



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Konzeption und Entwicklung einer hybriden App zur Organisationsverwaltung

Masterarbeit

zur Erlangung des Grades Master of Science (M.Sc.)
im Studiengang Computervisualistik

vorgelegt von
Lisa Werner

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Bastian Kraye, M.Sc.
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im September 2019

Abstrakt

Vereine, wie zum Beispiel die Pfadfinder, stützen sich auf die Arbeit ihrer ehrenamtlichen Mitglieder, welche eine Vielzahl von Aufgaben zu erledigen haben. Oftmals kommt es in den Reihen der Mitglieder zu plötzlichen Änderungen in den Organisationsteams und Ämtern, wobei Planungsschritte verloren gehen und Unerfahrenheit bezüglich der Planung besteht. Da die speziellen Anforderungen durch bereits existierende Tools nicht abgedeckt werden, wird zur Unterstützung von Vereinen in Bezug auf die genannte Problematik in dieser Arbeit ScOuT, ein Planungstool für die Organisationsverwaltung, konzipiert und entwickelt. Der Schwerpunkt lag darauf verschiedene geeignete Richtlinien und heuristische Methoden zu identifizieren und zu verwenden, um eine gebrauchstaugliche Benutzeroberfläche erstellen zu können. Das entwickelte Produkt wurde im Rahmen der Arbeit empirisch durch eine Benutzerumfrage bezüglich der Gebrauchstauglichkeit ausgewertet. Das Ergebnis dieser Studie zeigt, dass bereits ein hohes Maß des angestrebten Ziels durch den Einbezug der Richtlinien und Methoden erreicht werden konnte. Daraus lässt sich im großen Kontext schließen, dass mithilfe von benutzer-spezifischen Konzeptideen und der Anwendung geeigneter Richtlinien und Methoden eine zielführende Grundlage für eine gebrauchstaugliche Anwendung zur Unterstützung von Vereinen erstellt werden kann.

Abstract

Clubs, such as Scouts, rely on the work of their volunteer members, who have a variety of tasks to accomplish. Often there are sudden changes in their organization teams and offices, whereby planning steps are lost and inexperience in planning occurs. Since the special requirements are not covered by already existing tools, ScOuT, a planning tool for the organization administration, is designed and developed in this work to support clubs with regard to the mentioned problems. The focus was on identifying and using various suitable guidelines and heuristic methods to create a usable interface. The developed product was evaluated empirically by a user survey in terms of usability. The result of this study shows that already a high degree of the desired goal could be reached by the inclusion of the guidelines and methods. From this it can be concluded that with the help of user-specific concept ideas and the application of suitable guidelines and methods, a suitable basis for a usable application to support clubs can be created.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	4
2.1	Begriffsklärung	4
2.2	Pfadfinder	5
2.3	Themenbezogene Arbeiten	8
2.4	Technologie	14
3	Gebrauchstauglichkeit	16
3.1	Dialoggestaltung: DIN EN NORM 9241-110	17
3.2	Qualitätskriterien: ISO/IEC 25000	18
3.3	Nielsen Heuristik	19
3.4	Vuolle	20
3.5	Richtlinien und Eigenschaften im Überblick	21
4	Funktionales Konzept	24
4.1	Personas	26
4.2	Funktionsaufgaben	28
4.3	Umsetzungskonzepte	29
4.4	Implementation	34
5	Oberflächengestaltung	36
5.1	Oberflächenelemente	36
5.2	Anwendung von Usability-Methoden	39
5.3	Page-Einteilung und Navigation	46
5.4	App-Seiten im Detail	48
6	Evaluation	56
6.1	Evaluationsziele	56
6.2	Testverfahren	58
6.3	Auswertung	60
6.4	Gesamtbild	70
7	Fazit	73
	Anhang	77
A	Große Abbildungen	77
B	Interview-Ergebnisse	80
C	Fragebogen	84
	Literaturverzeichnis	89

Abkürzungsverzeichnis

Kurzform:	Langform:
APP	oder Applikation; eng.: application; kurz: App; deu. lang.: Anwendungssoftware
BP	Robert Stephenson Smyth Baden-Powell of Gilwell (Gründer der Pfadfinder)
CUSO	Computer System Usability Questionnaire
DiBü	Diözesanbüro
Fab	Ionic Komponente: <code>IonFloatingActionButton</code>
FAQ	deu.: häufig gestellte Fragen; eng.: Frequently Asked Questions
GUI	deu.: Benutzeroberfläche; eng.: graphical user interface
iFrame	Inlineframe
ISO	deu.: Internationale Organisation für Normung; eng.: International Organization for Standardization
Orgateam	Organisationsteam
QUIS	Questionnaire for User Interface Satisfaction
REST	Representational State Transfer
ScOuT	Scouts Organisation Tool (In der Masterarbeit entwickelte Anwendung)
SUS	System Usability Scale
UCD-Prozess	User-Centered-Design-Prozess; oder auch: Human-Centered-Design-Prozess
UX	deu. Nutzungserlebnis; engl. user experience;

1 Einleitung

Unerfahrenheit, Überlastung, Traditionsverlust und Mitgliederschwund plagen heute viele Vereine, wie zum Beispiel die Pfadfinder. Diese Probleme hängen alle stark zusammen und sorgen für erhöhte Schwierigkeiten bei der Ausübung von Ämtern und der Planung von Veranstaltungen. Die Folge ist meist der Verlust von Wissen über Aufgaben und Planungsschritte und eine Überforderung der Mitglieder in ihren Rollen. Einige Vereine versuchen diese Wissenslücken zu füllen, doch das Ergebnis sind meist unübersichtliche, statische Dokumente, welche aufwendig zu erstellen und teilweise auch wenig hilfreich in der Anwendung sind.

Um diese Schwierigkeiten zu reduzieren und das Vereinsleben zu erleichtern, wurde eine Anwendung mit dem Namen Scouts Organisation Tool (kurz: ScOuT) konzipiert und entwickelt. ScOuT hat das Ziel die Organisationsverwaltung zu vereinfachen. Dabei werden auf Grundlage der ermittelten Probleme zwei Schwerpunkte gesetzt. Zum einen werden Vereinsmitglieder bei der Durchführung aktiv unterstützt. Dies erfolgt mittels automatischen Erinnerungen an von den Mitgliedern angelegten Aufgaben. Zum anderen generiert ScOuT mittels der angelegten Aufgaben eine Wissensbibliothek in Form von Templates, welche wiederverwendet werden können, um neuen Organisationsteams oder Amtsträgern zu helfen.

Doch die reine Konzeption und Implementation einer Idee zu einem Anwendungstool reicht nicht aus, um das Produkt für Nutzer gebrauchstauglich zu machen. Denn ein geeignetes Konzept zur Lösung eines Problems ist für den Nutzer nur so hilfreich und interessant wie es auf der Benutzeroberfläche abgebildet werden kann, da der Nutzer keinen direkten Bezug zu Methoden und Objektstrukturen hat. Er setzt die Anwendung folglich mit der Oberfläche gleich [BN13]. Eine schlechte Benutzeroberfläche beeinflusst die Bedienung der Anwendung durch den Nutzer. Eine nicht gebrauchstaugliche Oberfläche entsteht, wenn Informationen und Funktionen falsch, unerwartet oder fehlerhaft präsentiert werden. So kann die Anwendung zwar Funktionen anbieten, der Nutzer diese aber nicht anwenden, finden oder zuordnen. Ebenso kann die Benutzung der Funktionen zu umständlich, langwierig oder komplex sein. Eine weitere Folge ist, dass der Benutzer die Funktion aufgrund fehlerhafter Präsentation zum falschen Zweck verwendet und deshalb die Funktionen nicht das Ergebnis liefern, welches er erwartet. Für den Nutzer ist dann die Funktion selbst nicht gut genug, obwohl eigentlich die Oberflächengestaltung versagt hat. B. Böttcher und M. Nüttgens (siehe [BN13]) beschreiben es wie folgt: „[...] dass die bloße Verfügbarkeit von Funktionen nicht mehr hinreichend ist und der Gebrauchstauglichkeit sowohl aus ökonomischer als auch ergonomischer Perspektive eine wesentlich höhere Bedeutung beigemessen werden sollte.“ um der Oberfläche ihre Wichtigkeit zu geben. Gleichzeitig sprechen sie als Lösung die Gebrauchstauglichkeit an. Durch gemachte negative Erfahrungen sinkt das Ansehen der Anwendung

in den Augen des Nutzers, was dazu führt, dass die Anwendung nicht mehr verwendet wird. Die Einhaltung von Ergonomie wirkt dagegen positiv auf den Nutzer und verbessert somit den Werbe-Wert [SR91]. Auch Nielsen beschreibt die Anwendung von Usability-Methoden als erhebliche Verbesserung des Produktes [Nie94]. Die Verschlechterung der Anwendung, lässt sich in einigen Zahlen wiedergeben. Zehn Prozent der Zeit am Computer werden von Nutzern darauf verwendet Probleme bei der Bedienung zu lösen, weil die Gebrauchstauglichkeit nicht bei der Implementierung der Anwendung einbezogen wurde [Ric97]. Deshalb empfinden rund 57% der Benutzer die Bedienung von Geräten als zu kompliziert [Ham06]. Dies hat zur Folge, dass in Amerika jedes dritte Computersystem für Büroarbeiten abgelehnt wurde. Wird dennoch ein neues System eingeführt, steigt der Prozentsatz der krankgemeldeten Mitarbeiter um 300% an. Doch nicht nur der Mensch selbst ist von der mangelhaften Befolgung des Prinzips der Gebrauchstauglichkeit betroffen. Durch falsche Bedienung entstehen jährlich Wirtschaftsschäden im Wert von circa einer Milliarden Euro in Deutschland und 30 Milliarden Dollar in den USA.

Die Gebrauchstauglichkeit, welche als Lösung für diese Probleme gilt, ist ein von der Internationale Organisation für Normung definierter Begriff, der das Ziel verfolgt die Verwendung von Anwendungen durch eine angepasste Benutzeroberfläche für den Nutzer möglichst effizient, effektiv und zufriedenstellend zu gestalten [DIN98]. Deshalb ist das Ziel dieser Forschung die Anwendung ScOuT zum Komfort des Nutzers nach verschiedenen Richtlinien zu entwickeln, um ein möglichst hohes Maß an Gebrauchstauglichkeit zu erreichen. Dadurch wird eine Verbesserung der Anwendung zur Unterstützung der Arbeit der Vereinsmitglieder sowohl auf struktureller Ebene als auch auf der Präsentationsebene angestrebt.

Die dafür gewählten Richtlinien der Dialoggestaltung, Qualitätskriterien, Nielsen Heuristik und die Dimensionen von Vuolle dienen dabei als Teilziele zur Erreichung der Gebrauchstauglichkeit. Zur Umsetzung dieser werden verschiedene Methoden angewendet. Darunter die Integration der potentiellen Anwender durch ein Interview von fünf Pfadfindergruppen zu Beginn der Konzeption. Ebenfalls wurden Personas und Funktionsaufgaben bei der Konzeption ermittelt, um die Gebrauchstauglichkeit bereits vor dem Design der Benutzeroberfläche einzubeziehen [Usa19], [Ric97]. Weiter blieb der Bezug zum Nutzer durch die Methode des Lauten Denkens während der Entwicklung der Benutzeroberfläche bestehen [BN13]. Die grobe und feine Einteilung der Oberfläche in App-Seiten und ihre Bausteine erfolgte nach der Untergliederung von Funktion und Information als ein Großteil der Forschungsarbeit an der Benutzeroberfläche [Ham06]. Die reine Anwendung der Richtlinien in Bezug auf Hilfe, Feedback, Fehlen, Terminologie und Farbe erfolgte ebenfalls. Zum Abschluss wurde erneut der Benutzer hinzugezogen um in einer Remote-Mobile-Umfrage als Maß zu dienen, die Erreichung der Gebrauchstauglichkeit zu prüfen [Ham06], [BN13].

Um nun diesen Prozess nachvollziehen zu können, beginnt die Ausarbeitung in Kapitel 2 mit den Grundlagen zur Darstellung der Pfadfinder und der dort bestehenden Problematik in der Vereinsarbeit. Um ein Verständnis über den Ist-Zustand zu erhalten, werden darauf aufbauend weiterführende Planungstools betrachtet, welcher durch bereits existierende Anwendungen gegeben ist. Die letzte Grundlage für die Konzeption von ScOuT ist die Wahl der Technologien und der Plattformen. Sind die Grundlagen für die Struktur und Funktionen abgeschlossen, geht Kapitel 3 tiefer auf die Gebrauchstauglichkeit, sowie ihre Teilziele ein.

Nach der Darbringung aller nötigen Grundlagen beginnt der Hauptteil mit Kapitel 4. Dort werden zunächst anhand des Interviews die Anforderungen an ScOuT erhoben. Anschließend wird ein Ausschnitt der Konzeption besprochen. Dieser Ausschnitt wird für den Rahmen der Masterarbeit auf die Kernideen gekürzt. In Kapitel 5, zweites Kapitel des Hauptteils, geht es um die Gestaltung der Oberfläche im Sinne der Gebrauchstauglichkeit. Dort beginnt damit das zuvor gewählte Framework für das Frontend bezüglich seiner Oberflächenelemente zu untersuchen. Daraufhin werden Methoden und Konventionen detailliert beschrieben, welche im Allgemeinen bei der Gestaltung der Benutzeroberfläche angewendet werden. Die Unterteilung in App-Seiten und ihre einzelnen Anordnungen schließen dieses Kapitel ab. Der Hauptteil endet mit Kapitel 6 über die Erstellung, Durchführung und Auswertung der Benutzerumfrage.

Abschließend wird im Schlussteil der Ausarbeitung eine Zusammenfassung der Durchführung und der Ergebnisse gegeben.

2 Grundlagen

Im Anschluss an die folgende Begriffsklärung wird sich zunächst mit der Problemstellung beschäftigt. Hierfür wird auf die Pfadfinder, im Speziellen die Gruppierung der Deutschen Pfadfinderschaft Sankt Georg (kurz: DPSG), eingegangen und durch die in der Vereinsarbeit auftretenden Probleme das Ziel der geplanten Anwendung motiviert. Daraufhin werden weiterführende Arbeiten dahingehend untersucht, ob sie Lösung und Teillösung für die genannten Probleme darstellen und wie aktuelle Standards zur Aufgabenverwaltung in bestehenden Tools aussehen. Wenn der Ist-Zustand des Problems und der aktuellen Lösungsansätze abgeschlossen ist, wird mit den aktuellen, technischen Möglichkeiten zur Umsetzung von ScOuT fortgefahren.

2.1 Begriffsklärung

Zunächst die Definition einiger Begriffe, welche in der Arbeit immer wieder verwendet werden.

Veranstaltung

Veranstaltungen sind Aktionen, Projekte, Events oder Ähnliche, welche zu einem bestimmten Zeitpunkt stattfinden und von Organisationsteams mittels Aufgabenlisten geplant werden.

Organisationsteam

Ein Organisationsteam (kurz: Orgateam) ist dafür verantwortlich eine gemeinsame Aufgabenliste zu erstellen und zu erledigen, um ein Ziel, wie eine Veranstaltung, zu planen.

Planung

Nach eigener Definition ist die Planung einer Veranstaltung in drei Abschnitte zu unterteilen: Vorbereitung, Durchführung, Nachbereitung. Die Vorbereitung selbst ist noch einmal zu untergliedern in die Definition und die Ausführung der Vorbereitungsschritte. Ein Vorbereitungsschritt, im Folgenden Aufgabe genannt, kann sowohl ganz zu Beginn der Eventplanung definiert werden als auch noch im späteren Verlauf der Vorbereitung.

Amt

Ein Amt entspricht einer Rolle, welche eine Person innerhalb eines Vereins inne hat. Amtsträger ist dabei die Bezeichnung für die Person und Amtsaufgaben sind die zu erledigenden Schritte zum Ausüben des Amtes und zum Erreichen dessen Zielen.

Aufgabe

Aufgaben sind Vorbereitungsschritte, welche ausgeführt werden müssen, um eine Veranstaltung zu planen oder ein Amt auszuüben.

Planungstool

Oder auch Organisationsverwaltungstool. Diese Tools sind Anwendungen (oder Applikation; eng.: application; kurz: App; deu. lang.: Anwendungssoftware), welche dem Nutzer dabei helfen Aufgaben zu verwalten und Veranstaltungen zu planen.

2.2 Pfadfinder

Die Pfadfinder (engl.: scout) sind ein Jugendverband, welcher 1907 von Robert Stephenson Smyth Baden-Powell of Gilwell (kurz: BP) in England als eine international, religiös und politisch unabhängige Erziehungsbewegung gegründet wurde. Auf Grundlage dieses Ursprungs entstanden weltweit viele verschiedene Verbände, die sich der Jugendarbeit und Erziehung widmen, darunter hier in Deutschland unter anderem die DPSG, welcher sich der katholischen Kirche zugehörig fühlt. Die DPSG, gegründet 1929, ist mit 95.000 Mitgliedern der größte Jugendverband in Deutschland (Stand 2019) [Ass07][Aut19][Mit19].

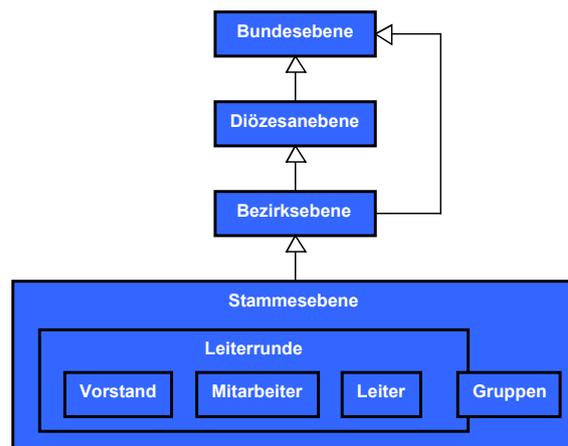


Abbildung 1: Struktur der DPSG

Die Struktur der DPSG unterteilt sich in drei bis vier Ebenen (vergleiche Abbildung 1). Auf oberster Ebene ist die Bundesebene. Die DPSG ist ein katholischer Pfadfinderverband, weshalb die Unterteilung der Bundesebene in die Diözesanebene auf Grundlage der katholischen Amtsbezirke erfolgt. Einige Diözesen unterteilen sich anschließend in Bezirke, welche sich wiederum in Stämme untergliedern, andere zerlegen sich direkt in die Stammesebene [Aut19][Mit19].

Die Stämme bilden dabei die Vereine vor Ort und leisten innerhalb der Struktur eines Stammes den Hauptanteil an der Jugendarbeit. Die innere Struktur basiert dabei auf den verschiedenen Stufen, welche sich nach dem

Alter gliedern. Es gibt die Stufen Biber (ab 4 Jahre), Wölflinge (ab 6 Jahre), Jungpfadfinder (ab 9 Jahre), Pfadfinder (ab 12 Jahre) und Rover (ab 15 Jahre). Innerhalb einer Stufe hat ein Stamm eine oder mehrere Gruppen, welche wöchentliche Gruppenstunden von eineinhalb Stunden unter Aufsicht der Gruppenleiter ausführen [Aut19][Mit19]. Doch neben den wöchentlichen Gruppenstunden bietet ein Stamm weitere Aktionen (= Veranstaltung der Pfadfinder) an, welche es zu organisieren gilt. Hervorzuheben wäre beispielsweise das Zeltlager, welches meist der Höhepunkt eines Pfadfinderjahres ist und einen besonders hohen Aufwand an Vorbereitung benötigt. Aber auch kleinere Aktionen, wie Geländespiele, Wanderungen, Stände auf Stadt- oder Gemeindefesten und viele mehr, benötigen ihre Planung. Des Weiteren gibt es Aktionen, die von der Bundesebene und der Diözesanebene angeboten werden. Darunter fallen sowohl Aktionen für den ganzen Stamm als auch verwaltungstechnische Treffen und Schulungen. Auch eine Teilnahme an einer solchen Aktion muss im Stamm geplant werden. Um all das zu regeln, gibt es in den Stämmen nicht nur die Gruppenleiter, sondern auch Vorstände und eine Vielzahl unterschiedlicher Ämter und Teams zur Organisation (kurz. Orgateams). Auch die Jugendlichen selbst sind oft Teil dieser Orgateams und übernehmen bereits früh Verantwortung für die Durchführung von Gruppenstunden und Aktionen.

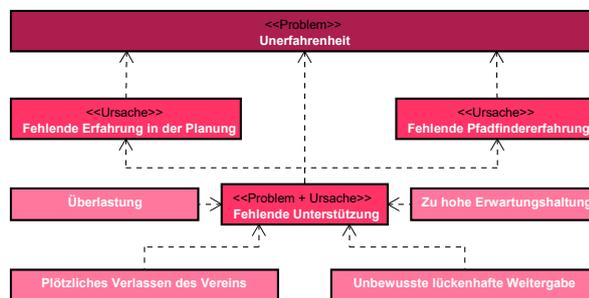


Abbildung 2: Übersicht über Probleme und deren Ursachen im Stammesleben

Obwohl es die Pfadfinder und auch die DPSG schon lange gibt, treten immer wieder Komplikationen im Stammesleben auf. Dabei stellen, nach langjähriger Beobachtung und Eigenerfahrung der Arbeit im Stamm, die unerfahrenen Orgateams und Amtsträger ein Hauptproblem dar. Eine Übersicht über die Probleme und Ursachen ist in Abbildung 2 zu sehen, welche im Folgenden genauer erläutert werden.

Die genannte Unerfahrenheit kann viele Ursachen haben. Obwohl Pfadfinder sich mit der Jugendarbeit beschäftigen, ist ein Einstieg in jedem Alter möglich. Mitglieder, die erst später dazustoßen, erfahren jedoch nicht die gemeinsame Entwicklung mit dem Stamm wie Mitglieder, die bereits länger involviert sind. Traditionen und regelmäßige Aktionen sind nicht ausreichend

bekannt. Diese fehlende Eigenerfahrung kann die Arbeit als Orgateam oder in einem Amt deutlich erschweren. Hinzu kommen noch weitere Ursachen für die Unerfahrenheit. Pfadfinder können bereits in sehr jungen Jahren anfangen. Durch die Struktur des Jugendverbandes dürfen auch Jugendliche früh Verantwortung für die Planung von Aktionen übernehmen. Diesen Orgateams mangelt es dann nicht an Erfahrung in Bezug auf das Pfadfinderleben, jedoch bringen sie meist wenig Erfahrung in Bezug auf die Planung als solche mit sich und das Ausmaß der notwendigen Vorbereitung wird häufig unterschätzt. Doch auch langjährige Pfadfinder übernehmen gelegentlich die Planung von Aktionen oder Ämtern, die sie immer nur als Teilnehmer oder Beobachter kannten. Ihnen ergeht es ähnlich wie den jüngeren Mitgliedern, auch wenn sie den Vorteil haben, Erfahrung von anderen Planungen und Ämtern mitzubringen. Doch nicht immer sind die Überschneidungen der Aufgaben groß genug, um das Amt oder die Planung direkt zu Beginn sinnvoll umzusetzen.

Obwohl das Problem der Unerfahrenheit immer wieder auftritt, werden unerfahrene Orgateams weiterhin unzureichend von anderen Mitgliedern des Stammes unterstützt. Auch dies hat verschiedene Ursachen. Eine davon ist die Überlastung der Mitglieder. Sie erledigen ihre Aufgaben nicht nur ehrenamtlich neben dem Berufsleben, sondern haben oft eine Häufung von Ämtern und Aufgaben aufgrund geringer oder insgesamt sinkender Mitgliederzahlen. Darunter leidet oft die Kommunikation. Eine weitere Ursache für die mangelhafte Weitergabe von Informationen ist der plötzliche Kommunikationsabbruch. Dies passiert, wenn ein Mitglied plötzlich den Stamm verlässt, sei es wegen eines Umzugs, Krankheiten oder eines Jobwechsels. Oft wird es in diesen Fällen versäumt, Informationen zur Planung einer Aktion oder Ausführung eines Amtes weiterzureichen. In anderen Fällen ist die Erwartungshaltung der erfahrenen Mitgliedern zu hoch. Sie gehen davon aus, dass als langjähriges Mitglied alles bekannt ist. Dabei gehen sie von Wissen aus, welches nicht vorhanden ist und versäumen es diese Lücken zu schließen.

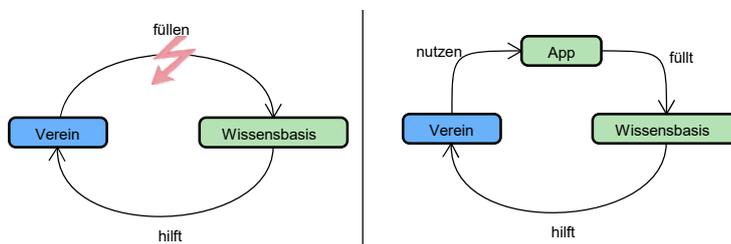


Abbildung 3: App als Lösung für Probleme

Die Idee ist es nun, diese Probleme im Stammesleben zu reduzieren, indem eine App (ScOuT) entwickelt wird, welche die Aufgaben in Orgateams

und Ämtern unterstützt. Dabei sollen Mitglieder ihre Erfahrung an andere übermitteln, um unerfahrenen Orgateams und Amtsträgern besser bei ihren neuen Aufgaben zu helfen. Es reicht jedoch nicht eine App zu konzipieren, welche eine reine Anhäufung von Wissen darstellt, denn das Füllen dieser Wissensbibliothek ist mit den selben Problemen belastet wie sonstige Wege zur Weitergabe von Informationen (siehe Abbildung 3 links). Zudem bereitet es nur zusätzliche Arbeit die Wissensbasis zu füllen. Deshalb sollte der Fokus für den Nutzer von der reinen Weitergabe und dem monotonen Aufschreiben von Informationen, Planungsschritten und Amtsaufgaben umgelenkt werden auf eine aktivere und direktere Nutzung. Es soll ein Nutzungsmehrwert für denjenigen entstehen, welcher die Wissensbasis füllt. Dies erfolgt mittels aktiver Unterstützung durch ScOuT. Zu diesem Zweck Hilft ScOuT dem Nutzer durch Erinnerungen. So findet, wie im rechten Teil der Abbildung 3 dargestellt ein Zwischenschritt zur Befüllung der Bibliothek statt, welcher den Eintragenden motivieren soll. Die eben genannte Wissensbibliothek kann dann durch sogenannte Templates automatisch in ScOuT erstellt werden. Templates können aus den von erfahrenen Nutzern angelegten Planungsschritten bestehen und von weniger erfahrene Nutzer wiederverwendet werden.

2.3 Themenbezogene Arbeiten

In der Welt der Applikationen gibt es eine Vielzahl von Tools zum Verwalten und Planen von Aufgaben oder Veranstaltungen, wodurch die Arbeit im Verein vereinfacht und die in Abschnitt 2.2 genannten Probleme gelöst werden könnten. Für die Problemlösung der Schwierigkeit im Pfadfinderalltag werden nun einige Anwendungen betrachtet und analysiert. Darunter fallen sowohl Anwendungen, welche den gesamten Prozess optimieren sollen, als auch solche, die eine konkretere Zielsetzung verfolgen.

Doodle ¹

Mit Doodle werden Termine abgeglichen, um für Treffen die beste Terminwahl vereinbaren zu können. Doodle bietet zudem die besondere Option, das Ergebnis der Umfrage für den Nutzer vorab zu analysieren. So werden bereits Termine hervorgehoben, an denen die meisten teilnehmen können.

Kommunikationswege

z.B. Slack², Discord³, Teamspeak⁴

Diese Anwendungen bieten Möglichkeiten in themenbezogenen Chats zu kommunizieren oder online in Gruppen zu telefonieren.

¹<https://doodle.com/de/>

²<https://slack.com/>

³<https://discordapp.com/>

⁴<https://teamspeak.com>

CloudService

z.B.: OwnCloud⁵, Dropbox⁶, GoogleDrive⁷

Mit CloudServices können Dateien, wie Dokumente, Tabellen und weitere Daten online verfügbar gehalten werden, sodass mehrere Personen darauf zugreifen können. GoogleDrive bietet neben der Speicherung die Möglichkeit diese Dokumente auch zu öffnen und zu bearbeiten, wodurch keine zusätzliche Anwendung benötigt wird, um in den Dateien zu arbeiten.

MindMeister ⁸

Diese Anwendung wird beim Brainstorming verwendet und dient der Erstellung von MindMaps.

