



HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL



Fachhochschule Nordwestschweiz
Hochschule für Gestaltung und Kunst



Masterarbeit

PatternFinder Konzept

von

Thomas Kneubühl
Manuel Schaffner
Markus Oberle
Roland Schild



MAS in Human Computer Interaction Design
Uni Basel, Fachhochschule Nordwestschweiz
und
HSR Hochschule für Technik Rapperswil
Prof. Dr. Lothar Müller

Betreuer:
Christian Hübscher
Thomas Bircher

Februar 2009

Abstract

Patterns sind bewährte Lösungsmuster zu wiederkehrenden Problemstellungen. Diese Lösungsmuster werden in der Softwareentwicklung verbreitet angewendet. Im Bereich User Interface Design (UI Design) existieren ebenfalls viele solcher Lösungsmuster. Die existierenden UI Design Lösungsmuster, sind im Internet weit verstreut und es fehlt an Unterstützung bei der Suche. Das Problem dabei ist, dass Lösungen, die nicht in angemessener Zeit auffindbar sind, keinen Nutzen bringen.

Patterns bringen grosse Vorteile mit sich. Sie verkürzen die Designzeit, vereinfachen die Kommunikation unter den Projektmitarbeitern, dokumentieren Problemstellungen und deren Lösungsansatz und helfen, Missverständnisse zu vermeiden. Sie verhindern Fehlentscheide und erhöhen die Qualität einer Software.

Im Rahmen dieser Masterarbeit wurde ein Konzept entwickelt, welches aufzeigt, wie diese UI Patterns organisiert und aufbereitet werden können, um sie für UI Designer zugänglich zu machen.

Im Rahmen einer Analyse (Kapitel 3) wurden existierende Pattern-Sammlungen zusammengetragen und ausgewertet. Die Struktur und Organisation der Sammlungen wurden untersucht. Es kann bereits vorweg genommen werden, dass keiner der bestehenden Ansätze die Patterns in angemessener Weise zur Verfügung stellt.

Interviews mit UI Designern und Softwareentwicklern gaben Aufschluss darüber, wie beim UI Design vorgegangen wird und wie der Einsatz von Patterns einen Mehrwert bringen kann.

Dieses Konzept ist die Grundlage für eine Applikation, die es erlaubt, schnell und effizient die notwendigen Patterns bei Problemstellungen im Bereich UI Design zu finden. Die involvierten Projektmitarbeiter sollen von dem Wissen und Erfahrungen der HCI Fachspezialisten profitieren können, ohne sich vertieft mit der Materie auseinandersetzen zu müssen.

Der in dieser Masterarbeit beschriebene Ansatz wird T-Modell genannt (Kapitel 4). Dieses Modell sieht vor, das Konzept direkt in die Arbeitsabläufe und Prozesse der Auftraggeberin zu implementieren.

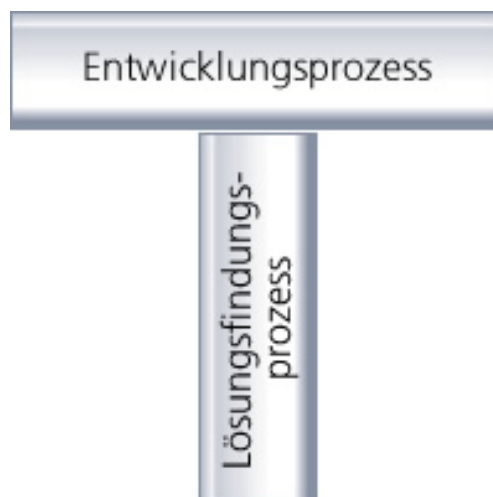


Abbildung 1 – Das T-Modell

Management Summary

Auftraggeberin

Die Masterarbeit entstand im Auftrag der PostFinance (nachfolgend Auftraggeberin genannt) an der Universität Basel, Fachhochschule Nordwestschweiz und HSR Hochschule für Technik in Rapperswil im Rahmen des Lehrgangs Master of Advanced Studies in Human Computer Interaction Design (MAS HCID).

Ausgangslage und Zielsetzung

Die User Interface Designer (UI Designer) unserer Auftraggeberin sind Fachspezialisten sowie Softwareentwickler, welche sich mit Design Aufgaben im Zusammenhang mit Software User Interfaces auseinandersetzen. Zu vielen Design Problemen gibt es inzwischen bewährte Lösungsmuster, welche das UI Design markant vereinfachen. Diese Lösungsmuster, auch Patterns genannt, sind nur schwer auffindbar und werden deshalb kaum eingesetzt.

Das Ziel dieser Masterarbeit ist die Erarbeitung eines Konzepts für die effiziente und effektive Nutzung solcher Patterns durch die UI Designer, ohne vertiefte Kenntnisse im Bereich Human Computer Interaction Design (HCID) zu benötigen.

Analyse

Um zu eruieren, wie die UI Designer in ihrer Arbeit vorgehen, wurde eine umfassende Analyse durchgeführt. Dazu wurden insgesamt sieben Personen interviewt (drei Fachspezialisten, vier Softwareentwickler). Bereits existierende Ansätze, Konzepte und Tools zu dieser Problemstellung wurden untersucht (Kapitel 3).

Resultate

Die Untersuchung hat ergeben, dass die bestehenden Ansätze für die Auftraggeberin ungeeignet sind und ein neuer Ansatz erarbeitet werden muss. Das in dieser Masterarbeit beschriebene Konzept weist folgende Vorteile auf (Kapitel 6):

- Das UI Design wird effizienter und dessen Qualität erhöht sich. Viele Probleme können mit diesen Lösungsmustern behoben werden. Dies führt zu einer einheitlicheren Gestaltung der User Interfaces.
- Patterns werden einfach gefunden und können effizient eingesetzt werden.
- Durch den Einsatz der Patterns verbessert sich die Kommunikation.
- Das Konzept verfolgt einen dynamischen Ansatz und lässt sich beliebig mit weiteren Inhalten erweitern.
- Der Anwender muss kein HCID Spezialist sein. Das benötigte Wissen wird ihm von HCID Spezialisten in Form von Patterns zur Verfügung gestellt.
- Das Konzept lässt sich auf einfache Art und Weise in die Arbeitsabläufe und Prozesse der Auftraggeberin integrieren.

Fazit

Durch die Umsetzung dieses Konzepts kann die Softwareentwicklung vor allem im Bereich des User Interface Designs vereinfacht und die Qualität der Software verbessert werden. Auf das in den Patterns steckende Fachwissen kann mit geringem Aufwand zugegriffen werden.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Kurzbeschreibung	7
1.2	Einführung ins Themengebiet	7
1.2.1	Pattern	8
1.2.2	Patternlanguage	9
1.2.3	Libraries	9
1.3	Aufgabenstellung	9
1.4	Zielpublikum	9
1.5	Abgrenzung	9
1.6	Randbedingungen und Besonderheiten	9
1.7	Geheimhaltung und Rechte	10
1.8	Risiken	10
1.8.1	Verwendbarkeit	10
1.8.2	Akzeptanz	10
1.8.3	Langzeitstudie	10
2	Der Weg zum Pattern Finder – Methodisches Vorgehen	11
2.1	Chronologisches Vorgehen	11
2.1.1	Analyse der Pattern Libraries / Konkurrenzanalyse	11
2.1.2	Analyse der Patternkonzepte / Konkurrenzanalyse	11
2.1.3	Benutzer- und Taskanalyse	11
2.1.4	Early Prototype	11
2.1.5	Testing 1	12
2.1.6	Konzeptionelles Design – Konzeptidee	12
2.1.7	Erster konkreter Prototyp	12
2.1.8	Testing 2	12
2.1.9	Konzeptionelles Design – erste Gruppierung der Inhalte	12
2.1.10	Konzeptionelles Design – Taskflow Entwurf	12
2.1.11	Konzeptionelles Design - Ordnungsstruktur	12
2.1.12	Konzeptionelles Design – zweite Gruppierung der Inhalte	13
2.1.13	Konzeptionelles Design – Taskflow ausarbeiten	13
2.1.14	Zweiter konkreter Prototyp	13
2.1.15	Testing 3	13
2.1.16	Konzeptionelles Design – Variation	13
2.1.17	Dritter konkreter Prototyp	13
2.2	Konzeptionelles Vorgehen	14
3	Gründe für die seltene Nutzung von Patterns – Problemanalyse	15
3.1	Die Analyse der Patternkonzepte - Papers	16
3.1.1	Schwergewicht auf inhaltlicher Struktur und Dokumentation von Pattern	16
3.1.2	Patterns als gemeinsame Sprache	16
3.1.3	Verknüpfung von Patterns ist wichtig	16
3.1.4	Benutzerbezogene Sichten auf Sammlungen	17
3.1.5	Ordnungsversuche nach HCI-Prinzipien	17
3.1.6	Einbettung in Entwicklungsprozess fehlt	18
3.1.7	Pflegeprozess wichtig	18
3.1.8	Kein Pattern-Tool mit Marktreife gefunden	18
3.1.9	Maschinelle Verarbeitung von Patterns ist komplex	18
3.2	Die Analyse der Pattern Libraries	19
3.2.1	Erkannte Positive „Pattern Finder“ Ansätze	19
3.2.2	Erkannte Painpoints	21
3.3	Die Benutzer- und Task-Analyse	24
3.3.1	Personas	24
3.3.2	Szenarios	25
3.3.3	Fazit der Benutzer- und Task-Analyse	26
3.4	Zusammenfassung	27

4	Das T-Modell als Konzept des PatternFinder – Die Lösung_____	28
4.1	Motivation und Nutzen	28
4.2	Das T-Modell Konzept	29
4.2.1	Entwicklungsprozess	29
4.2.2	Lösungsfindungsprozess	30
4.2.3	Top Down Ansatz	31
4.2.4	Bottom Up Ansatz	32
4.2.5	Alternative Navigationsmöglichkeiten	36
4.3	Task Model	37
4.3.1	Primärer Task Flow	38
4.3.2	Sekundäre Task Flow	40
4.4	PatternFinder Prototype	43
4.4.1	Screen Architektur	43
4.4.2	Wireframes	44
4.4.3	Screen Designs	47
4.4.4	Farbkonzept	50
4.4.5	UI Elemente und Umsetzungsdetails	50
4.4.6	Process Phase Navigation	50
4.4.7	Category Selection List	51
4.4.8	Tag List	51
4.4.9	Pattern List	52
4.4.10	Shadowpath Navigation	52
4.4.11	Reference Navigation	53
4.4.12	Pattern Detail	53
4.4.13	Pattern Actions	53
4.4.14	Pattern Tags	53
4.4.15	Prototyp aus Anwendersicht	54
4.4.16	Im Prototyp verwendete Patterns	59
4.5	Pflegeprozess	59
4.6	Einfluss der Findings im Bezug auf Konzept und Prototyp	60
5	Das Konzept im Test - Evaluation _____	63
5.1	Erster Test – Early-Prototyp	63
5.2	Zweiter Test – erster konkreter Prototyp	64
5.3	Dritter Test – zweiter konkreter Prototyp	65
5.4	Vierter Test – Dritter konkreter Prototyp	66
6	Schlussfolgerung _____	67
6.1	Nutzen für das Unternehmen	67
6.2	Empfehlungen für das weitere Vorgehen	68
6.2.1	Usability Test	68
6.2.2	Grafisches Design ausarbeiten	68
6.2.3	Realisierung	68
6.2.4	Inhalte erarbeiten	68
6.2.5	Betriebskonzept	68
7	Unsere Erfahrungen _____	69
8	Verzeichnisse_____	70
8.1	Studenten	70
8.2	Literatur und Quellen	71
8.3	Anhänge	75
8.4	Abbildungen	76
8.5	Glossar	78

1 Einleitung

Bei der Konzeption und Entwicklung von Applikationen gibt es immer wieder gleiche oder ähnliche Problemstellungen. Zu vielen dieser Problemstellungen gibt es bewährte Lösungsmuster. Diese sind allerdings nur schwer auffindbar und werden deshalb selten angewendet. Diese Masterarbeit befasst sich mit folgenden Fragestellungen:

- Wo findet man diese Lösungsmuster?
- Wie werden diese heute verwendet?
- Wie könnten diese Lösungsmuster besser gefunden werden?

1.1 Kurzbeschreibung

Unsere Auftraggeberin, die PostFinance, entwickelt Software, dessen Benutzerschnittstellen mit Storyboards spezifiziert werden.

Projektmitarbeiter, welche User Interface Design (UI Design) Aufgaben wahrnehmen, werden während ihrer Arbeit vor diverse Probleme gestellt. Viele dieser Probleme könnten mit Hilfe von Patterns (Kapitel 1.2) gelöst werden. Patterns sind in verschiedenen Formen und Medien niedergeschrieben. Dazu gehören unter anderem konventionelle Fachliteratur, Internetplattformen und Paper-Abhandlungen.

Die Schwierigkeit des UI Designers ist ein Problem zu erkennen, zu verstehen und eine optimale Lösung zu finden.

Das Ziel dieser Arbeit ist es, ein Konzept zu erarbeiten, welches aufzeigt, wie der UI Designer Antworten zu seinen Problemstellungen während seiner Arbeit findet. Die Motivation für diese Arbeit besteht darin, UI Designern eine praxisorientierte Hilfe bei der Analyse und dem Entwurf (Designentscheiden) von Benutzerschnittstellen zu bieten.

1.2 Einführung ins Themengebiet

Patterns können als „wiederholt bewährtes Lösungsmuster für ein Problem“ beschrieben werden.

A pattern is a three-part rule,
which expresses a relation between
a certain context, a problem, and a solution.

(Alexander, 1977)

Die Idee, Lösungsansätze zu beschreiben stammt von dem 1936 in Wien geborenen Bauarchitekten, Architektur- und System-Theoretiker Christopher Alexander. In seinem Buch „A Pattern Language: Towns, Buildings, Construction“ (Alexander, 1977) beschreibt er unterschiedliche Architektur Aspekte und Strukturen und führt sie logisch zusammen. Mit dem Ansatz, die sonst sehr kreative Disziplin der Architektur in einem strukturierten wieder verwendbaren Nachschlagewerk der Allgemeinheit zur Verfügung zu stellen, hat er neue Wege beschritten.

Erich Gamma übertrug diese Idee zusammen mit der Gang of Four (GoF) auf die Softwareentwicklung (Gamma, Helm, Johnson, & Vlissides, 1994). Solche Software Patterns sind weit verbreitet (z.B. Singleton, Strategy...). Patterns oder ähnliche Lösungsmuster werden heute in vielen Bereichen eingesetzt.

- Software Entwicklung (z.B. Singleton Pattern).
- HCI (z.B. Wizard).
- Chemie (z.B. die Position der Elemente im Periodensystem beschreibt deren Eigenschaften).
- Architektur (z.B. Beach House).
- Musik (z.B. Blues).
- Kochrezepte (z.B. Pasta Gerichte).
- Weitere...

Auch im UI Design wurde nach Rezepten gesucht, die wiederkehrende Aufgaben oder Probleme allgemein lösen können. So wie Gamma im Bereich der Softwareentwicklung, haben sich unter anderem Jennifer Tidwell, Martin van Weelie, Tom Erickson und Jan Borchers Ende der Neunzigerjahre mit dem Einsatz von Patterns im UI Design beschäftigt. Sie ahnten ein grosses Potential in den UI Patterns.

1.2.1 Pattern

Wie bereits erwähnt wird ein Pattern als „wiederholt bewährtes Lösungsmuster für ein Problem“ beschrieben.

Das Rad muss nicht immer neu erfunden werden, sondern es können vorhandene Erfahrungen und Erkenntnisse festgehalten, weitergegeben und genutzt werden. Diese Erkenntnisse können dazu beitragen, das Problem besser zu verstehen. So entsteht eine gemeinsame Sprache zwischen den verschiedenen Projektbeteiligten, um die benötigten Design Entscheide und deren Wirkung zu diskutieren. Durch Patterns wird diese Diskussion losgelöst von rein technischen Argumenten geführt.

Patterns sollen schlussendlich beim Bau von Systemen helfen, bei denen das System und die Benutzer partnerschaftlich miteinander kommunizieren. Pattern können zu Analyse, Design und auch zur Verifikation von Systemen verwendet werden.

Nachfolgend ein Beispiel eines Patterns aus dem Umfeld des UI Designs:

(Quelle: Tidwell J. , Designing Interfaces: Patterns for Effective Interaction Design)

Extras on Demand

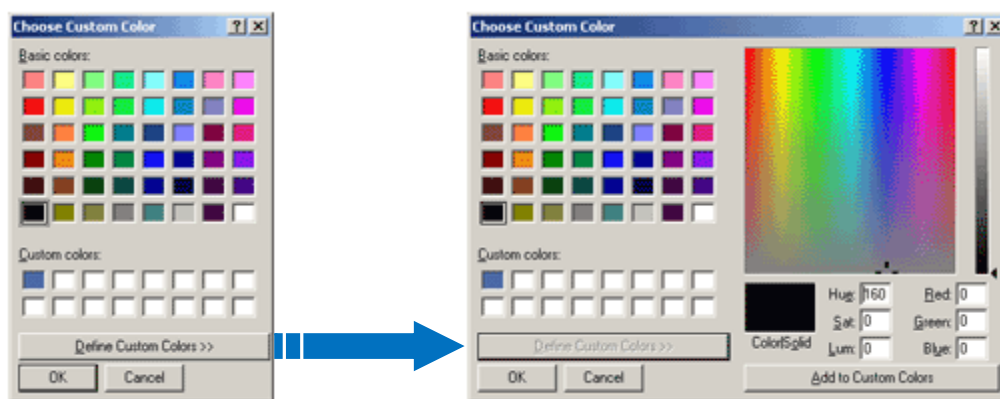


Abbildung 2 - The color dialog box in Windows 2000

- What:** Show the most important content up front, but hide the rest. Let the user reach it via a single, simple gesture.
- Use when:** There's too much stuff to be shown on the page, but some of it isn't very important. You'd rather have a simpler UI, but you have to put all this content somewhere.
- Why:** A simple UI is often better than a complex one, especially for new users, or users who don't need all the functionality you can provide. Let the user choose when to see the entire UI in its full glory -- they're a better judge of that than you are.
- If your design makes 80% of the use cases easy, and the remaining 20% are at least possible (with a little work on the user's part), your UI is doing as well as can be expected! When done correctly, Extras On Demand can save a lot of space on your interface.
- How:** Ruthlessly prune the UI down to its most commonly used, most important items. Put the remainder into their own page or section. Hide that section by default; on the newly-simplified UI, put a clearly-marked button or link to the remainder, such as "More Options." Many UIs use arrows or chevrons, ">>", as part of the link or button label. Others use "...", especially if the button launches a new dialog.
- That section should have another button or other affordance to let the user close it again. Remember, most users won't need it most of the time. Just make sure the entrance to and exit from this "extras" page are obvious.
- In some interfaces, the window literally expands to accommodate the details section, then shrinks down again when the user puts it away. See the Closable Panels pattern for one way to do this. Various desktop UIs provide another mechanism: a dropdown for fill color, for instance, contains a "More Fill Colors..." item that brings up a separate dialog box.

1.2.2 Patternlanguage

A person with a pattern language can design any part of the environment.
He does not need to be an "expert". The expertise is in the language.

(Alexander, 1977)

Wird eine Sammlung von Patterns so strukturiert und verknüpft, dass damit ein Problembereich/Domäne möglichst vollständig beschrieben werden kann, wird diese als Patternsprache (Vokabular oder Patternlanguage) bezeichnet. Eine Patternlanguage dient als gemeinsame, interdisziplinär verständliche Sprache (Lingua Franca) zu einem Problembereich und hilft Erfahrungen, Best Practices und mentale Modelle zu benennen, erinnern und vermitteln.

1.2.3 Libraries

Die Patterns sind üblicherweise in Libraries (Bibliotheken) geordnet. Die Analysen haben ergeben, dass die Libraries verschiedene Ausprägungen aufweisen und deshalb haben wir die Unterteilung in folgende drei Typen vollzogen:

- Pattern Auflistungen: Libraries, die Patterns auflisten – hier wird Wert auf den Inhalt der Patterns gelegt und das Erscheinungsbild der Library rückt in den Hintergrund.
- Pattern Kataloge: Libraries, die Patterns in eine gewisse Struktur bringen, jedoch keine weiterführenden Ansätze haben, um das Finden der Patterns zu vereinfachen.
- Pattern Finders: Libraries, die Ansätze implementiert haben und das Finden der Patterns vereinfachen.

1.3 Aufgabenstellung

Der Wunsch des UI Designers nach Unterstützung bei seiner Arbeit ergibt folgende Ziele für die Masterarbeit:

- Die UI Designer (Personas) und deren Aufgaben sowie Arbeitskontext (Szenarien) sind bekannt.
- Die Probleme beim UI Design sind bekannt und in ihrer Kritikalität gewichtet.
- Eine Konkurrenzanalyse über die heute verfügbaren Hilfsmittel der UI Designer (Fachliteratur, Guidelines, Tools etc.) wird durchgeführt.
- Ein praxistauglicher Weg um Patterns zu finden, ist beschrieben.
- Ein Konzept für einen «PatternFinder», der für die Auftraggeberin später umgesetzt werden soll, ist erstellt.

Mit dem PatternFinder soll künftig durch den vermehrten Einsatz von Patterns die Effizienz sowie die Qualität neuer Software verbessert werden. Design Entscheide könnten anhand von Patterns belegt werden. Zusätzlich helfen Patterns, die Kommunikation unter den Projektbeteiligten zu fördern.

Durch das Umsetzen der Materie können wir unser Wissen festigen und eine real existierende Lücke im Bereich HCI schliessen.

1.4 Zielpublikum

Dieses Konzept richtet sich an Personen, die sich mit UI Design auseinandersetzen, also Projektverantwortliche, Softwareentwickler, UI Designer und HCI Spezialisten. Das Konzept bietet auch weiteren Personen die Möglichkeit, sich einen Einblick in das Themengebiet zu verschaffen.

1.5 Abgrenzung

Die Realisierung der Lösung und das vollständige Ausarbeiten und Zuteilen von Patterns ist nicht Teil dieser Arbeit. Das Resultat ist ein Konzept für den optimalen Zugriff auf Patterns, jedoch nicht für eine Pattern Library (das Tool) selbst.

1.6 Randbedingungen und Besonderheiten

Die Auftraggeberin ermöglicht dem Projektteam den Zugang zu UI Designern und erlaubt, diese in die Arbeit mit einzubeziehen.

1.7 Geheimhaltung und Rechte

Die Ergebnisse sind frei verfügbar und können für Unterrichtszwecke verwendet und zitiert werden. Allfällige Medienaufzeichnungen und Namen von Testpersonen werden nicht ohne Erlaubnis der Auftraggeberin veröffentlicht.

1.8 Risiken

Jedes Projekt birgt Risiken. Schon zu einem frühen Zeitpunkt wurden diese erkannt und Massnahmen getroffen, um diese Risiken zu minimieren.

1.8.1 Verwendbarkeit

Wenn das Konzept nicht realisiert wird, kann ein Tool, welches auf diesem Konzept basiert nie eingesetzt werden. Folgende Punkte können zu dieser Situation führen:

- Das Konzept trifft nicht den Kern der Benutzerbedürfnisse. Bei den Tests dieses Konzeptes wird festgestellt, dass die Benutzer sich nicht zurechtfinden, eine falsche oder überhaupt keine Lösung erhalten oder überfordert sind. Die Folge wäre eine Ablehnung der Realisierung des Konzeptes.
- Das Konzept löst nicht die eigentlichen Probleme aus der Aufgabenstellung.

Während der Entstehung muss das Konzept laufend auf seine Konsistenz und Tauglichkeit geprüft werden. Regelmässige Tests in geeigneter Form liefern zu diversen Zeitpunkten jeweils fortlaufende Resultate und verhindern, dass das Konzept in eine falsche Richtung entwickelt wird.

1.8.2 Akzeptanz

Die Akzeptanz schwindet, wenn das Zielpublikum nicht motiviert werden kann, diese Informationsquelle zu nutzen. Sie sind es sich aufgrund ihrer Erfahrung als Anwender oder als Softwareentwickler gewohnt, frei entscheiden zu können, wie ein UI für einen bestimmten Anwendungsfall auszusehen hat. Die Rahmenbedingungen des Konzeptes schränken die Benutzer in ihrer Kreativität ein.

Die späteren Benutzer müssen bereits bei der Entstehung des Konzeptes so gut als möglich miteinbezogen werden. Damit tragen diese einen wesentlichen Teil zur Entstehung des Konzeptes bei und können ihren Beitrag zur Gestaltung leisten.

1.8.3 Langzeitstudie

Die Arbeit beschränkt sich auf ein Konzept und es findet im Rahmen dieser Masterarbeit keine Realisierung des Tools statt. Dieser Umstand verunmöglicht es, den Nutzen in Form einer Langzeitstudie zu messen.

Dieses Risiko kann nicht minimiert werden. Bei der Konzeption wird jedoch darauf geachtet, dass das Konzept auch realisiert werden kann.

2 Der Weg zum Pattern Finder – Methodisches Vorgehen

Alle Ergebnisse in diesem Dokument wurden anhand angewandter Methoden aus dem HCID Bereich erarbeitet. Dieses Vorgehen erleichtert das Strukturieren und die Nachvollziehbarkeit der Resultate. Dieses Kapitel beschreibt den Einsatz dieser Methoden.

Das methodische Vorgehen kann aus chronologischer und konzeptioneller Sichtweise betrachtet werden. Letzteres zeigt logische Zusammenhänge zwischen einzelnen Methoden und deren Ergebnissen auf.

2.1 Chronologisches Vorgehen

In der nachfolgenden Auflistung werden die angewandten Methoden sowie Tasks in chronologischer Reihenfolge mit den dazugehörigen In- und Outputs abgebildet.

2.1.1 Analyse der Pattern Libraries / Konkurrenzanalyse

Methode: Qualitativ- und quantitative Auswertung
Input: Pattern Libraries aus Fachliteratur und Internet
Output: Analyse der Pattern Libraries (Kapitel 3.2)
Beschreibung: Für die qualitative und quantitative Auswertung der Pattern Libraries wurden Kriterien festgelegt. Bei diesen Kriterien handelt es sich um allfällige Gemeinsamkeiten, Navigations- und Suchelemente, allgemeine konzeptionelle Implementierungen sowie Anzahl und Qualität der Pattern Libraries. Alle Libraries wurden analysiert, die Erkenntnisse in Excel Sheets dokumentiert und danach in einer Datenbank ausgewertet.

2.1.2 Analyse der Patternkonzepte / Konkurrenzanalyse

Methode: Mindmap
Input: Papers aus dem Web
Output: Analyse der Patternkonzept – Papers (Kapitel 3.1)
Beschreibung: Die Papers wurden im Internet recherchiert. Einige dieser Dokumente enthielten Referenzen auf weitere Papers, die auch untersucht wurden. Um die Aktualität der Papers zu überprüfen, wurden diese in zeitlicher Abfolge gebracht. Jedes Paper wurde zusammengefasst. In einem Mindmap wurden alle wichtigen Erkenntnisse zusammengetragen und visualisiert. Einzelne Autoren (Anhang 9) wurden, sofern bekannt, persönlich angeschrieben und nach dem aktuellen Stand ihrer Arbeit befragt.

2.1.3 Benutzer- und Taskanalyse

Methode: Interview
Input: Keine
Output: Personas, Szenarios (Kapitel 3.3)
Beschreibung: In Form von halbstrukturierten Interviews wurde eine Vielfalt an Daten erhoben. Diese Daten wurden anschliessend in Protokollen festgehalten. Aus den Interviewdaten wurden Personas und Szenarios abgeleitet.

2.1.4 Early Prototype

Methode: Prototyping
Input: Patternanalyse
Output: Early Prototype (Anhang 5.1 und 5.2)
Beschreibung: Aus den Erkenntnissen wurden erste triviale Prototypen erstellt. Diese Prototypen wurden entwickelt, um erste Vorstellungen für mögliche Konzepte bildlich festzuhalten. Diese Prototypen basieren nicht auf einem konkreten Konzept, sondern wurden intuitiv entwickelt.

2.1.5 Testing 1

Methode: Expert Review
Input: Early Prototype
Output: Konzeptidee, T-Model (Kapitel 4)
Beschreibung: Die Prototypen wurden jeweils von einem Teammitglied überprüft. Die Ergebnisse wurden anschliessend im Plenum diskutiert. Diese Diskussion ergab eine erste Vorstellung über mögliche Lösungsansätze. Die Prototypen wurden nicht weiter ausgearbeitet. In den nachfolgenden Task wurden neue Prototypen erstellt, in welche Erkenntnisse aus diesen Prototypen flossen.

