



Goethe-Universität Frankfurt

Bachelorarbeit

# TeamVision - visualisiertes Team- und Projektmanagement

eingereicht bei

Prof. Dr.-Ing. D. Krömker

Professur für Graphische Datenverarbeitung

*von*

David Veith

Projekt: **virtuelle Teams erfolgreich führen**

Projektstatus: **abgeschlossen**

Zeitraum: 27.07.2009 bis 28.09.2009

[Projekteigenschaften editieren](#) [Spezifikation anzeigen](#)

neuen Meilenstein

neue Komponente

Eingereicht am 28.09.2009

## **Eidesstattliche Erklärung**

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Frankfurt, den 28.09.2009

---

(David Veith)

## **Danksagung**

Zunächst einmal möchte ich Professor Detlef Krömker und insbesondere Dipl.-Inf. Sarah Voß für ihre Unterstützung und die Möglichkeit danken, meine Ideen zu realisieren und diese Arbeit zu verfassen. Außerdem geht ein besonderer Dank an Dipl.-Inf. Jörg Demmer sowie Dipl.-Inf. Daniel Schiffner und Dipl.-Inf. Sebastian Schäfer für ihre Ratschläge und Hilfsbereitschaft.

Weiterhin möchte ich meiner gesamten Familie danken, die mich immer unterstützt hat in meiner Wahl.

Ein besonderer Dank geht an meine Freundin Nadine Nienberg für ihre Unterstützung.



# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>xi</b>
Deutsch . . . . .	xi
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Aufgaben und Ziele dieser Arbeit . . . . .	2
1.3 Aufbau der Arbeit . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Grundlagen des Projektmanagements . . . . .	3
2.1.1 Was ist ein Projekt? . . . . .	3
2.1.2 Warum Management? . . . . .	3
2.1.3 Virtuelles Teammanagement . . . . .	4
2.1.4 Grundsätzliche Anforderungen an ein Tool für virtuelle Teams . . . . .	5
2.2 Dokumentation . . . . .	5
2.3 Wiki . . . . .	6
2.3.1 MediaWiki . . . . .	7
2.3.2 TWiki . . . . .	7
2.3.3 TikiWiki . . . . .	7
2.4 Visualisierung . . . . .	7
2.4.1 Bewertungskriterien für Visualisierung . . . . .	7
2.4.2 Farbe . . . . .	10
2.4.3 Diagramme . . . . .	11
<b>3 State-of-the-Art Analyse</b>	<b>13</b>
3.1 Auswahl von Tools zur Analyse . . . . .	13
3.2 Durchführung der Analyse . . . . .	14
3.2.1 Action Method . . . . .	14
3.2.2 Devunity . . . . .	15
3.2.3 Planzone . . . . .	17
3.2.4 TeamBox . . . . .	18

3.2.5	MediaWiki als Dokumentationswerkzeug . . . . .	19
3.2.6	TWiki als Dokumentationswerkzeug . . . . .	20
3.3	Visualisierungen von Projekten in baumartigen Strukturen . . . . .	20
3.3.1	project illustrator . . . . .	20
3.3.2	proscio . . . . .	21
3.3.3	Zusammenfassung . . . . .	21
3.4	Fazit der Analyse . . . . .	22
<b>4</b>	<b>Anforderungsanalyse</b>	<b>23</b>
4.1	Begründung der Notwendigkeit eines weiteren Tools . . . . .	23
4.2	Zielgruppe . . . . .	24
4.3	Motivation der Benutzer . . . . .	24
4.4	Beispieltool für virtuelles Projektmanagement . . . . .	24
4.5	Anforderungen virtuelles Team- und Projektmanagement . . . . .	25
4.6	Rollenspezifische Anforderungen . . . . .	25
4.7	Controlling: Bewertungskriterien für ein IT-Projekt . . . . .	27
4.8	Anforderungen an das Tool . . . . .	28
<b>5</b>	<b>Konzeption</b>	<b>29</b>
5.1	Ablaufplan und Aufbau von TeamVision . . . . .	29
5.2	Aufbau von TeamVision im Detail . . . . .	30
5.2.1	Projektverwaltung . . . . .	30
5.2.2	Team-, Benutzer-, Gruppen- und Rollenverwaltung . . . . .	31
5.2.3	Controlling . . . . .	32
5.2.4	Diskussion . . . . .	33
5.2.5	Entwicklungsdokumentation . . . . .	33
5.2.6	Verifikation . . . . .	34
5.2.7	Erweiterungsverwaltung . . . . .	35
5.2.8	Komponenten Konfiguration und Datenbankverwaltung . . . . .	35
5.2.9	Visualisierung . . . . .	36
5.2.10	Komponente: „Zuteilung Aufgaben Fremdumgebung“ . . . . .	37
5.3	Oberfläche von TeamVision . . . . .	37
5.3.1	Der Projektbaum . . . . .	38
5.3.2	Hilfen für Menschen mit Farbwahrnehmungsstörungen . . . . .	41
5.4	Aufbau der Datenbank . . . . .	41
5.5	Weitere mögliche Funktionen . . . . .	42
5.6	Fallbeispiel für eine Fremdkomponente: das MediaWiki . . . . .	42
5.7	Zusammenfassung des Konzepts . . . . .	44

---

<b>6 Implementierung</b>	<b>45</b>
6.1 Umfang der Implementierung . . . . .	45
6.2 Wahl der Entwicklungsumgebung und Sprachen . . . . .	45
6.3 Eingrenzung des Rollenmodells . . . . .	46
6.4 Struktur der Implementierung . . . . .	46
6.5 Anmeldevorgang und Login . . . . .	47
6.6 Oberfläche . . . . .	47
6.7 Grundfunktionen . . . . .	48
6.7.1 Profil . . . . .	48
6.7.2 Adressbuch . . . . .	49
6.7.3 ToDo-Listen . . . . .	50
6.7.4 Kalender . . . . .	51
6.8 Projektverwaltung . . . . .	52
6.8.1 Projektübersicht: Der Projektbaum . . . . .	52
6.8.2 Teams, Rollen und Zuständigkeiten im Projekt . . . . .	54
6.8.3 Details aus dem Projekt über die Zoomfunktion, Listen und Popups . . . . .	54
6.9 Kooperation mit einer Fremdunggebung am Beispiel MediaWiki . . . . .	54
<b>7 Zusammenfassung, Fazit und Ausblick</b>	<b>57</b>
7.1 Zusammenfassung und Fazit der Arbeit . . . . .	57
7.2 Ausblick . . . . .	59
<b>A Glossar</b>	<b>61</b>
<b>B Bilder zum Entwurf der Oberfläche</b>	<b>63</b>
<b>Literaturverzeichnis</b>	<b>65</b>



# Abbildungsverzeichnis

2.1	Herstellungsland eines Autotyps in zwei Darstellungen[Mac86] . . . . .	8
2.2	Durch Umstrukturierung präattentive Wahrnehmung verbessern . . . . .	9
2.3	Relevanz der visuellen Variablen nach Abbildungsarten aus[Mac86] . . . . .	9
2.4	Darstellung des RGB Modells im Einheitswürfel[Lic] . . . . .	10
2.5	Ein Liniendiagramm . . . . .	11
2.6	Ein Kreisdiagramm mit 6 disjunkten Teilmengen . . . . .	12
2.7	Ein Ganttendiagramm[Bus] . . . . .	12
3.1	Gesamtübersicht über ein Projekt in Action Method.[Beh] . . . . .	15
3.2	Devunity Visualisierung[Alo] . . . . .	16
3.3	Screenshots von Planzone[Aug] . . . . .	18
3.4	Screenshot Teambox[Tea] . . . . .	19
3.5	Ein Element des Projektbaums aus project illustrator[iata] . . . . .	20
3.6	Ein möglicher vollständiger Projektbaum[iata] . . . . .	21
3.7	Der Projektbaum als Gliederung der Aufgaben in einer Übersicht[EFK06] . . . . .	21
4.1	Wesentliche Aufgaben der einzelnen Rollen. . . . .	26
5.1	Der gedachte Ablaufplan des Tools. . . . .	29
5.2	Aufbau von TeamVision . . . . .	30
5.3	Projektstatus über Kreisdiagramm visualisiert . . . . .	32
5.4	Projektstatus für den Controller visualisiert . . . . .	33
5.5	Projektbaum in der Ansicht für Entwickler . . . . .	34
5.6	Abhängigkeiten mittels Zoomfunktionalität . . . . .	37
5.7	Entwurf Oberfläche TeamVision . . . . .	38
5.8	Farbwahl für Design . . . . .	41
5.9	TeamVision mit gedachtem Funktionsumfang . . . . .	44
6.1	Ausschnitt aus der Oberfläche von TeamVision . . . . .	48
6.2	Farbauswahl für die Statusanzeige des Projektbaums . . . . .	49
6.3	Ansicht des Adressbuchs in TeamVision . . . . .	50

6.4	Ansicht einer ToDo-Liste in TeamVision . . . . .	51
6.5	Darstellung eines Projektbaums über alle Ebenen . . . . .	56
B.1	Ausschnitt TeamVision: Navigationsbereich . . . . .	63
B.2	Entwurf der Oberfläche von TeamVision. . . . .	64

# Zusammenfassung

Trotz eines umfangreichen Angebots an Literatur und Ratgebern im Bereich des Projektmanagements scheitern auch heute noch viele IT-Projekte. Ursache sind oft Probleme im Projektteam oder Fehleinschätzungen in der Planung des Projektes und Überwachung des Projektstatus. Insbesondere durch neue Technologien und Globalisierung entstandene Arbeitsweisen wie das virtuelle Team sind davon betroffen. In dieser Arbeit wird auf die Frage eingegangen, was virtuelle Teams sind und welche Probleme die Arbeit von virtuellen Teams belastet. Dafür werden aktuell existierende Tools aus dem Bereich des Web 2.0 analysiert und aus dem Stand der angebotenen Tools vermeidbare Schwächen der Helfer herausgearbeitet. Anschließend wird ein mittels einer Anforderungsanalyse und eines Konzepts, welches neue Methoden zur Darstellung von Projektstatus und Verknüpfung mit Dokumentation und Kommunikation nutzt, das Tool „TeamVision“ erstellt, welches versucht, virtuelle Teams möglichst effizient zu managen, Probleme schnell zu erkennen und somit die Arbeit innerhalb des Teams zu beschleunigen. Hierbei wird insbesondere das Ergebnis der Analyse benutzt, dass viele Tools einzelne Verwaltungsaufgaben getrennt durchführen. Informationen müssen vom Nutzer selbst aus den verschiedenen Grafiken, Listen oder anderen Darstellungen gesammelt und selbst assoziiert werden. Die prototypische Implementierung von TeamVision versucht den Informationsfluss beherrschbar zu machen, indem Übersichten in einem Projektbaum zusammengefasst werden, der mittels Zoomfunktionen und visueller Hilfsmittel wie Farbgebung versucht, die Informationsbeschaffung zu erleichtern.



# Kapitel 1

## Einleitung

### 1.1 Motivation

Um eine möglichst hohe Flexibilität zu erreichen, setzen immer mehr Unternehmen auf kleinere Projektgruppen, um im permanenten Wandel wettbewerbsfähig bleiben zu können. (Vgl. [Lit02, S.7-8]) Neben den klassischen Strukturen im Projektmanagement kristallisiert sich eine immer mehr an Bedeutung gewinnende Struktur für den Erfolg eines Projektes heraus: Das Team.

Die Entwicklung von Software in einem Team stellt eine Herausforderung dar, die sich erst langsam im Bereich des Projektmanagements etabliert. Insbesondere so genannte „virtuelle Teams“, in welchen die Teammitglieder vernetzt über Distanz miteinander arbeiten und kommunizieren, sind dabei folgenden Faktoren ausgesetzt, welche regelmäßig zu strukturbedingten Problemen führen:

- erhöhtes Ausmaß an Abstimmungsproblemen und Missverständnissen
- Interaktionen zwischen Teammitgliedern sind häufiger nicht richtig abgestimmt
- plötzliche Ausreißer (Abhängigkeiten und Aussteiger)
- Verlust von Energie und Engagement über die Zeit
- unterschiedlich verstandene Arbeitsziele und Ergebnisverständnis aller beteiligten Personen

Adaptiert nach [Ger02]

Maßnahmen, um diese Probleme abzufangen werden oft noch in unzureichendem Umfang getroffen. Eine wichtige Rolle für Erfolg in der Softwareentwicklung spielt die Wartbarkeit der Software, daraus resultierend auch der Programmierstil des Entwicklerteams, sowie die Dokumentationsarbeit. Genau an diesem Punkt läuft es in vielen Projektgruppen nicht wie gewünscht. Eigene Erfahrungen haben gezeigt, dass im Programmieralltag oft wichtige Dokumente nicht auffindbar oder im schlimmsten Fall gar nicht vorhanden waren. Deshalb ist es für den Erfolg eines Projektes und daraus resultierenden Folgeaufträgen von entscheidender Bedeutung, dass sowohl das Team als auch der Teamleiter über den Projektstatus und somit den Status des Teams selbst informiert sind. Beim Betrachten von

Dokumenten oder Quellcode aus dem Projekt soll die Frage: „Warum steht denn das hier?“ [Wei04, S.40] gar nicht erst auftauchen.

## 1.2 Aufgaben und Ziele dieser Arbeit

Ein Ziel dieser Arbeit ist es, eine Software zur konzipieren, welche die strukturbedingten Probleme eines virtuellen Teams abfängt, rechtzeitig aufzeigt oder ganz verhindert. Eine Analyse einiger aktuell auf dem Markt existierender Tools soll zeigen, wo Bedarf besteht und noch Lücken in der Umsetzung vorhanden sind. Übereinstimmend mit der Meinung von Weinberg <sup>1</sup> bzgl. der Dokumentation von Software, welche besagt, dass der Quellcode für sich selbst sprechen können muss, wird im Rahmen dieser Arbeit ein Konzept entwickelt werden, welches dazu beitragen soll Dokumentation von Entwicklungsarbeit, deren Ideen und Resultate <sup>2</sup> sowie den Status des Projektes dem Benutzer bzw. Teammitglied in geeigneter Form zu präsentieren. Der Benutzer kann anhand der gefundenen Informationen seine Komponente bzw. deren Schnittstelle zu anderen Komponenten in geeigneter Weise entwickeln und validieren sowie ggf. eingreifen, falls eine Anforderung durch die eigene, aber auch durch Komponenten anderer Teammitglieder nicht erfüllt wird.

Darüber hinaus soll in der Implementierung im Rahmen dieser Arbeit erprobt werden, mit welchen Werkzeugen sich eine solche Dokumentationsumgebung kombinieren lassen kann.

## 1.3 Aufbau der Arbeit

Im zweiten Kapitel wird grundlegendes Wissen vorgestellt, welches zum Verständnis der Arbeit notwendig ist. Dabei wird insbesondere analysiert werden, was ein Projekt ausmacht und welche Besonderheiten bei der Betrachtung eines Teams bestehen. Außerdem wird auf verschiedene Aspekte der Visualisierung eingegangen, die später verwendet werden.

Im dritten Kapitel wird anhand einer State-of-the-Art Analyse untersucht, was für Arten von Tools, insbesondere im Web 2.0, vorhanden sind, was sie leisten können und wo noch Potential vorhanden ist. Des Weiteren wird eine Untersuchung dahingehend durchgeführt, was es zum Thema „Projektbaum“ an Implementierungen gibt, wofür drei Beispielanwendungen analysiert werden.

Im vierten Kapitel folgt eine Anforderungsanalyse an ein neues Tool, welche die wesentlichen Aufgaben eines solchen Tools festlegt und für das Konzept aufbereitet.

Das fünfte Kapitel beschäftigt sich mit einer theoretischen Umsetzung und der gedachten Funktionsweise eines Tools intern als auch extern durch bereitgestellte Schnittstellen, z.B. zur Anbindung an ein bereits bestehendes Tool.

Das sechste Kapitel beschäftigt sich mit der Implementierung und dabei aufgetretenen Problemen, bevor sich das siebente Kapitel abschließend einer zusammenfassenden Betrachtung dieser Arbeit und der aus ihr gewonnenen Ergebnisse widmet.

---

<sup>1</sup>Vergleich [Wei04, S.455f.]

<sup>2</sup>Definitionen, Konzepte und Schnittstellen sind festzuhalten und nicht detailgetreu deren Ausprägung im Quellcode

# Kapitel 2

## Grundlagen

### 2.1 Grundlagen des Projektmanagements

#### 2.1.1 Was ist ein Projekt?

Nach DIN 69901<sup>1</sup> ist ein Projekt definiert als „ein Vorhaben, das im Wesentlichen durch eine Einmaligkeit der Bedingungen in Ihrer Gesamtheit gekennzeichnet ist“.

Litke<sup>2</sup> zählt weitere wesentliche Merkmale auf, die zur Erkennung eines Projektes dienen:

- Zielorientierung
- Terminvorgabe
- Zeitliche Begrenzung
- Ressourcenbeschränkung
- Abgrenzung zu anderen Vorhaben

#### 2.1.2 Warum Management?

Aus den genannten Merkmalen, die Indikatoren für ein Projekt sind, sowie der Definition nach DIN 69901 erschließt sich, dass ein Projekt ein einmaliges Vorhaben ist, welches in Ressourcen wie Zeit, Budget, Personal etc. beschränkt ist. Daraus ergibt sich die Notwendigkeit, die Ressourcen so zu verwalten, dass das Projekt innerhalb seiner Vorgaben umgesetzt werden kann. Abweichender Ressourcenverbrauch, der über den zur Verfügung stehenden Ressourcen liegt, kann zum Scheitern des Projektes führen. Bestes Beispiel hierfür ist die unfertige Software „Duke Nukem Forever“ von 3D Realms. 1997 wurde mit der Entwicklungsarbeit begonnen und im Mai 2009 wurde das Projekt voraussichtlich für immer eingestellt. Die Kosten für dieses Scheitern lagen bei etwa 20 Millionen US-Dollar. [Gol]

---

<sup>1</sup>Aus [GPM09]

<sup>2</sup>[Lit02, S.15]

Obwohl dieses Beispiel extrem scheint, so scheitern viele Projekte an der Erfüllung der Zielsetzung, am Einhalten des Budgets oder der Zeit. Auch bringt ein Projekt immer das Risiko mit sich, dass beim Erreichen der Ziele die Konkurrenz schon weiter ist oder im schlimmsten Fall den Markt beherrscht. Folgendes Zitat eines Beraters für Projekte gibt Aufschluss darüber, woran insbesondere Softwareprojekte scheitern:

„Wir haben in den vergangenen Jahren über 50 Audits von Großprojekten vorgenommen. Ein wichtiges Ergebnis: Hinter Problemen und Krisen stand nie nur eine singuläre Ursache[...]. Klare Ziele fehlen, der Nutzen ist selten beschrieben. Es reicht nicht, die Einführung der Software x bis zum Tag y als ein Ziel zu definieren. Die strategische Zielsetzung und der Nutzen, das „Wozu“ für das Unternehmen und die Kunden müssen präzise und messbar festgelegt werden. Bei der Planung werden Termine oft viel zu knapp angesetzt, das Budget ist zu klein, der Elefant hinter der Mücke wird nicht gesehen.“ [HK05].

Folglich ist gutes Management in einem Projekt, die damit verbundene umsichtige Ressourcenverwaltung sowie das frühzeitige erkennen von Engpässen und Problemstellen in der Projektplanung essentiell für den Erfolg des Projektes.

### 2.1.3 Virtuelles Teammanagement

Als theoretischer Hintergrund eignet sich für die Betrachtung eines virtuellen Teams das Gruppenmodell von Tuckmann(1965). Adaptiert nach Staehle<sup>3</sup> lässt sich das Modell für virtuelle Teams in vier Phasen zusammenfassen:

1. Forming: Unsicherheit des Individuums in der Gruppe, Orientierungsphase. Mitglieder der Gruppe definieren die Aufgaben, Regeln und Ziele sowie geeignete Methoden, um diese zu erreichen.
2. Storming: Konflikte zwischen Teammitgliedern treten auf. Ablehnung der Aufgabenstellung sowie Polarisierung von Meinungen sind möglich.
3. Norming: Entwicklung von Gruppenkohäsion sowie Aufbau einer funktionalen Vertrautheit. Es findet ein Austausch von Meinungen statt.
4. Performing: Interpersonelle Probleme sind angesprochen und ausdiskutiert, Gruppenstruktur ist funktional zur Aufgabenerfüllung und das Rollenverhalten ist flexibel. Problemlösungen tauchen auf, Energie des Teams wird primär in eine effiziente Lösung von Aufgabenstellungen investiert - **Hauptarbeitsphase**.

Ein virtuelles Team muss alle vier Phasen durchlaufen, um eine hohe Effektivität zu erreichen.[Ger02] Insbesondere die Phase 2 wird oft übersprungen, was das „Wir Gefühl“ im Team und somit die Teamperformance insgesamt stören kann. Bei virtuellen Teams ist die Teamperformance ein häufiges Problem. Rahmenbedingungen, die für ein konstruktives Arbeiten im Team essentiell sind: Arbeitsregeln,

---

<sup>3</sup>Vgl. [Sta91]

eine klare Positions- und Rollenverteilung sowie eine gemeinsam definierte Roadmap zum Vorgehen. Außerdem muss eine funktionale Vertrautheit zwischen den Mitgliedern des Teams bestehen. Das heißt, dass ein Teammitglied sich auf ein anderes Teammitglied und seine Problemlösungen verlässt und ggf. auftretende Probleme direkt anspricht.

Elemente des Projektes mit denen alle Teammitglieder vertraut sein müssen, um eine gut organisierte und effiziente, virtuelle Teamarbeit zu gestalten, sind Ziel, Sinn und Zweck des Projektes. Außerdem sollte das Vorgehen gemeinsam in einer Roadmap und Regeln für die Zusammenarbeit festgelegt und auch durchführbar sein, damit alle Teammitglieder ihren „Platz“ haben. Weiterhin müssen sämtliche Teammitglieder über das notwendige technische Equipment verfügen sowie Kommunikationskanäle eingerichtet und ihr Einsatz deutlich geregelt sein. (Vgl. [Ger02])

#### **2.1.4 Grundsätzliche Anforderungen an ein Tool für virtuelle Teams**

Begründet in der Distanz der Teammitglieder zueinander ergeben sich für ein virtuelles Team besondere Anforderungen an ein Tool, welches die Dokumentationsarbeit für den Projekt- bzw. den Entwicklungsstand übernehmen soll. Drei wesentliche Rahmenbedingungen der Verfügbarkeit von Informationen, welche die Effizienz des Teams steigern: Zentrale und einheitliche Verwaltung der Information, wobei kein information hiding stattfinden darf, es sei denn, dass Abweichungen in den Regeln definiert sind. Die Verlässlichkeit der Informationen muss durch die Gewährleistung der Aktualität der Informationen sichergestellt sein. Die Nutzung des Tools muss verbindlich geregelt sein. Alle relevanten Informationen müssen festgehalten werden. Für die Dokumentation dürfen nur die festgelegten Tools verwendet werden. (Vgl. [Ger02])

## **2.2 Dokumentation**

Ziel der Softwareentwicklung ist die Erstellung eines Programms, welches lauffähig im System des Kunden ist und dabei die vom Kunden gestellte Spezifikation erfüllt. Probleme und Lösungen sowie Abweichungen von der Spezifikation in der Entwicklung müssen dokumentiert sein, um ggf. bei Fehlern Maßnahmen ergreifen zu können, wenn diese auf dem System des Kunden auftreten. Für Mängel ist eine Dokumentation eines Projektes von größerer Wichtigkeit. Im Gegensatz zu Fehlern<sup>4</sup>, die eine Abweichung zwischen Ist- und Sollverhalten darstellen, liegt ein Mangel vor, „[...]wenn eine gestellte Anforderung oder eine berechnete Erwartung nicht angemessen erfüllt wird.“[Dro06, S.73] Nach BGB, §459, Abschnitt 1 gilt:

„Der Verkäufer einer Sache haftet dem Käufer dafür, dass sie zu der Zeit, zu welcher die Gefahr auf den Käufer übergeht, nicht mit Fehlern behaftet ist, die den Wert oder die Tauglichkeit zu dem gewöhnlichen oder dem nach dem Verträge vorausgesetzten Gebrauch aufheben oder mindern. Eine unerhebliche Minderung des Wertes oder der Tauglichkeit kommt nicht in Betracht.“

---

<sup>4</sup>umgangssprachlich „Bugs“

Aus dem Gesetzestext lässt sich folgern, dass Software dann von der Haftung ausgeschlossen ist, wenn eine Software keine Fehler enthält, die den nach Vertrag vorausgesetzten Gebrauch beeinträchtigen. Aufgrund der nicht formalen Formulierung bleibt der Gesetzgeber jedoch dem Entwickler die Erklärung schuldig, welcher Fehler im Programmcode zu Regressansprüche bis hin zur völligen Auflösung des Vertrags führen kann, eventuell begleitet von Schadensersatzansprüchen durch den Kunden.

Zu Beachten ist weiterhin, dass Mängelansprüche nach §438 Abschnitt 1 Nr. 3 des BGB zwei Jahre lang durch den Kunden geltend gemacht werden können, im schlimmsten Fall schon lange nachdem die Entwickler das Projekt abgeschlossen haben und dem Vertreiber der Software nicht mehr zur Verfügung stehen. Für vorsätzliche verschwiegene Mängel gilt die zweijährige Verjährungsfrist nicht<sup>5</sup>. Der Kunde hat bei Mängeln das Recht auf Nacherfüllung, kann den Kaufpreis mindern oder vom Vertrag zurücktreten oder Schadensersatz verlangen<sup>6</sup>.

Die ganze Sachlage aus Sicht des Gesetzgebers ist im BGB §433-§480 geregelt, allerdings von weiteren Paragraphen aus dem BGB und anderen Gesetzbüchern begleitet und ergänzt durch Urteile vom diversen Gerichten. Dokumentation ist damit mehr als nur ein Hilfsmittel für ein Projektteam, um die Ziele zu erfüllen und interne Probleme zu erkennen. Dokumentation bedeutet ein Nachweis der Erfüllung der Spezifikation mit dem Kunden und deren Änderungen im Verlauf des Projektes durch beide Vertragspartner, um im Falle einer Beanstandung durch den Kunden entsprechend reagieren und Schaden vom Vertreiber abwenden zu können, wenn keine berechtigten Forderungen vorliegen.

