Konzipierung und Erstellung eines Parametertabellenkonfigurators und Praxisnahe Mitarbeit im releasten Produkt LC-VISION

Projektarbeit T1000
des Studiengangs Informatik
an der Dualen Hochschule Baden-Württemberg Ravensburg

von

Janik Frick

Kurs: TIT21, Matrikelnummer 4268671
Dualer Partner: Blum-Novotest GmbH, Standort Grünkraut
Betreuer: Mallmann, Guilherme, Dr.-Ing.

Inhaltsverzeichnis

1 Einleitung **2**

2 Softwareentwicklung **3**

2.1 Anforderungsanalyse . . . 3

2.1.1 Interviews . . . 3

2.1.2 Fragebögen . . . 3

2.1.3 Beobachtung . . . 4

2.1.4 Anforderungstypen . . . 4

2.2 Grobentwurf . . . 4

2.2.1 GUI mit Bidern . . . 4

2.2.2 Prototypen . . . 5

2.2.3 User-Tests . . . 5

3 Parametertabellenkonfigurator **5**

3.1 Problemstellung . . . 5

3.2 Anforderungen . . . 6

3.3 Entwurf Benutzeroberfläche . . . 6

3.4 Programmierung . . . 7

3.4.1 Technologien . . . 7

3.4.2 Software-Struktur . . . 8

3.4.3 Verarbeiten von Benutzeraktionen . . . 9

3.4.4 Herausforderungen und Lösungen . . . 9

3.5 Tests . . . 9

3.6 geplante Erweiterungen . . . 9

3.7 Reflexion . . . 9

4 LC-Vision **9**

1 Einleitung

Die Blum-Novotest GmbH(Blum) ist im Bereich Maschinenbau für Mess- und Prüftechnik tätig. Am Standort Grünkraut (Baden-Württemberg) ist der Hauptsitz der Firma. Dort befinden sich Entwicklung und Fertigung für den Bereich Messtechnik. Am zweiten Standort in Willich (Nordrhein-Westfalen) sind die Entwicklung und Fertigung im Bereich Prüftechnik untergebracht.
Produkte von Blum werden in vielen anspruchsvollen Industrien im Bereich der Qualitätssicherung eingesetzt. Die Kunden kommen unter anderem aus der Automobil-, Luftfahrt- und der Werkzeugmaschinenindustrie.
An die Produkte dieser Industrien werden höchste Ansprüche in Sachen Qualität gestellt. Daraus resultieren auch für die Produkte der Firma Blum höchste Ansprüche.
Um diese Ansprüche erfüllen zu können, werden sowohl bestehende Produkte laufend optimiert und weiterentwickelt, als auch neue Produkte entwickelt.
Die Arbeit wird im Umfeld von Blum am Standort Grünkraut verfasst.
Ziel der Arbeit ist es die vielfältigen Aspekte der Softwareentwicklung kennenzulernen und den Einstieg zu schaffen.
Um dieses Ziel zu erreichen, besteht der praktische Teil aus zwei Projekten.
Mit der Entwicklung eines „Parametertabellenkonfigurators“ sollen die verschiedenen Aufgaben der Softwareentwicklung veranschaulicht werden.
Durch die Mitarbeit am schon bestehenden Produkt „LC-Vision“ wird die Produktpflege und Weiterentwicklung thematisiert. Außerdem wird dabei das Einarbeiten in unbekannten Code relevant, was ein wichtiger Bestandteil der Produktpflege ist.

2 Softwareentwicklung

Softwareentwicklung besteht nicht nur aus dem Schreiben von Software. Auch die Entwicklung von Software ist ein Projekt. Man beginnt mit dem Sammeln der Anforderungen, startet die Entwicklung von Prototypen und geht in die Implementierung. Der Prozess endet mit der Produktpflege und dem Service.

2.1 Anforderungsanalyse

Eindeutige und präzise Anforderungen bilden die Grundlage für ein funktionierendes Projekt[1].
Daraus folgt, dass es nicht ausreichend ist, die Liste der Anforderungen zu lesen. Anforderungen müssen analysiert und mit weiterem Wissen über das Projektumfeld kombiniert werden.
Dafür gibt es mehrere Techniken, die eingesetzt werden können.