Toggle ⁹

Mit Toggle kann die Zeit erfasst werden, die zur Erledigung einer Aufgabe benötigt wurde. So ist es möglich den eigenen Aufwand einfacher zu messen und zu reflektieren.

Diese Anwendungen mit konkreten Aufgabenzielen entsprechen nur teilweise dem, was zur Unterstützung der Amtsaufgaben oder der Veranstaltungsplanung benötigt wird. Doodle kann durchaus dazu verwendet werden, um den Termin der Veranstaltung, die Treffen der Orgateams oder Termine für den Amtsträger zeitlich zu vereinbaren. Wer im Team plant, benötigt gut funktionierende Kommunikationskanäle für den Austausch untereinander. Vieles wird bei Organisationstreffen besprochen, doch oft fällt etwas an, was nicht bis zum nächsten Treffen warten kann. Chats und Online-Telefonie zählen deshalb mittlerweile zu den wichtigen Kommunikationswegen für solche Teams. Auch CloudServices sind generell beim Planen weit verbreitet, da auf diesem Weg alle möglichen Dokumente, Bilder und vieles mehr miteinander geteilt werden können. Hier würde sogar die Möglichkeit bestehen die Planung „per Hand“ in den Dokumenten zum Beispiel in Form von Protokollen festzuhalten. Doch diese Art der Veranstaltungsplanung erweist sich als äußerst umständlich. Auch MindMaster unterstützt die Planung mit hilfreichen Funktionen. So können die MindMaps für Reflexionen, die Gestaltung vom Programmablauf als auch für die Vorbereitung der Planung von verschiedenen Aufgaben verwendet werden. Für all diese Möglichkeiten bieten Tools mit dem Hauptziel der Veranstaltungsplanung jedoch bessere Alternativen. Lediglich Toogle wird seine Anwendung im Bereich von Firmen zur Einteilung der Arbeit der Angestellten finden und weniger in der Vereinsarbeit, da das Tracken der Zeit von einem einzelnen Vorbereitungsschritt für die Vereinsmitglieder wenig relevant ist. Obwohl diese Anwendungen also

⁵<https://owncloud.com/de/>

⁶<https://dropbox.com/>

⁷<https://google.de/drive/apps.html>

⁸<https://mindmeister.com/de>

⁹<https://toggl.com/>

durchaus ihre Möglichkeiten bieten, um die Planung von Veranstaltungen zu erleichtern, sind sie, jede Einzelne für sich, jedoch weniger hilfreich und bieten nicht die erwarteten Funktionen für die Unterstützung der Vereine. Einige der Anwendungen mit konkreten Aufgabenzielen werden deshalb in Tools zur Veranstaltungsplanung integriert.

Als Grundlage zum Planen einzelner Aufgaben oder einer Veranstaltung dienen die Vorbereitungsschritte. Dabei gilt es, die gesamte Planung der Veranstaltung in kleine Aufgaben zu unterteilen. Eine solche Aufgabe kann zum Beispiel die Terminvereinbarung eines Orgateamtreffens sein, wofür Doodle verwendet werden könnte. Eine weitere mögliche Aufgabe könnte die Ablaufplanung des Programms sein, was mit Doodle wiederum nicht möglich ist. Die Anwendung zum Planen von Veranstaltungen sollte sich also nicht an konkreten Aufgaben und vor allem deren Umsetzung aufhalten, sondern sich damit befassen die Menge aller Aufgaben zu verwalten. So bieten Anwendungen wie Trello¹⁰, ClickUp¹¹, Asana¹², ToDoIst¹³, Wunderlist¹⁴ und InLoox¹⁵ genau solche Funktionen. Die Unterstützung für Amtsaufgaben, welche nicht an eine Veranstaltung gebunden sind, werden dadurch hinreichend abgedeckt. Doch für die Planung von Veranstaltungen selbst müssen die Tools weiter geprüft werden. In all diesen Werkzeuge können Veranstaltungen abgebildet werden, sei es nun durch Boards, Spaces, Projekte oder Listen. Auch die Abbildung von Aufgaben wird in all diesen Anwendungen angeboten. Bei den meisten heißen sie Task, nur bei Trello werden sie Karten genannt. Die Erstellung von Aufgaben ist jedoch nur eine minimalistische Grundlage, da dies auch mit einfachen Dokumenten in einem CloudService oder dergleichen erreicht werden kann. Zur Differenzierung von den minimalistischen Funktionen bieten Planungstools die grundlegende Funktion, den Nutzer an jede Aufgabe zu erinnern. Diese Erinnerungen werden mittels konkreten Fristen eingerichtet und der Nutzer erhält Benachrichtigungen darüber, ob er aktuell etwas vorzubereiten hat. Die Planung wird durch diese Funktion ungemein erleichtert. Der Planende muss sich nicht mehr jeden Tag damit befassen, ob und was genau jetzt dringend ist. Neben der Frist können sowohl Aufgaben, als auch Veranstaltungen in den meisten Anwendungen optional durch zusätzliche Funktionen erweitert werden. Zu den Erweiterungen zählen Projektbeschreibungen, Dateianhänge, Kalender, Verknüpfungen mit Online-Videos sowie die Funktionen der zuvor genannten Anwendungen mit konkreten Aufgabenzielen und vieles mehr. Eine besondere Erweiterung ist die Checkliste. Eine Aufgabe kann zum Beispiel aus mehreren kleinen Schritten bestehen, welche zusammenhängen und keine eigene Aufgabe benötigen.

¹⁰<https://trello.com/>

¹¹<https://clickup.com/>

¹²<https://asana.com/de>

¹³<https://de.todoist.com/>

¹⁴<https://wunderlist.com/de/>

¹⁵<https://inloox.de/>

Des Weiteren können sie ebenfalls eine Packliste oder eine Einkaufsliste sein. Dieser besondere Typ von Aufgaben kann ohne Schwierigkeiten mittels einer Checkliste abgebildet werden. Da diese Erweiterung wohl die häufigste Verwendung findet, wird sie nahezu bei allen Anwendungen zum Planen von Veranstaltungen angeboten. Aus diesen bereits existierenden Grundlagen ergibt sich der in Abbildung 4 dargestellte Aufbau, welcher die verschiedenen Anwendungen zusammenfasst.

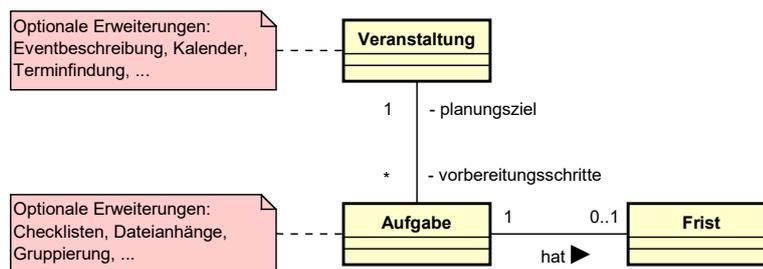


Abbildung 4: Zusammenfassendes Metamodell für Grundlagen bereits existierender Anwendungen zum Planen von Aufgaben und Veranstaltungen

Trotz der gemeinsamen Grundlage unterscheiden sich die Anwendungen durch optionale Erweiterungen, verschiedenen Funktionen und ihre Benutzeroberflächen. Im Folgenden werden einige Beispiele für die Unterschiede genannt und analysiert.

Trello

Trello bietet die Möglichkeit die Aufgaben (dort Karten genannt) in Listen zu gruppieren. Dies könnte für größere Orgateams den Vorteil bieten, dass die Organisation in Subgruppen unterteilt werden kann, wie zum Beispiel ein Küchenteam und ein Logistikteam. Der Nutzer hat zudem die Möglichkeit die Listen untereinander, die Karten innerhalb einer Liste, sowie Karten von einer Liste in eine andere zu verschieben. Des Weiteren ist der Informationsfluss in Trello sehr umfassend. Die dort integrierten Benachrichtigungen in Form von einem Badge geben dem Benutzer Hinweise bei jeglichen Änderungen in den Karten, Listen und Boards durch andere Nutzer. Badges sind dabei kleine, meist rote Zahlen, welche gut sichtbar an Buttons angebracht werden, um dem Nutzer mitzuteilen, dass hinter diesem Button Neuigkeiten, Aufgaben oder sonstige Veränderungen auf ihn warten. Dadurch ist auf einen Blick ersichtlich, ob sich etwas in der Organisation geändert hat oder Aufgaben bereits erledigt sind. In den Benachrichtigungen werden jedoch ausschließlich Aktivitäten durch Nutzer dargestellt. Erinnerungen an eine ablaufende Frist werden nicht in dem Badge angegeben. Die Frist wird lediglich deutlich auf den Karten angezeigt, jedoch muss der Nutzer manuell die Veranstaltung (dort Board genannt) öffnen, um diesen Hinweis sehen zu

können. Um Fristen nicht zu verpassen, müssen also regelmäßig alle Boards geöffnet und manuell überprüft werden, was bereits bei wenigen Veranstaltungen einen erhöhten Aufwand darstellt. Dieser Aufwand könnte minimiert werden, indem die Karten innerhalb der Listen durch Trello automatisch nach den Fristen sortiert werden, sodass dringende Aufgaben optisch schneller erfasst werden können. Jedoch müssen alle Karten und Listen ebenfalls vom Nutzer per Hand verschoben werden. Ebenso fehlt eine automatische Sortierung für Karten mit abgelaufenen Fristen oder erledigten Checklisten. Der Nutzer hat lediglich die Möglichkeit selbst Listen anzulegen, welche unerledigte und erledigte Karten enthalten und die Karten manuell nach der Erledigung zu verschieben, um den Überblick zu optimieren. Bei einer wie bereits erklärten Einteilung der Listen für Suborgateams würde dies das Board jedoch überfüllen.

Der Nutzer hat zudem in Trello die Möglichkeit Boards zu kopieren. Möchte der Nutzer nach dem selben Ablauf eine Veranstaltung mehrmals planen, so kann er dies problemlos mit einer Kopie machen. Jedoch müssen manuell alle Fristen neu gesetzt und bei der Nutzung von Erledigt- und Unerledigt-Listen die Karten verschoben werden. Das stellt einen erhöhten Zeitaufwand für den Nutzer dar.

ClickUp

Wie auch bei Trello können die Aufgaben (dort Task genannt) in ClickUp in verschiedene Listen sortiert und diese Tasks zwischen den Listen verschoben werden. Ebenso werden die Listen auch in ClickUp ausschließlich vom Nutzer sortiert. Das Tool zeigt Aufgaben mit abgelaufenen oder bald anstehenden Fristen (dort Due Date) nicht automatisch sortiert ganz oben an. Jedoch gibt es Filterfunktionen, welche manuell vom Nutzer aktiviert werden können. Eine simplere Variante, dringende Fristen einzusehen, bringen die drei verschiedenen Darstellungsmöglichkeiten der Veranstaltung (dort Projekt genannt) mit sich. Die Kalender-Ansicht ermöglicht es innerhalb eines Kalenders alle Task einzusehen, die eine Frist haben. Die Listen-Ansicht ermöglicht es dem Nutzer die Task sortiert entsprechend der zugehörigen Listen zu betrachten und die Board-Ansicht zeigt alle Tasks sortiert nach unerledigt (dort Open) und erledigt (dort Closed). Die letztere, automatisierte Ansicht ist ein Vorteil gegenüber Trello, bei welchem diese Sortierung per Hand erfolgen muss.

Während Trello mit seinen Karten nicht ausschließlich das konkrete Ziel verfolgt Aufgaben abzubilden, sind die Tasks in ClickUp deutlich stärker darauf bezogen. Dies zeigt sich insbesondere an der großen Fülle an optionalen Erweiterungen für die Aufgaben. Dazu zählt beispielsweise die sich regelmäßig wiederholende Aufgabe. Diese Einstellung ermöglicht es, dass der Tasks nach seiner Erledigung nach einem festen Zeitraum, wie zum Beispiel einer Woche, erneut auf unerledigt gesetzt wird. Dies ist besonders praktisch für Aufgaben, die wiederkehren, wie die Vorbereitung eines wöchentlichen Meetings für die Arbeit oder das Gießen der Pflanzen, welche privat zu Hause

immer wieder vergessen werden. Auch bei den Pfadfindern gibt es ab und an wiederkehrende Einzelaufgaben, wie die des Rasenmähens oder andere pflegende Aufgaben rund um ein eigenes Pfadfinderhaus. Für die Planung von Veranstaltungen ist diese Funktion jedoch wenig hilfreich, da es dort kaum Verwendungszwecke dafür gibt. Eine weitere Option sind die Subtasks, welche Unterschritte von Aufgaben sein können. Sie können an eine Aufgabe angehängt werden und bieten ähnliche viele optionale Möglichkeiten wie die normalen Tasks. Mehrschrittige Aufgaben sind immer wieder in der Planung von Veranstaltungen zu finden, weshalb diese optionale Erweiterung in ClickUp eine Bereicherung ist.

Eine für die Ziele von ScOuT interessante Funktion von ClickUp sind die Templates. Aus jedem Projekt kann ein Template erstellt werden und aus diesem Template wiederum eine Kopie des ursprünglichen Projektes. Dies entspricht in etwa der Funktion in Trello ein Board zu kopieren. Obwohl ClickUp die Funktion bietet, alle erledigten Task automatisch wieder als unerledigt zu markieren, ist diese nicht standardmäßig aktiviert und muss vom Nutzer für jedes zu erstellende Projekt einzeln ausgewählt werden. Auch Fristen werden standardmäßig mit ihrem konkreten Datum übernommen und der Nutzer muss sich aktiv dazu entscheiden, diese mit einer Abbildungsfunktion anzupassen. Für die Fristen bedeutet dies konkret, dass ein Starttermin beim Erstellen der Veranstaltung ausgewählt wird, auf welchen die früheste Frist abgebildet wird und alle späteren Fristen in der selben zeitlichen Relation wie im Template angepasst werden.

Asana

Asana ist in vielerlei Hinsicht ähnlich wie ClickUp. Dort gibt es ebenfalls Fristen (dort Fälligkeitsdatum genannt), die drei Darstellungsmöglichkeiten der Veranstaltungen (dort Projekt genannt) und die Möglichkeit die Aufgaben in Listen (dort Abschnitt genannt) zu gruppieren. Die Board-Ansicht unterscheidet sich jedoch dahingehend, dass es der Standard-Ansicht von Trello ähnelt und die Aufgaben nicht nach erledigt und unerledigt sortiert werden, denn Aufgaben verschwinden bei Asana, anders als bei Trello und ClickUp, wenn sie erledigt sind. Erst durch einen vom Nutzer eingestellten Filter werden erledigte Aufgaben wieder angezeigt. Wie bei ClickUp und Trello werden auch hier die Listen nicht automatisch sortiert, sodass der Nutzer nicht sofort sieht, wann ein Fälligkeitsdatum anfällt. Auch hier gibt es wieder Filter-Funktionen, welche das Sortieren nach Fristen ermöglichen. Anders als bei ClickUp können die eingestellten Filter bei Asana gespeichert werden, sodass eine dauerhafte Sortierung nach den Fristen möglich ist.

Wie in den anderen beiden Anwendungen gibt es auch hier eine Funktion, um von einem Projekt ein Duplikat anzulegen. Dabei können Fälligkeitsdaten mit übernommen werden. Anders als bei ClickUp gibt es keine Möglichkeit die Fristen automatisch anpassen zu lassen und auch bei Asana bleiben erledigte Aufgaben weiterhin in der Kopie erledigt.

Ähnlich dieser drei Beispiele sind die meisten auf dem Markt verfügbaren Anwendungen zur Planung und Organisation entwickelt. Obwohl diese Grundlagen bereits viele Funktionalitäten bieten, entsprechen die Funktionen nicht ganz dem, was Experten zum Planen von Veranstaltungen benötigen und kleinere Vereine sowie Dachverbände warten weiterhin auf Tools, welche den speziellen Anforderungen genügen.

2.4 Technologie

Um das Planungstool ScOuT nun mit der passenden Technologie umzusetzen, sodass die Organisation von Vereinen und ihren Veranstaltungen technisch erleichtert wird, müssen verschiedene Aspekte in Betracht gezogen werden. Wichtig ist dabei, dass alle Mitglieder eines Organisationsteams Zugriff auf die Technologie haben, da sonst unnötige zusätzliche Kommunikation entsteht und die Verteilung der Aufgaben schneller zu ungleichen Anteilen erfolgt. Deshalb wird sowohl die mobile Variante am Mobiltelefon als auch die Verwendung von ScOuT am Computer geprüft.

Nach den Statistiken von <https://de.statista.com/> beträgt der Marktanteil von Android in den letzten Jahren um die 80% und von iOS um die 20% [Ten19]. Doch auch andere Plattformen wie Windows Phone und andere Mobiltelefone sind in Deutschland vertreten. Ebenso gibt es noch Personen, gerade bei den Pfadfindern, die kein Mobiltelefon benutzen und somit auf eine Version der Technologie am Computer angewiesen sind. Da die Organisation nicht nur von einer Person, sondern von Organisationsteams abhängig ist, sollten möglichst viele der Plattformen abgedeckt sein. Deshalb reicht es nicht eine Native App zu erstellen, welche nur eine der Plattformen abdecken würde. Die Wahl bleibt folglich zwischen einer Web-App und einer hybriden App. Beide Technologien funktionieren auf allen Plattformen, da sie ihre Dienste über Webservices anbieten. Der Unterschied liegt darin, dass hybride Apps Schnittstellen zu den Endgeräten selbst anbieten, zum Beispiel auf Kontakte, das GPS und weitere Funktionen zugegriffen werden kann, wohingegen Web-Applikationen diese Möglichkeit nicht aufweisen. Um möglichst viele Plattformen abzudecken und den Zugriff auf die Funktionalitäten des Endgerätes nicht von vornherein wegfallen zu lassen, wird für die Entwicklung von ScOuT auf das Konzept von hybriden Apps zurückgegriffen.

Zur Umsetzung einer hybriden App gibt es verschiedene Frontend- und Backend-Frameworks, die getrennt oder in Kombination verwenden werden können. Ionic¹⁶, ein OpenSource Framework, welches zur Implementierung von Frontends genutzt wird, erweitert die Möglichkeiten von Angular¹⁷, einem TypeScript-basierten Frontend-Web-Applikationsframework,

¹⁶<https://ionicframework.com/>

¹⁷<https://angular.io/>

sowie ApacheCordova¹⁸, welches ein Framework zur Erstellung hybrider Applikationen für mobile Endgeräte ist. Ionic konzentriert sich dabei auf die Elemente, welche für die Gestaltung der Benutzeroberfläche verwendet werden und bietet eine Vielzahl vordefinierter Ionic-Tags, welche wie in HTML verwendet werden können. Aufgrund dessen, dass Ionic keine Funktionen zur Gestaltung des Backends bietet, wird hierfür eine eigenständige Technologie benötigt. Das Backend wird mit NodeJS¹⁹ umgesetzt. Zu Beginn der Entwicklung wurde das Framework Laravel²⁰ genutzt, welches auf NodeJS basiert und dieses erweitert. Aufgrund der Komplexität der Installation und in der Anwendung wurde es frühzeitig gegen das simplere NodeJS ausgetauscht. Neben dem Frontend und dem Backend wird zudem eine Datenbank zur vollständigen Entwicklung einer hybriden App benötigt. Hierfür wurde MongoDB²¹, eine dokumentorientierte strukturierte Datenbank, verwendet. Diese Form der Datenbank unterscheidet sich dahingehend von den traditionellen relationalen Datenbanken, dass die zu speichernden Daten nicht mehr in einer tabellarischen Form gespeichert werden, sondern in Dokumenten, welche eine Ansammlung (engl. Collection) von Objekten beinhalten.

Diese drei Komponenten müssen nun miteinander agieren, um ein Gesamtprogramm darzustellen. Frontend und Backend erreichen dies mittels Representational State Transfer (kurz: REST). REST ist ein Programmierparadigma für verteilte Systeme und wird bei Web-Services, wie Ionic und NodeJS, als Schnittstelle für die Kommunikation verwendet [FT00]. Dabei bietet NodeJS REST-konforme HTTP-Methoden an. Das Frontend greift auf diese Methoden zu und sendet mittels *POST*-Request eine Anfrage an das Backend. Die Antwort, welche es daraufhin vom Backend erhält, beinhalten die nötigen Informationen, um den eigenen Zustand anzupassen. Dieser Zustandstransformation ist namensgebend für REST. Ähnlich funktioniert die Kommunikation zwischen dem Backend und der Datenbank. Die Datenbank bietet die vier grundlegenden Methoden CRUD (Create, Read, Update, Delete) für persistenten Speicher, welche den HTTP-Methoden sehr ähneln. Das Backend kann so mit den Daten in der Datenbank arbeiten und auf Anfrage des Frontends Informationen weitergeben oder verändern.

¹⁸<https://cordova.apache.org/>

¹⁹<https://nodejs.org/en/>

²⁰<https://laravel.com/>

²¹<https://www.mongodb.com/>

3 Gebrauchstauglichkeit

Nachdem die Grundlagen für die Konzeption von Struktur und Funktionen im vorherigen Kapitel 2 behandelt wurden, werden nun die Grundlagen für die Erreichung einer gebrauchstauglichen Benutzeroberfläche dargestellt. Hierfür wird zunächst die Gebrauchstauglichkeit selbst eingeordnet und definiert und in den anschließenden Kapiteln werden verschiedene Richtlinien als Teilziele zur Umsetzung der Gebrauchstauglichkeit in ScOuT erstellt.

Die Ergonomie ist die Wissenschaft, welche die Arbeitsgeräte für eine Aufgabe so anpasst, dass ein optimales Arbeitsergebnis folgt. Dabei werden Arbeitsabläufe, Arbeitsbedingungen und die Anordnung der Gegenstände sowohl zeitlich als auch räumlich bestmöglich gestaltet [DINb]. Darunter fällt auch die Software-Ergonomie (engl. usability engineering). Software ist in diesem Fall das Arbeitsgerät, welches es im ergonomischen Sinne zu optimieren gilt. In der Software-Ergonomie soll die Software leicht verständlich und schnell benutzbar gestaltet werden [DIN02][Ric97]. Abbildung 5 veranschaulicht diesen Zusammenhang. Wie ebenfalls in der Abbildung zu sehen ist, wird das Ziel der Software-Ergonomie mit Hilfe der Gebrauchstauglichkeit (engl. usability) umgesetzt [Ric97][DIN98]. Nielsen, einer der Usability-Pioniere, bezeichnet deshalb die Gebrauchstauglichkeit als ein Attribut zur Messung der Qualität der Benutzerschnittstelle einer Software [BN13]. Um diese Ziele zu erreichen, wird sich in der Software-Ergonomie mit dem Design und der Verwendung von Benutzerschnittstellen beschäftigt. Dadurch fällt diese Wissenschaft in das Feld der Mensch-Computer-Interaktion (engl. human-computer interaction; kurz HCI) [Car18]. Wie in Abbildung 5 zu sehen ist, bewertet die Gebrauchstauglichkeit folglich die Oberfläche und die Dialogkontrolle einer Software.

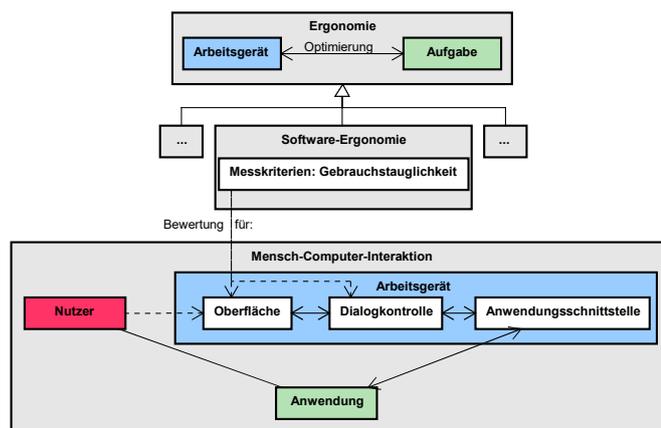


Abbildung 5: Einordnung der (Software-)Ergonomie, Gebrauchstauglichkeit und HCI

Nach der ISO DIN EN NORM 9241-11 umfasst die Gebrauchstauglichkeit drei Eigenschaften, um die Anpassung der Software im Sinne eines optimierten Arbeitsergebnisses zu erreichen [DIN98]:

Effektivität (engl. effectiveness) „Die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen.“

Effizienz (engl. efficiency) „Der im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzte Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen.“

Zufriedenheit (engl. user satisfaction) „Freiheit von Beeinträchtigung und positive Einstellung gegenüber der Nutzung des Produkts.“

Gebrauchstauglichkeit umfasst jedoch nicht das Nutzungserlebnis (engl.: user experience; kurz: UX). Dieses beinhaltet sowohl ästhetische Aspekte, also ansprechende Gestaltung, als auch emotionale Faktoren, wie Spaß bei der Nutzung (Joy of use) [DIS09]. Auch sollte der Begriff Gebrauchstauglichkeit nicht verwechselt werden mit dem der Benutzerfreundlichkeit (engl. user friendliness). Für die Nutzer klingt dieser zwar intuitiv richtig, jedoch haben die Internationale Organisation für Normung (eng.: International Organization for Standardization; kurz: ISO) aufgrund von mehrdeutigen Interpretationen dort eine Trennung gezogen. Benutzerfreundlichkeit umfasst nach dieser Definition sowohl Gebrauchstauglichkeit, als auch das Nutzungserlebnis und ist somit nicht alleine mit der Gebrauchstauglichkeit gleich zu setzen.

3.1 Dialoggestaltung: DIN EN NORM 9241-110

Zur Erreichung der in der Gebrauchstauglichkeit definierten Ziel-Eigenschaften einer Software wurden Richtlinien und Eigenschaften von verschiedenen Parteien, wie der Iso, festgelegt. Die DIN EN NORM 9241-110 bildet dabei einen Unterpunkt in der Norm der Gebrauchstauglichkeit selbst.

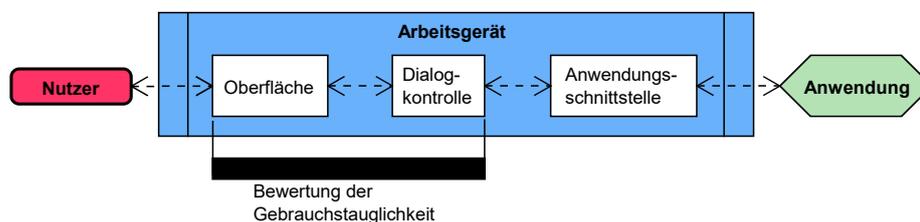


Abbildung 6: Ausschnitt aus Abbildung 5: Zusammenhang HCI und Gebrauchstauglichkeit

Wie in Abbildung 6 dargestellt, ist auch die Dialogkontrolle ein Teilgebiet der Gebrauchstauglichkeit [Ham06]. Die von der ISO aufgestellte DIN EN

NORM 9241-110 (ehemals ISO 9241-10) ist eine Richtlinie für die Bewertung genau dieser Dialoggestaltung [DIN08]. In der Norm wurden folgende sieben Eigenschaften für die Dialoggestaltung definiert:

Aufgabenangemessenheit „Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.“

Selbstbeschreibungsfähigkeit „Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt wird.“

Steuerbarkeit „Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“

Erwartungskonformität „Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z. B. den Kenntnissen aus dem Arbeitsgebiet, der Ausbildung und der Erfahrung des Benutzers sowie den allgemein anerkannten Konventionen.“

Fehlertoleranz „Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand durch den Benutzer erreicht werden kann.“

Individualisierbarkeit „Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe, individuelle Vorlieben des Benutzers und Benutzerfähigkeiten zulässt.“

Lernförderlichkeit „Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.“

3.2 Qualitätskriterien: ISO/IEC 25000

Die Qualitätssicherheit von Software hat sich ebenfalls mit der Gebrauchstauglichkeit befasst und dazu eigene spezifizierende Eigenschaften aufgestellt. Wie jedes Teilgebiet der Qualitätssicherung trägt auch die Gebrauchstauglichkeit zur Optimierung der Qualität der Software bei. In diesem Sinne wird die Benutzung der Software qualitativ durch die Einhaltung der Kriterien erhöht.

