Hochschule Esslingen
University of Applied Sciences

Project Name:	Valiquad
Document Title:	Project Manual
Autor(s):	Manuel Stübler
Created on:	22.10.10
Last modified:	15.11.10
Version:	v1.3
Status:	| | in progress | | presented |x| closed
Located at:	/docs/pm/project.odt

Revision History:

#	Date	Version	Section	Description	Author	Status
1	22.10.10	v1.0	all	First creation	Manuel Stübler	closed
2	23.10.10	v1.1	all	Refined description	Manuel Stübler	closed
3	02.11.10	v1.2	2,5	Refined description	Manuel Stübler	closed
4	15.11.10	v1.3	3	Updated artifacts	Manuel Stübler	closed

Table of Contents

List of Tables

1 Introduction

This document describes the internal structure of the "Valiquad" project. It defines the standards and roles that will be used within the project. It also lists the artifacts that will result from the development process and all responsibilities between the participating instances.

2 Project Description

Valiquad is an analyzing tool for tracking and remote monitoring the Quadrocopter with all of its stored and measured sensor data. The Quadrocopter is kind of a helicopter with four, oppositely located propellers. It is developed by the master's course at the "University of Applied Sciences Esslingen". The analyzing software, called Valiquad, is written in Java and uses radio-frequency to communicate with the helicopter. It can read out the parameters of the sensors that are mounted on the system and is also able to set new parameters for the controllers within the aircraft, to analyze the flight behavior with different settings.

Valiquad will primarily provide a view to the user, displaying the measured values over time in an adequate diagram. It will also display the current system relevant values in a separate widget. Therefore it is a GUI-based software and kind of a debugging and analyzing tool for the helicopter and its system parameters. This is necessary as it is not possible to use a stepwise debugger while the system is in the air, as this means that the micro-controller would be interrupted and therefore prevented from controlling the propellers, which would cause the helicopter to crash.

For the transmission of the data from the Quadrocopter to the analyzing software and vice versa, a special application layer protocol will be introduced, that matches the requirements of the system. This protocol will base upon a radio-frequency technology called ZigBee, which is one of the constraints to the system.

3 Process Overview

This is a brief description of the artifacts and products generated within the process of development.

Activity	Product	Customer Involvement
PM	Project manual (project.odt)	Information
RE	User manual (user.odt)	Information
SE-A	Requeriments document for the application (req-app.odt)	Information
SE-H	Requeriments document for the hardware (req-hw.odt)	Information
SE-A	Documentation of the software architecture and implementation (impl-app.odt)	
SE-H	Documentation of the software architecture and implementation (impl-hw.odt)	
SE-H	Protocol documentation (protocol.odt)	
IT	Testing and test case documentation (test.odt)	

Tabelle 3-1: Process Overview

4 Project Organization

This section describes the structure and organization used for the project.

4.1 *Rolls*

This table defines the rolls of the project.

Rolls
Project manager (PM)
Requirements Engineer (RE)
Software architect (SA)
Software engineer (application) (SE-A)
Software engineer (hardware) (SE-H)
Integration and test engineer (IT)
Customer (CUS)

Tabelle 4-2: Rolls

4.2 Tasks and Responsibilities

4.2.1 Project Manager (PM)

- Developing the project manual
- Tracing the development process
- Organizing team meetings
- Planning code reviews

4.2.2 Requirements Engineer (RE)

- Developing requirements and the according documentation
- Informing the customer about the project status
- Developing the solution concept and abstracting the development process
- Creating the user manual for the system

4.2.3 Software Architect (SA)

- Developing the overall software architecture and keeping track of the whole development process
- Creating UML-diagrams to illustrate the interaction between different software components
- Defining the API (application programming interface) between the upper and lower layer of the software

4.2.4 Software Engineer Application (SE-A)

- Developing and integrating the application specific software
- Creating the user interface of the software
- Evaluating different GUI-technologies
- Developing the view and controller according to the MVC-pattern

4.2.5 Software Engineer Hardware (SE-H)

- Developing and integrating the hardware specific software
- Developing the radio-frequency communication protocol and data synchronization between the application and the Quadrocopter
- Developing the model according to the MVC-pattern
- Implementing the communication protocol in Java for the application on the host and in C for the micro-controller on the remote system (the Quadrocopter)

4.2.6 Integration and Test Engineer (IT)

- Integrating and testing the final system
- Defining suitable test cases for the system
- Tracking the whole testing process of the system

4.3 Responsibilities

This list shows the role assignment in detail.

Roll	First name	Last name	Email	
PM, SE-H, SA	Manuel	Stübler	mastit10@hs-esslingen.de	
RE, SE-A, IT	Peter	Mayer	pemait01@hs-esslingen.de	

Table 4-3: Role assignment

4.4 Participation of the Customer

The following table describes the integration of the customer within the project.

No.	Activity	Method	Who?
1	System requirements	Review	PM, RE, CUS
2	Prototype presentation	Review	PM, RE, CUS
3	Product release	Review	PM, RE, CUS

Table 4-4: Involvement of the customer

5 Standards and Regulations

This section handles the defined standards and regulations for the overall development process.

5.1 Tools

The following table defines the tools that have to be used in the project.

Activity	Tool
Documentation	Open Office 3
Project Plan	GanttProject
Target IDE	CodeWarrior
Host IDE	Eclipse

Tabelle 5-5: Activity and Tools

5.2 Style Guideline

The code that is written in Java must adhere to the Java conventions. This can be accomplished by activating the Eclipse formatter using the pre-defined and built-in "Java Conventions" with a tab-size of 4.

The Java source-code has to be documented by using JavaDoc comments. This is used to automatically generate an API documentation out of the source files.

The code that is written in C for the micro-controller on the Quadrocopter has to keep to the established style that is already used within the existing source-code.