2.1.1 Interviews

Interviews bieten die Möglichkeit in direkten Gesprächen mit Personen aus dem Umfeld des Projekts Informationen zu sammeln. Diese Informationen können Risiken sein, die zu berücksichtigen sind, Schwierigkeiten bei der aktuellen Vorgehensweise, sowie Abläufe die beibehalten werden sollten.
Der Erfolg dieser Vorgehensweise hängt dabei von allen Beteiligten Personen ab. Fehlt Wissen über das Umfeld des Projekts kann es zu Schwierigkeiten bei der Erstellung der Fragen kommen, wodurch manche Aspekte nicht berücksichtigt werden. Auch die Befragten können Probleme haben ihre Gedanken und ihr Wissen wiederzugeben[2]. Das kann die Auswertung der Antworten erschweren.

2.1.2 Fragebögen

Fragebögen geben den Befragten Personen die Möglichkeit ohne direkte Beeinflussung durch den Ersteller der Fragen, Informationen bereitzustellen. Zusätzlich reduzieren sie den Zeitdruck unter dem geantwortet werden soll und alle Befragten antworten auf die gleiche Fragestellung.
Fragebögen können eingesetzt werden um Annahmen zu bestätigen oder nach Meinungen und Vorschlägen zu fragen[2].
Bei Einsatz dieser Methode ist darauf zu achten die Fragen offen zu formulieren, um die Antworten nicht in eine Richtung zu lenken.

2.1.3 Beobachtung

Die Beobachtungsmethode dient dazu aus beobachteten Vorgängen Informationen zu sammeln.
Diese Beobachtung kann offen oder verdeckt durchgeführt werden. Bei der verdeckten Vorgehensweise ist das beobachtete Verhalten natürlicher und realistischer, als bei der offenen Vorgehensweise[3].
Mit diesem Vorgehen lassen sich Abläufe bei Aufgaben nachvollziehen. Gründe für dieses Abläufe sollten durch andere Techniken in Erfahrung gebracht werden.

2.1.4 Anforderungstypen

Anforderungen können in zwei Typen unterteilt werden: Die funktionalen und die nicht funktionalen Anforderungen.
Funktionale Anforderungen geben vor, für welche Aufgaben das Produkt geplant wird und welche Funktionen dafür benötigt werden. Die Validierung dieser Anforderungen ist gegeben, da am Ende eindeutig ist, ob eine Funktionalität verfügbar ist.
Die nicht funktionalen Anforderungen geben vor, mit welchem Vorgehen und welchen Tools eine Funktion umgesetzt werden soll[4].
Nicht funktionale Anforderungen sind häufig unpräzise formuliert und somit problematisch in der Validierung[4].

2.2 Grobentwurf

Beim Grobentwurf werden die Hauptbestandteile der Software geplant. Dabei sollte der Kunde möglichst miteinbezogen werden, denn der Kunde hat großen Anteil am Gelingen des Projekts[5].
Um den Kunden einzubeziehen können Prototypen eingesetzt werden.
Hat das Programm eine Graphische Benutzeroberfläche(GUI) so kann der Grobentwurf unterschiedlich geplant werden.

2.2.1 GUI mit Bidern

Der Grobentwurf der GUI kann mit Hilfe von Bildern in einem beliebigen Bilderstellungsprogramm nachgebildet werden. Die erstellten Entwürfe dienen als Gesprächsgrundlage für eventuell notwendige Veränderungen.

2.2.2 Prototypen

Ein Prototyp ist eine funktionierende, begrenzte Version der Anwendung, die als Basis für Gespräche und die Entwicklung weiterer Prototypen dient[6].
Dieser Prototyp muss nicht durch Software erstellt werden, sondern kann auch mit einem Mockup-Tool[1] realisiert werden.