In der ISO/IEC 25000 (ehemals ISO/IEC 9126) wird die Qualitätssicherung in verschiedene Kategorien unterteilt. Darunter auch die Kategorie der Benutzbarkeit, welche sich mit dem Nutzungsaufwand beim Bedienen der Software beschäftigt. Diese Kategorie unterteilt die ISO/IEC 25000 in die fünf nachfolgenden Subkriterien [DINa]:

„**Attraktivität** beschreibt die Anziehungskraft der Anwendung gegenüber dem Benutzer.“

„**Bedienbarkeit** beschreibt den Aufwand für den Benutzer, bei der Bedienung der Anwendung.“

„**Erlernbarkeit** beschreibt den Aufwand für den Benutzer, die Bedienung der Anwendung zu erlernen.“

„**Verständlichkeit** beschreibt den Aufwand für den Benutzer, das Konzept der Anwendung zu verstehen.“

„**Konformität** beschreibt den Grad, in dem die Software Normen oder Vereinbarungen zur Benutzbarkeit erfüllt.“

3.3 Nielsen Heuristik

Nielsen gilt als einer der Usability-Pioniere und bezeichnet die Gebrauchstauglichkeit als das Attribut zur Messung der Qualität guten Interaktionsdesigns [BN13]. Mittels heuristischer Evaluation ermittelte er die zehn Usability Heuristiken für eine bessere Gestaltung der Benutzeroberfläche [Gro19][Nie94] (Übersetzung von: [Bor19]).

Sichtbarkeit des Systemstatus (eng.: visibility of system status) „Das System informiert den Nutzer immer darüber, was gerade passiert – rechtzeitig und durch angemessenes Feedback.“

Übereinstimmung von System und Wirklichkeit (eng.: match between system and the real world) „Das System spricht die Sprache des Nutzers – mit ihm vertrauten Wörtern, Phrasen und Konzepten. Entlehnt aus der echten Welt erscheinen Informationen in ihrer natürlichen und logischen Ordnung.“

Nutzerkontrolle und Freiheit (eng.: user control and freedom) „Nutzer führen Aktionen oft unbeabsichtigt durch. Auswege wie „Rückgängig“, „Wiederholen“ und „ESC“ sind deshalb immer möglich und sichtbar.“

Beständigkeit und Standards (eng.: consistency and standards) „Nutzer müssen nicht überlegen, ob unterschiedliche Wörter, Situationen und Aktionen das Gleiche meinen. Die Konventionen des Betriebssystems werden eingehalten.“

Fehlervermeidung (eng.: error prevention) „Besser als jede gute Fehlermeldung ist ein sorgfältiges Design, welches Fehler gar nicht erst auftreten lässt. Das System vermeidet fehleranfällige Situationen oder warnt den Nutzer und lässt ihn die Aktion bestätigen.“

Wiedererkennung statt Erinnerung (eng.: recognition rather than recall) „Durch sichtbare Objekte, Aktionen und Optionen muss der Nut-

zer weniger im Gedächtnis behalten. Anleitungen zum Gebrauch des Systems sind sichtbar oder leicht zu erreichen.“

Flexibilität und Effizienz (eng.: flexibility and efficiency of use) „Kurzbefehle und andere Abkürzungen – unsichtbar für Neulinge – beschleunigen bei fortgeschrittenen Nutzern die Bedienung. Zusätzlich sind häufige Aktionen individuell anpassbar.“

Ästhetisches und minimalistisches Design (eng.: aesthetic and minimalist design) „Dialogfenster enthalten keine überflüssigen oder nur selten gebrauchten Informationen. Denn jede zusätzliche Information steht in Konkurrenz mit den relevanten Informationen und mindert deren Sichtbarkeit.“

Hilfestellung beim Erkennen, Bewerten und Beheben von Fehlern (eng.: help users recognize, diagnose, and recover from errors) „Fehlermeldungen sollten in klarer Sprache (kein Code) formuliert sein, das Problem exakt beschreiben und eine konstruktive Lösung vorschlagen.“

Hilfe und Dokumentation (eng.: help and documentation) „Obwohl es besser ist, wenn der Nutzer ein System ohne Hilfe benutzen kann, ist es manchmal Nötig, eine Dokumentation bereitzustellen. In dem Fall sind die Informationen einfach zu finden und konzentrieren sich auf die Aufgabe des Nutzers. Die Dokumentation enthält konkrete Schritte zur Ausführung und beschränkt sich auf das Wesentliche.“

3.4 Vuolle

Vuolle beschäftigte sich nicht nur mit der allgemeinen Oberflächengestaltung, sondern der von einem Mobiltelefon. Obwohl es in der Auswahl heutiger Mobiltelefone auch recht große Displays gibt, sind die Maße dennoch nicht mit denen eines Tablets oder Monitors zu vergleichen. In Anbetracht dessen, dass ScOuT auf allen Geräten funktionstüchtig sein soll, muss die Gebrauchstauglichkeit für Mobiltelefone im Besonderen einbezogen werden. Vuolle stellt dafür neun Usability Dimensionen auf [VAK⁺08][JABH15]:

Installation (eng.: installation) Die Anwendung soll mit geringem Aufwand zu installieren oder einzurichten sein.

Erlernbarkeit (eng.: learnability) siehe Abschnitt 3.2 Erlernbarkeit.

Bedienbarkeit (eng.: ease of use) Die Beweglichkeit von einem Aufgabenschritt zum nächsten soll innerhalb der Anwendung bequem und einfach sein.

Effizienz (eng.: efficiency) siehe Kapitel 3 Effizienz.

Effektivität (eng.: effectiveness) siehe Kapitel 3 Effektivität.

Zufriedenheit (eng.: user satisfaction) siehe Kapitel 3 Zufriedenheit.

Telefon-Kontext (eng.: mobile context) Akkuverbrauch, Bildschirmgröße, Informationseingabe, Umgebung, Eignung in Bezug auf die Aufgabe, Informationsaustausch und der Wechsel der Anwendung sollen einbezogen werden. (weitere Informationen folgen unter Abschnitt 5.2)

Sicherheit (eng.: safety) Die Nutzung der Anwendung soll kein Sicherheitsrisiko hervorrufen und die Umgebung nicht vollständig ausblenden.

Support (eng.: support) Es soll jederzeit klar sein, von wem Hilfe zu bekommen ist und die Hilfeinformationen sollen nützlich sein.

3.5 Richtlinien und Eigenschaften im Überblick

Auf den ersten Blick scheint es, dass die meisten definierten Eigenschaften sich stark unterscheiden und alle vier Quellen völlig verschiedene Richtlinien/Heuristiken aufstellen. Doch genauer betrachtet gibt es einige Überschneidungen. Tabelle 1 soll dabei einen zusammenfassenden Überblick bieten. Effektivität, Effizienz und Zufriedenheit bilden dabei für alle das zu erreichende Ziel. Lediglich Vuolle nennt sie in seiner eigenen Liste zusätzlich.

Das Lernen der Anwendung wird zum Beispiel bei drei Richtlinien angesprochen. Die Qualitätskriterien beschreiben in der Eigenschaft Erlernbarkeit und Verständlichkeit, dass Ziel den Lernaufwand des Nutzers sowohl in der Bedienung, als auch dem Verständnis vom Konzept der Anwendung zu verringern. Vuolle verwendet die selbe Definition von Erlernbarkeit. Ähnlich wird in der Lernförderlichkeit der Dialoggestaltung davon gesprochen, den Nutzer beim Lernen anzuleiten.

Mit dem Thema Fehler beschäftigen sich die ISO 9241 und ganz besonders die Nielsen Heuristik. Dort gibt es in der Fehlertoleranz, Fehlervermeidung und der Hilfestellung beim Erkennen, Bewerten und Beheben von Fehlern, zwei Schwerpunkte in Bezug auf das Thema. Zum einen sollen Fehler möglichst gar nicht erst entstehen und zum anderen soll dem Nutzer das Beheben des Fehlers durch ausreichende Hilfen durch das System vereinfacht werden.

Eng verbunden mit dem Thema der Fehler ist das der Hilfe. Denn Fehlervermeidung erfolgt dann, wenn der Nutzer ausreichend Hilfe erhalten hat das System zu erlernen. Vuolle fordert deshalb einen Support, während in der Nielsen Heuristik von einer Dokumentation die Rede ist.

Eine weitere Oberflächeneigenschaft, welche die ISO 9241 und Nielsen beschreiben, ist der Informationsaufbau. Während Nielsen über rechtzeitiges und angemessenes Feedback in der Sichtbarkeit des Systemstatus sowie ein minimalistisches Design spricht, fordert die ISO 9241 auf ähnliche Weise Rückmeldungen des Systems für seine Selbstbeschreibungsfähigkeit.

Die Steuerung der Oberfläche ist ein weiteres Thema, mit welchem sich die Richtlinien befassen. Dabei geht es in der Steuerbarkeit der Dialoggestaltung um die Rücksichtnahme auf die Fähigkeiten und Wünsche des Nutzers.

Genauso beschreibt die Nutzerkontrolle und Freiheit in der Nielsen Heuristik ähnliche Ziele an die Oberflächensteuerung. Die Bedienbarkeit von Vuolle hat zwar eine andere Definition, aber das selbe Ziel wie die Bedienbarkeit der Qualitätskriterien. Beide verlangen den Aufwand der Steuerung zu verringern und dem Nutzer diese zu vereinfachen.

Eigenschaft	ISO 9241- 110	ISO/ IEC 25000	Nielsen	Vuolle
Effektivität	✗	✗	✗	✓
Effizienz	✗	✗	✗	✓
Zufriedenheit	✗	✗	✗	✓
Fehler	✓	–	✓, ✓	–
Hilfe	–	–	✓	✓
Lernen	✓	✓, ✓	–	✓
Informationsaufbau	✓	–	✓, ✓	–
Steuerung	✓	✓	✓	✓
Nutzeranpassung	✓	–	✓	–
Einheitlichkeit	✓	–	✓, ✓, ✓	–
Aufgabenangemessenheit	✓	–	–	–
Attraktivität	–	✓	–	–
Konformität	–	✓	–	–
Installation	–	–	–	✓
Telefon-Kontext	–	–	–	✓
Sicherheit	–	–	–	✓

Tabelle 1: Richtlinien und Eigenschaften im Überblick (✗= Übergeordnetes Ziel; ✓= einbezogene(s) Thema/Eigenschaft; – = kein direkter/schwacher Zusammenhang)

Eine individuelle Anpassung des Systems an den Nutzer wird sowohl in der Individualisierbarkeit der Dialoggestaltung, als auch bei Nielsen gefordert. Nielsen spielt dabei in der Flexibilität und Effizienz speziell auf die Fähigkeiten des Nutzers an und die darauf angepassten, individuellen Steuerungsmöglichkeiten.

Das Thema der Einheitlichkeit wird bei Nielsen mehrmals angesprochen. Sei es die Übereinstimmung von System und Wirklichkeit, wodurch eine Einheit zwischen ihnen gebildet wird, oder die Einheitlichkeit innerhalb des Systems durch Beständigkeit und Standards und eine damit verbundene Wiedererkennung statt Erinnerung. Doch auch die Erwartungskonformität der ISO 9241 befasst sich mit einheitlichen Dialogen im System und in Bezug auf die Umgebung des Nutzers.

Obwohl nun bereits einige Zusammenhänge hervorgehoben wurden, gibt es noch viele weitere. So sind verschiedene Themen, wie zum Beispiel die

Einheitlichkeit, der Informationsaufbau und die Hilfe, ausschlaggebend für das einfache Erlernen der Anwendung. Diverse Übergänge von Fehlervermeidung und Hilfe sind zudem fließend. Insgesamt können einige Eigenschaften nicht voneinander getrennt werden und sollten in der Gestaltung der Oberfläche als Gesamtes betrachtet werden. Es gibt jedoch neben den verknüpfbaren Eigenschaften, noch weitere, welche keinen schwachen oder starken Zusammenhang aufzeigen. Dennoch haben auch diese ihre Verbindlichkeit zur Oberflächengestaltung

4 Funktionales Konzept

Der Ablauf der Umsetzung des angestrebten Tools zur Erleichterung der Vereinsarbeit erfolgt aus einer Mischung von verschiedenen softwareergonomischen Vorgehensmodellen. Dabei wird der User-Centered-Design-Prozess (kurz: UCD-Prozess) einbezogen [BN13]. Dieses Vorgehensmodell stellt die Gebrauchstauglichkeit in den Mittelpunkt der Planung und der Einbezug von Usability-Methoden wird im Besonderen berücksichtigt. Des Weiteren werden das Vorgehen in [Rut02] für die Einordnung der Usability-Methoden im Software-Lebenslauf betrachtet. Hier werden insbesondere die Schritte der funktionellen Konzeption in Bezug auf die Gebrauchstauglichkeit genauer eingeteilt. Ebenso werden die Schritte der Festlegung des Systemdesigns aus [Ham06] einbezogen. Ein besonderes Augenmerk wird dabei auf die Anforderungserhebung gelegt, in welcher darauf geachtet wird, dass Funktionen dem Menschen oder der Maschine zugeordnet werden. Daraus ergibt sich der in Abbildung 7 angestrebte Ablauf zur Konzeption und Implementation des Planungstools.

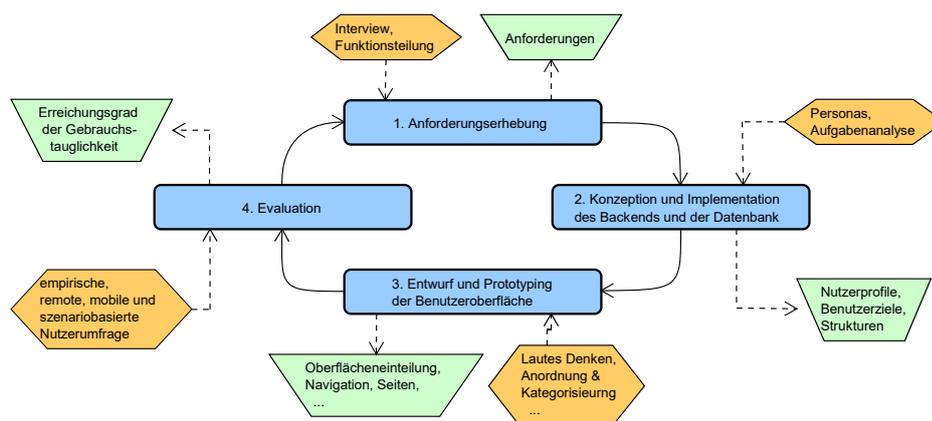


Abbildung 7: Planungszyklus für die Umsetzung der geplanten App

Wie in Abbildung 7 dargestellt besteht der erste Schritt zur Entwicklung von ScOuT darin die Anforderungen zu erheben. Hierfür wurde ein Interview mit vier Stämmen aus der Diözese Limburg sowie dem Diözesanbüro (kurz: DiBü) selbst durchgeführt [BN13][Rut02][Usa19][Ham06].

Die Inhalte des Interviews basieren dabei auf der Grundlage des in den Abschnitten 2.2 und 2.3 zusammengetragenen Ist-Zustands der Pfadfinder und der bereits existierenden Tools. Dies hat den Zweck den Soll-Zustand des geplanten Systems genauer zu definieren. Dieser angestrebte Zustand wird in den Anforderungen an das Tool festgehalten.

Im ersten Bestandteil des Interviews wurde zunächst über die Problemstellung in der Vereinsarbeit diskutiert. Jeder der Stämme und auch das DiBü haben bereits viele Wege für die Weitergabe von Informationen ausprobiert. Darunter fallen nicht nur digitale Wege, wie CloudServices, Homepages und Planungstools, sondern auch handgeschriebene Notizbücher und mündliche Überlieferungen. Der Wunsch nach einem Planungstool, welches die Weitergabe von Informationen vereinheitlicht und erleichtert, ist während des Interviews bei allen Befragten aufgekommen. Um die Vernetzung nun zu verbessern, wurden verschiedene Funktionswünsche geäußert. Darunter fallen Mailinglisten, Erinnerungshinweise, Checklisten, integrierte Kalender und E-Mail Accounts, Suchfunktionen, sowie ein gemeinsames und halb öffentliches Ablagesystem mit amtsbezogenen Zugriffsrechten. Doch all diese Funktionen sind bereits in den in Abschnitt 2.3 genannten Tools weitestgehend abgedeckt. Die einzigen Anforderungen der Stämme und des DiBüs, welche sich nicht mit bereits existierenden Tools umsetzen lassen, sind spezielle Anforderungen an Templates. Aus den Interviews war zu entnehmen, dass die Funktionen der vorhandenen Templates für die Wiederverwendung in der Vereinsarbeit zu umständlich sind. Jedes Template erfordert, wie bereits in Abschnitt 2.3 vorgestellt, zu viele nachträgliche Anpassungen durch den Nutzer. Aufgrund dessen ergaben sich folgende Anforderungen, um spezifisch den Vereinen zu helfen:

1. Veranstaltungen sollen einen konkreten Zeitraum haben;
Ursache: Existierende Tools stellen eine relativ unkontrollierte Abarbeitung von Aufgaben dar, weil Projekte keinen Zeitraum haben. Für die Planung von Veranstaltungen ist dieser Zeitraum jedoch als eine Deadline essentiell.
2. Die Frist, bis wann eine Aufgabe erledigt sein sollte, ist automatisch auf den Veranstaltungsbeginn gesetzt;
Ursache: Das Setzen der Frist per Hand wird unnötig, sobald Anforderung 1. umgesetzt wird.
3. Es soll ein Frist-Erinnerung eingeführt werden, welcher den Nutzer erinnert, wann eine Aufgabe erledigt werden sollte;
Ursache: Dient der automatischen Sortierung der Aufgaben nach Dringlichkeit und ersetzt die ursprüngliche Frist.
4. Die Aufgaben sollen automatisch nach Eigenschaften Erledigt/Unerledigt und Frist-Erinnerung wartend/Frist-Erinnerung aktiv sortiert werden;
Ursache: Das Sortieren per Hand, wie es in einigen der existierenden Tools notwendig ist, stellt einen erhöhten Arbeitsaufwand für den Nutzer dar. Zudem sollte durch diese Sortierung der Nutzer nicht manuell nach den aktiven Frist-Erinnerung suchen müssen.

5. Die Frist-Erinnerung soll kein konkretes Datum haben, sondern ein relativer Wert abhängig vom Veranstaltungsbeginn sein;
Ursache: Bei der Verwendung von Templates werden die relativen Werte eins zu eins übernommen, wodurch dem Nutzer Arbeit erspart wird alle Aufgaben mit neuen Fristen zu versehen oder Abhängigkeiten immer wieder neu eingeben zu müssen.
6. Das Template soll alle Aufgaben immer als unerledigt und nie als erledigt übernehmen;
Ursache: Auch hier wird dem Nutzer die Arbeit erspart, alle Aufgaben per Hand wieder auf unerledigt zu setzen.

Diese Aufstellung der Anforderungen verändert stark die Funktionsteilung, wie sie in den Tools in Abschnitt 2.3 vorgestellt wurden. In der Funktionsteilung werden die Funktionen in menschliche und maschinelle unterteilt [Ham06]. In der Teilung dieser Anforderungen liegt nun ein stärkerer Fokus auf der Übernahme von Funktionen durch die Maschine, um dem Mensch die Arbeit zu erleichtern.

Neben der Spezifikation des Ziels der App wurden im Interview weitere Punkte besprochen. Der zweite Teil des Interviews soll dabei eine direkte Grundlage für den zweiten genannten Schritt in Abbildung 7 sein. Ziel ist es dabei die Zielgruppe von ScOuT besser kennen zu lernen. Zu diesem Zweck wurden Fragen bezüglich der Personengruppen und Ämter in den Vereinen gestellt. Ebenso wurden die Aktionen (=Veranstaltung der Pfadfinder) der Pfadfinder ausführlich besprochen, wodurch ein Überblick über die Vielfalt der Aufgaben ermittelt werden konnte. Vollständige Listen dazu befinden sich im Anhang B und werden in den nachfolgenden Abschnitten 4.1, 4.2 und 4.3 genauer untersucht und einbezogen.

Deshalb beschäftigen sich die nachfolgenden Abschnitte, wie im Planungszyklus in Abbildung 7 dargestellt, mit der funktionellen Konzeption eines Planungstools, welches genau dieses Defizit beheben soll. Die funktionale Konzeption umfasst dabei sowohl die Planung des Backends, als auch der Datenbank, welche mit den in Abschnitt 2.4 recherchierten Frameworks umgesetzt werden.

4.1 Personas

Teil der Stakeholder-Analyse ist es die Nutzerprofile zu bestimmen. Hierfür wird die Usability-Methode Personas verwendet [Usa19]. Personas oder Nutzerprofile sind Benutzergruppen mit ähnlichen Charaktereigenschaften und Verhaltensweisen gegenüber dem System. Ziel dieser Analyse ist es die Benutzer von ScOuT besser kennen zu lernen und deren Ziele in Bezug auf die Nutzung von ScOuT ermitteln zu können. Wie in Abbildung 7 in Kapitel 4 dargestellt, befindet sich diese Usability-Methode am Anfang des zweiten Schrittes des Planungszyklus.

Die Ermittlung der Personas erfolgt anhand der Vereinsanalyse, der Struktur von Veranstaltungen und den geführten Interviews. Aufgrund dessen, dass das Produkt in erster Linie ein Hilfs-Tool für Vereine darstellen soll, werden Firmen, private Nutzer und Sonstige in dieser Arbeit nicht weiter beachtet, obwohl diese ScOuT am Ende auch nutzen könnten. Die ermittelten Personas beziehen sich folglich ausschließlich auf Vereinsmitglieder und Personen, welche indirekt mit der Vereinsarbeit zusammenhängen.

Gäste

Diese Benutzergruppe besteht aus Personen, welche nicht aktiv Aufgaben erledigen, sondern lediglich Teilnehmer von Veranstaltungen sind und Informationen einsehen möchten. Darunter fallen zum Beispiel Eltern und die Jugendlichen.

Orgateam

Die Persona Orgateam umfasst Gruppen von Vereinsmitgliedern, welche die Planung einer Veranstaltung anstreben. Teilweise sind Orgateams an Ämter gebunden, doch die Aufgaben beim Planen einer Veranstaltung unterscheiden sich nicht, weshalb der Bezug zu Ämtern vernachlässigt werden kann.

SubOrgateam

Die Persona SubOrgateam ist dem Orgateam sehr ähnlich. Der einzige Unterschied von einem Orgateam, welches die Hauptverantwortung für die Planung von Veranstaltungen trägt, zu einem SubOrgateam ist der, dass das SubOrgateam keinen Gesamtüberblick über die ganze Veranstaltung benötigt, sondern nur aufgabenspezifische Informationen.

Helfer

Die Helfer bilden eine Benutzergruppe, welche überwiegend Informationen über die Veranstaltung abrufen möchte, um zu erfahren wann und wo ihre Hilfsarbeit stattfindet. Sie unterscheiden sich jedoch von den Gästen, da die Helfer teilweise für bestimmte Programmstücke eigene Materialien mitbringen müssen und somit selbst aktive Aufgaben haben.

Ämter

Personen, welche Ämter in einem Verein inne haben, müssen eine Vielzahl von Aufgaben erledigen. Das Hauptaugenmerk liegt bei dieser Persona auf den amtsabhängigen Aufgaben. Obwohl diese auch teilweise das Planen einer Veranstaltung umfassen, wird sich bei dieser Personengruppe nur auf die amtspezifischen Aufgaben konzentriert, da diese bereits in der Persona Orgateam abgedeckt sind.

4.2 Funktionsaufgaben

Nachdem die Nutzerprofile definiert wurden, müssen nun ihre konkreten Bedürfnisse ermittelt und in Teil-Aufgaben unterteilt werden [Ric97]. In diesem Sinne sind mit dem Begriff „Aufgaben“ nicht zu erledigende Tätigkeiten des Nutzers gemeint. Nach der Aufgabenangemessenheit aus den Eigenschaften der Dialoggestaltung (vergleiche Abschnitt 3.1) bedeuten Aufgaben hier die benötigten Funktionen, welche der Nutzer in ScOuT erwartet [Ram19]. Zur Vermeidung von Doppeldeutigkeiten werden die nach der Aufgabenangemessenheit genannten Aufgaben im Folgenden immer Funktionsaufgaben genannt.

Diese Usability-Methode gehört noch zum Schritt 2 des in Abbildung 7 aus Kapitel 4 dargestellten Planungszyklus. Dieser Schritt befasst sich mit der in Abschnitt 3.1 genannten Eigenschaft der Aufgabenangemessenheit. Erst mit schriftlich fixierten Funktionsaufgaben wird der Erreichungsgrad der Aufgabenangemessenheit für die spätere Evaluation messbar (siehe [Ram19]).

Dabei wird jedoch beachtet, dass ein zu großer Anstieg an Funktionen im Gegensatz zur Gebrauchstauglichkeit steht. Denn große Mengen an Funktionsmöglichkeiten, können sowohl die Erlernbarkeit, als auch die Verständlichkeit (siehe Kapitel 3) verringern, wie auch die Umsetzung von Selbstbeschreibungsfähigkeit, Steuerbarkeit und Lernförderlichkeit (siehe Abschnitt 3.1) erschweren [Ham06].

In der nachfolgenden Liste werden die spezifischen Funktionsaufgaben der Personas aufgelistet.

Funktionsaufgaben für Gäste

- Einsehen von Veranstaltungs-Informationen
- Anmelden als Teilnehmer

Funktionsaufgaben für Orgateam

Das Planen von Veranstaltungen besteht aus folgenden Funktionsaufgaben:

- Erstellen, Bearbeiten und Löschen von Veranstaltungen
- Erstellen, Bearbeiten und Löschen von Aufgaben innerhalb einer Veranstaltung
- Zuweisen von SubOrgateams
- Zuweisen von Helfern
- Erstellen, Bearbeiten, Löschen und Nutzen von Templates von Veranstaltungen

Funktionsaufgaben für SubOrgateam

- Erstellen, Bearbeiten und Löschen von Aufgaben innerhalb einer Veranstaltung

Funktionsaufgaben für Helfer

- Einsehen von spezifischen Veranstaltungs-Informationen (Programmplan für Hilfsarbeiten bei der Durchführung von Veranstaltungen)
- Optional: Erstellen, Bearbeiten und Löschen von Aufgaben innerhalb einer Veranstaltung

Funktionsaufgaben für Ämter

- Erstellen, Bearbeiten und Löschen von Aufgaben für amtsspezifische Ziele (unabhängig von Veranstaltung)

Des Weiteren gibt es noch eine Vielzahl an Funktionen, wie die Einbindung von Doodle, dem Kalender, E-Mail-Addressbücher oder einem Cloud Service. Obwohl domänenübergreifende Funktionen, wie zum Beispiel, der Zugriff auf das Kontaktbuch durch ein Navigationsgerät, im Gegensatz zur Gebrauchstauglichkeit stehen [Ham06], gehören sie dennoch zu dem heutigen Standard. Die Nutzer erwarten diese Funktionen, was sich auch im Interview widerspiegelt hat (siehe Kapitel 4). Im Rahmen der Masterarbeit wird jedoch nicht weiter auf diese eingegangen.

4.3 Umsetzungskonzepte

Die hier genannten Ideen zur Umsetzung der Funktionsaufgaben für die Personas basieren auf eigenen, konzeptionellen und softwaretechnischen Überlegungen und wurden mittels verschiedener Methoden in Form von Aktivitätsdiagrammen, UseCase-Diagrammen, Klassendiagrammen und Zustandsdiagrammen entworfen. Nachfolgend findet sich eine grobe Zusammenfassung des konzeptionellen Entwurfs. Grundlage dafür ist ein Konzept, in welchem der registrierte Nutzer Vereine anlegen, bearbeiten und löschen kann, andere Nutzer in die Vereine einladen, sowie selbst durch Einladungen einem Verein beitreten kann.

Einsehen von Informationen und Anmeldung:

Ziel dieser Funktion ist es für Gäste relevante Informationen über eine Veranstaltung (z. B. Teilnehmerbeitrag, Zeitraum, Lokation) und ein Anmeldeformular, sowie Informationen für Helfer (z. B. Schichtplan, Programmablauf) öffentlich zugänglich zu machen.

- A) Gäste müssen sich ebenfalls in ScOuT registrieren und mit der Rolle „Gast“ einem Verein beitreten. Dann können sie die genannten Funktionen nutzen. Nachteil: Nicht jeder möchte sich dort registrieren.

- B) Alle Veranstaltungen werden auf einer zur App gehörenden Website öffentlich gelistet und jeder kann über diesen Zugriff die Funktionen nutzen. Nachteil: Gäste müssen den Vereinsnamen innerhalb von ScOuT wissen, um die richtigen Veranstaltungen zu finden.
- C) ScOuT bietet eine Schnittstelle pro Verein an, welche Funktionen und Informationen bereitstellt und auf den Webseiten der Vereine integriert werden kann.