## 2.3 Wiki

Ein Wiki, hawaiisch für „schnell“, auch WikiWiki oder WikiWeb genannt, ist bis auf wenige Ausnahmen wie TiddlyWiki<sup>7</sup> eine Client-Server Hypertext-Anwendung, die sich dadurch auszeichnet, dass ein User Informationen sowohl lesen als auch dynamisch ändern kann. Die Informationen werden dabei meist in einer Datenbank gespeichert. Einige Wikis wie das TWiki setzen auf direktes Umwandeln von Informationen in HTML-Dokumente und ergänzen die Verlinkung in den übergeordneten Verzeichnissen. Es ist möglich, durch Erweiterungen eine Datenbank in TWiki einzubinden. Wikis können sowohl im Inter- als auch im Intranet eingesetzt werden, wobei der Einsatz im Intranet meist für Firmen von Interesse ist, um ihre Dokumentation zu verwalten.

Die Idee zum Wiki stammt aus der Anfangszeit des World Wide Web (kurz: WWW), ursprünglich von Berners-Lee, und wurde 1995 das erste Mal von Theoretikern für Experimente im Netz umgesetzt. Seitdem haben sich Wikis im Bereich des Wissensmanagement etabliert. (Vgl. Fazit: [ben06]) Bekannte Anwendungen, die auf dem Prinzip des Wikis basieren sind MediaWiki, TWiki und TikiWiki. (Vgl. [c2w])

---

<sup>5</sup>Vgl. §438 Abschnitt 3

<sup>6</sup>§437 BGB, siehe auch <http://dejure.org/gesetze/BGB/437.html>

<sup>7</sup>nur clientseitig, realisiert mit HTML und JavaScript [www.tiddlywiki.com](http://www.tiddlywiki.com)

### 2.3.1 MediaWiki

Das MediaWiki ist ein frei verfügbares Wiki, basierend auf PHP. Wikipedia<sup>8</sup> basiert auf dem MediaWiki-Framework, welches seit 2002 als Grundlage für Wikipedia entwickelt wurde.[We] Vertrieben wird das MediaWiki von der Wikimedia Foundation und dient auch als Grundlage vieler anderer Wikis wie DBWiki oder aber auch für alltägliche Belange wie das FahrradWiki oder „Haus-&Wohnung“<sup>9</sup>. [Wick] Insbesondere am Wiki „Haus-&Wohnung“ wird deutlich, worin das Problem vieler Wikis liegt. Die Qualität des Wikis ist vom Inhalt und der Anzahl der Autoren abhängig.

### 2.3.2 TWiki

TWiki ist eine auf PEARL basierende Anwendung, die als Intranetumgebung für mittelständige und große Unternehmen gedacht ist, welche mit TWiki ihre Dokumente verwalten wollen. Unternehmen wie Yahoo! nutzen TWiki für Projektmanagement.(Vgl. [Thob]) Die Arbeitsweise von TWiki lässt sich mit einem Zitat der Herstellerseite beschreiben: „TWiki is [...] written in Perl. It reads a text file, hyperlinks it and converts it to HTML on the fly.“ [Thoa]

Einsatzgebiete des TWiki reichen von Qualitäts- und Projektmanagement über Stadt- und Landschaftsplanung bis hin zur Dokumentation von Forschungsergebnissen.

### 2.3.3 TikiWiki

TikiWiki vereint mehrere Features von Web 2.0 in einer Software. Sich selbst beschreibt TikiWiki als Groupware oder Content Management System(kurz: CMS). In TikiWiki enthalten sind Wikis, Foren, Blogs, Bildgalerien, Linkübersichten und weitere Inhalte, um WWW-Inhalte zu gestalten.(siehe [tik])

## 2.4 Visualisierung

### 2.4.1 Bewertungskriterien für Visualisierung

Die Visualisierung beschäftigt sich mit der Darstellung von Daten, um durch geeignete Repräsentation die Aufnahme von Informationen zu erleichtern. Es existieren zwei grundsätzliche Richtungen in der Visualisierung, zwischen denen zu unterscheiden ist: Wissenschaftlich-technische Visualisierung und Informationsvisualisierung. In dieser Arbeit wird nur auf die Informationsvisualisierung eingegangen, da sich diese mit Bereichen wie Oberflächendarstellungen im Webbrowser, um Informationen besonders gut aufnehmen zu können, beschäftigt.

Um eine Visualisierung zu bewerten, sind zwei<sup>10</sup> grundlegende Kriterien von Bedeutung: Expressivität und Effektivität der Visualisierung.

<sup>8</sup>Online verfügbar Enzyklopädie, von jedem Benutzer editierbar, siehe [www.wikipedia.de](http://www.wikipedia.de)

<sup>9</sup>Siehe [hau]

<sup>10</sup>die Angemessenheit wurde nicht berücksichtigt, da sie im Grunde immer dann gegeben ist, wenn Expressivität und Effektivität erfüllt sind (Vgl. [Hei00, S.12])

Durch die Expressivität wird die Unverfälschtheit der Visualisierung beschrieben. Für die Expressivität gilt: „[...]nur die in den Daten enthaltenen Informationen und nur diese sollen durch die Visualisierung dargestellt werden.“[Hei00, S.9]

Die nachfolgende Darstellung zeigt ein Beispiel für expressive und ein Gegenbeispiel für nicht expressive Darstellung.

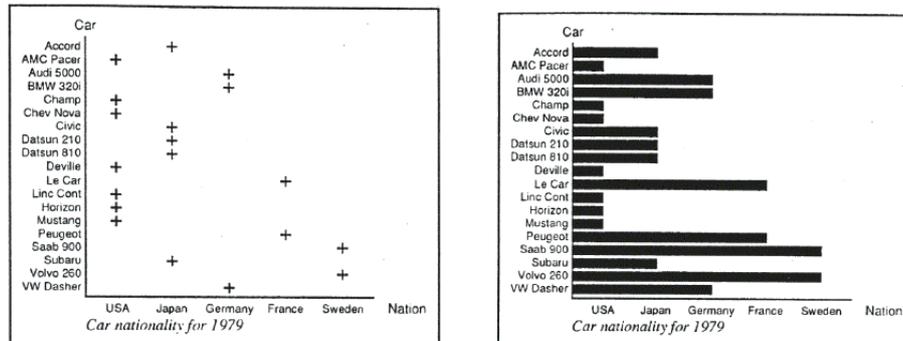


Abbildung 2.1: Herstellungsländ eines Autotyps in einem Scatterplot und in einem Balkendiagramm[Mac86]

Im Scatterplot ist die Zuordnung vom Autotyp zu Herstellungsländ eindeutig, während im Balkendiagramm eine qualitative Bewertung suggeriert wird (Länge der Balken steht für Qualität(Vgl.[Mac86]). Die Effektivität ist ein Maß für die Geschwindigkeit, in der ein Betrachter Informationen aus einer Darstellung entnehmen kann. Die Effektivität hängt somit direkt von den Fähigkeiten des menschlichen Sehapparates ab Informationen zu registrieren und zu ordnen und der Nutzung der Fähigkeiten des Sehapparates durch die Darstellung. Es wird unterschieden zwischen präattentiver und kognitiver Wahrnehmung.

Präattentive Wahrnehmung ist durch eine kurze Zeitspanne von 10 Millisekunden pro Element<sup>11</sup> gekennzeichnet. In dieser Zeit werden Reizmerkmale erkannt und Gruppierungen automatisch vorgenommen. Begleitet wird die präattentive Wahrnehmung von unfreiwilligen Augenbewegungen. Der präattentiven Wahrnehmung sind allerdings auch Grenzen gesetzt. Durch zu viele gleichartig gestaltete Elemente ist es für den Betrachter schwierig, spontan einen Bereich als wesentliches Reizmerkmal zu identifizieren.

Die präattentive Wahrnehmung kann in der Visualisierung durch eine förderliche Anordnung verbessert werden, indem die Konzentration gezielt auf eine bestimmte Textur gelenkt wird.

Kognitive Prozesse in der Wahrnehmung werden durch die präattentive Wahrnehmung nicht beeinflusst, benötigt allerdings mit *mindestens* 40 Millisekunden<sup>12</sup> pro Element wesentlich mehr Zeit als die präattentive Wahrnehmung.

In der folgenden Grafik wurde eine Übersicht von Wahlmöglichkeiten zunächst nach Zeilen und dann nach Spalten geordnet.

<sup>11</sup> Siehe [War00, S.165]

<sup>12</sup> Siehe [AT88]

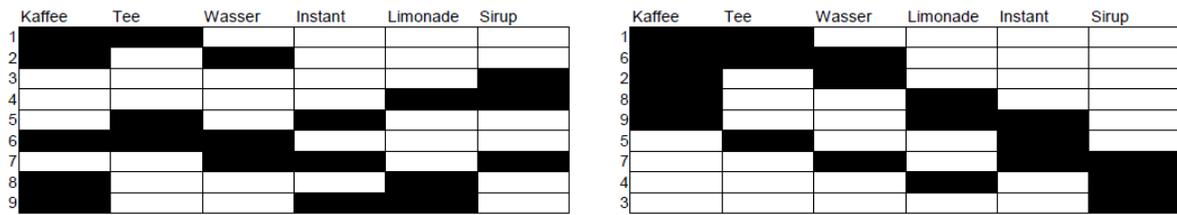


Abbildung 2.2: Durch Umstrukturierung präattentive Wahrnehmung verbessern<sup>13</sup>

Durch Versuche<sup>14</sup> wurde festgestellt, dass eine Hierarchie in der Wahrnehmung von verschiedenen visuellen Variablen existiert. Einige visuelle Variablen, die in dieser Arbeit in der Implementierung auftauchen:

- Position (x,y)
- Farbe
- Länge
- Verbindung

Die visuellen Variablen werden während des Prozesses des präattentiven Sehens unterschiedlich stark wahrgenommen. Es lässt sich eine Hierarchie der visuellen Variablen erstellen. Die Hierarchie ändert sich je nach der Art der Darstellung. Während die Position der Variablen „Position“ immer an erster Stelle bleibt, ist die „Länge“ in einer quantitativen Darstellung von entscheidender Bedeutung und ist somit in der Hierarchie an zweiter Stelle, während sie in einer ordinalen oder nominalen Darstellung nur eine untergeordnete Rolle spielt.

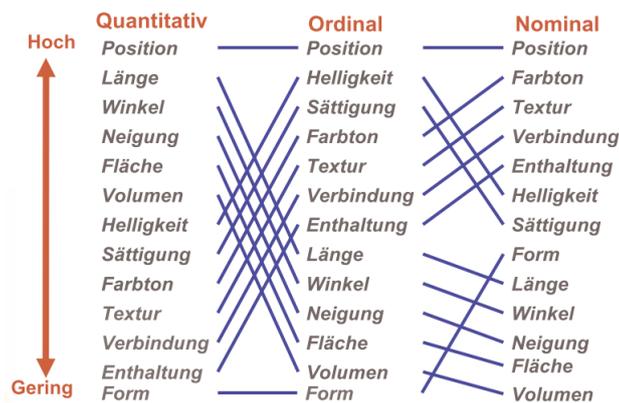


Abbildung 2.3: Relevanz der visuellen Variablen nach Abbildungsarten aus[Mac86] mit Adaptionen<sup>15</sup>

Zusammengefasst lässt sich sagen: Eine Visualisierung mit hoher Güte ist expressiv und effektiv,

<sup>13</sup>Vergleiche Verfahren anhand eines anderen Beispiels[Spe00, S.15]

<sup>14</sup>Siehe [WSC84]

<sup>15</sup>Grafik entnommen aus [Bre07]

achtet dabei auf die Struktur der Daten und die Wahrnehmungsfähigkeiten des menschlichen Sehapparates und versucht unter Ausnutzung des präattentiven Sehens zuerst auf wesentliche Informationen zu lenken und erst dann auf zweitrangige Informationen. Beispielsweise soll ein Pilot zuerst wahrnehmen, dass er Treibstoff verliert und nicht, ob sich die Wetterlage ändert.

### 2.4.2 Farbe

Din 5033 definiert Farbe wie folgt:

„Farbe ist ein durch das Auge vermitteltler Sinneseindruck, also eine Gesichtsempfindung. Die Farbe ist diejenige Gesichtsempfindung eines dem Auge strukturlos erscheinenden Teiles des Gesichtsfeldes, durch die sich dieser Teil bei einäugiger Beobachtung mit unbewegtem Auge von einem gleichzeitig gesehenen, ebenfalls strukturlosen angrenzenden Bezirk allein unterscheiden kann.“[Nor79]

Nach dem ersten Grassman'schen Gesetz (1853) besteht zwischen vier Farben immer eine eindeutige lineare Beziehung. Eine Farbe benötigt zu ihrer Beschreibung drei voneinander unabhängige Farben. Es folgt, dass Farbe eine dreidimensionale Größe ist, Farben somit als Vektor eines dreidimensionalen Raums aufgefasst werden können. Die Vektoren des Farbraums heißen Farbvalenzen, die linear unabhängigen Basisvektoren Primärvalenzen.

Im RGB-Modell, welches in dieser Arbeit verwendet wird, sind alle Darstellbaren Farben als Punkte im Einheitswürfel aufgefasst. Der Ursprung(0,0,0) ist als Schwarz definiert und der Endpunkt der Hauptdiagonalen im Einheitswürfel(1,1,1) als Weiß. Grauwerte liegen auf der Hauptdiagonalen. Rot (1,0,0), Grün(0,1,0) und Blau(0,0,1) bilden die orthogonalen, d.h. unabhängige, Basisvektoren des Einheitswürfels, mit denen sich alle anderen Farbwerte durch Vektoroperationen berechnen lassen. Ein Farbwert C ergibt sich aus der Gleichung  $C \equiv R + gG + bB$ <sup>16</sup>, wobei R,G,B die Basisvektoren und r,g,b Skalare sind.

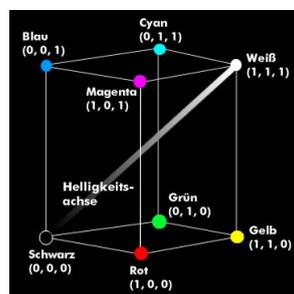


Abbildung 2.4: Darstellung des RGB Modells im Einheitswürfel[Lic]

Vorteile des RGB-Modells sind die Einfachheit und leichte Portierbarkeit, weshalb RGB z.B. in

<sup>16</sup>Aus [War00, S.107]

HTML und in der Bildverarbeitung zum Einsatz kommt. Problem des RGB-Modells zur Farbdarstellung ist die Definition der Basisvektoren R,G,B. Es existieren hierfür keine Richtwerte und je nach Monitor, Einstellungen des Monitors und der Grafikkarte und Betriebssystem kann sich die Farbdarstellung gravierend unterscheiden. Außerdem ist das RGB-Modell nicht auf die Farbwahrnehmung des Auges ausgerichtet. Zur schnellen Aufnahme von Informationen durch das Auge soll nach Bauer's Regel die konvexe Hülle des Einheitswürfels verwendet werden.

Farbe hat sowohl in der Natur als auch in der menschlichen Kultur eine symbolische Bedeutung. So kann eine rote Färbung des Tieres zum Ausdruck bringen, dass es hochgradig giftig ist wie z.B. die rote Speikobra. Im deutschen Verkehrswesen wird rot für Gefahrenzeichen verwendet und blaue Verkehrszeichen haben einen hinweisenden Charakter.

Ein weiterer Aspekt bei der Verwendung von Farben zur Visualisierung von Informationen ist, dass es Menschen gibt, die bestimmte Farbtöne nicht wahrnehmen oder unterscheiden können. Dies kann so weit gehen, dass ein Mensch gar keine Farben wahrnehmen kann. Deshalb muss eine gute Visualisierung diesen Menschen andere Hilfen wie geometrische Symbole oder Texte geben, um Elemente zu ordnen bzw. zu unterscheiden.

### 2.4.3 Diagramme

In diesem Abschnitt werden einige gebräuchliche Diagrammtypen im Bereich des Projektmanagements vorgestellt. Im Projektmanagement wird oft das Gantt-Diagramm verwendet, weshalb es dafür diverse Programme gibt, die sich mit der Erstellung von Gantt-Diagrammen beschäftigen.<sup>17</sup>

**Liniendiagramm:** Daten werden aus einer zweidimensionalen Relation (a,b) in ein zweidimensionales Diagramm mit zwei orthogonalen Achsen x,y übertragen. Dabei bildet sich die Skala für x aus dem Wertebereich von a bzw. die Skala für y aus dem Wertebereich von b. Eine Nebenbedingung für ein Liniendiagramm ist, dass für jeden Wert aus a nur ein Wert aus b zugeordnet werden kann. Die im Diagramm dargestellten Wertepaare (a,b) werden durch Linien miteinander verbunden. Liniendiagramme werden benutzt, um Messungen oder zeitliche Abläufe darzustellen.

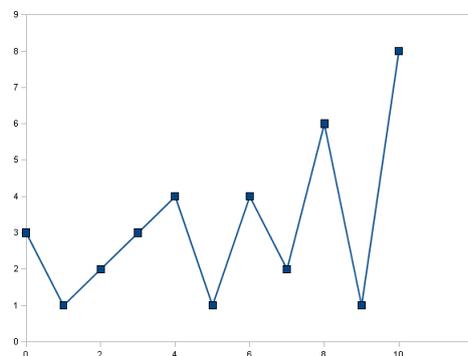


Abbildung 2.5: Ein Liniendiagramm

<sup>17</sup>Frei verfügbar ist das Gantt Project (<http://www.ganttproject.biz/>)



# Kapitel 3

## State-of-the-Art Analyse

### 3.1 Auswahl von Tools zur Analyse

Als Ausgangspunkt der Analyse wurde die Website Web2Null<sup>1</sup> gewählt. Aus den dort aufgelisteten Tools wurden diejenigen ausgewählt, die sich mit dem Management von Projekten, Kommunikation im Team oder Entwicklung beschäftigen, möglichst ein gutes Userfeedback erhalten haben und auf der Hersteller- bzw. Anbieterseite dafür werben, dass das entsprechende Tool sich durch Besonderheiten von anderen Tools abheben kann, also einen Mehrwert besitzen. Die gewählten Tools mussten außerdem frei verfügbar oder zumindest eine freie Testversion besitzen und online für jedes Teammitglied erreichbar sein.

Folgende Projekte wurden als empfehlenswert für eine genauere Analyse im Rahmen dieser Arbeit befunden:

- **Action Method** [Beh] wirbt mit einer Benutzeroberfläche mit Drag & Drop, was in vielen grafischen Oberflächen als Benutzungsschnittstelle zur Verfügung gestellt wird und somit eine leichte Bedienbarkeit suggeriert. Aufgaben eines Projektes sollen mit Action Method verwaltet und klar zugewiesen werden können.
- **Devunity** [Alo] konzentriert sich auf die gemeinsame Entwicklung im Team über das Internet. So soll Quellcode gemeinsam gelesen und bearbeitet werden können.
- **Planzone** [Aug] wirbt mit umfangreichen Funktionen für Projektmanagement wie ToDo-Listen, einem Planungstool, einer Dokumentenverwaltung, aus der sich Nutzer bei Bedarf benötigte Dokumente downloaden können sowie der Möglichkeit, ein Wiki über das Projekt zu führen und ein Erinnerungstools für Aufgaben, welches auf Emails basiert.
- **TeamBox** [Tea] behauptet von sich „ein Platz für ihr Team“ zu sein. TeamBox ist das einzige Tool in dieser Analyse, welches explizit auf Teams eingeht und deren Kommunikation erleichtern möchte.

---

<sup>1</sup>siehe [act]

Ausgeschlossen wurde GoalsOnTrack [Van], da zwar mit einem Tool geworben wird, welches die Ziele eines Projektes verfolgen und damit sicher stellen soll, dass das Projekt immer auf Kurs bleibt, was eine Analyse durchaus erstrebenswert gemacht hätte. Allerdings ist GoalsOnTrack nur bei einer kostenpflichtigen Mitgliedschaft einsichtig. Aufgrund der Informationen auf der Herstellerseite kann keine genügende Analyse durchgeführt werden.

Da inzwischen auch mit Wikis in vielen Unternehmen Dokumentation und Projektmanagement durchgeführt wird<sup>2</sup>, werden das MediaWiki und das TWiki als größte Vertreter aus dem Bereich der Dokumentation in die Analyse mit einfließen.

## 3.2 Durchführung der Analyse

### 3.2.1 Action Method

Action Method wirbt mit einer klaren Verwaltung von Aufgaben<sup>3</sup>, doch schon kurz nach dem Login macht sich Nüchternheit breit. Die Werkzeuge sind in ihrer Anwendung begrenzt und eine Zusammenarbeit zwischen den einzelnen Werkzeugen ist nur mangelhaft vorhanden. So kann ein Event oder eine Referenz in Form eines Links oder Dokumentes nicht mit einer Aufgabe, in Action Method als „Step“ bezeichnet, verknüpft werden. Die Oberfläche ist klar strukturiert, hat jedoch keine so genannten „Eye-Catcher“. Alles wirkt monoton gestaltet und Aufgaben lassen sich nur in drei festen Farben gestalten: ein helles Blau, ein Orangeton oder ein helles Grau. Die beworbene Drag & Drop Funktionalität kommt nur in der Projektübersicht zum tragen und ist schlecht durchdacht. Beim Bewegen eines Elementes springen alle Elemente, über die mit der Maus gefahren wird hin und her und stören so teils die gewünschte Ordnung. Auch besitzt die Drag & Drop Funktion einige Fehler. So kann es passieren, dass ein Element nicht immer an einen anderen Platz verschoben werden kann. Positiv an Action Method fällt die Diskussionsfunktionalität<sup>4</sup> auf. Diese Funktionalität stellt eine wesentliche Anforderung an effizientes, virtuelles Teammanagement dar. Es ist möglich Ideen für später zu hinterlegen als so genannte *Backburners*<sup>5</sup>. Auch hier fehlt wieder die Möglichkeit, eine Idee an eine Aufgabe zu binden.

Ein gravierendes Problem in Action Method ist die Persistenz der Dokumentation. Allein durch das Anklicken der Checkbox in einer Aufgabe wird diese unwiderruflich gelöscht. Damit wird es unmöglich nach dem Projekt dieses zu validieren und Fehlerquellen zu finden.

**Fazit:** Action Method dient bestenfalls als persönlicher Projektplaner, der sich Ideen für später merken oder aktuelle Aufgaben anzeigen kann. Für eine Projektdokumentation fehlt es Action Method an den entsprechenden Funktionalitäten. Informationen werden nicht persistent gemacht und es fehlt eine Übersicht über das Gesamtprojekt und dessen Historie.

---

<sup>2</sup>Vgl. Kapitel 2.3.2

<sup>3</sup>Wer mehr als 50 Aufgaben verwalten will, muss eine Gebühr an den Hersteller entrichten

<sup>4</sup>Vgl. Anforderung an Beispieltool [Ger02]

<sup>5</sup>dt.: Nachbrenner

The screenshot displays a project overview in the Action Method system, organized into five main sections:

- Discussions (1 of 1 discussion topics):** A table with columns for Title, Latest Post, and Notification. One entry is visible: "Diskussion" with 1 post by 1 person, dated 07/31/2009 at 04:35 AM by You, and a notification toggle set to OFF.
- Action Steps (1 Results):** A dropdown menu is set to "Active". Below it, a light blue box contains a checkbox and the label "Step1".
- References (1 Results):** A table with columns for Link, Title, and Shared. One entry is visible: "Ref1".
- Backburners (1 Results):** A grey box contains the text "Ich hab da mal ne Idee für später". Below the text are links for "Edit", "Sharing", and "Trash".
- Events (1 Results):** A single entry is visible: "Event 1" with 0 people, dated 08/01/2009, and an "Edit" link.

Abbildung 3.1: Gesamtübersicht über ein Projekt in Action Method.[Beh]

### 3.2.2 Devunity

Devunity ist jedem Nutzer in vollem Umfang frei zugänglich. Es ist dem Nutzer freigestellt ein Projekt zu erstellen oder einem bereits bestehenden Projekt beizutreten. Da Devunity auf die Entwicklung von Software ausgerichtet ist, muss bei der Erstellung eine oder mehrere Programmiersprachen gewählt werden sowie die Art der Lizenz. Ein Projekt kann privater Natur oder Open Source sein. Die Entwicklung mit Devunity ist auf die Script- bzw. Metasprachen PHP, Python, JavaScript, Perl, HTML, ASP, CSS und RUBY beschränkt. Dadurch ist das Einsatzgebiet von Devunity auf die Entwicklung von Applikationen für das World Wide Web beschränkt. Devunity stellt für die Entwicklung auf Wunsch die API Schnittstelle bekannter Webseiten wie Flickr, Google, Yahoo, Twitter oder Wordpress bereit. Für jedes Projekt kann eine Beschreibung in Form eines Textes erstellt werden. Weiterhin besitzt jedes Projekt einen Terminkalender, mit dem Events verknüpft werden können. Bei Problemen in der Entwicklung kann im Forum von Devunity um Rat gefragt werden. Dort können alte Beiträge gelesen und Lösungen in einem „Code Basket“ mitgenommen werden. Diskussionen werden grundsätzlich

um Programmcode gestaltet.

Quick Issue close

Title:

Type:  Bug  ToDo

Severity:  Blocker  Critical  Normal  Enhancement

Priority:  High  Medium  Low

Privacy:  Private  Public

Assigned to:  ▼

Tags: (Seperated by Commas)

Abbildung 3.2: Devunity überzeugt im Detail mit intelligent gestalteten Menüs und Visualisierungen.[Alo] Leider funktionieren viele Funktionalitäten nur eingeschränkt.