2.2.3 User-Tests

Bei User Tests wird einer oder mehreren Testpersonen, abhängig von der Verfügbarkeit, der aktuelle Prototyp vorgelegt. Dann kann man der Testperson eine konkrete Aufgabe geben, oder sie den Prototyp frei entdecken lassen. Bei beiden Vorgehensweisen werden Probleme dokumentiert. Hat man mehrere Testpersonen zur Verfügung können auch Gruppentests durchgeführt werden. Bei diesen arbeiten mehrere Personen an einer Aufgabe. BEi Einzel-Tests lassen sich Probleme im Ablauf besser erkennen. In Gruppen-Tests hingegen werden die tatsächlichen Probleme der Testpersonen besser erkenntlich.[7].

3 Parametertabellenkonfigurator

Der „Parametertabellenkonfigurator“ ist ein internes Projekt, bei dem Mitarbeitende der Abteilung Service-International die Rolle des Kunden übernehmen. Dadurch wird der Einstieg in die Softwareentwicklung vereinfacht.
Das hat mehrere Gründe. Zum Einen ist der Kunde direkt vor Ort und Fragen lassen sich schneller klären, zum Anderen gibt es keinen finanziellen Druck, der bei externen Auftraggebern hinzukommen würde. Auch der zeitliche Druck ist stark reduziert, da das Produkt eine Unterstützung bieten soll. Die Arbeit kann aber auch ohne diese Unterstützung ungehindert fortgeführt werden.

3.1 Problemstellung

Bei bestehenden Programmen für numerische Steuerungen müssen während der Inbetriebnahme Parametertabellen in einem beliebigen Texteditor angepasst werden. Aufgrund der Anzahl, wie auch der Einstellmöglichkeiten der Parameter, wird hierfür eine separate Installationsanleitung benötigt. Der Prozess der Parametrierung ist daher Fehleranfällig und von der Erfahrung des Inbetriebnehmers abhängig. Um den hohen Anforderungen der Kunden gerecht zu werden, gilt es potentielle Fehlerquellen zu eliminieren. Hierfür soll selbstständig ein „Parametertabellenkonfigurator“ entwickelt werden.

[1] Tool um eine GUI zu simulieren

3.2 Anforderungen

Die funktionellen Anforderungen wurden vom Kunden direkt zu Projektstart bereitgestellt. Dennoch war damit die Anforderungsanalyse noch nicht abgeschlossen, denn es fehlten noch Informationen über den Hintergrund des Projekts.
Dieser wurde während Gesprächen mit dem Kunden von Zeit zu Zeit deutlicher; Es geht darum ein Tool zu entwickeln, welches vor allem die unerfahreneren Mitarbeiter im Service bei der Konfiguration der Produkte von Blum zu unterstützen.
Die funktionellen Anforderungen besteht aus zwei Teilen:
Die Parametertabellen und die Beschreibung der einzelnen Parametern.
Für die Parametertabellen müssen Funktionen für das Laden, das Bearbeiten und das Speichern entwickelt werden.
Im Bereich der Beschreibungen geht es darum, die richtige Beschreibung zu laden und anzuzeigen. Da die Beschreibungen nicht angepasst werden müssen, wir hier keine Funktion zum Speichern benötigt.

3.3 Entwurf Benutzeroberfläche

Nach dem alle Anforderungen besprochen war, konnte mit den Überlegungen für die Benutzeroberfläche begonnen werden.
Um die Überlegungen schnell in einen sichtbaren Entwurf zu übertragen, wird das Programm „Balsamiq Mockups 3“ verwendet. Mit diesem Tool können alle gängigen Designs nachgebaut werden und auch mit Funktionalität, zum Beispiel das anklicken von Buttons oder das auswählen von Optionen in Drop-down Menüs können somit schnell und einfach visualisiert werden. Das ist hilfreich, um Designvorschläge und Ideen besprechen zu können, ohne das man die Änderungen aufwendig programmieren muss.
Zusätzlich bietet der erstellte Entwurf eine gute Unterstützung für die Implementierung des Designs, da man sich den Aufbau der Software in Bezug auf Menüs und Buttons nicht merken muss, sondern diesen immer konkret vorliegen hat.
Für den „Parametertabellenkonfigurator“ fiel die Wahl auf ein Design, dass aus mehreren Fenstern aufgebaut ist, um die Bedienbarkeit möglichst einfach zu halten, denn die Anzahl an Auswahlmöglichkeiten würde innerhalb eines Fensters die Übersichtlichkeit deutlich reduzieren.
Wäre alles in einem Fenster, müsste durch „ausgrauen“ oder ausblenden der Optionen dargestellt werden, welche momentan verfügbar sind. Das ist zwar möglich, ersteres führt aber bei manchen Anwendern zu Verwirrung, oder Unsicherheiten, was den Konfigurationsprozess der Produkte von Blum negativ beeinflussen könnte, zweites ist optisch nicht ansprechend, da immer wieder unerwartete, freie Flächen entstehen und eventuell Layouts verschoben werden.