Aufgrund der in A) und B) genannten Nachteile, stellt C) eine sinnvolle Umsetzungsmöglichkeit dar und sollte bei der Implementierung den anderen beiden Alternativen vorgezogen werden.

Verarbeiten von Veranstaltungen:

Verarbeiten beinhaltet in diesem Fall: Erstellen, Bearbeiten und Löschen, sowie das Zuweisen von Helfern zu einem Programmpunkt.



Abbildung 8: Grundlegende Strukturen einer Veranstaltung.

Für die Umsetzung dieser Funktionen muss definiert werden, dass eine Veranstaltung immer von einem konkreten Verein geplant wird. Dies bedeutet, dass der Nutzer zunächst einem Verein angehören muss. Die Veranstaltung selbst muss für die Funktionen eigene Objektstrukturen haben. Diese werden in Abbildung 8 dargestellt. Die Veranstaltung beinhaltet essentielle Eigenschaften, wie einen Namen und eine Lokation. Zudem kann sie verschiedene Programmpläne haben. Die Programmpunkte sind dann die spezifischen Einträge im Plan. Ein Beispiel wäre die Veranstaltung „Stammesleitung“ mit dem Programmplan „Ablauf“, welcher Programmpunkte enthält wie „1. Reflexion vergangener Aktionen“, „2. Besprechung zukünftiger Aktionen“, „3. Besprechung sonstiger Themen“. Sind diese Inhalte in ScOuT alle erfüllt, sollte die Verarbeitung von Veranstaltungen alle grundlegenden Funktionen abdecken. Diese möglichen Bedingungen erfüllen nicht nur alle in Abschnitt 2.3 erarbeiteten Eigenschaften des Ist-Zustand, sondern auch die aus dem Interview gewünschten Anforderungen an einen Soll-Zustand. Während in existierenden Apps Veranstaltungen terminlos sind und somit auch Projekte ohne festen Zeitraum abgebildet werden können, wird in der eigens entwickelten Struktur die feste Angabe von Start- und End-Termin für den Veranstaltungszeitraum erwartet.

Verarbeiten von Aufgaben:

Verarbeiten beinhaltet in diesem Fall: Erstellen, Bearbeiten und Löschen, sowie das Zuweisen von SubOrgateams und Helfern zu einer Aufgabe.

Im Falle der Aufgaben, welche zu einer Veranstaltung gehören, gilt wie beim Verarbeiten der Veranstaltung selbst, dass der Nutzer zuvor einem Verein beigetreten ist und in diesem Verein Veranstaltungen angelegt hat, welchen Aufgaben zugeordnet werden können. Für die Aufgaben der Ämter sollte der Nutzer in dem Verein bereits eine Struktur für die verschiedenen Ämter angelegt haben, sodass die Aufgaben Ämtern zugewiesen werden können. Im Falle eines Wechsels des Amtsträgers wird die Aufgabe durch diese Zuweisung weitergereicht.

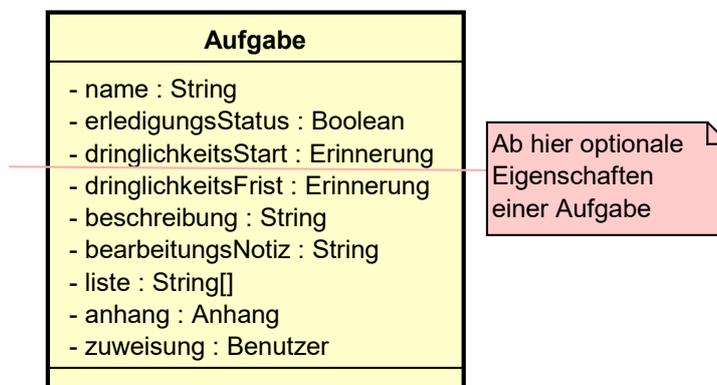


Abbildung 9: Grundlegende Strukturen einer Aufgabe

Wie für die Veranstaltung wird für die Aufgabe ebenso eine Objektstruktur entwickelt. Diese wird in Abbildung 9 dargestellt. Die ersten drei Attribute Name, erledigungsStatus und dringlichkeitsStatus sind für die Objektstruktur der Aufgabe essentiell und müssen beim Erstellen einer Aufgabe verpflichtend angegeben werden. Wie in Kapitel 4 mit Hilfe des Interviews erarbeitet, wird die ehemals in Abschnitt 2.3 genannte Frist durch das Attribut dringlichkeitsStart ersetzt. Die Frist kündigte in bereits existierenden Tools immer an, bis wann die Aufgabe erledigt werden sollte. Durch die Anpassung der Veranstaltung ist jedoch automatisch der Starttermin der Veranstaltung selbst die Frist. Das Attribut dringlichkeitsStart erweitert diese Änderung des Konzepts zu anderen Tools dahingehend, dass nun angegeben werden kann, ab wann die Aufgabe erledigt werden soll. Die unerledigten Aufgaben erhalten dadurch eine zweiteilige Hierarchie, welche Aufgaben mit erreichtem dringlichkeitsStart in die wichtigere Ebene der Hierarchie verschiebt. Das Fristen-Konzept in ScOuT wird neben dem Veranstaltungs-Start zusätzlich durch eine optionale Version der ursprünglichen Frist aus anderen

Tools erweitert, denn manche Aufgaben müssen in einem bestimmten Abstand vor dem Veranstaltungsbeginn erledigt werden. Wie in Abschnitt 2.1 bereits erklärt, gehört zur Planung einer Veranstaltung nicht nur die Vorbereitung, sondern auch die Durchführung und die Nachbereitung. Deshalb sollte bei dem dringlichkeitsStart, sowie der dringlichkeitsFrist darauf geachtet werden, dass diese auch in und nach dem Veranstaltungszeitrum gelegt werden können.

Eine weitere Neuerung der in Abschnitt 2.3 ermittelten Aufgaben-Struktur ist die Notiz-Funktion. Eine Beschreibung der Aufgabe, welche es sonst auch in anderen Tools gibt, soll beinhalten, wie und wofür eine Aufgabe zu erledigen werden kann und ist somit fest an die Aufgabe gebunden. Die Notiz dagegen bietet ein dynamisches Feld, in welchem zum Beispiel (Zwischen-)Resultate festgehalten werden können, während und nachdem die Aufgabe erledigt wurde.

Das Attribut der Zuweisung gab es bereits auch in anderen Tools um Aufgaben einem einzelnen Mitglied des Orgateams aufzutragen. Da im Fall von ScOuT jedoch auch Helfer und SubOrgateams zugewiesen werden sollen, welche nicht Teil des Orgateams selbst sein müssen, erhält auch diese Funktion eine neue Bedeutung im Vergleich zu den existierenden Tools.

Angepasst an die Standards, welche bei Planungstools existieren, fehlen nur noch die Checklisten und Anhänge als optionale Ergänzungen für die Aufgaben.

An dieser Stelle möchte schon einmal angemerkt werden, dass noch einige weitere Ideen bei der Konzeption bedacht wurden, welche die Grundlagen um ein vielfaches erweitern. Hier ein einzelnes Beispiel als Andeutung an die Ideen: Es soll spezielle Aufgabentypen geben, welche bestimmte Aufgaben gezielter unterstützen. Der Grundentwurf beinhaltet nur eine Beschreibung, was gemacht werden soll und wann es erledigt werden soll. Die speziellen Aufgaben dagegen bieten zusätzliche Funktionen, beispielsweise die Kontakt-Aufgabe. Für eine Veranstaltung muss eine Lokation gebucht und dafür verschiedene Personen angefragt werden. Werden zum Beispiel immer die selben fünf Personen angefragt, könnten diese Kontakte bereits als Liste von Kontakt-Strukturen verfügbar gemacht werden. Jeder einzelnen Struktur in der Liste würden Zustände zuzuordnen sein, wie „Kontaktaufnahme erfolgt - Aufgabe erledigt“, was in diesem Fall heißen würde, die Lokation wäre gebucht. Dieser Kontakt würde dann in der Aufgabe gleichbedeutend der bearbeitungsNotiz Resultate angeben. Die anderen Kontakte werden in den Hintergrund gelegt und erst beim Erstellen einer Kopie der Veranstaltung mit Hilfe des Templates wieder hervorgehoben. Auf Grundlage ausführlicher Evaluation der Veranstaltungen (siehe Anhang B) und deren Vorbereitung gab es Überlegungen zu Kontaktaufnahmen, Routenplanung, Programmplanung, E-Mail-Versand, Datenverwaltung, Packlisten, Reiseplanung sowie Abrechnungsmodule. Diese Konzeptideen haben jedoch eine so geringe Priorität, dass sie hier nicht ausführlich beschrieben werden.

Verarbeiten von Templates:

Verarbeiten beinhaltet in diesem Fall: Erstellen, Bearbeiten, Löschen und Nutzen von Templates.

Das Objekt Template entspricht grundlegend den Kopien, Duplikaten und anderen Varianten aus den in Abschnitt 2.3 genannten Tools und dient folglich dazu die Veranstaltung erneut oder ähnliche Veranstaltungen damit zu planen.

Wie in diesem Abschnitt bereits genannt, sind einige der Template-ähnlichen Funktionen für den Nutzer umständlich oder bieten nicht genau das, was für Vereine benötigt wird, um wiederkehrende Veranstaltungen zu planen. Dazu zählt zum Beispiel das häufige Problem, dass sich Nutzer per Hand oder auf anderen umständlichen Wegen darum kümmern müssen, dass Fristen neu gesetzt und Aufgaben von dem Status Erledigt wieder in den Status Unerledigt gesetzt werden müssen.

Diese Probleme sollen mit dem eigenen Konzept des Templates behoben werden, was folgende Auswirkungen hat: In Veranstaltungen soll für den Benutzer einstellbar sein, ob die Lokation, einzelne Programmpläne oder einzelne Programmpunkte fest sind. So kann ein Ablauf, welcher immer derselbe, ist im Template problemlos übernommen werden und muss nicht jedes Mal aufs Neue vom Nutzer gesetzt werden. Da die Veranstaltungsbeschreibung dieselbe bleibt, muss der Nutzer beim Anlegen einer Veranstaltung auf Basis eines Templates nur noch den Namen und den Zeitraum angeben, sowie die nicht als fest markierten Eigenschaften ergänzen. Auch bei der Aufgabe ergeben sich Anpassungen für das Template. Der Erledigungs-Status wird automatisch auf Unerledigt gesetzt, was dem Nutzer Arbeit ersparen soll. Das Notizfeld, welches als dynamische Hilfe angedacht ist, wird ebenfalls zurückgesetzt, ebenso die Checkpunkte der Listen und die Zuweisung der Benutzer. Eine Ausnahme bildet die Zuweisung an ein Amt. Diese Zuweisung bleibt auch bestehen, wenn eine neue Veranstaltung auf Grundlage eines Templates erstellt wird. Ein essentieller Unterschied zu den in Abschnitt 2.3 genannten Tools ist die Anpassung der beiden Erinnerungen für die Dringlichkeit der Aufgabe. Während diese bei den anderen Tools konkrete Daten haben und einzeln per Hand angepasst oder alle zusammen manuell abgebildet werden müssen, wird hier die Erinnerung anders konfiguriert. Die Erinnerungen werden nicht anhand eines konkreten Datums vom Nutzer erstellt, sondern in Form einer Relation zum Veranstaltungsbeginn, wie zum Beispiel ein Monat und sieben Tage vorher. Diese Darstellung des Abstands zu der Veranstaltung anstelle eines festen Zeitpunktes ergab sich deutlich aus den Interviews und bildet eine essentielle Anforderung der Pfadfinder an ScOuT.

4.4 Implementation

Im Rahmen der Masterarbeit wurde zwar ein umfassendes Konzept für ScOuT als Hilfst-Tool für Vereine entwickelt, doch im Folgenden wird dessen Umfang auf ein Minimum reduziert, welches lediglich die großen Unterschiede zu den in Abschnitt 2.3 genannten Tools beinhaltet und somit die aus dem Interview geforderten wichtigen Anforderungen darstellt.

Bei dieser Reduktion fallen alle konzeptionellen Punkte bezüglich der Ämter und ihrer Aufgaben weg. Die Aufgaben der Ämter, welche nicht an Veranstaltungen gebunden sind, werden bereits hinreichend durch die existierenden Tools abgedeckt und können mit einfachen Aufgabenlisten unterstützt werden. Die anderen Aufgaben von Ämtern werden wie bisher mit den Aufgaben betrachtet, welche an Veranstaltungen gebunden sind.

Ebenso wird alles im Zusammenhang mit den Personas Gast, Helfer und SubOrgateam nicht implementiert, da diese Funktionen eine zu geringe Schnittmenge mit den wichtigen Anforderungen aus dem Interview darstellen.

Übrig bleiben die Funktionen und Objekte, welche mit der Persona Orgateam verbunden sind, stellt Abbildung 24 im Anhang einen Gesamtüberblick über das Ergebnis der implementierten Struktur dar. Darunter fallen nun: Der Nutzer, sein Verein, die Veranstaltungen und die Aufgaben. Durch die Verwendung von MongoDB (siehe Abschnitt 2.4) erhalten alle Objekte eine einzigartige Identifikationsnummer (eng.: unique identificationnumber; kurz uID) vom Typ *ObjectID*. Der Nutzer wird durch die Objektstruktur *User* abgebildet. Er muss sich mittels einer E-Mail-Adresse und eines Passworts registrieren und einen Anzeigenamen angeben. Während das Attribut *mail* als Loginname dient, stellt *name* den Anzeigenamen bereit, welcher anderen Nutzern, zum Beispiel im selben Verein, angezeigt wird. Der Nutzer kann beliebig viele Vereine in seinem Feld *teams* speichern. Die Vereine heißen in dem System nun *Team*, da ScOuT zwar als Ziel hat die Vereinsarbeit zu erleichtern, jedoch für jedermann offen ist und sich auch Zusammenschlüsse, wie eine Arbeitsgruppe, mit dem Begriff Team identifizieren können sollte. Jedes *Team* hat eine Liste von Mitgliedern. Darin sind alle Mitglieder mit ihrer uID gespeichert. Ebenso sind darin Nutzer gespeichert, welche noch nicht Teil des *Team* sind, sondern erst eingeladen wurden. Die Unterscheidung erfolgt mittels des *MemberStatus*. Jedes *Team* kann beliebig viele Veranstaltungen, welche mit dem Namen *Event* versehen sind, speichern. Wie in der Konzeption entwickelt, erhält *Event* verpflichtende Attribute für den Zeitraum der Veranstaltung. Jedes *Event* kennt mittels *teamref* sein zugehöriges *Team*. Zuletzt wird das Orgateam im Feld *members* mittels der uID der Nutzer abgelegt. In das Orgateam können nur Nutzer aufgenommen werden, welche in der Mitgliederliste *members* des *Team* sind und den *MemberStatus MEMBER* haben. Für jedes *Event* können wiederum beliebig viele Aufgaben gespeichert werden. Aufgaben werden in ScOuT *Task* genannt. Wie in der

Konzeption erarbeitet, wird der Dringlichkeits-Start als Abstand in Tagen, Wochen und Monaten zum Veranstaltungsbeginn angegeben und kann somit als einfache natürliche Zahlen gespeichert werden. Einfachheitshalber wird für die ersten Implementationsschritte der Dringlichkeits-Start auf den Zeitraum vor der Veranstaltung beschränkt. Auch der durch die Templates hervorgerufene Unterschied zwischen Beschreibung und Ergebnisnotiz wurde bereits durch die Attribute *description* und *note* dargestellt. Die wünschenswerten Attribute Checkliste und Anhang, sowie die Zuweisung von einzelnen Nutzern wurde für diesen Stand der Implementation weggelassen, da sie zu den Standards der existierenden Planungstools gehören und keine zu testende Neuerung darstellen. Als letztes Objekt wurde das *Template* abgebildet. Jedes *Team* kann beliebig viele *Templates* haben. Das *Template* selbst speichert durch die Reduktion aller Funktionen nur wenige wiederverwendbare Attribute eines *Event* ab, darunter die Beschreibung, die Team-Referenz und die Aufgaben. Der Zeitraum des *Event* muss für jede Instanz individuell angegeben werden, weshalb dieses Attribut nicht im *Template* übertragen wird. Auch werden alle Aufgaben übernommen. Dabei ist die Kopie des Dringlichkeits-Start der entscheidende Unterschied zu anderen Planungstools und wird zusammen mit dem Namen und der Beschreibung im *Template* gespeichert. Nicht gespeichert wird die konkrete Notiz und der Erledigungs-Status, da diese beim Erstellen eines neuen *Event* aus dem *Template* zurückgesetzt werden.

Mit dieser Implementations-Grundlage ist ScOuT fast voll funktionstüchtig und stellt die im Konzept entwickelten Unterschiede zu existierenden Planungstools dar, welche die Arbeit für Vereine verbessern sollen und in den anderen Tools nicht geboten wurden.

5 Oberflächengestaltung

Nachdem nun Schritt 2 aus Abbildung 7 im vorherigen Kapitel 4 abgeschlossen ist, folgt nun Schritt 3 im Planungszyklus von ScOuT.

Die Benutzeroberfläche (eng.: graphical user interface; kurz: GUI) wird mit dem in Abschnitt 2.4 vorgestellten Frontend Framework Ionic umgesetzt. Die GUI dient dem Nutzer als Schnittstelle mit dem Backend von ScOuT und muss deshalb alle für den Benutzer relevanten Informationen und Funktionen abbilden. Mit dem Zweck diese vollständig darstellen zu können, werden verschiedene Usability-Methoden zur Oberflächengestaltung angewendet, bei welchen das Ziel ist, die Gebrauchstauglichkeit (siehe Kapitel 3) zu erreichen.

Im folgenden Verlauf wird immer wieder der Unterschied von Informationen und Funktionen beachtet [Ham06]. Diese Trennung der Oberflächenelemente beruht auf ihrem unterschiedlichen Zweck, dem Nutzer zu dienen. Funktionen werden dabei durch sogenannte Bedienelemente dargestellt und dienen der Dialogkontrolle. Informationen dagegen dienen rein der Präsentation des Informationsflusses.

5.1 Oberflächenelemente

Durch die Nutzung von Ionic als Frontend Framework gibt es bereits eine grob vordefinierte Struktur und Komponenten (siehe Dokumentation von Ionic²²), um die Oberfläche mit Elementen auszustatten. In Ionic wird die Oberfläche Page genannt. Jede Page erhält eine einheitliche Struktur, welche in Abbildung 10 a) dargestellt wird. Diese entsteht auf Grundlage des `IonHeader`, welcher ausnahmslos immer als Balken am Kopf der Page dargestellt wird und dem analogen `IonFooter` am unteren Ende der Page. Dazwischen ist der `IonContent`.

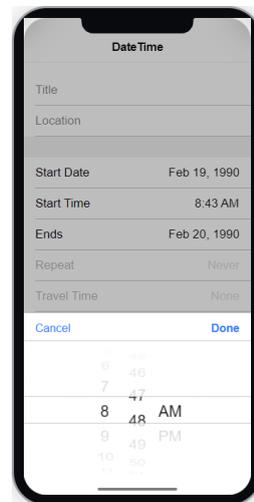
Die im Anhang abgelegte Abbildung 25 verschafft einen Überblick über die Oberflächenelemente, welche für die Gestaltung des `IonContent` verwendet wurden. Diese wurden zur besseren Umsetzung der Gebrauchstauglichkeit in vier Kategorien eingeteilt und werden im Einzelnen nachfolgend aufgelistet [Ham06].

Zuvor jedoch noch generelle Anmerkungen zu Ionic. Die Elemente von Ionic können fast beliebig ineinander verschachtelt werden. `IonCardSubtitle` ist also nicht ausschließlich an die `IonCard` gebunden, auch wenn der Name dies vermuten lässt. Strikte Verbindungen, wie zum Beispiel die `IonFloatingActionButton` mit dem Element `IonFabList`, wurden deshalb in der Abbildung 25 mit Assoziationen verdeutlicht. Des Weiteren gilt, dass Ionic nicht mit Enumeration-Klassen arbeitet, sondern mit einer vordefinierten Liste von Strings.

²²<https://ionicframework.com/docs/components/>



(a) Page: Basis Struktur



(b) IonDateTimePicker Beispiel

Abbildung 10: Ionic-Page Struktur und Beispiel für IonDateTimePicker

Strukturelemente

Die Strukturelemente sind Ionic Komponenten, welche zur Anordnung der anderen Komponenten dienen.

Es ist darauf hinzuweisen, dass wie in Abbildung 25 im Anhang zu sehen ist, die Komponenten `IonButtons` und `IonButton`-Elemente sich unterscheiden und sind und beim Lesen auf das Merzhahl „s“ geachtet werden muss.

- Das Element `IonButtons` ist dafür da eine Liste mit `IonButton` darzustellen. Dieses Strukturelement wird immer in einer `IonToolbar` verwendet. Die `IonToolbar` ist dafür da `IonButtons` relativ zur Seite zu positionieren und die `IonButton` innerhalb von `IonButtons` nach Regeln anzuordnen. Die `IonToolbar` hat eine rechteckige Form, welche waagrecht über die gesamte Breite der Oberfläche geht.
- Der `IonFloatingActionButton` (kurz: Fab) soll einen besonderen Button strukturieren. Dabei sorgt das Element `IonFloatingActionButton` ähnlich der `IonToolbar` dafür, den Fab räumlich anzuordnen. `IonFabList` hat dieselbe Funktion wie `IonButtons` und listet eine Menge von `IonButton`-Elementen auf, welche zu dem Fab gehören.
- Die `IonCard` ist eine zweiteilige Struktur. Sie hat einen Kopf, welcher mit `IonCardHeader` dargestellt wird, und einen Körper mit `IonCardContent`. Die `IonCard` hat dabei eine rechteckige Form mit abgerundeten Ecken. Die Hintergrundfarbe von Kopf und Rumpf können getrennt angesteuert werden.

- Die `IonList` ist ein Strukturelement, welches `IonItem` beinhaltet und diese senkrecht anordnet. Das `IonItem` selbst dagegen ist als Strukturelement betrachtet eine Liste, welche alle darin enthaltenen Komponenten waagrecht anordnet. Beide Elemente werden als ein Bereich sichtbar, welcher nach unten durch eine Linie abgetrennt wird. Diese Linie kann mittels der Enumeration `Line` verschieden stark ausgeprägt sein oder sogar entfallen.

Kontrollelemente

Kontrollelemente sind Ionic Komponenten, welche auf „Click()“-Events reagieren und Funktionen ausführen, wie den Wechsel der Page oder die Kommunikation mit dem Backend.

- Das eben erwähnte `IonItem` ist die einzige Komponente, welche sich zwei Kategorien zuordnen lässt. Neben seiner Struktureigenschaften kann das `IonItem` durch die Eigenschaft `detail` zu einem anklickbaren Kontrollelement werden. Dabei wird zum Aussehen des `IonItem` eine kleine, offene Pfeilspitze auf der rechten Seite ergänzt.
- Der `IonFabButton` hat nur die Funktion, die in der `IonFabListe` angelegten `IonButton` in die Richtung des Attributes `side` auszuklappen und somit ein verstecktes Menü einzublenden. Sowohl der `IonFabButton`, als auch die `IonButton`-Elemente in der `IonFabListe` haben immer eine Kreisform und bestehen aus einem `IonIcon` ohne zusätzlichen Text.
- Der `IonButton` (außerhalb der `IonFabList`) dagegen stellt einen Standardbutton da, welcher eine rechteckige Form hat und sowohl durch `IonIcon` als auch Text beschriftet werden kann.

Eingabelemente

Die Eingabelemente sind in Ionic Komponenten, in welchen der Nutzer freie Texte eingeben oder aus einer vordefinierten Liste von Optionen auswählen kann.

- `IonInput` stellt dabei das klassische Textfeld dar, in welchem der Nutzer Texteingaben machen kann. In Ionic kann vorgegeben werden, welcher Typ der Eingabe gemacht werden kann.
- Der Nutzer kann in `IonSelect` aus einer Liste von festgelegten Möglichkeiten auswählen. Diese Liste kann beliebig groß sein und wird als Text dargestellt.
- Dagegen kann bei dem Eingabelement `IonToggle` nur zwischen zwei Möglichkeiten ausgewählt werden, welche das als Schalter dargestellte Element nach rechts oder links schieben, um „an“ oder „aus“ zu simulieren.

- Der `IonDateTimePicker` ermöglicht es den Nutzer die Uhrzeit und das Datum auszuwählen. Dafür wird dieses Eingabeelement als eine Menge von Zahlenrädern, ähnlich dem Spielautomaten „Einarmiger Bandit“, dargestellt, welche mittels Scrollen angepasst werden kann. Ein Beispiel dafür ist in Abbildung 10 b) zu sehen.

Präsentationselemente

Präsentationselemente sind Ionic Komponenten, welche ausschließlich zur Darstellung von Informationen dienen.

- Das `IonIcon` ist dabei das einzige Element, welches Bilder darstellt. Ionic bietet dafür eine Vielzahl vordefinierter Icons²³.
- `IonLabel`, `IonCardTitle` und `IonCardSubtitle` sind Elemente zur Textdarstellung. `IonCardSubtitle` wird in einer kleineren Schrift dargestellt als `IonLabel` und `IonCardTitle` und ist standardmäßig in dunkelgrauer Schrift. Die anderen beiden sind dagegen in schwarzer Schrift.

5.2 Anwendung von Usability-Methoden

Zur Durchsetzung der Eigenschaften der Gebrauchstauglichkeit wurden verschiedene Methoden in der Konzeption angewendet. Die Methoden befassen sich dafür mit verschiedenen Bausteinen der GUI. Darunter fällt zum einen die Dialogkontrolle. Zum anderen bestimmen die Methoden die Werte der Eigenschaften der GUI-Elemente. Zu den Eigenschaften zählen räumliche Positionen, Größen, Farben und mehr. Schließlich sind genau diese Elemente und ihre Eigenschaften das, was eine Oberfläche ausmacht. Usability-Methoden arbeiten folglich mit ihnen und passen sie so an, dass die Gebrauchstauglichkeit erreicht wird.

Lautes Denken

Um den Bezug zum Nutzer im Laufe der Entwicklung nicht zu verlieren, wurden zu mehreren Zeitpunkte einzelne Nutzer hinzugezogen und eine Evaluation von ScOuT in Form von „Lautem Denken“ (engl. Thinking Aloud) durchgeführt. Diese Methode ermöglicht es frühzeitig Probleme zu erkennen, die während der Bedienung auftreten und den Grad der Gebrauchstauglichkeit reduzieren. Dabei bekommt der Nutzer eine Aufgabe gestellt, welche er mit Hilfe von ScOuT durchführen soll. Während er die Aufgabe bearbeitet, spricht er Gedanken, Meinungen und Gefühle laut aus. Diese werden protokolliert. Anschließend können die Protokolle dafür verwendet werden Denkprozesse nachzuverfolgen und Probleme zu identifizieren [BN13].

²³<https://ionicons.com/>

Grobe Oberflächeneinteilung

Die Einteilung der Oberfläche sollte nicht nur konsistent sein, sondern auch bestimmten Regeln folgen. Eine grundlegende Konsistenz wird alleine dadurch erreicht, dass wie in Abbildung 10 in Abschnitt 5.1 Ionic die Oberfläche in Kopf, Körper und Fuß einteilt. Diese Einteilung kann für Status-, Arbeits- und Softkey-Bereiche genutzt werden [Ham06]. Softkeys sind digitale Buttons, welche im Gegensatz zu analogen Schaltern sogenannten Hardkeys stehen. Der Statusbereich soll Hinweise über den Systemzustand mitteilen. In der Ionic-App wird dort ein Titel angegeben, welcher nach Möglichkeit den Inhalt der Page zusammenfasst. So wurden Überschriften gewählt wie „Deine Teams“ für eine Liste aller Teams, in welchen der Nutzer Mitglied ist und „Event: {Eventname}“ für die Einzelansicht einer Veranstaltung. Ebenso hat nicht jedes Smartphone einen Hardkey, welcher die „Zurück“-Funktion abbildet, weshalb dieser Hardkey ebenfalls fest in der groben Oberfläche eingeplant werden sollte. Der Hardkey für die „Zurück“-Funktion wird oben rechts im Statusbereich platziert, da seine einzige Funktion ist die Page zu wechseln und somit den Status zu ändern. Der Softkey-Bereich dagegen wird im Fußbereich durch ein festes Menü dargestellt. Durch dieses dauerhaft verfügbare Navigationsmenü erhält der Nutzer ein erhöhtes Gefühl von Sicherheit und Kontrolle. Da auf einem Smartphone-Bildschirm nicht genügend Platz ist, um alle wichtigen Navigationsfunktionen mit Buttons abzubilden, wird an letzter Stelle im Fußbereich ein erweitertes Menü mittels eines `IonFloatingActionButton` integriert. Der Arbeitsbereich benötigt den größten Platz, da dort flexibel die meisten GUI-Elemente angeordnet werden, um für die Page spezifische Informationen und Funktionen anzubieten.