Devunity zeichnet sich durch die Möglichkeit aus, einen kostenlosen SVN Server für das Projekt nutzen zu können, um so Code zu Versionen und die Entwicklung persistent zu machen. Mit dem Editor können die Team-Mitglieder eines Projektes gleichzeitig an Programmcode arbeiten und miteinander über einen Chat kommunizieren. Leider war es zum Zeitpunkt als diese Analyse durchgeführt wurde nicht möglich, den Editor hinreichend zu testen. Im Internet Explorer 8 funktionierte er aufgrund von JavaScript Fehlern gar nicht, im FireFox 3.5.2 war es nicht möglich, Dateien zu öffnen und in Opera 9.64 konnten geöffnete Dateien nicht editiert werden. Einzig die Tatsache, dass Devunity von sich selbst sagt, dass es sich noch in der Alpha-Phase der Entwicklung befindet, rechtfertigt in irgendeiner Weise die gravierenden Probleme mit dieser Funktionalität, obwohl Devunity explizit mit dieser Möglichkeit wirbt und sich dadurch von anderen Plattformen abheben möchte. Als Kommunikationsmöglichkeiten zwischen Team-Mitgliedern wird auf soziale Netzwerke und Kommunikation via Chat verwiesen. Es besteht die Möglichkeit Kontaktdaten zu hinterlassen.

**Fazit:** Devunity ist aufgrund des Fehlens von Tools für Projektmanagement wie Zeitplänen oder Aufgabenverteilung, weshalb ein weiteres Tool parallel zu Devunity notwendig wäre, und der Tatsache, dass einige Funktionalitäten nur umständlich oder gar nicht funktionieren, nicht für virtuelles Teammanagement empfehlenswert<sup>6</sup>.

<sup>6</sup>Vgl. Bedingungen für technische Funktionalität bzw. zentrale Verwaltung Kapitel 2.1.3 und 2.1.4

### 3.2.3 Planzone

Planzone stellt sich selbst als Tool für Projektmanagement und Zusammenarbeit dar und ist in den Sprachen Englisch, Französisch und Deutsch verfügbar. Nach der Anmeldung startet ein Wizard, der dabei hilft, ein Projekt zu erstellen. Die Oberflächen sind klar strukturiert und der Benutzer wird durch klare Anweisungen und Hervorhebungen durch das Menü geleitet. Planzone besitzt eine umfangreiche Toolsammlung wie das Erstellen von Aufgaben, wobei Aufgaben mit anderen Team-Mitgliedern geteilt bzw. mehreren Team-Mitglieder zugewiesen werden können. Eine Aufgabe kann direkt mit dem Projekt oder einer Unteraufgabe verknüpft sein. Wird eine Aufgabe abgeschlossen, so bleibt diese als abgeschlossen im Speicher erhalten und kann jederzeit wieder betrachtet werden. Planzone verfügt über zwei umfangreiche Zeitplaner. Im persönlichen Planer kann sich der Benutzer seine eigenen Aufgaben über alle Projekte ansehen, im Projekt den gesamten Projektverlauf. Neben der Arbeitszeiteinteilung kann auch eine Rückmeldung über die aktuelle Planung erfolgen, um z.B. bei einem Ausfall schneller reagieren zu können. Weiterhin verfügt Planzone über die Möglichkeit, Diskussionen zu starten, Dokumente abzulegen, ein Wiki anzulegen und automatisierte Berichte für das aktuelle Projekt automatisch für Excel generieren zu lassen die dem Projektleiter durch entsprechende Einstellung zu bestimmten Zeiten regelmäßig zugeschickt werden. Problem hierbei ist, dass Exceldokumente nur in Excel immer korrekt dargestellt werden und Excel nicht frei verfügbar ist. Auch ist nur die Zusammenfassung, nicht aber die Projektdetails, in jeder Version von Planzone sofort verfügbar. Um die Projektdetails sichtbar zu machen, muss erst eine Lizenz für Planzone in entsprechendem Umfang erworben werden. Auffallend an Planzone ist die vielfältige Nutzung von Gantt diagrammen und Übersichten. In Planzone fehlt allerdings, ebenso wie in Action Method die Möglichkeit, Dokumente direkt mit Aufgaben zu verknüpfen. Manche Menüs wirken etwas überladen und es ist nicht immer sofort ersichtlich, welcher Button wofür zuständig ist.

Ein besonderes Feature in Planzone ist die Zuteilung von Aufgaben in Unteraktivitäten mittels Drag & Drop. Diese Funktionalität kann z.B. bequem eingesetzt werden, um jedem Teammitglied offene Aufgaben entsprechend zuzuteilen, indem ein Ordner „offene Aufgaben“ heißt und jedes Team-Mitglied seine Ordner hat, in die bei Bedarf offene Aufgaben verschoben werden können.

**Fazit:** Planzone ist ein umfangreiches Tool für Projektmanagement, in dem detailliert Aufgaben und deren Status festgelegt und betrachtet werden kann. Durch automatische Berichte und Statistiken entfällt teilweise Arbeit sich anhand der Daten den Projektstatus erarbeiten zu müssen und kann sich der Benutzer auf Problemstellen konzentrieren. Planzone bietet somit alle notwendigen Tools für die erfolgreiche Umsetzung eines Projektes. Virtuelles Teammanagement ist in Planzone jedoch nicht ausreichend integriert, insbesondere durch fehlende Lenkung bei der Dokumentation und Verknüpfung der Dokumentation mit den Aufgaben kann die Qualität je nach Team schwanken. Auch hat das Team durch die komplexen Übersichten, in denen eine effektive visuelle Hervorhebung fehlt, nicht immer sofort den Überblick, wo es eventuell Probleme geben könnte, weil eine Aufgabe viel zu komplex ist. Die Oberfläche wirkt überladen und durch den Einsatz von vielen sanften Tönen werden wichtige Informationen nicht genügend hervorgehoben.

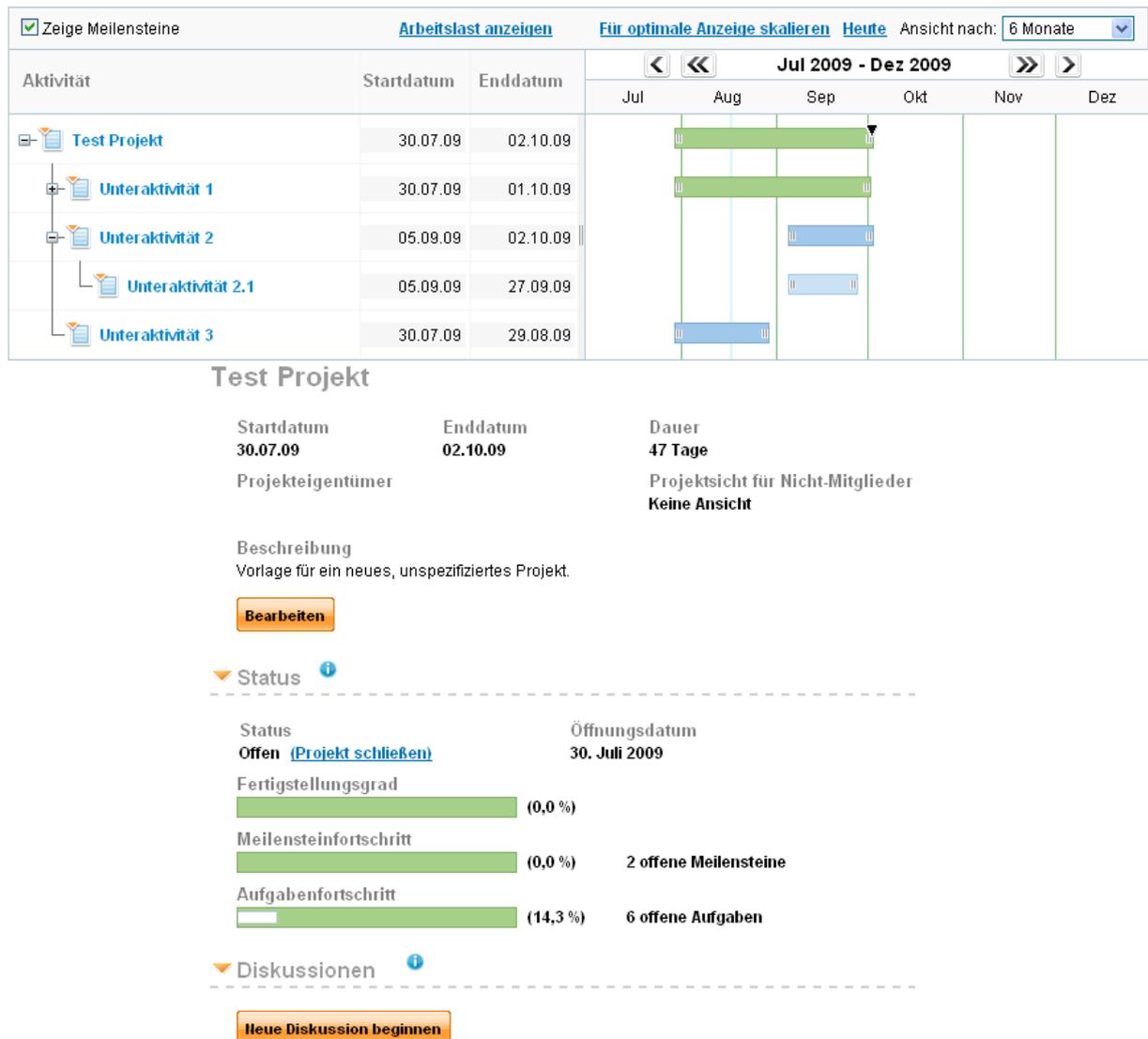


Abbildung 3.3: Planzone überzeugt mit klar strukturierten Übersichten, die alle wichtigen Informationen zu vermitteln wissen.[Aug]

### 3.2.4 TeamBox

TeamBox ist ein Open Source Tool<sup>7</sup>, welches sich ausschließlich der Kommunikation im Team widmet. So fehlen Tools zum Projektmanagement bis auf eine Aufgaben und eine ToDo-Liste sowie die Möglichkeit Seiten („Pages“) anzulegen, auf denen Informationen festgehalten werden können. Sonst werden im Internet gebräuchliche Kommunikationsformen angeboten: ein Chat, ein Nachrichtensystem und eine Adressliste. Um das Tool zu nutzen, muss einer recht undeutlichen AGB zugestimmt werden, in der von möglichen Änderungen die Rede ist, die jederzeit erfolgen können. Auch von Ko-

<sup>7</sup>Die API steht kostenlos zum Download zur Verfügung

sten ist die Rede, wobei die Dienste, auf welche diese fallen könnten, nicht näher genannt werden. Es ist allerdings davon auszugehen, dass der Account beim Erstellen erst einmal kostenlos ist, da auch keine Kreditkartenangaben abgegeben werden müssen. Aus Sicht der Visualisierung kann sich TeamBox nicht hervorheben. Die Seiten sind einheitlich gestaltet und es wird nicht versucht, Informationen visuell hervorgehoben.

**Fazit:** TeamBox zeichnet sich weder durch Mehrwert gegenüber anderen Tools noch durch irgendeine Besonderheit aus. Es werden Funktionen aus Wikis wie die Seiten genutzt, um dynamisch Inhalte anzulegen, allerdings ohne größere Navigationsstruktur. Die Aufgabenübersicht gestaltet sich recht dürftig und die Kommunikationsformen sind in Messengertools wie Pidgin<sup>8</sup> mit diversen Protokollen in besserer Form vertreten. TeamBox beachtet nicht den Hintergrund virtueller Teams, legt keine Dokumentation an und ist somit als Tool für Projekte in Unternehmen nicht einsetzbar.

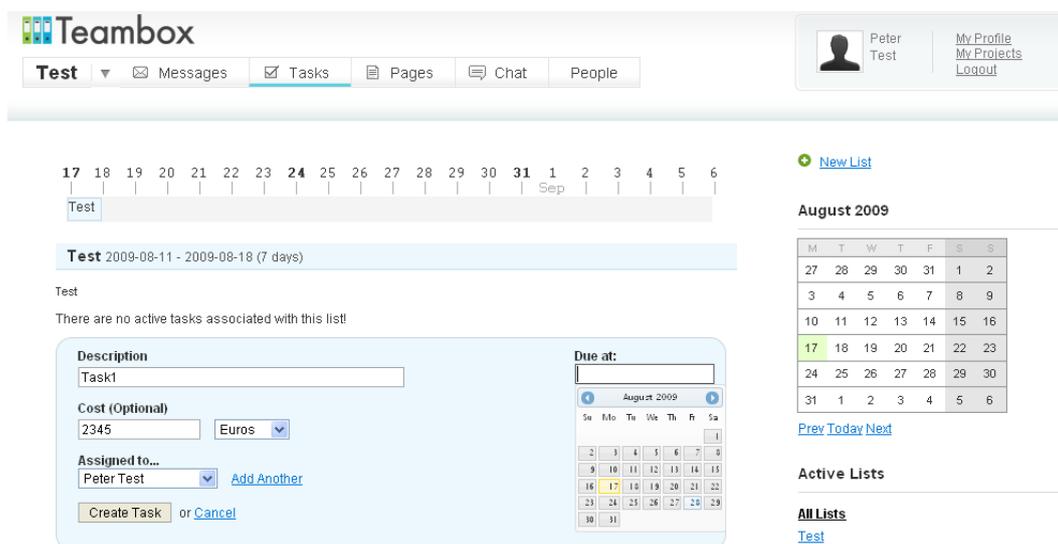


Abbildung 3.4: Teambox setzt auf Kommunikation und schlichte Übersichten.[Tea]

### 3.2.5 MediaWiki als Dokumentationswerkzeug

Wie in den Grundlagen<sup>9</sup> erwähnt, handelt es sich bei dem MediaWiki um die Basis für Wikipedia, welche in PHP geschrieben wurde und u.a. der Bereitstellung und Bearbeitung von Informationen im Netz dient. Dadurch, dass das MediaWiki eine Datenbank nutzt, kann es mit PHP Skripten leicht angepasst werden. MediaWiki bietet von sich aus Schnittstellen zur API<sup>10</sup>, um neue Funktionalitäten zu integrieren oder um Funktionen aus dem MediaWiki automatisch aufzurufen wie zum Beispiel einen Artikel anlegen und mit Inhalt füllen. Dadurch ist das MediaWiki bestens geeignet, um Dokumentation on-the-fly im MediaWiki oder externen Tools zu erstellen. Durch eine entsprechende Erweiterungen könnte mit einem Wiki der Projektstatus visualisiert werden. Alle notwendigen Funktionalitäten

<sup>8</sup>Siehe [Pid]

<sup>9</sup>Vgl. Kapitel 2.3.1

<sup>10</sup>Siehe [Wika]



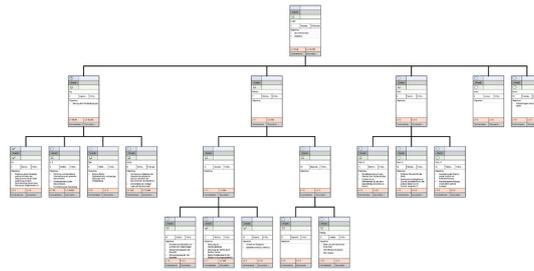


Abbildung 3.6: Ein möglicher vollständiger Projektbaum[iata]

### 3.3.2 proscio

Die Software proscio<sup>12</sup> ist als Controllingsoftware für Projekte gedacht und beschäftigt sich hauptsächlich mit dem Verwalten des Budgets und anderen Ressourcen wie Mitarbeitern eines Projektes. Im Gegensatz zu project illustrator ist proscio eine Applikation für den Desktop PC. Auch in dieser Software ist Werkzeug in Form eines Projektbaums integriert, welches bei der Gliederung des Projektes helfen soll. Im Gegensatz zu project illustrator ist der Projektbaum bei proscio nicht zur Visualisierung, sondern zur Gliederung des Projektes gedacht.

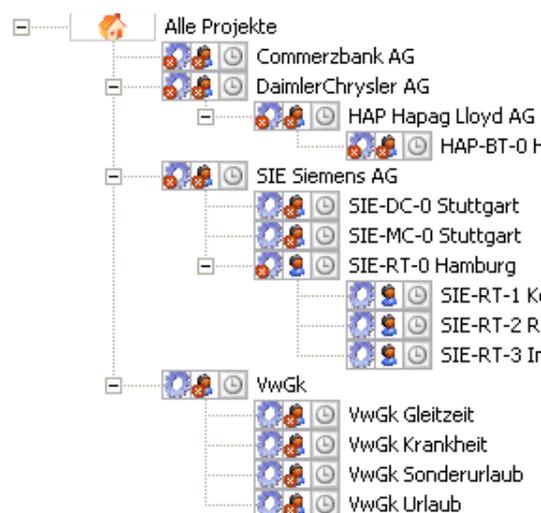


Abbildung 3.7: Der Projektbaum als Gliederung der Aufgaben in einer Übersicht[EFK06]

### 3.3.3 Zusammenfassung

Die zwei ausgewählten Tools, in deren Beschreibung das Wort „Projektbaum“ gefunden werden konnte, haben diesen auf verschiedene Arten umgesetzt. Während project illustrator sich auf die plattformunabhängige Visualisierung des Projektstatus konzentriert, sieht proscio im Projektbaum eine Methode, um ein Projekt zu gliedern und aus dieser Gliederung heraus auf einem Desktop PC zu verwalten.

<sup>12</sup>Siehe [EFK06]

### 3.4 Fazit der Analyse

Aus den vier von Web2Null gewählten Tools ist nur Planzone für den Einsatz in größeren Projekten und somit im professionellen Bereich zum empfehlen. Allerdings ist Planzone in diesem Fall kostenpflichtig, da 25MB Speicher und die Beschränkungen nicht ausreichend sind, um ein größeres Projekt gut führen zu können. Einige Features und Sonderwünsche wie den Einsatz im Intranet stehen erst gegen einen entsprechenden Vertrag mit Planzone zur Verfügung. Planzone sorgt allerdings nicht automatisch für eine qualitativ hochwertige Dokumentation, da diese weiterhin vom Team selbstständig erstellt und gepflegt werden muss in Form von Dokumenten, die nicht direkt online einsehbar<sup>13</sup> und editierbar sind.

Bei den vorgestellten Wikis hat jedes seine Stärken und Schwächen im Einsatz als Dokumentationswerkzeug. Während TWiki insbesondere im Intranet und der Umwandlung von Dokumenten punkten kann, ist MediaWiki auf dynamisch im Inter- bzw. Intranet generierte Inhalte ausgerichtet. Beide eignen sich zur Erstellung einer Projektdokumentation. Die Wahl hängt von den Anforderungen und den gewünschten Erweiterungen bzw. deren Umfang ab.

---

<sup>13</sup>sofern es sich nicht um Dokumente handelt, die ein Browser selbst öffnen kann

# Kapitel 4

## Anforderungsanalyse

### 4.1 Begründung der Notwendigkeit eines weiteren Tools

Aus der State-of-the-Art Analyse ist deutlich geworden, dass viele Hersteller von Software im Bereich Projektmanagement im Web 2.0 auf teils ungelenkte Kommunikation und Aufgabenverwaltung im Team setzen, also keine Hilfen bieten zu erkennen wo sich kritische Abschnitte im Projekt befinden. Dadurch gehen strategisch wichtige Informationen für die Projektbearbeitung verloren. Jedes Tool ist auf eine bestimmte Sichtweise ausgerichtet und die Verknüpfung von Informationen ist in allen untersuchten Tools beschränkt. Auf die Dokumentation des Projektes und deren dauerhafte Verwahrung wurde oft keinen Wert gelegt. Eine explizite Problemverwaltung ist nur in Devunity in Form von Bugtracking vorgefunden worden. Bei anderen Tools wie Planzone könnte eine Problemverwaltung entweder durch direktes einfügen in die Aufgabenbeschreibung oder in Form von Dokumenten erstellt werden, was allerdings die Effektivität im Finden von bekannten Problemstellungen einschränkt. Auch durch die Nutzung der Diskussionsfunktion wird keine unmittelbare Übersicht zu einzelnen Problemdetails geschaffen.

Ein weiterer Aspekt ist, dass die meisten Tools ohne entsprechende Lizenz nur eingeschränkt nutzbar sind. Eine Integration in bereits bestehende Umgebungen einer Firma wie ein Wiki als Erweiterung zur Dokumentation war in keinem Tool vorgesehen. Dies wäre notwendig, um eine zentrale Verwaltung von Informationen sicher zu stellen.<sup>1</sup>

Weiterhin sind die bisher vorgestellten Tools in ihrem Funktionsumfang nicht oder nur unter großem Aufwand erweiterbar. Sind weitere Funktionen gewünscht, so muss entweder ein anderes Tool gewählt oder eine Anpassung des aktuellen Tools vorgenommen werden. Beides ist meist mit hohem Zeitaufwand verbunden. In der Praxis kommt es häufiger vor, dass mehrere Tools benutzt werden. So wird z.B. Mantis für Bugtracking eingesetzt, BSCW oder SVN für die Verwaltung von Dokumenten und ein Forum, ein Wiki oder Emails für Kommunikation.

Somit bietet sich im Rahmen dieser Arbeit die Entwicklung eines Tools an, welches an bereits bestehende Umgebungen einer Firma angefügt werden kann oder bei der Entscheidung für ein Tool beliebig

---

<sup>1</sup>Vgl. Kapitel 2.1.4

erweiterbar ist ohne Einschränkung auf bestimmte Funktionen wie Specialpages beim MediaWiki<sup>2</sup>.

## 4.2 Zielgruppe

Das Tool, welches im Rahmen dieser Arbeit konzipiert wird, richtet sich an Entwickler bzw. Entwicklerteams im Umfang von etwa fünf bis zwanzig Personen. Das Team soll in Zuständigkeiten und Rollen unterteilbar sein und jedes Teammitglied je nach Zuteilung in dem entsprechenden Projekt bzw. einer Aufgabe des Projektes Tools zur Verfügung gestellt bekommen, die es ermöglichen, den Stand des Projektes sowie Probleme auf visuellem Wege möglichst schnell zu erfassen und ggf. zu aktualisieren.

## 4.3 Motivation der Benutzer

Damit die Benutzer dazu motiviert sind, das Tool dauerhaft zu benutzen, muss es in der Bedienbarkeit möglichst einfach zu handhaben, also intuitiv, sein. Würde das Tool beispielsweise an ein Wiki gekoppelt, so darf die Manipulation des Artikels z.B. hinsichtlich seines Namens nur gewünschte Folgen haben, z.B. indem dadurch die Kategorie oder die Komponente einer Aufgabe geändert wird, aber nicht, dass dadurch Aufgaben plötzlich nicht mehr zugeordnet oder gefunden werden können und somit keinem Projekt mehr angehören. Auch darf die Historie der Aufgabe und somit ihre Dokumentation nicht verloren gehen. Das Tool muss somit in jedem Anwendungsschritt möglichst robust und gleichzeitig klar ersichtlich sein, welche Reaktion auf eine Interaktion erfolgt.

Ein weiterer Aspekt der Motivation ist die Selbsterklärung des Tools durch verständliche Texte und Icons sowie Hilfen bei größeren Werkzeugen. Auch darf der Funktionsumfang nicht zu groß sein, da eine Überladung von Informationen und Tools dazu führt, dass viele Tools nicht benutzt werden und dadurch wichtige Informationen unter Umständen nicht ausreichend dokumentiert werden, während weniger wichtige Informationen zu umfangreich bearbeitet werden. Eine Priorisierung der Tools z.B. durch Einschränkung des Umfangs der Interaktionsmöglichkeiten der Tools ist eine mögliche Maßnahme.

## 4.4 Beispieltool für virtuelles Projektmanagement

Grundlage für das zu erstellende Tool ist ein Beispieltool, welches von Graf und Jordan<sup>3</sup> vorgestellt wird. Es enthält grundlegende Funktionen im Bereich des Teammanagement, welche für effektives Arbeiten unerlässlich sind. Das Beispieltool enthält im einzelnen Personen- und Adresslisten, visualisiert den Projektstatus für jeden Arbeitsschritt („Arbeitspakete“) sowie entscheidende Meilensteine. Weiterhin enthalten sind eine Dokumentbibliothek, ein Kommunikationsforum für Diskussionen sowie ein Terminkalender. Aus der Zusammenstellung wird ersichtlich, dass bei Graf und Jordan im

---

<sup>2</sup>Siehe [Wikb]

<sup>3</sup>Siehe [Ger02]

virtuellen Teammanagement weniger Wert auf die Dokumentation als auf die Kommunikation der Arbeitsergebnisse gelegt wird. Das Diskussionsforum ist jedoch als Dokumentationswerkzeug in der Hinsicht von besonderem Interesse, dass dadurch indirekt auch Entscheidungsprozesse während der Diskussion dokumentiert werden. Dies erleichtert eine Analyse von Entscheidungen z.B. für Folgeprojekte, insbesondere dann, wenn diese in größerem zeitlichen Abstand bzw. anderem personellen Hintergrund erfolgen.

## 4.5 Anforderungen virtuelles Team- und Projektmanagement

Die im Beispieltool von Graf und Jordan genannten kommunikativen Komponenten, die erforderlich für virtuelles Teammanagement sind, wie ein Diskussionsforum und ein Terminkalender sollten in jedem Fall im Konzept festgehalten werden, da sie eine grundlegende Anforderung des Teams an die Kommunikation darstellen.

Damit sich jedes Team-Mitglied über den Projektstatus einen Überblick verschaffen kann, sind Übersichten über den gesamten Projektfortschritt sowie die verbleibende Restzeit und unerledigte Aufgaben erforderlich. Die Betrachtung dieser Informationen muss aus Sicht der Visualisierung möglichst effektiv gestaltet sein, d.h. es dürfen nicht zu viele Informationen auf einmal angezeigt werden. Eine Zoomfunktion, die Details verschiedener Aufgaben dem Benutzer auf Wunsch verdeutlicht, ist deshalb eine essentielle Anforderung an das Konzept bzw. Tool, welches in dieser Arbeit entwickelt wird.

Eine weitere wichtige Funktionalität, die alle Bereiche tangiert, ist die Möglichkeit, Informationen und Aufgaben sowie Aufgaben untereinander zu Verknüpfen, um es den Benutzer zu erleichtern Zusammenhänge zwischen den einzelnen Arbeitsschritten herzustellen und zu validieren.

