

Pflichtenheft

Projekt: Projektname

Version 0.1

Michael Trummer

Name	Datum	Unterschrift
Hans Mustermann	1. März 2017	

Änderungsübersicht

Datum	Version	Autor	Beschreibung
2017-03-01	0.1	M. Trummer	Anhand Word-Vorlage von Prof. R. Bonderer erstellt

Die Information in dieser Tabelle muss nur dann aktualisiert werden, wenn eine neue Version des Dokumentes erzeugt, bzw. freigegeben wird, nicht aber jedes Mal wenn das Dokument 'angefasst' wird.

- Alle Abschnitte, die im Style "Comment" geschrieben sind (wie dieser Text), dienen nur für Erklärungen, die dem Autoren des Dokumentes helfen sollen. Diese Kommentare müssen im endgültigen Dokument entfernt oder unsichtbar gemacht werden.
- Im Dokument werden verschiedene Felder, z.B. für das Datum, den Projektnamen, etc. verwendet. Diese sollen deshalb unbedingt zu Beginn unter den Doc properties (Dokumenteigenschaften) festgelegt werden.
- Die hier präsentierte Pflichtenheftvorlage ist angelehnt an **IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE Std 830-1998**. Sie kann auch für Projekte verwendet werden, die nicht nur aus Software bestehen.
- Gemäss DIN 69901-5 umfasst das Pflichtenheft die "vom Auftragnehmer erarbeiteten Realisierungsvorgaben aufgrund der Umsetzung des vom Auftraggeber vorgegebenen Lastenhefts", d.h. das Lastenheft beinhaltet die Kundenanforderungen, im Pflichtenheft sind technische Vorgaben an die Entwicklungsgruppe formuliert, z.B. allenfalls notwendige Vorgaben für die Programmiersprache, die Plattformen, Betriebssystem, etc... Im Pflichtenheft darf keinesfalls das Design beschrieben werden.
- Im internationalen Umfeld werden statt der DIN-Normen eher die IEEE-Normen angewandt. Im IEEE Standard 830 wird eine "Software Requirements Specification" formuliert, welche sowohl das Lastenheft als auch das Pflichtenheft beinhaltet. Diese Vorlage verfolgt diesen Ansatz. Teilweise werden Hinweise in Englisch direkt aus diesem Standard verwendet. Weitere Informationen zu den einzelnen Punkten finden Sie direkt in [1].
- The Requirements Specification should address the product, not the process of producing the product. Project requirements represent an understanding between the customer and the supplier about contractual matters pertaining to production of the product and thus should not be included in the Requirements Specification. These normally include items such as
 - Cost
 - Delivery schedules
 - Reporting procedures
 - Development methods
 - Quality assurance
 - Validation and verification criteria
 - Acceptance procedures

Project requirements are specified in other documents, typically in a software development plan, a software quality assurance plan, or a statement of work.

Inhaltsverzeichnis

Abbildungsverzeichnis	4
Tabellenverzeichnis	4
1. Einleitung	5
1.1. Zweck	5
1.2. Produktüberblick	5
1.3. Definitionen, Akronyme und Abkürzungen	5
1.4. Referenzen	5
2. Allgemeine Beschreibung	6
2.1. Systemübersicht	6
2.2. Produktfunktionen	7
2.3. Benutzereigenschaften	7
2.4. Einschränkungen	8
2.5. Annahmen und Abhängigkeiten	8
2.6. Priorisierung der Anforderungen	8
3. Externe Schnittstellen	9
4. Funktionale Anforderungen	10
4.1. Überblick über die Systemfunktionen	10
4.2. Actors	10
4.3. Kurzbeschreibung der Use Cases	10
4.4. Spannungsteiler	10
5. Sonstige Anforderungen	13
A. Referenzen	14

Abbildungsverzeichnis

2.1 Kontextdiagramm (Festlegung der Systemgrenze) 6

Tabellenverzeichnis

1. Einleitung

1.1. Zweck

Im vorliegenden Dokument sind die Anforderungen definiert, welche im Projekt Projektname umgesetzt werden müssen. Es beschreibt den Auftrag zwischen Auftraggeber und Auftragnehmer. Der Ausdruck Pflichtenheft ist hier im Sinne der IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE Std 830-1998 verwendet. Die dort definierte Requirements Specification beinhaltet sowohl die Benutzeranforderungen (Lastenheft gemäss DIN 69901-5) als auch Realisierungsvorgaben an die Entwicklungsgruppe (Pflichtenheft gemäss DIN 69901-5).

1.2. Produktüberblick

Hier soll ein Überblick gegeben werden über die Teile, welche im Rahmen dieses Projekts entwickelt werden sollen, d.h. eine summarische Beschreibung darüber, welche Funktionalität das Produkt haben, und eventuell auch welche es nicht haben soll. Die zu entwickelnden Teile sollen namentlich erwähnt werden. Ausserdem kann auch erläutert werden, wie das Produkt eingesetzt werden soll.