Anordnung der Informationen und Funktionen

Obwohl der Begriff Arbeitsbereich zunächst vermuten lässt, dass dort nur aktive Elemente für den Nutzer liegen, um Aufgaben abzuarbeiten, umfasst er nicht nur Funktionen, sondern auch Informationen. Dort wird folglich alles präsentiert, was der Nutzer zur Bedienung von ScOuT benötigt [Ham06]. Für die Darstellung von Informationen werden die in Abschnitt 5.1 genannten Ionic Komponenten vom Typ Präsentationselement verwendet. Funktionen werden mittels einer Kombination aus Eingabeelementen und Kontrollelementen implementiert. Die Strukturelemente dienen der Anordnung der Funktionen und Informationen. Für eine gebrauchstaugliche Anordnung müssen jedoch gleich mehrere Regeln eingehalten werden.

Eine der Regeln besagt, dass Informationen und Funktionen, welche sinngemäß zusammen gehören, auch zusammen angeordnet werden und, welche voneinander getrennt sind, auch räumlich zu trennen sind. Dabei können verschiedene Gründe dazu führen, dass Informationen und Funktionen zusammen gehören. Es hängt dabei zum Teil davon ab, welche Informationen und Funktionen auf einer Page zusammengefasst werden. Ein Kriterium zur

nahen Anordnung kann zum Beispiel sein, dass Bedienelemente, welche von der Seite weg navigieren, aufgrund ihrer Funktionalität zusammen gehören. Während Bedienelementen, welche die Seite nicht wechseln ebenfalls zusammengehören, jedoch von wegführenden Bedienelementen zu trennen sind. Innerhalb dieser Gruppierung können jedoch wieder andere Gründe für eine Trennung von Funktionen auftreten. So kann es sein, dass ein Bedienelement nur mit mehr Informationen dem Nutzer seine Funktionalität übermitteln kann. Diese Funktion benötigt dann gegebenenfalls mehr Platz und grenzt sich durch Eingabelemente und Präsentationselemente durch die reine Form der Kontrollelemente ab, wodurch eine räumlich getrennte Anordnung dieser Funktion zu anderen sinnvoll erscheint. Doch nicht nur die Gruppierung von Informationen und Funktionen spielt eine Rolle beim Design der Oberfläche, sondern auch die Anordnung der Gruppen. Für den Nutzer wichtige Gruppen sollten schneller erreichbar sein und somit eher im direkten Sichtfeld positioniert werden als unwichtigere Gruppen. Gruppen werden dann wichtig, wenn sie zum Beispiel häufiger benötigt werden oder wenn das System eine Eingabe fordert, um weiter verfahren zu können. Zuletzt gilt noch bei der Anordnung von Informationen und Funktionen zu beachten, dass eine Überfüllung der Page mit diesen nicht förderlich ist. Dies beeinflusst die Übersichtlichkeit einer App-Seite [Ham06].

Feedback und Hilfe

Alle der in Kapitel 3 genannten Eigenschaften beschäftigen sich stark mit dem Thema der leichten Erlernbarkeit der Systemkonzeption und dem Konzept der Hilfe, sollte es zu Problem kommen. Dabei ist eine Abhängigkeit der Eigenschaften zu erkennen. Ist das System selbstbeschreibend, dann kann der Nutzer es leichter erlernen.

Nun gilt es also das System und die Oberfläche so zu gestalten, dass der Nutzer möglichst alle Informationen für die Bedienung zur Verfügung gestellt bekommt. Dafür wird dem Nutzer nach seiner Registrierung zuerst ein Tutorial angeboten, welches besonders bei den Tests des Lauten Denkens viel Anklang fand. Doch nicht nur zu Beginn der Erstanwendung, sondern auch zu jedem späteren Zeitpunkt sollte das System den Nutzer unterstützen ScOuT zu verstehen. Es soll, wie von Nielsen gefordert, den Systemzustand sichtbar machen. Für den allgemeinen Systemzustand wurde bereits der Page-Title im `IonHeader` eingeführt. Die Seiten zeigen dann im `IonContent` Informationen über die in der Datenbank liegenden Objekte. Doch auch Teilzustände des Systems müssen dem Nutzer mitgeteilt werden [Ham06]. Besonders wichtig für den Nutzer ist es zu erfahren, wenn sich ein Zustand ändert. Dies erfolgt, wenn der Nutzer selbst über das Backend auf die Datenbank zugreift. Deshalb wurde bei Buttons, welche Funktionen beinhalten, die die Datenbank verändern, ein Hinweisfeld ergänzt. Dieses informiert den Nutzer über eine erfolgreiche Änderung. Das Feedback ist dabei möglichst nah an dem But-

ton selbst positioniert, da der Blick des Nutzers beim Klicken örtlich dort in der Nähe ist. Der Nutzer benötigt jedoch nicht nur Feedback über Änderungen, sondern auch über das Fehlen von Änderungen, wenn zum Beispiel eine Funktion aufgrund fehlerhafter Eingabe nicht erfolgt. Das Feedback in dem Hinweisfeld kann also auch dazu dienen den Nutzer darüber zu informieren, dass ein Zustand nicht geändert wurde.

Doch nicht nur direkte Informationen über das System reichen aus, um die in der ISO/IEC 25000 geforderte leichte Erlernbarkeit der Bedienung des Systems und die Lernförderlichkeit der ISO 9241-1110 zu gewährleisten. Auch ist es, trotz größter Bemühungen, nicht möglich, das System in vollen Zügen selbstbeschreibungsfähig zu machen ohne andere Designvorschriften zu verletzen. Nielsen und Vuolle definieren deshalb weitere Eigenschaften, welche sich mit zusätzlichen Hilfen befassen. Darunter fallen Anleitungen, Dokumentationen und Support. Diese sollen nach ihrer Definition für den Benutzer klar erreichbar sein. Deshalb findet sich die Hilfe in ScOuT im erweiterten Menü unter dem erwartungskonformen Icon des Fragezeichens. Nicht nur die leichte Navigation dort hin, sondern auch der Inhalt der Hilfe wird von Nielsen und Vuolle durch Eigenschaften charakterisiert. Nielsen beschreibt sie als Anleitung für den Nutzer. Dies ist bereits mit dem Tutorial gegeben, weshalb die Hilfe dem Nutzer die Möglichkeit gibt, dieses problemlos immer wieder einsehen zu können. Die Hilfe wurde zudem um eine Sitemap und ein Lexikon erweitert. Die Sitemap soll dem Nutzer dabei helfen im Falle einer Orientierungslosigkeit, Pages direkt öffnen zu können. Das Lexikon gibt dem Nutzer eine Übersicht über Begriffe und deren Erläuterung, die im Zusammenhang mit dem Aufgabenzweck der Anwendung stehen. Gerade die Begriffe werden auch im Tutorial erklärt, doch dort ist die Begriffsklärung nur ein Teil, um die damit verbundenen Funktionsaufgaben zu erledigen. Deshalb wurden diese Begriffsdefinition noch einmal extrahiert und konzentriert.

Zusammen mit dem Feedback über aktuelle oder geänderte Systemzustände und den zusätzlichen Hilfen sollte nun ein gewisser Grad an Selbstbeschreibungsfähigkeit des Systems vorhanden sein und die Erlernbarkeit für den Nutzer einfacher möglich sein.

Eingabe und Fehler

Neben dem großen Thema Feedback und Hilfe stellen Fehler einen weiteren Punkt dar, welcher in mehreren Eigenschaften in Kapitel 3 genannt wird. In Bezug auf die Fehler müssen nun mehrere Schritte unterschieden werden:

1. Fehlerursache: Fehler können beim Navigieren erfolgen, indem der Nutzer auf die falschen Kontrollelemente klickt, oder bei der Bestätigung einer falschen Eingabe.

2. Fehlervermeidung: Nielsen besagt, dass das System dabei helfen soll Fehler zu vermeiden. Gut beschriftete Kontrollelemente können dabei helfen falsche Navigationen zu umgehen. Mit gut ist dabei gemeint, dass der Nutzer die Beschriftung, egal ob durch Text oder Icon, verstehen oder reproduzieren kann und keine Missverständnisse auftreten. Durch einheitliche Beschriftung, konsistenten Aufbau und eindeutige Symbole, welche sogar im Tutorial erklärt werden, ist dies problemlos zu verwirklichen. Gerade bei der Wahl der Icons in der Ionic-App wurde deshalb Wert darauf gelegt, dass der Nutzer sie versteht, da einige Buttons nur aus dem Icon bestehen und keinen zusätzlichen Text haben. Die Fehlervermeidung von falschen Eingaben ergibt sich auf demselben Weg wie bei der Navigation, indem dem Nutzer immer deutlich gemacht wird, was von ihm erwartet wird und was ihn erwartet. Das heißt, dass alle Eingabefelder beschriftet sind und dem Nutzer gesagt wird, was dort einzugeben ist. Im Falle der Registrierung, bei welcher es Uneindeutigkeiten zwischen Nutzernamen und Anzeigenamen gibt, wurden sogar zusätzliche Hinweise angegeben.
3. Fehlererkennung: Nielsen besagt, dass das System dabei helfen soll Fehler zu erkennen. Im Falle einer falschen Navigation wird der Nutzer selbst erkennen, dass er die Page verlassen hat und auf welcher Seite er sich befindet, da dies durch die Titel im **IonHeader** immer mit dem Nutzer kommuniziert wird. Zusätzlich gibt der Seiteninhalt im **IonContent** Aufschluss darüber, wo der Nutzer sich gerade befindet. Bei einer falschen Eingabe erkennt das System diese und informiert den Nutzer mittels einer Fehlermeldung. Der Nutzer muss hier also keine Eigenleistung dafür erbringen.
4. Fehlerbehebung: Sowohl die ISO 9241-110, als auch Nielsen besagen klar, dass die Fehlerbehebung dem Nutzer so wenig Arbeit wie möglich machen soll. Um dies zu erreichen, sind die Fehlermeldungen so detailliert, dass direkt aus ihnen heraus zu lesen ist, was der Fehler ist und der Behebungsschritt für den Benutzer eindeutig wird. Fehlermeldungen wie „Die Eingabe ist falsch“ werden also ersetzt durch „Die Eingabe des Teamnamen überschreitet die maximale Länge von 30 Zeichen.“ oder „Du musst einen Namen für den Task angeben.“. Dem Nutzer wird also direkt gesagt, was zu tun ist, um den Fehler zu beheben. Durch das Abfragen dieser falschen Eingaben wird die Änderung der Datenbank-Sätze über das Backend abgebrochen. Es muss also nicht ein falscher Datensatz gesucht und gelöscht werden, sondern der Vorgang wird vom System für den Nutzer unterbrochen, was wiederum der Fehlervermeidung zu Gute kommt. Die Fehlerbehebung im Falle einer falschen Navigation lässt sich jederzeit über den Zurück-Button im **IonHeader** korrigieren.

Terminologie

Im Zusammenhang mit dem zuvor erwähnten Lexikon in der Hilfe wurden Entscheidungen bezüglich der Terminologie getroffen. Die Terminologie ist eines von vielen Dingen auf der GUI, die einheitlich sein soll und somit den Eigenschaften der Beständigkeit von Nielsen und der Erwartungskonformität der ISO 9241-110 im Kapitel 3 zu Gute kommt. Neben dieser Anforderung an die Kontinuität wird zudem die Wahl der Begriffe von zusätzlichen Eigenschaften zur Erreichung der Gebrauchstauglichkeit beeinflusst. Nielsen sagt, dass das System mit der Wirklichkeit übereinstimmen soll. Ebenso sollen die Begriffe den Nutzer nicht daran hindern das Konzept und den Dialog zu verstehen und zu erlernen. Deshalb sollte bei der Wahl darauf geachtet werden, dass die Begriffe nah am Alltagsgebrauch sind und von den Nutzern verstanden werden können.

Wie in Abschnitt 4.4 bereits für das Backend und die Datenbank eingeführt, werden die Begriffe Verein, Veranstaltung und Aufgabe auch auf der Oberfläche der App durch die Begriffe Team, Event und Task ersetzt. Der Begriff „Team“ ist dabei umfassender als Verein. Würde es weiterhin Verein heißen, könnten Familiengruppen, Arbeitsgruppen oder andere Gruppierungen sich ausgeschlossen fühlen. Team ist zudem ein gängiger Begriff in anderen Tools, wie in Abschnitt 2.3 zu sehen ist. Aus demselben Grund wurde Aufgabe zu Task umbenannt, da dieser Begriff den Nutzern bereits bekannt und vertraut sein sollte. Da es in den anderen Tools keine Veranstaltungen in dem Sinne gibt, sondern Projekte geplant werden, muss hier ein eigener Begriff gewählt werden. Der Begriff Veranstaltung ist dabei zu lang und der bei den Pfadfindern übliche Bezeichner Aktion nicht allgemeingültig genug. Deshalb wurde sich an dem gängigen Wort Eventplanung orientiert und Veranstaltungen werden auf der GUI als Event bezeichnet.

Designentscheidungen bezüglich Ionic

Einzelne Entscheidungen zur Oberflächengestaltung entstanden weniger auf Grundlage von den in Kapitel 3 vorgestellten Usability-Eigenschaften oder Methoden zur Umsetzung dieser, sondern aus den Möglichkeiten, welche Ionic bietet.

So wurde die **IonCard** als Stilmittel ernannt, um innerhalb einer Page grundlegende Strukturen zu erschaffen. Die **IonCard** dient dadurch der Gliederung der Informationen und Funktionen auf einer Page. Obwohl eine Page bereits nur Informationen enthalten sollte, welche sinngemäß zusammen gehören, können diese dennoch feiner gruppiert werden. Um diese Gruppen auf der Page unterscheiden zu können, erhält jede Gruppe eigene **IonCard**. Ein Beispiel hierfür ist eine Page, welche alle Teams eines Nutzers auflistet. Diese Teams können in zwei Gruppen geteilt werden. Zum einen die Teams, in welchen der Nutzer Mitglied ist, und zum anderen die Teams, zu welchen er eine Einladung erhalten hat und noch kein Mitglied ist. Die **IonCard**

wurde für diese Strukturierung den anderen Strukturelementen vorgezogen, da sie Listen-unabhängig ist und sowohl Funktionen als auch Informationen zusammen darstellen kann. Die anderen Strukturelemente sind entweder an Buttons oder an eine Liste von `IonItem` gebunden. Deshalb wird im Folgenden die `IonCard` als Strukturkarte bezeichnet.

Eine weiteres Stilmittel wurde mittels der verschiedenen Kontrollelemente eingeführt. Der `IonButton` wurde für beliebige Funktionen verwendet, welche jedoch nicht Teil einer dynamischen Liste von Funktionen sind. Dabei wurde nicht unterschieden, ob er Funktionen zum Öffnen einer anderen Page beinhaltet oder Funktionen zur Kommunikation mit dem Backend. Das `IonItem` mit einem Detail-Pfeil wird dagegen verwendet, wenn eine dynamische Liste zum Öffnen anderer Pages verwendet wird. Hier ist die Funktion klar auf das Öffnen beschränkt und beinhaltet nie die Interaktion mit dem Backend. Ein weiterer Unterschied der Nutzung besteht darin, dass der `IonButton` ausschließlich als beschrifteter Text einen funktionsbeschreibenden Namen wie beispielsweise „Speichern“ trägt, während dieser bei dem `IonItem` entfällt, da die Funktion immer Öffnen ist. Dagegen hat das `IonItem` die Möglichkeit mehr Informationen Preis zu geben, welche die anschließend zu öffnende Seite betreffen. Dort werden zum Beispiel Informationen wie das zugehörige Team von einem Event angegeben. `IonItem`, welche nicht Teil einer dynamischen Liste sind sondern einer statischen, werden meist mit einem `IonIcon` gekennzeichnet. Auf eine statische Liste von `IonButton` anstelle der `IonItem` mit `IonIcon` wurde verzichtet (siehe nachfolgender Paragraph „Farben“). Das `IonItem` mit Detail-Pfeil wird im Folgenden mit dem kürzeren Namen Detail-Item bezeichnet.

Farben

Neben der Anordnung von GUI-Elementen und dem Inhalt dieser, gibt es noch die Eigenschaft der Farbe einer Komponente, welche beachtet werden muss. Damit befasst sich ein weiteres Unterkapitel der ISO 9241. Die ISO 9241-8 zur Anforderung an Farbdarstellung (mittlerweile ersetzt und unterteilt in ISO 9241-302, ISO 9241-303 und ISO 9241-305) [DIN97] stellt dafür verschiedene Regeln auf. Diese besagen zum einen, dass maximal sechs verschiedene Farben neben Schwarz und Weiß verwendet werden sollen, damit die Informationsaufnahme nicht durch komplexe Farbmuster erschwert wird. Ebenso sollten Farben jederzeit dieselbe Bedeutung haben, damit der Nutzer nicht darin gehindert wird das Konzept des Systems zu verstehen und für ihn die Orientierung erleichtert wird. Die Wahl der Farbe soll sich dabei nach Bedeutungen aus der Alltagswelt richten, um den Nutzer nicht durch unerwartet fehlerhafte Interpretationen zu verwirren.

Wie in der Abbildung 25 im Anhang gezeigt, bietet Ionic bereits eine Reihe von neun vordefinierten Farben. Obwohl die Farbe `light` Weiß entspricht, ist die Liste immer noch um eins länger, als in der ISO 9241-8 gefor-

dert. Deshalb werden nicht alle vordefinierten Farben von Ionic verwendet. Die dunkelblaue Farbe `primary` wurde ausschließlich als Hintergrundfarbe für die Kopf und Fußzeile eingesetzt. Sie soll einen Rahmen für ScOuT bilden, welcher sowohl einem ästhetischen Zweck als auch Kontinuität dient. Die im `IonFooter` gezeigten Funktionen sind jederzeit verfügbar. Das Hellblau der Farbe `secondary` zeigt dem Nutzer Interaktionsmöglichkeiten mit dem System an. Zu Beginn der Konzeption der Oberfläche wurde aufgrund der Einheitlichkeit diese Farbe sowohl für Box-Buttons, als auch Detail-Items verwendet. Bereits beim Lauten Denken stellte sich heraus, dass eine zu große Menge an hellblauer Flächen das Auge schnell überforderte und Informationen der Detail-Items schlechter wahrgenommen werden konnten. Deshalb wurden die großen Flächen von `IonButton` in der Farbe `secondary` größtenteils aus dem Detail-Item entfernt. Kleine Flächen der `IonIcons` dagegen bleiben in dem Detail-Item. Die restlichen Hintergründe wurden in hellgrau und weiß gehalten, was den Ionic Farben `light` und `medium` entspricht. Dabei wurden lediglich die `IonCard-Header` in Hellgrau gefärbt, um sie farblich dezent abzuheben. Es stellte sich beim Lauten Denken heraus, dass das Auge nun leichter mehrere aufeinander folgende Karten unterscheiden konnte. Bisher wurden, wenn Weiß nicht dazu gezählt wird, drei von sieben Farben für den Hintergrund verwendet, wodurch die Möglichkeit für vier weitere bei der Schrift bleiben. Ionic selbst verwendet für helle Hintergründe wie weiß und hellgrau automatisch die Farben schwarz und dunkelgrau und für farbige Hintergründe in den Farben `primary` und `secondary` eine weiße Schrift. Dadurch ist automatisch ein hoher Kontrast zwischen der Schriftfarbe und der Hintergrundfarbe gegeben, weshalb daran nichts geändert wurde. Lediglich die Farbe `danger`, welche in einem intensiven Rot von Ionic angeboten wird, wurden genutzt, um verschiedene Texte einzufärben. Darunter fallen die Texte, welche dem Nutzer aufgrund ihrer Dringlichkeit direkt ins Auge fallen sollen. Dabei werden zum einen Tasks rot hervorgehoben, welche dringend erledigt werden sollen, sowie Feedbackhinweise, welche eine Fehlermeldung an den Nutzer übermitteln.

5.3 Page-Einteilung und Navigation

Nachdem verschiedene Styleguids mit Hilfe der Methoden definiert wurden, ist der erste Schritt in der Entwicklung der Oberfläche die Einteilung der App-Seiten. Ziel dieser Unterteilung ist es zusammengehörende Informationen und Funktionen kompakt darzustellen und von anderen zu trennen. Dies dient dem Zweck, dass der Nutzer sich besser orientieren kann und fördert die in Abschnitt 3.2 genannten Eigenschaften Bedienbarkeit, Erlernbarkeit und Verständlichkeit. Zum Erreichen dieser Eigenschaften werden die Informationen anhand von Zusammenhängen in ihrer Objektstrukturen in Kategorien eingeteilt. Jede dieser Kategorien stellt anschließend eine eigene Page dar. Da Informationen auch mehreren Kategorien angehören können, sind sie oft

auch auf mehreren Pages zu finden. Daraus ergibt sich folgende Liste an Pages:

Home Umfasst Informationen: Willkommensnachricht (uneingeloggter Nutzer); zusammenfassende Account Informationen (eingeloggter Nutzer)

Account Informationsdarstellung über den Account des Nutzers

Teamliste Liste aller Teams des Nutzers

Teamansicht Informationen zu einem einzelnen Team

Eventliste Liste aller Events des Nutzers; kann nach einem Team gefiltert werden

Eventansicht Informationen zu einem einzelnen Event

Taskliste Liste aller dringender (Task ist unerledigt; Reminder ist aktiv) und offener (Task ist unerledigt; Reminder ist inaktiv) Tasks des Nutzers; kann nach einem Event oder nach einem Team gefiltert werden

Taskansicht Informationen zu einem einzelnen Task

Hilfe Informationen, welche dem Nutzer helfen sollen ScOuT zu verstehen und zu nutzen

Einige der Pages zum Darstellen von Informationen sind jedoch mit Informationen überfüllt und teilweise auch mit zu vielen Funktionen bestückt. Deshalb werden diese weiter in Subpages unterteilt. Darunter fallen sowohl die Teamansicht, welche um eine Page für eine Teammitgliederliste erweitert wird, als auch die Eventansicht, welche ebenfalls eine Subpage für die Mitgliederliste des Orgateams, als auch eine Subpage für alle Task eines Events unabhängig von ihrem Dringlichkeits- und Erledigungs-Status auflistet. Die meisten Unterteilungen erhält die Page Hilfe, welche in eine Nutzungsanleitung, eine Begriffsübersicht und eine Sitemap unterteilt wird. Ebenso gibt es in der Hilfe vier Subpages, welche die Nutzungsanleitung noch einmal in ein Tutorial unterteilen.

Des Weiteren gibt es Pages, welche nicht zur Informationsdarstellung, sondern rein zum Anbieten einer konkreten Funktionen benötigt werden. Darunter fallen: Registrierung, Login, Erstellung eines Teams, eines Events und eines Tasks.

Da darauf geachtet wird, dass eine Page nicht mit Funktionen überfüllt wird, steigt die Anzahl der Pages und wird mit weiteren Strukturen und Funktionen auch weiterhin mehr werden. Deshalb sollte bei der Aufteilung der Pages auch ihre Erreichbarkeit im Blick behalten werden [Ham06]. Dabei gilt es nicht nur zu prüfen, ob jede Page erreicht wird, sondern wie weit Wege sind, um von A nach B zu gelangen und somit die Komplexität der Navigation zu bestimmen [Rau92]. Abbildung 26 im Anhang stellt dabei die

wichtigsten Übergänge dar. Zur Vereinfachung des Diagramms wurden einige Übergänge weggelassen und die Pages farblich sortiert. In der Abbildung sind Pages im Zusammenhang mit dem Team in Rot markiert, Pages bezüglich des Events mit Blau und alles im Zusammenhang mit Tasks in grün. Die restlichen Pages wurden in der Abbildung gelb angegeben. In eckigen Kästen, da sie keine Page sind, sind zudem das Menü und das Menu im Fab dargestellt.

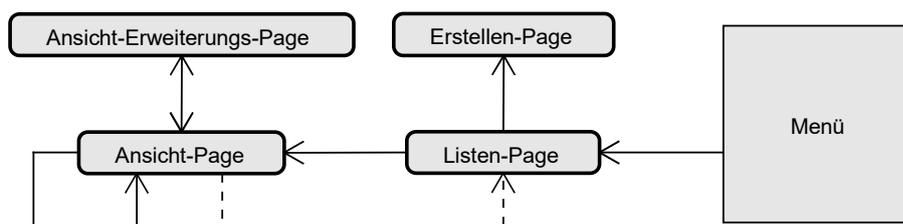


Abbildung 11: Vereinfachung der Navigationsübersicht in Abbildung 26 aus dem Anhang.

Das Diagramm stellt Folgendes dar: Das Menü in dem `IonFooter` ist im eingeloggten Zustand von jeder Page erreichbar. Diese Verbindungen wurden der Übersichtlichkeit halber weg gelassen. Über das Menü können die Seiten Teamliste, Eventliste und Taskliste geöffnet werden, ebenso die Startseite und das FloatingMenü. Die Startseite und das FloatingMenü können außerdem im nicht eingeloggen Zustand erreicht werden. Während die Registrierung nur über die Login-Page und die Startseite erreicht werden können, kann der Login selbst über alle Seiten erreicht werden, welche nur im eingeloggten Zustand eingesehen werden können. Zwischen den Seiten für Teams, Events und Tasks sind zwei Navigationsarten dargestellt. Die eine erfolgt in beide Richtungen und wird deshalb über eine Verbindungen mit Pfeilspitzen an beiden Enden dargestellt. Die andere öffnet eine gefilterte Variante der Page und wird mit gestrichelten Linien markiert. Im Allgemeinen kann gesagt werden, dass alle Listen eine Ansicht öffnen können. Ebenso können alle Ansichten ihre Erweiterungen öffnen, wie die Eventansicht die Orgateam-Page, und alle Erweiterungen können zurück zur Ansicht schalten. Aus den Ansichten können zudem wieder gefilterte Listen geöffnet werden oder andere Ansichten, wie das zu dem Event gehörende Team. Die soeben beschriebene Navigationsstrategie bleibt also innerhalb der Page-Kategorien konsistent, wie in Abbildung 11 nochmal vereinfacht dargestellt.

5.4 App-Seiten im Detail

Im Folgenden werden einige App-Seiten im Einzelnen betrachtet und die Designentscheidungen erläutert. Dafür werden die Informationen der Seiten

genauer aufgeschlüsselt und die Zusammenhänge zu den Funktionen dargestellt, um so eine Anordnung nach den in Abschnitt 5.2 dargestellten Regeln zu erreichen.

Teamliste

Die Teamliste gibt dem Nutzer eine Übersicht über alle Teams, welche ihn betreffen. Innerhalb dieser Liste gibt es zwei untergeordnete Informationskategorien. Die eine beinhaltet Teams, in welchen der Nutzer Mitglied ist, und die andere Teams, zu welchen der Nutzer eingeladen wurde. In der Strukturkarte der Teams, in denen er Mitglied ist, erhält der Nutzer in einem kompakten Überblick mehr Informationen über das Team. Neben dem Namen des Teams wird dort die Anzahl der anstehenden Events und dringenden Tasks angezeigt, da diese Informationen den Nutzer am meisten interessieren dürften. Die Team-Einladung enthält lediglich die Information, ob es eine Einladung gibt und für welches Team sie ist. Dort werden dem Nutzer die zugehörigen Funktionen der Annahme und Ablehnung der Einladung angeboten. Eine letzte Funktion auf dieser Seite ist das Erstellen neuer Teams.