2.1.6 Konzeptionelles Design – Konzeptidee

Methode: Brainstorming
Input: Analyse Pattern Library, Analyse Patternkonzepte, Early Prototyps
Output: Konzeptidee, T-Model (Kapitel 4)
Beschreibung: In einem Brainstorming wurden Ergebnisse und Ideen zusammengetragen. Aus diesen Inputs entstand eine erste Konzeptidee.

2.1.7 Erster konkreter Prototyp

Methode: Prototyping
Input: Konzeptionelles Design
Output: Papierprototyp (in Visio erstellt) (Anhang 6.1)
Beschreibung: Aus den Erkenntnissen wurde ein erster konkreter Prototyp erstellt. Dieser Prototyp wurde entwickelt, um die Konzeptidee ein erstes Mal zu testen.

2.1.8 Testing 2

Methode: Usability Testing
Input: Erster konkreter Prototyp
Output: Testergebnisse (Anhang 6.2)
Beschreibung: Der Test wurde mit drei Personen, welche den Prototyp zuvor noch nie gesehen haben, durchgeführt. Zwei dieser Personen sind nicht im Projektteam, haben aber alle fundierte HCID Kenntnisse (Studienkollegen).

2.1.9 Konzeptionelles Design – erste Gruppierung der Inhalte

Methode: Card Sorting
Input: Analyse der Pattern Libraries (Kapitel 3.2)
Output: Erste Ordnungsstruktur
Beschreibung: Diese Ergebnisse dienen als Grundlage für die Vertiefung des konzeptionellen Designs. Die Begriffe bzw. Gruppennamen stammen aus dem Input.

2.1.10 Konzeptionelles Design – Taskflow Entwurf

Methode: Konzeptvisualisierung
Input: Analyseergebnisse (Kapitel 3)
Output: Objektübersicht (Kapitel 4)
Beschreibung: Alle bisher zusammengetragenen Fakten wurden auf dem Whiteboard zusammengetragen. Dadurch konnten die Objekte für das Konzept visualisiert und später ein erster Taskflow abgeleitet werden.

2.1.11 Konzeptionelles Design - Ordnungsstruktur

Methode: Brainstorming
Input: Analyse Pattern Library, Papers (Kapitel 3.2 und 3.1)
Output: Sackgasse: Top Down Ansatz (Kapitel 4.2.3)
Beschreibung: In einem Brainstorming wurde versucht, Ordnungsstrukturen zu definieren (Top Down Ansatz) und Patterns darin einzuordnen. Es konnte keine einheitliche und sinnvolle Ordnungsstruktur gefunden werden. Die Zusammenhänge sind nicht praxisnah und zu komplex.

2.1.12 Konzeptionelles Design – zweite Gruppierung der Inhalte

Methode: Card Sorting
Input: Analyse Pattern Library
Output: Inhalt des Prototypen (Kapitel 4.2.4), Idee für Taskflow
Beschreibung: Für den weiteren Verlauf des Projektes wurden Inhalte (konkrete Patterns) definiert. Bei dieser Arbeit entstand die Idee des Bottom Up Ansatzes, bei welchem die Ordnungsstruktur von Charakteristiken aus Patterns abgeleitet wurde.

2.1.13 Konzeptionelles Design – Taskflow ausarbeiten

Methode: Brainstorming
Input: Taskflow, Personas, Szenarios
Output: Taskmodel (Kapitel 4.3)
Beschreibung: Aus den bestehenden Szenarien wurde der Taskflow abgeleitet und auf Flipcharts aufgezeichnet. Dabei wurden weitere Taskflows entdeckt.

2.1.14 Zweiter konkreter Prototyp

Methode: Prototyping
Input: Bisherige Erkenntnisse
Output: Papierprototyp (Anhang 7)
Beschreibung: Aus den Erkenntnissen wurde ein zweiter konkreter Prototyp erstellt.

2.1.15 Testing 3

Methode: Usability Walkthrough
Input: Zweiter konkreter Prototyp
Output: Vertiefte Erkenntnisse
Beschreibung: Diese Testergebnisse lieferten die Grundlage für die Erweiterung des Prototypen.

2.1.16 Konzeptionelles Design – Variation

Methode: Variation und Entscheide
Input: Erkenntnisse aus Testing 3
Output: Optionen für die Verfeinerung des Prototypen
Beschreibung: Einzelne Bereiche im Prototyp wurden mit Hilfe von Variationen ausgearbeitet.

2.1.17 Dritter konkreter Prototyp

Methode: Prototyping
Input: Bisherige Erkenntnisse
Output: Papierprototyp (Anhang 8)
Beschreibung: Aus den Erkenntnissen wurde der aktuelle Prototyp erstellt. Dieser Prototyp bildet das abschliessende Ergebnis dieser Arbeit. Die abschliessenden Tests konnten aus Zeitmangel nicht durchgeführt werden.

2.2 Konzeptionelles Vorgehen

Nachdem im vorhergehenden Kapitel alle Tasks sowie die angewandten Methoden erläutert sind, werden in diesem Kapitel die Zusammenhänge dieser Tasks untereinander illustriert. Die Arbeiten nach den Analysetätigkeiten wurden immer von mehreren Ergebnissen aus verschiedenen Tasks beeinflusst. Durch ein iteratives Vorgehen konnten alle Einflüsse Schritt für Schritt in das Konzept eingebaut werden.

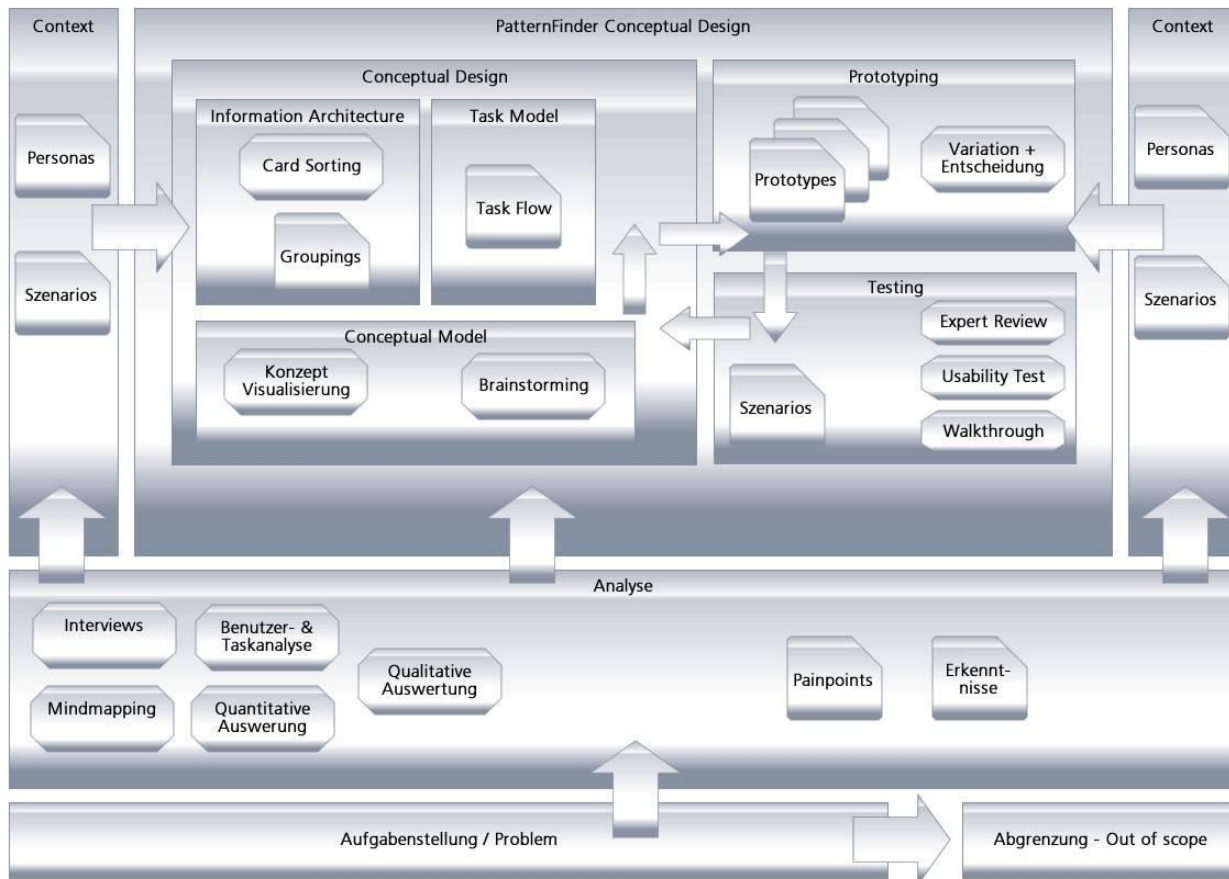


Abbildung 3 – Konzeptionelles Vorgehen

In der Abbildung sind die Methoden sowie die Ergebnisse ersichtlich. Das Fundament bildet die Analyse. Aus der Benutzer- und Taskanalyse sowie den qualitativen- und quantitativen Auswertungen der Pattern Libraries und Konzepten wurden erste Erkenntnisse gewonnen. Auch erste Painpoints konnten entdeckt werden.

Personas und Szenarios wurden aus den Analyseergebnissen abgeleitet. Damit waren die Grundlagen für die Erarbeitung des Konzepts geschaffen.

Die Information Architecture beschreibt die Inhalte des Konzeptes. Diese wurden mittels Card Sorting in logische Kategorien eingeteilt. Diese Kategorien bilden die Filter in der Konzeptidee. Parallel dazu wurde der Taskflow definiert. Bei der Erarbeitung wurde festgestellt, dass es nicht nur einen, sondern mehrere Taskflows gibt (Kapitel 4.3). Aus den Personas wird abgeleitet, dass Lukas von Fach (sekundäre Persona, Kapitel 3.3.1.2) aufgrund seines Wissenstandes und seines Fokus das System auf eine andere Art verwendet als dies Peter S. Coder (primäre Persona, Kapitel 3.3.1.1) tut. Zusammen mit diesen Erkenntnissen wurde ein erstes konzeptionelles Model entwickelt. Dieses wurde in einem ersten Prototyp eingearbeitet.

Natürlich wird ein konzeptionelles Design nicht auf einen Schlag erschaffen. Deshalb wurden mehrere Iterationen benötigt, um das aktuelle Ergebnis zu erreichen. In jeder Iteration wurde der jeweilige Prototyp getestet. Für diese Tests wurde jeweils eine zu diesem Zeitpunkt optimale Testmethode ausgewählt.

3 Gründe für die seltene Nutzung von Patterns – Problemanalyse

Die Problemanalyse besteht aus drei Teilen:

- Existierende Literatur und Veröffentlichungen aus dem akademischen Umfeld wurden gesammelt und bewertet. Es existieren viele Veröffentlichungen über Patterns. Wir sind der Frage nachgegangen, weshalb sich diese Konzepte in der Praxis nie durchsetzen konnten (Kapitel 3.1).
- Existierende Sammlung von Patterns wurden im Internet gesucht, zusammengetragen und qualitativ sowie quantitativ ausgewertet (Kapitel 3.2).
- Die Vorgehensweise der UI Designer bei der Auftraggeberin wurden analysiert und daraus Personas sowie Szenarios erstellt (Kapitel 3.3).

3.1 Die Analyse der Patternkonzepte - Papers

Aus 31 verschiedenen Papers zur Strukturierung und Ordnung von Patterns können nützliche Erkenntnisse für das PatternFinder Konzept entnommen werden. Diese Papers wurden von HCI Experte sowie Software Spezialisten geschrieben.

Die Erkenntnisse werden nachfolgend aufgeführt und bewertet. Die Antworten der Autoren aus den Rückfragen zum aktuellen Stand der Arbeiten sind darin eingearbeitet. Weitere Informationen zu den Erkenntnissen sind im Anhang 9 zu finden. Die konkrete Verwendung im PatternFinder ist in Kapitel 4.6 beschrieben.

3.1.1 Schwergewicht auf inhaltlicher Struktur und Dokumentation von Pattern

Die UCD-Pattern-Community befasste sich bisher schwergewichtig mit den inhaltlichen und visuellen Strukturen und dem Dokumentieren von Patterns. Daher sind auch zahlreiche unterschiedliche Patternsammlungen auf unterschiedlichen Medien in heterogener Qualität und Form vorhanden (Kapitel 3.2). Durch diese Vielfalt sind die vorhandenen Patterns nur schwer zugänglich.

Autoren und Nutzer von Patterns sind nicht dieselben Personen und haben verschiedene mentale Modelle. Die inhaltliche Struktur und die Formulierung sind bezüglich der Verständlichkeit für die Nutzung wichtig. Die meisten Konzepte verwenden dieselben inhaltlichen Grundelemente.

Allerdings wird vorgeschlagen, zusätzlich zur Problemsituation und Lösungssituation auch die gegensätzlichen Kräfte als Problemkomponenten, die Wirkungsweise bezüglich UCD Komponenten oder den Nutzen bezüglich UCD Aspekten mit zu beschreiben und allenfalls die Kontextbeschreibung zu ersetzen.

Die Verständlichkeit der vorhandenen Patterns für den Leser ist nicht das Problem. Es spielt keine Rolle, ob dies nun die gefühlsbetonte Textform nach Alexander oder die übersichtliche Tabellenform nach Welie, Tidwell oder Gamma ist. Jedoch ist es schwierig ein geeignetes Pattern zu finden, wenn man nicht weiss, wonach und wo man suchen soll. Daher steht für das PatternFinder Konzept das praxisbezogene Finden und Nutzen von Pattern im Vordergrund und nicht die Struktur und Dokumentation der Patterns.

(Borchers, 2001), (Erickson, Towards a Pattern Language for Interaction Design , 1997), (Gaffar, Javahery, Seffah, & Sinnig, 2003), (Guy, 2004), (Hughes, 2007), (Kurt & Ebert, 2004), (Lancock, Malone, & Wheeler, 2005), (Mahemoff & Johnston, Principles for a Usability-Oriented Pattern Language, 1998), (Pedro J. Molina, 2002), (Seffah & Javahery, 2002), (van Welie & van der Veer, Pattern Languages in Interaction Design: Structure and Organization , 2003), (Welie, 2000)

3.1.2 Patterns als gemeinsame Sprache

Da Patterns bekannte Situationen (Problem- und Lösungssituation) ansprechend und gut widererkennbar beschreiben (Patternname ist selbstsprechend, PainPoint ist spürbar), dienen Pattern auch als gemeinsame Sprache (LinguaFranca) und helfen ein gemeinsames Verständnis einer Situation, auch ausserhalb des gegebenen Frameworks, klar zu benennen und Erfahrungen auszutauschen. Dies auch im Gegensatz zu Guidelines, welche in sehr spezifischer Form konkrete Oberflächeneigenschaften individuell strukturiert beschreiben und daher nur von Experten einigermaßen gleich verstanden werden. Lassen sich mit Patterns Probleme eines Themenbereichs beschreiben, so spricht man auch von einer mehr oder weniger vollständigen Patternsprache.

(Borchers, 2001), (Erickson, Towards a Pattern Language for Interaction Design , 1997), (Janietz, 2005)

3.1.3 Verknüpfung von Patterns ist wichtig

Eine bi- oder unidirektionale Verknüpfung von Patterns wird als nützlich erachtet, da einerseits weitere abhängige und verknüpfte Patterns gefunden, andererseits auch ein einzelnes Pattern durch dessen Umfeld besser verstanden werden kann. Verknüpfungen sind nicht nur zwischen Patterns, sondern auch mit Metadaten möglich. Mit diesen Metadaten lassen sich die Patterns zusätzlich finden. Die Verknüpfung ist also zentral für das Finden von Pattern. Dabei können die Beziehungen auch typisiert (z.B. has-a, is-a, related-to) oder hierarchisch sein. Die Orientierung in einem solchen Netz kann durch Visualisierung als Karte gesteigert werden. Der PatternFinder versucht durch Verknüpfungen verschiedene Wege zu Pattern zu ermöglichen.

(Borchers, 2001), (Granlund, Lafrenière, & Carr, 2001), (Holger Kampffmeyer, 2007), (Lancock, Malone, & Wheeler, 2005), (van Welie & van der Veer, Pattern Languages in Interaction Design: Structure and Organization , 2003), (van Welie, Interaction Patterns in User Interfaces , 2000)

3.1.4 Benutzerbezogene Sichten auf Sammlungen

Es ist wichtig zu erkennen, ob ein Pattern aus Sicht des Benutzers (Problem oder Aufgabe des Benutzers z.B. schnell die richtige Adresse finden) oder des Entwicklers (Problem oder Aufgabe des Entwicklers z.B. viele Daten übersichtlich darstellen) eine Situation beschreibt. Auch zum Finden von Patterns ist es wichtig, dass diese mit dem Denkmuster und Vokabular des jeweiligen Benutzers geordnet sind. Dies wird durch verschiedene Sichten erreicht. Verschiedene Sichten können auch einem Benutzer helfen, das Problem oder die Lösung aus verschiedenen Blickwinkeln zu betrachten und den Lösungsansatz kreativ (neue Kombinationen) zu überprüfen. Der PatternFinder soll unterschiedlichen Benutzern ermöglichen, mit unterschiedlichen mentalen Modellen und Vokabularen passende Patterns zu finden.

(Fincher & Windsor, Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design , 2000), (Kurt & Ebert, 2004), (Scott, 2005), (Seffah & Javahery, 2002), (van Welie & van der Veer, Pattern Languages in Interaction Design: Structure and Organization , 2003)

3.1.5 Ordnungsversuche nach HCI-Prinzipien

In den Papers werden verschiedene Möglichkeiten diskutiert, wie Patterns in definierte Strukturen zugeordnet werden können. Im Konzept des PatternFinders wird versucht, auf diesen aufzubauen.

Folgende Ordnungsstrukturen werden in den analysierten Papers genannt:

- **Nach den UCD-Komponenten (Benutzer, Aufgabe, Umfeld und System)** - Dies sind die spezifizierbaren Einflusskomponenten eines jeden interaktiven Problemlösungssystems.
(Fincher & Windsor, Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design , 2000), (Greene, Matchen, Jones, Thomas, & Callery, 2003), (van Welie, Interaction Patterns in User Interfaces , 2000)
- **Nach UCD-Aspekten wie den 5E (Effective, Efficient, Engaging, ErrorTolerant, EasyToLearn), den ergonomischen Kriterien oder den kognitiven Dimensionen** - Hier werden die messbaren Wirkungen in den Vordergrund gestellt.
(Fincher, Patterns for HCI and Cognitive Dimensions: two halves of the same story? , 2002), (van Welie, Interaction Patterns in User Interfaces , 2000)
- **Nach durchzuführender Tätigkeit, Aktivität, Absicht oder Ziel (Sicht Endbenutzer oder Entwickler)** - Hier wird das Gewicht auf die Interaktionen, welche zum Finden eines Problems und überführen in einen Lösungszustand erforderlich sind, gelegt. Diese können auch hierarchisch voneinander abhängig sein (z.B. Ziel - Shopapplikation, Erlebnis - einkaufen, Tätigkeit - Einkaufskorb füllen...). Diesen Zielen (..needs to..) können auch Bedingungen (use to..) und diesen dann die Patterns als Lösung zugeordnet werden. Diese Anordnung hilft, erst das Problem (z.B. gegensätzlichen und zu beeinflussenden Kräfte) besser zu verstehen, bevor über mögliche Lösungen nachgedacht wird. So soll ein Pattern nicht über die Lösung, sondern über das zu lösende Problem gefunden werden.
(Borchers, 2001), (Deng, Kemp, & Todd, 2005), (Greene, Matchen, Jones, Thomas, & Callery, 2003), (Guy, 2004), (Holger Kampffmeyer, 2007), (Schmettow, 2005), (Scott, 2005), (van Welie & van der Veer, Pattern Languages in Interaction Design: Structure and Organization , 2003), (van Welie, Interaction Patterns in User Interfaces , 2000)
- **Nach UCD-Schichten oder Projektschritten wie den 5S (Strategy, Scope, Structure, Skeleton, Surface), Baxley (Structure, Behaviour, Presentation) oder modellbasierten Ansätzen (Businessobject-, User-, Task-, Dialog-, Layout-, und Presentation Model)** - Hier wird der schrittweise Aufbau resp. Die Konkretisierung in den Vordergrund gestellt und die Patterns in der benötigten Sequenz geordnet. Die möglichen Lösungen werden so durch die getroffenen Entscheide schrittweise eingegrenzt. Diese Ordnung wurde auch von unseren HCI-Kollegen beim Cardsorting der Patterngruppen gewählt.
(Granlund, Lafrenière, & Carr, 2001), (Pedro J. Molina, 2002), (Radeke, Forbrig, Seffah, & Sinnig, 2006), (Sinnig, Gaffar, Seffah, & Forbrig, 2004)
- **Nach einem Entwicklungsprozess** - siehe nachfolgendes Kapitel 3.1.6

3.1.6 Einbettung in Entwicklungsprozess fehlt

Die beschriebenen Tools haben spezifische Sichtweisen auf einzelne Problembereiche im System Design (z.B. Generieren von Code aus Softwarepatterns) und ihr Einsatz ist nicht im ganzen Entwicklungsprozess möglich. Ein benutzerzentriertes Design soll über den ganzen Entwicklungszyklus (Anforderungsaufnahme / Analysis- und Problemspace: richtiges Problem lösen, Systemdesign / Solutionspace: richtige Lösung finden, Implementierung: richtig lösen...) von Pattern unterstützt werden können. Dabei sind Patterns wie Werkzeuge zur Lösung eines Problems zu verstehen. Nur Methode und Werkzeug zusammen haben einen grossen praktischen Nutzen. Daher basiert der PatternFinder auf der Einbettung in einen Entwicklungsprozess.

(Borchers, 2001), (Fincher & Windsor, Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design , 2000), (Granlund, Lafrenière, & Carr, 2001), (Mahemoff & Johnston, Principles for a Usability-Oriented Pattern Language, 1998), (Pedro J. Molina, 2002), (Sinnig, Gaffar, Seffah, & Forbrig, 2004)

3.1.7 Pflegeprozess wichtig

Die Definition eines transparenten Pflegeprozesses (wie wird ein Pattern in die Sammlung eingefügt, bewertet und an Veränderungen angepasst) mit Validierung und Versionisierung wird als wichtig erachtet, um eine Sammlung mit wachsender Qualität am Leben zu erhalten. Daher wird der Pflegeprozess für den PatternFinder kurz aufgezeigt.

(Borchers, 2001), (Deng, Kemp, & Todd, 2005), (Fincher & Windsor, Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design , 2000), (Gaffar, Javahery, Seffah, & Sinnig, 2003), (Guy, 2004), (Lancock, Malone, & Wheeler, 2005)

3.1.8 Kein Pattern-Tool mit Marktreife gefunden

Die Papers wurden hauptsächlich im Rahmen von Dissertationen oder Konferenzen verfasst und die insgesamt 14 Tools wurden als Proof Of Concept entworfen und sind mangels Ressourcen oder konkreten Anwendungsfällen nicht zur Marktreife gelangt. Keines dieser Tools war für uns verfügbar und uns ist keines bekannt, das im professionellen Einsatz verbreitet ist. Dies ist ein Grund, weshalb Patterns insbesondere fürs UI Design nicht explizit genutzt werden. Einzig Softwarepatterns sind aufgrund der konkreten Vermittlung an Schulen bekannt und werden eingesetzt. Ziel ist es, den PatternFinder zur Marktreife und zur Verbreitung zu führen.

(Deng, Kemp, & Todd, 2005)

3.1.9 Maschinelle Verarbeitung von Patterns ist komplex

Zur maschinenlesbaren Verarbeitung (z.B. Finden mit Hilfe von SemanticWeb oder regelbasierten Ansätzen wie Ontologien) ist eine Volltextsuche ungenügend und ist ein einheitliches, austauschfähiges Datenformat (z.B. das XML-basierte PLML), sowie eine einheitlich strukturierte Sammlung (z.B. Metadatenstruktur oder Abbildung in UML) von Pattern erforderlich. Beides ist heute nur vereinzelt vorhanden resp. eine Konvertierung nur manuell möglich.

(Borchers, 2001), (Deng, Kemp, & Todd, 2005), (Gaffar, Javahery, Seffah, & Sinnig, 2003), (Greene, Matchen, Jones, Thomas, & Callery, 2003), (Holger Kampffmeyer, 2007) (Janietz, 2005), (Scott Henniger, 2003), (Seffah & Javahery, 2002), (van Welie & van der Veer, Pattern Languages in Interaction Design: Structure and Organization , 2003)

Die Einordnung der Pattern kann schnell komplex werden, wenn z.B. die Regeln (Mapping, Parametrisierung, Frage-Antwort, Ziel-Bedingungskarte...) entsprechend erweitert werden müssen, und kann daher nur von Experten durchgeführt werden (Formalismus ist starr). Dazu kommt, dass auch die zu lösenden Probleme maschinenlesbar formuliert (z.B. Model Driven Architecture) werden müssen, um das dazu passende Pattern zu finden oder direkt irgendwelchen Output (Design, Codegerüste...) zu generieren.

Normalerweise werden die Probleme resp. Bedürfnisse sowie das Projektvokabular erst im Laufe der Zeit klar (während dem Lösungsprozess, wenn Lösungsansätze greifbar werden oder durch die Auseinandersetzung oder Diskussion z.B. mit Patterns). Zusätzlich beeinflusst das neue System mit seinen Möglichkeiten wiederum dessen Umgebung. Aus diesen Gründen werden beim PatternFinder nur teilweise automatisierte Ansätze verfolgt.

(Deng, Kemp, & Todd, 2005), (Holger Kampffmeyer, 2007), (Pedro J. Molina, 2002), (Scott, 2005)

3.2 Die Analyse der Pattern Libraries

Es wurden 34 verschiedene UI Pattern Libraries analysiert. Diese sind als Webseiten oder als Bücher verfügbar. Dieses Kapitel beschreibt die Findings aus der Pattern Library Analyse. Die detaillierte Analyse ist im Anhang 4 zu finden.

Es wurden insgesamt 1590 UI Design Patterns zusammengetragen. Die grosse Anzahl gibt einen Eindruck über die Vielfalt der UI Patterns. Um sich darin zurechtzufinden, müssen diese Patterns geordnet werden.

Während der Analyse wurden Ansätze gesammelt, die für das Finden von Patterns förderlich oder hinderlich sind. Es gibt eine Menge guter Ideen, eine umfassende PatternFinder Lösung wurde nicht gefunden.

Nun stellt sich die Frage, wie man die Ansätze verwenden kann, für welche Benutzer sie geeignet sind und wie sie in einem Entwicklungsprozess eingebunden werden können. In diesem Kapitel werden die analysierten Ansätze kurz beschrieben.

3.2.1 Erkannte Positive „Pattern Finder“ Ansätze

3.2.1.1 Topic Map

Die Topic Map bietet eine übersichtliche Darstellung der Kategorien. Jede Kategorie ist in einem Satz beschrieben und die dazugehörigen Patterns sind aufgeführt. Die Beschreibung der Kategorienamen und der Beispiele helfen ein besseres Bild darüber zu bekommen, welche Art von Pattern sich dahinter versteckt.



Abbildung 4 - Topic Map
(Timmesfeld, Schmitt, Vucica, & Schreiner, 2005)

3.2.1.2 Tags und Finden über Tags

Durch Tags kann man Patterns mit mehreren Begriffen beschreiben. Tags können auch von einer Community vergeben und verwaltet werden.

Ein Vorteil der Tags ist, dass die Attribute und Titel der Patterns nicht bekannt sein müssen. Die Metabeschreibung der Patterns mit Tags hilft beim Finden.

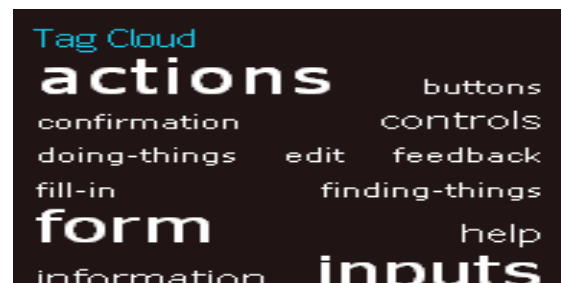


Abbildung 5 – Tags
(Lammi, 2008)

3.2.1.3 Ansprechende Kurzübersicht

Eine Kurzbeschreibung des Patterns verbunden mit einem Bild, hilft bei der visuellen Orientierung.

Die Kurzbeschreibung hilft, mehrdeutige oder unbekannte Labels zu klassifizieren und zu erfassen.