## 4.6 Rollenspezifische Anforderungen

Obwohl in der Anforderung von Graf und Jordan an ein Tool für ein virtuelles Team kein information hiding betreiben<sup>4</sup> sollte, kann es manchmal notwendig sein, bestimmte Informationen nicht jedem Nutzer zu präsentieren. Insbesondere nicht Team-Mitglieder eines Projektes wie Kunden oder Mitglieder eines anderen Projektes sollen keine internen Informationen erhalten, sofern ihnen kein direkter Zugang gestattet wird, sondern nur die Informationen, die vom Projektteam für den Kunden bzw. Mitglieder eines anderen Projektes gedacht sind. Es ergibt sich für die einzelnen Rollen folgende Präsentationsvorgabe der Informationen:

- **Abteilungsleitung:** Die Informationen der gesamten Abteilung müssen präsentiert werden. Dies schließt insbesondere eine Verwaltung für alle Projekte auf oberster Ebene mit ein.

---

<sup>4</sup>Vgl. Kapitel 2.1.4

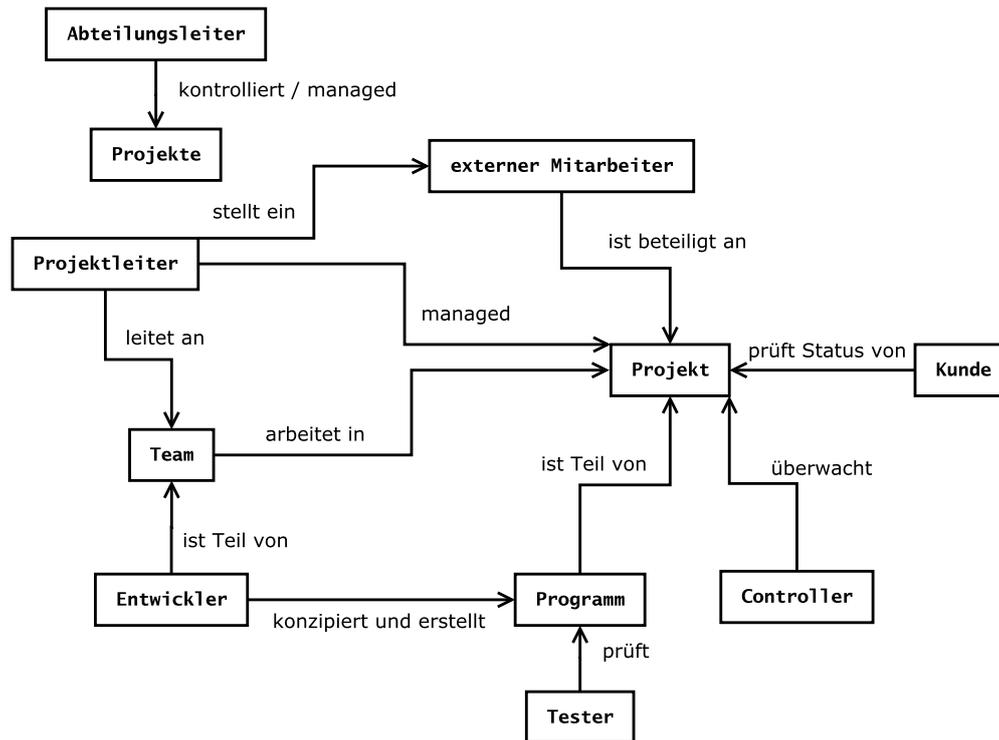


Abbildung 4.1: Wesentliche Aufgaben der einzelnen Rollen.

- **Projektleitung:** Alle Informationen der zu leitenden Projekte sind sichtbar sowie öffentliche Informationen anderer Projekte. Auf Einladung durch den Abteilungs- oder Projektleiter soll auch weitere Information eines anderen Projektes sichtbar werden können.
- **Entwickler:** Information aus Projekten sichtbar, an denen der Entwickler als Team-Mitglied beteiligt ist.
- **Tester:** Information zugeteilter Projekte sichtbar, die für den Test notwendig sind. Für den Tester kann zusätzlich eine Testspezifikation erstellt werden, die zu erfüllen ist.
- **Beobachter / Controller:** Projektstatus wird visualisiert durch den aktuellen Fortschritt. Projektleitung bzw. Entwickler können Informationen zur Beobachtung freigeben. Dabei kann eine Information, die die Projektleitung erstellt, nicht vom Entwickler freigegeben werden.
- **Kunde:** Ähnlich dem Beobachter kann nur der aktuelle Stand des Projektes sowie eine aktuelle Darstellung der erfüllten Anforderungen oder für den Kunden hinterlassene Informationen wie Notizen oder Termine präsentiert werden.
- **externer Mitarbeiter:** Da die Aufgaben eines externen Mitarbeiters recht unterschiedlich sein können, müssen seine Aufgaben und Berechtigungen vom Projektleiter definiert werden.

Das Management für Rollen und deren Berechtigung muss eindeutig und je nach Projekt differenzierbar sein. Wenn ein Kunde in einem Projekt erweiterte Rechte bekommt, so soll dies nur für dieses

Projekt geschehen. Eine weitere Möglichkeit ist es Projekttypen vorzudefinieren wie z.B. ein Entwicklungsprojekt, ein Controllingprojekt, ein Folgeprojekt oder ein Projekt in enger Zusammenarbeit mit Kunden oder externen Partnern.

## 4.7 Controlling: Bewertungskriterien für ein IT-Projekt

Bei der Durchführung eines Projektes stellt sich immer die Frage, ob die Anforderungen an die Entwicklung innerhalb der erforderlichen Zeit mit dem zur Verfügung stehenden Budget erreicht werden können.

Gerald Weinberg beschreibt in seinem Buch „Die Psychologie des Programmierers“<sup>5</sup> die Problematik ein Programm zu bewerten. Wichtige Kriterien sind, ob rechtzeitig fertig gestellt wurde, was es gekostet hat und ob die Anforderungen erfüllt wurden. Wie genau der Quellcode aussieht spielt dabei eine untergeordnete Rolle, da es eine komplexe Aufgabe ist, Code zu bewerten. Dies soll jedoch nicht heißen, dass ein ineffizientes Programm nicht von einem effizienten Programm zu unterscheiden ist, oder die Erfüllung von Richtlinien in der Programmierung nicht zu befolgen sind.

Nach Weinberg ist die Erfüllung der Spezifikation der wichtigste Aspekt bei der Verifikation eines Programmes, wichtiger als die Effizienz, Anwendbarkeit und Produktionskosten. Funktioniert ein Programm nicht, so ist es wertlos. Weinberg bemerkt jedoch in der aktuellen deutschen Ausgabe<sup>6</sup>, dass inzwischen die Funktionalität auch im Sinne der Benutzbarkeit eine wichtigere Rolle spielt als noch vor 25 Jahren und die reine Erfüllung der Spezifikation und die Laufzeit, welche vor 25 Jahren noch ein höheres Maß für die Qualität eines Programms gewesen ist.

Es ergibt sich eine Hierarchie in der Kontrolle der Entwicklung. Besonders hohe Priorität hat dabei, dass die Spezifikation in der gegebenen Zeit erfüllt wird. Dabei darf allerdings die Benutzbarkeit des Programms nicht vernachlässigt werden. Auch muss überprüft werden, ob sich Anforderungen in der Spezifikation nicht gegenseitig behindern und den Erfolg des Projektes gefährden. Dies muss zwar schon in der Konzeption des Projektes bzw. des Programmes bei den Verhandlungen mit dem Kunden geschehen, allerdings fallen viele Probleme erst in der Entwicklung auf. Deshalb ist es während der Entwicklung wichtig, jederzeit zu prüfen, ob Spezifikationen auch erfüllbar sind und nicht wertvolle Zeit daran verloren geht im Versuch, Spezifikationen zu erfüllen. In Kapitel 2.1.2 wurde erwähnt, dass in vielen Projekten „[...]der Elefant hinter der Mücke[...]“ nicht gesehen wird. Stellt sich im Laufe des Projektes heraus, dass es mit den gegebenen Ressourcen nicht möglich ist, die Spezifikation zu erfüllen, oder dass die Erfüllung einer Spezifikation aufwändiger ist als erwartet, so muss umgehend eine Gegenmaßnahme ergriffen werden wie der Kontakt mit dem Kunden, eine Änderung der Spezifikation oder eine Streichung von nicht notwendigen Anforderungen. In jedem Fall ist es sinnvoll bei der Vertragsaushandlung die Komplexität von Anforderungen festzulegen und zu prüfen, um während des Projektes bei Änderungen der Spezifikation oder deren Bewertung vertragsrechtlich nicht in Problemsituationen zu geraten.

---

<sup>5</sup>Siehe [Wei04, S.53-60]

<sup>6</sup>[Wei04] ist eine Neuauflage zum 25. Jubiläum des Buches

Die Rolle des Controllers bzw. der Projektleitung wird somit begleitet von der Anforderung, den Projektstatus nicht nur zu überwachen, sondern auch aus den dargestellten Informationen Rückschlüsse über den zukünftigen Verlauf und das Endergebnis des Projektes ziehen zu können und ggf. einzugreifen.

## 4.8 Anforderungen an das Tool

Das Tool soll den Projektstatus dokumentieren, visualisieren und überprüfbar machen. Dabei sollen dem Benutzer so viele Informationen wie möglich zu einem Teilbereich des Projektes über eine Anzeige erreichbar machen in Form von direkter Visualisierung oder Verweisen. Dafür wird eine Struktur benötigt, die das Projekt gliedern und anzeigen kann wie ein Projektbaum. Die Oberfläche soll möglichst einfach, dabei jedoch auch sehr ausdrucksstark gestaltet sein. Das Tool darf nicht überladen wirken, also nicht mit zu vielen Informationen und Anzeigen von wichtigen Thematiken ablenken. Die einzelnen Komponenten des Tools müssen zusammenarbeiten können, um Zusammenhänge im Projekt besser darstellen zu können. Das Tool benötigt ein tiefgehendes, hierarchisch aufgebautes Rollensystem, welches für jedes Projekt separat verwaltet werden kann.

Für die Dokumentation des Projektes wird die Möglichkeit benötigt, Ergebnisse der Entwicklung, Modulkataloge und Konzepte sowie Testpläne, Testverfahren und Ergebnisse aus den Tests festzuhalten. Um Entscheidungsprozesse zu dokumentieren, ist ein Diskussionsforum von Vorteil, da dieses automatisch Beratungen über Probleme protokolliert ohne Mehraufwand.

Bei der Visualisierung muss das Tool alle Kriterien erfüllen, die eine gute Visualisierung ausmachen.<sup>7</sup> Weit verzweigte Strukturen benötigen eine Zoomfunktion, wobei dem Benutzer immer klar sein muss, was er sieht und wieso er es dargestellt bekommt.

Um Fremdumgebungen anzubinden, muss das Tool eine Verwaltung über die Aufgaben besitzen, die von Fremdumgebungen abgenommen werden, aber auch Möglichkeiten, die durch Fremdumgebungen hinzu gekommen sind. Hierfür wird eine Schnittstelle zu Fremdumgebungen sowie eine Datenbankverwaltung benötigt, wobei diese auch für Erweiterungen benötigt werden.

Für die Überwachung des Projektes muss es in den Visualisierungen wie dem Projektbaum die Möglichkeit geben, kritische Abschnitte zu erkennen und darzustellen. Optionale Meldungen an Projektleiter oder den Controller sind eine wünschenswerte Möglichkeit, um bei Problemen, die erkannt wurden, gleich Maßnahmen ergreifen zu können, ohne das explizit nachgeschaut werden muss.

---

<sup>7</sup>Vgl. Kapitel 2.4.1

# Kapitel 5

## Konzeption

### 5.1 Ablaufplan und Aufbau von TeamVision

Um den vielen verschiedenen Anforderungen gerecht werden zu können, ist das Tool in verschiedene Komponenten und Module unterteilt, welche sich um die verschiedenen Aufgabenbereiche kümmern sollen. Um besser zu verstehen, welche Inhalte das Tool benötigt, zunächst einmal ein Ablaufplan, der den generellen Ablauf des Tools im Einsatz beschreibt:

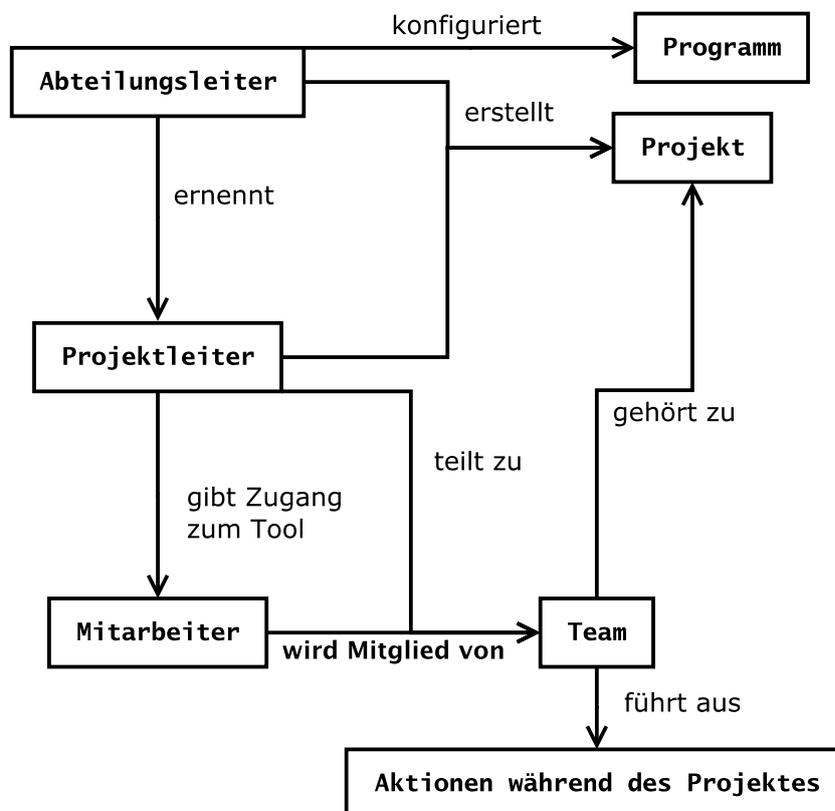


Abbildung 5.1: Der gedachte Ablaufplan des Tools.

Der Ablaufplan beschreibt nicht den internen Zusammenhang des Tools, erklärt aber die wesentlichen Aufgaben. Nicht näher beschrieben ist der Punkt „Aktivitäten während des Projektes“. Dies beinhaltet alle Funktionen des Tools, die im Laufe des Projektes in Anspruch genommen werden. In der Grafik wurden Mitarbeiter aus Übersichtsgründen nicht näher umschrieben, auch aus dem Grund, da es in Kapitel 4<sup>1</sup> schon ein Rollendiagramm gibt, welches die Aktivitäten der einzelnen Projektmitglieder skizziert.

Da für jede Rolle unterschiedlichste Aufgaben anfallen und diese sich teilweise überschneiden können, wird es für jeden Teilbereich eine eigenständige Komponente geben, die in weitere Module unterteilt sein kann. Die folgende Grafik zeigt, welche Komponenten für das Tool geplant sind:

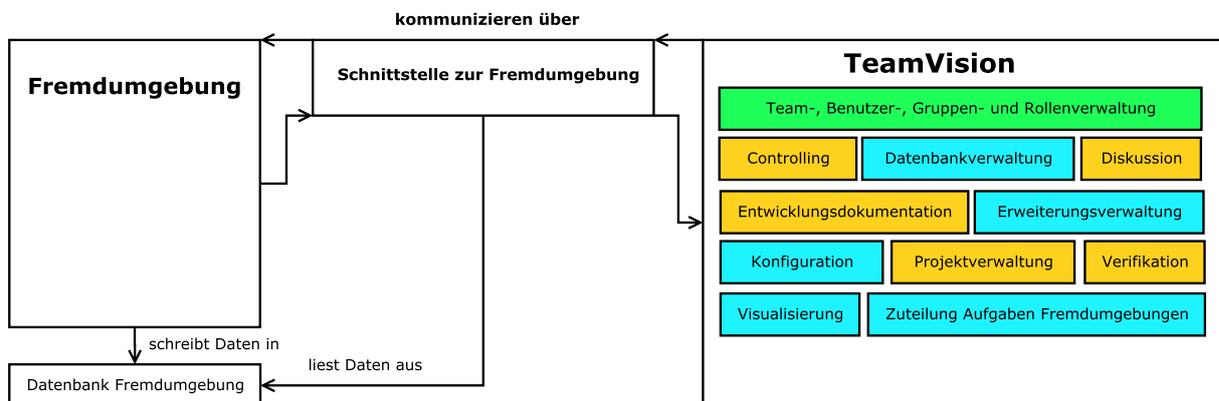


Abbildung 5.2: Aufbau von TeamVision: logische Zusammenhänge zwischen Komponenten sind farblich hervorgehoben.

## 5.2 Aufbau von TeamVision im Detail

### 5.2.1 Projektverwaltung

Die Projektverwaltung ist neben dem Anlegen von neuen Projekten auch für die Verwaltung von existierenden Projekten zuständig. Der Status kann hier editiert werden, indem neue Aufgaben erzeugt, bestehende Aufgaben abgeschlossen oder aus dem Projekt genommen werden, indem sie ausgeschlossen werden. Dies kann der Fall sein, wenn die Spezifikation sich geändert hat. Solche Aufgaben müssen dann in einem Pool verwaltet werden, welcher ausgeschlossene Aufgaben verwaltet. Diese fließen nicht mehr in den Projektstatus ein, es sei denn eine Aufgabe wird reaktiviert. In der Projektverwaltung wird außerdem der geplante zeitliche Ablauf festgehalten in Form von zeitlichen Angaben wie geplante Arbeitsstunden für eine Aufgabe und Fristen für Aufgaben oder ihnen übergeordneten Meilensteinen. Um besonders wichtige Termine oder Meetings dem Team für bessere Planung des Vorgehens sichtbar zu machen, ist ein Wochenkalender angedacht, der für jede Woche anstehende

<sup>1</sup>Vgl. Kapitel 4.6

Termine darstellt. Außerdem können im Kalender neue Termine hinzugefügt werden, die neben dem ganzen Projektteam auch nur eine Gruppe betreffen können. Die Projektverwaltung kümmert sich um das Anlegen der Struktur für den Projektbaum. Dabei werden Daten von Knoten und Kanten mit entsprechenden Metadaten in die Datenbank geschrieben.

Diese Komponente verwaltet weiterhin die Auftragslage des Projektes sowie die Anforderungen des Kunden in Form von Spezifikation, Auftragsbeschreibung und Projektnotizen.

### **5.2.2 Team-, Benutzer-, Gruppen- und Rollenverwaltung**

Aufgrund der Freiheit, dass ein Benutzer in jedem Team und somit in jedem Projekt andere Berechtigungen haben kann, muss TeamVision eine klar strukturierte, leicht veränderbare Rollenverwaltung mitbringen. Dafür soll ein Benutzer eine globale Rolle haben wie z.B. dieser Benutzer ist ein Projektleiter, womit er in TeamVision Projekte anlegen und verwalten kann. Durch andere Projektleiter kann der Benutzer dann in deren Projekte eingeladen werden und dort in verschiedenen Rollen auftreten, vom Projektleiter bis zum Beobachter, einzig die Rolle des Kunden ist nicht möglich. Folglich kann ein Projektleiter, welcher in anderen Projekten keine Rolle hat, diese auch nicht sehen, ausgenommen in der Konfiguration wurde die Voreinstellung abgewandelt. Hat ein Projektleiter eine Beobachterrolle inne, so soll er eventuell mehr sehen können als ein Entwickler, der eine Beobachterrolle im gleichen Projekt hat, um Managementenerfahrung zu sammeln, während der Entwickler bei der Programmierung eventuell eine beratende Rolle inne haben soll.

Einen Sonderfall stellt der Abteilungsleiter dar, da er durch seine Rolle schon global in TeamVision alle Rechte hat. Er kann in einem Projekt keine weitere Rolle annehmen, aber jederzeit eine Statusmeldung zu einem Projekt hinterlassen, um zu kennzeichnen, dass er in einem Projekt auch als Entwickler tätig ist.

Weiterhin hat jeder Benutzer ein Profil, in dem er seine Fähigkeiten eintragen kann. Um die Fähigkeiten von Benutzern vergleichbar zu machen, z.B. um ein Team aus möglichst erfahrenen Programmierern in der benötigten Sprache bilden zu können. Außerdem besitzt jeder Benutzer eine private ToDo-Liste und für jedes Projekt eine öffentliche ToDo Liste. D.h. der Benutzer kann sich eine Gesamtübersicht verschaffen und darüber hinaus zu jedem Projekt sichtbare Angaben machen, was noch zu erledigen ist. Um die Verwaltung der Listen zu erleichtern, soll der Benutzer in einem Menü seine Listeneinträge hinzufügen, ändern oder löschen können und je nach Zuteilung werden diese dann in der entsprechenden Anzeige sichtbar gemacht.

Ein Abteilungs- bzw. Projektleiter kann neben dem erstellen von Projekten Teams- und Projektgruppen bilden, wobei die Gruppen immer Untereinheiten vom Team sind. Eine Gruppe kann im Gegensatz zum Team auch vom Team selbst angelegt werden. Diese Gruppe kann dann auch vom Team editiert werden.

Zu jedem Projekt gehören auch die Möglichkeiten Regeln und Richtlinien vorzugeben, die auch aus alten Projekten übernommen und ggf. abgewandelt werden können.

### 5.2.3 Controlling

Um eine erfolgreiche Überwachung eines Projektes zu gestalten werden Anforderungslisten, Zeitpläne und Deadlines sowie der aktuelle Projekt- und Teamstatus benötigt. Damit können mögliche Probleme wie Überschreitung des Projekttermins, Nichterfüllung der Spezifikation, Personalausfall und Personalmangel rechtzeitig entgegen gewirkt werden.

Dazu soll jedem Projekt eine Spezifikation zugeordnet sein und jeder Abschnitt der Spezifikation muss als offen oder abgeschlossen gekennzeichnet werden können. Der Projektstatus soll unter anderem über den Projektbaum und einige Statusdiagramme dargestellt werden. Ein mögliches Statusdiagramm wäre ein Kreisdiagramm, welches den aktuellen Stand in fertiger und unfertiger Anteil darstellt. Dem gegenübergestellt wird dann abgelaufene Zeit und noch vorhandene Zeit.

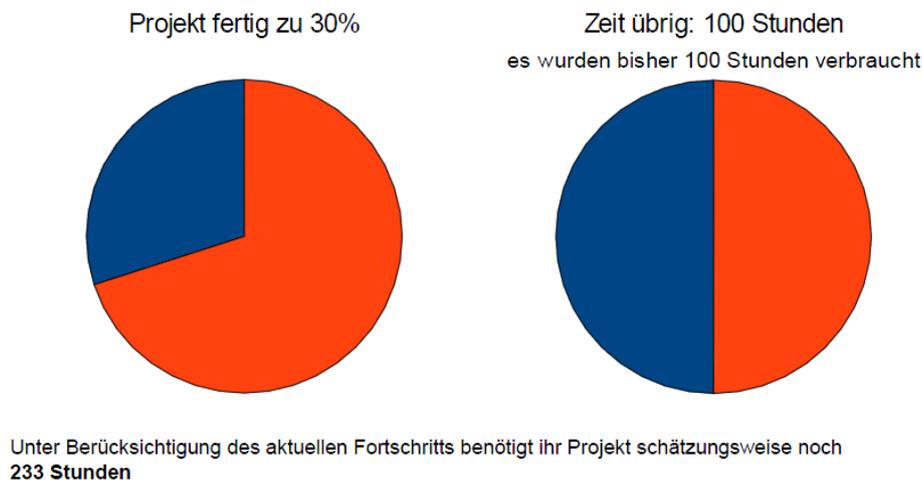


Abbildung 5.3: Der zeitliche Projektstatus mit zwei Kreisdiagrammen visualisiert mit Schätzung der verbleibenden Projektdauer.

Da der zeitliche Ablauf innerhalb eines Projektes auf mehrere Arten gesehen werden kann, soll das Kreisdiagramm zwei Modi unterstützen: Zeit in Arbeitsstunden und kalendarische Zeit. So kann es z.B. sein, dass das Projektteam mehr Arbeitsstunden benötigt, aber kalendarisch gut in der Zeit liegt. Dies könnte ein Indikator dafür sein, dass mehr Budget anfallen könnte. Wenn hingegen die Arbeitsstunden im Rahmen liegen, aber der kalendarische Zeitverlauf problematisch ist, kann dies auf einen Personalmangel hindeuten. Die kalendarische Anzeige soll wahlweise in Tagen, Wochen oder Monaten gestaltet werden können.

Der Projektbaum bietet dem Controller die Möglichkeit zu erkennen, welche Aufgaben des Projektes nicht abgeschlossen sind und wo mehr Zeit benötigt wird als vorgesehen. Dafür soll der Projektbaum eine Controllerdarstellung besitzen, in der nur die benötigte Zeit sowie der Status einer Aufgabe oder eines Astes angezeigt wird. Außerdem sollen besonders kritische Pfade besonders hervorgehoben werden.