Obwohl diese Seite als Hauptziel die Auflistung aller Teams hat, in denen der Nutzer Mitglied ist, sollte die Strukturkarte für ausstehende Team-Einladungen noch davor angeordnet werden. Die Funktion der Annahme oder Ablehnung einer solchen Einladung ist erfordert eine einmalige Aktion und es wird vom Nutzer erwartet, dass diese wahrgenommen wird, da das System dort auf eine Reaktion des Nutzers wartet. Es wird zudem davon ausgegangen, dass ein Nutzer relativ wenige Teameinladungen erhält und diese Strukturkarte deswegen weitestgehend in ihrer Größe kompakt bleibt und die Liste der Teams, selbst auf einem Smartphone, nicht vollständig vom Bildschirm schiebt. Dagegen ist die Information, dass ein Nutzer aktuell keine Einladung hat, relativ unwichtig und wird nur in dem seltenen Fall benötigt, wenn der Nutzer eine Einladung erwartet. Die Funktion zum Erstellen eines neuen Teams ist an keine Strukturkarte gebunden. Obwohl ihre Nutzung vermutlich deutlich geringer ausfällt, als die analogen Funktionen der Erstellung eines Events oder eines Tasks, wird sich für eine konsistente Anordnung dieser ähnlichen Funktionen entschieden. Da die Liste der Teams lang werden kann, wäre eine Platzierung der Erstell-Funktionen davor oder dahinter unpraktisch für den Nutzer und würde einen unnötig hohen Aufwand zum Scrollen mit sich bringen. Deshalb wird hier die erweiterte Fußleiste für diese Funktionen gewählt.

Basierend auf diesen Überlegungen ergibt sich das in Abbildung 12 erarbeitete Schema. Ausstehende Team-Einladungen werden an erster Stelle angezeigt, gefolgt von der Liste aller Teams, in welchen der Nutzer Mitglied ist. Erst zum Schluss folgt die Strukturkarte, falls keine Team-Einladung vorhanden ist. Die Funktion zum Erstellen eines neuen Teams wurde jederzeit bedienbar in die erweiterte Fußzeile eingefügt.

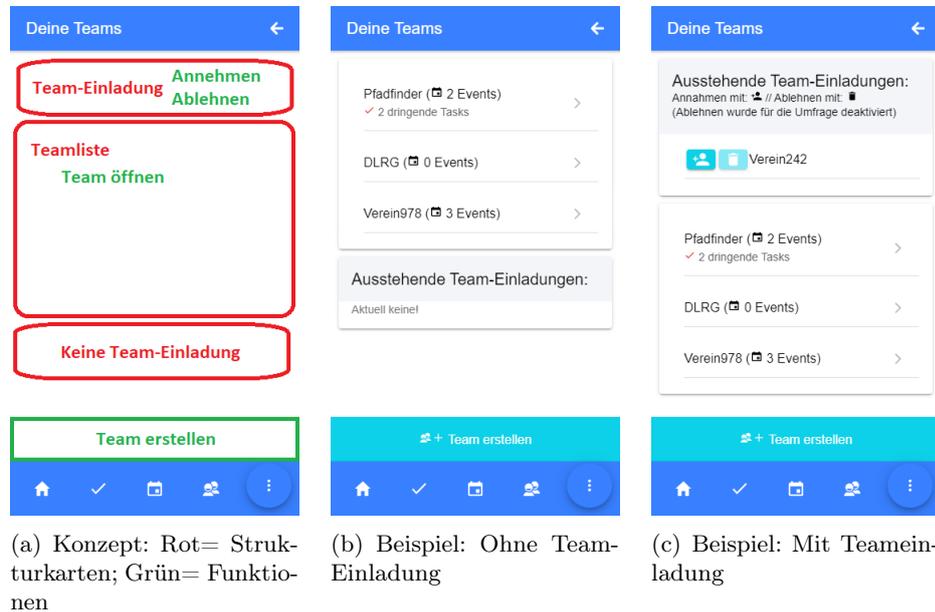


Abbildung 12: Konzept und zwei Beispiele der Page „Teamlist“

Eventliste

Zweck dieser Page ist es, dem Nutzer alle Events aufzulisten, in welchen er Mitglied des Orgateams ist. Neben dem Namen des Events wird das Datum angegeben, an dem dieses Event beginnt, das zugeordnete Team, sowie die Anzahl der dringenden und offenen Tasks. Hier sind es folglich mehr zusätzliche Informationen als bei der Teamliste. Der Eventbeginn ist entscheidend, um dem Nutzer einen schnelleren Überblick geben zu können, wie nah welches Event ist und ihm somit die Möglichkeit zu bieten seine Prioritäten danach zu richten. Das Team wird angegeben, falls es vorkommt, dass in verschiedenen Teams Events geplant werden, welche den selben Namen tragen und der Nutzer sie dadurch trotzdem korrekt zuordnen kann. Während dringende Tasks eigentlich überall angezeigt werden, da sie aufgrund ihrer Dringlichkeit für den Benutzer besonders hohe Relevanz haben, werden hier jedoch zusätzlich offene Tasks angezeigt. Der Nutzer sollte auf einen Blick sehen können, wie viel für dieses Event noch zu erledigen ist. Für weitere Informationen zu einem Event wird dem Nutzer die Funktion zum Öffnen des Events angeboten.

Wie in Abbildung 13 dargestellt, gibt es zusätzliche Informationen bezüglich eines Filter. Wird die Eventliste über ein Team geöffnet, so werden alle Events gefiltert und nur die angezeigt, welche zu diesem Team gehören. Die Filterung nach einem Team wird dem Nutzer dann in dieser Strukturkarte angezeigt. Ebenso wie in der Teamliste gibt es auch hier eine Funktion, um neue Events zu erstellen.

Wie in Abbildung 13 dargestellt, ist diese Funktion analog zur Teamliste in der erweiterten Fußzeile platziert. Die Strukturkarte für den Filter ist vor der Eventliste angeordnet. Der Nutzer sollte direkt als erstes darüber informiert werden, wenn ein Team-Filter aktiv ist und dies nicht erst nach langem Scrollen bei einer vollen Eventliste durch Zufall mitgeteilt bekommen, wenn er am Ende der Liste angekommen ist. Diese Strukturkarte wird ausgeblendet, wenn es keinen aktiven Filter gibt, um Platz zu sparen.

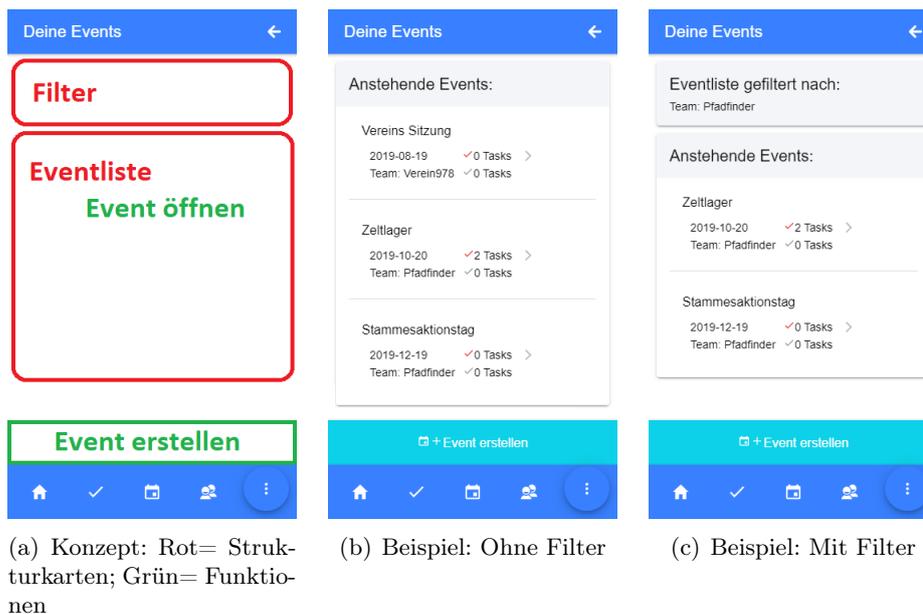
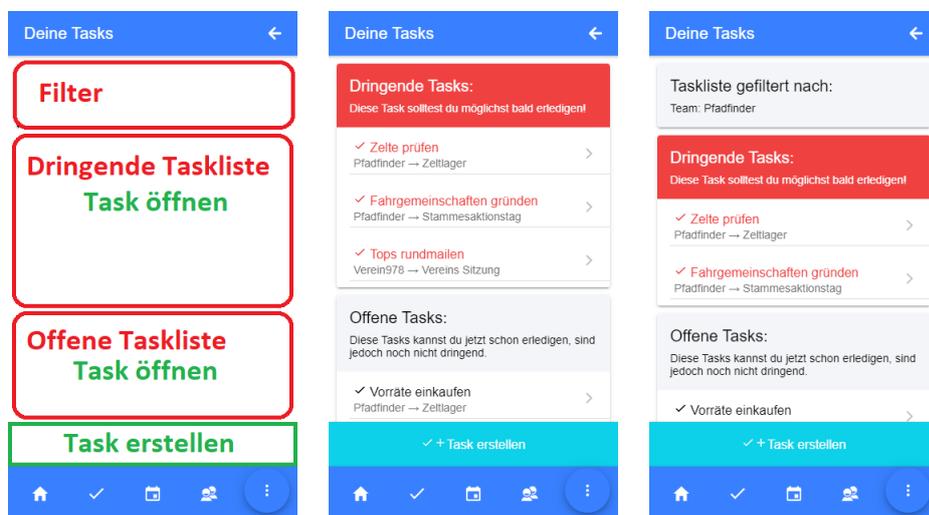


Abbildung 13: Konzept und zwei Beispiele der Page „Eventliste“

Taskliste

Wie in Abbildung 14 dargestellt, gibt es drei Strukturkarten auf dieser Page, wodurch sie sich leicht von der Eventliste und der Teamliste unterscheidet. Während bei den anderen beiden eine vollständige Liste angeboten wird und die Eventliste lediglich optional nach Teams gefiltert werden kann, ist die Taskliste von vornherein vorgefiltert. Dort werden nur alle vom Nutzer unerledigten Task angezeigt. Unter diesen werden sie von ScOuT nochmal zusätzlich sortiert nach dringenden Tasks und offenen Tasks. Dringende Tasks sind die unerledigten Tasks, bei welchen der Dringlichkeits-Start (siehe Abschnitt 4.3 und 4.4) bereits aktiv ist. Offene Tasks dagegen sind unerledigt und haben noch keinen aktiven Dringlichkeits-Start. Durch diese Sortierung soll die Darstellung der Liste der unerledigten Tasks für den Nutzer bereits vorverarbeitet werden und ihm die Entscheidung, um welchen Task er sich als nächstes kümmern möchte, deutlich erleichtern. Dabei werden die dringenden Tasks immer vor den offenen Tasks angezeigt, wie in Abbildung 14

dargestellt. Zu den Informationen, welche dem Nutzer zusätzlich zum Namen des Task angezeigt werden, gehören hier der Teamname und der Eventname. Dies erfolgt auf derselben Grundlage wie bei der Eventliste, dass es gleichnamige Task für verschiedene Events und Teams geben kann und der Nutzer durch diese Information die Zuordnung direkt erkennt. Wie auch die Eventliste, kann die Taskliste nach Team und Event gefiltert werden. Dabei bleiben weiterhin nur die unerledigten Tasks in der Anzeige. Um zu den anderen Pages mit Listen konsistent zu bleiben, kann auch in dieser Liste die Taskansicht geöffnet und Tasks erstellt werden. Letztere Funktion befindet sich erneut in der erweiterten Fußzeile.



(a) Konzept: Rot= Strukturkarten; Grün= Funktionen

(b) Beispiel: Ohne Filter

(c) Beispiel: Mit Filter

Abbildung 14: Konzept und zwei Beispiele der Page „Taskliste“

Teamansicht

Während die Listen eine Übersicht über alle Objekte einer Objektstruktur geben, stellt eine konkrete Ansicht eines Objekts alle Informationen zu diesem bereit. Im Fall der Teamseite sind diese Informationen die Events und die Mitglieder. Über die Events werden dem Nutzer zudem Informationen darüber gegeben, wie viele dringende und offene Tasks aktuell innerhalb dieses Teams sind. Diese Basisinformationen sind alle mit Funktionen zum Öffnen der gefilterten und ungefilterten Listen verknüpft. Die zweite Strukturkarte spaltet die Informationen weiter auf, um den Nutzer mit genaueren Angaben zu unterstützen. Deswegen wird dort eine Liste aller Events des Teams mit Angabe der dringenden Tasks angezeigt. Auch die Informationen bezüglich der Teammitglieder erhielten in der ursprünglichen Planung eine

Strukturkarte. Doch die Liste der Mitglieder, die Liste der eingeladenen Mitglieder und die Funktion zum Einladen neuer Mitglieder überfüllte die Seite für die Nutzung am Smartphone, weshalb sie eine eigene Page erhalten.

Wie in Abbildung 15 gezeigt, erfolgt die Anordnung der beiden Strukturkarten auf der Ansichts-Page nach dem Prinzip, dass zuerst die allgemeinen, zusammenfassenden Informationen bereitgestellt werden, welche anschließend in genauere Informationen aufgeteilt werden.

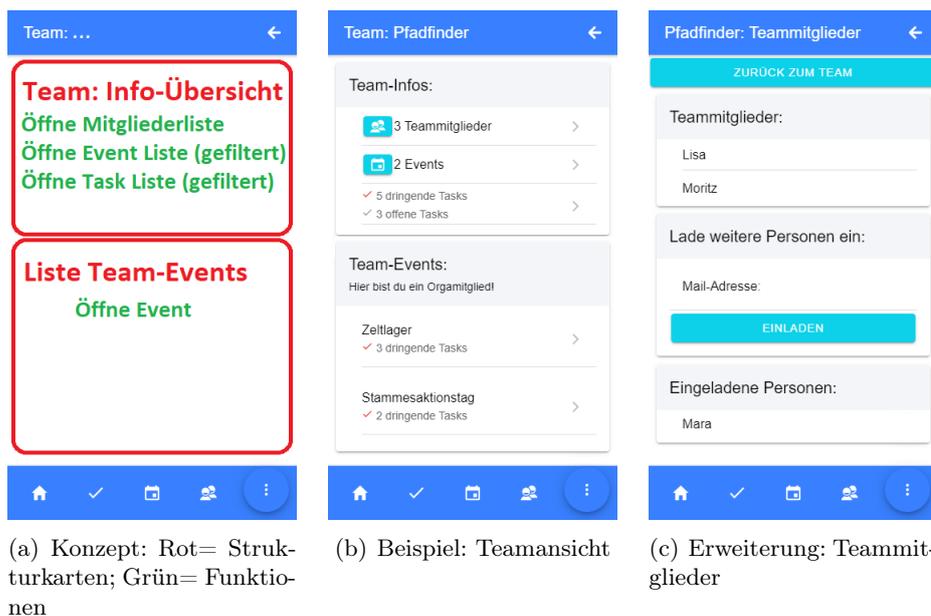


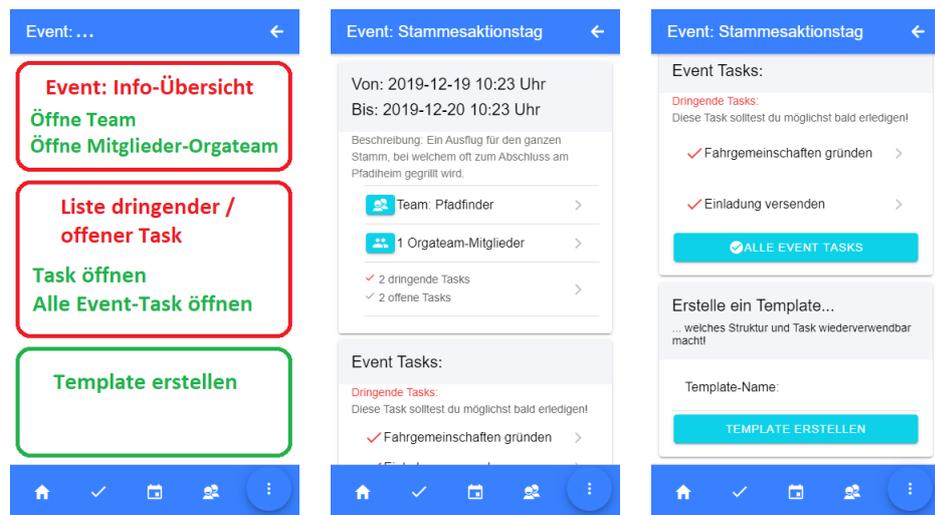
Abbildung 15: Konzept und ein Beispiel der Page „Teamansicht“ mit einem Beispiel der Page „Teammitglieder“

Eventansicht

Ähnlich wie zuvor bei der Teamansicht beschrieben, zeigt auch diese Page zunächst eine Zusammenfassung über die Informationen zu dem Event an und anschließend eine genauere Aufschlüsselung weiterer Informationen. Zu den Informationen in der allgemeinen Übersicht zählen der Event-Zeitraum, die Beschreibung, das zugehörige Team, die Orgateam-Mitgliederliste und die Anzahl der dringenden und offenen Tasks. Bis auf den Zeitraum und die Beschreibung können über die Informationen wieder andere Pages geöffnet werden. Im Falle der Taskliste wird hier die gefilterte Variante geöffnet. In der zweiten Strukturkarte werden alle dringenden Tasks des Events aufgelistet, sodass diese direkt von dort geöffnet werden können, um sich als Nutzer um die Erledigung dieser zu kümmern. Hat dieses Event keine dringenden Tasks, dann werden die offenen Tasks angezeigt. Die Informationen bezüglich der Tasks zu einem Event werden für den Nutzer vorgefiltert, sodass

sie innerhalb der Eventansicht die Page nicht überfüllen. Des Weiteren gibt es in der zweiten Karte die Funktion eine Taskliste zu öffnen, welche alle Tasks des Events beinhaltet, sogar die bereits erledigten. Die Einsicht der erledigten Tasks hat relativ geringe Priorität und wird lediglich benötigt, um gegebenenfalls ein Event, welches als Template dienen soll, anzupassen. Deshalb wurden in allen anderen Pages immer nur die dringenden und offenen Tasks priorisiert. Damit der Nutzer die Übersicht über die erledigten Tasks jedoch nicht verliert, hat er die Möglichkeit sie über das Event einzusehen.

Eine Erweiterung der Eventansicht im Vergleich zu der Teamansicht ist die dritte funktionelle Karte, mit welcher der Nutzer ein Template des Events anlegen kann. Aktuell ist das Anlegen eines Templates sehr simpel gehalten, doch die Überlegung besteht auch diese Karte auf eine eigene Page auszulagern, um weitere Informationen und Funktionen anbieten zu können. Ein weiterer Grund diese Funktion auszulagern ist der, wie in Abbildung 16 zu sehen ist, dass die Page der Eventansicht bereits so voll ist, dass die Karte zum Anlegen eines Templates im Smartphone nicht mehr direkt zu sehen ist und der Nutzer diese Funktion gegebenenfalls suchen muss, wenn er sich noch nicht gut mit dem Aufbau von ScOuT auskennt.



(a) Konzept: Rot= Strukturkarten; Grün= Funktionen

(b) Beispiel: Eventansicht

(c) Beispiel: Eventansicht Fortsetzung

Abbildung 16: Konzept und ein zweigeteiltes Beispiel der Page „Eventansicht“

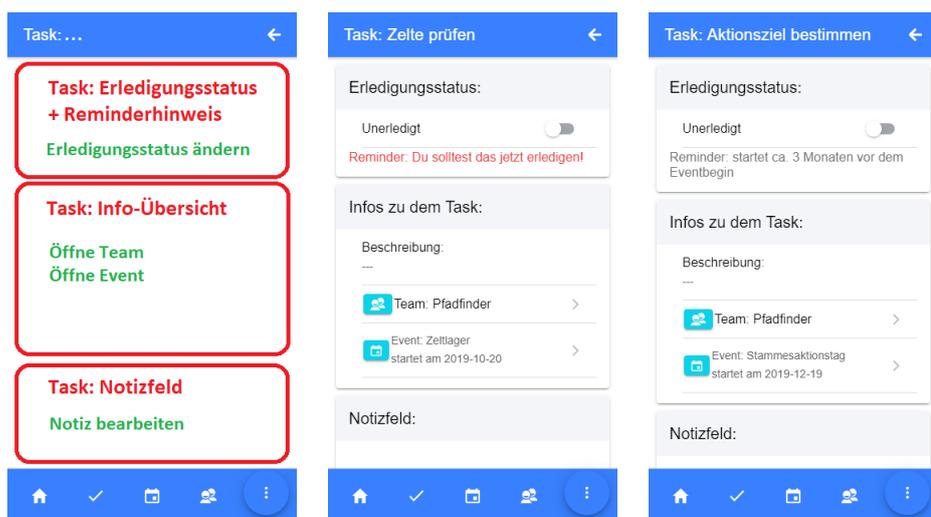
Taskansicht

Bei der Darstellung der Informationen eines Tasks gibt es keine Untergliederung in allgemein und spezifisch. Stattdessen werden die allgemeinen Informationen in ihrer Wichtigkeit unterteilt, weshalb es in Abbildung 17

zwei Strukturkarten für diese gibt. Die erste beinhaltet den Erledigungs-Status des Task und die Funktion diesen zu ändern, sowie einen Reminder, welcher den Dringlichkeitstatus interpretiert und dem Nutzer Informationen dazu gibt. Diese beiden Inhalte sind für den Nutzer wichtiger als andere Informationen über den Task, da sie zum erfolgreichen Erledigen der Aufgabe und somit dem Planen der Veranstaltung dienen. Zusätzlich trägt die Tatsache, dass das System hier eine Interaktion des Nutzers, ähnlich wie bei der Team-Einladung, erwartet, dazu bei, diese Strukturkarte als erstes auf der Page anzuordnen.

Die zweite Strukturkarte beinhaltet, entsprechend der anderen Pages zur Ansicht konkreter Objekte, einen Überblick über die allgemeinen Informationen eines Tasks. Darunter fallen die Beschreibung, das Team und das Event mit seinem Eventbeginn. Wie in den anderen Ansichten auch, können hier das Team und das Event direkt geöffnet werden.

Zuletzt gibt es noch eine funktionelle Karte für die Notiz. Die Notiz enthält nur Informationen, welche der Nutzer eingegeben hat und kann direkt dort auch vom Nutzer angepasst werden.



(a) Konzept: Rot= Strukturkarten; Grün= Funktionen

(b) Beispiel: Taskansicht eines unerledigten und dringenden Task

(c) Beispiel: Taskansicht eines nicht dringenden Task

Abbildung 17: Konzept und zwei Beispiele der Page „Taskansicht“

6 Evaluation

Nachdem die Grundlagen von ScOuT sowohl für die Funktionen und Strukturen, als auch für die Oberfläche implementiert wurden, steht aus, diese auf ihre Gebrauchstauglichkeit zu prüfen. Hierfür wird zunächst abgegrenzt, welche Bereiche wie weit in der Entwicklung vorangeschritten sind, um nur jene zu evaluieren, welche dafür auch bereit sind. Unfertige Bereiche werden zu einem späteren Zeitpunkt geprüft.

Mit Hilfe der Einschränkung wird anschließend das genaue Ziel der Evaluation festgelegt und die Prüfung in Form einer Benutzerumfrage erstellt. Da das Ziel der Masterarbeit sowohl die Konzeption der Anwendung, als auch der Oberfläche umfasst, werden beide Teile in der Umfrage ermittelt und anschließend in der Auswertung betrachtet.

6.1 Evaluationsziele

Bevor jedoch die Umfrage erstellt werden kann, muss zunächst genau definiert werden, welches Ziel mit ihr verfolgt wird. Die Umfrage soll sich nur mit den Fragen beschäftigen, welche zu diesem Zeitpunkt der Entwicklung geeignet sind. Ziele bezüglich bisher nicht existierenden Funktionen oder noch nicht umgesetzten Usability-Richtlinien sollten erst evaluiert werden, wenn diese implementiert und beachtet wurden. Deshalb wird nachfolgend der aktuelle Stand von ScOuT kritisch untersucht.

Die in Abschnitt 3.1 genannten Eigenschaften der Selbstbeschreibungsfähigkeit und Fehlertoleranz fordern, dass das System die Nutzer zu jeder Zeit vollständig über das System aufklären sollte, um Fehler zu vermeiden. Dies ist überwiegend in der Konzeption der Benutzeroberfläche einbezogen worden. Jedoch gibt es noch Verbesserungsmöglichkeiten im Bereich der Eingabeelemente. Diese sind zwar beschriftet und an kritischen Stellen, wie der Registrierung, wo Nutzernamen und Anzeigenamen erklärt werden, durch zusätzliche Informationen ausgestattet, jedoch reicht dies nicht für jeden Nutzer an Hilfestellungen. Nützliche Informationen zum Verständnis der Eingabeformulare in der Hilfe anzusiedeln, würde dem Nutzer jedoch nur zusätzlichen Aufwand bringen. Des Weiteren wäre eine sinnvolle Strukturierung der Informationen über alle Eingabefelder in ScOuT in der Hilfe schwierig. Diese Informationen sollten folglich nicht weit entfernt zum Eingabefeld selbst sein, sondern möglichst nah. Um dies umsetzen zu können, wäre es möglich an jedes Eingabefeld das konventionelle kleine Icon von einem *i* im Kreis anzubringen, welches beim Anklicken Informationen über das Eingabefeld angibt. Diese Informationen sollten enthalten, welche Eingabe in dem Feld erwartet wird, wofür sie vom System benötigt wird sowie die Angabe eines minimalistischen Beispiels.

Weitere bisher nicht vollständig implementierte Funktionen betreffen die Fehlertoleranz und die Steuerbarkeit der Dialoggestaltung aus Abschnitt 3.1

und die Nutzerkontrolle und Freiheit der Nielsen Heuristik in Abschnitt 3.3, da es im aktuellen Zustand von ScOuT keine Möglichkeiten gibt erstellte Teams, Events und Tasks zu bearbeiten oder zu löschen, sowie aus Teams auszutreten oder Teameinladungen abzulehnen. Diese Funktionen müssen in der weiteren Entwicklung noch implementiert und ihre Darstellung in der GUI integriert werden, damit die Fehlerkorrektur vereinfacht werden kann und die Steuerbarkeit dem Nutzer mehr entgegen kommt.

Ebenso wurden noch nicht alle Ansprüche an die Lernförderlichkeit der Dialoggestaltung und die Hilfe und Dokumentation von Nielsen aus den Abschnitten 3.1 und 3.3 umgesetzt. Es gibt zwar bereits ein Tutorial und die Hilfe, doch damit sind noch nicht alle Standards erfüllt, welche es benötigt um eine vollständige Umsetzung der Usability-Eigenschaften zu erreichen. Zum einen fehlt dafür neben der Hilfe und dem Tutorial noch ein Bereich für häufig gestellte Fragen (eng.: Frequently Asked Questions; kurz: FAQ). Zum anderen ist das Tutorial noch nicht optimiert. Der Nutzer erhält zwar beim ersten Gebrauch ein kurze, einführende Anleitung zu der Anwendung, welche auch jederzeit wieder eingesehen werden kann, jedoch sind die Hilfestellungen nicht nah genug an den dazugehörenden Funktionen angesiedelt. Es benötigt etwas Zeit die schriftlichen Anleitung zu lesen und der Nutzer könnte zumEnd ehin die ersten Informationen über das Team bereits wieder vergessen haben und das obwohl dies die ersten Schritte sind, welche er in ScOuT gehen sollte. Es würde sich daher empfehlen eine geführte Tour während der Benutzung zu machen. Die helfenden Informationen würden spezifisch dann auftauchen, wenn der Nutzer sie zum ersten Mal benötigt und nicht alle zusammen auf einmal nach der Registrierung.

Gar nicht einbezogen wurden bisher die Individualisierbarkeit der Dialoggestaltung und die dazu ähnliche Flexibilität und Effizienz von Nielsen aus den Abschnitten 3.1 und 3.3, die Attraktivität der Qualitätskriterien in Abschnitt 3.2, sowie die Installation, die Sicherheit und einige Punkte des Telefon-Kontexts von Vuolle aus Abschnitt 3.4. Für die persönliche Gestaltung könnten in Zukunft verschiedene Farbschemata angeboten werden, zwischen welchen der Nutzer wählen kann. Zudem würden personalisierte Dialogkontrollen unterschiedlichen Benutzertypen helfen die Steuerung anzupassen. Im Falle der von Vuolle genannten Eigenschaften wurden zum Beispiel die Installation, die Sicherheit und der Akkuverbrauch noch nicht untersucht, da ScOuT im aktuellen Entwicklungsprozess ausschließlich im Browser getestet wurde und nicht als hybride App am Mobiltelefon selbst. Andersherum wurde ScOuT noch nicht vollständig in Bezug auf die Eigenschaften der Bildschirmgrößen konzipiert, da die Größe trotz der Tests im Browser immer auf die Größe eines Smartphones reduziert wurde und nie die volle Größe eines Monitors hatte. Was bereits im Browser getestet wurde ist die Informationseingabe. Hier wurde festgestellt, dass der von Ionic vorgegebene `IonTimeDatePicker` möglicherweise am Handy ein sinnvolles Eingabeelement darstellen mag, jedoch für die Bedienung mit Maus und Tastatur

ungeeignet ist. Da Ionic trotzdem eine keine alternative Komponente bereitstellt, muss speziell für diesen Gebrauch ein eigenes Element implementiert werden.