Abbildung 6 – Kurzübersicht
(Toxboe)

3.2.1.4 Ranking nach Patternbewertung

Bei der Auswahl mehrerer Patterns im gleichen Themenbereich (z.B. Navigation) kann eine Bewertung oder Kommentar helfen. Patterns können durch eine Community bewertet und das Vertrauen in ein Pattern gestärkt werden.

Die Bewertungen können bei der Sortierung von Suchresultaten der Patterns einbezogen werden.

Die Validierbarkeit und Korrektheit wird bei Patterns immer wieder in Frage gestellt. Die Gefahr besteht, dass die Bewertungen sehr auf der Meinung des UI Designers beruht und User Centered Design Aspekte nicht berücksichtigt werden.

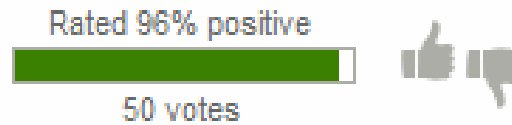


Abbildung 7 – Ranking nach Patternbewertung (Toxboe)

3.2.1.5 Interaktive Map

Eine interaktive Map führt den Benutzer durch das hierarchisch strukturierte Pattern. Die Idee, eine Struktur grafisch zu visualisieren, ist nicht neu. Der Vorteil besteht hauptsächlich darin, verwandte Patterns einfacher zu finden. Jedoch müssen die Patternbezeichnungen sehr sprechend sein.

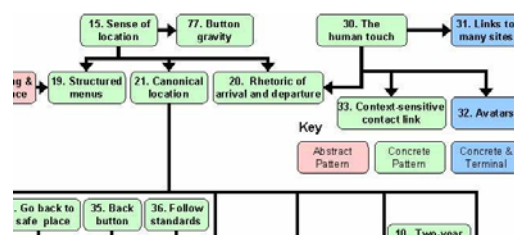


Abbildung 8 - Hierarchische Darstellung (Graham, 2002)

3.2.1.6 Globale Navigation

Damit jederzeit auf ein Pattern oder eine Kategorie zugegriffen werden kann, ist die Globale Navigation hilfreich. Wenn unklar ist, wonach man sucht, ist die globale Navigation ein Nachteil.



Abbildung 9 - Globale Navigation (Folmer)

3.2.1.7 Guide

Ein Guide führt den Benutzer mit vordefinierten Fragen zu einer Lösungsmenge. Dieser funktioniert ähnlich wie ein Filter.

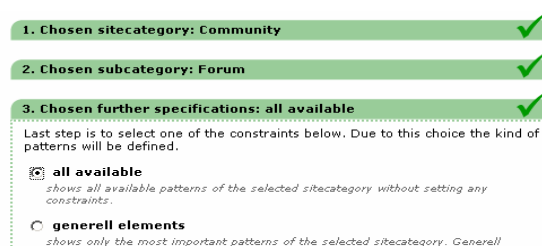


Abbildung 10 - Guide (Timmesfeld, Schmitt, Vucica, & Schreiner, 2005)

3.2.1.8 Pattern nach Aufgaben sortieren

Statt die schwierig zu findenden Kategoriennamen zu wählen, gibt es auch den Ansatz, die Patterns nach Aufgaben zu kategorisieren (Morville & Rosenfeld, 2006). Dabei handelt es sich um Aufgaben, die das Pattern erfüllen sollte (z.B. Eingabe vom Benutzer einholen, eine Auswahl machen) oder Aufgaben des UI Designers, die durch das Pattern unterstützt werden (z.B. Font wählen, Bildschirminhalt anordnen, Widgets auswählen, Struktur planen). Siehe auch Kapitel 3.1.5.

3.2.2 Erkannte Painpoints

3.2.2.1 Einstieg verwirrend

Beim Einstieg erkennt man nicht immer auf Anhieb, um was es geht. Weder textuell noch visuell ist auf erkennbar, dass es sich um eine Pattern Sammlung handelt. Viel Text als Einleitung und kein offensichtlicher Einstieg in die Patterns.

3.2.2.2 Schlechtes Labeling

Die Analyse hat gezeigt, dass Bezeichnungen des Patterns und auch der Gruppierungen ein inkonsistentes Erscheinungsbild haben und schwer verständlich sind. Für viele Bezeichnungen müssen die Benutzer gute Englischkenntnisse oder ein Designerglossar zur Verfügung haben.

Bei den meisten Pattern Libraries ist kein einheitliches Konzept fürs Labeling erkennbar.

Die Organisation nach Themen kann mehrdeutig sein und damit zu Problemen führen (Morville & Rosenfeld, 2006).

3.2.2.3 Schlechte Suchunterstützung

Die Mehrzahl der Library Anbieter bieten keine Suche an, sei es, weil manche nur wenig Inhalt haben oder weil niemand den Suchmechanismen Interesse geschenkt hat.

Suchmechanismen, die bei gängigen CMS oder Wikis implementiert sind, reichen nicht aus und es fehlen gute Suchalgorithmen. Hilfestellung zur Verfeinerung der Suche fehlen oder die Resultatmenge wird unübersichtlich dargestellt.

User needs

Patterns that meet a direct need of the user.

Navigating around

- [Accordion](#)
- [Headerless Menu](#)
- [Breadcrumbs](#)
- [Discreet Navigation](#)

Searching

- [Advanced Search](#)
- [Autocomplete](#)
- [Frequently Asked Q](#)

Abbildung 11 – Pattern nach Aufgaben sortieren
(van Welie, A Pattern Library for Interaction Design)



Abbildung 12 – Einstieg verwirrend
(Graham, 2002)



Abbildung 13 – Schlechtes Labeling
(Guy, 2004)

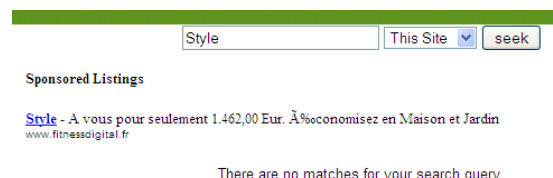


Abbildung 14 – Schlechte Suchunterstützung
(Bolchini & Community)

3.2.2.4 Technische Plattformen

Die Inhalte werden stark an die technischen Gegebenheiten angepasst. Oft fehlt die Möglichkeit, zusätzliche Navigationshilfen und Filter zu integrieren.

Wikis sind schnell installiert und es wird auch ein Community Gedanke unterstützt. Sie sind aber nicht dazu geeignet, eine gute Auffindbarkeit zu garantieren. Die Struktur wächst unkontrollierbar und Standardsuchen sind nicht genügend gut implementiert.



Abbildung 15 – Technische Plattformen (Stimmt.AG)

3.2.2.5 Falsche Hoffnungen

Durch ein gutes Design könne falsche Hoffnungen geweckt werden, sofern sich dahinter schlechte oder unvollständige Inhalte verbergen.

Es gibt auch den umgekehrten Fall einer Library mit vielen Pattern, die ein schlechtes Interface Design und schlechte Grafiken aufweisen und die Verständlichkeit der Patterns dadurch beeinträchtigen.

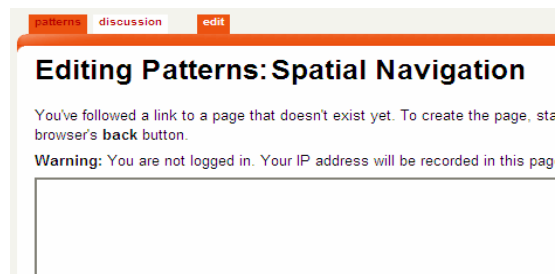


Abbildung 16 – Falsche Hoffnungen
Nicht beschriebenes Pattern im Wiki (Community)

3.2.2.6 One Way Navigation

Mit dem Back Button ist kein Zurück (Browser) möglich. Der Weg führt nur über die Einstiegsseite zu anderen Inhalten.

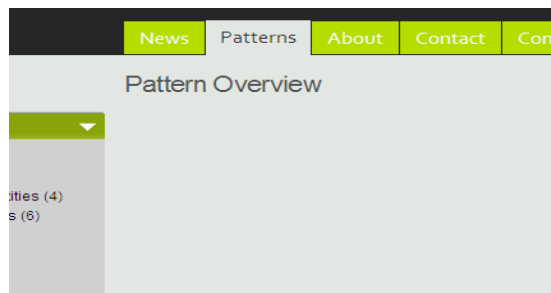


Abbildung 17 – One Way Navigation -
Leere View, der Weg zurück funktioniert nicht. (Behrens, 2008)

3.2.2.7 Code-Fehlermeldungen

Eine fehlerhafte Implementation ergibt ein unprofessionelles Bild der Library. Die Glaubwürdigkeit an den Inhalt leidet.



Abbildung 18 – Fehlermeldungen
(Timmesfeld, Schmitt, Vucica, & Schreiner, 2005)

3.2.2.8 Pattern Inhalte sind nicht vertrauenswürdig

Viele Libraries sind nicht aktualisiert, die Links und Bilder sind veraltet oder die Seiten nicht fertig gestellt und mit tbd (to be done) versehen.

Dies lässt das Vertrauen in den Inhalt der Library schwinden. Auch Libraries mit zu wenigen Patterns, können als unfertig und nicht brauchbar wahrgenommen werden.

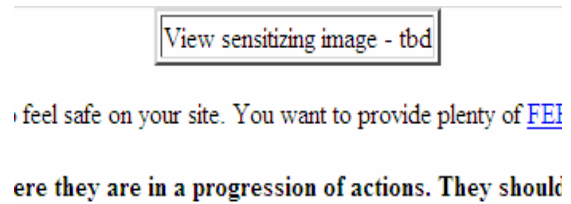


Abbildung 19 – Pattern Inhalte nicht vertrauenswürdig
(Graham, 2002)

3.3 Die Benutzer- und Task-Analyse

Bereits zu einem frühen Zeitpunkt wurde eine Benutzeranalyse mit Interviews durchgeführt. Der Interviewbogen wurde bewusst offen strukturiert, um möglichst vielseitige Informationen zusammenzutragen. Die Daten stammen aus sieben Interviews. Die interviewten Personen sind drei Fachspezialisten sowie vier Softwareentwickler. Alle beschäftigen sich mit User Interface Design, allerdings mit unterschiedlichen Ausprägungen.

Nach dem 30-minütigen Interview konnten in einem offenen Gespräch weitere wertvolle Informationen gesammelt werden.

Die Ergebnisse wurden analysiert und in Personas und Szenarios abgeleitet.

3.3.1 Personas

Aus den Interview-Daten konnten zwei Personas abgeleitet werden.

3.3.1.1 Primäre Persona



Peter S. Coder

Peter ist der Meinung, dass zwischen Fach und IT viele Unklarheiten entstehen, da es kein gemeinsames Vokabular gibt.

Das UI Design ist Bestandteil seiner täglichen Arbeit. Designentscheide fällt er aufgrund seiner langjährigen Erfahrung als Softwareentwickler intuitiv.

Peter kennt sich mit Softwarepattern (z.B. GOF) aus und setzt diese bei seiner täglichen Arbeit ein. Von UI Patterns hat er nie gehört, stellt sich aber vor, dass diese auf Widgetebene basieren.

Bei konkreten Problemstellungen sucht er in Fachliteratur nach Lösungen. Er kennt vielseitige Lösungswege zu den Themen Softwarearchitektur und -entwicklung von Konferenzbesuchen.

Peter kennt die technischen Möglichkeiten von UI Elementen genau und nervt sich, wenn diese ungenau spezifiziert sind.

Er ist der Meinung, dass die Kommunikation zwischen Fach und IT mit dem Einsatz von Patterns in den Spezifikationen verbessert werden könnte. Es darf ihn aber nicht in seiner Kreativität einschränken.

Abbildung 20 - Persona Peter S.Coder

3.3.1.2 Sekundäre Persona



Lukas von Fach

Lukas erstellt seit kurzem UI Designs und diese mit gesundem Menschenverstand. Er orientiert sich an Software, die er schon kennt und übernimmt daraus Ideen für sein Design. Als ehemaliger Poweruser kennt er das Fach und dessen Bedürfnisse genau.

Die technischen Bezeichnungen der UI Elemente kennt er nur teilweise. Der Unterschied zwischen einem Radio Button und einer Checkbox sind ihm nur teilweise bekannt.

Im Vordergrund steht viel mehr die Präsentation der Businesslogik. Patterns sind für ihn dasselbe wie UI Elemente.

Informationen zu Problemlösungen sucht Lukas gezielt in Fachliteratur. Literatur zu UI Design ist ihm jedoch fremd. Er wäre motiviert, UI Patterns einzusetzen, hat jedoch Angst, dass seine Arbeit durch zusätzliche Vorgaben aufwendiger wird.

Abbildung 21 - Persona Lukas von Fach

3.3.2 Szenarios

Die UI Design Arbeiten werden hauptsächlich während der Spezifikationsphase durchgeführt. Weitere Anpassungen werden jedoch auch später im Projekt, z.B. während der Realisierung vorgenommen.

Die UI Design Arbeiten sind im Entwicklungsprozess unserer Auftragsgeberin nicht verankert, deshalb können die einzelnen Tätigkeiten nicht direkt in einen Prozess eingeordnet und abgebildet werden. Zwei Szenarios verdeutlichen die Vorgehensweise.

3.3.2.1 UI Design durch Peter S. Coder (Primäre Persona)

Die Anforderungen für das neue System werden von der Fachabteilung in Textform definiert und an Peter S. Coder übergeben. Peter S. Coder studiert diese Anforderungen und versucht sich ein Bild zu machen, wie der Endanwender dieses System oder vor allem das User Interface nutzen möchte.

Mittels Papierprototypen, welche er selbständig entwirft, bespricht er die Machbarkeit mit dem Fach noch bevor er mit der Umsetzung beginnt.

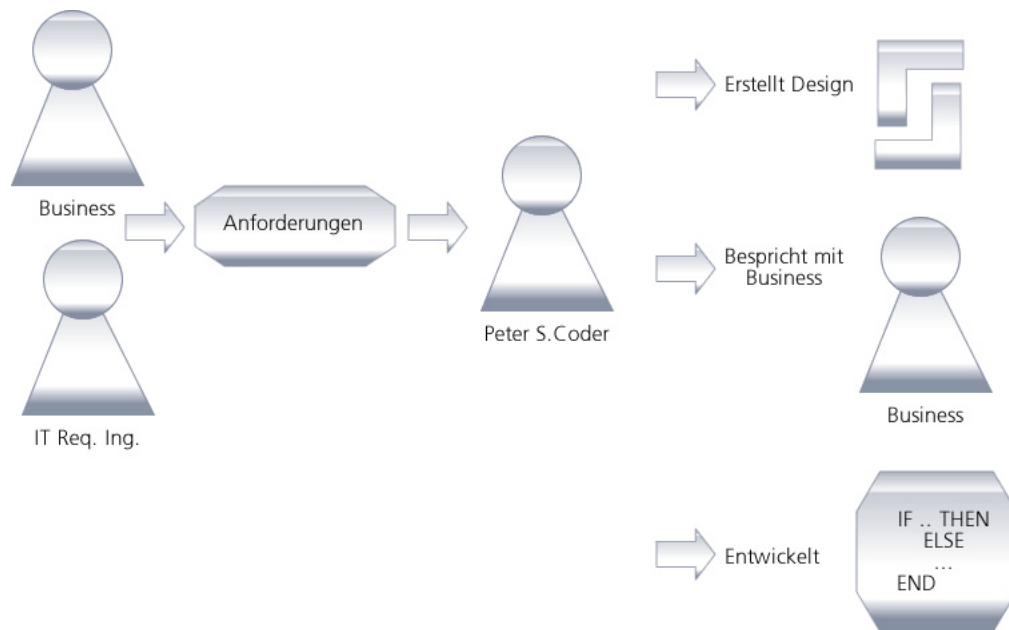


Abbildung 22 - Szenario 1

Mit diesen Papierprototypen demonstriert Peter S. Coder neben den UI Elementen auch die Screen Abfolge und die Interaktionselemente. Bei seinen Design Vorschlägen achtet Peter S. Coder auf Einfachheit. Von den zwei bis drei Vorschlägen präsentiert er jeweils nur einen. Bei diesem ist die technische Machbarkeit bereits garantiert.

Nach erfolgreicher Rücksprache mit dem Fach macht er sich an die Realisierung der Ergebnisse. Während der Realisierungsphase finden weitere Besprechungen der Zwischenergebnisse mit dem Fach statt.

3.3.2.2 UI Design durch Lukas von Fach (Sekundäre Persona)

Lukas von Fach erhält die fachlichen Anforderungen in Textform geliefert. Da er früher diese Systeme selbst als Anwender bedient hat, versteht er diese genau und weiss, wie er diese in einem UI umsetzen muss. Als Vorlagen dienen Visio Zeichnungen, welche ihm viel Arbeit abnehmen. Nachdem Lukas von Fach die erste Version seines UI Entwurfs fertig gestellt hat, bespricht er diese mit den IT-Spezialisten, um die technischen Machbarkeit sicherzustellen.

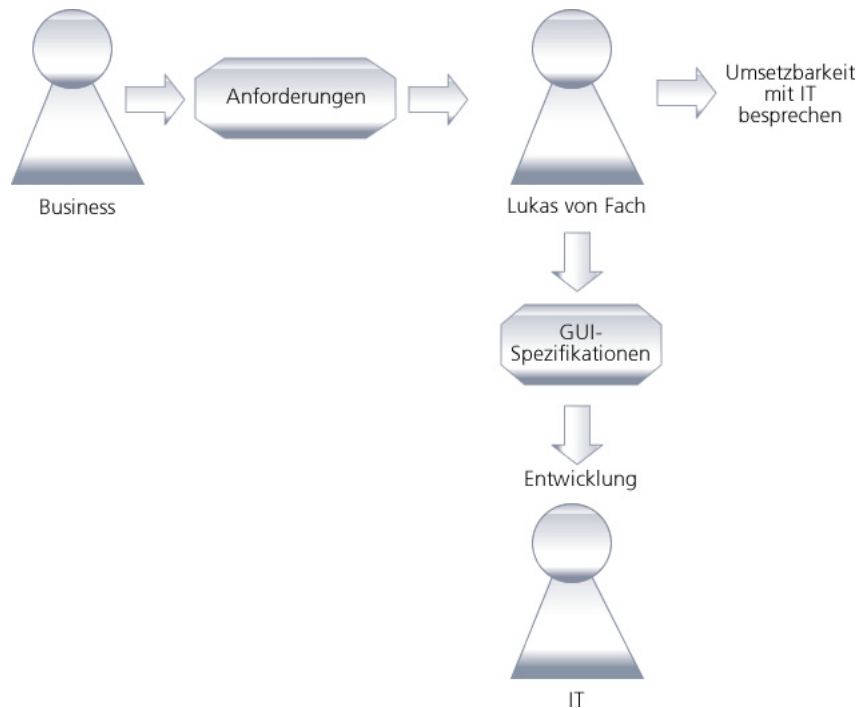


Abbildung 23 - Szenario 2

Nachdem er die Korrekturen aus den Besprechungen in das UI Design integriert hat, startet Lukas von Fach eine weitere Überprüfung. Dabei achtet er vor allem auf das korrekte Wording, sowie die Vollständigkeit der einzelnen Attribute.

3.3.3 Fazit der Benutzer- und Task-Analyse

Grundsätzlich geht aus der Benutzer- und Task-Analyse hervor, dass es bei unserer Auftraggeberin nicht „den“ UI Designer gibt. Die UI Design Arbeiten werden nicht von ausgebildeten Spezialisten gemacht, sind reine Erfahrungssache und werden durch Bauchentscheide unterstützt. Die Disziplin UI Design ist nicht im Entwicklungsprozess verankert.

Die Auswertung der Interviewdaten haben ergeben, dass es im Bereich des UI Designs ein grosses Verbesserungspotential gibt. Folgende Probleme wurden erkannt:

- **Patterns sind unbekannt** - Software Patterns werden in der Softwareentwicklung eingesetzt, UI Patterns sind jedoch völlig unbekannt. Für die beschriebenen Personas sind Patterns ein (teilweise) unbekannter Begriff und deren Vorteile werden nicht genutzt.
- **Schlechte Design Entscheide** - Die technischen Möglichkeiten bilden die alleinige Grundlage für das Erstellen eines UI. Die einfachste und schnellste Variante wird bevorzugt. Es werden viele Bauchentscheide getroffen. Die Einführung eines konzeptionellen Vorgehens würde die Qualität, sowie die Effizienz verbessern.
- **Kein benutzerzentrierter Ansatz** - Der Endanwender wird bei Design Entscheiden ignoriert. Peter S. Coder und Lukas von Fach meinen, sie kennen die Benutzer, erstellen jedoch das Design aus ihrer individuellen Sicht. Ansätze aus anderen Applikationen werden bedenkenlos kopiert, auch wenn diese in einem komplett anderen Zusammenhang eingesetzt werden.
- **Kein UI Design Prozess** - Die UI Entwicklung ist für Lukas von Fach eine komplizierte Angelegenheit. Ein strukturiertes Vorgehen für das erstellen von UI's hat er nie gelernt. Dass die Businessobjekte sowie die Businesslogik auf dem UI erscheinen, hat für ihn den höchsten Stellenwert.

- **Fehlende Ausbildung** - Sämtliche im UI Design involvierten Personen sind nicht oder nur teilweise ausgebildet. Dieser Nachteil kann mit dem Einsatz von Patterns teilweise behoben werden.
- **Kommunikationsprobleme** - Zwischen den Projektmitarbeitern bestehen Kommunikationsprobleme. Peter S. Coder meint, es fehle ein gemeinsames Vokabular. Patterns könnten in einer Spezifikation als Referenz eingesetzt werden. Dies trägt zu einer besseren Kommunikation bei.

3.4 Zusammenfassung

UI Patterns sind noch sehr unbekannt und schwer zugänglich. Deshalb werden Patterns bei unserer Auftraggeberin kaum oder überhaupt nicht eingesetzt.

UI Design ist bei der Auftraggeberin nicht Bestandteil des Entwicklungsprozesses. Der benutzerzentrierte Ansatz muss integriert werden, damit das UI Design den nötigen Stellenwert bekommt. Der Entwicklungsprozess kann nicht von heute auf morgen umgestellt werden. Es ist wichtig, dass der UI Designer bei seiner Arbeit während des Entwicklungsprozesses unterstützt wird.

Das Hilfsmittel sind Patterns. Da diese mit wenig Aufwand eine grosse Anzahl Probleme lösen, besteht eine reelle Chance, dass diese Verwendung finden und akzeptiert werden.

Durch die Verwendung von Patterns werden UI Designer bei ihren Entscheidungen unterstützt. Damit werden die UI Design Arbeiten teilweise standardisiert.

Die Hilfestellung soll sich keinesfalls nur auf Patterns beschränken, vielmehr können auch Templates, Methoden, Guidelines, sowie weiterführende Informationen wie z.B. eine Testanleitung zur Verifikation des Designs angeboten werden.

4 Das T-Modell als Konzept des PatternFinder – Die Lösung

4.1 Motivation und Nutzen

Die während des UI Designs eines SW-Entwicklungsprozesses auftretenden Probleme sind wiederkehrend. Hier drei Beispiele:

- Das Plazieren von mehreren Buttons.
- Einer ganzen Applikation ein bestechendes Look & Feel zu geben.
- Das Layout eines Formulars zu gestalten.

Mehr oder weniger gute Lösungen zu diesen Problemen wurden bisher individuell entwickelt. Lösungsansätze zu diesen Problemen sind aber bereits mehrheitlich in bestehenden Pattern Libraries dokumentiert. Zu den oben genannten Aufgaben bestehen folgende Patterns:

- Button Groups.
- Corner Treatements.
- Right/left alignment.

Um in den zahlreichen Pattern Libraries mit über 1'500 Patterns die richtige Lösung in einer vertretbaren Zeit zu finden, braucht es gute Kenntnisse der einzelnen Patterns oder zumindest der Pattern Libraries.

Nachfolgend sind einige Gründe aufgelistet, welche die Schwierigkeiten aufzeigen, weshalb Patterns schlecht zu finden oder anzuwenden sind (Kapitel 3.2.2):

- Die Such- und Navigationsmöglichkeiten der Pattern Libraries sind unzureichend.
- Mit einem guten Design aber unvollständigen Inhalten schwindet die Motivation für den Gebrauch dieser Pattern Libraries.
- Durch eine unklare Einstiegsseite werden Pattern Libraries nicht als solche erkannt.
- Das Labeling (Überschriften, Namen, etc.) ist nicht auf die jeweiligen Benutzer zugeschnitten.
- Die Qualität der technischen Umsetzung der Pattern Libraries ist unzureichend (viele Fehler).

Ein weiterer Grund ist die fehlende Einbettung der UI Design Disziplinen in den verwendeten Entwicklungsprozess. Das heisst, die einzelnen Aufgaben werden nicht strukturiert angegangen, was den Einsatz von Patterns zusätzlich erschwert.

Die Motivation für Verbesserungen besteht darin, die bekannten Lösungsansätze (Patterns) für ein breiteres Publikum zugänglich zu machen, das heisst, auch für Benutzer ohne spezifische UI Design Kenntnisse. Das Potential, welches sich hinter diesen Patterns befindet ist gross.

Durch ein effizientes Finden und das Wiederverwenden von Lösungsansätzen wird der Entwicklungsprozess verkürzt und die Qualität des Endproduktes gesteigert. Patterns helfen zu kommunizieren, dokumentieren und standardisieren. Sie verhindern Fehlentscheide im UI Design und verringern Missverständnisse.

In diesem Kapitel wird immer wieder Bezug auf das Kapitel 4.6 genommen. Darin werden die Einflüsse der Findings im Bezug auf das Konzept und den Prototypen beschrieben.

4.2 Das T-Modell Konzept

Die Konzeptidee des PatternFinders resultiert aus den folgenden zwei Komponenten:

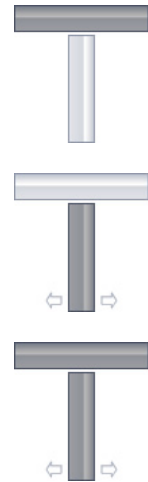
- **SW-Entwicklungsprozess** - Ein Entwicklungsprozess unterstützt den Benutzer dabei, das UI Design in einem strukturierten Prozess durchzuführen und die Menge der Patterns dadurch einzuschränken.
- **Lösungsfindungsprozess** - Basiert auf der Idee der Tags und der individuellen Kategorienbildung. Der Benutzer wird durch vertraute Begriffe aus seinem Vokabular zur Lösung geführt.

Das konzeptionelle Modell basiert auf diesen beiden Komponenten und wird wie folgt visualisiert:

Die Komponente des Entwicklungsprozess (Kapitel 4.2.1): Sie stellt die horizontale Projekt-Zeitachse des Modells dar. (Findings Kapitel 4.6.1.6 / 4.6.1.8)

Der Lösungsfindungsprozess (Kapitel 4.2.2) ist in der vertikalen Achse abgebildet. Diese vertikale Achse wird je nach Phase in der horizontalen Zeitachse (Entwicklungsprozess Phase) von links nach rechts verschoben. Je nach horizontaler Position ist die Menge und Art der möglichen Lösungen unterschiedlich.

Das Zusammenführen dieser beiden Dimensionen stellt die Basis des T-Modells dar. Das Konzept Modell ist ein grosses T mit einer verschiebbaren vertikalen Achse.



4.2.1 Entwicklungsprozess

Je nach Firmenkultur werden unterschiedliche Entwicklungsprozesse eingesetzt. Jeder Prozess besteht aus verschiedenen Phasen. Im Kapitel □ wurde erwähnt, dass eine Anbindung der Patterns an einen Entwicklungsprozesse und deren Phasen Vorteile mit sich bringt. Dadurch werden Aspekte des UCD in diesem Prozess unterstützt ohne ihn zu verändern und eine erste Gruppierung der Patterns erreicht.



Abbildung 24 – Iterativer Entwicklungsprozess

Drei Beispiele bekannter Entwicklungsprozesse:

- **OOSE Object-Oriented Software Engineering**



Abbildung 25 – OOSE

- **RUP Rational Unified Process**



Abbildung 26 - RUP

- **Hermes**



Abbildung 27 - Hermes

Im Lehrgang MAS HCID wurde der UCD Ansatz „The Usability Engineering Lifecycle“ (Mayhew, 2008) eingesetzt. Dieser Ansatz wird durch den übergreifenden Entwicklungsprozess OOSE optimal integriert. Deshalb wird in diesem Konzept der OOSE Prozess stellvertretend für andere Entwicklungsprozesse eingesetzt.