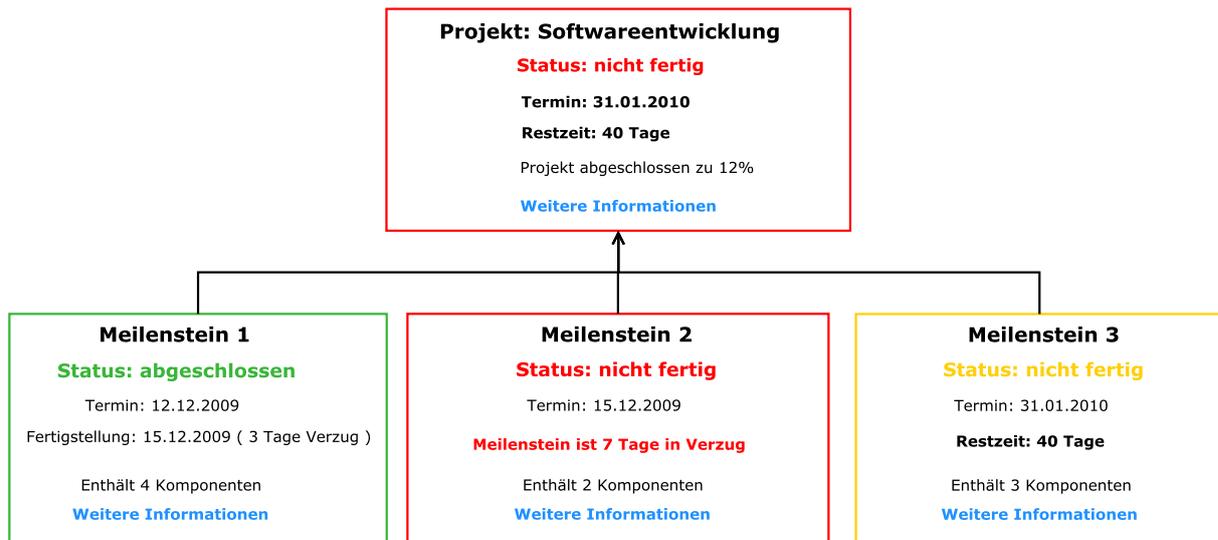


Abbildung 5.4: Projektstatus für den Controller visualisiert. Der zweite Meilenstein befindet sich im Verzug, weshalb das Projektelement als Wurzel des Baums eine rote Umrandung hat und der Status hat eine rote Schriftfarbe.

#### 5.2.4 Diskussion

Die Komponente Diskussion stellt nur die Funktionalität zur Verfügung, eine Diskussion zu erstellen, einen Beitrag zur Diskussion zu liefern oder eine Diskussion als Ersteller oder Moderator zu editieren oder zu schließen. Die Funktionalitäten um die Diskussion bekommen eine eigene Komponente, da keine direkte Gemeinsamkeit zu anderen Funktionen besteht und die Komplexität dieser Komponente stark variieren kann - je nachdem, welche Extras eingebaut werden wie Grafiken einfügen oder Schrift editieren.

#### 5.2.5 Entwicklungsdokumentation

In der Entwicklungsdokumentation sollen Entwickler den Entwicklungsstand und Probleme festhalten. Um den Entwicklungsstand festzuhalten, gibt es einmal die Entwicklungsansicht des Projektbaums, in der Aufgaben als erledigt oder offen deklariert sind und ggf. mit Fristen versehen. Außerdem gibt es einen Modulkatalog, in dem jedes Modul einer Komponente kurz in Form von Eingabe, Ausgabe und Zweck beschrieben ist.

Probleme in der Entwicklung können im Tool selbst in Form von Diskussionen, Bugtracking oder ähnliches festgehalten werden, aber auch in Fremdungumgebungen wie Mantis oder einem Wiki, wenn es eingebunden wurde. Gleiches gilt für Dokumentation und Konzepte, die während der Entwicklung entstehen. Diese kann ebenfalls vom Tool selbst oder von einer Fremdungumgebung verwaltet sein.

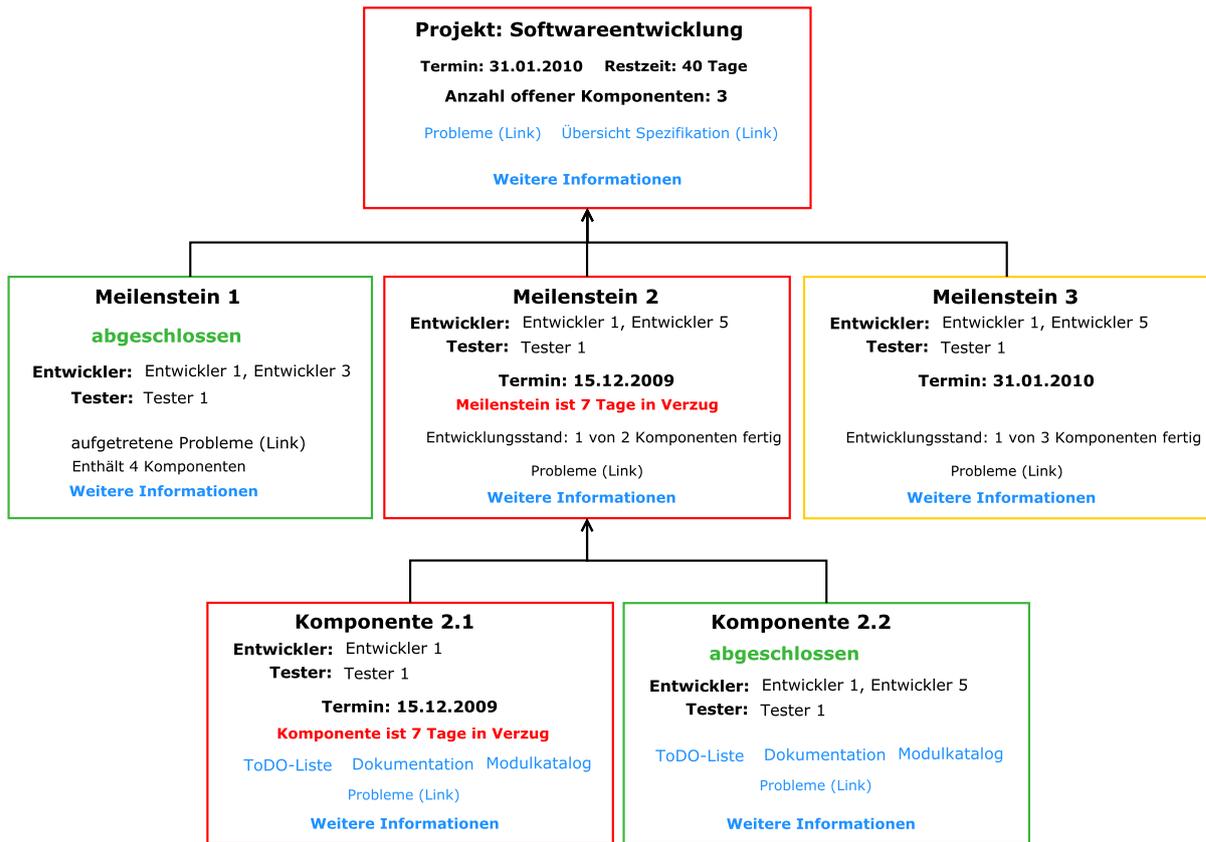


Abbildung 5.5: Projektbaum in der Ansicht für Entwickler. Hier wurde die dritte Ebene für den Meilenstein 2 skizziert, welche auf Wunsch dargestellt werden kann.

## 5.2.6 Verifikation

Für die Verifikation von Projekten stellt TeamVision verschiedene Möglichkeiten bereit, um Testabläufe zu beschleunigen. So können Testpläne erstellt sowie Testmodelle vorgegeben werden. Ein Tester kann anhand von Vorgaben einen oder mehrere Tests durchführen und Ergebnisse dann in Form von Text und Notizen im Projektbaum hinterlegen. Dafür wird es eine Ansicht des Projektbaums für Tester geben. Für die erfolgreiche Durchführung von Tests soll es ein Planungstool in TeamVision geben, welches die Anforderungen an Tests einzelner Aufgaben festlegen kann sowie die Möglichkeit bietet, direkt auf die Erfüllung von Spezifikationen testen zu lassen. Dazu kann beispielsweise gefordert werden, dass während eines Tests bestimmte Funktionen auf ihr Soll- und Istverhalten zu prüfen sind. Bei Nichterfüllung könnten die Ergebnisse des Tests in Fehler und Mängel unterteilt werden, um eine Priorisierung bei der Behebung zu erreichen, wobei ein Mangel eine hohe Priorität hat und Fehler selbst priorisiert werden müssen. Diese Priorisierung wird nicht vom Entwickler, sondern vom Projektleiter oder vom Entwickler selbst vorgenommen, abhängig davon, ob der Entwickler die Rechte zur Priorisierung hat. Eine weitere Genauigkeit der Tests kann dadurch erreicht werden, dass Projektleiter bzw. Entwickler in der Entwicklung zusätzlich Informationen zu Aufgaben hinterlassen können, damit der Tester sich mehr auf kritische Komponenten fixiert, welche eventuell schon in

einem vorherigen Projekt Probleme bereitet haben.

### 5.2.7 Erweiterungsverwaltung

Für eine möglichst hohe Flexibilität im Funktionsumfang müssen neue Komponenten eingebunden werden können. Deshalb soll es eine Komponente geben, in der Erweiterungen registriert werden. Das wäre der Name der Komponente, wo Sie sichtbar sein soll und welche Rolle welche Rechte bekommt. Die Erweiterungen werden über Schablonen erstellt. Zur Kodierung der Rechte ist ein Bitstring von Vorteil, welcher mit 0 bzw. 1 den Zugriffsmöglichkeiten auf die Komponente darstellt. Diesen String soll es dann für jede Rolle geben und die verschiedenen Optionen werden durch die Position dargestellt. Beispiel: Abteilungsleiter="111". Das heißt, dass der Abteilungsleiter vollen Zugriff hat. Er darf lesen, schreiben, löschen. Ein Beobachter könnte dann den String "100" haben und ein Kunde, der die Erweiterung nicht nutzen können soll, hat dann "000". Diese Strings müssten im Quelltext der Erweiterung selbst festgehalten werden. Die Erweiterungsverwaltung kann somit als ein großes Array verstanden werden, welches Erweiterungen registriert. Aus Gründen der Sicherheit wäre eventuell ein Schutzmechanismus von Vorteil, welcher die Komponente validiert. Dieser könnte prüfen, ob die Schablone für die Erweiterung korrekt umgesetzt wird. Die *Schablone* soll wie folgt aussehen:

**Konfiguration** in Form von vordefinierten Variablennamen, für die Werte gesetzt werden.

Eine Funktion `getHTML()`, die den HTML-Text der Komponente als String zurück gibt.

Installationsskript Datenbank für eigene Tabellen, sofern welche benötigt in der Erweiterung: `installDB()`.

**Beispiel einer Erweiterung unter Verwendung von Pseudocode in PHP-Syntax:**

```
1 <?php
2 //dies ist eine Erweiterung, die "Hallo Welt" ausgibt
3 //Berechtigungen
4 $rolle1="110"; $rolle2="101"; ... $rolleN="100";
5
6 function string getHTML(){
7     $htmlcode="<p>\ "Hallo Welt\"</p>";
8     return $htmlcode;
9 }
10 function installDB(){
11 //Pseudocode
12     $sql="Lege Tabelle X an mit Zeilen A,B,C";
13     execute_sql($sql);
14 }
15 ?>
```

### 5.2.8 Komponenten Konfiguration und Datenbankverwaltung

Mit Hilfe dieser Komponente kann der Administrator bzw. der Abteilungsleiter TeamVision den gewünschten Optionen entsprechend anpassen. Dafür wird ein Auswahlformular mit den gewünschten Optionen angezeigt, unter denen man wählen kann. Eine Option wäre die Möglichkeit, Projektleiter

generell zu allen Projekten oder Projekten mit bestimmten Optionen Zutritt zu verschaffen als eine bestimmte Rolle wie z.B. Beobachter.

Die Datenbankverwaltung stellt Funktionen bereit, mit denen die Datenbank eingerichtet und angesprochen werden kann. Durch diese Funktionen soll die Möglichkeit gegeben werden, administrative Operationen direkt von TeamVision aus durchzuführen. Damit besteht auch die Möglichkeit, dass TeamVision beratend in den Administrativen Prozess eingreift.

### **5.2.9 Visualisierung**

Die Visualisierung kümmert sich um grafische Darstellung der Komponenten durch Zeichenfunktionen, Layout Steuerung und Funktionen wie die Zoomfunktion. Außerdem ist die Visualisierung für die Zeichnung der verschiedenen Ansichten des Projektbaums zuständig, welcher für diverse Komponenten unterschiedliche Informationen darstellt.<sup>2</sup> Die Visualisierung ist damit eng mit der Oberfläche von TeamVision, den Funktionen für den Projektbaum sowie weitere Komponenten gebunden und damit eine der wichtigsten Kernelemente von TeamVision.

#### **5.2.9.1 Zoomfunktion**

Wie aus den bisherigen Abbildungen zum Thema Projektbaum bzw. den im Konzept dargestellten Grafiken vielleicht schon ersichtlich geworden ist, besteht eine der größten Schwierigkeiten in TeamVision darin, die großen Datenmengen auf dem Monitor so darzustellen, dass der Betrachter sich innerhalb kürzester Zeit mit dem Projektstatus vertraut machen kann. Deshalb sollen nicht alle Informationen automatisch, sondern erst auf Wunsch, dargestellt werden. Wird eine Ansicht ausgewählt, so werden erst die höheren Ebenen dargestellt. Sobald der Benutzer z.B. mehr Informationen zu einem Meilenstein oder einem anderen Element aus dem Projektbaum erfahren möchte, so muss er diesen auswählen, um sich mehr Informationen anzeigen zu lassen. Gleiches gilt auch für Kindelemente. Deren Details sind ebenfalls erst auf Wunsch sichtbar. Sobald eine andere Komponente gewählt wird, so sollen andere Teilbäume wieder ausgeblendet werden. Obwohl dadurch ein detaillierter Vergleich zwischen zwei oder mehr Teilbäumen ausgeschlossen ist, erscheint es aufgrund der Datenmenge nicht sinnvoll, möglichst viele Daten gleichzeitig zu visualisieren. Dennoch sollte auf Wunsch die Möglichkeit gegeben sein, mehrere Teilbäume gemeinsam zu betrachten. Dafür könnte sich ein Projektbaum eignen, der die Projektstruktur, nicht aber die Details offen legt. Der Status könnte dann nur über Farben visualisiert werden.

Um aus Elementen selbst weitere Informationen zu beziehen, sollen weitere Fenster geöffnet werden, welche diese Informationen enthalten, die dann nach Betrachtung der Information wieder geschlossen werden können.

---

<sup>2</sup>Controllingansicht und Entwickleransicht sind in den zugehörigen Komponenten beschrieben

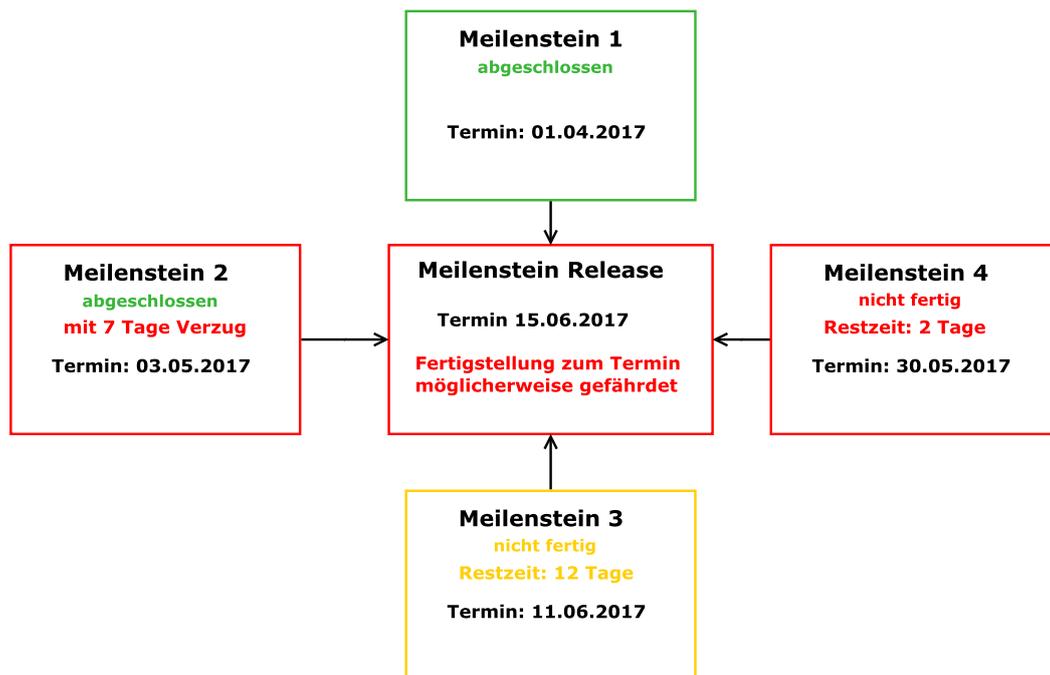


Abbildung 5.6: Abhängigkeiten werden mittels der Zoomfunktionalität dargestellt. In der Grafik ist die Fertigstellung eines Meilensteins durch die Verzögerung anderer Meilensteine in Gefahr.

### 5.2.10 Komponente: „Zuteilung Aufgaben Fremdumgebung“

Wenn Fremdumgebung eingebunden ist, dann werden eventuell einige bereitgestellte Funktionen nicht benötigt oder es kann sein, dass z.B. Daten aus der Fremdumgebung und nicht mehr der eigenen Datenbank geholt werden müssen. Dann ist es nicht mehr Aufgabe der eigenen Datenbankverwaltung, die Daten zu verwalten, sondern die Verwaltung muss über eine Schnittstelle zur Fremdumgebung geschehen. Eventuell muss die Fremdumgebung angepasst werden, damit diese keine Daten korrumpiert, wenn z.B. ein bestimmtes Format festgelegt worden ist.

Weiterhin ist es möglich, dass Aufgaben im Ganzen von einer Fremdumgebung übernommen werden. So kann es sein, dass der Projektbaum im Element einen Link auf eine Auflistung setzt, die in einer Fremdumgebung vorhanden ist wie z.B. eine Auflistung von Bugs für eine Komponente oder eine Aufgabe.

## 5.3 Oberfläche von TeamVision

Die Oberfläche wird in zwei Bereiche unterteilt: der Anzeigebereich und der Navigationsbereich, wobei der Navigationsbereich wiederum unterteilt ist in zwei Unterbereiche. Dadurch soll dem Benutzer die Orientierung in der Navigation erleichtert werden. Der erste Unterbereich des Navigationsbereiches ist das Hauptmenü, welches möglichst unverändert bleiben soll. In diesem Menü befinden sich

Links wie „Übersicht“, „Recherche“ oder „Logout“. Der zweite Bereich beinhaltet Optionen, die sich je nach ausgewählter Funktionalität ändern soll. So kann z.B. ein Projektstatus nur dann abgefragt werden, wenn ein Projekt ausgewählt wurde. Da Anzeigen in diesem Bereich dynamisch sind, ist dieser vom statischen Bereich abgekoppelt. Für den Benutzer wird dieser Bereich als „Aktivitäten“ benannt.

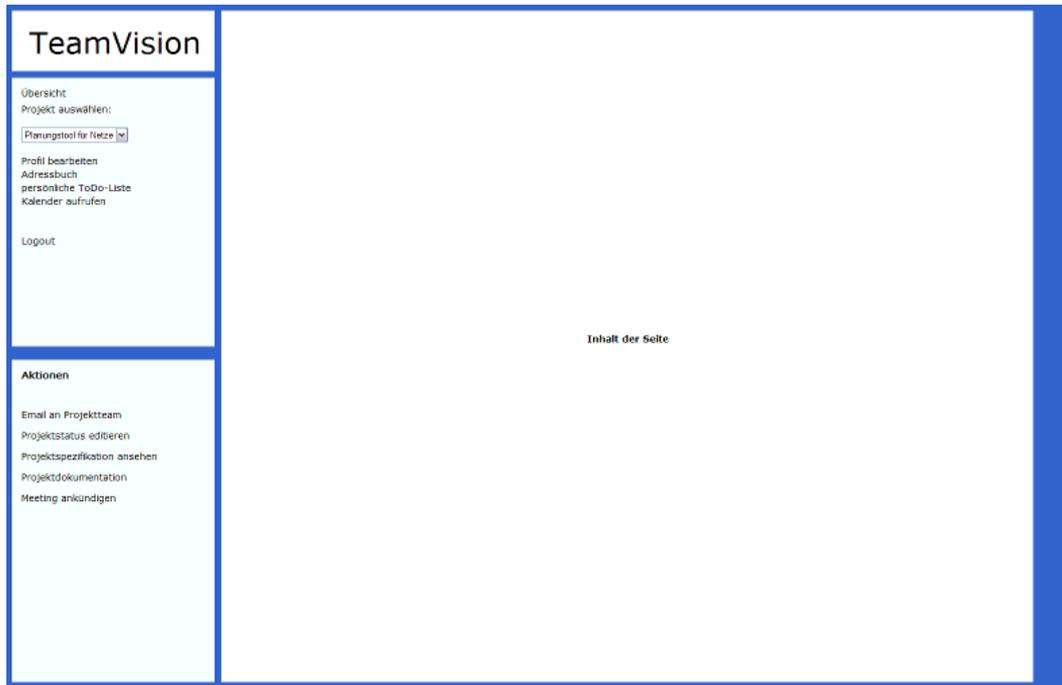


Abbildung 5.7: Ein Entwurf der Oberfläche von TeamVision.<sup>3</sup>

### 5.3.1 Der Projektbaum

Der Projektbaum soll als Übersicht des Projektstatus dienen, damit der Benutzer erkennen kann, wie der Status des Projektes insgesamt ist und in welchem Bereich des Projektes Bedarf an weiterer Planung besteht oder sogar Maßnahmen ergriffen werden müssen, um den Erfolg des Projektes nicht zu gefährden. Um den verschiedenen Rollen gerecht zu werden, soll es nicht nur eine Darstellung des Projektbaums geben, sondern vielmehr soll die Darstellung des Projektbaums auf die Rollen zugeschnitten sein. Im Konzept enthalten sind ein Projektbaum für den Controller, den Entwickler und den Tester sowie ein Projektbaum, der das Projekt und seinen Status möglichst einfach visualisiert und somit einen Überblick verschaffen soll.

Eine Besonderheit des Projektbaums soll seine Flexibilität bzgl. des Aufbaus sein, aber auch die Zusammenführung von verschiedenen Informationen in einem begrenzten Bereich, um langes Suchen zu ersparen und durch Zoomfunktionen benötigte Details abzurufen. Umgekehrt können durch den

<sup>3</sup>Im Anhang befindet sich eine größere Darstellung des Entwurfs.

Projektbaum an entsprechender Stelle Informationen hinzugefügt werden. Ein weiterer Mehrwert gegenüber bestehenden Tools soll dadurch entstehen, dass die Möglichkeit besteht Elemente des Baums untereinander mittels Abhängigkeiten zu verknüpfen und somit die Auswirkung von Abhängigkeiten auf Aufgaben zu visualisieren wie z.B. die mögliche Feststellung „durch diese vielen Abhängigkeiten ist es kaum möglich den Termin rechtzeitig wahrzunehmen, da die Abhängigkeiten selbst erst kurz vor dem Termin fertig sind“. Somit wird die Problematik des Gantt-Diagramms umgangen, dass nur eine begrenzte Anzahl an Abhängigkeiten dargestellt werden kann. Die Anzeige der Abhängigkeiten erfolgt in einem gesonderten Fenster, da der Projektbaum sonst durch die vielen Verbindungen an Übersicht verlieren würde und alle Teilbäume mit Abhängigkeiten dargestellt werden müssten.

Beim Entwurf des Projektbaums hat es sich als effektiv herausgestellt, Farbe des Rahmens eines Elements als Statusmeldung zu nutzen. Würde der gesamte Hintergrund gefärbt, so könnten die Informationen innerhalb des Elementes eventuell je nach Farbe nicht mehr vollständig wahrgenommen werden. Dabei soll der Status eines Elementes von seinen Kindern beeinflusst werden. Dabei entscheidet das Element, welches eine Terminvorgabe am schlechtesten erfüllt. Elemente, welche abgeschlossen sind, werden generell grün dargestellt. Es spielt dabei keine Rolle, ob sie ihren Termin eingehalten haben. Eine mögliche Ausnahme hiervon wäre die Darstellung von Abhängigkeiten, da sich eine Verspätung negativ auswirkt. Elemente, die innerhalb des Zeitplans liegen, werden in einem Gelb dargestellt und Elemente, die sich dem Abschlusstermin nähern oder sogar mit einer Verspätung zu rechnen ist, werden in einem Rotton dargestellt. Diese Farbaufteilung entspricht dem Ampelsystem und dürfte somit einer Mehrheit der Nutzer bekannt sein, womit die Symbolwirkung der Farbe voll ausgeschöpft werden kann.

Eine weitere Funktionalität, die der Projektbaum unterstützen könnte, ist der PDF Export. Es ist denkbar, dass ein Nutzer den aktuellen Stand ausdrucken, verschicken oder mitnehmen möchte. PDF ist dafür besonders gut geeignet, da das PDF-Format weit verbreitet ist und viele kostenlose Programme zur Verfügung stehen, um PDFs darzustellen. Weiterhin unterstützen viele Programmiersprachen, insbesondere auch Websprachen wie PHP den PDF Export.

### 5.3.1.1 Definition des Projektbaums

Der Projektbaum besitzt eine Wurzel, die das Projekt präsentiert. Die Wurzel besitzt 1 bis n Kinder, die die 1 bis n der Wurzel direkt zugeordneten Komponenten oder Meilensteine repräsentieren. Einem Meilenstein sind 1 bis m Komponenten untergeordnet. Jeder Komponente im Projektbaum sind 1 bis k Aufgaben zugeordnet.

Zwischen einzelnen Unterelementen des Projektbaums können Abhängigkeitsbeziehungen bestehen. Dabei kann ein Element von einem anderen oder mehreren Elementen abhängig sein. Abhängigkeiten sind nicht auf Geschwister- oder Elterknoten beschränkt. In der Darstellung der Abhängigkeiten sind direkte Abhängigkeiten zwischen zwei Komponenten in beide Richtungen besonders hervorzuheben, da sie besonders kritisch sind. Dies könnte z.B. der Fall sein, wenn bei zwei Komponenten die gegenseitigen Schnittstellen angepasst werden.

Jeder Knotentyp des Projektbaums enthält neben möglicher Verknüpfungen zu anderen Knoten auch Metadaten und ggf. Links zu Funktionen des Elements.