Use Case ("Funktionalität")

1. Eingangs- und Ausgangsspannung angeben durch den Benutzer
2. Der Benutzer muss eine E-Serien (E12 oder E24) auswählen
3. Minimaler und Maximaler Widerstandswert angeben durch den Benutzer
4. Herausgabe des geeigneten Widerstandsverhältnisses
5. Minimierung des Abstands von beliebigen Widerstandswerten der E-Reihe zum optimalen Widerstandsverhältnis

1.3. Definitionen, Akronyme und Abkürzungen

(Optional) Define all terms, acronyms, and abbreviations used in this document. This often goes into a separate document in a larger project.

Spezielle Bezeichnungen für Abläufe und Prozesse werden hier niedergeschrieben.

1.4. Referenzen

Siehe Anhang A auf Seite 14 dieses Dokuments.

2. Allgemeine Beschreibung

In diesem Kapitel sollen Hintergrundinformationen gegeben werden, keine spezifischen Anforderungen. Diese folgen im nächsten Kapitel. In diesem Kapitel muss genau geklärt werden, was zum System gehört und was nicht, d.h. die Systemgrenze muss hier zwingend festgelegt werden.

2.1. Systemübersicht

Definieren Sie das Umfeld des Systems mit den Schnittstellen des Systems zu seinem Umfeld und das System selbst. Hier muss die Systemgrenze in Form eines Kontextdiagramms gezogen werden. Aus diesem Abschnitt muss eindeutig hervorgehen, was innerhalb der Systemgrenze liegt (was muss entwickelt werden?) und was ausserhalb.



Abbildung 2.1: Kontextdiagramm (Festlegung der Systemgrenze)

This subsection should also describe how the system operates inside various constraints. For example, these constraints could include

1. System interfaces
2. User interfaces
3. Hardware interfaces
4. Software interfaces
5. Communications interfaces
6. Memory
7. Operations
8. Site adaptation requirements

2.) Erstellung eines GUI für die erwähnte Applikation -> Verwendung von Qt & C++.

2.2. Produktfunktionen

Hier soll eine **Zusammenfassung** der Hauptfunktionen angeboten werden, welche das System erfüllen soll.

2.3. Benutzereigenschaften

Festlegung der Benutzer (Benutzergruppen), welche mit dem System arbeiten sollen, z.B. Routinebenutzer, Servicetechniker, Administrator. Dazu gehört auch die Definition der Kenntnisse und Erfahrungen der einzelnen Gruppen. Falls die funktionalen Anforderungen mittels Use Case – Diagrammen formuliert werden, dann entspricht dies den Actors, welche Personen repräsentieren.

2.4. Einschränkungen

This subsection of the SRS should provide a general description of any other items that will limit the developer's options. These include:

1. Regulatory policies:
2. Hardware limitations (e.g., signal timing requirements)
3. Interfaces to other applications
4. Parallel operation
5. Audit functions
6. Control functions
7. Higher-order language requirements: C++
8. Signal handshake protocols (e.g., XON-XOFF, ACK-NACK)
9. Reliability requirements:
10. Criticality of the application: Nein
11. Safety and security considerations: Nein

GUI-Programmierung mit Qt. Angewiesen auf seine Vorlage.

2.5. Annahmen und Abhängigkeiten

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the system but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

2.6. Priorisierung der Anforderungen

Falls die Anforderungen unterschiedliche Prioritäten haben, bzw. einzelne Anforderungen erst in einer späteren Version implementiert werden sollen, dann kann das hier aufgelistet werden, z.B.

- Muss-Anforderung: dies ist eine Anforderung, welche für das System essentiell und unabdingbar ist, bzw. das System würde keinen Sinn ergeben, wenn diese Anforderung nicht implementiert wäre.
- Soll-Anforderung: eine Soll-Anforderung ist nicht unabdingbar, trägt jedoch zur wesentlichen Verbesserung des Systems bei. Sie soll wenn möglich realisiert werden.
- Wunsch-Anforderung (nice to have): diese Anforderung trägt zur Verbesserung des Systems bei, ist jedoch nicht unbedingt notwendig. Es ist ein Plus wenn diese Anforderung realisiert werden kann.

3. Externe Schnittstellen

Ab diesem Abschnitt sollen alle Anforderungen in einem Detaillierungsgrad geschrieben sein, der es den Entwicklern ermöglicht, ein System zu entwerfen, welches diese Anforderungen erfüllt, d.h. der Entwickler sollte nun klar wissen, welche Anforderungen umgesetzt werden müssen. Ebenfalls müssen Testingenieure aufgrund dieser Anforderungen ihre (System-)Testfälle definieren können.

Jede Anforderung sollte eindeutig identifizierbar sein, beispielsweise durch eine eindeutige Nummer.

In diesem Kapitel sollen, allenfalls in Abschnitten organisiert, die folgenden Schnittstellen beschrieben werden:

- Benutzerschnittstellen : JA -> GUI
- Hardwareschnittstellen: NEIN
- Softwareschnittstellen: NEIN
- Kommunikationsschnittstellen: NEIN

4. Funktionale Anforderungen

Die funktionalen Anforderungen sind die wichtigsten Teile zur Beschreibung des Systems. Für die Beschreibung der funktionalen Anforderungen bieten sich Use Cases an, nicht nur für Software, sondern auch für das ganze System. Jeder Use Case beschreibt dabei eine Funktion. Im folgenden Raster ist vorgesehen, dass bei jedem Use Case auch nicht funktionale Anforderungen wie Antwortzeiten, etc. direkt beim Use Case stehen, falls diese zu dieser Funktion gehören.