Zeit →

Abbildung 28 - Usability Engineering Lifecycle

Je nach Phase im Entwicklungsprozess treten unterschiedliche Designprobleme auf verschiedenen Ebenen auf. Es macht Sinn, diese bereits auf die einzelnen Phasen einzuschränken.

4.2.2 Lösungsfindungsprozess

Die grosse Anzahl der Patterns kann durch die Unterteilung in die unterschiedlichen Prozessphasen bereits eingeschränkt werden.

Die Restmenge der möglichen Lösungsansätze ist unter Umständen noch nicht überschaubar. Das heisst, dass ein zusätzliches Verfahren zur Lösungsfindung benötigt wird.

Die Schwierigkeit ist nun, die einzelnen Patterns für den Benutzer verständlich zu kategorisieren. Mit der Anzahl möglicher Patterns steigt auch der Anspruch an die Qualität der Kategorisierung

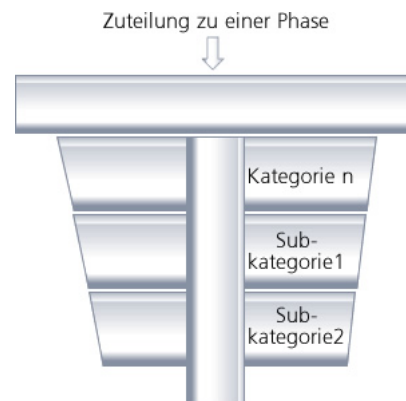


Abbildung 29 - Lösungsfindungsprozess

Die Einschränkung der Lösungsmenge wird in sämtlichen analysierten Pattern Libraries über die Selektion von Kategorien vollzogen. Nachfolgend sind drei frei ausgewählte Beispiele von Kategoriennamen aufgelistet:

- User Needs.
- Application Needs.
- Context of design.

Eine strukturierte Methode zu dieser Kategorisierung ist wichtig. Dazu wurden zwei Ansätze verwendet:

- Top Down.
- Bottom Up.

4.2.3 Top Down Ansatz

In den untersuchten Pattern Papers erfolgt die Kategorisierung (Kapitel 3.1.5) über einen Top Down Ansatz. Das heisst, es werden zuerst Kategorien gebildet und anschliessend die Patterns diesen Kategorien zugeordnet. Dieser Umstand führt zu Kategoriebezeichnungen, welche Autorenspezifisch und nicht auf den Benutzer zugeschnitten sind.

Bei unserem Versuch des Top Down Ansatzes wurden mittels Brain Storming eigene Kategorien gebildet. Anschliessend wurden einige dieser Kategorien zu einer Gruppe zusammengefasst. Die Begriffe entstanden einerseits aus dem Fachbereich HCID und dem allgemeinen IT Umfeld. Nachfolgend ein paar Beispiele dieser Kategorien:

- Feeling: Metapher, Moodboard, Bilder, Farben
- UI Layer: Baxley, 5S, ISO
- PatternFinder User: Personas, Business View, Profistufe
- Usability: 5E, ergonomische Kriterien

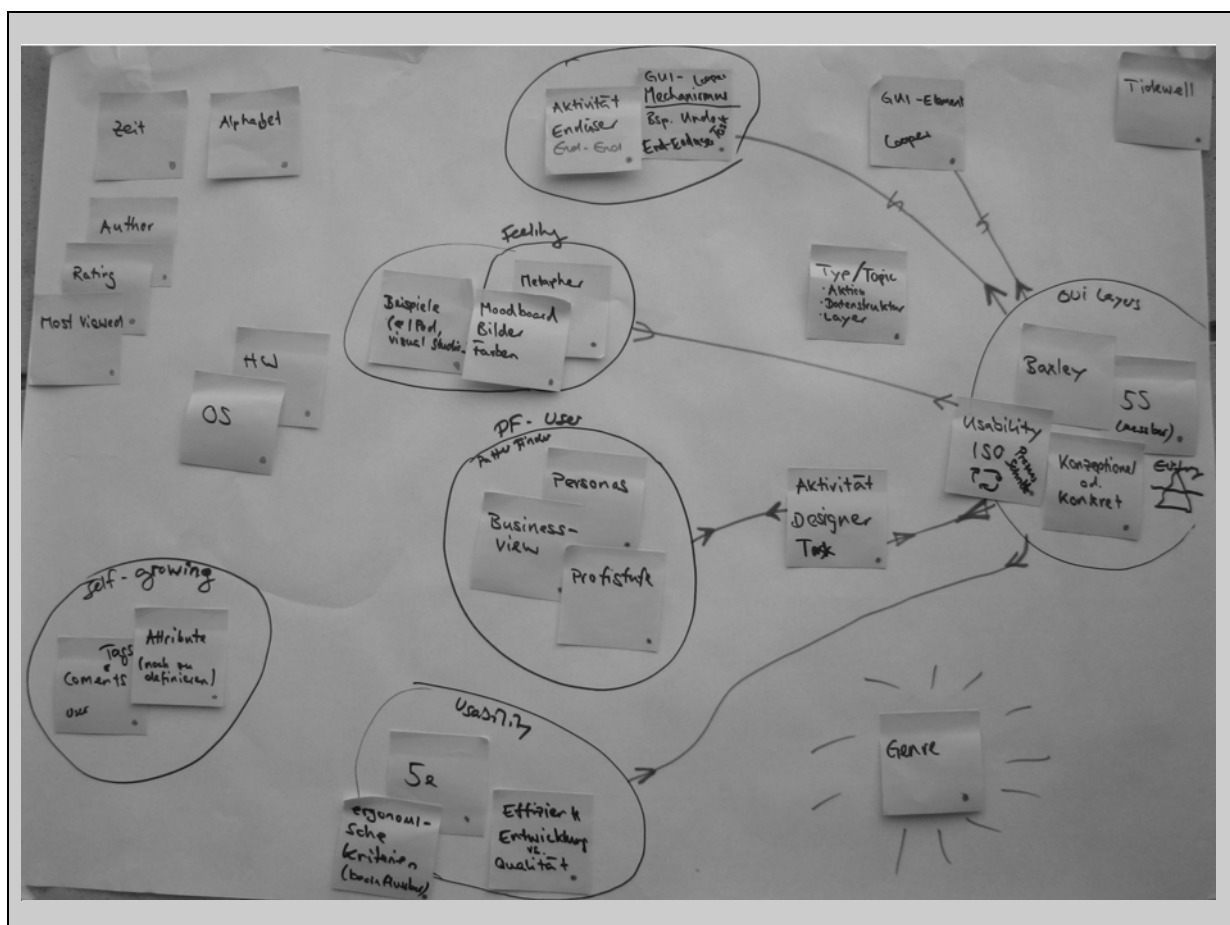


Abbildung 30 – Gruppierung Top Down

Anschliessend wurden mögliche Verbindungen und Hierarchien zwischen diesen Kategorien untersucht. Dieser Weg hat sich als äusserst vielschichtig und nicht praxistauglich herausgestellt, da die Kategorien und deren Verbindungen in einer unüberschaubaren Menge untereinander in Beziehung standen. Insbesondere das Festhalten der hierarchischen Sicht und die gegenseitige Beeinflussung waren schwierig miteinander in Einklang zu bringen.

Aufgrund dieser Probleme wurde nicht versucht, Patterns zu diesen Kategorien zuzuordnen. Dieser Ansatz wurde nicht mehr weitergeführt.

Wenn wir auf die analysierten Pattern Tools zurückblicken, entsteht der Eindruck, dass diese von UCD Spezialisten für UCD Spezialisten gemacht wurden. Diese Tools lassen sich nur schwer in der Praxis anwenden.

4.2.4 Bottom Up Ansatz

Der Bottom Up Ansatz verfolgt den umgekehrten Weg. Die für den Lösungsfindungsprozess notwendigen Kategorien werden nicht vorgängig festgelegt, sondern ergeben sich durch die Eigenschaften oder Charakteristiken der Patterns.

Es werden nicht die Patterns kategorisiert, sondern deren Eigenschaften. Für diesen Ansatz werden sogenannte Tags eingesetzt. (Findings Kapitel 4.6.1.2 / 4.6.1.3 / 4.6.1.4 / 4.6.1.5 / 4.6.1.16)

Der Vorteil dieses Ansatzes ist, dass die Titel der Patterns und Kategorien nicht bekannt sein müssen. Tags können sehr intuitiv und mehrschichtig sein und sind näher beim Benutzer. Den Metabeschreibungen (Tags; Kapitel 3.2.1.2) der Patterns sind keine Grenzen gesetzt und können auf mehrere Benutzerprofile ausgerichtet sein.

Da ein Tag auf mehrere Patterns zutreffen kann, entstehen automatisch Verbindungen unter Patterns und unter den zusammengefassten Gruppen (Kapitel 3.1.9).

Als Vorbereitung zur Realisierung dieses Ansatzes, wurden 30 Patterns (Tidwell J. , 2005) mit unterschiedlichen Gesichtspunkten ausgewählt. Diese Patterns wurden auf je einer Karte beschrieben (Vorderseite). Auf der Rückseite wurde Tags (Eigenschaften oder Charakteristiken) aufgelistet.

Vorderseite der Karte

- Name des Patterns.
- Beschreibung des Patterns.
- Beispiel Anwendung.

Rückseite der Karte

- Eigene Tags. Dies sind Stichworte, die man mit diesen Patterns in Verbindung bringt oder bringen könnte.

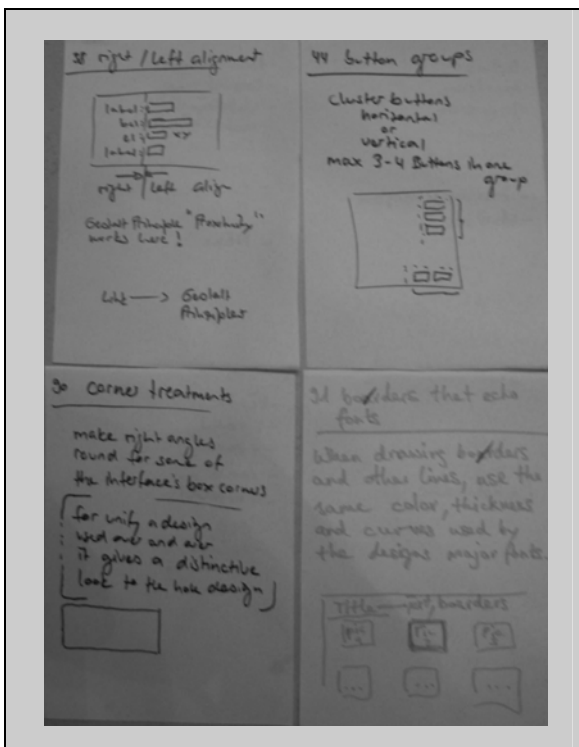


Abbildung 31 - Pattern Card – Bottom Up

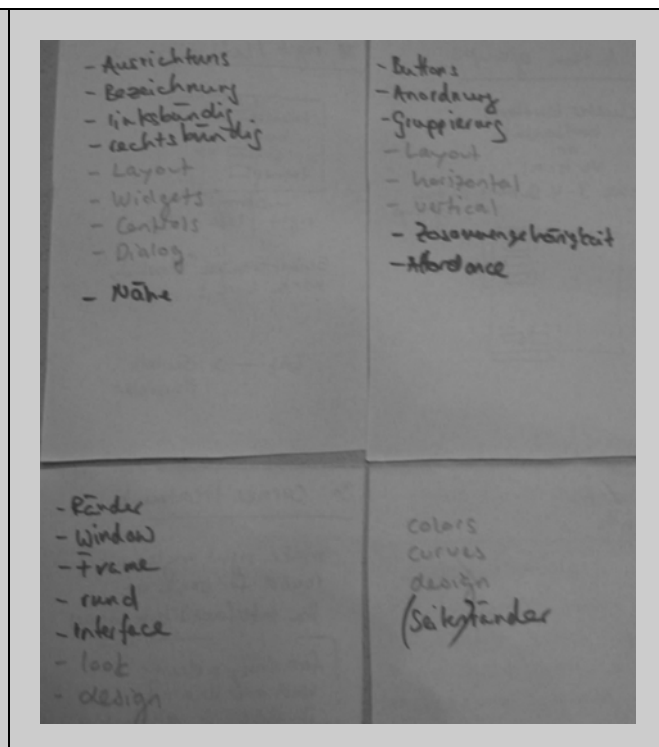


Abbildung 32 - Pattern Tags – Bottom Up

Beispiel aus obigem Bild:

Aus den Tags „Sichtbarkeit“, „Wichtigkeit“, „zurechtfinden“, „Gewohnheit“, „Orientierung“ und „erinnern“ entstand der Kategoriename „Userneeds“.

Die Kategorien entstehen somit nicht aus irgendwelchen Begriffen, sondern aus den Tags, die durch die Benutzer den einzelnen Patterns zugeteilt werden. Das heisst, je nach Wissensstand oder Gebiet aus dem die Benutzer stammen, bilden sich diese Tags und deren Kategorien. So wächst und gestaltet sich der Lösungsfindungsprozess individuell nach Benutzergruppen und deren Vokabular (Findings Kapitel 4.6.1.14).

Grundsätzlich könnte bei der Auslieferung des PatternFinders ein Basisstamm an Tags und Kategorien zu einer definierten Auswahl an Patterns mitgegeben werden.

Die erwähnte Zuteilung zu einer Phase schränkt die Menge der Patterns bzw. der Tags auf der horizontalen Achse ein.

In einem zweiten Schritt werden die Tags durch die primären Kategorien weiter eingeschränkt. Bestehen in diesen Kategorien weitere Unterkategorien, werden diese jeweils schrittweise gegen unten weiter eingeschränkt. (Findings Kapitel 4.6.1.7)

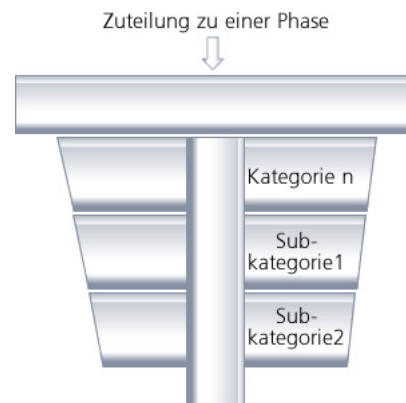


Abbildung 34 – Kategorien - Subkategorien

Im unten stehenden Beispiel wurden nach dem Usability Engineering Lifecycle (Mayhew, 2008) folgende Phasen definiert:

- Requirement Analyse.
- Conceptual Design.
- Screen Design Standards.
- Detailed UI Design.
- Software Development.
- UI Testing.

Der Phase Screen Design Standards wurden folgende Oberkategorien zugeordnet:

- Genre.
- Design.
- Container.
- Layout.
- User Needs.
- Unterstützung.

Das Pattern „Button Group“ hat die Tags „Buttons“, „Anordnung“, „Gruppierung“, und „Affordance“ etc. Der Tag „Anordnung“ erscheint in den Kategorien „Container“ und „Layout“. Die Kategorie „Layout“ erscheint sowohl unter der Phase Screen Design Standards und Detailed UI Design.

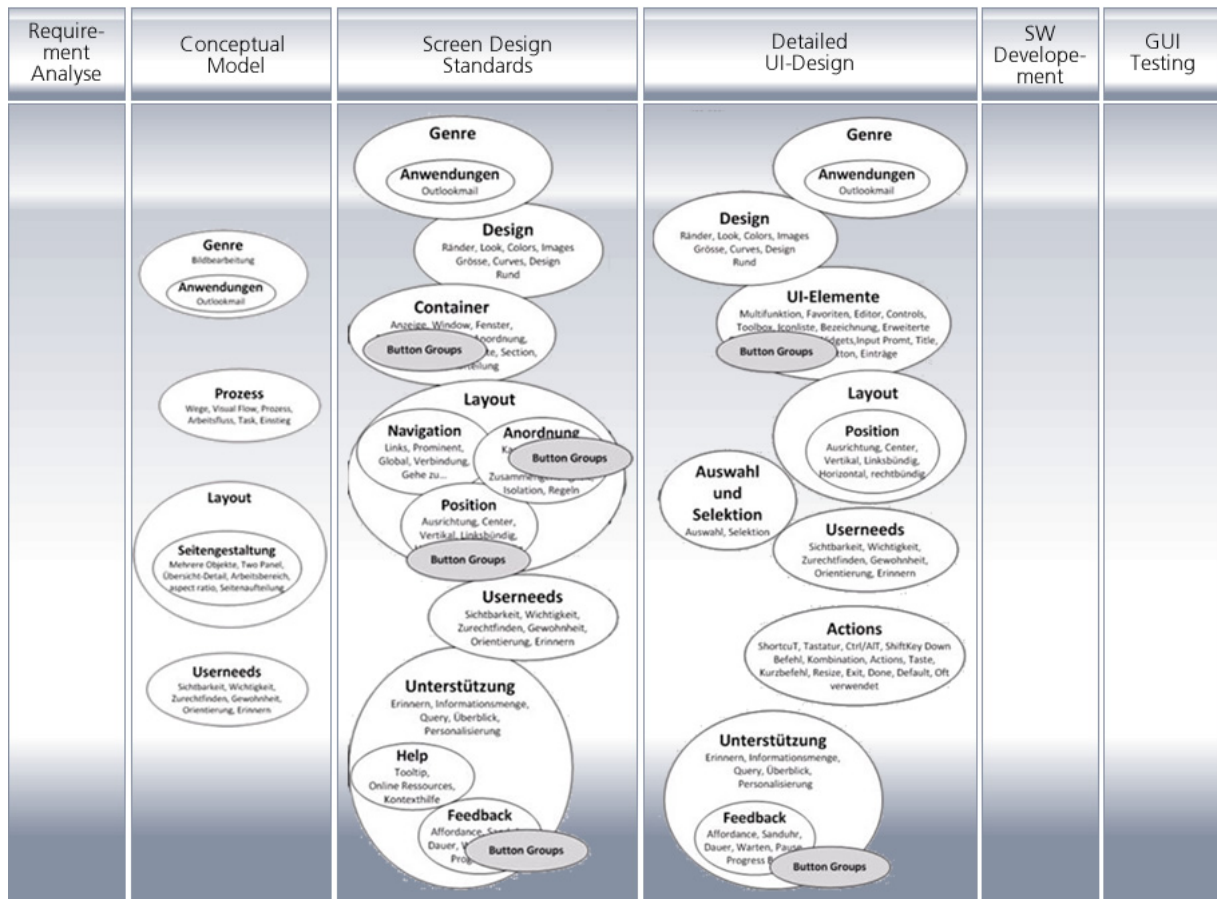


Abbildung 35 - Phase – Beispiel Kategorien aus Bottom Up

Dadurch kann das Pattern über unterschiedliche Wege gefunden werden. Um das Konzept zu testen, wurden die im Bottom Up Ansatz erarbeiteten Kategorien und Tags verwendet. Dieser dynamische Layer der Tags erlaubt eine flexible, austauschbare, in der Menge nicht eingeschränkte Möglichkeit von Gruppierungen und Hierarchien aufzubauen. Im Weiteren ist dieses Konzept offen, um Tags oder Kategorien firmen- oder personenspezifisch zu gestalten.

Ein Tag kann zu mehreren Kategorien oder Pattern gehören. Hier das Datenmodell für den DB-Designer:

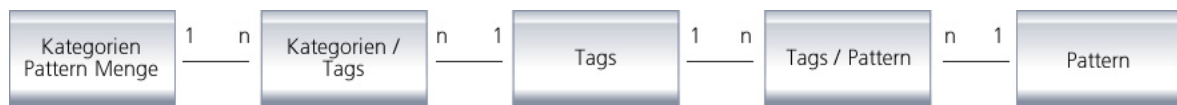


Abbildung 36 – Kategorien-Tags-Pattern Datenmodell

4.3 Task Model

Anhand des Eingangs dieses Kapitels gewählten Beispiels „das Plazieren von mehreren Buttons“ wird der primäre Task Flow, wie die primäre Persona zu der gewünschten Lösung gelangt, mit folgender Ausgangssituation beschrieben:

- Der Benutzer hat das Problem erkannt und ist an einer Lösung interessiert, die ihm noch nicht bekannt ist.

Zusätzlich werden die sekundären Task Flows konzeptionell beschrieben. Die Task Flows zeichnen sich durch folgende unterschiedliche Ausgangssituationen aus:

- Der Benutzer hat kein konkretes Problem. Er zeigt jedoch Interesse an den Beschreibungen (Shadow Path).
- Der Benutzer sucht ein ihm bereits bekanntes Pattern (direkte Navigation, Auswahl über Index).
- Der Benutzer sucht zu einer Anforderung lediglich weitere Informationen (verwandte Patterns, Shadow Path).

4.3.1 Primärer Task Flow

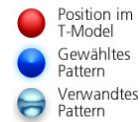
Ausgangslage: Das Problem ist erkannt, die Lösung ist jedoch unbekannt.

Szenario: Peter S. Coder ist in seiner Arbeit auf ein Problem gestossen. Er ist sich über dieses Problem im Klaren und kann dieses auch artikulieren.

Task: Das Plazieren von mehreren Buttons

Die Position im T-Modell wird jeweils mit einer Grafik am rechten Rand visualisiert.

Legende:



1. Peter S. Coder startet den PatternFinder. Sämtliche Phasen werden zur Auswahl angezeigt (Findings Kapitel 4.6.1.15).

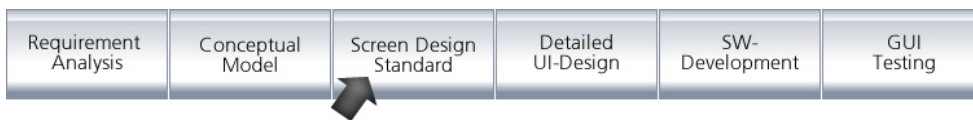


Abbildung 38 – Phasen Auswahl



2. Die Entwicklung befindet sich in der Phase Screen Design Standards. Deshalb wählt Peter S. Coder diese aus. Sämtliche Tag-Kategorien, die Tags selber und die dazugehörigen Patterns werden angezeigt. Die Menge ist immer noch zu gross um eine vernünftige Lösung zu finden. Peter S. Coder kann unter folgenden Gruppen (Abbildung 40) auswählen. Dahinter verbergen sich Tags und Patterns.

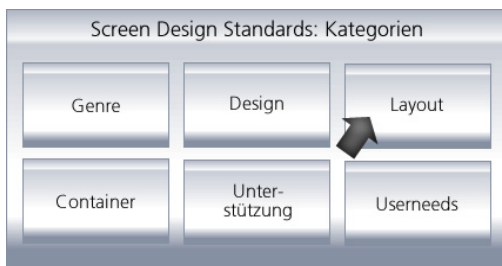


Abbildung 39 – Kategorien Auswahl



3. Peter S. Coder entscheidet, dass die Kategorie „Layout“ sein Problem am besten beschreibt und wählt diese aus. Sämtliche Tag-Subkategorien, die Tags selber und die dazugehörigen Patterns der Kategorie „Layout“ werden angezeigt.

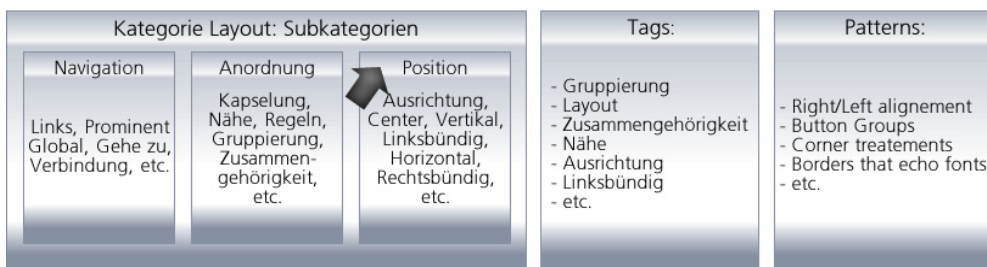


Abbildung 40 – Subkategorien Auswahl



4. Da ihm zum Thema Plazieren von mehreren Buttons das Stichwort „Position“ in den Sinn kommt, wählt er dieses aus. Sämtliche Tags und die dazugehörigen Patterns der Sub-Kategorie „Position“ werden angezeigt. Es besteht keine weitere Sub-Kategorie. Die Tags könnten jedoch statt Kategorien zur weiteren Einschränkung verwendet werden.

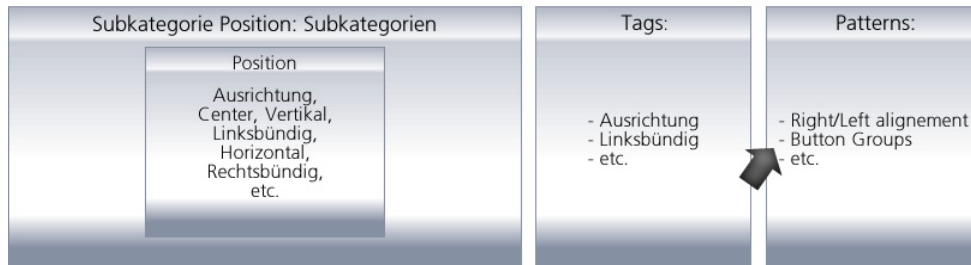


Abbildung 41 – Tag Auswahl

5. Auf den ersten Blick sieht er das Pattern „Button Groups“ und wählt dies zur näheren Begutachtung aus. Peter S. Coder hat ein Pattern für das Plazieren von mehreren Buttons gefunden, was ihm bei der Lösung des Problems hilft.

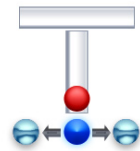


Abbildung 42 – Pattern Präsentation

4.3.2 Sekundäre Task Flow

Ausgangslage 1: Der Benutzer hat kein konkretes Problem, er zeigt jedoch Interesse an den Beschreibungen (Shadow Path).

Szenario: Lukas von Fach beschäftigt sich aktuell mit dem Design eines UI's und ist sich keiner Probleme bewusst. Trotzdem möchte er von den Möglichkeiten des PatternFinders profitieren und die Qualität seiner Arbeit positiv beeinflussen.

Er lässt sich die ausführlichen Beschreibungen der einzelnen Phasen anzeigen. Er entdeckt im Text einen Beitrag zu einer Phase, die seiner aktuellen Arbeit sehr nahe kommt. Dieser Text ist mit einem Link unterlegt. Er wählt diesen aus. Er erhält die Beschreibung diverser Lösungskategorien der ausgewählten Phase. Er studiert die Beschreibungen der angezeigten Lösungskategorien und erhält Hinweise auf diverse Patterns und lernt dabei neues über strukturiertes UI Design.

Er entscheidet sich für eine bestimmte Kategorie und erhält eine Auswahl aller Patterns dieser Kategorie. Er studiert die für ihn neuen Informationen und entscheidet sich anschließend für ein bestimmtes Pattern. Er nimmt sich Zeit, dieses genau zu studieren.

Dieser Task wird dazu beitragen, dass er sein Wissen im UI Design verbessert. Zusätzlich lernt er, den PatternFinder als unterstützendes Tool einzusetzen.

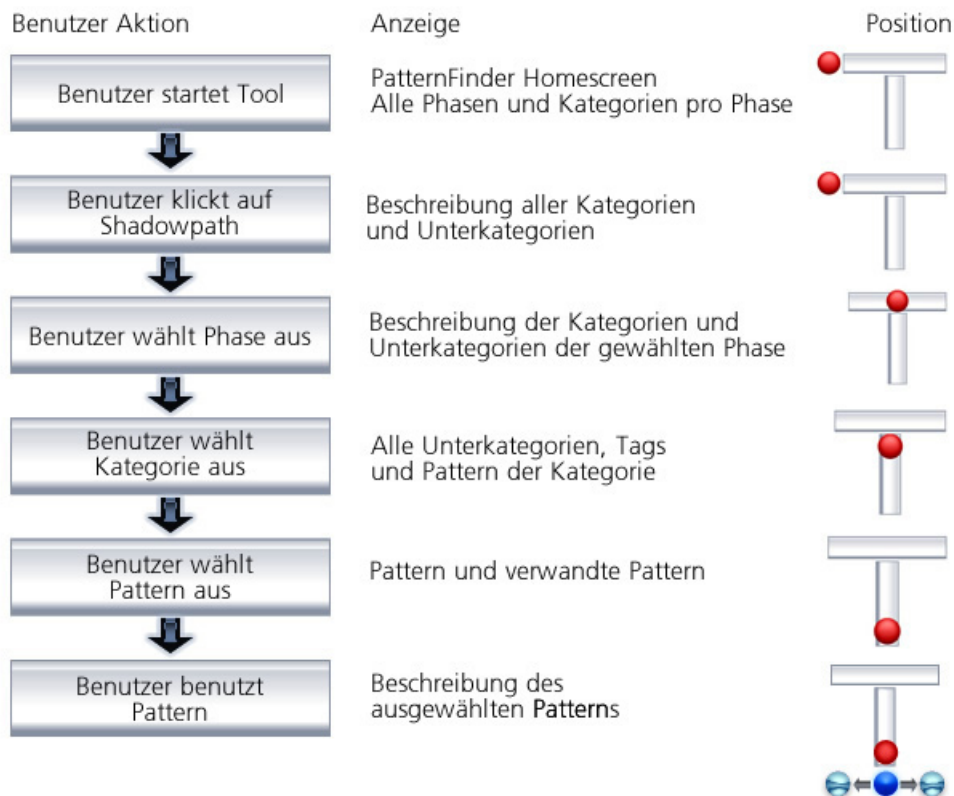


Abbildung 43 – Task Flow: Kein Problem, hat nur Interesse (Shadow Path über Beschreibung)

Ausgangslage 2: Der Benutzer sucht ein ihm bereits bekanntes Pattern.

