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

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 "Parametertabellenkonfiguratorsßollen 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

2.1 Anforderungsanalyse

Bei der Anforderungsanalyse geht es darum möglichst viele Informationen zu sammeln. Diese Informationen können aus allen Bereichen und von allen Personen die im Projektumfeld tätig sind. Ziel davon ist es, nicht nur zu wissen, welche Funktionen von dem Produkt erwartet werden, sondern auch um herauszufinden, welche Standards es in diesen Bereichen gibt und welche Faktoren berücksichtigt werden müssen. Dazu zählen beispielsweise Sicherheitsstandards und das spätere Einsatzgebiet.
Mit Hilfe dieser zusätzlichen Informationen über das Projektumfeld können manche Faktoren schon früh berücksichtigt werden, wodurch man beispielsweise Schnittstellen für externe Sicherheitssoftware schon einplanen kann.

2.1.1 Anforderungstypen

Anforderungen können in zwei Typen unterteilt werden: Die funktionellen und die nicht funktionellen Anforderungen.
Bei den funktionellen Anforderungen handelt es sich um die Anforderungen die direkt vom Kunden kommen. Diese Anforderungen definieren welche Funktionen das Produkt bereitstellen soll.
Diese Anforderungen sind häufig einfach zu definieren, da es eine konkrete Problemstellung gibt, die durch dieses Produkt gelöst werden soll. Die klare Definition erleichtert die Kommunikation über diese Art von Anforderungen. Außerdem ist dadurch das Testen gut möglich, da das Verhalten des Produkts klar vorgegeben ist. Diese Anforderungen haben die höchste Priorität, da von ihnen der Erfolg des Produktes abhängt.
Die nicht funktionellen Anforderungen sind wesentlich komplexer zu definieren. Diese geben vor, mit welchen Technologien und Vorgehensweisen die funktionellen Anforderungen umgesetzt werden sollen.
Diese werden meistens durch Personen mit entsprechendem Fachwissen definiert. Sie können sowohl vom Kunden, als auch vom Hersteller definiert kommen.
Diese Anforderungen sind als Orientierungshilfe gut geeignet, sind aber nicht zwingend umzusetzen, da bei der Definition nicht alle Details bedacht werden können und somit kommt es in diesem Bereich immer wieder zu Änderungen.

2.2 Prototyping

Nach dem die wichtigsten Anforderungen herausgefiltert wurden, beginnt man damit einen Prototypen zu entwickeln.
Ein Prototyp dient dazu die gewünschten Abläufe und das "Graphical User Interface"(GUI) so umzusetzen, dass man klar erkennen kann, ob die Umsetzung in die richtige Richtung geht. Dabei müssen zum Beispiel angezeigte Texte nicht zwingend mit denen zusammenpassen, was später angezeigt wird. Prototypen dienen als gute Grund-

lage für interne und externe Gespräche über den aktuellen Stand und den Fortschritt des Projekts. Während der Entwicklung können auch GUI-Tests verwendet werden.

2.2.1 GUI-Tests

Hat das Produkt ein GUI können Prototypen und GUI-Tests gut eingesetzt werden, um Informationen über den Anwender zu sammeln. Diese Daten können eingesetzt werden, um das GUI zu verbessern und weiterzuentwickeln.

Click-Test
Für den Click-Test kann man entweder ein statisches Bild eines GUI-Bereichs (zum Beispiel einen Screenshot), oder ein Mockup Tool[1] verwenden.
Für die Durchführung zeigt man einer Versuchsperson einen Ausschnitt des GUI und gibt ihr eine Aufgabe die zu diesem Ausschnitt passt. Die Versuchsperson hat die Aufgabe die entsprechende Stelle zu finden und zu zeigen. Die Person, die den Test durchführt hat die Aufgabe währenddessen möglichst genau darauf zu achten, was die Versuchsperson macht und wie sie reagiert.
Möchte man die Versuchsperson nicht durch Anwesenheit unter Druck setzen, kann man fortgeschrittene Hilfsmittel wie Heatmap-Tools[2] nutzen. Die Heatmap gibt durch farbliche Kennzeichnungen Aufschluss darüber, an welchen Stellen von den Versuchsperson(-en) besonders häufig nach der Funktion für die Aufgabe gesucht wurde. Ohne ein Heatmap-Tool muss diese Aufgabe von der durchführenden Person mit Notizen oder Erinnerungen selbst übernehmen.
Je nach dem wie schnell die Testpersonen die richtige Stelle gefunden haben, muss das GUI nochmal angepasst werden, oder es kann so übernommen werden. 5-Sekunden Test

Beim 5-Sekunden Test wird den Versuchspersonen für 5 Sekunden ein statischer Ausschnitt des GUI gezeigt. Danach werden Fragen bezüglich des Bildes gestellt. Es kann beispielsweise gefragt werden, was positiv oder negativ aufgefallen ist.
Je nach Antworten können erste Vermutungen über die Qualität des GUI angestellt werden. Da die Informationen aber nur in einer sehr kurzen Zeitspanne für die Versuchspersonen verfügbar sind, können nicht alle Aspekte wahrgenommen werden.
Deshalb sollte der 5-Sekunden Test mit einer anderen Testmethode kombiniert werden. Besonders häufig genannte Störfaktoren, die die Testpersonen schon innerhalb 5 Sekunden abgeschreckt haben, können aber auch schon auf Basis dieses Tests identifiziert und angepasst werden. Im Gegensatz sollten positive Rückmeldungen auf jeden Fall nochmal geprüft werden, da positive Auffälligkeiten nach längerer Zeit auch störend wirken können.