**Wurzelement:** Dem Wurzelement kann keine Abhängigkeit zugewiesen werden, da die Wurzel das gesamte Projekt darstellt. Das Wurzelement enthält verschiedene Metadaten wie z.B. die Statusanzeige mit Texten wie „Projekt ist noch in Planung“, „nicht fertig“ oder „abgeschlossen“, wobei diese Anzeige nicht in jeder Ansicht erscheint mit Ausnahme von „abgeschlossen“. Die Statusanzeige ist farblich hervorgehoben entsprechend der Ampelfarben. Es ist auch denkbar, die Anzahl der Aufgaben, unterteilt in die noch offen sind sowie eine Anzeige von fertig gestellten Aufgaben als Information anzuhängen, die über eine Zoomfunktion aufgerufen werden kann. Dem Projektleiter ist somit die Möglichkeit geboten z.B. schon abgeschlossene Aufgaben zu betrachten und ggf. wieder zu eröffnen oder abschließende Notizen hinzuzufügen.

**Meilenstein:** Ein Meilenstein fasst im Idealfall mehrere Komponenten zusammen, die gemeinsam entwickelt werden müssen. In der Softwareentwicklung ist es oft der Fall, dass erst bei Fertigstellung mehrerer Komponenten eine Nutzung oder ein genauer Test möglich ist. Einem Meilenstein sind Informationen über den Status des Meilensteins sowie eine Anzeige der aktuell kritischen Komponente bzw. Aufgabe zugeordnet, die eine Verzögerung wahrscheinlich machen könnte. Dies kann nur aufgrund von Schätzfunktionen geschehen, da aus dem aktuellen Status keine Angabe über die zukünftige Entwicklung gegeben werden kann. Eine mögliche Funktion wäre Bearbeitungsstand in Prozent verrechnet in vergangene Zeit in Prozent. Wenn dabei die Abweichung größer als ein Schwellwert ist, so kann die Komponente bzw. Aufgabe als kritisch eingestuft werden, abhängig von der Komplexität des Elementes. Die Abschätzung der Abweichung soll zusammen mit einer Ansicht nach der Grafik aus Kapitel 5.2.3 in der Zoomfunktion zu finden sein.

**Komponente:** Komponenten repräsentieren übergeordnete Teilaufgaben aus dem Projekt, die mehrere disjunkte Aufgaben beinhalten. Eine Komponente muss mindestens eine Aufgabe beinhalten. Als Metadaten sind der Komponente Status, Anzahl der Aufgaben, „kritische Aufgabe“ sowie der Fertigstellungstermin zugeordnet.

**Aufgabe:** Eine Aufgabe ist eine in sich geschlossene, sozusagen atomare Aktion im Projekt. Sie besitzt zwei Zustände: beendet oder nicht beendet. Für den Fall, dass eine Aufgabe erst fertig gestellt werden kann, wenn eine andere Aufgabe abgeschlossen ist, so kann eine Aufgabe mit einer anderen Aufgabe verknüpft werden. Metadaten einer Aufgabe sind Zeitraum, zuständige Entwickler, mögliche Testpersonen, sowie eine Dokumentation der Aufgabe und ggf. zugehörige Diskussionen.

### 5.3.2 Hilfen für Menschen mit Farbwahrnehmungsstörungen

Aufgrund der Vielzahl von unterschiedlichen Störungen in der Farbwahrnehmung kann kein allgemeingültiges Design gefunden werden, welches allen Anwendern gerecht wird. Beachtet man dann auch noch den geringen Prozentsatz von Nutzern mit Problemen in der Farbwahrnehmung, so ergibt sich, dass auf eine volle Ausschöpfung von Farbeindrücken nicht verzichtet werden sollte. Um aber dennoch eine Lösung bereitzustellen, welche jedem Nutzer gerecht wird, wird dem Nutzer eine Möglichkeit gegeben, seine Farben selbst zu wählen, wenn er im Benutzerprofil angibt, Probleme mit Farbwahrnehmung zu haben. Diese Angabe ist für andere Benutzer nicht sichtbar.

	Rot	Grün	Blau
Linkfarbe	 <input type="text" value="FF"/>	<input type="text" value="00"/>	<input type="text" value="00"/>
Schriftfarbe	 <input type="text" value="00"/>	<input type="text" value="FF"/>	<input type="text" value="00"/>
Menüfarbe	 <input type="text" value="00"/>	<input type="text" value="00"/>	<input type="text" value="FF"/>

Abbildung 5.8: Mit einem einfachen RGB Menü kann der Benutzer das Ergebnis seiner Farbwahl sofort erkennen.

Da es sein kann, dass ein Nutzer Farben gar nicht wahrnimmt, ist es angedacht, dass in jeder Visualisierung Informationen durch Text oder durch Symbole in ihrer Bedeutungen zusätzlich hervorgehoben werden, wenn dies auch nicht so effektiv wie durch Farbgebung geschehen kann.

## 5.4 Aufbau der Datenbank

TeamVision soll durch eine relationale Datenbank ergänzt werden, welche Benutzereingaben sowie Einstellungen persistent macht. Erweiterungen soll die Möglichkeit geboten werden, weitere Relationen anzulegen, die unabhängig von den Relationen agieren, welche in diesem Konzept beschrieben werden.

Aufgrund der Vielzahl der Komponenten von TeamVision dürfte die Anzahl der Relationen den Rahmen dieser Arbeit übersteigen. Deshalb werden hier nur einige der benötigten Relationen aufgelistet:

**Users:** Relation, in der Benutzerdaten wie Login, die globale Rolle in TeamVision und personenbezogene Daten gespeichert werden

**Rollen / RolleAktion:** In diesen Relationen werden die Rechte gespeichert, die eine Rolle bzgl. der verschiedenen Interaktionsmöglichkeiten besitzt. Dazu wird eine Rolle mit einer Aktion verknüpft, um die Relation möglichst flexibel für jede neue Aufgabe zu halten.

**Projekt:** Mit dieser Relation soll das Projekt repräsentiert werden, wer es leitet und welchen Status es hat.

**Team:** In dieser Relation wird festgehalten, welche Benutzer von TeamVision in einem Team an einem Projekt arbeiten.

**Gruppe:** In dieser Relation werden Gruppen, die innerhalb eines Projektes gebildet werden festgehalten. Hierbei ist zu beachten, dass eine Gruppe sich in der Konfiguration unterschiedlich verhält. Dies hängt davon ab, ob eine Gruppe von einem Projektleiter oder beispielsweise von einem Entwickler erstellt worden ist.

**Projektbaum (mit weiteren Relationen):** Für den Projektbaum werden diverse Relationen benötigt, um die verschiedenen Arten von Elementen, Abhängigkeiten, Notizen, Details, etc. zu speichern.

## 5.5 Weitere mögliche Funktionen

Im bisher vorgestellten Konzept wurde nur ein Bruchteil der Ideen eingearbeitet, die für die Erstellung des Konzepts vorgelegen haben. Deshalb sollen drei Ideen erwähnt werden, welche TeamVision durch eine Implementierung bereichern würden. Dabei handelt es sich um eine Recherchefunktion, eine Funktion um Nachrichten an Teammitglieder, Gruppen oder das ganze Team zu versenden und eine Newsfunktion.

Die Recherchefunktion soll dazu dienen, um Projekte nach bestimmten Begrifflichkeiten zu durchsuchen, um für ein aktuelles Projekt schon auf frühere Erfahrungen zurückgreifen zu können. Dafür könnten Aufgaben, Komponenten, die Meilensteine und das Projekt selbst mit Suchworten belegt werden. So könnte anhand einer Eingabe die Suchworte mittels einer Datenbank verglichen werden und ähnliche bzw. genaue Treffer würden dann in absteigender Reihenfolge aufgelistet werden. Je nachdem wie genau bzw. wie flexibel die Suchfunktion ist, könnte sie bei einer größeren Datenbank dazu beitragen, Probleme, die eventuell schon in früheren Projekten aufgetreten sind, schneller zu beheben. Die Nachrichtenfunktion dient dazu, z.B. als Controller bei der Feststellung von Problemen z.B. direkt mit dem Entwickler über TeamVision in Kontakt zu treten oder eine Notiz für ihn zu hinterlassen, die auf dem Startbildschirm des betreffenden Benutzers beim nächsten Login eingeblendet wird.

Die Newsfunktion soll als automatischer Überblick über aktuelle Geschehnisse dienen. Wird beispielsweise eine Aufgabe abgeschlossen, so könnte in den News eine Zeile erscheinen „Die Aufgabe 123 wurde am 01.12.2009 um 14:35 abgeschlossen (1 Tag vor dem Termin)“.

## 5.6 Fallbeispiel für eine Fremdkomponente: das MediaWiki

Durch das Einbinden einer Fremdkomponente können verschiedene Aufgaben von TeamVision an diese abgegeben werden. Im Falle des MediaWiki wäre dies z.B. die Funktionalität, Dokumentation zu Aufgaben zu erzeugen. TeamVision könnte dem MediaWiki über dessen API mitteilen, einen neuen Artikel mit einer Schablone für den Inhalt anzulegen. Es besteht auch die Möglichkeit über TeamVision den Artikel zu verändern und dann über das MediaWiki abzuspeichern. Somit besteht die Möglichkeit einen leistungsfähigeren Editor einzubinden, welcher mehr Funktionen besitzt als der Editor im MediaWiki. Als Bedienungen muss eine Konvention eingeführt werden, wie die Arti-

kel zu benennen sind. Das MediaWiki besitzt in diesem Falle schon Regeln, indem Kategorien bzw. Namen durch einen Doppelpunkt voneinander getrennt werden, was dann wie folgt aussieht: Projekt:Meilenstein:Komponente:Aufgabe. Weiterhin zu beachten ist dann, dass durch die Änderung des Artikelnamens auch TeamVision beeinflusst wäre oder aber der Name des Artikels nicht geändert werden darf. Die zweite Option erscheint sinnvoller, da ansonsten ein hoher Entwicklungsaufwand in TeamVision betrieben werden muss, sofern ein Artikel umbenannt wird, da dieser erst gefunden und neu zugeordnet werden muss.

Weiterhin könnten im MediaWiki die Teams eingeteilt werden, indem der Benutzername mit Rolle in einem Artikel zum Projekt eingetragen wird. Dies dürfte dann aber auch nur durch Personen mit Berechtigungen wie Abteilungsleiter oder Projektleiter geschehen, um Missbrauch vorzubeugen. MediaWiki ist allerdings darauf ausgelegt, dass möglichst alle Nutzer einen Artikel verändern können. Der Artikel müsste gesperrt werden, was zur Folge hat, dass sensible Daten dem Team im Wiki verborgen oder eventuell unbefugten Benutzern des Wikis zugänglich sind.

Um die Anbindung zwischen TeamVision und einem MediaWiki zu realisieren, müsste die Schnittstelle eine Kommunikation zur API des MediaWiki aufbauen sowie das MediaWiki bei Veränderung eines Artikels, welcher Informationen für TeamVision speichert, ein Event werfen, welches Prozeduren einleitet, um TeamVision zu aktualisieren. Für Dokumentationsaufgaben allein wäre dies unkritisch, sofern der Artikelname nicht geändert werden kann, weil damit diese Rückkopplung vom MediaWiki zu TeamVision entfällt.

## 5.7 Zusammenfassung des Konzepts

TeamVision besitzt eigene Werkzeuge, die dabei helfen sollen, die Arbeit zwischen verschiedenen Rollen möglichst weit zu verknüpfen, wobei die Hauptaufgabe in der Verknüpfung zwischen Projektleitung, Entwicklung, Verifikation und Controlling liegt. Des Weiteren ist TeamVision dazu gedacht an Fremdumgebungen anzuknüpfen oder durch eigene Erweiterungen ergänzt zu werden. Die folgende Grafik soll zusammenfassend darstellen, welche Grundfunktionen zu jedem Modul gedacht sind:

Team-, Benutzer-, Gruppen- und Rollenverwaltung	Benutzer anlegen, administrieren, Teams bilden, Team verwalten, Rollen erstellen, Rollen zuteilen, Gruppen erstellen, Gruppe verwalten, Benutzerprofil anlegen & verwalten, ToDo-Listen
Controlling	Ansichten: Zeitpläne, Arbeitsfortschritt, Spezifikation, Projektstatus kontrollieren und Probleme lokalisieren
Diskussion	Diskussionsforum, Moderation der Diskussionen, Dateianhang
Entwicklungsdokumentation	Entwicklungsfortschritt sowie aufgetretene und aktuelle Probleme festhalten, Dokumentation in Form von Artikeln, Dokumenten, Konzepten und dem Modulkatalog
Projektverwaltung	Projekte anlegen und managen, Projektbaum als Projektübersicht*, Bestandteile des Projekts festlegen, zeitlicher Ablauf, Anforderungen, Team*, Kalender / Termine
Verifikation	Testpläne (Vorgehen, Vorgaben, Ziele), Fehler- und Mangelliste sowie Ist-Verhalten in Form von Testberichten
Datenbankverwaltung	Daten in Datenbank übertragen, Recherche durchführen, Installation und Löschen von Tabellen
Erweiterungsverwaltung	Erweiterung installieren, zugänglich machen, deinstallieren
Konfiguration	Einstellungen und Verhalten von TeamVision anpassen
Visualisierung	Zeichenfunktionen für Darstellungen von Projektstauts in Diagrammen, Layoutverwaltung, Darstellung des Projektbaums
Zuteilung Aufgaben Fremdumgebungen	Aufgaben an Fremdumgebung abgeben oder neue Aufgaben aus Fremdumgebung in Bestand übernehmen, Schnittstellenverwaltung

Abbildung 5.9: Die Grafik stellt die vorgestellten Kernmodule von TeamVision sowie die angedachten Grundfunktionen dar.

# Kapitel 6

## Implementierung

### 6.1 Umfang der Implementierung

Aufgrund des zeitlich begrenzten Umfangs der Arbeit kann nicht das gesamte Konzept umgesetzt werden. Vielmehr wird sich die Implementierung im Rahmen dieser Arbeit in Form eines Prototyps auf wesentliche Aspekte des Konzepts beschränken. Dieses beinhaltet den Projektbaum, da dieser für alle Rollen ein umfassendes Werkzeug darstellt in TeamVision. Weiterhin ist eine einfache Anbindung an eine Fremdunggebung Bestandteil des Prototyps. Es wurde eine Anbindung an ein MediaWiki durchgeführt, um Artikel zu Elementen eines Projektbaums zu erzeugen und Text anzugeben. Dadurch soll demonstriert werden, dass Aufgaben auch durch bestehende Umgebungen übernommen werden können, sofern eine entsprechende Einbindung in TeamVision geschieht.

### 6.2 Wahl der Entwicklungsumgebung und Sprachen

Da TeamVision als Webapplikation gedacht ist, wird HTML, JavaScript und PHP eingesetzt. PHP eignet sich aus mehreren Gründen für die Entwicklung von TeamVision. Einerseits bietet PHP die Möglichkeit, dynamische Inhalte zu erzeugen und anzuzeigen, andererseits wird PHP fortlaufend entwickelt und insbesondere im Bezug auf Sicherheit und Laufzeit aktualisiert. So wird es in PHP 6 viele der in PHP 4 und teilweise auch in 5 bestehenden Probleme nicht mehr geben, da entsprechende Fehlerquellen in Form von global verfügbaren Arrays<sup>1</sup> oder automatischen Filterfunktionen nicht mehr genutzt werden können. Mit Ajax können Webseiten im Verhalten ähnlich zu Desktopapplikationen gestaltet werden. Damit kann TeamVision direkt auf Benutzereingaben reagieren und den Seiteninhalt während der Eingabe anpassen.

Die Möglichkeit in Pearl zu programmieren wurde verworfen, da im Pearl-Projekt momentan Probleme mit dem Release einer neuen Version bestehen<sup>2</sup> und es unklar ist, welche Änderungen sich in der

---

<sup>1</sup>Gemeint sind die so genannten „register globals“, vor dessen Nutzung schon durch den Entwicklern von PHP, Rasmus Lerdorf, in seinem Buch „Programmieren mit PHP“ abgeraten wird. [Ras03] In der Version 6 werden sie endgültig nicht mehr nutzbar sein.

<sup>2</sup>Die Version 6 wurde laut Susanne Schmidt schon 2000 angekündigt, aber es scheint unwahrscheinlich, dass vor 2010 ein Release zu erwarten ist.[Sch09]

nächsten Version durchsetzen können. In PHP können die meisten Änderungen bezogen auf PHP 6 schon berücksichtigt werden, damit für eine Aktualisierung auf eine modernere Version später wenig bis gar kein Aufwand besteht.

### 6.3 Eingrenzung des Rollenmodells

Bei der Implementierung wurde die Umgebung für den Abteilungsleiter bzw. Projektleiter entwickelt, da dieser die umfassendsten Werkzeuge benötigt. Das Rollenmodell wurde nicht in der Anforderung bzw. im Konzept vorgestellten Form umgesetzt, sondern auf die Beschränkung von Interaktionsmöglichkeiten und Editierfähigkeiten beschränkt. So kann ein Abteilungsleiter jedes Projekt bearbeiten und in ihm mitwirken, während ein Nutzer mit einer anderen Rolle nur bei Projekten mitwirken kann, in deren Team er Mitglied ist.

Weiterhin gilt, dass jeder Benutzer, sofern er kein Abteilungsleiter ist, zwei Rollen in TeamVision einnehmen kann. Zum einen eine globale und zum anderen eine an das Projekt gebundene Rolle. In einem Projekt greift die projektgebundene Rolle, die von einem Abteilungs- oder Projektleiter geändert werden kann. Für allgemeine Aktionen in TeamVision ist die globale Rolle zuständig, welche in der Datenbank direkt in der Nutzertabelle gespeichert wird. Damit kann ein Nutzer die Berechtigung erhalten, ein neues Projekt zu erstellen. Weiterhin gilt, dass aufgrund des eingeschränkten Umfangs des Prototyps die projektgebundene Rolle hauptsächlich dazu genutzt wird, um Personen Zuständigkeiten im Projekt zuzuteilen.

### 6.4 Struktur der Implementierung

Im Gegensatz zum Konzept ist es in der Implementierung nicht ohne weiteres möglich gewesen, für jeden Aufgabenbereich ein Modul zu erstellen, ohne Interaktionen zwischen Modulen aufzuteilen oder die Grenzen zwischen Zuständigkeiten wegzulassen. Bei der Implementierung mussten ebenfalls umgebungsspezifische Einflüsse beachtet werden. Die Benutzungsschnittstelle von TeamVision wird über die Datei `index.php` auf der höchsten Ebene aufgerufen. Die `index.php` ruft dann entsprechend des Parameters aus der URL wie `page` und `form` die gewünschte Seite auf. Der Inhalt der einzelnen Ordner:

**Config:** Konfigurationsdateien, beispielsweise für TeamVision und das MediaWiki

**CSS:** Stylevorgaben für Oberfläche von TeamVision mittels CSS-Dateien

**Extension:** Erweiterungen, die in TeamVision eingebunden werden sollen, finden hier ihren Platz

**Functions:** Module, die importiert werden können, um bestimmte Funktionen nutzen zu können

**Icons:** Bilder, die in TeamVision genutzt werden

**Install:** Installationsscript für die Datenbank, ReadMe zur Installation

**JavaScript:** ausgelagerter JavaScriptcode

**MediaWiki:** Schnittstelle zur Fremdunggebung MediaWiki

**pages:** in diesem Ordner befinden sich die Module, die die Grundfunktionalitäten von TeamVision

repräsentieren

## 6.5 Anmeldevorgang und Login

Der Anmeldevorgang erfolgt über ein Registrierungsformular, in dem der Nutzer Login, Passwort, Emailadresse und Adressdaten eingeben kann. Es ist angedacht gewesen, eine Aktivierung über Email oder den Administrator möglich zu machen. Dafür wurde im Code eine Weiche für eine Aktivierungsoption eingebaut. Zu dieser Funktionalität, die in der Konfiguration aktiviert werden kann, gibt es jedoch kein Modul, welches den Aktivierungsprozess durchführt.

Der Aktivierungsprozess könnte über Email erfolgen oder vom Administrator in Form des Abteilungsleiters, welcher einzelne Benutzer jederzeit aktivieren bzw. deaktivieren kann. Bei der Emailaktivierung müsste in der Datenbank eine Tabelle angelegt werden, welche einen Schlüssel in Form eines Hash speichert, der aus der Email übermittelt werden würde.

Da TeamVision den Zugang zu teils sehr vertraulichen Projektdaten gewährt, kann eine Mindestgröße für Passworte verlangt werden. Außerdem kann der Hashalgorithmus jederzeit in der Konfiguration verändert werden. Voreingestellt ist die Mindestlänge von 8 Zeichen für Passworte und der Hashalgorithmus „sha512“, welches mit einer 512 BIT Verschlüsselung als recht sicher gilt. Da die Hashfunktion in PHP den Algorithmus dynamisch nach dem Parameter wählt, kann die Hashfunktion jederzeit ersetzt werden. Einzig die Passworte müssten dann neu in die Datenbank eingetragen werden, wenn dies in einer bereits bestehenden Installation von TeamVision geschieht.

Beim Login wird der Hash der Eingabe mit dem Hash aus der Datenbank verglichen und anschließend eine Session gestartet, in der Benutzername, Rolle und der Status als eingeloggt festgehalten werden. Der Status hilft dabei, den Zugang zu allgemeinen Modulen zu regeln. Die Rolle ist für Weichen in Modulen gedacht, wo die Möglichkeiten durch die Rolle begrenzt sind. Der Benutzername wird für die Abfragen aus der Datenbank benötigt.

Neu registrierte Benutzer haben die Rolle eines Gastes.

## 6.6 Oberfläche

Gemäß dem Entwurf aus dem Konzept wurde die Oberfläche von TeamVision in drei Bereiche unterteilt: den Anzeigebereich, den statischen Navigations- und den dynamischen Aktionsbereich.

Die Oberfläche wird von der `index.php` erzeugt, indem drei DIVs<sup>3</sup> verwendet werden, in denen der gewünschte Inhalt angezeigt wird. Damit der Inhalt aus dem Anzeigebereich nicht über das DIV hinaus gerendert wird, ist das DIV für den Anzeigebereich mit dem Befehl „`overflow: auto`“ versehen. Somit wird bei einem Überlauf des anzuzeigenden Inhalts aus dem DIV in der Richtung des Überlaufs ein Scrollbalken eingeblendet. Durch die Option `auto` wird der Scrollbalken nicht eingeblendet, wenn er nicht benötigt wird. Dadurch wird unnötige Irritation und Ablenkung vermieden. Die Option des

<sup>3</sup>Das Tag DIV definiert in HTML einen Bereich. Beispiel: `<div> Dies ist ein Bereich </div>`

Überlaufs wird insbesondere für die Darstellung des Projektbaums benötigt.

Der Navigationsbereich ist für alle eingeloggtten Benutzer gleich bis auf den Inhalt der Auswahlbox für Projekte. Dort werden nur Projekte gelistet, zu denen der Benutzer Zugang erhalten hat. Weiterhin im Navigationsbereich verlinkt sind das Adressbuch, das Profil, die ToDo-Liste und die Kalenderfunktionalität.

Im Aktionsbereich sind für einen Abteilungsleiter momentan unter anderen das Erstellen eines neuen Projekts und im Projekt selbst die Erstellung sowie Verwaltung eines Teams vorgesehen.

The screenshot shows the TeamVision interface. On the left is a navigation menu with the title 'TeamVision'. Below the title is a dropdown menu labeled 'Projekt auswählen:'. Below that are links for 'Profil bearbeiten', 'Adressbuch', 'ToDo-Listen', and 'Kalender aufrufen'. At the bottom of the menu is a 'Logout' link. Below the menu is an 'Aktionen' section with links for 'Neues Projekt anlegen' and 'Mitarbeiter editieren'.

On the right is a calendar for September 2009. The calendar shows the days of the week (Montag, Dienstag, Mittwoch) and the dates from 08 to 20. There are 'Termin' (Meeting) entries on the 11th, 18th, and 20th. Below the calendar is a form for entering a meeting for the date 21.09.2009. The form includes fields for 'Beginn:' (11:00) and 'Ende:' (12:00), a checkbox for 'Privater Termin?', and a text field for 'Termin (kurz):'.

Abbildung 6.1: Ausschnitt aus der Oberfläche von TeamVision

## 6.7 Grundfunktionen

### 6.7.1 Profil

Im Profil kann der Benutzer seine globale Rolle einsehen, seine Adressdaten aktualisieren und sein Passwort ändern. Außerdem ist es im Profil möglich, Farbeinstellungen für den Projektbaum vorzunehmen, die auf die Oberfläche ausgedehnt werden könnten. Die Farbeinstellungen werden in der Datenbank hinterlegt und in der aktuellen Session vermerkt. In die einzelnen Felder des Formulars für Farbanpassungen können nur gültige HEX-Ziffern eingegeben werden, also 0-9 und A-F (groß und klein). Wird nur eine Ziffer eingegeben, so wird eine 0 angehängt. Nach der Änderung eines Farbkanals ändert sich auch die Vorschau für die aktuell gewählte Farbe. Dafür wird eine JavaScript-Funktion

aufgerufen, die nach einer Änderung an einem Farbkanal den Wert für die Vorschau, dargestellt durch ein span im HTML-Quelltext.

		R-Kanal	G-Kanal	B-Kanal	
Projektbaum: Element in Planung		<input type="text" value="AA"/>	<input type="text" value="AA"/>	<input type="text" value="AA"/>	Standardwert: AA AA AA
Projektbaum: Element aktiv		<input type="text" value="FF"/>	<input type="text" value="CE"/>	<input type="text" value="07"/>	Standardwert: FF CE 07
Projektbaum: Element in Verzug		<input type="text" value="FF"/>	<input type="text" value="00"/>	<input type="text" value="00"/>	Standardwert: FF 00 00
Projektbaum: Element abgeschlossen		<input type="text" value="38"/>	<input type="text" value="B5"/>	<input type="text" value="38"/>	Standardwert: 38 B5 38
Projektbaum: Element abgebrochen		<input type="text" value="00"/>	<input type="text" value="00"/>	<input type="text" value="00"/>	Standardwert: 00 00 00

Abbildung 6.2: Farbauswahl für die Statusanzeige des Projektbaums.