Es wurden, wie in Abschnitt 4.4 bereits verkündet einige Funktionalitäten von ScOuT im Rahmen der Masterarbeit weggelassen, obwohl diese zum Teil explizit in dem Interview gewünscht waren. Diese Anforderungen sollten ebenfalls noch implementiert und in der GUI gebrauchstauglich angeboten werden. Darunter fällt zum Beispiel das komplexe Zuordnungssystem von Orgateams, Suborgateams, Helfern und Einzelpersonen für einen einzelnen Task. Ebenso wurden die Möglichkeiten des Dringlichkeits-Systems noch nicht vollständig ausgeschöpft.

Die genannten Funktionen und Konzepte der Oberflächengestaltung werden erst in Zukunft bearbeitet und können deshalb zum aktuellen Zeitpunkt noch nicht bewertet werden. Die nachfolgende Evaluation bezieht sich folglich auf die bereits bestehenden Funktionen und Usability-Richtlinien.

6.2 Testverfahren

ScOuT unterscheidet sich in einigen Funktionen wie dem neuen Dringlichkeits-Status von bereits existierenden Tools. Bei der Umsetzung dieser Neuerungen wurden verschiedene Methoden zur Gestaltung der Oberfläche angewandt (siehe Kapitel 3), welche als Ziel verfolgen die Richtlinien von Nielsen, Vuolle und ISO zu erreichen (siehe Abschnitt 5.2). Dabei waren sowohl die Richtlinien teils heuristisch ermittelt, als auch manche der Methoden heuristisch umgesetzt. Nun wird ScOuT dahingehend geprüft, ob sie dieses Ziel auf heuristischem Weg erreicht hat. Hierfür wird ein Nutzertest in Form eines Fragebogens erstellt. Dieses Testverfahren gehört zu den empirischen Bewertungsmethoden. Durch die Teilnahme realer Probanden, anstelle von virtuellen, ist der Bezug zum Benutzer stärker, als bei formalen oder kriterienorientierten Bewertungen. Der Bezug zum System und damit die Validität kann bei Fragebögen jedoch unterschiedlich ausfallen [Ham06]. Ebenso kann der Fragebogen in die subjektiven Methoden eingeordnet werden, welche durch die direkte Beurteilung durch Endbenutzer weiche Daten gewinnt [BN13]. Um den Fragebogen zu delokalisieren wird zudem nach dem Remote-Usability-Test die Umfrage online durchgeführt. Dies ermöglicht ortsunabhängige Tests und die Teilnahme der Pfadfinder aus den Interviews [BN13]. Für diesen Zweck wurden das Frontend und das Backend von ScOuT jeweils bei dem Online-Hosting Anbieter heroku²⁴ und die Datenbank beim MongoDB Cloud-Service²⁵ gehostet. ScOuT wurde nun als Website via Inlineframe (kurz: iFrame) in der Umfrage eingebunden. Des Weiteren wurde zur speziellen Prüfung der Gebrauchstauglichkeit nach dem Mobile-Usability-Test

²⁴<https://heroku.com>

²⁵<https://cloud.mongodb.com>

die zu bewertende Anzeigefläche des iFrame auf die Größe eines Smartphones reduziert. So kann direkt geprüft werden, ob die Gebrauchstauglichkeit, trotz dieser Einschränkung, dennoch gegeben ist [BN13]. Zusätzlich wurde der Fragebogen nach dem szenariobasierten Usability-Test aufgebaut, bei welchem der Nutzer verschiedene Aufgaben erhält, welche in ScOuT erledigt werden sollen [BN13].

Der Aufbau der Umfrage orientiert sich nach dem Trichterprinzip [Red]. Demnach werden die Fragen so gegliedert, dass allgemeinere zu Beginn der Umfrage gestellt und die Fragen zum Ende hin immer konkreter werden. Dadurch ergibt sich der folgende Aufbau:

- a) Einleitungstext + Datenschutzabfrage
- b) Abfrage demographischer Daten
- c) Abfrage von Vorkenntnissen und Zielen
- d) Aufgabe
- e) Konkrete Fragen zum Produkt und der Gebrauchstauglichkeit
- f) Reflexionsmöglichkeit

Unabhängig vom Inhalt der Fragen wurde beim Aufbau jeder einzelnen darauf geachtet, dass dieser Regeln folgt. Darunter die Regeln, dass Fragestellungen ungestützt und offen, einfach und nicht suggestiv sind [Sch19].

Bei der Erstellung der Inhalte der konkreten Produktfragen wurden bereits etablierte Listen an Fragen zum Thema Gebrauchstauglichkeit betrachtet. Darunter Questionnaire for User Interface Satisfaction (kurz: QUIS), System Usability Scale (kurz: SUS) und Computer System Usability Questionnaire (kurz: CUSO) [BN13]. Obwohl diese Fragebögen zum Standard zu gehören scheinen, sind die Fragen doch eher unzeitgemäß aufgebaut und entsprechen nicht den oben genannten Regeln für Fragestellungen. Zudem decken die Listen an Fragen nicht alles ab, was spezifisch für den aktuellen Stand von ScOuT relevant ist. Aus diesen Gründen wurde sich lediglich an den Frageinhalten und Themen orientiert um einen eigenen spezifischen Fragebogen zu erstellen. Der gesamte Fragebogen befindet sich im Anhang C und wird im folgenden Absatz beschrieben.

Inhalt des Fragebogens

Die Abfrage der demographischen Daten (siehe Fragebogenaufbau b) im vorherigen Abschnitt) umfasst das Alter und das Geschlecht (siehe Anhang C Fragebogenseite 2). Dies dient lediglich der groben Einschätzung der Probandengruppe. Weitere demographische Daten hätten für diese erste Umfrage zur Gebrauchstauglichkeit noch keine große Relevanz, weshalb sie nicht

abgefragt werden. In zukünftigen Fragebögen können zur Untersuchung der Personas jedoch zusätzliche demographische Daten erhoben werden.

Anschließend wurden die Vorkenntnisse und Ziele der Probanden ermittelt, welches ebenfalls dem Zweck dient die Teilnehmer besser einordnen zu können (siehe Anhang C Fragebogenseite 2). Für die Vorkenntnisse wurde eine Liste von Tools vorgeschlagen und zwischen den Erfahrungen mit dem Computer oder einem Smartphone unterschieden. ScOuT selbst steht in dieser Liste nicht zur Auswahl, da das Tool bisher nicht öffentlich verfügbar ist und einzig die Teilnehmer der Methode Lautes Denken aus Abschnitt 5.2 Erfahrung mit ScOuT sammeln konnten.

Die Aufgabe im Fragebogaufbau d) (siehe Anhang C Fragebogenseite 3) befasst sich mit den Kernfunktionen der Anwendung wie dem Erstellen eines Teams, einer Veranstaltung und mehrerer Tasks. Zudem sollen die Probanden mit dem Aufgabenschritt 5 Erfahrung mit dem Template machen, welches den maßgeblichen Unterschied zu weiterführenden Arbeiten, wie in Abschnitt 2.3 genannten Tools, macht.

Anschließend folgen die konkreten Fragen zum Produkt und zur Gebrauchstauglichkeit. Beginnend auf den Fragebogenseiten 4 und 5 (siehe Anhang C) wird zunächst der Erreichungsgrad der Aufgabe abgefragt. Dies verschafft einen allgemeinen Überblick über Probleme und Ursachen bei der Bedienung der Anwendung. Danach folgen konkretere Fragen, welche sich stärker an den angewandten Usability-Methoden aus Abschnitt 5.2 orientieren und prüfen sollen, ob die Anwendung der Methoden ihren Zweck erfüllt haben. Fragebogenseite 6 befasst sich deshalb mit der Terminologie und mit der Anordnung von wichtigen Informationen. Um den Zeitaufwand für das Ausfüllen des Fragebogens zu reduzieren, wurde als wichtigste Information der neu entwickelte Dringlichkeitsstatus im Zusammenhang mit dem Reminder und dem Erledigungs-Status des Task abgefragt. Weitere konkrete Fragen befassten sich mit der Nutzung der Hilfe-Page (im Fragebogen: Hilfe-Funktion) und Fehlermeldungen.

Zum Abschluss werden auf der Fragebogenseite 10 der Fragenbogaufbau mit f) Reflexionsmöglichkeiten beendet. Dort ist eine letzte konkrete Frage zum Gebrauch von Templates.

6.3 Auswertung

Die Umfrage wurde von 193 Personen aufgerufen und gestartet. 64 davon füllten sie bis zur letzten Seite aus. Die meisten Abbrüche gab es auf Seite 2, welche mit den demographischen Daten, Vorkenntnissen und potentiellen Nutzungszielen den Fragebogen einleitet. Die Umfrage wurde an die Stämme der DPSG aus Limburg versendet, die am Interview teilgenommen haben. Des Weiteren wurde sie über den Uni-Newsletter, sowie private Kontakte gestreut. Die Probanden beschränken sich somit nicht mehr ausschließlich auf die Pfadfinder und Vereine.

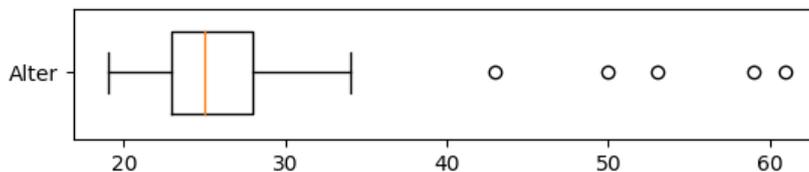


Abbildung 18: Umfrageergebnisse: Altersangaben

Von den 64 Personen waren 31 weibliche Probanden, 32 männliche und eine Person gab beim Geschlecht divers an. Das durchschnittliche Alter lag bei 27 und es gab eine Spannweite von 19 bis 61. Wie in Abbildung 18 zu sehen ist, liegt der Schwerpunkt jedoch zwischen 20 und 30 und es gibt lediglich wenige Ausreißer über dem Alter von 40 Jahren.

Bei der Abfrage der Vorkenntnisse bezüglich bereits existierender Tools, muss bedacht werden, dass der Begriff Event zu diesem Zeitpunkt der Umfrage nicht definiert ist. Deshalb muss davon ausgegangen werden, dass auch ein Treffen unter Freunden, bei welchem nur der Zeitpunkt und der Ort geplant werden müssen, für die Probanden bereits als Event gilt. Die Vorkenntnisse der Probanden von bereits existierenden Tools zum Planen von Veranstaltungen beschränken sich bei den reinen Smartphone-Vorkenntnissen auf sechs und bei denen am Computer auf vier Personen. Keinerlei Vorkenntnisse gaben 30 Probanden an und Vorkenntnisse auf beiden Plattformen 24. Obwohl die Hälfte der Probanden angab, dass sie bereits Erfahrungen mit solchen Tools gesammelt haben, haben nur drei Personen angegeben, dass sie mit mehr als drei der gefragten Tools gearbeitet haben. Somit haben die meisten Probanden im Durchschnitt erst Erfahrung mit einem Tool gehabt. Neben den vorgeschlagenen Planungstools, welche aufgrund ihrer umfangreichen Funktionen gewählt wurden, nannten die Probanden individuell noch weitere. Hierunter fielen Tools mit konkreten Anwendungszielen, welche bereits auch in Abschnitt 2.3 genannt wurden, wie Doodle, Google Drive und Slack. Des Weiteren wurde ein Tool namens Terminplaner 4.0²⁶ genannt, welches den Funktionen von Doodle entspricht, sowie bringabottle und Kadermanager. Bringabottle²⁷ ist ein Tool zum expliziten Planen von Veranstaltungen. Wie der Name es vermuten lässt, geht es darum zu managen, wer was zu einer Party mit bringt. Der Einladende kann eine Liste erstellen und die eingeladenen Personen können sich in der Liste eintragen. So wird sicher gestellt, dass nichts doppelt mitgebracht wird und der Einladende einen Überblick darüber behält, was noch fehlt. Kadermanager²⁸ ist speziell für Sportvereine mit Mannschaftssport entwickelt. Im Mannschaftssport bezeichnet ein Kader die Gruppe an möglichen Mitspielern bei einem Spiel oder Training. Dies

²⁶<https://terminplaner4.dfn.de/>

²⁷<http://www.bringabottle.de>

²⁸<https://kadermanager.de/>

entspricht einer Teilnehmerliste. Durch einfaches Anklicken der Buttons „In“ und „Out“ können die Vereinsmitglieder angeben, wann sie teilnehmen. Der Trainer kann so den Kader mit einem besseren Überblick im Voraus planen. Abbildung 19 a) gibt einen Überblick über die bevorzugten Tools. Trello, Doodle und ToDoIst sind dabei die meist genutzten Favoriten, wohingegen ClickUp von keinem Probanden verwendet wurde und Inloox, Terminplaner 4.0, Slack, Kadermanager und Bringabottle zu Sonstiges zusammengefasst wurden, da sie nur von Einzelpersonen als Vorkenntnisse genannt wurden. Auf Grundlage dieser Ergebnisse, sollten in zukünftigen Arbeiten an ScOuT auch die hier neu vorgestellten Tools einbezogen werden, da diese oder ähnliche Anwendungszwecke ebenfalls zu den Anforderungswünschen der Nutzer gehören könnten. Gerade das Abfragen der Teilnehmer und die Zuordnung von Materiallisten würde seinen Verwendungszweck beim Planen von Veranstaltungen finden.

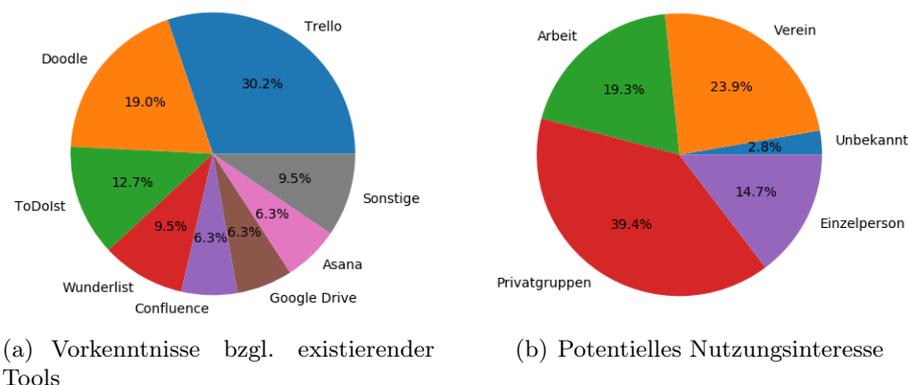


Abbildung 19: Umfrageergebnisse: Nutzungsziel und Vorkenntnisse

Wie bereits zu Beginn dieses Abschnittes erwähnt, sind die Probanden nicht nur Vereinsmitglieder der Pfadfinder. Dies spiegelt sich auch in den potentiellen Nutzungszielen aus Frage 3 der Umfrage wider. Wie in Abbildung 19 b) zu sehen ist, besteht ein höheres Interesse daran ein Planungstool eher für private Gruppen zu nutzen. Vier Probanden gaben an, dass sie zum aktuellen Zeitpunkt noch nicht wissen, wofür sie ein solches Tool nutzen sollten. Das Interesse der Nutzung für die Arbeit, Vereine oder für sich als Einzelperson hält sich die Waage zwischen 16 und 26 Interessierten. Da das geplante Produkt noch nicht erhältlich ist, sollte noch erwähnt werden, dass alle Probanden potentielle zukünftige Nutzer sind.

In zukünftigen Umfragen sollten die Probandengruppen weiter spezifiziert werden. Sobald ScOuT verfügbar gemacht wird, sollte die Erfahrungsdauer mit dem Tool stärker einbezogen werden. So kann zwischen potentiellen zukünftigen Nutzern, vollkommen unerfahren, relativ neuen Nutzern und erfahrenen Nutzern unterschieden werden um diese Fähigkeitenlevel in

ihrem Stadium besser zu unterstützen und sich tiefergehend mit der Eigenschaft Erwartungskonformität der Dialoggestaltung und Flexibilität der Nielsen Heuristik in Kapitel 3 zu befassen.

Nachdem nun ausführlich die Probanden beschrieben wurden, folgt die Aufgabe zur Nutzung von ScOuT in der Umfrage und anschließend die einzelnen Fragen, welche sich sowohl mit der Gebrauchstauglichkeit befassen, als auch mit den Unterschieden zu bereits existierenden Planungstools.

Erledigungsgrad der Aufgabe

Beginnend zur Auswertung der Umfrage wird geprüft, ob bei der Benutzung von ScOuT Probleme auftraten. 52 Probanden gaben an, die Aufgabe vollständig erledigt zu haben, was daraufhin deutet, dass sie keine oder nur minimale Schwierigkeiten hatten. Dagegen gaben 11 Probanden an, die Aufgabe nur teilweise erledigt zu haben. Eine einzelne Person, hat die Aufgabe gar nicht erledigt.

Die Einzelperson gab bei Frage 9 als Begründung „Sonstiges“ als Problem an. Durch das Hosting von Frontend, Backend und Datenbank bei den kostenlosen Versionen von heroku und MongoDB traten teilweise erhöhte Wartezeiten auf, bis ScOuT im iFrame angezeigt wurde. Die Wartezeit könnte gegebenenfalls dem Nutzer zu lange gedauert haben, weshalb keine Applikation angezeigt wurde und er deshalb die Aufgaben nicht erledigt hat. Ebenso kann ein Grund für das nicht Anzeigen der Applikation sein, dass beim Laden ein Fehler aufgetreten ist. Auch berichteten einige der Probanden, dass die Applikation nicht in allen Browsern funktioniert. Leider wurde bei der Erstellung der Umfrage versäumt ein Textfeld anzulegen, welches dem Probanden ermöglicht Feedback zu geben, was der genaue Grund ist. Zudem sollte in zukünftigen Umfragen nach der Angabe, dass der Proband keine einzige Aufgabe erledigt hat, gegebenenfalls eine andere Umfrage folgen. Denn, wenn die Anwendung, wie in diesem Fall vermutet, nicht angezeigt wurde, kann der Proband nichts über die Oberfläche und die Funktionen aussagen. Da der Proband zudem angab, dass er, obwohl keine Aufgabe erledigt wurde, die Hilfe-Funktion nicht brauchte, wird dies als Bestätigung angesehen, dass ScOuT gar nicht erst angezeigt und somit nicht bedient wurde. Aus diesem Grund wird der Datensatz des Probanden für den Rest der Auswertung ausgeschlossen.

Neben dem einen Probanden, welcher trotz nicht Erledigung der Aufgaben den Fragebogen ausfüllte, gab es von den 193 Personen, die den Fragebogen starteten genau acht, die nicht weiter als Seite 3, also der Aufgabe, kamen. Diese könnten den Fragebogen ebenfalls abgebrochen haben, weil ScOuT gar nicht oder zu langsam angezeigt wurde. Allerdings könnte die Ursache ebenso sein, dass sie zu große Probleme bei der Bedienung der Applikation hatten, weshalb sie hier erwähnt werden sollten.

Das Ergebnis von 52 vollständig erledigten Aufgaben und 11 teilwei-

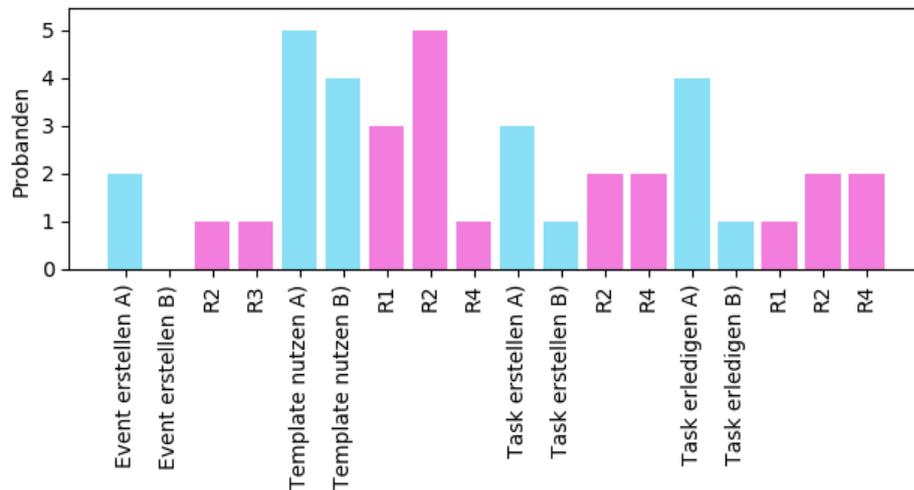


Abbildung 20: Umfrageergebnisse: Erledigungsgrad der Aufgabe: A) teilweise erledigt B) komplett unerledigt; Ursachen: R1 = Funktion nicht gefunden, R2 = Funktion nicht verstanden, R3 = Funktion funktionierte nicht, R4 = Sonstige Ursache

se erledigten Aufgaben, ist für den frühen Entwicklungsstand von ScOuT ein positives Ergebnis. Die Schwierigkeiten und Probleme der 11 Probanden verteilten sich dabei auf unterschiedliche Schritte und Ursachen. Probleme aufgrund des Nicht-Verstehens der Aufgabe wurden aus der Auswertung der Umfrage ausgeschlossen, da diese keine Hinweise auf die Gebrauchstauglichkeit in ScOuT geben. Abbildung 20 verschafft einen Überblick darüber, bei welchen Aufgaben es Probleme gab und was dafür von den Probanden als Ursache angegeben wurde. Die zwei blauen Balken geben jeweils an, ob A) dieser Aufgabenteil vielleicht oder teilweise erledigt wurde und B) ob er gar nicht erledigt wurde. Die magentafarbenen Balken benennen jeweils für den davor genannten Aufgabenschritt die Menge der Fehlerursachen R1 = Funktion nicht gefunden, R2 = Funktion nicht verstanden, R3 = Funktion hat nicht funktioniert und R4 = Sonstiges. Sind Probleme aufgetreten, weil die Funktion nicht gefunden wurde, so sollte der Aufbau der Informationen und Funktionen auf den einzelnen Seiten überdacht werden. Die Problemquelle des Nicht-Verstehens der Funktion sollte durch das im Ausblick (siehe Abschnitt 6.1) vorgeschlagene, geführte Tutorial behoben oder zumindest verringert werden können. Im Falle, dass eine Funktion nicht funktionierte, sollten zu dieser Funktion umfassende Testfälle implementiert werden um Fehler gegebenenfalls aufzuspüren und zu beheben.

Insgesamt hatte von den elf Probanden niemand Schwierigkeiten mit der Erstellung oder dem Beitritt in ein Team. Zudem gab es zwei Probanden mit Problemen beim Erstellen eines Events und neun beim Erstellen eines Events mit Template. Probleme beim Erstellen und Erledigen eines Task betraf vier

und sechs Probanden. Auffällig ist an dieser Stelle, dass das neu entwickelte Template nicht hinreichend in ScOuT integriert wurde. Hier fanden drei Personen die Funktion zum Erstellen des Templates oder die Funktion der Erstellung eines Events aus dem Template nicht. Fünf weitere gaben an, dass sie die Funktion gar nicht verstanden. In Anbetracht der 52 erfolgreich erledigten Aufgaben ist der Anteil gering, jedoch ist dies ein eindeutiger Hinweis darauf, dass Neueinsteiger in ScOuT gerade für diese Funktion, welche es so in anderen Anwendungen nicht gibt, besser herangeführt werden müssen als an bereits bekannte Funktionen.

Hilfe-Seite und Fehlermeldungen

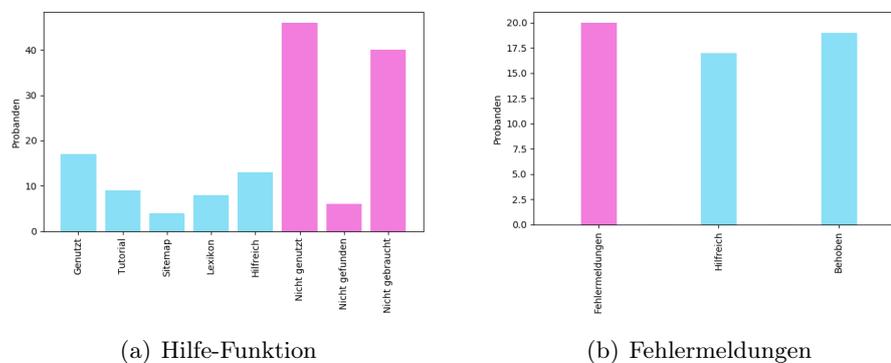


Abbildung 21: Umfrageergebnisse: Hilfe und Fehlermeldungen

Die Ergebnisse der Fragen zur Hilfe spiegeln das zuvor vermutete Problem wieder. Wie in Abbildung 21 a) zu sehen, gab es von 17 Personen, welche die Hilfe nutzten 4, welche sie nicht hilfreich fanden. Alle vier gaben an, dass der Grund dafür eine mangelnde Erklärung rund um das Template war. Die Hilfe einmal genauer betrachtet, zeigt zwar, dass der Begriff im Lexikon deutlich unter dem Event hervorgehoben wird und aufgrund der kurzen Übersicht leicht zu finden sein sollte. Jedoch wird dort lediglich der Zusammenhang zwischen Event und Template erklärt und nicht beschrieben, wo und wie dieses Template genutzt werden kann, da dies nicht Teil des Lexikons ist. Das Tutorial, welches dem Nutzer besagte Bedienung beschreiben soll, ist für das Team, das Event und den Task zwar scheinbar ausführlich genug, nicht jedoch für das Template. Ebenso gibt die Sitemap keine Informationen preis, da sie nur zu App-Seiten verlinkt, das Template selbst aber keine eigenen Seiten hat. Auf Grundlage dieser Ergebnisse in der Umfrage sollte die Hilfe überarbeitet werden und über eine stärkere Einbindung des Template in ScOuT nachgedacht werden. Ein weiterer Punkt, welcher sich bei der Evaluation herausstellte, betrifft den Ort der Hilfe. Sechs Personen gaben an, sie gesucht, aber nicht gefunden zu haben. Eine mögliche andere Position

für die Hilfe wäre dauerhaft oben rechts bei dem Zurück-Button angezeigt zu werden. So wird die Gefahr verringert, dass der Nutzer Hilfe braucht, aber nicht bekommt und das Design wird nicht völlig umgestellt.

Nach den Ergebnissen der Umfrage wurden 20 Personen Fehlermeldungen angezeigt. An dieser Stelle muss jedoch angemerkt werden, dass im Nachhinein festgestellt wurde, dass der Begriff der relevanten Fehlermeldungen hätte präzisiert werden müssen. Ein Serverfehler von Seiten des Host-Services sollte an dieser Stelle nicht abgefragt werden, da diese Fehler nicht in der Hand der Entwickler liegen. Die Fragen zielten stattdessen auf die in Abschnitt 5.2 beschriebenen Fehlermeldungen, welche bei Eingaben in Formularen als rote Hinweise auftauchen können, ab um dieses Missverständnis in zukünftigen Umfragen zu vermeiden, könnte ein Beispielbild an der Frage angebracht werden. Da an dieser Stelle das Ergebnis jedoch sehr positiv ausfällt und wie in der Abbildung 21 b) zu sehen ist, von den 20 Fehlern 17 hilfreich waren und 19 erfolgreich behoben werden konnten, stellt sich lediglich für drei Fehlermeldungen und einen Fehler die Frage, ob die Umsetzung im Sinne der Gebrauchstauglichkeit nicht sinnvoll war. Diese sind im Zusammenhang mit 63 Probanden nahezu vernachlässigbar.

Objekte und Informationspräsentation

Zu den Hauptstrukturen in ScOuT zählen Team, Event und Task. Deshalb wurde in den Fragen 10 und 11 die gewählten Terminologie, der semantische Zusammenhängen zwischen ihnen und die Präsentation der wichtigen Informationen der Objekte abgefragt.

