Project Management Information

Smart Door

CSIR
ZEBRA-V
Eduan Bekker (12214834)
Albert Volschenk (12054519)

Zühnja Riekert(12040593)

Publication Date: 20/10/2014

Change Log

17/10/2014	Version 0.1	Document Created	Albert Volschenk
17/10/2014	Version 0.2	Document Layout	Albert Volschenk
18/10/2014	Version 0.3	Software development process	Albert Volschenk
18/10/2014	Version 0.4	Team profile	Albert Volschenk
19/10/2014	Version 0.4.1	Edited team profile	Zühnja Riekert
19/10/2014	Version 0.4.2	Edited team profile	Eduan Bekker
19/10/2014	Version 0.5	Problem management, project progress	Albert Volschenk
19/10/2014	Version 0.6	Edited document	Eduan Bekker
19/10/2014	Version 0.6.1	Edited document	Zühnja Riekert

Contents

The Software Development Process

The Scrum methodology was followed during the software development process. There are several reasons why the scrum methodology was used, the main reasons are:

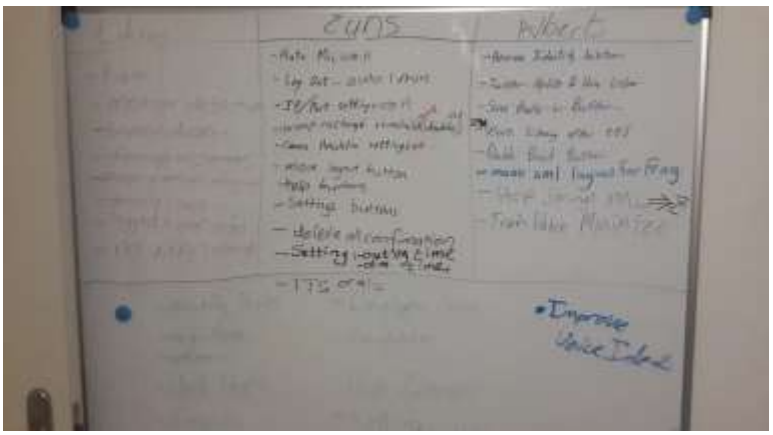
- It encourages teamwork.
- Problems can be solved in a team environment.
- It promotes short feedback cycles and continuous delivery.
- It provides good project tracking.

Daily scrum meetings were held to discuss work that was previously done, work that has to be done now and potential problems that may occur in the near future.

During the monthly sprint meetings, the discussions were focused on what stories needed to be completed during the next sprint. The spike periods, where each team member needs to research a certain topic or technology, was also set during the sprint meetings.

During the software development process, an informal scrum methodology was used.

- At each sprint meeting all of the stories that needed to be done was written on a large white board (underneath the team member whose responsibility it was to complete the story).
- Then followed a discussion on the priority of each story.
- Each team member would then choose a story and put a blue magnet next to it. This magnet system indicated each team member's current story.
- When a team member finished a story, he/she would then indicate (with an arrow) which team member should test the story. Then the member would move his/her blue magnet to a new story and start with that new story.
- After the testing of the completed story is done and the tester is satisfied with the completed story, the story would be striked out with a green line. This indicated to the rest of the team that the story is completed. If the tester found some bugs he would notify the story's creator, which will then later fix the bug. The process would repeat itself until the tester is satisfied with the story.
- At the next sprint meeting, the finished stories would be wiped out to make room for the next sprint's stories.



Team Profile

Eduan Bekker

- Team leader
- Face Researcher
    - Face, eyes and nose detection
    - Facial recognition
    - Group rectangles algorithm
- Door and door's circuit.
- Raspberry Pi Server
- Documentation

Zühnja Riekert

- Database
- User Interface
    - Custom theme and layouts
- Text to speech
- Usability optimization
- Internal system integration
- System settings
- Documentation

Albert Volschenk

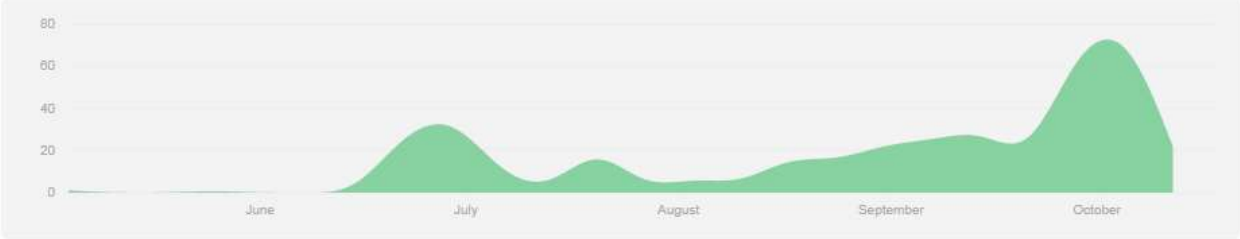
- Twitter
- Voice Researcher
    - Voice recognition
    - Voice identification
- User commands
- Github "expert"
- Client Communicator
- Raspberry Pi Server
- Door and door's circuit.
- Documentation

Problem Management

A Simple honesty system was used for solving problems. As soon as someone ran into a problem, that person would notify the group. The person would then be given time to do research about the problem for a set amount of time. If the research period expires and the user still can't fix the problem then the group will discuss it during one of the daily scrum meetings, where alternatives would be considered. Once the group settled on a decent alternative or a solution could not be found, the client would be informed and asked for their opinion.

Project Progress

The main progress started around June/July with a constant flow of commits. Then at the start of August the commit rate dropped significantly because of the need to do much more research about problems or new unfamiliar technologies. Halfway through September some of the research started to pay of and the commit rate started to climb again. From there on it was a race against time to get everything into a working system. All of the rise and then sudden drop in commit rate during the year was due to problems that occurred, then after a sufficient amount of research the problem would be fixed and the commit rate would rise again.



Unimplemented Functionality

- Due to a compatibility bug with GCC 4.8 and Android version prior to 4.3 the application does force close on said devices. Direct contact with the project owner of JavaCV has been made and a fix is in the works. As of the publication date of this document, it has not yet been resolved.
- It was not possible to implement the original sought after voice Authentication system. But after discussing the problem with the clients, it was decided that voice identification would be an acceptable alternative.
- Due to time constraints some functionality was omitted. The voice command system only works on the logged in page and the application has no “bonus features” such as integration with the currently existing cmore system, a booking system or a “Where is” feature.
- No logging system has been implemented.
- Some of the usability problems identified from usability tests could not be improved due to time constraints for instance there is no visual feedback that training on faces is occurring.
- Needs more furrow unit testing.

Main Risks and Challenges

Since there were some very complex and relatively undocumented parts of the project, the main risk factor was to get a decently working system with the limited time available. From the large list of challenges that the group faced, a couple of them stood out.

The fact that almost every technology was new to the members of our group meant that it was quite the challenge to learn the new systems and technologies in a short period of time. Android, face detection, voice identification and object database systems were just some of the main new technologies that needed to be learned for this project. Each of these held major problems of their own which needed much attention. For instance, it was difficult to make the application compatible with different types of Android devices. Each device had different quality of hardware, for instance processing power or camera quality which had a major impact on the performance of the application.

The voice and face identification was a challenge to implement on a normal java system, but to get it to work on an Android device was nearly impossible.

In the end most of the challenges were overcome and a working application was created.