Diese Problematik kann durch das aufteilen in mehrere Fenster behoben werden, da nur die im Fenster angezeigten Optionen auch verfügbar sind.
Da von Anfang mindestens ein zweites Fenster geplant war, um die Beschreibungen der einzelnen Parameter in einer vergrößerten Ansicht anzeigen zu können, fügte sich auch das Fenster für die Auswahl der benötigten Parametertabelle problemlos in das Design ein.
Das Hauptfenster ist in zwei Bereiche unterteilt. Der obere ist für die Parametertabelle allgemein, er besteht aus eine Anzeige, um zu sehen welche Parametertabelle ausgewählt ist, ein Button um eine andere Parametertabelle auswählen zu können und ein Feld im der der Inhalt angezeigt wird und bearbeitet werden kann. Um den Einrichtungsprozess zu unterstützen, werden alle bearbeiteten Zeilen farblich markiert.
Im unteren Teil geht es um die einzelnen Parameter. Es gibt eine Anzeige für den aktuell ausgewählten Parameter, einen Bereich um die Beschreibung, wenn vorhanden mit Bild, anzuzeigen und einen Button um die Beschreibung isoliert und vergrößert in einem neuen Fenster zu öffnen. Um auch hier die Bedienbarkeit zu erleichtern gibt es zwei weitere Buttons die es ermöglichen schnelle zu vorherigen, beziehungsweise dem nachfolgenden Parameter zu springen.

3.4 Programmierung

Für die Programmierung des „Parametertabellenkonfigurator“ kommt die C++-Erweiterung Qt zum Einsatz. Qt wird verwendet, da diese spezielle Komponenten beinhaltet, die für das Entwickeln von Programmen mit GUI konzipiert sind. Dazu gehört neben vielen vordefinierten Klassen für Widgets auch der Signal-SLot Mechanismus, der später noch genauer thematisiert wird.
Im Projekt kommen „Clean-Code Prinzipien“ für eine erhöhte Les- und Wartbarkeit des Codes und die objektorientierte Programmierung zum Einsatz.

3.4.1 Technologien

Um den Prozess der Programmierung zu vereinfachen kamen zusätzliche Tools zum Einsatz.
Optimiert für die Programmiersprache Qt wird der „Qt Creator“ [2] als Entwicklungsumgebung verwendet. Dieser bietet zum Beispiel einen speziellen Designer, mit dem man das GUI visuell designen kann, ohne alles zu programmieren.
Da das persönliche Ziel des Projekts aber im Einstieg in die Softwareentwicklung liegt, wurde auf diese Möglichkeit verzichtet und der Aufbau des GUI erfolgte vollständig per Programmierung.
Für die Versionsverwaltung kommt Git zum Einsatz. Lokal wird die Zusatzsoftware Git-