Szenario: Peter S. Coder erhält eine Spezifikation, die mit Patterns beschrieben wurde. Davon muss er einen grossen Teil umsetzen. Um ein Pattern korrekt umzusetzen, möchte er weitere Details zu diesem Pattern erhalten.

Er gibt im PatternFinder die mitgelieferte eindeutige Identifikation dieses Patterns ein. Er erhält dieses Pattern zur Ansicht und wählt es für die Detailinformationen aus. Er bemerkt anschliessend, dass er auch die Möglichkeit hätte, ein Pattern über dessen Namen im Index anzuwählen.



Abbildung 44 – Task Flow: Sucht bestimmtes Pattern

Ausgangslage 3: Der Benutzer sucht zu einer Anforderung lediglich weitere Informationen.

Szenario: Lukas von Fach bekommt eine UI Spezifikation und will sie mit Patterns ergänzen. Er gibt einen Begriff, z.B. UNDO in das Suchfeld ein. Er erhält eine Auflistung, unterteilt in Phasen, Kategorien und Patterns, bei welchen dieser Suchbegriff vorkommt. Ob dieser Suchbegriff im Text, in der Beschreibung oder als Tag gefunden wird, ist für ihn irrelevant.

Er hat nun die Möglichkeit, eines der gefundenen Resultate auszuwählen. Wählt er eine Phase aus, wird diese im PatternFinder geöffnet. Wählt er eine Kategorie aus, erhält er alle Pattern dieser Kategorie. Er kann auch direkt eines der gefundenen Patterns anwählen und erhält die Detailinformationen dieses Pattern mit sämtlichen Verbindungen.

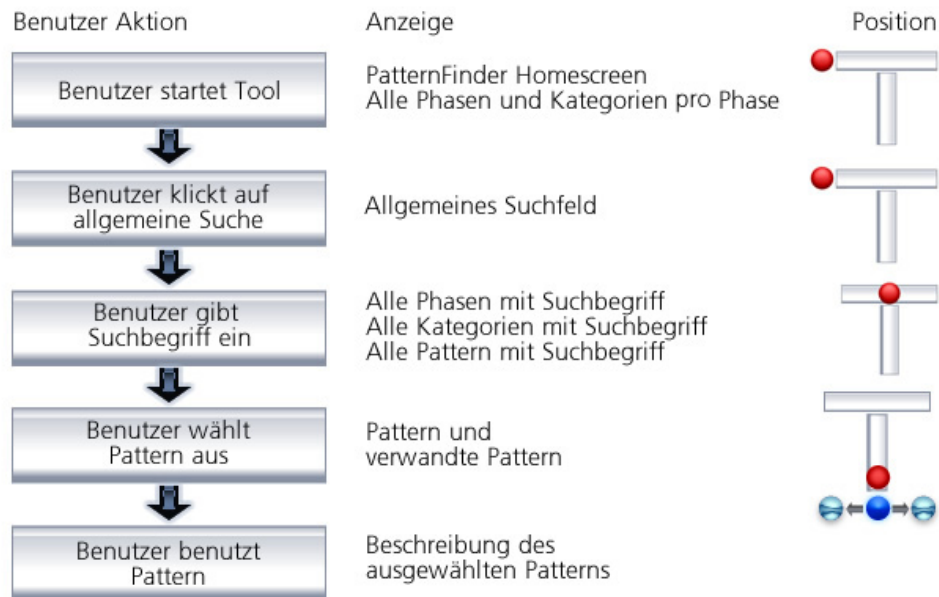


Abbildung 45 – Task Flow: Sucht zu Anforderung lediglich Information

4.4 PatternFinder Prototype

4.4.1 Screen Architektur

Um das Konzept zu prüfen und visualisieren wird ein Prototyp erstellt. Er wird auf den primären Task Flow ausgerichtet. Peter S. Coder hat das Problem erkannt und sucht unbekannte Patterns zur Lösung seines Problems.

Im Prototyp wird das Konzeptionelle T-Modell in drei Screens umgesetzt:

Einstieg in den Lösungsfindungsprozess

Der Einstieg in den Lösungsfindungsprozess soll geleitet sein. Auf dem ersten Screen wird der Entwicklungsprozess abgebildet. Es muss eine Prozessphase ausgewählt werden, um mit einer eingeschränkten Resultatmenge in den Lösungsprozess einsteigen zu können.

Lösungsfindung über Kategorie, Tag und Selektion

Ist eine Prozessphase ausgewählt, kann stufenweise zur möglichen Lösung navigiert werden. Dies erfolgt über den in der Information Architektur erarbeiteten Weg. Die Lösungsmenge wird über die Selektion von Kategorien und Tags eingeschränkt.

Grundlegend anders ist, dass die Patterns nicht direkt einer Kategorie zugeordnet werden, sondern dass die Tags kategorisiert werden (siehe Kapitel 4.2.4).

Es erscheinen nur noch die Patterns, die mit den in der Kategorie vorhanden Tags gekennzeichnet sind.

Lösung – das Patterndetail

Wurde ein Pattern ausgewählt, werden die Details dazu angezeigt. Von diesem Pattern gibt es eine zusätzliche Navigation über Querverbindungen zu anderen Kategorien und Patterns (siehe Kapitel 4.2.5)

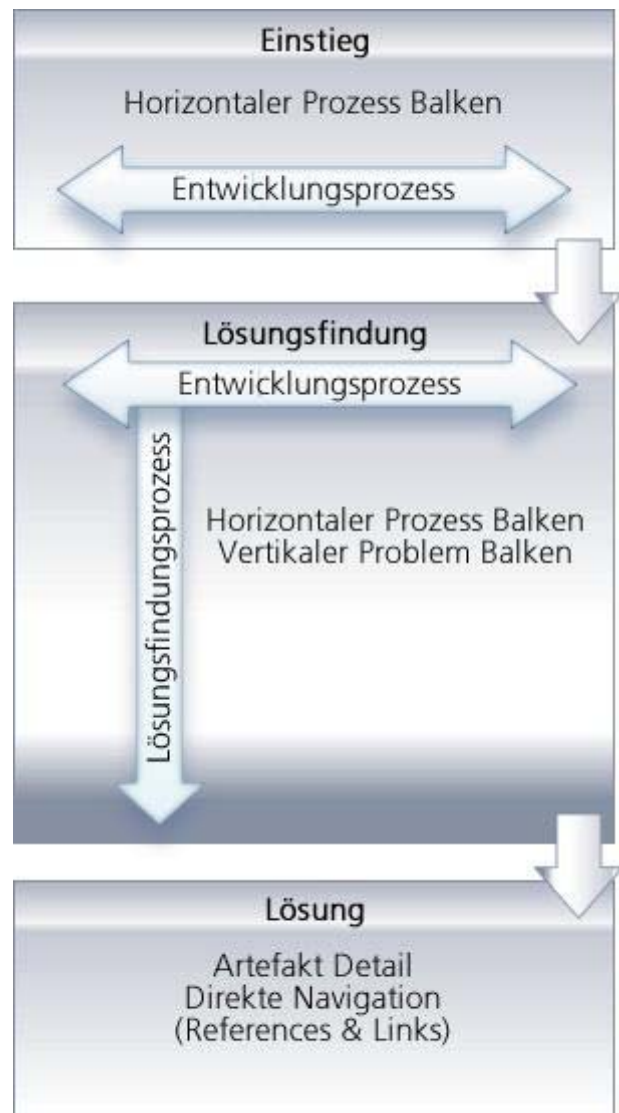


Abbildung 46 – Task Flow: Prototyp

Die Wireframes im Kapitel 4.4.2 visualisieren das Layout und die Anordnung der verschiedenen Elemente.

4.4.2 Wireframes

4.4.2.1 Einstieg in den Lösungsfindungsprozess

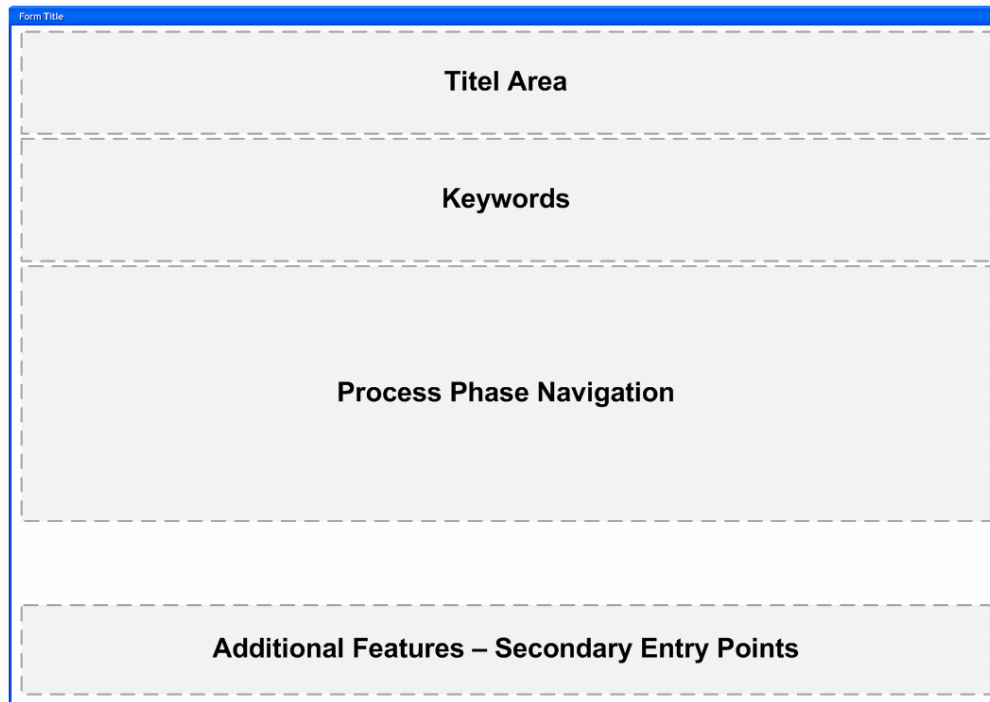


Abbildung 47 - Wireframe Welcome Page

Der Welcome Page dient als Einstieg und ist in folgende Bereiche aufgeteilt:

- **Title Area** - Mit einem ansprechenden Titel werden die positiven Gefühle des Benutzers geweckt. Das Thema Pattern soll damit schmackhaft gemacht werden.
- **Keywords** - Eine kurze Einführung in das Thema.
- **Process Phase Navigation** - Dieser Bereich widerspiegelt die Phasen des Entwicklungsprozesses und des horizontalen Prozessbalken. Informationen zum aktuellen Entwicklungsprozess werden hier angezeigt (Kapitel 4.4.10).
- **Additional Features** - Hier besteht die Möglichkeit, das Prozess Modell zu wechseln. Auch Auflistungen von „Last Viewed“ oder „Best Rated“ Patterns können hier als alternativer Einstieg untergebracht werden. Diese können einen passiven Pattern Lernprozess anregen. Auf diese Features wird nicht weiter eingegangen, da sie für das Hauptszenario keine Rolle spielen.

4.4.2.2 Lösungsfindung über Kategorie- und Tag-Selektion

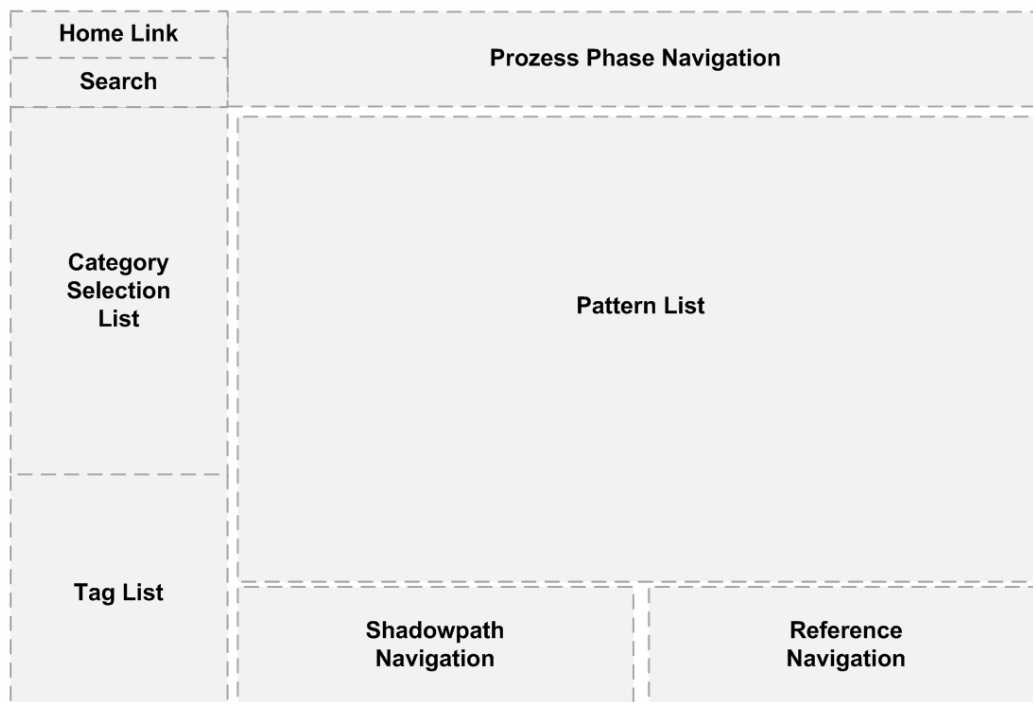


Abbildung 48 - Wireframe Lösungsfindung

Der PatternFinder hilft die grosse Zahl an Patterns über die Lösungsfindungsseite einzuschränken und auf das gesuchte, aber vorher noch unbekannte Pattern zu stossen.

Die Lösungsfindungsseite ist in folgende Bereiche aufgeteilt:

- **Prozess Phase Navigation** – Wichtig ist, dass hervorgehoben wird, wo im Prozess die Problemlösung zu finden ist. Auch ein Wechsel zwischen den Phasen ist möglich (Kapitel 4.4.6).
- **Category Selection List** – Hier wird über die ersten Stufen nach Kategorien und Unterkategorien der Problemlösung navigiert (Kapitel 4.4.7).
- **Tag List** – Die Tag List ermöglicht eine erweiterte n-Stufige Navigation nach der Auswahl der ersten Stufen (Kapitel 4.4.8).
- **Pattern List** – Die Patterns sind der Hauptaspekt des PatternFinders. Deswegen werden sie an zentraler Position angezeigt. Um das Finden der Patterns in der Liste zu unterstützen, wird eine Kurzübersicht verwendet (Kapitel 3.2.1.3).
- **Shadow Path Navigation** – Diese Art von Navigation soll auch den Benutzern gerecht werden, die keine konkrete Lösung zu einem konkreten Problem suchen (Kapitel 4.2.5). Es soll zusätzlich das Lernen und den Aufbau von UI Design Fachwissen unterstützen (Kapitel 4.4.10).
- **Reference Navigation** – Durch den Ansatz mit Tags, ist es möglich, Referenzen automatisch zu generieren (Kapitel 4.2.5).
- **Search** – Die Suche wird in diesem Rahmen nicht weiter behandelt. Es wird jedoch auf Morville & Rosenfeld, 2006 verwiesen, um Informationen zur Optimierung der Suche zu erhalten.
- **Home Link** – Führt zurück auf die Einstiegsseite

4.4.2.3 Die Lösung – Das Pattern Detail

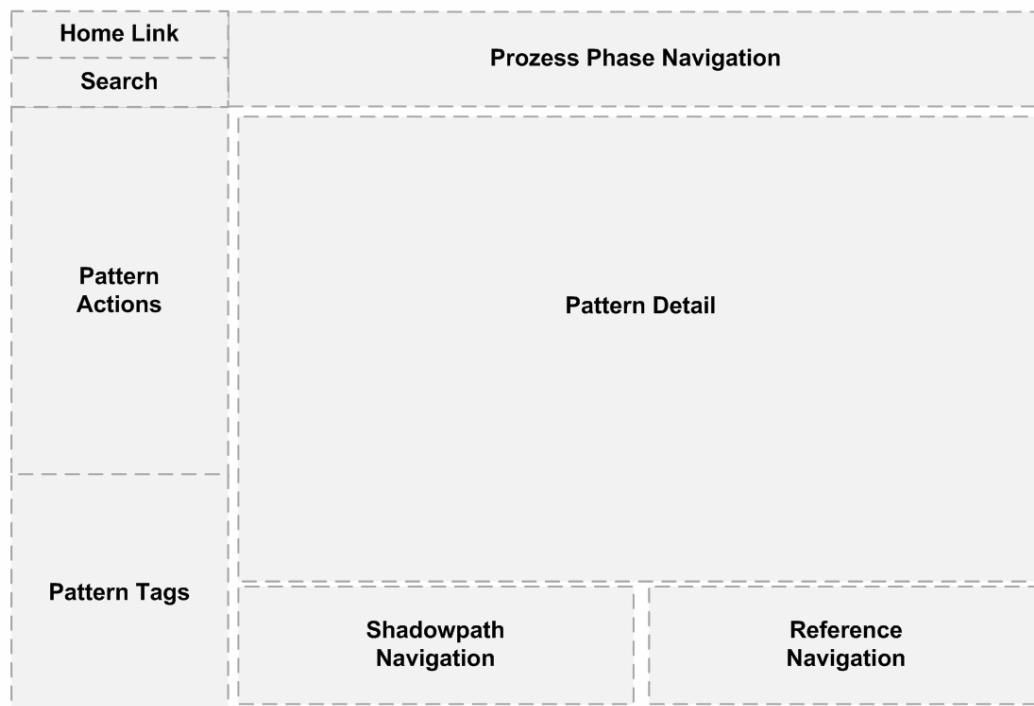


Abbildung 49 - Wireframe Lösung

Ein Pattern wurde ausgewählt und wird nun präsentiert.

Dieser Screen besteht aus folgenden Bereichen:

- **Prozess Phase Navigation** – Wichtig ist, dass hervorgehoben wird, wo im Prozess die Problemlösung zu finden ist. Auch ein Wechsel zwischen den Phasen ist möglich (Kapitel 4.4.6).
- **Pattern Actions** – Hier werden die weiteren unterstützenden Aktionen angezeigt, um das gefundene Pattern zu verwenden. Unter anderem wird es möglich sein, zu drucken, Bookmarks zu setzen und das Pattern in geeigneter Form ins Clipboard (Copy & Paste) zu kopieren (Kapitel 4.4.13).
- **Pattern Tags** – Die Pattern Tags werden angezeigt. Eine Selektion ist möglich, wobei sich das Patterndetail schliesst und die Patternliste (Lösungsfindung), angepasst auf den selektierten Tag, wieder angezeigt wird. Die Auswahl der Category Selection List geht nicht verloren (Kapitel 4.4.14).
- **Pattern Detail** – Das Detail eines Patterns muss gewisse Standards einhalten (Kapitel 3.2.1.1) Es wird eine Navigation zurück auf die Pattern Liste angeboten.
- **Shadow Path Navigation** – Dieser Bereich soll das Lernen und den Aufbau von UI Design Fachwissen unterstützen und zeigt Tipps und Tricks an, die zum Pattern und zu den Tags des Pattern passen (Kapitel 4.4.10).
- **Reference Navigation** – Durch den Ansatz mit Tags zu arbeiten, ist es möglich, Referenzen zwischen Lösungsidentitäten automatisch zu generieren (Kapitel 4.2.5).
- **Search** – Die Suche wird in diesem Rahmen nicht weiter behandelt. Es wird jedoch auf Morville & Rosenfeld, 2006 verwiesen um Informationen zur Optimierung der Suche zu erhalten.
- **Home Link** – Führt zurück auf die Einstiegsseite

4.4.3 Screen Designs

Form Title

Pattern Finder

Find a solution to your Design Problem

Each Problem occurs somewhere in a Process...
 If you don't know or use a Development Process try the Mayhew Process

In which Process Phase are you ?

The Mayhew Process

Requirements Engineering	Conceptual Model	Screen Design Standards	Detailed UI-Design	Software Development	GUI Testing
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat mehr >>	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat mehr >>	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat mehr >>	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat mehr >>	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat mehr >>	Testing und Deployment der Software Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat mehr >>

You should now be able to choose a Projectphase Level in the Pattern Finder Guide
 Are you using a different Project Process ?
[Change to RUP](#)
[Change to HERMES](#)

Last Viewed Patterns
[Breadcrumbs](#)
[Button Group](#)
[Faceted Navigation](#)

Best Rated
[Directory Navigation](#)
[Home Link](#)
[Double Tab Navigation](#)

Abbildung 50 - Welcome Screen

Form Title

Pattern Finder
Find a solution to your Design Problem

Direct Pattern Search

Enter Text

Search

Choose a Category

Unterstützung

Feedback

Sanduhr, Affordance, Erinnern, Feedback,

Help

Sanduhr, Affordance, Erinnern,

Design

Ränder, Look, Colors, Images, Grösse, Curves

Container

Anzeige, Window, Fenster, Dialog, Gruppierung

Layout

Navigation, Anordnung, Position ...

User Needs

Sichtbarkeit, Wichtigkeit, Zurechtfinden,

Genre

Anwendungen ...

Choose a Tag

Affordance, Anordnung,

Button, Gruppierung, Gruppierung,

Horizontal, Layout, Vertikal,

Zusammengehörigkeit

Chosen Tags:

Requirements Engineering

Conceptual Model

Screen Design Standards

Detailed UI-Design

Software Development

GUI Testing

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat

UI Patterns Browse Results (245 Patterns found)

Browsing by: [Screen Design Standards](#) | [Unterstützung](#) | [Feedback](#)

Button Groups			
<input type="checkbox"/>	Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Lorem Ipsum dolores Reoal molw dnnwbw smoxw
<input type="checkbox"/>	Corner treatments Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Ipsum Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Ipsum Lorem Ipsum dolores Reoal molw dnnwbw smoxw
<input type="checkbox"/>	Borders that echo fonts Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw
<input type="checkbox"/>	Right/left alignment Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw
<input type="checkbox"/>	Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw
<input type="checkbox"/>	Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw	<input type="checkbox"/> Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw

[< previous](#) Page 1 2 3 [next >](#)

Screen Design Standards

Die Phase oder die Aktivität Requirements Engineering.
Inkl. User / Kontext Analyse
Lorem ipsum dolores Lorem
ipsum dolores Lorem ipsum dolores Lorem ipsum
dolores Lorem ipsum dolores Docum dolores Lorem ipsum dolores
Lorem ipsum dolores Lorem ipsum dolores Lorem
ipsum dolores Lorem ipsum dolores Lorem ipsum Dolor Lorem

Referenzen

Unterstützung ist eine Kategorie, die auch in der Detailed UI-Design verwendet wird. [Klicken Sie hier für weitere Informationen.](#)

Tags:
Folgende Patterns haben ähnliche Tags:
[Corner Treatments](#), [Breadcrumbs](#), [Progressbar](#)

Abbildung 51 - Lösungsfindungs Screen

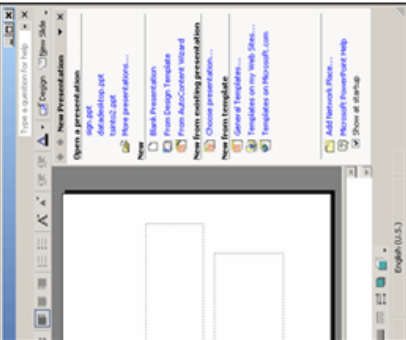
Form Title	<h1>Pattern Finder</h1>									
Direct Pattern Search <input type="text"/> Enter Text <input type="button" value="Search"/>	print pattern bookmark pattern copy to clipboard back to patternlist									
Pattern Options	<p>Browsing by: Screen Design Standards Layout Anordnung Button Groups Pattern</p> <h2 style="color: blue;">Pattern Detail (REF 04020)</h2> <h3>Button Groups Pattern</h3> <p>What: Instead of using menus, present a large group of related actions on a UI panel that's richly organized and always visible.</p> <p>Use when: You need to present many actions -- too many for a Button Group. You could put them into a menu, but maybe you don't have a menu bar at all, or you'd rather make the actions more obvious. Same for pop-up menus; they're just not visible enough. Your users may not even realize the pop-up menus exist. Or maybe your set of possible actions is too complex even for a menu. Menus are best at showing a flat set of actions (since pull-right menus, or cascading menus, are hard for some users to manipulate), in a very simple, linear, one-line-per-item presentation. If your actions need to be grouped, and if those groups don't fit the standard toplevel menu names -- such as File, Edit, View, Tools, or Format -- then you might want a different presentation altogether. This pattern can take up a lot of screen space, so it's not usually a good choice for small devices.</p> <p>Why: The "Use When" section already hinted at the two main reasons for using action panels: visibility and freedom of presentation. You could enhance the effect by underlining them when the mouse rolls over them. This is what Microsoft does in its interfaces that use action panels. Feel free to experiment. However, usability-test it to make sure users actually see the actions as clickable things, not ordinary text or pictures.</p> <p>By placing the actions out on the main UI, and not hidden inside a traditional menu, you make those actions fully visible to the user. Really, action panels are menus, in the generic sense; they just aren't found in menu bars, dropdowns, or pop-ups. Users don't have to do anything to see what's on an action panel -- it's right there in front of them -- so your interface is more discoverable. This is particularly nice for users who aren't already familiar with the traditional document model and its menu bars.</p> <p>There are many ways to structure objects on an interface: lists, grids or tables, hierarchies, and just about any custom structure you can devise. But button groups and traditional menus only give you a list (and not a very long one). Action panels are freiform -- they give you as much freedom to visually organize verbs as you have for nouns. Use it wisely!</p> 									
Pattern Tags	Affordance. Anordnung, Button, Gruppierung, Horizontal/Layout, Vertikal, Zusammengehörigkeit									

Abbildung 52 – Lösungs Screen

4.4.4 Farbkonzept

Jede Entwicklungsphase hat eine primäre und eine sekundäre Farbe.

Requirements Engineering	Primärfarbe
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat	Sekundärfarbe 60% Transparente Primärfarbe

Abbildung 53 - Primär und Sekundärfarben

Um den Benutzer zu leiten, werden die aktuellen Auswahlelemente orange gefärbt. Dies soll die Aufmerksamkeit auf die aktuelle Auswahl lenken.



Abbildung 54 – Selektionsfarbbeispiel

Für die Navigationselemente werden entweder Hyperlinks (blau unterstrichen) oder grössere Selektionsflächen verwendet. Selektionsflächen werden weiss auf blau gekennzeichnet.



Abbildung 55 - Selektionsfläche

Auf die Ausarbeitung des detaillierten Farbkonzepts und auf ein weiteres Ausarbeiten der Affordance des UI Designs wird im Rahmen dieser Arbeit verzichtet.

Der Scope liegt auf dem Lösungs- und Patternfindungsprozess und nicht auf der abschliessenden visuellen Ausarbeitung.

4.4.5 UI Elemente und Umsetzungsdetails

In diesem Kapitel werden die wichtigsten UI Elemente des PatternFinder Prototyps beschrieben.