Bei der Use Case Definition ist wichtig, dass die Granularität nicht zu fein gewählt wird. Allfällige Ausnahmefälle einer Funktion sollen beispielsweise in der Beschreibung des entsprechenden Use Cases geschehen und nicht allenfalls in einem "Unter-Use Case".

Statt mittels Use Cases kann dieses Kapitel auch mit **Systemfunktion 1**, **Systemfunktion 2**, etc. gegliedert werden.

4.1. Überblick über die Systemfunktionen

Hier kommt ein Use Case Diagramm hinein, das alle Funktionen zeigt. Allenfalls genügt ein Link zum Abschnitt Systemübersicht.

4.2. Actors

Kurzbeschreibung der Actors.

4.3. Kurzbeschreibung der Use Cases

Jeder einzelne Use Case soll kurz beschrieben werden.

Berechnung von 2 Widerstandswerten von einer bestimmten E-Serie, basierend auf einer Eingabe- und Ausgabespannung.

4.4. Spannungsteiler

Ist für jeden Use Case zu beschreiben. Die folgenden Unterkapitel sollen für jeden einzelnen Use Case vollständig vorhanden sein. Falls bei einem Abschnitt nichts zu schreiben ist, dann soll dies entsprechend vermerkt werden, z.B. falls ein Use Case keine Vorbedingungen braucht oder keine nicht-funktionalen Anforderungen vorhanden sind, kann bei diesem Abschnitt einfach das Wort "keine" stehen.

Berechnung von 2 Widerstandswerten von einer bestimmten E-Serie, basierend auf einer Eingabe- und Ausgabespannung.

4.4.1. Vorbedingungen

Zustand des Systems bevor der Use Case eintritt, z.B. kann hier stehen, dass ein System erfolgreich initialisiert sein muss, damit diese Funktion ausgeführt werden kann.

Keine Vorbedingungen.

4.4.2. Nachbedingungen

Zustand des Systems nachdem der Use Case durchlaufen ist, z.B. kann hier bei einem Kalibrations-Use Case stehen, dass das System kalibriert oder allenfalls in einem Fehlerzustand ist.

Widerstandswerte angegeben oder Fehlermeldung.

4.4.3. Nicht-funktionale Anforderungen

Zusicherungen, die für Design und Realisierung wichtig sind, wie z.B. Antwortzeit, Häufigkeit, Priorität usw.

4.4.4. Hauptszenario

Beschreibung des Use Cases, ggf. gegliedert in Einzelpunkte. Beschrieben wird der Normalfall. Variationen werden mit Unterszenario-Nummer erwähnt (*[S-1]*, *[S-2]*, usw.) und separat als Unterszenarien beschrieben. Fehlerfälle werden mit Fehlerszenario-Nummern angegeben (*[E-1]*, *[E-2]*, usw.) und separat als Fehlerszenarien beschrieben.

Beispiele: Falls gewünscht können zusätzliche Informationen erfasst werden *[S-1]*.

Beim Einlesen der Daten können die Fehler *[E-1]*, *[E-2]* oder *[E-3]* auftreten.

4.4.5. Unterszenarien

[S-1] Zusatzinformationen erfassen

[S-2] ...

4.4.5.1. [S-1] Zusatzinformationen erfassen**4.4.5.2. [S-2] ...****4.4.6. Fehlerszenarien**

[E-1] Daten nicht verfügbar

[E-2] Falsches Datenformat

[E-3] ...

4.4.6.1. [E-1] Daten nicht verfügbar**4.4.6.2. [E-1] Falsches Datenformat****4.4.6.3. [E-2] ...****4.4.7. Regeln**

Gültigkeits- und Validierungsregeln, Berechnungsformeln usw.

4.4.8. Anmerkungen**4.4.9. Beispiele**

5. Sonstige Anforderungen

In diesem Kapitel sollen alle bisher noch nicht spezifizierten Anforderungen, möglicherweise in Unterabschnitten, definiert werden, z.B.

- Leistungsanforderungen (Performance), die nicht direkt einer Systemfunktion zugeordnet werden konnten.
- Entwurfseinschränkungen (Design Constraints): Anforderungen, welche die Entwickler bei der Wahl des Designs einschränken.
- Qualitätsanforderungen wie Zuverlässigkeit, Verfügbarkeit, Sicherheit (Safety und Security, ist nicht dasselbe!)
- Wartbarkeit
- Portabilität
- Logging und Tracing
- Service und Support

A. Referenzen

- [1] IEEE. *Recommended Practice for Software Requirements Specification*. ANSI/IEEE Std 830-1998, 1998.

Do not repeat items covered in other documents or in a global project definitions and acronyms document