## 6.7.2 Adressbuch

Das Adressbuch besteht aus mehreren Adressbüchern, die nach Projekten unterteilt sind. Zusätzlich gibt es eine Gesamtansicht, in der alle Kontakte aus den verschiedenen Projekten gelistet sind. Im zugeklappten Zustand sieht der Benutzer von einem Adressbuch nur den Namen des Adressbuchs in Form des Projektnamens sowie ein Icon in Form eines Pfeils und eine Trennlinie. Klickt der Benutzer auf das Icon, so wird dieses um 90° rotiert und das Adressbuch ausgeklappt. Der Inhalt des jeweiligen Adressbuchs wird mit Ajax mittels einer HTTP-Request<sup>4</sup> ermittelt. Dabei wurde im Adressbuch ein Verfahren zur Steigerung der Sicherheit in der Datenübermittlung erprobt, welche es Angreifern schwerer machen soll, über den selben Weg Daten zu erlangen. Aufgrund der Tatsache, dass ein HTTP-Request leicht gefälscht werden oder durch Skripte beispielsweise in PHP selbst emuliert werden kann, wurde ein Konzept erarbeitet, um die Sicherheit in der Übermittlung sensibler Daten zu steigern.

**Datenübermittlung im Adressbuch sicherer gestalten:** Um unnötigen Traffic und Programmieraufwand zu sparen, wurden HTTP-Requests verwendet um Adressdaten für die Anzeige zu ermitteln statt die Seite immer wieder neu zu laden. Da dabei Parameter zum Funktionsaufruf offen verfügbar sind und die Skripte, die aufgerufen werden keine Sessiondaten nutzen können, da der Header der Seite nicht nachträglich geändert werden kann, wurde ein Schlüssel eingeführt.

Der Schlüssel, welcher in der Konfiguration eingestellt werden kann, wird mit dem Benutzernamen als Hash in der JavaScript-Funktion übermittelt. Der gesendete Hash wird mit einem Hash verglichen, der aus dem Schlüssel und dem übermittelten Benutzernamen gebildet wird. Die Chance, dass beide Eingaben passend gefälscht sind, ist gering, da dazu dem Angreifer der völlig willkürlich definierte Schlüssel bekannt sein muss, der dem Angreifer auch aus dem Quelltext der Website nicht ersichtlich

<sup>4</sup>Ein HTTP-Request ist eine Anfrage an den Server mittels HTTP. Die Antwort wird von Ajax weiter verarbeitet.

wird, da er dort nicht auftaucht.

Sind dem Angreifer Username und Hash bekannt, dann hat dieser mit hoher Wahrscheinlichkeit schon Zugriff auf die Nutzerdaten mittels einer Schadsoftware gehabt. Das vorgestellte Verfahren soll nur davor schützen, dass kein Nutzer mit höherer Berechtigung gekapert wird, indem einfach der Parameter für den Nutzernamen geändert wird.

Eine alternative Lösung über Sessionids wurde nicht angestrebt, da hier andere Gefahren wie Hijacking<sup>5</sup> bestehen. Probleme bei der Entwicklung des Kalenders (siehe 6.7.4) haben gezeigt, dass bei der Verwendung von HTTP-Requests Vorsicht geboten ist. Die sicherste Lösung ist es, die Interaktionen in einen Parameter, der über GET oder POST übermittelt wird, einzubauen und damit ständiges neu laden der Seite zu provozieren unter Verzicht auf Ajax.

Eine weitere Idee die Sicherheit zu erhöhen ist es, einen temporären Parameter zu erstellen, der als Hash nur eine bestimmte Zeit eine Gültigkeit besitzt. Dafür müsste TeamVision jedoch während der Session diese in der Datenbank festhalten. Die Nutzung von Ajax wäre dann wesentlich sicherer, da ein Einbruchversuch durch Hijacking nur während einer gültigen Session von Erfolg gekrönt wäre, sofern nicht eine Kontrolle mittels der übermittelten IP-Adresse aktiviert ist, was allerdings nicht in allen Netzen funktioniert.

#### ▶ Gesamtansicht

---

#### ▼ Computergrafik

<b>Name Nachname</b>	069-555-44-12	e@mail.de
	Adresse:	Straße Nr. 10, 6000 Frankfurt
<b>Veith David</b>	069-555-444-1	e@mail.de
	Adresse:	Straße Nr. 1 a, 60000 Frankfurt

---

#### ▶ Datenstrukturen

---

Abbildung 6.3: Dem Benutzer steht für jedes Projekt ein zusätzliches, automatisch generiertes Adressbuch zur Verfügung.

### 6.7.3 ToDo-Listen

Angelehnt an die Idee des Adressbuchs, Listen für die einzelnen Projekte erst auf Wunsch angezeigt zu bekommen, werden die einzelnen ToDo-Listen für die Projekte ebenfalls erst durch einen Klick auf ein Icon in Form eines Pfeils angezeigt. Es wird unterschieden zwischen der persönlichen ToDo-Liste und den einzelnen Listen für die Projekte. Eine aufgeklappte ToDo-Liste präsentiert sich in Form einer Tabelle mit drei Spalten und einem Button unter der Liste. In der Tabelle werden zeilenweise die Elemente in Form einer Notiz, des Fälligkeitsdatum sowie eines Löschbuttons gelistet. Der Button

<sup>5</sup>Diebstahl von Identitäten im Internet durch das Benutzen manipulierter Adresseingaben, in denen Parameter an eine URL angehängt werden

unter der Tabelle dient dazu, um neue Einträge zu generieren.

Zu jedem Projekt gibt es die Möglichkeit, sich eine ToDo-Liste anzeigen zu lassen, bei der es sich um eine Gesamtansicht der eingetragenen Punkte der einzelnen Nutzer handelt.

▼ Computergrafik				
✓	Beleuchtung	Fälligkeit: 27.09.2009	<input checked="" type="checkbox"/>	✗
✓	Renderpipeline	Fälligkeit: 27.09.2009	<input checked="" type="checkbox"/>	✗
	Clipping korrigieren	Fälligkeit: 27.09.2009	<input type="checkbox"/>	✗
<b>Punkt zur ToDo-Liste hinzufügen</b>				
► Datenstrukturen				

Abbildung 6.4: In jeder ToDo-Liste sind dem Benutzer diverse Interaktionsmöglichkeiten mittels Ajax geboten wie ein ToDo ein erledigt markiert oder löschen.

#### 6.7.4 Kalender

Der Kalender für TeamVision wurde in Form eines Wochenkalenders umgesetzt. Dazu wurde die Klasse Calendar (calendarclass.php) in PHP umgesetzt, die unter der Verwendung der Klasse day (dayclass.php) mit einer Datumsangabe, welche für die Funktion showWeek(\$day,\$month,\$year) benötigt wird, einen Kalender für die Woche erstellt, in der dieser Tag ein Wochentag ist.

Die Interaktionen des Benutzers mit dem Kalender waren ursprünglich wie beim Adressbuch und der ToDo-Liste über Ajax gesteuert worden. Es kam jedoch beim hinzufügen von Terminen zu Problemen, da während einer Interaktion, beispielsweise durch den Wechsel der Woche, die schon gesendeten Daten weiterhin im Hintergrund übermittelt und damit in der Datenbank eingetragen wurden.

Die Lösung hierfür ist eine Erlaubnis für das hinzufügen von einem Termin in die Datenbank, die erst erteilt werden muss. Dies geschieht, sobald das Formular für das hinzufügen eines Termins aufgerufen wird. Nach dem Bestätigen werden die Termindaten eingefügt und die Erlaubnis wird dabei wieder entzogen. Beim nächsten Wechsel der Woche wird die Datenbank nicht angesprochen, da keine Erlaubnis vorliegt.

Dem Benutzer sind folgende Interaktionsmöglichkeiten im Bezug auf den Kalender im Prototyp gegeben: Eine Woche vor oder zurück gehen Termine eintragen und Termine anzeigen lassen in einem Popup - dabei für bestimmte Uhrzeiten oder den ganzen Tag. Um das Popup für die Gesamtansicht der Termine an einem bestimmten Wochentag abzurufen, muss der Name des Wochentags angeklickt werden. Bei den Terminen werden zwei Kategorien unterschieden: Termine privater Natur und Termine auf ein Projekt bezogen. Der Nutzer kann im Formular für das Anlegen neuer Termine explizit angeben, dass ein neuer Termin privater Natur ist. Wird keine Angabe gemacht, so wird angenommen, dass der Termin nicht auf ein Projekt bezogen ist. Wählt der Nutzer ein Projekt aus, so wird der Termin allen Teammitgliedern des Projektes angezeigt.

Da es passieren kann, dass zu einer Uhrzeit mehrere Termine stattfinden, werden Termininformationen nicht im Kalender dargestellt. Stattdessen wird ein Link platziert, der auf das Popup mit den Detailinformationen verweist. Dies liegt auch daran, dass hierfür ein ausgereiftes Konzept der visuellen Darstellung von mehreren kollidierenden Terminen erforderlich ist und es kein Ziel dieser Arbeit war, einen Kalender zu entwickeln, der schon existierende Software ablöst.

## 6.8 Projektverwaltung

Im Konzept wurde für die Projektverwaltung eine strikte Trennung der Ansichten im Projektbaum vorgesehen, so dass ein Controller Informationen zum Stand des Projektes im Bezug auf Arbeitszeiten, Terminlichkeiten und Budget auf einen Blick erhalten kann. Ein Entwickler hingegen kann den Entwicklungsstand verfolgen und ggf. bei Problemen über Lösungen diskutieren. Der Projektleiter kann die verschiedenen Übersichten nutzen, um sich ein Bild vom Projektstatus zu verschaffen und bei Problemen die Ursache zu ergründen.

Dieses Konzept wurde im Prototyp nur vereinfacht umgesetzt, so dass es einen Projektbaum mit einer Ansicht gibt, die ein Gefühl dafür vermitteln soll, welche Ideen aus dem Konzept auch praktisch realisiert werden können.

### 6.8.1 Projektübersicht: Der Projektbaum

Im Projektbaum ist, wie im Konzept beschrieben, das Projekt durch die Wurzel präsentiert. An die Wurzel können Meilensteine und Komponenten als Kinder angehängt werden. Die Meilensteine können wieder zwei verschiedene Typen von Kindelementen fassen: Komponenten bzw. Aufgaben. Eine Komponente kann nur Aufgaben als Kinder haben und Aufgaben selbst haben keine Kinder. Aufgrund der Einschränkungen der Möglichkeiten, Kinder zu generieren, ist der Projektbaum auf eine Tiefe von 3 beschränkt. Abhängigkeiten zwischen den einzelnen Elementen wurden im Prototyp nicht implementiert.

In TeamVision sind alle Elemente des Projektbaums bis auf die Wurzel auch als eigenständiger Eintrag in der Datenbank vorhanden, da die Wurzel aus den Projektinformationen generiert wird. Jedes Element des Projektbaums hat eine eigene Statusanzeige. Diese zeigt an, ob ein Element noch geplant wird, aktiv ist, abgeschlossen oder abgebrochen wurde. Weiterhin wird angezeigt wie viele Tage noch bis zum Erreichen der Deadline vorhanden sind. Dies wird jedoch nur berechnet, sofern das Element aktiv ist.

Die einzelnen Knoten des Projektbaums werden durch Bereiche (DIV) dargestellt. Die Elemente einer Ebene sind in einer Tabelle nebeneinander als jeweils eine Spalte eingebettet. Die Tabelle ist wiederum in einen Bereich eingebettet, der den Abstand zu den anderen Bereichen verwaltet. Durch die Struktur des Projektbaums sind insgesamt drei Ebenen unter der Wurzel zu zeichnen. Ob und welche Ebene zu zeichnen ist, wird über Parameter aus der URL ausgelesen, welche als GET-Parameter bezeichnet werden. Der erste Ansatz mit Ajax direkt die Werte an den Server zu übermitteln und

den Baum als fertige Zeichnung zu erhalten hat gezeigt, dass dies nicht flexibel genug ist, um nach der Rückkehr aus Untermenü die zuletzt gewählte Ansicht wieder herzustellen. Die Lösung mittels GET-Parametern hat auch den Vorteil, dass neue Daten leicht angehängt und weiter verarbeitet werden können. So wurde z.B. anschließend die Position des Elternelements für die jeweilige Ebene sowie die zuletzt erreichte Scrollposition übermittelt, um zum einen die Kinder unter ihrem Elterknoten zu zentrieren sowie die aktuelle Position im Anzeigefenster beizubehalten. Damit wird der Gedanke von Ajax verfolgt, eine Anwendung auf den Benutzer so weit wie möglich wie eine Desktopanwendung wirken zu lassen, so dass nach dem erneuten laden einer Seite nicht erst wieder mühselig auf die zuletzt besuchte Position gescrollt werden muss.

Um dem Benutzer die Zuordnung zu den Elterknoten zu erleichtern, werden bei der Auswahl der Detailfunktion eines Elterknotens der Elterknoten und seine Kinder deren Hintergrund in der gleichen Farbe gefärbt. Ebenso geschieht dies mit der folgenden Ebene. Der Elterknoten für die dritte Ebene wird etwas dunkler gezeichnet und seine Kinder erhalten im Hintergrund den gleichen Farbton. Nicht davon betroffen ist der Rahmen um das Element.

Der Rahmen eines Elementes gibt zusammen mit einer Statusmeldung (in Planung, aktiv, abgeschlossen, abgebrochen, Problem) den aktuellen Status des Elements an. Daran kann der Benutzer sofort erkennen, welche Elemente des Projektbaums genauer analysiert werden müssen. Eine Hilfe dabei ist, dass ein Elterknoten sich färbt, sobald ein Kind in Verzug ist oder ein Problem gemeldet wurde. Im Prototyp wurde die Anzeige auf aktive Elemente zunächst beschränkt. Es wurden Überlegungen durchgeführt, ob dies auch für andere Statusmeldungen der Eltern gelten soll, allerdings sind abgeschlossene Elemente nicht mehr von Bedeutung. Einzig die Wurzel des Projekts weist dann noch darauf hin, dass ein Problem vorliegt, da sie den Baum nicht traversiert, sondern alle Knoten unabhängig in einer SQL-Query auf Probleme prüfen lässt. Eine mögliche Lösung ist ein Icon, welches auch bei abgeschlossenen Elementen usw. anzeigt, dass ein Kind ein Problem hat. Diese Lösungsidee wird jedoch als kritisch bewertet, da dieses Icon leicht übersehen werden könnte in größeren Baumstrukturen. Es wird die Annahme getroffen, dass Nutzer möglichst sinnvolle Eingaben machen und deshalb für eine abgeschlossene Komponente gilt, dass auch die Kinder abgeschlossen sind, weshalb bei einer Statusänderung eines Elterknotens auf einen nicht aktiven Zustand (außer Problem) die Kinder den gleichen Zustand übernehmen müssen.

Bei der Implementierung des Baumstruktur und dessen Darstellung sind einige Probleme aufgetreten. Unter anderem ist es in HTML nicht vorgesehen Zeichnungen zu erstellen. Deshalb wird eine Bibliothek für JavaScript von Walter Zorn<sup>6</sup> in der Version 3.05 verwendet, um Linien zu zeichnen. Die Bibliothek zeichnet, indem ganz viele kleine Bereiche erzeugt werden, die gefärbt sind. Daraus wird dann sukzessive die Zeichnung aufgebaut. Es hat sich jedoch gezeigt, dass diese Zeichnungen nicht immer korrekt dargestellt werden, da die Linien zu dünn sind und vom Browser deshalb manchmal teils gelöscht werden. Um diese Problematik zu umgehen, wurde eine Linie vom Elterknoten auf das Kindelement direkt unter dem Knoten gezeichnet. Dabei musste noch das Problem umgangen werden,

---

<sup>6</sup>Siehe: <http://www.walterzorn.de/jsgraphics/jsgraphics.htm>

dass die Position der Elemente auf dem Bildschirm bzw. ihr Offset nicht genau genug ermittelt werden kann für eine saubere Zeichnung. Die Lösung hierfür ist ein Bereich, der zwischen den Bereichen für die Ebenen liegt<sup>7</sup>. Einzig zwischen der Wurzel und der ersten Ebene wurde keine Linie gezeichnet, da die Wurzel eine feste Position hat.

Abbildung 6.5 am Ende des Kapitels zeigt die aktuelle Gestalt des Projektbaums.

## 6.8.2 Teams, Rollen und Zuständigkeiten im Projekt

Dem Abteilungsleiter bzw. Projektleiter ist es möglich über ein Formular Benutzer dem Team zuzuweisen und ihnen dabei Rollen zuzuweisen. Weiterhin verfügen beide Rollen über die Möglichkeit das Projektteam zu verwalten und Rollen zu ändern. Dabei kann ein Abteilungsleiter keine andere Rolle im Projekt einnehmen die des Abteilungsleiters. Im Gegensatz zum Konzept kann der Abteilungsleiter im Prototyp keine weitere Rollenausprägung erhalten.

Zu jedem Element des Projektbaums mit Ausnahme der Wurzel können Personen zugeteilt werden. Dabei kann ein Entwickler nur als Entwickler und ein Tester nur als Tester zugeteilt werden. Mitarbeiter mit anderen Rollen werden unter der Kategorie „andere Aufgaben“ zugeteilt. Dabei ist im Prototyp keine weitere Differenzierung ihrer Aufgaben möglich.

## 6.8.3 Details aus dem Projekt über die Zoomfunktion, Listen und Popups

Meilensteine, Komponenten und Aufgaben besitzen eine Zoomfunktionalität. Diese Funktion wird durch den Link auf den Namen des jeweiligen Elements aufgerufen. Dabei wird ein neues Fenster als Popup geöffnet, in dem Details zum jeweiligen Element wie dessen Beschreibung und die zuständigen Mitarbeiter gelistet sind. In diesem Popup ist auch ein kleines Formular gelistet, mit dem ein Artikel in einem MediaWiki angelegt bzw. ergänzt werden kann.

Die Wurzel verfügt über eine Funktion, mit der die Projektspezifikationen in einem Popup aufgerufen werden. Dort können der Abteilungs- bzw. der Projektleiter sowohl die erfüllten als auch die offenen Spezifikationen einsehen und den Status oder die Projektspezifikation selbst ändern.

## 6.9 Kooperation mit einer Fremdunggebung am Beispiel MediaWiki

Im Konzept war es vorgesehen, dass eine Besonderheit von TeamVision die Möglichkeit sein soll, Aufgaben an Fremdunggebungen abzugeben. Es erwies sich jedoch als nicht machbar, eine Implementierung zu entwickeln, die von sich aus alle Fremdunggebungen unterstützt. Deshalb wurde eine beispielhafte Fremdunggebung ausgewählt, welche im projektmanagement zum Einsatz kommt. Die Wahl ist auf das MediaWiki gefallen.

Das MediaWiki bietet eine API<sup>8</sup> als Schnittstelle, die über HTTP-Requests angesprochen werden

<sup>7</sup>Gemeint ist ein DIV, welches von zwei DIVs umgeben ist.

<sup>8</sup>API steht für Application Programming Interface und ist somit eine programmierbare Schnittstelle für Anwendungen

kann. Diese Requests müssen jedoch als POST erfolgen, d.h. die Daten werden nicht in der URL angezeigt oder übermittelt. Dies dient laut MediaWiki der Sicherheit. Wenn ein Programm mit der API arbeiten will, ist der Ablauf für das Erstellen eines Artikels wie folgt:

1. **Login:** Bevor das MediaWiki genutzt werden kann, muss eine Authentifizierung stattgefunden haben. Bei Erfolg wird ein Cookie gesetzt und eine Sitzung gestartet. Dies geschieht automatisch über die Antwort der API. Wenn kein Cookie gesetzt werden kann, so muss bei den weiteren Anfragen die Daten, die im Cookie gesetzt werden würden, durch die Anfrage selbst übermittelt werden.
2. **Erlaubnis für die Bearbeitung eines Artikels holen:** Bevor in einem Artikel geschrieben werden kann, muss erst ein Token vom MediaWiki abgefragt werden, welches die Erlaubnis gibt, einen Artikel anzulegen bzw. zu editieren. Dabei muss der Titel des Artikels angegeben werden.
3. **Artikel anlegen / bearbeiten:** Als letzter Schritt muss nur noch die Seite in Form von Titel und Inhalt mit dem Token, welches für die Seite erhalten wurde, an das MediaWiki gesendet werden.

Im Prototyp wird aus Gründen der Einfachheit als Benutzername immer der in der Konfiguration des MediaWikis angegebene Benutzer verwendet. In einem realen Einsatzszenario die Authentifizierung mit dem Account des Benutzers erfolgen und dann auf dessen Namen ein Artikel anlegt oder ergänzt werden.

Im Prototyp ist die Implementierung der Anbindung an die Ferndumgebung durch drei Dateien erfolgt. Die Datei „mediawikifunctions.php“ beinhaltet alle Funktionen, die aufgerufen werden, um mit der API zu kommunizieren. „addarticle.php“ enthält den Code, der nötig ist, um Login durchzuführen und den Artikel anzulegen. Dort werden auch die Antworten der API, welche im XML Format übergeben werden, geparkt. Es wurde jedoch nur die minimal nötigen Informationen extrahiert und keine Fehlerbehandlung durchgeführt. Die letzte Datei „formforpopup.php“ beinhaltet Quelltext in HTML, der ein Formular zum Erzeugen eines WikiArtikels aufruft.

Der Prototyp verfügt nicht über die Fähigkeit bestehende Texte eines Artikels zu verändern.

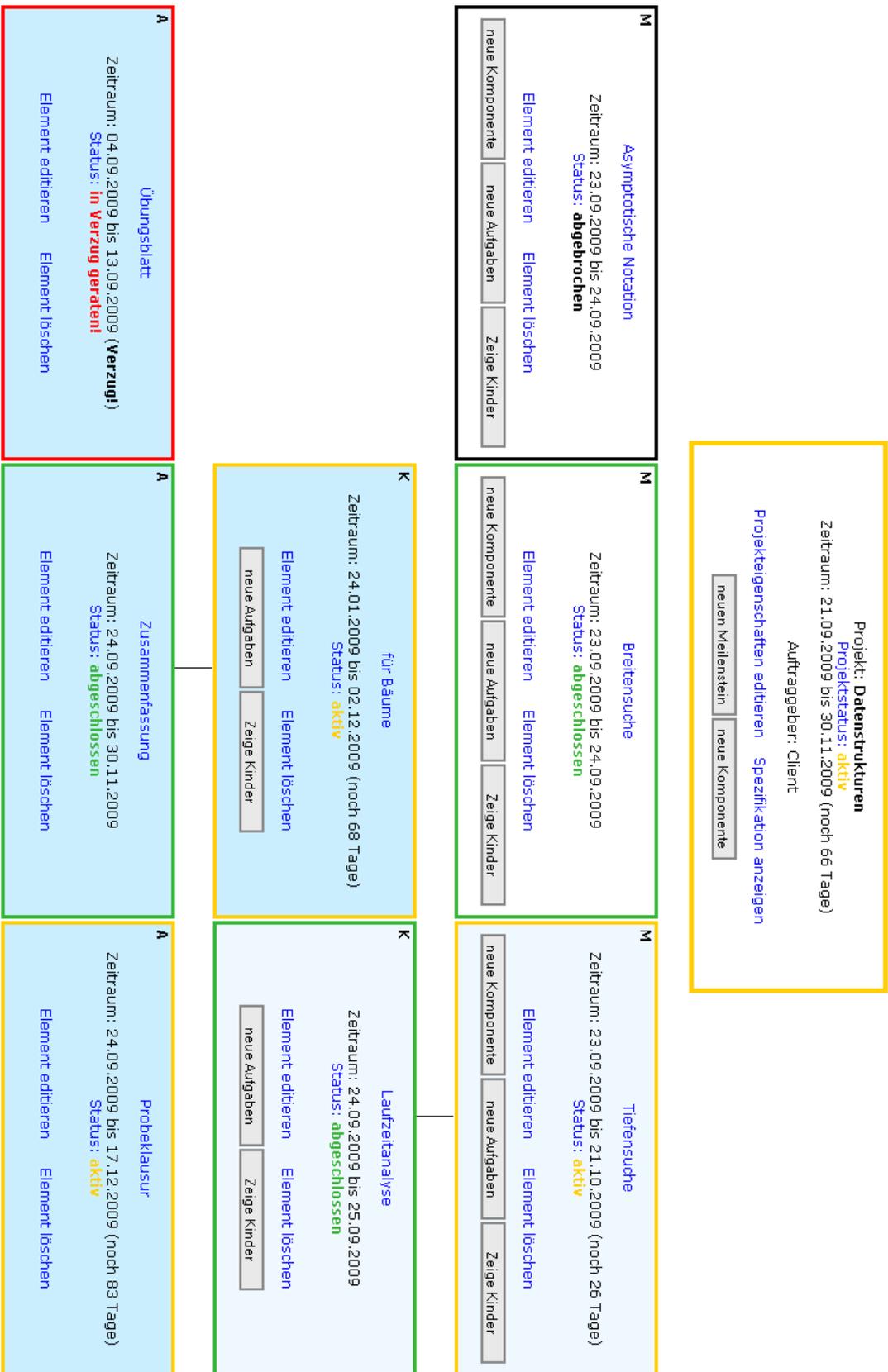


Abbildung 6.5: Darstellung eines Projektbaums über alle Ebenen.

## Kapitel 7

# Zusammenfassung, Fazit und Ausblick

### 7.1 Zusammenfassung und Fazit der Arbeit

Das wesentliche Ziel dieser Arbeit ist es gewesen, ein Verständnis dafür zu entwickeln, welche Kriterien entscheidend für den Erfolg eines Projektteams sind, sowie diese Kriterien im Konzept von TeamVision umzusetzen. In der Einleitung und in den Grundlagen wurde hierzu dargestellt, was ein virtuelles Team ist und welchen Einflüssen es ausgesetzt ist. Als wesentliche Punkte waren dabei die strukturbedingten Probleme (räumliche Trennung) und das in der Praxis häufige Überspringen des Stormings als eine wichtige Phase in der Bildung des Teams, in dem durch die gemeinsame Konfliktlösung während der Teambildung eine funktionale Vertrautheit aufgebaut wird, die letztendlich für die Effizienz des Teams von entscheidender Bedeutung ist.