4.4.6 Process Phase Navigation

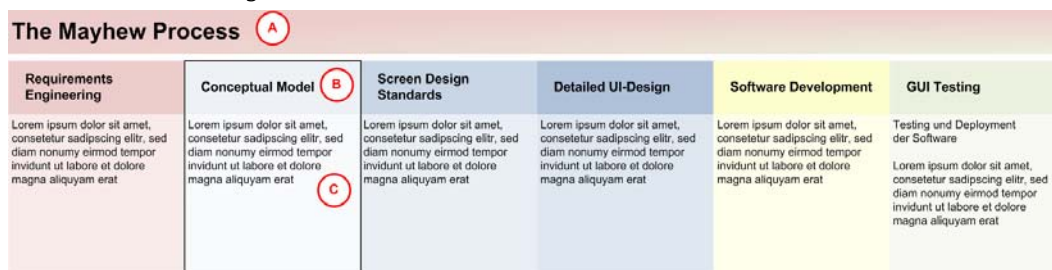


Abbildung 56 - Process Phase Navigation UI Element

- A) Der Entwicklungsprozessname z.B. RUP, Usability Engineering Lifecycle (Mayhew, 2008) oder Hermes
- B) Titel der Prozessphase – der Titel ist selektierbar – und gibt Feedback auf Mouseover. Der Titel hat die primäre Phasenfarbe.
- C) Die Beschreibung der Prozessphase hat die sekundäre Phasenfarbe.

Nach der Auswahl einer Phase sieht im Lösungsfindungsscreen die Navigation wie folgt aus:

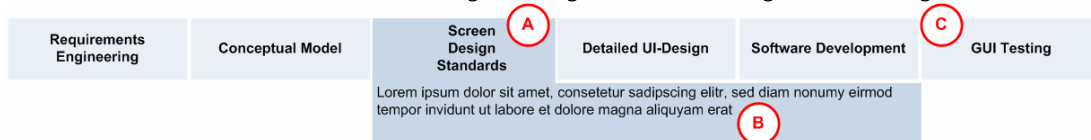


Abbildung 57 - Process Phase Navigation UI Element (in Lösungsfindungs Screen)

- A) Die ausgewählte Entwicklungsphase mit der primären Phasenfarbe
 - B) Die textuelle Kurzbeschreibung der aktuellen Entwicklungsprozessphase – hinterlegt mit der primären Phasenfarbe.
 - C) Andere Entwicklungsphasen: Diese ist mit der sekundären Phasenfarbe des aktuellen Entwicklungsprozess gekennzeichnet.
- (Findings Kapitel 4.6.1.24)

4.4.7 Category Selection List

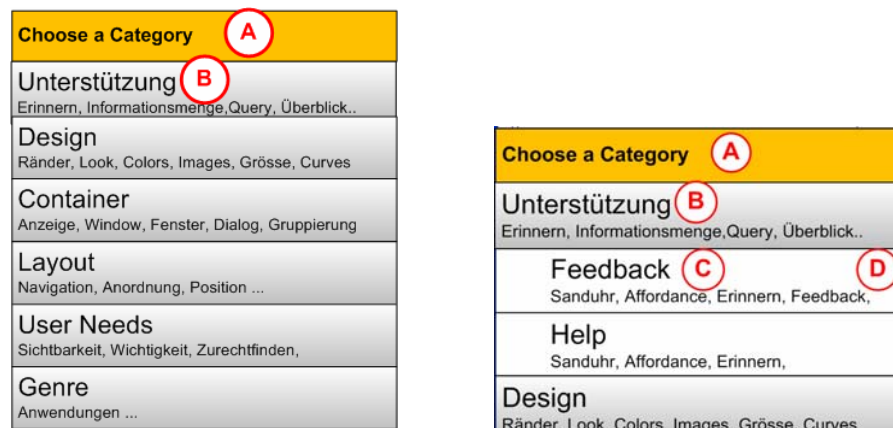


Abbildung 58 – Category Selection

- A) Der Titel der Selection List: Aufforderung zur Auswahl
- B) Der Titel der Stufe 1 Kategorie: Das ganze Element ist auswählbar. Wenn eine Kategorie noch Unterkategorien hat, werden diese bei einer Auswahl wie in C angezeigt.
- C) Der Titel der Unterkategorie
- D) Die häufigsten Tags der Kategorie werden automatisch als Beschreibung der Kategorie verwendet.

(Findings Kapitel 4.6.1.21 / 4.6.1.25)

4.4.8 Tag List

- A) Der Titel der Tag List: Aufforderung zur Auswahl
- B) Die Tags werden nicht wie in einer Tag Cloud zufällig dargestellt sondern aufgelistet. Die Grösse der Font variiert aber je nach Anzahl Vorkommnisse der Tags.
- C) Die ausgewählten Tags werden hier aufgeführt.
- D) Durch ein Cancel Button kann man alle bereits selektierten Tags wieder aus der Auswahl entfernen. Die Auswahl der Category Selection List geht nicht verloren.

(Findings Kapitel 0)



Abbildung 59 - Tag List UI Element

4.4.9 Pattern List

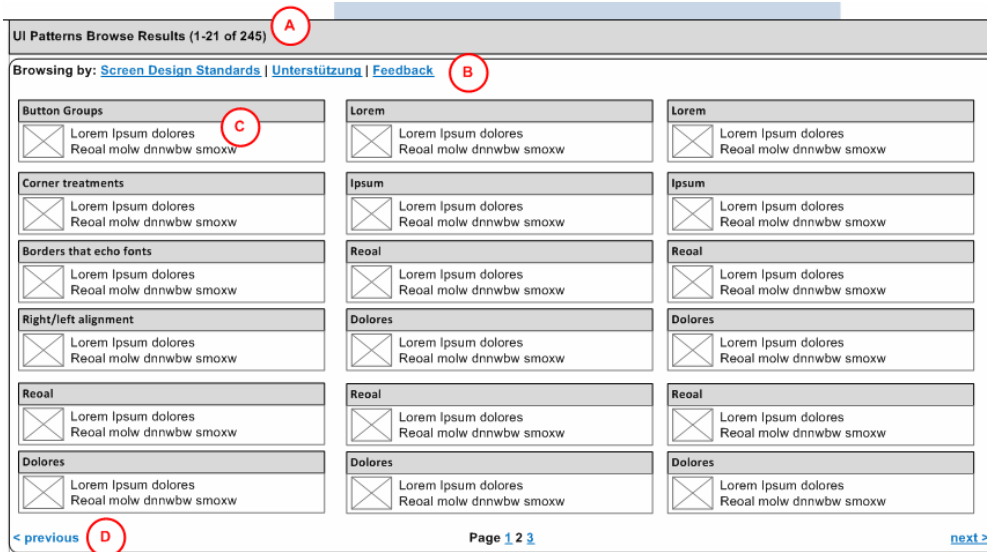


Abbildung 60 - Pattern List UI Element

- A) Im Titel wird jeweils die Anzahl der Resultate angezeigt.
- B) Die „Browsing by“ Linie zeigt alle momentan ausgewählten Filter in der gewählten Reihenfolge. Wenn ein Pattern oder ein Link über eine sekundäre Navigaton aufgerufen wird, ist dieses Navigationselement leer. Beim Selektieren auf einen vorherigen Filter (z.B. Unterstützung) werden die später ausgewählten Filter (Feedback) zurückgesetzt und die angepasste Patternliste auf dem Lösungsfindungs Screen aktualisiert.
- C) Das Pattern Detail wird mit einem Titel, einer Kurzbeschreibung und einem Bild versehen.
- D) Da nicht alle Patterns auf einen Blick dargestellt werden können gibt es eine „Pagination“.

(Findings Kapitel 4.6.1.22)

4.4.10 Shadowpath Navigation

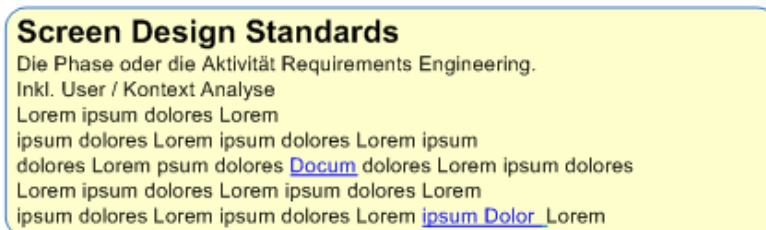


Abbildung 61 - Shadowpath Navigation UI Element

Dieses Element gibt Hinweise zum aktuell ausgewählten Thema. Beim Selektieren der Screen Design Standards Phase gibt es weiterführende Erklärungen mit Hyperlinks. Ziel dieser Navigation ist ein Lernprozess und nicht das Finden eines Patterns. Bei einer weiteren Auswahl oder Selektion eines Hyperlinks ändert sich die Shadow Path Navigation. Die Shadow Path Navigation muss von Experten erstellt und mit Phasen, Kategorien und Tags oder Patterns verknüpft werden.

(Findings Kapitel 4.6.1.4)

4.4.11 Reference Navigation

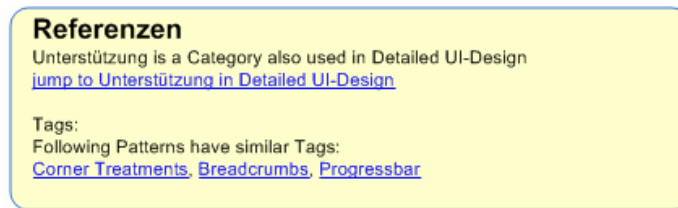


Abbildung 62 – Reference Navigation UI Element

Der Inhalt und die Links der Referenced Navigation können über die Tags automatisch berechnet werden. Beispiel: Ein Tag „direct manipulation“ kommt oft zusammen mit dem Tag „zoom in“ vor. Daraus kann man ein Link von einem Pattern das nur „zoom in“ hat zu allen Patterns die „direct manipulation“ haben generieren (Kapitel 4.3.2.). Es ist auch möglich, manuelle Verlinkungen hinzuzufügen.

4.4.12 Pattern Detail

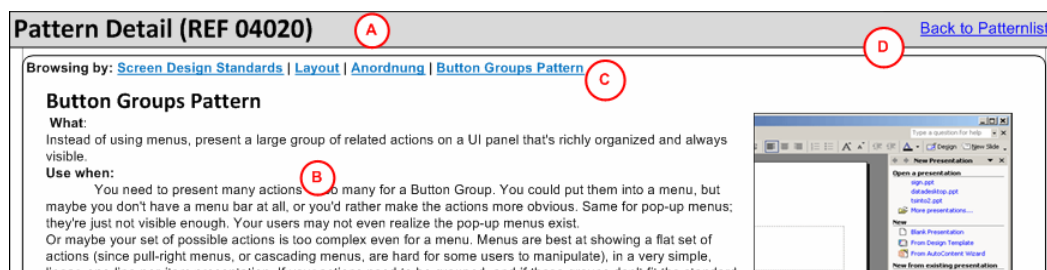


Abbildung 63 – Pattern Detail UI Element

- A) In diesem Bereich werden der Name und die eindeutige Referenznummer eines Patterns dargestellt. Diese können über die Suche mit der Referenznummer wiedergefunden werden.
- B) Der Inhalt des Patterns: Für Details zu Patterninhalten siehe Anhang 4 und 9.
- C) Die „Browsing by“ Linie zeigt alle momentan ausgewählten Filter in der ausgewählten Reihenfolge. Wenn ein Pattern oder ein Link über eine sekundäre Navigaton aufgerufen wird, ist dieses Navigationselement leer. Beim Selektieren auf einen vorherigen Filter (z.B. Unterstützung) werden die später ausgewählten Filter (Feedback) zurückgesetzt und die angepasste Patternliste auf dem Lösungsfindungs Screen aktualisiert.
- D) Ein Hyperlink „zurück“ auf den Lösungsfindungs Screen mit der Patternliste ermöglicht das zurückkommen auf die letzte Auswahl.

4.4.13 Pattern Actions

Hier werden die unterstützenden Aktionen angeboten, um das gefundene Pattern zu verwenden. Unter anderem wird es möglich sein zu drucken, Bookmarks zu setzen und das Pattern in geeigneter Form ins Clipboard (Copy & Paste) zu kopieren.

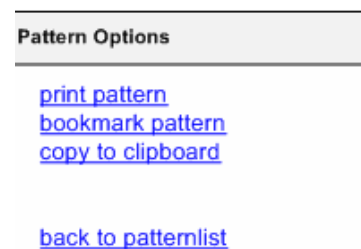


Abbildung 64 – Pattern Options UI Element

4.4.14 Pattern Tags

Auf dem „Pattern Tags“ werden die Tags des Patterns dargestellt. Optionale Erweiterung: Es wäre denkbar, dass Benutzer neue Tags erfassen können.



Abbildung 65 – Pattern Tags UI Element

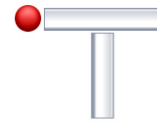
4.4.15 Prototyp aus Anwendersicht

Als Beispiel wird der primäre Task Flow aus Kapitel 4.3.1 visualisiert.

Peter S. Coder muss in seinem Design mehrere Buttons plazieren, er weiss aber nicht so recht wie. Er will im Patternfinder nachschauen, ob es zu diesem Thema ein Pattern gibt.

Die Screens befinden sich in vergrößerter Version im Anhang 8.

1. Peter S. Coder hat das Problem erkannt und ist an einer Lösung interessiert, die ihm noch nicht bekannt ist. Als erstes muss Peter S. Coder die Phase im Prozess auswählen. Damit er den Phasenentscheid fällen kann, werden ihm die nötigen Erklärungen geboten. Da Peter S. Coder nun schon seit einem halben Jahr mit dem Patternfinder arbeitet und die Designabteilung vom Entwicklungsprozess „Hermes“ auf den „Usability Engineering Lifecycle“ (Mayhew, 2008) umgestiegen ist, weiss er schon, in welcher Phase er sich mit seinem Design befindet. Er wählt die Detailed Screen Design Standards aus.



Form Title

Pattern Finder

Find a solution to your Design Problem

The Pattern Finder helps you to find a Pattern for your Design Problem

In which process phase are you ? Choose a the phase:

The Mayhew Process

Requirements Engineering	Conceptual Model	Screen Design Standards	Detailed UI-Design	Software Development	GUI Testing
<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat</p>	<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat</p>	<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat</p>	<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat</p>	<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat</p>	<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat</p>

You should now be able to choose a Projectphase Level in the Pattern Finder Guide
Are you using a different Project Process ?

[Change to RUP](#)
[Change to HERMES](#)

Last Viewed Patterns

[Breadcrumbs](#)
[Button Group](#)
[Faceted Navigation](#)

Best Rated

[Directory Navigation](#)
[Home Link](#)
[Double Tab Navigation](#)

Abbildung 66 – Phasen Auswahl Screen

2. Nach dem Einstieg in die Screen Design Standards wird Peter S. Coder der Lösungsfindungs-Screen präsentiert. Sämtliche Tag-Kategorien, die Tags selber und die dazugehörigen Patterns werden angezeigt. Die Menge ist immer noch zu gross und unüberschaubar. Peter S. Coder kann unter folgenden Gruppen auswählen. Dahinter verbergen sich Tags und Patterns.



Form Title

Pattern Finder

Find a solution to your design problem

Direct Pattern Search

Requirements Engineering
Conceptual Model
Screen Design Standards
Detailed UI-Design
Software Development
GUI Testing

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat.

Choose a Category

- Unterstützung**
Eintragen, Informationsmenge Query, Überblick...
- Design**
Ränder, Look, Colors, Images, Grösse, Curves
- Container**
Anzeige, Window, Fenster, Dialog, Gruppierung
- Layout**
Navigations, Anordnung, Positionieren
- User Needs**
Sichtbarkeit, Wichtigkeit, Zurechtfinden
- Genre**
Anwendungen ...

Choose a Tag

[Links Ausrichtung](#)
[Horizontal Kapselung](#)
[Rechtsbündig Verbindung](#)
[Prominent Nähe Regeln Gehe zu](#)
[Ipsum Dolores Rece psum Dolore](#)
[Rece psum Dolores Rece ipsumdolores](#)
[Dorem Ipsum ipsumdolores Dore](#)
[ipsumdolores Dorem ipsumdolores Dorem](#)
[Dorem Ipsum ipsumdolores Dore](#)
[ipsumdolores Dorem ipsumdolores](#)
[DoremDorem Ipsum ipsumdolores Dore](#)
[ipsumdolores Dorem ipsumdolores](#)
[DoremDorem Ipsum ipsumdolores Dore](#)

UI Patterns Browse Results (80 Patterns found)

Browsing by: [Screen Design Standards](#)

Button Groups	Lorem	Lorem
<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw
Corner treatments	Ipsum	Ipsum
<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw
Borders that echo fonts	Reaal	Reaal
<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw
Right/left alignment	Dolores	Dolores
<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw
Reaal	Reaal	Reaal
<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw
Dolores	Dolores	Dolores
<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw	<input type="checkbox"/> Lorem Ipsum dolores Reaal molw dnnwbw smoxw

< previousPage 123next >

Screen Design Standards

Die Phase oder die Aktivität Requirements Engineering.
 Inkl. User / Kontext Analyse
 Lorem ipsum dolores Lorem
 ipsum dolores Lorem ipsum dolores Lorem ipsum
 dolores Lorem psum dolores **Docum** dolores Lorem ipsum dolores
 Lorem Ipsum dolores Lorem ipsum dolores Lorem
 ipsum dolores Lorem ipsum dolores Lorem **ipsum Dolor** Lorem

Referenzen

Unterstützung ist a Category also used in Detailed UI-Design
[jump to Unterstützung in Detailed UI-Design](#)

Tags:
 Following Patterns have similar Tags:
[Corner Treatments](#), [Breadcrumbs](#), [Progressbar](#)

Abbildung 67 – Kategorien Auswahl Screen

3. Peter S. Coder sieht, dass die Kategorie „Layout“ sein Problem am besten beschreibt und wählt diese aus. Sämtliche Tag-Subkategorien, die Tags selber und die dazugehörigen Patterns der Kategorie „Layout“ werden nun angezeigt.



Form Title
Pattern Finder
Find a solution to your design problem

Requirements Engineering	Conceptual Model	Screen Design Standards	Detailed UI-Design	Software Development	GUI Testing
		Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat			

Direct Pattern Search

Choose a Category

Unterstützung
Erinnern, Informationsmenge, Query, Überblick...

Design
Ränder, Look, Colors, Images, Größe, Curves

Container
Anzeige, Window, Fenster, Dialog, Gruppierung

Layout
Navigation, Anordnung, Position

Navigation Links, Prominent, Global, Gehe zu, Verbindung...

Anordnung Kapselung, Nähe, Regeln, Gruppierung, Zusammengehörigkeit...

Position Ausrichtung, Center, Vertikal, Linksbündig, Horizontal, Rechtsbündig...

User Needs
Sichtbarkeit, Wichtigkeit, Auffinden

Genre
Anwendungen ...

Choose a Tag

Ausrichtung
[Horizontal](#), [Kapselung](#),
[Rechtsbündig](#), [Verbindung](#),
[Center](#), [Vertikal](#), [Ipsum](#), [Dolores](#),
[Recel](#), [Ipsumdolores](#), [Dorem](#), [Ipsum](#)

UI Patterns Browse Results (15 Patterns found)

Browsing by: [Screen Design Standards](#) | [Layout](#)

Button Groups Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Lorem Lorem Ipsum dolores Reoal molw dnnbw smoxw	Lorem Lorem Ipsum dolores Reoal molw dnnwbw smoxw
Corner treatments Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Ipsum Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Ipsum Lorem Ipsum dolores Reoal molw dnnwbw smoxw
Borders that echo fonts Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw
Right/left alignment Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw	
Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Reoal Lorem Ipsum dolores Reoal molw dnnwbw smoxw	
Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw	Dolores Lorem Ipsum dolores Reoal molw dnnwbw smoxw	

Page 1

Screen Design Standards

Die Phase oder die Aktivität Requirements Engineering.
Inkl. User / Kontext Analyse
Lorem ipsum dolores Lorem
ipsum dolores Lorem ipsum dolores Lorem ipsum
dolores Lorem psium dolores [Docum](#) dolores Lorem ipsum dolores
Lorem ipsum dolores Lorem ipsum dolores Lorem
ipsum dolores Lorem ipsum dolores Lorem [ipsum Dolor](#) Lorem

Referenzen

Unterstützung ist a Category also used in Detailed UI-Design
[Jump to Unterstützung in Detailed UI-Design](#)

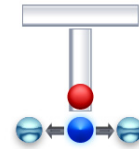
Tags:
Following Patterns have similar Tags:
[Corner Treatments](#), [Breadcrumbs](#), [Progressbar](#)

Abbildung 68 – Subkategorien Auswahl Screen

-

Abbildung 69 -Abbildung 36 – Tag Auswahl ScreenAbbildung 70 -Abbildung 36 – Pattern Auswahl Screen

5. Auf den ersten Blick sieht er das Pattern „Button Groups“ und wählt dies zur näheren Begutachtung aus. Peter S. Coder hat ein Pattern für das Plazieren von mehreren Buttons gefunden, was ihm bei der Lösung des Problems hilft.



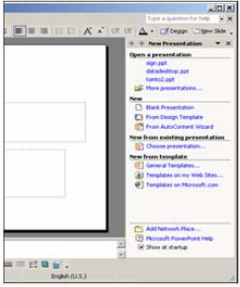
Form Title		Requirements Engineering	Conceptual Model	Screen Design Standards	Detailed UI-Design	Software Development	GUI Testing
Pattern Finder Find a solution to your design problem		Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat					
Direct Pattern Search Enter Text <input type="text"/> <input type="button" value="Search"/>							
Pattern Options print pattern bookmark pattern copy to clipboard back to patternlist		Pattern Detail (REF 04020) Back to Patternlist					
		Browsing by: Screen Design Standards Layout Position Button Groups Pattern Button Groups Pattern What: Instead of using menus, present a large group of related actions on a UI panel that's richly organized and always visible. Use when: You need to present many actions -- too many for a Button Group. You could put them into a menu, but maybe you don't have a menu bar at all, or you'd rather make the actions more obvious. Same for pop-up menus; they're just not visible enough. Your users may not even realize the pop-up menus exist. Or maybe your set of possible actions is too complex even for a menu. Menus are best at showing a flat set of actions (since pull-right menus, or cascading menus, are hard for some users to manipulate), in a very simple, linear, one-line-per-item presentation. If your actions need to be grouped, and if those groups don't fit the standard toplevel menu names -- such as File, Edit, View, Tools, or Format -- then you might want a different presentation altogether. This pattern can take up a lot of screen space, so it's not usually a good choice for small devices. Why: The "Use When" section already hinted at the two main reasons for using action panels: visibility and freedom of presentation. You could enhance the effect by underlining them when the mouse rolls over them. This is what Microsoft does in its interfaces that use action panels. Feel free to experiment. However, usability-test it to make sure users actually see the actions as clickable things, not ordinary text or pictures.  By placing the actions out on the main UI, and not hidden inside a traditional menu, you make those actions fully visible to the user. Really, action panels are menus, in the generic sense; they just aren't found in menu bars, dropdowns, or pop-ups. Users don't have to do anything to see what's on an action panel -- it's right there in front of them -- so your interface is more discoverable. This is particularly nice for users who aren't already familiar with the traditional document model and its menu bars. There are many, many ways to structure objects on an interface: lists, grids or tables, hierarchies, and just about any custom structure you can devise. But button groups and traditional menus only give you a list (and not a very long one). Action panels are freeform -- they give you as much freedom to visually organize verbs as you have for nouns. Use it wisely!					
Pattern Tags Affordance. Anordnung, Button, Gruppierung, Horizontal, Layout, Vertikal, Zusammengehörigkeit		Affordance Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum doloresdolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem		Referenzen Other Patterns have common Tags. Corner Treatments (Tags: Affordance)			

Abbildung 71 -Abbildung 36 – Pattern Detail

4.4.16 Im Prototyp verwendete Patterns

Neben den Findings aus der Analysephase, die in den Prototypen eingeflossen sind, wurden für die Auswahl und Entwicklung der UI Elemente auch Patterns verwendet (Anhang 3).

Patterns	UI Element / Screen
Welcome Page	Einstieg in den Lösungsfindungsprozess
Clear Entry Points	
Doormat Navigation	Process Phase Navigation
Trail Menu	Category Selection List
Faceted Navigation	Tag List
Tag Cloud Pattern	
Home Link	Home Link
Pagination	Pattern List
Center Stage	
Double Tab Navigation	

4.5 Pflegeprozess

Aus der Analyse ist hervorgegangen, dass ein klar definierter Pflegeprozess für den Aufbau und den Betrieb eines Tools sehr wichtig ist (Kapitel 3.1.7). In diesem Konzept wurde vorgängig bereits beschrieben, wie ein Pattern, ein Tag, eine neue Benutzergruppe (Sicht mit Vokabular der Benutzergruppe) hinzugefügt oder der Entwicklungsprozess ausgetauscht werden kann.

Zusätzlich könnten die Patterns auch versionisiert und bewertet werden, um die zeitliche Veränderung und deren Gültigkeit zu sehen.

Detaillierte Ansätze zu diesem Thema sind Anhang 10 zu finden.

4.6 Einfluss der Findings im Bezug auf Konzept und Prototyp

Legende:

EK: Einfluss auf Konzept

EP: Einfluss auf Prototyp

4.6.1.1 Inhaltliche und visuelle Struktur der Patterns ist massgebend

EK / EP	Kommentar	Finding Herkunft
- -	Die Struktur eines Patterns ist nicht Bestandteil dieser Arbeit.	Kapitel 3.1.1

4.6.1.2 Der Pattern Autor ist nicht Benutzer des Patterns

EK / EP	Kommentar	Finding Herkunft
+ +	Naming des Patterns wird via Tags den Benutzern nähergebracht. Benutzer werden beim setzen der Tags miteinbezogen	Kapitel 3.1.1

4.6.1.3 Sichtweise des Benutzers ist nicht die des Autors

EK / EP	Kommentar	Finding Herkunft
+ +	Naming des Patterns wird via Tags den Benutzern nähergebracht. Benutzer werden beim setzen der Tags miteinbezogen	Kapitel 3.1.4

4.6.1.4 Unterschiedliche Personengruppen benötigen unterschiedliche Sichten

EK / EP	Kommentar	Finding Herkunft
++ ++	Die Kategorien, Subkategorien und Tags werden von den Benutzern oder deren Vertretern selber gebildet. Verschiedene Einstiegsmöglichkeiten müssen geboten werden.	Kapitel 3.1.4

4.6.1.5 Ordnungsprinzipien haben einen hohen Stellenwert

EK / EP	Kommentar	Finding Herkunft
++ ++	Die Kategorien, Subkategorien und Tags werden von den Benutzern oder deren Vertretern selber gebildet. Verschiedene Einstiegsmöglichkeiten müssen geboten werden.	Kapitel 3.1.5

4.6.1.6 UCD-Komponente, -Aspekte und -Schichten haben Einfluss

EK / EP	Kommentar	Finding Herkunft
++ ++	Der gesamte Entwicklungsprozess und dessen UCD-Komponente fliessen in das Modell des Konzeptes ein.	Kapitel 3.1.5

4.6.1.7 Interaktionsarten, d.h. Problem bestimmen und Lösung finden

EK / EP	Kommentar	Finding Herkunft
++ +	Bildet die Basis des Modells.	Kapitel 3.1.5

4.6.1.8 Patterns sind wie Werkzeuge - ohne Einbettung in eine Methode oder einen Entwicklungsprozess sind diese kaum von Nutzen

EK / EP	Kommentar	Finding Herkunft
++ ++	Motivation der Diplomarbeit.	Kapitel □

4.6.1.9 Es ist wichtig, dass eine schrittweise, aufbauende und transparente Eingrenzung der Lösungen präsentiert wird

EK / EP	Kommentar	Finding Herkunft
++ +++	Bottom Up Ansatz des Lösungsfindungsprozesses des Konzeptes.	Kapitel 3.1.5

4.6.1.10Es gibt massgebende Verbindungen (Brother, Parent, Child) unter Patterns

EK / EP	Kommentar	Finding Herkunft
++ ++	Kapitel 4.2.5 - Alternative Navigationsmöglichkeiten	Kapitel 3.1.3

4.6.1.11Brother, Parent, Child Verbindungen müssen visualisiert werden

EK / EP	Kommentar	Finding Herkunft
+ ++	Kapitel 4.2.5 - Alternative Navigationsmöglichkeiten	Kapitel 3.1.3

4.6.1.12Es bestehen verschiedene Blickwinkel zur Überprüfung der Lösungsansätze

EK / EP	Kommentar	Finding Herkunft
+ -	Ist Bestandteil des Bottom Up Ansatzes. Dabei werden die Benutzer bei der Kategorisierung der Inhalte miteinbezogen	Kapitel 3.1.4

4.6.1.13Die Patterns sind zu textlastig

EK / EP	Kommentar	Finding Herkunft
- -	Die Struktur eines Patterns ist nicht Bestandteil dieser Arbeit. Ist aber bei der Umsetzung einer Software von Bedeutung.	Kapitel 3.2

4.6.1.14Falsche Hoffnungen werden geweckt

EK / EP	Kommentar	Finding Herkunft
++ +	Qualität und Transparent des gesamten Konzeptes ist balanciert. Sollte im Prototypdesign einfließen.	Kapitel 3.2.2.5

4.6.1.15Einstieg verwirrend

EK / EP	Kommentar	Finding Herkunft
++ ++	Durch das T-Modell des Konzeptes entschärft	Kapitel 3.2

4.6.1.16Schlechtes Labeling

EK / EP	Kommentar	Finding Herkunft
++ +	Die Kategorien, Subkategorien und Tags werden von den Benutzern oder deren Vertretern selber gebildet. Im Prototyp wurde jedoch das Labeling aus Zeitgründen noch nicht endgültig abgeschlossen.	Kapitel 3.2.2.2

4.6.1.17Schlechte Suchunterstützung

EK / EP	Kommentar	Finding Herkunft
- -	Auf ein Suchkonzept wird in dieser Arbeit verzichtet. Prototyp und Konzept konzentrieren sich auf das Finden und nicht auf das Suchen. Es wird jedoch auf (Morville & Rosenfeld, 2006) verwiesen.	Kapitel 3.2.2.3

4.6.1.18One Way Navigation

EK / EP	Kommentar	Finding Herkunft
- -	Dem sollte bei der technischen Umsetzung Rechnung getragen werden.	Kapitel Fehler! Verweisquelle konnte nicht gefunden werden.

