computer-vision as possible. Another important feature, persistence of the work-flow data from the server can be created using a database, making the data retrievable after program termination as pointed out by Elmasri and Navathe (2014, p. 17).

Of course, different companies and recording goals exists. The product requested by the client is adjusted to match their particular work-flow and does not exist in it's required form. PolyCast is already able to produce stunning video and audio recordings, but does not yet have the perfect work-flow-improving software just yet.

²<https://secure.justmacros.tv>

6 Timeframe and budget

In this section, the target time frame and budget to develop and launch the product will be addressed to. Almost every project is limited by time and budget, however this project is only limited by time due to its educational purpose.

This project has no possibility to extend the time frame, making the team of developers extremely restricted by its deadlines. From start to end, ten weeks can be used to optimize and finish the product. After these weeks, the product will be released and probably used by the client, PolyCast.

Due to the educational purpose of the project, namely a university course, no budget has been given. This does not pose a problem because the team is given all the expertise of the PolyCast employees and will be granted the ability to test the product using the camera's already available at the clients office using an internet connection. Protocols and other tools are also documented, providing the team with a large source of information. Since this is all available, it is highly unlikely that any additional costs, except travelling expenses will be made. The teams expertise, combined with the client and TU Delft expertise is assumed to be sufficient for the delivery of a great product.

References

- Black, R., van Veenendaal, E., & Graham, D. (2012). *Foundations of software testing*. Cheriton House, North Way, Andover, Hampshire, United Kingdom: Cengage Learning EMEA.
- Elmasri, R., & Navathe, S. (2014). *Fundamentals of database systems* (6th ed.). Edinburg Gate, Harlow, United Kingdom: Pearson Education Limited.
- He130-w/k hd integrated camera manual*. (n.d.). Panasonic Corporation. Retrieved 25-04-2016, from http://pro-av.panasonic.net/en/sales_o/broch_pdf/aw-he130.pdf
- Kawadkar, H. (2014). Test driven design. In *Contemporary computing and informatics (ic3i), 2014 international conference on* (p. 236 - 237). Mysore.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. the proceedings of the seventh ieee international conference on* (Vol. 2, p. 1150 - 1157). Kerkyra.
- Miller, A. (2008). A hundred days of continuous integration. In *Agile, 2008. agile '08. conference* (p. 289 - 293). Toronto, CA.

Glossary

Agile development set of principles for software development, which provides the opportunity to assess the direction of a project throughout the development lifecycle. 6

Continuous Integration a development practice that requires the developers to integrate code into a shared repository several times a day. 7

Internet Protocol protocol which enables computers to communicate with each other over the Internet. 3

SIFT Algorithm scale-invariant feature transform: an algorithm that detects and describes local features in images. Also known as an 'image descriptor'. 8

Test Driven Design a method of software development in which testing is done before the code is written. 7