[2] Herausgegeben von der Qt Company

Kraken verwendet. Mit GitKraken wird die lokale Arbeit vereinfacht, da man ohne das Terminal die für Git-Relevanten Funktionen schnell nutzen kann. Dazu gehört unter anderem das Laden von alten Softwareständen, die durch einfaches anklicken eines Commits[3] direkt in der Entwicklungsumgebung bearbeitet werden können. Die Verwendung von Git erleichtert auch Software-Reviews[4], da man den Zugang zu einem Repository kontrolliert freigeben kann und somit unabhängig von der Anwesenheit des Entwicklers Zugang zum letzten auf dem Server gespeicherten Zustand möglich ist. Auch das Arbeiten auf mehreren Geräten ist durch den Serverzugriff problemlos möglich (wenn Git auf dem entsprechenden Gerät installiert ist).
Zusätzlich kommt die Pipeline-Funktion von GitLab zum Einsatz.
GitLab Pipeline
Die GitLab Pipeline ermöglicht es jeden auf den Server geladenen Commit automatisiert zu verarbeiten.
Für die Weiterverarbeitung können Shell-Scripts[5] verwendet werden.
Möglichkeiten für die Verarbeitung sind Scripts zum Beispiel das Bauen der Software, oder das Erstellen von Softwarepaketen. Für Softwarepakete gibt es die Option Artefakte[6] zu erstellen. Dadurch hat man automatisiert alle Dateien, die gebraucht werden, an einem Ort zur Verfügung.

3.4.2 Software-Struktur

Die Software ist modular aufgebaut, das bedeutet, dass sie aus mehreren möglichst unabhängigen Modulen besteht. Durch diese modulare Struktur wird das Ziel verfolgt Komponenten bei Bedarf anpassen oder austauschen zu können, ohne viel an den anderen Modulen anpassen zu müssen.
Außerdem ist die Software objektorientiert programmiert werden, da sich GUI-Elemente gut als einzelne Objekte eignen. Jedes Fenster der Anzeige ist eine eigene Klasse und ein Modul. Um die Module zu verwalten gibt es zwei Manager-Klassen. Der WindowManager ist dafür verantwortlich das richtige Fenster zur richtigen Zeit anzuzeigen, oder zu verbergen. Die zweite Manager-Klasse ist der DataManager. Er ist die Verbindungsstelle vom Logikmodul mit den GUI-Modulen. Er ist dafür verantwortlich die Benutzeraktionen entgegenzunehmen und dann die richtigen Daten an bereitzustellen.

[3] gespeicherter Zwischenstand des Projekts
[4] Feedback zum Code
[5] Dateien in denen mehrere Befehle stehen, die durch Aufruf der Datei in einem Kommandozeilenprogramm hintereinander ausgeführt werden
[6] Dateien, die nach Ausführung der Pipeline heruntergeladen werden

3.4.3 Verarbeiten von Benutzeraktionen

Benutzer führen ihre Aktionen auf der GUI aus. Die GUI muss auf verschiedene Aktionen der Benutzer reagieren können. Dafür gäbe es die Möglichkeit, dass sich die verschiedenen benötigten Klassen untereinander kennen. Diese Möglichkeit sorgt aber für starke Abhängigkeiten, wodurch die Modularisierung der Software

3.4.4 Herausforderungen und Lösungen

3.5 Tests

3.6 geplante Erweiterungen

3.7 Reflexion

4 LC-Vision

Literatur

[1] S. Lane, P. O'Raghallaigh, and D. Sammon, "Requirements gathering: the journey," *Journal of Decision Systems*, vol. 25, no. sup1, pp. 302–312, 2016.

[2] S. Tiwari, S. S. Rathore, and A. Gupta, "Selecting requirement elicitation techniques for software projects," in *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, pp. 1–10, IEEE, 2012.

[3] R. Silhavy, P. Silhavy, and Z. Prokopova, "Requirements gathering methods in system engineering," *Recent Researches in Automatic Control*, pp. 105–110, 2011.

[4] J. Eckhardt, A. Vogelsang, and D. M. Fernández, "Are "non-functional"requirements really non-functional? an investigation of non-functional requirements in practice," in *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, (New York, NY, USA), p. 832–842, Association for Computing Machinery, 2016.

[5] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Commun. ACM*, vol. 48, p. 72–78, may 2005.

[6] R. Budde, K. Kautz, K. Kuhlenkamp, and H. Züllighoven, "What is prototyping?," *Information Technology & People*, 1992.

[7] J. C. Bastien, "Usability testing: a review of some methodological and technical aspects of the method," *International Journal of Medical Informatics*, vol. 79, no. 4, pp. e18–e23, 2010. Human Factors Engineering for Healthcare Applications Special Issue.