4.6.1.19Diverse Code Fehlermeldungen

EK / EP	Kommentar	Finding Herkunft
- -	Dem sollte bei der technischen Umsetzung Rechnung getragen werden.	Kapitel 3.2.2.7

4.6.1.20 Tags und Finden über Tags

EK / EP	Kommentar	Finding Herkunft
++ ++	Ein zentrales Finding, dass das Konzept und den Prototyp beeinflusst hat	Kapitel 3.2.1.2

4.6.1.21 Topic Map

EK / EP	Kommentar	Finding Herkunft
- +	Fließt ins Design des Prototyps ein. Siehe Category Selection List – Kapitel 4.4.2 - Wireframes	Kapitel Fehler! Verweisquelle konnte nicht gefunden werden.

4.6.1.22 Ansprechende Kurzübersicht

EK / EP	Kommentar	Finding Herkunft
- +	Fließt ins Design des Prototyps ein. Siehe Pattern List – Kapitel 4.4.2 - Wireframes	Kapitel

4.6.1.23 Ranking und Patternbewertung 3.2.1.3

EK / EP	Kommentar	Finding Herkunft
- -	Wurde im Konzept und Prototyp nicht berücksichtigt – ist aber eine Erweiterungsmöglichkeit die nicht ausgeschlossen wird.	Kapitel 3.2.1.4, Kapitel 3.1.7

4.6.1.24 Globale Navigation

EK / EP	Kommentar	Finding Herkunft
- +	Fließt Teilweise ins Design des Prototyp ein. Siehe Prozess Phase Navigation – Kapitel 4.4.2 - Wireframes	Kapitel 3.2.1.6

4.6.1.25 Guide

EK / EP	Kommentar	Finding Herkunft
- +	Fließt ins das Design des Prototyps ein. Siehe Category Selection List und Tag List – Kapitel 4.4.2 - Wireframes	Kapitel 3.2.1.7

4.6.1.26 Pattern nach Aufgaben sortieren

EK / EP	Kommentar	Finding Herkunft
- -	Kann durch das Erarbeiten der Kategorien mit den Benutzern zusammen implizit einfließen.	Kapitel 3.2.1.8, Kapitel 3.1.5

4.6.1.27 Interaktive Map

EK / EP	Kommentar	Finding Herkunft
- -	Wurde im Konzept und Prototyp nicht berücksichtigt – ist aber eine Erweiterungsmöglichkeit als alternative Navigation, die nicht ausgeschlossen wird.	Kapitel 3.2.1.5

5 Das Konzept im Test - Evaluation

Das ganze Vorgehen betreffend Entscheiden und Ergebnissen wurde iterativ ausgelegt und alle Resultate getestet (Kapitel 2.1). Diese Schritte dienten der Sicherstellung der Praxistauglichkeit, sowie der Entscheidungsfindung. Es wurden folgende Evaluationen durchgeführt:

5.1 Erster Test – Early-Prototyp

Für den ersten Test wurden die sogenannten Early-Prototypen verwendet. Diese Prototypen entstanden intuitiv und hatten den Zweck, Ideen zu sammeln und im Team zu vermitteln. Die Prototypen wurden dem Team vorgestellt. Diese Prototypen zusammen mit den Feedbacks stellten den Einstieg in die Design-Phase dar.

Die folgenden Abbildungen geben einen kleinen Einblick in die ersten Prototypen, welche die Grundlage für die ersten Test lieferten. Aus den unterschiedlichen Ansätzen entstand die Idee des T-Models, welche im Kapitel 4 erläutert wurden.

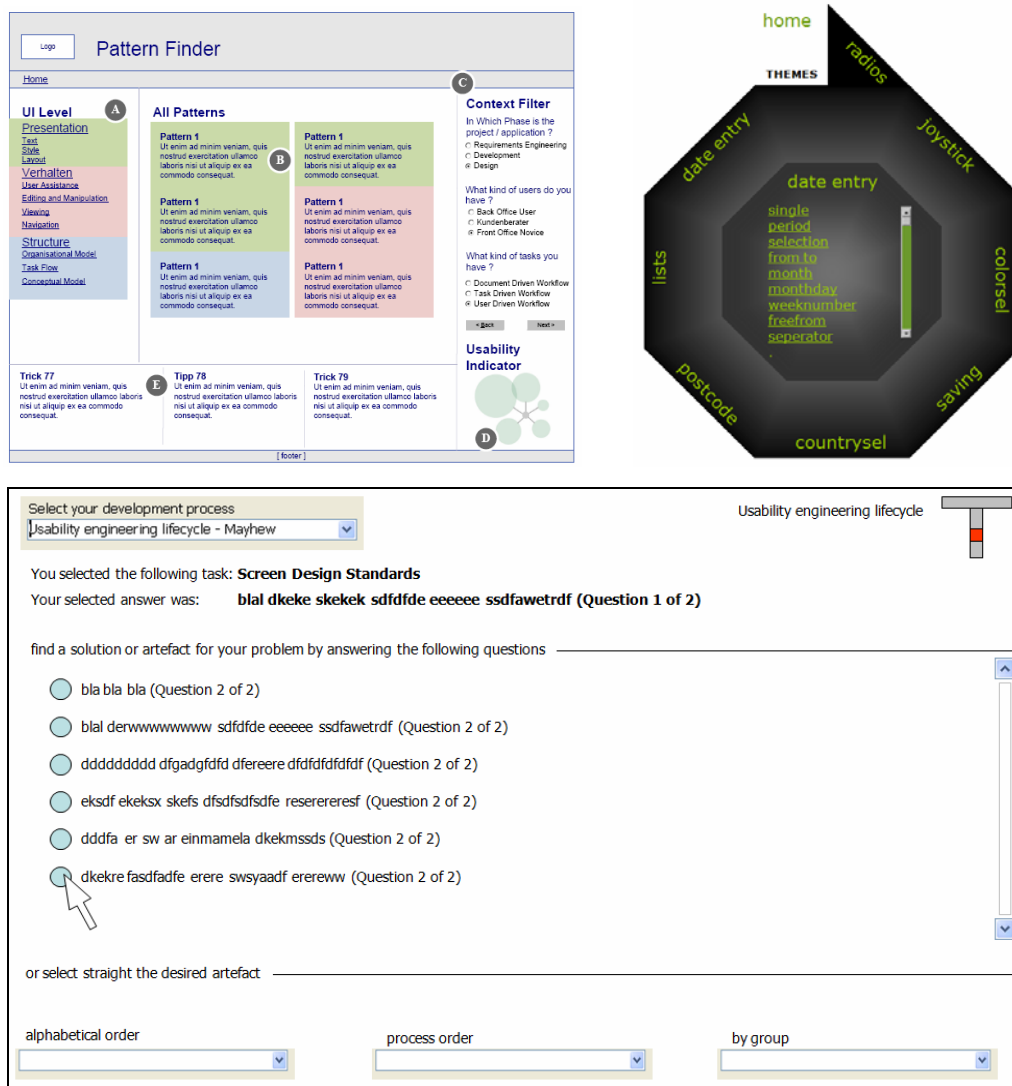


Abbildung 72 - Early Prototypen 1 bis 3

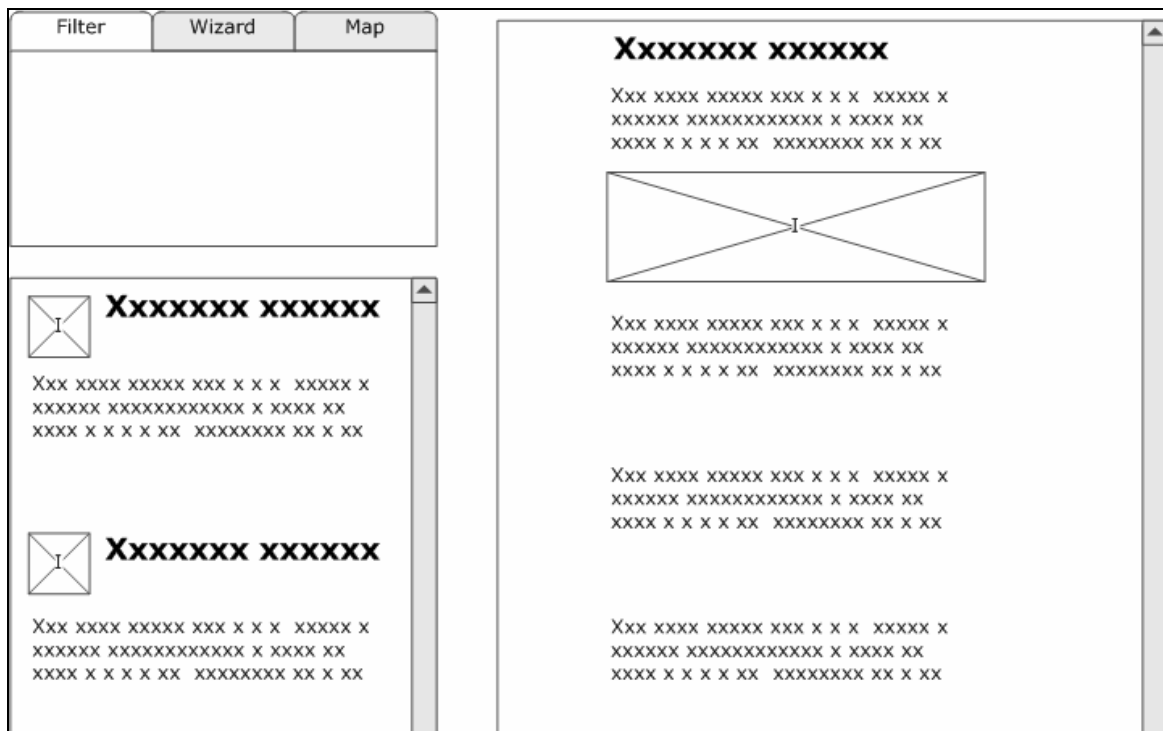


Abbildung 73 - Early Prototyp 4

5.2 Zweiter Test – erster konkreter Prototyp

Der zweite Test wurde mittels Usability Test am ersten konkreten Prototypen durchgeführt. Als Testpersonen standen uns Studienkollegen aus dem Lehrgang MAS HCID zur Verfügung. Es wurden insgesamt drei Test durchgeführt.

Der Prototyp wurde in Visio erstellt, ausgedruckt und in Bereiche zerschnitten. Mit diesen Stücken, welche einzelne Bereiche aus dem UI abbildeten, konnten die Interaktion zwischen Testperson und System simuliert werden. Neben der Testperson und dem Testleiter war eine weitere Person als Beobachter in die Tests involviert. Die Ergebnisse wurden in Protokollen festgehalten und flossen in die nachfolgenden Arbeiten ein (Anhang 6.1 und 6.2).

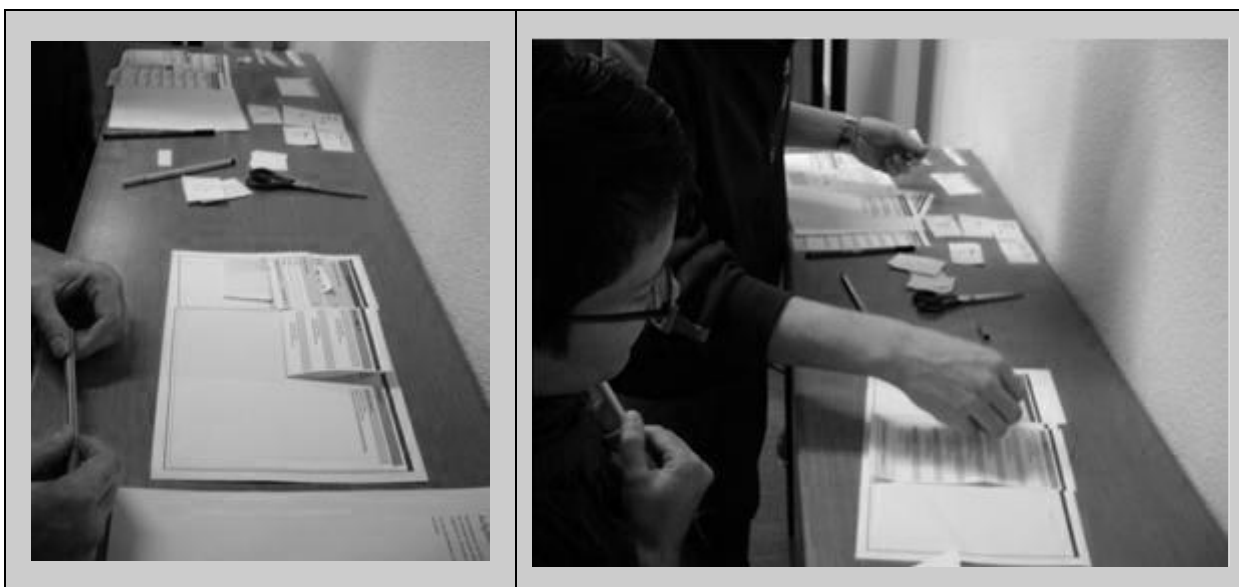


Abbildung 74 – Der zweite Prototyp im Test

5.3 Dritter Test – zweiter konkreter Prototyp

Der zweite konkrete Prototyp basiert auf dem ersten Prototypen. Dieser Prototyp wurde ebenfalls in Visio erstellt. Der Test wurde in Form eines Walkthrough durchgeführt. Die Ergebnisse wurden anschliessend in den Prototypen eingearbeitet.

Screen Design Standards

Choose a Category	UI Patterns 11-18 from 30	Guidelines (3)
Unterstützung <small>Sanduhr, Affordance, Erinnern, Informationsmangel, Feedback, Progressbar, Dauer, Überblick, Query, Warten, Help, Pause, Tooltips, Online Ressourcen, Kontexthilfe</small>	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div> <div style="background-color: #d9d9d9; padding: 2px;"> Datatips Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div> <div style="background-color: #d9d9d9; padding: 2px;"> Auto completion Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div> <div style="background-color: #d9d9d9; padding: 2px;"> Illustrated choices Lorem Ipsum dolores Reoal molw dnnwbw smoxw </div>	<div style="background-color: #d9d9d9; padding: 2px;"> Styleguide Lorem Ipsum dolores Reoal molw dnnwbw smoxw C:\Documents and Settings\knube\Desktop\downloads\ </div> <div style="background-color: #d9d9d9; padding: 2px;"> CI / CD Guidelines Lorem Ipsum dolores Reoal molw dnnwbw smoxw C:\Documents and Settings\knube\Desktop\downloads\ </div> <div style="background-color: #d9d9d9; padding: 2px;"> Wording Guidelines Lorem Ipsum dolores Reoal molw dnnwbw smoxw C:\Documents and Settings\knube\Desktop\downloads\ </div> <div style="background-color: #d9d9d9; padding: 2px;"> Methoden (2) </div> <div style="background-color: #d9d9d9; padding: 2px;"> Moodboard Lorem Ipsum dolores Reoal molw dnnwbw smoxw C:\Documents and Settings\knube\Desktop\downloads\ </div> <div style="background-color: #d9d9d9; padding: 2px;"> Prototyping Lorem Ipsum dolores Reoal molw dnnwbw smoxw C:\Documents and Settings\knube\Desktop\downloads\ </div>
Choose Tags	<< show previous 7 show next 7 >>	
Sanduhr, Affordance, Erinnern, Informationsmangel, Feedback, Progressbar, Dauer, Überblick, Query, Warten, Help, Pause, Tooltips, Online Ressourcen, Kontexthilfe choosen Tags: -	Screen Design Standards Die Phase oder die Aktivität Requirements Engineering. Inkl. User / Kontext Analyse Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores ipsum dolores Lorem ipsum dolores Lorem ipsum dolores ipsum dolores Lorem ipsum dolores Docum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores Lorem ipsum dolores ipsum dolores Lorem ipsum dolores Lorem ipsum dolores ipsum Dolor Lorem	Aussenbeziehungen Unterstützung is a Category also used in Detailed UI-Design Tags: Progress Bar, Informationsmenge, Query, Überblick, Personalisierung, Affordance, Sanduhr, Dauer, warten, Pause, Erinnern jump to Unterstützung in Detailed UI-Design

Abbildung 75 – Der dritte Prototyp

5.4 Vierter Test – Dritter konkreter Prototyp

Der vorliegende Prototyp (Anhang 8) liefert die Grundlage für den Abschlusstest. Dieser konnte im Rahmen der Masterarbeit aus Zeitgründen nicht durchgeführt werden. Der Prototyp ist für diesen Test vorbereitet.

Es wird empfohlen, nachfolgende Überlegungen in den Test mit einzubeziehen:

- Definition von Testfällen: Es sollen Fragestellungen aus echten oder virtuellen Projekten anhand der Szenarios (Kapitel 3.3.2) erarbeitet werden, welche durch die Testperson im Rahmen der Tests zu lösen sind.
- Auswahl von 10 bis 20 Patterns und Implementierung derselben in das System
- Erweiterung des Prototypen zu einem interaktiven Prototypen
- Entwerfen eines kurzen Fragebogens mit offenen und geschlossenen (Likert-Skala) Fragen. Inhaltlich können diese auf SUMI (Software Usability Measurement Inventory) oder ISO basieren, sich auf spezifische Funktionalitäten beziehen oder allgemein den Eindruck der Testperson erfragen.
- Festhalten von Fragestellungen bezüglich Erkenntnisgewinn: welche Aspekte sollen worauf geprüft werden

Zur Evaluation wird ein Usability Test mit mindestens fünf Personen (Fachspezialist, Entwickler, UI Designer, HCID-Spezialist) vorgeschlagen. Die Stärken und Schwächen des Prototypen sollen mit verschiedenen, teilweise parallelen Methoden wie Beobachtung beim Usability Test, Aufzeichnung der Interaktionen, Expert Review und Interview der Testpersonen erhoben werden. Anschliessend können die Ergebnisse aus diesen unterschiedlichen Quellen gesammelt, nach Aspekten gruppiert und in einer Matrix aus Aspekten und Testmethoden auf Korrelation untereinander untersucht werden. Dadurch können die Faktoren Benutzer, Umfeld und Aufgabe angemessen getestet werden. Es wird so geprüft, ob und wie die Aufgaben gelöst werden können, welchen Eindruck die Testpersonen vom Prototypen haben und welche Verbesserungsvorschläge bezüglich Praxisnutzen sie einbringen.

6 Schlussfolgerung

Bereits bei der Formulierung der Aufgabenstellung zeigten sich die Komplexität und der Schwierigkeitsgrad des Projektes. Während der Umsetzung der Aufgabe stellte sich heraus, dass es nicht nur eine richtige Lösung für die Problemstellung gibt. Nachhaltige Nachforschungen und Analysen führten zu der, in dieser Masterarbeit beschriebenen, Lösung.

Die Lösung zeichnet sich durch folgende Eigenschaften aus:

- **Verschiedene Anwendersichten** - Jede Anwendergruppe hat aufgrund ihres Wissenstandes, sowie ihres Vokabulars eine individuelle Sicht auf eine Problemstellung. Solche unterschiedlichen Sichten werden durch dieses Konzept unterstützt, sodass eine grosse Benutzervielfalt einen optimalen Nutzen hat.
- **Verschiedene Prozesse** - Der horizontale Balken im T-Modell kann unterschiedliche Prozesse abbilden. Dadurch kann das Konzept in verschiedenen Unternehmungen in ihre individuellen Prozesse eingebettet werden.
- **Verschiedene Inhalte** - Als Inhalte können neben Patterns auch Auszüge aus Guidelines und weiterführende Ergebnisse eingebunden werden.
- **Anwender muss kein HCID-Spezialist sein** - Wer mit diesem Tool (Konzept nach der Realisierung) arbeitet, braucht über keine fundierten HCID-Kenntnisse zu verfügen. Der Anwender wird anhand seines Wissenstandes zur geeigneten Lösungen geführt.
- **Flexibilität** - Die Inhalte sind nicht starr zugeordnet. Das Konzept sieht vor, dass diese Inhalte aus verschiedenen Problemstellungen heraus gefunden und verwendet werden können.
- **Bottom Up Ansatz** - Dieser Ansatz baut darauf, die einzelnen Inhalte zu charakterisieren. Diese Charakteristiken werden jeweils nach den Bedürfnissen verschiedener Benutzergruppen gruppiert. Diese Gruppierung bildet zugleich die Hierarchie ab.
- **Dynamischen Hierarchie** - Die bereits erwähnte Flexibilität bringt den Vorteil einer dynamischen Hierarchie mit sich. Diese Hierarchie muss nicht nachgeführt werden. Sie ergibt sich aufgrund des Bottom Up Ansatzes automatisch aus den gebildeten Charakteristiken.
- **Erweiterbarkeit** - Das Konzept lässt eine grosse Erweiterbarkeit zu. Neue Inhalte sowie weitere Anwendersichten für neue Benutzergruppen können laufend hinzugefügt werden.

6.1 Nutzen für das Unternehmen

Das Tool, welches aus diesem Konzept realisiert werden kann, bringt einem Unternehmen einen konkreten Nutzen durch den Einsatz von Pattern. Dies funktioniert aber nur, wenn eine Person:

- weiss, wo diese Pattern zu finden sind.
- jeweils das richtige Pattern auswählt.
- weiss, wie mit einem Pattern umzugehen ist.

Alle diese Punkte führen zum Schluss, dass die Patterns geordnet werden müssen. Diese Ordnung muss auf einzelne Benutzergruppen abgestimmt und optimiert werden. Dadurch werden die Pattern auf einmal für eine grosse Benutzervielfalt zugänglich. Dies wird mit dem PatternFinder erreicht.

Folgende Vorteile ergeben sich daraus:

- **Effizienzsteigerung beim UI Design** - Die UI's werden einheitlicher und nach bekannten Richtlinien konzipiert. Bauchentscheide kommen seltener vor. Das UI Design basiert auf fundiertem und bewährtem Wissen.
- **Ein Pattern unterstützt durch ein gemeinsames Verständnis die Kommunikation zwischen den einzelnen Fachbereichen** - Ein Pattern kann in einer Spezifikation als Referenz eingesetzt werden.
- **Bestehende Guidelines können ausschnittsweise eingesetzt werden** – und anstelle der Patterns als Lösung präsentiert werden.
- **Die Mitarbeiter profitieren von implizitem Expertenwissen.**
- **UI Design Disziplinen können in bestehende Entwicklungsprozesse eingebettet werden.**

6.2 Empfehlungen für das weitere Vorgehen

6.2.1 Usability Test

Während der Projektarbeit wurden die jeweiligen Ergebnisse laufend methodisch getestet (Kapitel 5). Die Testresultate wurden ständig in die nachfolgenden Entwicklungsstufen eingearbeitet. Die Menge der Testresultate war grösser als erwartet und die Einarbeitung dieser Resultate in den Prototypen nahm mehr Zeit in Anspruch als zu Beginn des Projektes geplant. Dieser Umstand führte dazu, dass der aktuelle Prototyp nicht abschliessend getestet werden konnte. Um die Qualität des endgültigen Prototypen sicherstellen zu können, empfehlen wir ausdrücklich, diese Tests durchzuführen. Allfällig weitere Ergebnisse aus diesem Test sind einzuarbeiten.

6.2.2 Grafisches Design ausarbeiten

Neben dem bereits vorhandenen konzeptionellen Design und Interaktionsdesign muss das grafische Design erarbeitet werden.

6.2.3 Realisierung

In dieser Arbeit wurden die Ergebnisse in kleinen Iterationen entwickelt. Für die Umsetzung des Tools empfehlen wir, dieses Vorgehen beizubehalten. Damit lässt sich die Qualität der Arbeit stets sicherstellen.

6.2.4 Inhalte erarbeiten

Die Inhalte (Pattern, Guidelines und weiterführende Ergebnisse...) des aktuellen Prototypen wurden für die Erarbeitung und Tests des PatternFinder Konzepts zusammengestellt und sind nicht vollständig. Grundsätzlich ist der PatternFinder ein inhaltsunabhängiges Konzept, welches für jede Anwendergruppe individuell ergänzt werden kann oder muss.

Im Bezug auf das T-Modell muss zusätzlich ein horizontaler Prozess definiert werden (Kapitel 4.2). Dazu eignet sich der bereits im Betrieb eingesetzte Entwicklungsprozess. Grundsätzlich ist die Wahl dieses Prozesses frei, dieser sollte jedoch gelebt werden.

6.2.5 Betriebskonzept

Das Betriebskonzept beschreibt die Pflichten und Rechte unterschiedlicher Benutzer. Dieses Betriebskonzept beinhaltet einerseits die Wartung und andererseits den Zugriff der eigentlichen User auf das spätere Tool. Dieses Konzept ist nicht Bestandteil dieser Arbeit und muss zusätzlich erstellt werden.

7 Unsere Erfahrungen

Manuel Schaffner

Für mich war es sehr interessant und lehrreich, einmal eine wissenschaftlich-theoretische Arbeit inkl. Literaturstudium zu machen, wie dies in der HCID-Praxis nur selten vorkommen wird. Inhalt war, ein Konzept zu finden, das sich in der Praxis bewährt. Dabei war es herausfordernd, in der Fülle von Infos weder den Überblick noch wichtige Infos zu verlieren. Schlussendlich ist es uns aber gelungen, einen neuen Ansatz für den Praxiseinsatz von Pattern zu finden.

Die Intensivwoche in Spanien war sehr hilfreich, da konzentriert zusammengearbeitet werden konnte und durch die schöne Umgebung eine kurze Erholungsphase (kurzes Abschalten) möglich war. Nach der Analyse wurde hier der inhaltliche Durchbruch der Lösung erarbeitet.

Dass die ungewohnte Art des wissenschaftlichen Schreibens am Schluss noch einen Zusatzaufwand benötigt, wurde uns bereits am Schreibkurs vor Augen geführt.

Die Zusammenarbeit zwischen Kollegen benötigte zwar zusätzliche Diskussionen, brachte aber ebenso wertvolle unterschiedliche Ansichten zu Tage und wir halfen uns gegenseitig, nicht in einer Sackgasse zu landen. Es hat gut geklappt und Spass gemacht.

Roland Schild

Mit grosser Bereitschaft und Motivation wurde die Masterarbeit in Angriff genommen. Jedoch auch mit einer gewissen Naivität. Die Kräfte liessen gegen Ende der Arbeit (auch bedingt durch meinen Job) deutlich nach. Ich war freudig erstaunt, dass ich nach der ersten gröberen Analyse und Kritik durch den Coach noch einmal Kräfte für den nötigen Endsprint aufbringen konnte.

Einer der Highlights, auch aus Sicht der Ergebnisse, war die Intensiv Woche auf Mallorca. Eine ganze Woche mit der Materie hat uns den Durchbruch gebracht.

Die Team Arbeit war angenehm aber auch teilweise schwierig, da vom totalen Pragmatist bis zum kleinlichsten Pedant alle Charaktere im Team vorhanden sind.