2.2.2 Rolle des Kunden

Für die Entwicklung eines Prototypen müssen wichtige Entscheidungen treffen: Wie sehr wird der Kunde einbezogen?

[1] Tool um das GUI zu simulieren
[2] Tool um die Mausbewegungen aufzuzeichnen

Unterstützt der Kunde von Beginn an bei der Erstellung des Prototypen und ist somit laufend bei den Entscheidungen involviert, oder dient er als Partner für das Testen der Prototypen? Unabhängig von der Entscheidung bei dieser Frage ist sicher, dass der Kunde eine entscheidende Rolle für den Erfolg spielt. Denn von ihm wird die Entscheidung getroffen, ob und wie ein Prototyp nochmals überarbeitet werden soll.
Diese Verantwortung bringt aber auch hohe Anforderungen mit sich. "The success of agile development hinges on finding customers who will actively participate in the development process. Further, the customers are expected to be 'Collaborative, Representative, Authorized, Committed, and Knowledgeable'" [2]
Fehlen einer verantwortlichen Person beispielsweise wichtige Berechtigungen um Entscheidungen treffen zu können, kann es zu Verzögerungen und dem Fehlen wichtiger Freigaben kommen. Fehlt das Engagement beim Kunden werden möglicherweise Anforderungen unpräzise formuliert oder es fehlen wichtige Anforderungen.
Diese Probleme beeinflussen den Entwicklungsprozess negativ. Insbesondere unpräzise Anforderungen könnten durch entstehende Missverständnisse ein Projekt zum Scheitern bringen.

2.3 Implementierung

Abhängig davon, in welcher Form ein finaler Prototyp entwickelt wurde, gestaltet sich die Implementierung der Software.
Wurde der Prototyp von Beginn an in Code geschrieben, ist der Aufwand dieser Phase eher klein. Denn der finale Prototyp ist meistens so weit entwickelt, dass neben des GUI auch der größte Teil der zu Grunde liegenden Logik schon besteht und funktioniert. Es wird noch der Rest der Logik entwickelt. Dann ist das Produkt fertig für die Abschluss-Tests.
Wurde ein Mockup-Tool[3] verwendet, kommt jetzt der Zeitpunkt, an dem das Ergebnis des Mockups in Code zu implementieren.
Mit der ersten Variante dauert es zwar länger einen finalen Prototypen zu erstellen, da Softwarelösungen an manchen Stellen aufwendig zu finden sind. Ist der Prototyp aber fertig, ist man auch mit der Implementierung (fast) fertig.
Mit einem Mockup-Tool[4] ist die Erstellung des Prototyps deutlich schneller, da man hier noch keine komplexe Logik entwickeln muss. Für die Entwicklung des GUI kann das Mockup als gute Orientierung dienen, da hier die Abläufe einfach nachvollziehbar sind. Aber die Logik muss noch von Beginn an neu entwickelt werden, was zeitaufwendig ist.
Ist nur ein kurzer Zeitraum zur Verfügung um die oberflächlichen Abläufe und das Design mit einem Prototypen zu erstellen, während der Zeitraum für die Implementierung größer ist, bietet sich die Verwendung eines Mockup-Tools an. Ist es dagegen relevant schnell alle Abläufe und Werte zu besprechen, sollte man den Prototypen direkt mit Code erstellt werden.
Wenn genug Zeit verfügbar ist kann man zwischen beiden Varianten entscheiden.

[3] Siehe Fußnote 1
[4] Siehe Fußnote 1

2.4 Tests

Um die Funktionalität der Software zu prüfen, gibt es mehrere Optionen wie man vorgehen kann.

2.5 Objektorientierte Programmierung

2.6 Clean-Code

3 Parametertabellenkonfigurator

Der "Parametertabellenkonfiguratorïst 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.

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 äusgrauenöder 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"(2.6) für eine erhöhte Les- und Wartbarkeit des Codes und die objektorientierte Programmierung (2.5) 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"[5] 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 GitKraken 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[6] direkt in der Entwicklungsumgebung bearbeitet werden können. Die Verwendung von Git erleichtert auch Software-Reviews[7], 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[8] 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[9] zu erstellen. Dadurch hat man automatisiert alle Dateien, die gebraucht werden, an einem Ort zur Verfügung.

[5] Herausgegeben von der Qt Company
[6] gespeicherter Zwischenstand des Projekts
[7] Feedback zum Code
[8] Dateien in denen mehrere Befehle stehen, die durch Aufruf der Datei in einem Kommandozeilenprogramm hintereinander ausgeführt werden
[9] Dateien, die nach Ausführung der Pipeline heruntergeladen werden

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.

3.4.3 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] Kirandeep Kaur. “Imprecise Software Requirements: A Software Development Risk”. In: *IOSR Journal of Computer Engineering* 7 (Jan. 2012), S. 10–12. DOI: 10.9790/0661-0711012.

[2] Sridhar Nerur, RadhaKanta Mahapatra und George Mangalaraj. “Challenges of Migrating to Agile Methodologies”. In: *Commun. ACM* 48.5 (Mai 2005), S. 72–78. ISSN: 0001-0782. DOI: 10.1145/1060710.1060712. URL: https://doi.org/10.1145/1060710.1060712.