

# Contents

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Softwareentwicklung</b>	<b>3</b>
2.1	Anforderungsanalyse . . . . .	3
2.1.1	Anforderungstypen . . . . .	3
2.2	Prototyping . . . . .	3
2.2.1	GUI-Tests . . . . .	4
2.3	Implementierung . . . . .	4
<b>3</b>	<b>Parametertabellenkonfigurator</b>	<b>6</b>
3.1	Problemstellung . . . . .	6
3.2	Anforderungen . . . . .	7
3.3	Entwurf Benutzeroberfläche . . . . .	7
3.4	Programmierung . . . . .	8
3.4.1	Software-Struktur . . . . .	8
3.4.2	Verarbeiten von Benutzeraktionen . . . . .	8
3.4.3	Herausforderungen und Lösungen . . . . .	8
3.5	Tests . . . . .	8
3.6	geplante Erweiterungen . . . . .	8
3.7	Reflexion . . . . .	8
<b>4</b>	<b>LC-Vision</b>	<b>8</b>

# 1 Einleitung

Die Blum-Novotest GmbH ist im Bereich der Mess- und Prüftechnik tätig. Die Produkte kommen in vielen anspruchsvollen Industrien zum Einsatz, zum Beispiel in der Automobilbranche und der Medizinbranche. An die Produktion werden in diesen Bereichen hohe Ansprüche im Bereich der Präzision gestellt. Um diese Versprechungen und Anforderungen seitens der Kunden einzuhalten und zu erfüllen nutzen international viele Unternehmen die Produkte von Blum, mit Hilfe derer die Präzision der Fertigung geprüft werden kann. Aus den Präzisionsansprüchen an die Kunden-Werkstücke resultieren folglich auch hohe Ansprüche an die Geräte und Software, die in der Qualitätssicherung verwendet werden.

Die Arbeit an einem Projekt/Produkt ist eigentlich nie wirklich vorbei.

Das hat mehrere Gründe: Es werden immer wieder Fehler gefunden, die behoben werden müssen, um die Qualitätsversprechen und Erwartungen einzuhalten. Außerdem kommen immer wieder Anfragen von Kunden, die sofern es wirtschaftlich ist, umgesetzt werden sollten, um die Kunden nicht an andere Hersteller zu verlieren. Und auch den Entwicklern kommen immer wieder neue Ideen für neue Funktionen haben, die das Einsatzgebiet der Produkte erweitern. Aus diesen Gründen wird die Konfiguration von Produkten immer aufwändiger und komplexer. Um dieser Problematik entgegen zu wirken wird ein "Parametertabellenkonfigurator" erstellt.

Der Bereich der Produktpflege und -erweiterung wird durch die Mitarbeit am bestehenden Produkt "LC-Vision" thematisiert.

Beide Projekte dienen dazu den Einstieg in die Abläufe der Softwareentwicklung verständlich und praxisnah kennenzulernen.

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

Für die Entwicklung eines Prototypen müssen wichtige Entscheidungen treffen:

Wie sehr wird der Kunde einbezogen?

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?

### 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 href Mockup Tool<sup>1</sup> 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<sup>2</sup> 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.

## 2.3 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

---

<sup>1</sup>Tool um das GUI zu simulieren

<sup>2</sup>Tool um die Mausbewegungen aufzuzeichnen

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 <sup>3</sup>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 <sup>4</sup>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 hat kann man zwischen beiden Varianten entscheiden.

## 3 Parametertabellenkonfigurator

### 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.

---

<sup>3</sup>Siehe Fußnote 1

<sup>4</sup>Siehe Fußnote 1

## 3.2 Anforderungen

### 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 dass 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 einer Anzeige, um zu sehen welche Parametertabelle ausgewählt ist, ein Button um eine andere Parametertabelle auswählen zu können und ein Feld in dem 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**
  - 3.4.1 Software-Struktur**
  - 3.4.2 Verarbeiten von Benutzeraktionen**
  - 3.4.3 Herausforderungen und Lösungen**
- 3.5 Tests**
- 3.6 geplante Erweiterungen**
- 3.7 Reflexion**
- 4 LC-Vision**