Markus Oberle

Die Masterarbeit war für mich persönlich eine grosse Herausforderung. Bereits zu Beginn der Arbeit war mir bewusst, dass wir uns auf ein sehr breites Themengebiet einlassen würden. Zwischenzeitlich war es sicher auch nicht ganz einfach, den Fokus nicht aus den Augen zu verlieren. Die Intensivwoche Anfangs November 2008 in Mallorca war sicherlich einer der Höhepunkte. Das angenehme Mittelmeerklima und die Abgeschiedenheit liess ein effizientes Arbeiten zu, ohne dass die Gemütlichkeit zu kurz kam. In dieser Woche konnten die grössten Fortschritte erzielt werden. Dank der Motivation und der Freude an der Sache war es gerade mit meinen Teamkollegen eine schöne, interessante und auch sehr lehrreiche Zeit. Ich möchte mich bei Thomas, Manuel und Roland für die tolle Zusammenarbeit bedanken und wünsche uns allen einen erfolgreichen Abschluss und weiterhin beruflich wie auch privat viel Erfolg.

Thomas Kneubuehl

Jetzt haben wir ein Prototyp und er sieht so aus wie so manche andere Webseite - Aber wenn ich reflektiere, steckt dahinter wirklich viel Arbeit und ein gutes Konzept. Es gibt ähnliche Webseiten die auch das Problem des Findens von grossen Mengen von Daten haben. Manche Ansätze haben auch Ähnlichkeiten mit unseren. Doch sind sie nicht die selben.

Bei Delicious.com werden auch Tags verwendet um zu der Webseite zu kommen die wir suchen aber im Vorherein nicht kennen. Auch bei Epicurious.com gibt es eine Navigation über Attribute.

Doch unser Ansatz geht noch weiter und schliesst somit auch eine Lücke im HCID Bereich.

Man könnte noch Jahre weiterforschen sich um die Patterns und deren Inhalt mehr Gedanken machen. Das Thema Patterns ist riesig und interessant.

Wenn der PatternFinder wirklich realisiert und ausgebaut und abgefüllt wird, ist eine solche einfach zugänglich Sammlung von konzentriertem Expertenwissen ein wahrer Schatz für jeden UI Designer.

Die Masterarbeit zu viert zu schreiben war eine grosse Herausforderung. Es war schwierig und benötigte einige Diskussionen um sich wieder gemeinsam den selben Fokus zu haben. Es gibt auch Vorteile in Workshops und im kreativen Design Prozess wenn man verschiedenen Sichten einbringen kann. Die Intensiv-Woche in Spanien ohne Internet war eine super Idee und hat uns auch einen riesen Schritt weitergebracht.

8 Verzeichnisse

8.1 Studenten

Thomas Kneubühl, 32
PostFinance, Bern

Markus Oberle, 31
PMI.AG, Aarau

Manuel Schaffner, 37
BIT, Bern

Roland Schild, 48
Credit Suisse, Bern



Abbildung 76 - Thomas
Kneubühl



Abbildung 77 - Markus
Oberle



Abbildung 78 - Manuel
Schaffner



Abbildung 79 - Roland
Schild

8.2 Literatur und Quellen

Alexander, C. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press.

Baxley, B. (2002). *Making the Web Work*. New Riders Publishing.

Behrens, C. (2008). *Information Design Patterns*. Abgerufen am 18. Juli 2008 von <http://interface.fh-potsdam.de/infodesignpatterns/patterns.php>

Bergin, J. (2007). *Fourteen Pedagogical Patterns*. Von <http://csis.pace.edu/~bergin/PedPat1.3.html> abgerufen

Bolchini, D., & Community. (kein Datum). *Hypermedia Design Patterns Repository*. Abgerufen am 18. Juli 2008 von <http://www.designpattern.lu.unisi.ch/PatternsRepository.htm>

Borchers, J. (2001). *A Pattern Approach to Interaction Design*. Wiley.

Brierley, M. (kein Datum). *UI Design Patterns Library*. Abgerufen am 18. Juli 2008 von Sakai Library: <http://bugs.sakaiproject.org/confluence/display/DESPAT/Library>

Community. (kein Datum). *Open Source Design Pattern Library*. Abgerufen am 18. Juli 2008 von <http://uidesignpatterns.org>

Cooper, A. (2007). *About Face 3.0*. John Wiley & Sons.

Coram, T., & Lee, J. (kein Datum). *Experiences - A Pattern Language for User Interface Design*. Abgerufen am 8. Juli 2008 von <http://www.maplefish.com/todd/papers/experiences/Experiences.html>

Deng, J., Kemp, E., & Todd, E. (2005). *Managing UI Pattern Collections*. Abgerufen am 18. Juli 2008 von <http://delivery.acm.org/10.1145/1080000/1073951/p31-deng.pdf>

Eckstein, J. (2000). *Learning to Teach and Learning to Learn - Running a Course*. Abgerufen am 18. Juli 2008 von <http://www.pedagogicalpatterns.org/examples/LearningAndTeaching.pdf>

Erickson, T. (1998). *Supporting Interdisciplinary Design: Towards Pattern Languages for Workplaces*. Abgerufen am 18. Juli 2008 von <http://www.visi.com/~snowfall/Patterns.WrkShpRep.html>

Erickson, T. (1997). *Towards a Pattern Language for Interaction Design*. Abgerufen am 18. Juli 2008 von <http://www.visi.com/~snowfall/Patterns.Chapter.html>

Fachhochschule Potsdam, F. D., & Community. (2008). *Pattern Browser*. Abgerufen am 18. Juli 2008 von <http://interface.fh-potsdam.de/patternBrowser/code/pattern/pattern.php?4,1,1,1,1,8>

Fincher, S. (2002). *Patterns for HCI and Cognitive Dimensions: two halves of the same story?*. Abgerufen am 18. Juli 2008 von <http://www.visi.com/~snowfall/FincherOnPatterns.pdf>

Fincher, S., & Windsor, P. (2000). *Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design*. Abgerufen am 18. Juli 2008 von <http://www.cs.kent.ac.uk/people/staff/saf/patterns/chi00.pdf>

Folmer, E. (kein Datum). Abgerufen am 2008. Juli 2008 von Interaction Design Pattern Library for Games: <http://www.helpyouplay.com>

Gaffar, Javahery, Seffah, & Sinnig. (2003). *A Pattern Framework for Eliciting and Delivering UCD Knowledge and Practices*. Abgerufen am 18. Juli 2008 von <http://www.dsinnig.com/pdfs/Patterns.pdf>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Garrett, J. J. (2000). *The Elements of User Experience*. Peachpit Press.

Glushko, B. (kein Datum). *Web Patterns*. Abgerufen am 18. Juli 2008 von Center for Document Engineering, U.C. Berkeley School of Information:
http://groups.ischool.berkeley.edu/ui_designpatterns/webpatterns2/webpatterns/home.php

Graham, I. (2002). *A Pattern Language for Web Usability*. Abgerufen am 18. Juli 2008 von
<http://www.trireme.com/WU/home.htm>

Granlund, A., Lafrenière, D., & Carr, D. A. (2001). *A Pattern-Supported Approach to the User Interface Design Process*. Abgerufen am 18. Juli 2008 von
<http://www.cdt.luth.se/%7Edavid/papers/HCIInt2001Final.pdf>

Greene, S. L., Matchen, P. M., Jones, L., Thomas, J. C., & Callery, M. (2003). *Tool-based Decision Support for Pattern Assisted Development*. Abgerufen am 18. Juli 2008 von
<http://www.cs.kent.ac.uk/people/staff/saf/patterns/chi2003/submissions/greeneMatchenJonesThomasCallery.doc>

Griffiths, R. N., & Pemberton, L. (1998). *Teaching Usability Design Through Pattern Language*. Abgerufen am 18. Juli 2008 von <http://www.it.bton.ac.uk/staff/lp22/CHIpaper.html>

Guy, E. S. (2004). *Appropriating Patterns for the Activity Theory Toolkit*. Abgerufen am 18. Juli 2008 von
http://www.cmis.brighton.ac.uk/staff/esg1/papers/ATIT2004_Proceedings_Appropriating%20Patterns.pdf

Henniger, S., Keshk, M., & Kinworthy, R. (2003). *Capturing and disseminating usability patterns with semantic web technology*. Abgerufen am 18. Juli 2008 von
<http://www.cs.kent.ac.uk/people/staff/saf/patterns/chi2003/submissions/Henninger-patterns-workshop.doc>

Holger Kampffmeyer, S. Z. (2007). *Finding the Pattern You Need: The Design Pattern Intent Ontology*. Abgerufen am 18. Juli 2008 von http://www.steffen-zschaler.de/publications/models07_dprio.pdf

Hughes, M. (2007). *A Pattern Language for User Assistance*. Abgerufen am 18. Juli 2008 von
<http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery/p27-hughes.pdf>

Janietz, S.-U. (2005). *Patterns & Pattern-Languages in der Mensch-Computer-Interaktion*. Abgerufen am 18. Juli 2008 von <http://www.suja.de/Text/?id=90>

Juan Deng, E. K. (2006). *Focussing on a standard pattern form: the development and evaluation of MUIP*. Abgerufen am 18. Juli 2008 von <http://portal.acm.org/citation.cfm?id=1152771>

Kuittinen, J. (kein Datum). *Gameplay Design Patterns*. Abgerufen am 18. Juli 2008 von
<http://www.gameplaydesignpatterns.org>

Kurt, S., & Ebert, D. (2004). *Pattern Languages in Interaction Design*. Abgerufen am 18. Juli 2008 von
<http://skurt.de/wp-content/uploads/2007/09/semmodben.pdf>

L., I. M. (2003). *Patterns for Personal Web Sites*. Abgerufen am 18. Juli 2008 von
<http://www.rdrop.com/%7Ehalf/Creations/Writings/Web.patterns/index.html>

Laakso, S. A. (2003). *User Interface Design Patterns*. Abgerufen am Juli. 19 2008 von
<http://www.cs.helsinki.fi/u/salaakso/patterns/index.html>

Lammi, J. (2008). *UI Pattern Factory*. Abgerufen am 20. Juni 2008 von <http://uipatternfactory.com>

Lancock, M., Malone, E., & Wheeler, C. (2005). *Implementing a Pattern Library in the Real World: A Yahoo! Case Study*. Abgerufen am 18. Juli 2008 von http://iasummit.org/2005/finalpapers/52_Presentation.pdf

Mahemoff, M., & Johnston, L. (1998). *Principles for a Usability-Oriented Pattern Language*. Abgerufen am 18. Juli 2008 von <http://mahemoff.com/paper/principles/>

Mahemoff, M., & Lorraine, L. (1998). *Pattern Languages for Usability: An Investigation of Alternative Approaches*. Abgerufen am 18. Juli 2008 von <http://mahemoff.com/paper/candidate/>

Martin, D., Rodden, T., Sommerville, I., Rouncefield, M., & Hughes, J. (kein Datum). *Patterns of Cooperative Interaction*. Abgerufen am 19. Juli 2008 von <http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/pointer.html>

Mayhew, D. J. (2008). *The Usability Engineering Lifecycle*. Morgan Kaufmann.

Molina, P. J. (2007). *Conceptual User Interface Patterns*. Von <http://pjmolina.com/cuip/> abgerufen

Morville, P., & Rosenfeld, L. (2006). *Information Architecture for the World Wide Web, Third Edition*. O'Reilly Media. O'Reilly Media.

Pedro J. Molina, S. M. (2002). *User Interface Conceptual Patterns*. Abgerufen am 18. Juli 2008 von <http://www.dlsi.ua.es/~santi/papers/UIConceptualPatterns.pdf>

Perzel, K., & Kane, D. (1999). *Usability Patterns for Applications on the World Wide Web*. Abgerufen am 18. Juli 2008 von http://jerry.cs.uiuc.edu/~plop/plop99/proceedings/Kane/perzel_kane.pdf

Radeke, F., Forbrig, P., Seffah, A., & Sinnig, D. (2006). *PIM Tool: Support for Pattern-Driven and Model-Based UI Development*. Abgerufen am 18. Juli 2008 von http://www.dsinnig.com/pdfs/PIM_final.pdf

Richter, M. (2007). *Usability Engineering kompakt*. Heidelberg: Spektrum Akademischer Verlag.

Schmettow, M. (2005). *Towards a Pattern Based Usability Inspection Method for Industrial Practitioners*. Abgerufen am 18. Juli 2008 von http://www.se-hci.org/bridging/interact2005/03_Schmettow_Towards_UPI.pdf

Schuler, D., & Community. (2002-2008). *Public Sphere Project*. Abgerufen am Juli. 18 2008 von <http://www.publicsphereproject.org/patterns/pattern.pl/public>

Scott, B. (2005). *Mind mapping Design patterns*. Abgerufen am 18. Juli 2008 von <http://looksgoodworkswell.blogspot.com/2005/05/mind-mapping-design-patterns.html>

Seffah, A., & Javahery, H. (2002). *On the Usability of Usability Patterns*. Abgerufen am 18. Juli 2008 von http://www.adana1.tesweb.org/dyn_files/sites/27/files/45.pdf

Shneiderman, B. (1997). *Designing the User Interface. Strategies for Effective Human-Computer Interaction*. Addison-Wesley.

Sinnig, D., Gaffar, A., Seffah, A., & Forbrig, P. (2004). *Patterns, Tools and Models for Interaction Design*. Abgerufen am 18. Juli 2008 von <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-103/sinnig-et-al.pdf>

Stimmel, C. L. (1999). *Hold Me, Thrill Me, Kiss Me, Kill Me*. Abgerufen am Juli. 18 2008 von Patterns for developing Effective Concept Prototypes: <http://jerry.cs.uiuc.edu/plop/plop99/proceedings/stimmel/HoldMeThrillMeKissMeKillMe1.PDF>

Stimmt.AG. (kein Datum). *Uni Basel Pattern Catalogue*. Abgerufen am 18. Juli 2008 von http://interaction-patterns.org/wiki/index.php?title=Main_Page

Tidwell, J. (1997). *COMMON GROUND: A Pattern Language for HCID*. Abgerufen am Juli. 9 2008 von http://www.mit.edu/~jtidwell/common_ground.html

Tidwell, J. (kein Datum). *Designing Interfaces: Patterns for Effective Interaction Design*. Abgerufen am 8. Juli 2008 von <http://designinginterfaces.com>

Tidwell, J. (2005). *Designing Interfaces: Patterns for Effective Interaction Design*. In J. Tidwell. O'Reilly Media, Inc.

Timmesfeld, F., Schmitt, S., Vucica, M., & Schreiner, M. (2005). *PatternCube*. Abgerufen am 18. Juli 2008 von <http://patterncube.com>

Toxboe, A. (kein Datum). *User Interface Design Patterns*. Abgerufen am 18. Juli 2008 von <http://ui-patterns.com>

van Duyne, D., Hong, J., & Landay, J. (2006). *The Design of Sites*. Abgerufen am 19. Juli 2008 von <http://www.designofsites.com>

van Duyne, D., Landay, J., & Hong, J. (2006). The Design of Sites. In *Patterns for Creating Winning Web Sites*. Prentice Hall PTR.

van Welie, M. (kein Datum). *A Pattern Library for Interaction Design*. Abgerufen am 8. Juli 2008 von <http://www.welie.com/patterns>

van Welie, M. (2000). *Interaction Patterns in User Interfaces*. Abgerufen am 18. Juli 2008 von <http://www.cs.vu.nl/~martijn/patterns/PLoP2k-Welie.pdf>

van Welie, M., & van der Veer, G. C. (2003). *Pattern Languages in Interaction Design: Structure and Organization*. Abgerufen am 18. Juli 2008 von <http://www.welie.com/papers/Welie-Interact2003.pdf>

van Welie, M., van der Veer, G. C., & Eliens, A. (2000). *Patterns as Tools for UI Design*. Abgerufen am 18. Juli 2008 von <http://www.cs.vu.nl/%7Emartijn/gta/docs/TWG2000.pdf>

Waloszek, G. (2003). *Introduction to User Interface Patterns at SAP*. Abgerufen am 18. Juli 2008 von http://www.sapdesignguild.org/community/design/patterns_sap.asp

Welie, M. v. (2000). *Interaction Patterns in User Interfaces*. Abgerufen am 18. Juli 2008 von <http://www.cs.vu.nl/~martijn/patterns/PLoP2k-Welie.pdf>

William, L., Kritina, H., & Jill, B. (2003). *Universal Principles of Design*. Rockport Publishers.

Yahoo Design Pattern Library. (2005-2008). Abgerufen am 19. Juli 2008 von <http://developer.yahoo.com/ypatterns/>

8.3 Anhänge

Nr	Anhang	Dokument
1-n	Protokolle	Anhang_01_nn.doc
2	Planung und Zeitrapport	Anhang_02_Termine&Tasks.xls
3	Für Prototyp verwendete Pattern	Anhang_03_Patterns_2008_12_12_1530.doc
4	Patterlib Analyse Anhang	Anhang_04_PatternAnalysis_2009_01_14_0200.doc
5-1	Early Prototype Konzept	Anhang_05_01_Early_Concepts.doc
5-2	Early Prototype	Anhang_05_02_Early_Prototypes.ppt
6-1	Papier Prototyp 1	Anhang_06_01_Prototyp_V1_0.ppt
6-2	Papier Prototyp Test Ergebnisse	Anhang_06_02_UsabilityTest_Ergebnisse.doc
7	Zweiter Prototyp	Anhang_07_Prototyp_V2_0.ppt
8	Final Prototyp	Anhang_08_Final_Prototyp.ppt
9	Paper Analyse Anhang	Anhang_09_PaperAnalyse_2009_01_14_1800.doc
10	Wartung und Erweiterungen	Anhang_10_Wartung_Erweiterung.doc

8.4 Abbildungen

Abbildung 1 – Das T-Modell	3
Abbildung 2 - The color dialog box in Windows 2000	8
Abbildung 3 – Konzeptionelles Vorgehen	14
Abbildung 4 - Topic Map	19
Abbildung 5 – Tags	19
Abbildung 6 – Kurzübersicht	19
Abbildung 7 – Ranking nach Patternbewertung	20
Abbildung 8 - Hierarchische Darstellung (Graham, 2002)	20
Abbildung 9 - Globale Navigation (Folmer)	20
Abbildung 10 - Guide (Timmesfeld, Schmitt, Vucica, & Schreiner, 2005)	20
Abbildung 11 – Pattern nach Aufgaben sortieren	21
Abbildung 12 – Einstieg verwirrend (Graham, 2002)	21
Abbildung 13 – Schlechtes Labeling	21
Abbildung 14 – Schlechte Suchunterstützung (Bolchini & Community)	21
Abbildung 15 – Technische Plattformen (Stimmt.AG)	22
Abbildung 16 – Falsche Hoffnungen	22
Abbildung 17 – One Way Navigation - Leere View, der Weg zurück funktioniert nicht. (Behrens, 2008)	22
Abbildung 18 – Fehlermeldungen (Timmesfeld, Schmitt, Vucica, & Schreiner, 2005)	22
Abbildung 19 – Pattern Inhalte nicht vertrauenswürdig (Graham, 2002)	23
Abbildung 20 - Persona Peter S.Coder	24
Abbildung 21 - Persona Lukas von Fach	24
Abbildung 22 - Szenario 1	25
Abbildung 23 - Szenario 2	26
Abbildung 24 – Iterativer Entwicklungsprozess	29
Abbildung 25 – OOSE	29
Abbildung 26 - RUP	29
Abbildung 27 - Hermes	29
Abbildung 28 - Usability Engineering Lifecycle	30
Abbildung 29 - Lösungsfindungsprozess	30
Abbildung 30 – Gruppierung Top Down	31
Abbildung 31 - Pattern Card – Bottom Up	32
Abbildung 32 - Pattern Tags – Bottom Up	32
Abbildung 33 – Tag Kategorisierung	33
Abbildung 34 – Kategorien - Subkategorien	34
Abbildung 35 - Phase – Beispiel Kategorien aus Bottom Up	35
Abbildung 36 – Kategorien-Tags-Pattern Datenmodell	35
Abbildung 37 – Pattern Beziehung und Navigation	36
Abbildung 38 – Phasen Auswahl	38
Abbildung 39 – Kategorien Auswahl	38
Abbildung 40 – Subkategorien Auswahl	38
Abbildung 41 – Tag Auswahl	39
Abbildung 42 – Pattern Präsentation	39
Abbildung 43 – Task Flow: Kein Problem, hat nur Interesse (Shadow Path über Beschreibung)	40
Abbildung 44 – Task Flow: Sucht bestimmtes Pattern	41
Abbildung 45 – Task Flow: Sucht zu Anforderung lediglich Information	42
Abbildung 46 – Task Flow: Prototyp	43
Abbildung 47 - Wireframe Welcome Page	44
Abbildung 48 - Wireframe Lösungsfindung	45
Abbildung 49 - Wireframe Lösung	46
Abbildung 50 - Welcome Screen	47
Abbildung 51 - Lösungsfindungs Screen	48
Abbildung 52 – Lösungs Screen	49
Abbildung 53 - Primär und Sekundärfarben	50
Abbildung 54 – Selektionsfarbbeispiel	50
Abbildung 55 - Selektionsfläche	50
Abbildung 56 - Process Phase Navigation UI Element	50
Abbildung 57 - Process Phase Navigation UI Element (in Lösungsfindungs Screen)	51
Abbildung 58 – Category Selection	51

Abbildung 59 - Tag List UI Element.....	51
Abbildung 60 - Pattern List UI Element.....	52
Abbildung 61 - Shadowpath Navigation UI Element.....	52
Abbildung 62 – Reference Navigation UI Element	53
Abbildung 63 - Pattern Detail UI Element.....	53
Abbildung 64 – Pattern Options UI Element.....	53
Abbildung 65 – Pattern Tags UI Element.....	53
Abbildung 66 – Phasen Auswahl Screen	54
Abbildung 67 – Kategorien Auswahl Screen	55
Abbildung 68 – Subkategorien Auswahl Screen.....	56
Abbildung 69 -Abbildung 36 – Tag Auswahl Screen	57
Abbildung 70 -Abbildung 36 – Pattern Auswahl Screen.....	57
Abbildung 71 -Abbildung 36 – Pattern Detail	58
Abbildung 72 - Early Prototypen 1 bis 3.....	63
Abbildung 73 - Early Prototyp 4.....	64
Abbildung 74 – Der zweite Prototyp im Test	64
Abbildung 75 – Der dritte Prototyp	65
Abbildung 76 - Thomas Kneubühl	70
Abbildung 77 - Markus Oberle.....	70
Abbildung 78 - Manuel Schaffner	70
Abbildung 79 - Roland Schild.....	70

8.5 Glossar

	Begriff	Erläuterung
A	Alexanderform	Darstellungsform von Pattern nach Ch. Alexander in Textform, mit Artefakt
B	Benutzer- und Taskanalyse	Usability-Methode zur Untersuchung der Benutzer und deren Tätigkeiten
	Benutzeranalyse	Analyse der Eigenschaften von Benutzern
	Bidirektionale Verknüpfung	Verknüpfung in beide Richtungen: Wovon abhängig und wie weiter
	Brainstorming	Methode zur Ideenfindung
C	Card Sorting	Verfahren, mit dem logische und benutzerfreundliche Strukturen entwickelt werden können. Begriffe werden auf (bewegliche) Karten geschrieben und gruppiert.
	CMS	Content Management System
	Community	Gemeinschaft
	Conceptual design	Konzeptionelles Design
	Conceptual model	Konzeptionelles Modell – dahinterliegendes Konzept, welches alles zusammenhält
	Contextual Navigation	Theamatisch weiterführende Navigation
D	Designer	Derjenige der formt und gestaltet
	Entwicklungsprozess	Vorgehen bei der Entwicklung eines Systems (RUP, HERMES,...)
	Expert Review	Untersuchung einer Applikation auf Usability durch einen Experten
F	Flash	Software von Adobe
G	Global Navigation	Haupt Navigation
	GUI	Siehe UI
H	HCI	Human Computer Interaction – Interaktion des Benutzers mit dem System
	HCID	Human Computer Interaction Design – Lehre von der Interaktion zwischen Mensch und Maschine
	HT	Entwicklungsprozess mit Phasen (horizontales T)
	HTML	Hyper Text Markup Language – technische Form einer Webseite
I	IA	Konzeption und Definition der Struktur eines Informationssystems
	Information Architecture	Siehe IA
	Informationsarchitektur	Siehe IA
	Internetplattform	Plattform (Sammlung von Webseiten) auf dem Internet
	Interview	Befragung durch Fragesteller mit dem Ziel, persönliche Informationen oder Sachverhalte zu ermitteln
	Iteratives Prototyping	Erstellen Prototyp – Test – Anpassung – Test – Anpassung – etc.
K	Kategorien	Die Tags sind zu Kategorien zusammengefasst und diese allenfalls wiederum zu Oberkategorien
	Kontext	Umfeld eines Systems
	Kontext Analyse	Analyse der Arbeit und deren Umstände
	Konzept Visualisierung	Grafische Darstellung der Konzeptergebnisse
	Kritikalität	Kritische Masse
L	Labeling	Benennungen
	Lösungsentität	Etwas das die Lösung enthält zB. Pattern, Methode, Guideline, Style Guide, Vorlage, weiterführende Informationen
M	Mallorca	Ferieninsel auf den Balearen (Spanien). Ort, an dem der Durchbruch dieser Arbeit erzielt wurde.

	MAS	Master of Advanced Studies
	Mindmapping	Gedächtniskarte, ist eine grafische Darstellung, die Beziehungen zwischen verschiedenen Begriffen aufzeigt
	Model driven architecture	Ansatz bei welchem ein Problem als Model abgebildet wird und daraus die Lösung generiert wird
	Moodboard	Grafisches Bild, welches eine Stimmung vermittelt
O	OOSE	Object Oriented Software Engineering – Entwicklungsprozess von Software
P	Pagination	Ein Pattern, das die Navigation zwischen Seiten beschreibt
	Painpoints	Einzelne Punkte unter denen man leidet
	Paper	Wissenschaftliche Publikation in Kurzform (1-10 Seiten)
	Pattern	Wiederholt bewährter Lösungsmuster für ein Problem. Beschreibt auf verständliche und exemplarische Art (Beispiel, Gefühl/Situation) Problem und Lösung, teils auch Kontext.
	Pattern Language	Vokabular mit welchem die Patterns beschrieben sind
	Pattern Library	Sammlung von Patterns
	Pattern Tool	Software zur Verwaltung von Patterns
	Patternsprache	Siehe Patternlanguage
	Persona	Repräsentative Mitarbeiter für die Benutzung einer Software
	PLML	Pattern Language Markup Language
	Qualitative Analyse	Inhaltliche Untersuchung
	Quantitative Analyse	Mengenmässige Untersuchung (Statistische Auswertung)
R	Related Patterns	Verwandte Patterns
S	Semantic Web	Maschinell lesbare Semantik einer Webpage
	Shadowpath	Zweiter Navigationsweg (z.B. über Beschreibungen von Objekten)
	Sitemaps	Inhaltsverzeichnis einer Web Applikation
	Social Classification	Klassifizierung durch die Benutzer
	Social Taging	Markieren von Objekten über Begriffe durch die Benutzer
	Storyboard	Drehbuch
	Styleguide	Beschreibt wie (aber nicht warum/wann) ein User Interface grafisch gestaltet werden muss
	Szenario	Beschreibung einzelner Arbeitsschritte in prosa
T	Tag	Begriff/Markierung der einer oder mehreren Lösungsentitäten zugeordnet werden. Ein solcher Begriff/Tag verweist somit auf eine oder mehrere Lösungsentitäten.
	Tag Kategorien	Gruppierung von Tags zu Kategorien
	Tag Subkategorien	Gruppierung von Tags zu Subategorien
	Task Analyse	Analyse der Aufgaben mit einem System
	Taskflow	Abfolge einzelner Arbeitsschritte. Kann ein Szenario beinhalten.
	Testing	Evaluation von Ergebnissen
U	UCD	User Centered Design – Projektvorgehensweise bei der der Benutzer im Zentrum steht
	UI	User Interface – Benutzeroberfläche als Schnittstelle zum System
	UI Design	User Interaction Design – Gestaltung der Interaktion zwischen Benutzer und Software
	UML	Unified Modeling Language
	Usability Test	Test einer Applikation auf deren Usability
V	Variationen und Entscheidung	Kreativitäts- und Auswahlverfahren im grafischen Design
	Visio	Microsoft Visualisierungs-Software
	VT	Lösungsfindungsprozess mit stufenweiser Eingrenzung der Lösungsmenge (vertikales T)

W	Walkthrough	Verfahren bei Usability-Tests mit Benutzerinteraktion
	Web 2.0	Interaktive und kollaborative Elemente des Internets
	Widget	Kleines Computerprogramm, das nicht als eigenständige Anwendung betrieben, sondern in ein UI eingebunden wird.
	Wiki	Schnell änderbare Seitensammlung
	Wireframes	Zeichenform für das fundamentale Layout eines Interfaces
X	XML	Extensible Markup Language