Im dritten Kapitel wurde durch die Analyse einiger Web 2.0 Tools festgestellt, dass die existierenden Tools zu wenig auf die Bedürfnisse der Benutzer eingehen. Der Funktionsumfang wird vom Anbieter bestimmt und nicht von der Art des Projektes oder besonderen Anforderungen an das Tool. Auch ist die Bedienung der Tools meist umständlich geregelt, indem Informationsvermittlung zu verschiedenen Sachverhalten in getrennten Ansichten erfolgt, obwohl diese erst in Zusammenhang gebracht Probleme aufzeigen oder an wichtige, noch offene Punkte im Projekt erinnern können.

Das vierte Kapitel hat die Anforderungen an TeamVision für IT-Projekte mit verschiedenen Rollen aufgezeigt. Anhand eines einfachen Rollenmodells, bei dem in den gegebenen Rollen nicht weiter differenziert wird, wurde aufgezeigt, welche unterschiedlichen Anforderungen die jeweiligen Rollen an ein Tool stellen, welche durch ihre Aufgaben im Projekt bestimmt sind. So muss ein Entwickler in der Entwicklungsphase immer auf dem aktuellen Stand der ihn betreffenden Entwicklung sein, insbesondere, wenn er nicht alleine an einem Modul arbeitet. Dagegen muss ein Controller jederzeit den gesamten Projektstatus erfassen können.

Ein Aspekt, der sich in Kapitel 4, aber auch in Kapitel 5 gezeigt hat, ist der Mehrwert, den eine Umsetzung der Anforderungen und des Konzepts von TeamVision mit sich bringen kann. Durch die Vernetzung von Informationen und das Einbinden von Fremddumgebungen, ist ein neben einer langsam voran schreitenden Ersetzung alter Tools auch die Möglichkeit gegeben, TeamVision durch weitere

Programme zu ergänzen, dabei aber immer zentral aus TeamVision heraus auf Informationen zuzugreifen.

In Kapitel 5 wurde das Konzept vorgestellt, welches im Projektbaum die Kernkomponente von TeamVision sieht, die jeder Rolle dabei hilft, möglichst schnell und einfach den Projektstatus zu erfassen und Informationen schnell zu erreichen. Die Erfassung der Informationen wird dadurch beschleunigt, dass Zoomfunktionen eingebaut sind, die Detailinformationen auf Wunsch präsentieren.

Mit der prototypischen Implementierung im sechsten Kapitel wurde die Umsetzbarkeit des Konzepts an entscheidenden Punkten ausprobiert. Nachdem anfangs einige Standardwerkzeuge wie Adressbuch, ToDo-Liste und Kalender mit dem Grundgerüst umgesetzt wurden, hat die Umsetzung des Projektbaums wesentlich mehr Zeit in Anspruch genommen und den erheblichen Entwicklungsaufwand von Webapplikationen mit HTML 4 aufgezeigt. Um die grafische Oberfläche für den Benutzer zu gestalten musste erheblicher Entwicklungsaufwand betrieben werden. Jede noch so kleine Änderung musste präzise geplant werden, damit keine Verschiebung in der Oberfläche auftritt. Hauptursache liegt darin, dass HTML das Seitenlayout während des Parsens des Quelltextes erstellt und dadurch die Anfangselemente nicht mehr durch spätere Elemente beeinflusst werden können. Auch sind die Freiheitsgrade der Darstellung eingeschränkt z.B. durch das FF von nativen Zeichenfunktionen in HTML.

Die Implementierung konnte jedoch aufzeigen, dass die Grundidee möglichst viele Funktionen in einer Ansicht in Form des Projektbaums unterzubringen, funktioniert. Um die Informationsvernetzung zu maximieren müssen jedoch komplexe Zoomfunktionen geschrieben werden, die nur zu einem geringen Teil implementiert wurden.

Obwohl die Schnittstellen von TeamVision bzgl. der Arbeit mit Fremdumgebungen in der Implementierung stark eingeschränkt wurden, konnte der Prototyp aufzeigen, dass eine Integration von Fremdumgebungen grundsätzlich möglich ist. Der Nutzungsgrad der Fremdkomponente ist dabei von der Schnittstelle dieser Komponente abhängig. Am Beispiel des MediaWiki ist deutlich geworden, dass aufgrund der spezifischen API kein generisches Konzept erarbeitet werden kann, welches auf alle Fremdumgebungen angewendet werden kann. Jedoch kann für jede Fremdumgebung ein Modul geschrieben werden, welches die API bzw. Schnittstelle der Fremdumgebung nutzt. Dann muss nur noch das Modul in TeamVision importiert werden und ist damit nutzbar. D.h., wenn das Modul für eine Fremdumgebung einmal geschrieben ist, so kann jeder, der TeamVision einsetzen möchte, dieses nutzen, um eine verbreitete Fremdumgebung wie das MediaWiki oder Mantis zu nutzen.

Damit ergibt sich auch ein weiterer Mehrwert von TeamVision: Im Gegensatz zu den meisten Tools ist der Code frei zugänglich und kann weiter entwickelt werden, womit auch die Möglichkeit gegeben ist, diese Weiterentwicklungen allen Nutzern zugänglich zu machen.

Das Fazit für diese Arbeit lautet somit wie folgt: TeamVision kann seine Zielgruppe erreichen, bedarf jedoch in der Umsetzung weiterer Ergänzungen, um den Grad der Informationsvernetzung zu erreichen. Insbesondere durch das Fehlen von Abhängigkeiten im Projektbaum und der rollenspezifischen Ansichten sowie der Controllingwerkzeuge wie in Abb.5.3 und 5.4 vorgestellt. Gegenüber den

in der State-of-the-Art Analyse untersuchten Tools bietet TeamVision jedoch den Vorteil, dass einzelne Funktionalitäten des Tools nicht nebeneinander existieren, sondern so weit wie möglich versuchen sich untereinander zu vernetzen.

## 7.2 Ausblick

Das Konzept von TeamVision ist auf eine bestimmte Zielgruppe fokussiert und in einem einfachen Rollenmodell auf eine Abteilung beschränkt worden. Wenn jede Instanz von TeamVision als eine Abteilung verstanden werden kann, könnte eine Schnittstelle erstellt werden, die mehrere Abteilungen bzw. ein ganzes Firmennetz vernetzt. Somit könnte TeamVision in Firmennetzen oder auch zwischen mehreren Firmen eingesetzt werden. Dabei wäre eine Import- und eine Exportfunktion von Nöten. Ein Einsatzszenario für die Kommunikation zwischen Firmen ist ein Projekt für eine Produktentwicklung, bei der mehrere große Hersteller gemeinsam ein Produkt auf den Markt bringen möchten, indem sie ihr Wissen und ihre Ressourcen vereinen.

Das Rollenmodell könnte verfeinert und durch ein Modul ergänzt werden, welches dem Administrator bzw. Abteilungsleiter Änderungen an den Berechtigungen der einzelnen Rollen erlaubt.

Weiterhin könnte der Projektbaum erweitert werden, indem er einen neuen Ansatz in der Visualisierung verfolgt, um schneller zu vermitteln, in welchen Teilbäumen für den Benutzer aktuell relevante Informationen vorhanden sind. Dazu könnte eine Darstellung verwendet werden, in der die Knoten auf kleine Formen reduziert werden und nur über die Farbe verdeutlicht wird, welcher Teilbaum aktuell kritisch ist. So könnten eventuell auch komplexere Baumstrukturen sowie deren Beziehungen untereinander in einer Ansicht gezeichnet werden. Auch Filterfunktionen für nicht benötigte Informationen könnten dabei helfen, die Übersichtlichkeit des Projektbaums zu steigern, indem beispielsweise nur aktive oder problematische Elemente angezeigt werden. Auch neue Darstellungsformen könnten erprobt werden, um die Datenflut im Projektbaum zu beherrschen und einen noch schnelleren Überblick über Probleme im Projekt zu ermöglichen.

Bei der Einbindung des MediaWiki könnte die Schnittstelle weiter ausgebaut werden, so dass mehr Informationen automatisch in das Wiki übertragen, aber auch mehr Informationen wieder aus dem Wiki ausgelesen werden, damit Artikel nicht nur erzeugt oder ergänzt, sondern auch bearbeitet werden können. Auch die Entwicklung einer API zu TeamVision wäre eine Möglichkeit, TeamVision selbst für andere Umgebungen nutzbar zu machen. Wenn diese API beispielsweise die Anzeige von bestimmten Funktionen wie dem Projektbaum übermitteln könnte, dann würde TeamVision damit auch an Attraktivität gewinnen, da dadurch die gewohnte Umgebung nicht gewechselt werden muss, sondern durch TeamVision erweitert wird.

In TeamVision wurden psychologische Aspekte wie Freiheiten in der Benutzung nur bedingt berücksichtigt. Da für den Erfolg eines Projektes jedoch die psychologische Komponente eine große Rolle spielt, wie in Kapitel 2.1.3 mit der Gruppentheorie und dessen Anwendung auf das virtuelle Team gezeigt werden konnte, ist eine Evaluation von TeamVision und eine Verbesserung der Oberflä-

che anhand der Evaluationsergebnisse eine Möglichkeit, die Benutzbarkeit und ggf. auch die sinnige Vernetzung von Informationen voranzutreiben, indem die Bedürfnisse der Teammitglieder stärker in TeamVision eingebracht werden.

Eines sollte in jedem Fall deutlich geworden sein: Immer komplexer werdende Aufgaben in Projekten durchzuführen ist nur dann von Erfolg gekrönt, wenn die Informationen über das Projekt möglichst eng miteinander verknüpft werden, so dass der Benutzer gezielter und vor allem schneller an entscheidende Informationen gelangt.

# Anhang A

## Glossar

**Ajax:** steht für „Asynchronous JavaScript and XML“. Ajax beschreibt ein Konzept für asynchrone Datenübertragung, welches die Möglichkeit schafft, Webanwendungen dynamischer zu gestalten und Inhalte an Interaktionen mit dem Benutzer zu orientieren, beispielsweise anhand einer Auswahl an mehreren Optionen das Layout gestalten.

**BSCW:** Das BSCW (Basic Support for Cooperative Work) ist eine von ORBI Team und Fraunhofer FIT entwickelte Groupware mit der Dokumente über einen Webserver miteinander geteilt werden können. Dabei werden Dateien in Oberflächen mittels einer Ordnerstruktur dargestellt. Angelegte Dateien können bei Bedarf verändert oder entfernt werden. Die Kosten liegen bei mindestens 1770 Euro, wenn ein eigener Server verwendet wird.

**GIMP:** GIMP (GNU Image Manipulation Program) ist ein frei verfügbares Bildbearbeitungsprogramm unter GPL Lizenz. Es unterstützt alle gängigen Bildformate und häufig gebräuchliche Operationen in Form von vordefinierten Filtern (Weichzeichnen, Schärfe, Farbanpassung,...).

**HTML:** Hypertext Markup Language oder auch Hypertext ist eine Auszeichnungssprache zur Strukturierung von Webinhalten, die vom Webbrowser dargestellt werden können. Aktuell wird HTML 4.01 verwendet, welches aufgrund seines Alters nur eingeschränkte Möglichkeiten zur Darstellung von Multimediainhalten besitzt. Deshalb werden Websites oft durch weitere Techniken wie Ajax oder Flashinhalte unterstützt. Der kommende Standard HTML 5 soll Multimediainhalte unterstützen.

**Mantis:** Bei Mantis handelt es sich um einen so genannten Bugtracker. Bugtracker werden in der Entwicklung dazu verwendet Fehlverhalten von Software zu erfassen und deren Lösung zu verwalten und dokumentieren. Mantis ist in PHP geschrieben und frei verfügbar.

**nominal:** Werte, die unterschieden, aber nicht in eine Rangfolge gebracht werden können. Beispiel: Geschlecht (männlich, weiblich).

**ordinal:** Werte, die in eine Reihenfolge gebracht werden können wie größter Wert, zweitgrößter Wert,..., n-ter Wert. Beispiel: Volumenangaben (Metropole, Großstadt, Stadt, Kleinstadt, Dorf,...)

**quantitativ:** Werte, die durch eine Zahlenangabe oder Menge dargestellt werden. Damit basiert die Ordnung auf der Größe der Zahl bzw. Kardinalität der Menge. Beispiel: Fördermenge in Tonnen

**PHP:** An C Syntax angelehnte Skriptsprache, welche 1995 von Rasmus Lerdorf entwickelt wurde. PHP wurde für die Darstellung dynamischer Webinhalte entwickelt, kann aber auch Grafiken und PDF erzeugen sowie XML-Daten verarbeiten und besitzt Schnittstellen zu diversen Datenbanken wie MySQL. Am 30.06.2009 ist Version 5.3.0 erschienen.

**SVN:** Software zur Versionsverwaltung, bei der Entwickler Änderungen auf einen Webserver laden, damit andere Entwickler an dieser Version Veränderungen vornehmen können. SVN beinhaltet Logs von Änderungen und das Speichern älterer Versionen, um Änderungen zwischen Versionen zu vergleichen.

**XML:** XML ist die Abkürzung für Extensible Markup Language (dt.: „erweiterbare Auszeichnungssprache“). XML ist eine Auszeichnungssprache mit der Daten strukturiert und dabei hierarchisch geordnet werden können. XML wird oft für Kommunikation zwischen Computern genutzt. Ein XML-Dokument besitzt ein Wurzel- oder Dokumentelement, gefolgt von einem Inhalt oder einem oder mehreren Kindelementen.

**Beispiel:** <wurzel><text>Dies ist ein Text.</text></wurzel>

## Anhang B

# Bilder zum Entwurf der Oberfläche

In diesem Abschnitt sind einige Grafiken, die im Kapitel Konzeption aufgrund ihrer Größe nicht vollständig dargestellt werden konnten, noch einmal in größerem Format dargestellt oder Ausschnitte noch einmal näher beleuchtet.

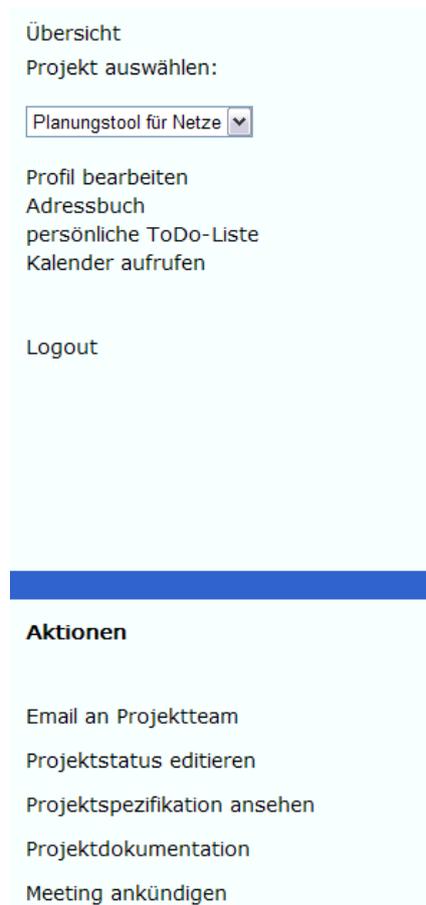


Abbildung B.1: Um die Trennung der Navigationsbereiche deutlich hervorzuheben, haben diese Bereiche einen Abstand von 20px. Zusätzlich hat der zweite Bereich die Überschrift „Aktionen“.

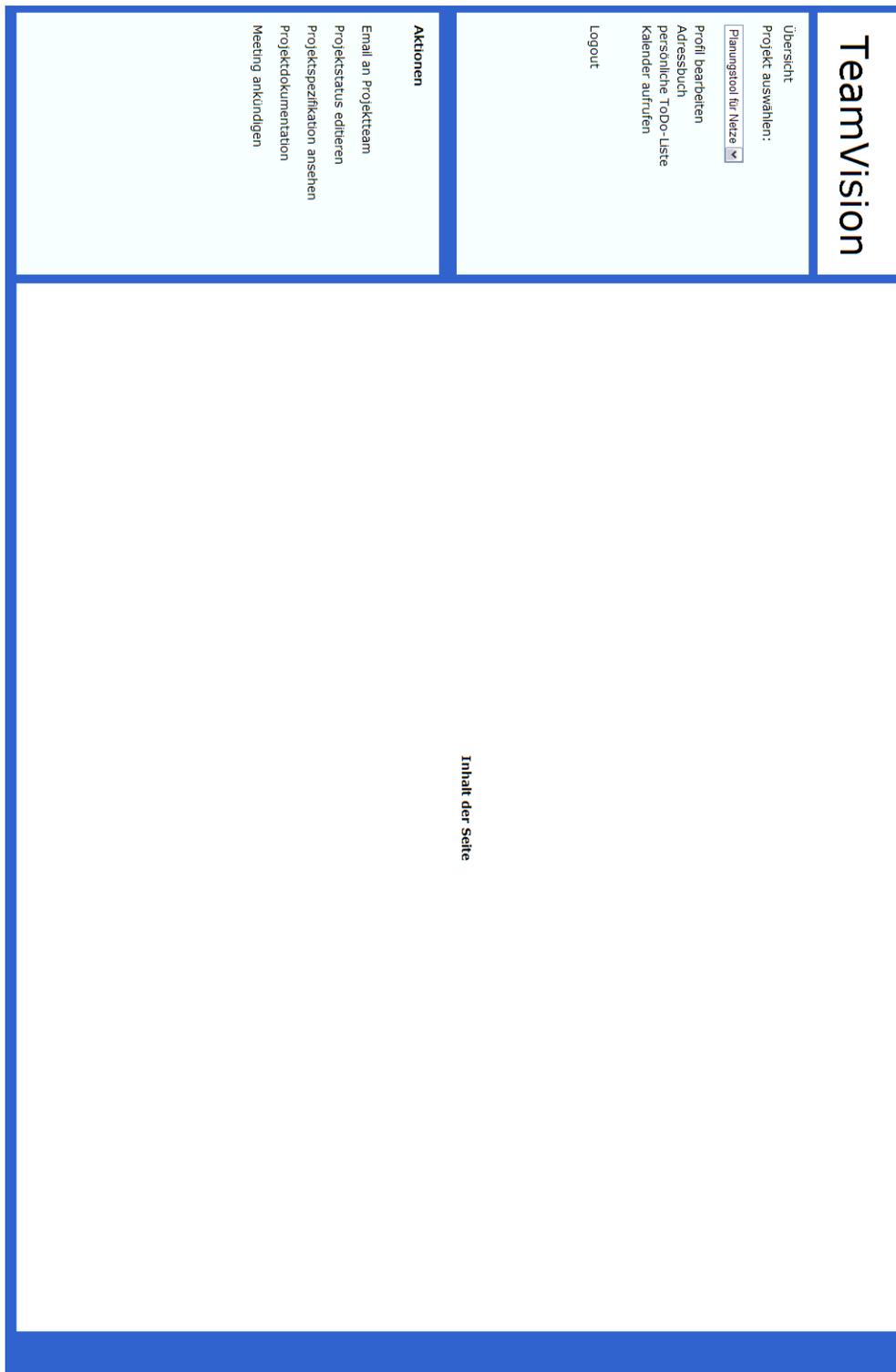


Abbildung B.2: Entwurf der Oberfläche von TeamVision.

# Literaturverzeichnis

- [act] active value GmbH. <http://www.web2null.de/category/projektmanagement>. Aufgerufen 28.07.2009.
- [Alo] Leeron Shalev - CTO Alon Carmel - CEO. <http://www.devunity.com>. Aufgerufen 28.07.2009.
- [AT88] Stephen Gormican Anne Triesman. Virtuelles Teammanagement im Projekt. In *Psychological Review*, vol. 95(1):pp. 15–48, 1988.
- [Aug] Augeo Software SAS. <http://www.planzone.de/>. Aufgerufen 28.07.2009.
- [Beh] Behance LLC. <http://www.actionmethod.com>. Aufgerufen 28.07.2009.
- [ben06] Oliver bendel. Das 1x1 der Wikis und Weblogs. In *Wissensmanagement*, vol. 03/06:pp. 22–25, 2006.
- [Bre07] Dr. Tobias Breiner. Vorlesung: Visualisierung im Wintersemester 06/07 an der Universität Frankfurt, 2007.
- [Bus] Frank Busch. [http://de.wikipedia.org/w/index.php?title=Datei:Gantt\\_diagramm.png&filetimestamp=20040702191119](http://de.wikipedia.org/w/index.php?title=Datei:Gantt_diagramm.png&filetimestamp=20040702191119). Aufgerufen 06.08.2009.
- [c2w] <http://c2.com/cgi/wiki?WikiEngines>. Aufgerufen 06.08.2009.
- [Dro06] Prof. Dr. Oswald Drobnik. Softwaretechnik, 2006. Teil der Veranstaltung PRG-1 im WS 06/07 an der Universität Frankfurt.
- [EFK06] EFK GmbH. [http://www.project-controlling.com/html/proscio\\_-\\_project\\_controlling.html](http://www.project-controlling.com/html/proscio_-_project_controlling.html), 2006. Aufgerufen 17.08.2009.
- [Ger02] Gabriele Jordan Gerhard Graf. Virtuelles Teammanagement im Projekt. In *projektManagement aktuell*, vol. 03:pp. 21–28, 2002.
- [Gol] Golem.de. <http://www.golem.de/0905/67212.html>. Aufgerufen 29.07.2009.
- [GPM09] GPM Deutsche Gesellschaft für Projektmanagement e. V. DIN 69901-1: Projektmanagement - Projektmanagementsysteme - Teil 1: Grundlagen. Berlin, Deutschland, 2009.

- [hau] <http://www.haus-wohnung.eu/Wohnen>. Aufgerufen 06.08.2009.
- [Hei00] Wolfgang Müller Heidrun Schumann. *Visualisierung*. Springer, Berlin Heidelberg, Deutschland, 2000. ISBN 3-540-64944-1.
- [HK05] Bernhard Hirth Hans Königes. <http://www.computerwoche.de/karriere/karriere-gehalt/564909/index2.html>, 2005. Aufgerufen 29.07.2009.
- [iata] iatrocon GmbH. [https://www.ixperio-shop.de/Lizenzen\\_Lizenz\\_fuer\\_project\\_illustrator-58684-670598.html?PHPSESSID=1fba36d096b6ee7531e5834e03afedc2](https://www.ixperio-shop.de/Lizenzen_Lizenz_fuer_project_illustrator-58684-670598.html?PHPSESSID=1fba36d096b6ee7531e5834e03afedc2). Aufgerufen 17.08.2009.
- [iatb] iatrocon GmbH. <https://www.ixperio.de/?q=wichtiges-%C3%BCber-project-illustrator>. Aufgerufen 17.08.2009.
- [Lic] Prof.Dr. Irene K. Lichtscheidl. Additive Farben - RGB Farbmodell <http://www.univie.ac.at/fotografie/digital/rgb.htm>. Aufgerufen 06.08.2009.
- [Lit02] Hans-Dieter Litke. *Projektmanagement*. GU, München, Deutschland, 2002. ISBN 3-7742-4920-2.
- [Mac86] Jock Mackinlay. Automating the design of graphical presentations of relational information. In *ACM Transactions on Graphics (TOG)*, vol. 5:pp. 110–141, 1986.
- [Nor79] Normenausschuss Farbe im Deutsche Institut für Normung e. V. DIN 5033: Farbmessung, Grundbegriffe der Farbmeterik. Berlin, Deutschland, 1979.
- [Pid] Pidgin Community. <http://www.pidgin.im/>. Aufgerufen 08.08.2009.
- [Ras03] Kevin Tatroe Rasmus Lerdorf. *Programmieren mit PHP*. O'Reilly, Balthasarstr. 81, 60670 Köln, 2003. ISBN 3-89721-177-7.
- [Sch09] Susanne Schmidt. <http://www.heise.de/open/Die-Zukunft-von-Perl--/artikel/132658/0>, 2009. Aufgerufen 22.09.2009.
- [Spe00] Robert Spence. *Information Visualization*. Addison-Wesley, Oxford, England, 2000. ISBN 0-201-59626-1.
- [Sta91] Wolfgang H. Staehle. *Management: Eine verhaltenswissenschaftliche Perspektive*. Vahlen, München, Deutschland, 1991. ISBN 3800623447.
- [Tea] Teambox. <http://www.teambox.com>. Aufgerufen 28.07.2009.
- [Thoa] Peter Thoeny. <http://twiki.org/>. Aufgerufen 28.07.2009.
- [Thob] Peter Thoeny. <http://twiki.org/cgi-bin/view/Main/TWikiInstallation>. Aufgerufen 06.08.2009.

- [tik] <http://info.tikiwiki.org/tiki-index.php>. Aufgerufen 06.08.2009.
- [Van] Vancouver IT Services, Inc. <http://www.goalsontrack.com>. Aufgerufen 28.07.2009.
- [War00] Colin Ware. *Information Visualization: Perception for Design*. Academic Press, San Diego, USA, 2000. ISBN 1-55860-511-8.
- [We] Artikel wurde in Kooperation mehrerer registrierter Benutzer auf Wikipedia erstellt. <http://de.wikipedia.org/wiki/Wiki>. Aufgerufen 28.07.2009.
- [Wei04] Gerald M. Weinberg. *Die Psychologie des Programmierers*. mitp, Bonn, Deutschland, 2004. ISBN 3-8266-1465-8.
- [Wika] Wikimedia Foundation Inc. <http://www.mediawiki.org/wiki/Manual:Extensions>. Aufgerufen 08.08.2009.
- [Wikb] Wikimedia Foundation Inc. [http://www.mediawiki.org/wiki/Manual:Special\\_pages](http://www.mediawiki.org/wiki/Manual:Special_pages). Aufgerufen 17.08.2009.
- [Wikc] Wikimedia Foundation Inc. <http://www.mediawiki.org/wiki/MediaWiki>. Aufgerufen 28.07.2009.
- [WSC84] Robert McGill William S. Cleveland. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. In *Journal of the American Statistical Association*, vol. 79, No. 387.:pp. 531–554, 1984.