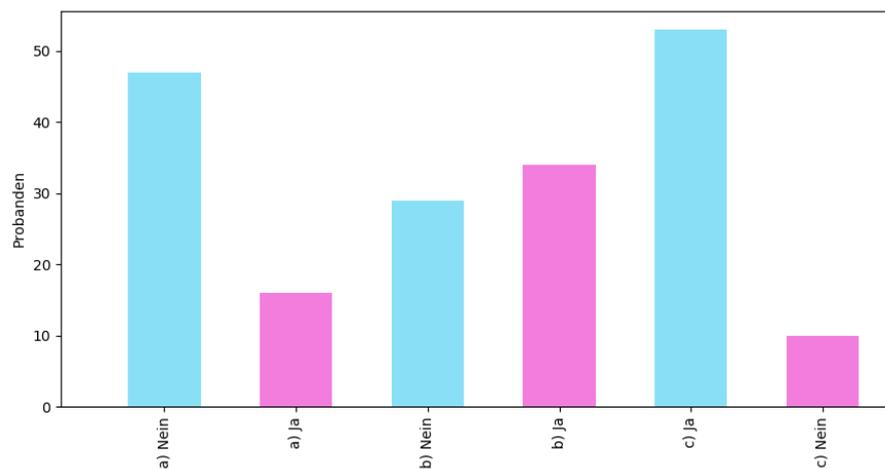


Abbildung 22: Umfrageergebnisse: Informationsdarstellung. Frage a) Richtig = Nein, Falsch = Ja; Frage b) Richtig = Nein, Falsch = Ja; Frage c) Richtig = Ja, Falsch = Nein

Zunächst wurde ein Bild von einer Task-Page gezeigt und dann Fragen

zum Erledigungs-Status und den damit zusammenhängenden Attributen gestellt. Wie in Abbildung 22 zu sehen ist erkannten 47 Probanden richtig, dass der Task noch nicht erledigt ist. Obwohl in der Abbildung 28 im Fragebogen (siehe Anhang) zu sehen ist, wie der Status auf „Unerledigt“ steht, gaben dennoch 16 Personen an, dass der Task bereits erledigt ist. Möglicherweise kann dieser doch recht hohe Irrtum reduziert werden, indem das „Unerledigt“ mit roter Farbe deutlicher hervorgehoben wird, wie es auch mit dem `IonIcon` für unerledigte Task gemacht wurde. Möglicherweise ist auch die Kombination aus Zustand und `IonToggleButton` nicht ganz klar. In einigen Fällen zeigen die Beschriftungen der Buttons an, welcher Zustand erreicht wird, wenn er betätigt wird. In anderen Anwendungen zeigen `ToggleButtons` an, welcher Zustand es gerade vorliegt. Es sollte der verfügbare Oberflächenplatz dahingehend geprüft werden, ob der aktuelle Status „Aktuell Unerledigt“ zusammen mit der Button-Beschriftung „Wechseln zu Erledigt“ oder einer ähnlichen Variante angegeben werden kann oder ob der `ToggleButton` eventuell durch eine andere Button-Form ersetzt werden sollte, wie zum Beispiel ein Checkmark-Kästchen.

Bei der zweiten Frage bezüglich des Zustandes des Reminders wurde ein Fehler begangen, welcher vermutlich ein Mitgrund für das schlechte Ergebnis in der Umfrage ist. Erwartet wurde, dass der Reminder als nicht aktiv gesehen wird. Ein aktiver Reminder würde die rote Nachricht „Reminder: Du solltest das jetzt erledigen!“ angeben. Jedoch endete die Umfrage am 13.09.2019, während das Event nach den Informationen weiter unten auf der Ansicht am 21.08.2019 startet. Da im Reminder angegeben wurde, dass er einen Tag vor Beginn startet, haben vermutlich viele Personen angegeben, dass er bereits aktiv ist. Deshalb muss diese Frage aus der Auswertung ausgeschlossen werden.

Das Ergebnis für die Frage, ob der Task im aktuellen Zustand erledigt werden kann ist dagegen mit 53 richtigen Antworten und 10 falschen Antworten positiv ausgefallen. Fragwürdig ist hier nur die Differenz von sechs Personen, welche sagen, dass der Task bereits erledigt ist, ihn aber scheinbar dennoch jetzt erledigen können.

Auch der gefragte Zusammenhang der Begriffe in Frage 11 lieferte ein zufriedenstellendes Ergebnis. Die drei Lückentexte wurden von 58, 57 und 59 von je 63 Probanden richtig beantwortet. Die grundlegende Terminologie scheint also weitestgehend vom System vermittelt worden zu sein und die Probanden verstanden auch die Zusammenhänge zwischen den Objektstrukturen.

Funktionsumfang

Wie zum aktuellen Entwicklungsstand von ScOuT zu erwarten war, fällt auch die Frage 20 der Umfrage aus und 17 Probanden gaben an, dass der Funktionsumfang noch nicht vollständig ist. Die große Anzahl von 46 Pro-

banden, welche sagten, dass der Funktionsumfang erwartungskonform ist beziehen sich vermutlich ausschließlich auf in den Aufgaben gefragten Funktionen. Positiv ist an dieser Stelle, dass der Funktionsumfang von niemandem als zu komplex benannt wurde. Das bedeutet, dass alle bisher implementierten Funktionen soweit stimmig zueinander passen.

Von den 17 Probanden gaben alle an, welche Funktionen ihnen fehlten. Fünf von ihnen fiel auf, dass zur Eigenschaft Steuerbarkeit noch nicht alle Optionen implementiert wurden. Dies wurde bereits in Abschnitt 6.1 als Ausblick gegeben. Ebenso entsprechend des Ausblicks stellten drei Probanden fest, dass noch keine Zuordnung von Einzelpersonen oder Untergruppen der Orgateams für einen einzelnen Task möglich ist. Ein weiterer Proband erkannte die Probleme mit dem `IonDateTimePicker` und seiner Nutzung am Computer anstelle des Handys. Die noch nicht vollständige Ausschöpfung des Dringlichkeits-Systems ist ein weiteres bereits im Ausblick geplantes Ziel, welches von zwei Probanden erkannt wurde. Ein Proband gab an, dass mehr Details zu den Veranstaltungen gewünscht sind. Diese wurden bereits in Abschnitt 4.3 bedacht, jedoch ebenfalls im Rahmen der Masterarbeit vernachlässigt. Ein weiterer Proband merkte an, dass das Starttutorial zwar eine gute Erklärung darstellt, jedoch, wie ebenfalls im Ausblick angemerkt, als geballte Information am Anfang zu viel ist und besser auf die Pages verteilt werden sollte, auf denen die Informationen benötigt werden. Die letzte bereits bekannte Anforderung an das System ist ein Anmeldesystem für Teilnehmer an der Veranstaltung (siehe Abschnitt 4.2). Dieses wurde ebenfalls von einem Probanden angemerkt. Die Tatsache, dass die noch nicht implementierten Konzeptideen und die Richtlinien für die Gebrauchstauglichkeit auch den Probanden auffielen, bestätigt, dass diese sinnvolle nächste Schritte in der Umsetzung von ScOuT sind.

Neben den bereits bekannten noch fehlenden Funktionen, gab es zusätzlich neue Ideen der 17 Probanden. Darunter eine Chat-Funktion für einzelne Veranstaltungen, Darstellung eines Kalenders für Veranstaltungen und Aufgaben anstelle der Listendarstellung, eine Historie für Aufgaben, wann wer was erledigt hat, sowie Angaben zur Einschätzung der Dauer von Aufgaben.

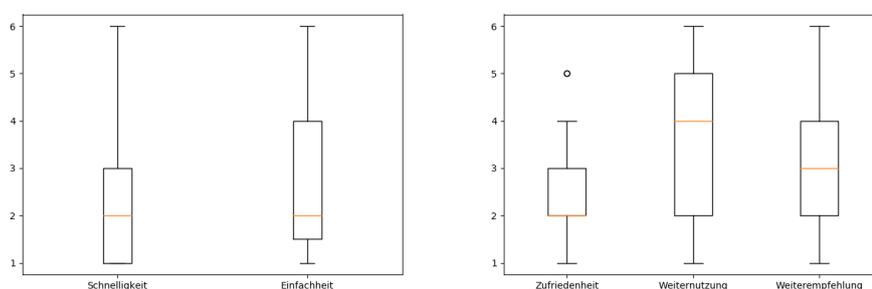
Ein Proband bemängelte zudem, dass es nicht möglich, ist Aufgaben direkt aus Veranstaltungen heraus zu erstellen und die Angabe von Team und Veranstaltung immer wieder hinzugefügt werden müsse. Dieser Hinweis deutet auf eine nicht effiziente Oberflächengestaltung hin. Eine Korrektur dessen sollte jedoch mit wenigen Änderungen möglich sein.

Abschließend wurde noch einmal kritisiert, dass die Funktionen des Templates in der Hilfe nicht ausreichend erklärt wurden.

Ziel des Templates war es den Nutzer besser in der Planung regelmäßig stattfindender Veranstaltungen zu unterstützen. Abbildung 23 a) zeigt das Ergebnis dieses Versuchs aus Frage 19. Die meisten Nutzer gaben an, dass die Erstellung einer Veranstaltung mit dem Template schneller ging als die Ver-

anstellung manuell anzulegen. Dennoch zeigt das Kastendiagramm ebenfalls an, dass es einige Probanden nicht als schneller empfanden. Die neun Probanden, welche die Tendenz vollständig auf die manuelle Erzeugung der Veranstaltung setzten, entsprechen jedoch genau den neun Probanden, welche auch beim Erledigungsgrad der Aufgabe Schwierigkeiten mit dem Template angaben.

Die Abfrage der Einfachheit von Template und manueller Erstellung fällt zwar ebenfalls positiv für das Template aus, neigt sich jedoch etwas mehr in Richtung der händisch angelegten Veranstaltungen. Scheinbar erledigten diese Probanden zwar die Aufgabe, hatten dabei jedoch wie die Neun, welche es nicht oder nur teilweise schafften, Schwierigkeiten dabei.



(a) Template Nutzung im Vergleich zur manuellen Erstellung einer Veranstaltung; Tendenz zu 1 = Template, Tendenz zu 6 = Manuell

(b) Abschließende Fragen; Tendenz zu 1 = positiv, Tendenz zu 6 = negativ

Abbildung 23: Umfrageergebnisse: Templatennutzung und abschließende Fragen

Letzte Fragen

In Abbildung 23 b) sind die Ergebnisse der letzten drei Fragen der Umfrage zu sehen. Der meisten aller Probanden gaben ihre Zufriedenheit bei der Nutzung von ScOuT zwischen gut und befriedigend an. Dieses Ergebnis ist für den aktuellen Stand von ScOuT durchaus positiv. In Zukunft, besonders nach der Fertigstellung aller Grundlagen, sollte eine ähnliche Fragestellung jedoch ein Ergebnis aufzeigen, welches sich stärker von dem Bereich der drei entfernt und näher um die zwei ballt.

Die Probanden gaben dagegen eine deutlich größere Spannweite für das eigene Interesse einer Weiternutzung von ScOuT an. Dies kann daher kommen, dass die Erinnerung an zu erledigende Aufgaben für Einzelpersonen bereits durch einige fertige Produkte abgedeckt ist und nicht jeder die hier entwickelte Anwendungsziele für die Nutzung in Vereinen oder anderen großen Gruppen benötigt. Obwohl das damit fehlende Interesse der eigenen Nutzung von ScOuT bekundet wurde, würden wieder mehr Probanden die Anwendung

an andere weiterempfehlen. Die eigene Zufriedenheit bei der Nutzung scheint damit bestätigt zu sein. Die Probanden vermitteln so, dass sie sich wohl fühlten, jedoch keinen eigenen Verwendungszweck darin sehen, Bekannten, welche dafür Verwendung haben, ScOuT jedoch weiter empfehlen würden.

6.4 Gesamtbild

Auf Grundlage der in Abschnitt 6.3 ausgewerteten Fragen, stellt sich nun die Frage, welches Gesamtergebnis in Bezug auf die Einhaltung der Gebrauchstauglichkeit erreicht wurde.

Die Eigenschaft der Konformität aus den Qualitätskriterien aus Abschnitt 3.2 beschreibt den Grad der Einhaltung von Normen und Vereinbarungen. Im Falle dieser Masterarbeit sind die Normen, welche in den vier Unterabschnitt von Kapitel 3 genannten Eigenschaften. Für die Bewertung des Einhaltungsgades werden die Eigenschaften wie in Tabelle 2 aufgelistet. Diese Auflistung stammt aus der Zusammenfassung der Eigenschaften in Abschnitt 3.5. Die Konformität misst sich folglich an der Einhaltung der zwölf Eigenschaften der Tabelle 2. In der zweiten Spalte dieser Tabelle wird aufgelistet, welche Eigenschaften bei der Implementierung einbezogen wurden. Von den gruppierten Eigenschaften im oberen Viertel wurde lediglich die Nutzeranpassung, also die Individualisierbarkeit der Oberfläche, der Steuerung und des Systems vollständig außen vor gelassen. Die Hilfe und die Steuerung wurden einbezogen, jedoch gibt es bereits, wie im Ausblick in Abschnitt 6.1 genannt, einige Punkte, welche erst in Zukunft geplant sind. Von den Eigenschaften zur Erreichung der Gebrauchstauglichkeit, welche nicht gruppiert wurden, wurde nur die Aufgabenangemessenheit vollständig in der Konzeption der Strukturen in Abschnitt 4.2 und der von Vuolle geforderte Telefon-Kontext nur teilweise einbezogen. Für letzteren gibt es ebenfalls in Abschnitt 6.1 bereits einen Ausblick. Insgesamt wurde folglich eine Konformität von vier vollständigen und drei unvollständigen Eigenschaften der zwölf Punkte angestrebt. Der genau Erreichungsgrad der Konformität hängt nun von der Erreichung jeder einzelnen angestrebten Eigenschaft ab.

Fehler Zur Erinnerung: Beim zusammengefassten Thema geht es um die Vermeidung, Erkennung, Darstellung und Behebung von Fehlern (siehe Abschnitt 3.5). Nach den Fragen zu den Fehlermeldungen (siehe vorangehenden Abschnitt „Hilfe-Seite und Fehlermeldungen“) und den Ergebnissen des Erledigungsgrades der Aufgabe, bei welcher lediglich ein einzelner Proband bei einem einzelnen Aufgabenschritt angab, dass eine Funktion nicht funktionierte, wird der Versuch diese Richtlinien einzuhalten als erfolgreich angesehen.

Hilfe Der Erfolg der Hilfe-Seiten dagegen ist eher mäßig, da, wie in den vorangehenden Abschnitt mehrfach festgehalten, die Hilfe für die Funkti-

Eigenschaft	Einbezug bei der Implementation	Erreichungsgrad
Fehler	✓	Erfolgreich
Hilfe	✓, ✗	Leicht verbesserungswürdig
Lernen	✓	Leicht verbesserungswürdig
Informationsaufbau	✓	Leicht verbesserungswürdig
Steuerung	✓, ✗	Unzureichend → Prüfung ausstehend
Nutzeranpassung	✗	–
Einheitlichkeit	✓	Erfolgreich
Aufgabenangemessenheit	✓	Sinnvoll
Attraktivität	✗	–
Installation	✗	–
Telefon-Kontext	✓, ✗	Sinnvoll
Sicherheit	✗	–

Tabelle 2: Überblick der Richtlinien und Eigenschaften in Bezug auf ihre Einhaltung und ihren Erreichungsgrad (✓ = einbezogen; ✗ = nicht einbezogen)

onsbeschreibung, die Funktionsanwendung und die Funktionsplatzierung des Templates unzulänglich ist.

Lernen Aufgrund der mangelhaften Hilfe bezüglich des Templates und der vereinzelt Angabe, dass eine Funktion nicht verstanden wurde (siehe Abbildung 20), lässt sich ableiten, dass an der Lernförderlichkeit von ScOuT noch zu arbeiten ist. Die Ergebnisse sind jedoch nicht vollständig schlecht ausgefallen, sondern lediglich nicht perfekt.

Informationsaufbau Die Abfrage der Informationsdarstellungen der Taskansicht wurde nach den Ergebnissen der Umfrage (siehe vorangehender Abschnitt „Objekte und Informationspräsentation“) zwar größtenteils erfolgreich richtig beantwortet, jedoch wurden anhand der Ergebnisse auch bereits Verbesserungsvorschläge gemacht. Dies spiegelt sich auch in dem Erledigungsgrad wider, bei welchem zumindest ein Proband angab, dass er nicht wusste, wo er einen Task erledigen soll. Auch wurden die Informationsdarstellung bezüglich des Templates nach dem Erledigungsgrad der Aufgaben nur teilweise erreicht. Dagegen lieferte die Frage zur Terminologie ein sehr positives Ergebnis.

Steuerung Zur Steuerung wurde keine direkte Frage gestellt, da ein großer Bestandteil das Bearbeiten, Löschen und ähnliche Funktionen beträgt, welche noch nicht implementiert wurden. Ohne also eine Frage gestellt zu haben, kann hier direkt gesagt werden, dass dies nur im aller nötigsten Ansatz eingehalten wurde und noch nicht erfolgreich sein kann. Zusätzlich wurde bei der Frage des Funktionsumfangs von einem Probanden angegeben, dass das System bei der Erstellung einer Aufgabe zu wenig Unterstützung beziehungsweise eine zu umständliche Steuerung für die Eingabe von Team und Veranstaltung verlangt. Zur Optimierung der Navigationswege und Anordnung der Bedienelemente, wodurch ebenfalls die Steuerung des Nutzers verbessert wird, könnte die Methode des Baums der Wahrscheinlichkeit angewendet werden [Ham06]. Diese erfasst nicht aufgrund von Vermutungen, wie es bisher der Fall war, sondern aufgrund von Statistiken, welche Wichtigkeit die Bedienelemente und Navigationspfade haben anhand der Häufigkeit der Nutzung.

Einheitlichkeit Das Ergebnis 11 ist ein Indikator dafür, dass die Einheitlichkeit erfolgreich umgesetzt wurde und die Überlegungen zur Terminologie sinnvoll waren.

Aufgabenangemessenheit und Bildschirmgröße (Telefon-Kontext) Diese beiden Einzelpunkte wurden zwar bei der Entwicklung einbezogen, jedoch nicht explizit in der Umfrage abgefragt. Zur Aufgabenangemessenheit kann jedoch gesagt werden, dass die Fülle an bereits existierenden und auch viel genutzten Anwendungen die Annahme stärkt, dass diese technische Unterstützung zu Planung von Aufgaben sinnvoll ist. Obwohl sich bewusst dazu entschieden wurde eine Mobile-Umfrage durchzuführen, weshalb das iFrame auf eine Bildschirmgröße reduziert wurde, kann diese Art der Abfrage jedoch nicht vollständig mit der Nutzung am Smartphone verglichen werden. Die Anwendung würde sich an die korrekte Größe des Smartphone anpassen, wohingegen das iFrame immer seine exakte Größe beibehält, weshalb es möglich ist, dass bei einzelnen Smartphones der Platz auf dem Display nicht ausgenutzt wurde und vielleicht sogar überschritten wurde. Da jedoch alle Objektinformationen, Strukturen und Funktionen erfolgreich auf dieser Größe platziert wurden, ist die Umsetzung vermutlich sinnvoll erfolgt.

7 Fazit

Ziel dieser Masterarbeit war es eine hybride App basierend auf einem geeigneten Konzept zu entwickeln, um Vereine, speziell die Pfadfinder, bei ihrer ehrenamtlichen Arbeit zu unterstützen und die Abläufe in solchen Organisationen zu verwalten. Zu diesem Zweck wurde die Anwendung ScOuT unter dem Gesichtspunkt der Gebrauchstauglichkeit implementiert, um den Nutzern ein effektives, effizientes und zufriedenstellendes Hilfs-Tool zur Verfügung zu stellen.

Rückblickend wurden verschiedene Schritte durchlaufen um diese Masterarbeit vorzubereiten und durchzuführen. Im eigenen Pfadfinder-Stamm gab es mehrere Personalwechsel, wodurch verschiedene Schwierigkeiten bei der Ausführung der Aufgaben auftraten. Das Interesse daran diese Hindernisse zu mindern war groß, doch der Einbezug von digitalen, statischen Hilfsmitteln reichte nicht aus. Auch in Gesprächen mit dem DiBü wurden Probleme ermittelt. Während das DiBü Schwierigkeiten hatte eine Fülle von über 40 Veranstaltungen im Jahr zu bewältigen, hatte der Stamm überwiegend mit Unerfahrenheit und fehlerhafter Kommunikation zu kämpfen. Auch die Verwendung von bereits existierenden Anwendungen brachte weder das DiBü noch den Stamm weiter. Diese Erkenntnisse wurden ausführlich analysiert und in den Kapiteln 1 und 2 erläutert. Ergebnis dieser Analyse ist eine Auflistung verschiedener Ursachen für die Problematiken im Alltag des Stammesleben, allen voran die Unerfahrenheit einiger Mitglieder in Bezug auf die Planung von Veranstaltungen und die Ausübung von Ämtern. Des Weiteren wurde eine kritische Analyse bereits existierender Planungstools durchgeführt, bei welcher es darum ging deren Ist-Zustand zu ermitteln. Dieser Zustand wurde in Bezug auf die Bedürfnisse der Pfadfinder als unzureichend eingestuft, da zwar grundlegende Funktionen wie das Anlegen von Aufgaben oder deren Zusammenschluss möglich sind, jedoch die Wiederverwendung solcher Strukturen nur umständlich zu handhaben ist. Die grundlegende Idee von ScOuT besteht darin, den Nutzer aktiv bei seiner Planung durch Erinnerungen der App zu unterstützen und die Planung mittels Templates für andere Mitglieder des Vereins wiederverwendbar zu machen.

Der Schwerpunkt lag dabei auf der für den Nutzer optimierten Anwendung. Denn der Benutzer ist derjenige, welcher die Anwendung gerne bedienen soll. Wird die Anwendung für den Nutzer ungeeignet dargestellt, so ist sie nutzlos und verursacht meist mehr Umstände, anstatt sie zu beseitigen. Die von der ISO definierte Gebrauchstauglichkeit für die Benutzeroberfläche entspricht dabei genau dem angestrebten Ziel. Eine Anwendung soll effizient sein, also den Nutzer wenig Aufwand kosten. Eine Anwendung soll effektiv sein, wodurch sie dem Nutzer helfen soll ein Ziel möglichst genau und vollständig zu erreichen. Zuletzt umfasst die Gebrauchstauglichkeit die Zufriedenheit des Nutzers bei der Bedienung der Anwendung. In die-

sem Zusammenhang wurden Recherchen zur Erreichung und Prüfung der Gebrauchstauglichkeit durchgeführt. In Kapitel 3 erfolgt die Auflistung von verschiedenen Richtlinien, welche dazu dienen die Gebrauchstauglichkeit zu erreichen. Sie definieren dafür nötige Teilziele präziser und untergliedern Effizienz, Effektivität und Zufriedenheit unter anderem auf die Komponenten der Benutzeroberfläche. Zu den betrachteten Richtlinien gehört die ISO 9241-110, welche ein Unterabschnitt der Norm für die Gebrauchstauglichkeit ist, die ISO/IEC 25000, welche als Bestandteil der Qualitätskriterien einen engeren Bezug zur Gebrauchstauglichkeit ist, die zehn Heuristiken von Nielsen, welcher als Pionier der Gebrauchstauglichkeit gilt, als auch die neun Dimensionen von Vuolle, welcher sich speziell mit der Gebrauchstauglichkeit in Bezug auf das Mobiltelefon beschäftigt. Die daraus resultierenden Eigenschaften wurden im letzten Abschnitt des Kapitels für die weitere Planung der Anwendung in übergreifende Themen zusammengefasst.

Nach der Recherche für die Grundlagen wurde zu Beginn der konkreten Konzeption ein Interview mit vier Stämmen der Diözese Limburg und dem DiBü durchgeführt, um den Bezug zum zukünftigen Nutzer zu steigern und ihn in die Erhebung der Anforderungen einzubeziehen. Auf der Grundlage der Anforderungen und Einfluss durch bereits existierende Planungstools wurde ein Konzept für ScOuT entwickelt, welches zu großen Teilen in Kapitel 4 zusammengefasst ist. Dabei ging es in erster Linie um Strukturen und damit verbundene Funktionen der Aufgaben, Veranstaltungen und Templates. Wichtige Bestandteile bildeten neben der einfachen Wiederverwendbarkeit von Templates die Funktionen für die Erinnerungen des Nutzers an unerledigte Aufgaben. Diese beiden Konzeptideen sollten ScOuT von bereits existierenden Planungstools differenzieren und die Schwierigkeiten bei der ehrenamtlichen Arbeit der Pfadfinder schmälern. Die Kernpunkte der Konzeption wurden letztendlich in der Implementierung des Backends der hybriden App umgesetzt.

Mit dem Backend als Grundlage wurde anschließend eine Benutzeroberfläche entwickelt, welche sich nach den Teilzielen der Gebrauchstauglichkeit richtet. Hierfür wurden zunächst in Kapitel 5 die in dem Framework Ionic verfügbaren Oberflächenelemente kategorisiert, um Struktur-, Kontroll-, Eingabe- und Präsentationselemente zu trennen. Im Anschluss wurde unter dem Einsatz von verschiedenen Methoden die Einteilung der App-Seiten, sowie die Gestaltung der Inhalte jeder einzelnen Seite geplant. Schwerpunkte waren dabei Anordnung von Informationen und Funktionen, sowie die gesonderte Betrachtung von Feedback, Hilfe und Fehlermeldungen. Während der Implementierung der Oberfläche wurde die Methode des Lauten Denkens wiederholt angewendet, um frühzeitig grobe Mängel in der Oberflächengestaltung aufzudecken.

Die daraus resultierende Anwendung ScOuT wurde im letzten Abschnitt der Masterarbeit evaluiert. Das für die Evaluation durchgeführte Vorgehen wird in Kapitel 6 beschrieben. Zunächst wurde abgegrenzt, welche Berei-

che der Anwendung bereits funktionstüchtig sind und welche Teilziele der Gebrauchstauglichkeit einbezogen wurden, da nur der aktuelle Stand von ScOuT in der Evaluation betrachtet werden kann. Als Evaluationswerkzeug wurde eine Benutzerumfrage gewählt. Die darin enthaltenen Fragen wurden beeinflusst durch bekannte Umfragen zur Gebrauchstauglichkeit auf die abgegrenzten Bereiche aufgestellt. Darunter gab es Fragen zur Hilfe, zu Fehlermeldungen, zur Anordnung der Informationen, zur Terminologie, als auch zum Template. Auf Grundlage der Benutzerumfrage wurde abschließend die Auswertung durchgeführt, welche den Erreichungsgrad der Gebrauchstauglichkeit widerspiegelt.

Ziel der Masterarbeit war es eine hybride App mit effizienten, effektiven und zufriedenstellenden Eigenschaften zur Erleichterung der Vereinsarbeit zu entwickeln. Durch die Verwendung einer hybriden App, wurde eine Nutzbarkeit für die breite Masse erreicht. Da existierende Planungstools den Bedürfnissen in der Vereinsarbeit nicht gerecht wurden, sollten gerade die speziellen Unterschiede von ScOuT, welche in Zusammenarbeit mit den Pfadfindern der Diözese Limburg konzipiert wurden, auch dieses Kriterium erfüllen. Dabei wurde vorausgesetzt, dass die Erfahrung der Pfadfinder sie zu Experten macht, welche wissen, was ihre eigenen Bedürfnisse eigentlich sind. Die Erreichung der Gebrauchstauglichkeit lässt sich durch die Umfrage evaluieren. Die Einhaltung aller Teilziele war im Rahmen der Masterarbeit nicht möglich, weshalb bereits gesagt werden kann, dass der Einbezug der Gebrauchstauglichkeit bei der Gestaltung der Benutzeroberfläche nicht voll erzielt wurde. Dagegen ist innerhalb der gesteckten Teilziele eine Tendenz zum angemessenen Erreichen der Gebrauchstauglichkeit festzustellen. Die Erreichung bezüglich Fehler und Einheitlichkeit wurde als erfolgreich eingestuft. Hier gibt es demnach kaum bis keine Verbesserungen, welche dieses Potential steigern würden. Bezüglich der Hilfe, dem Lernen und dem Informationsaufbau ist die Bezeichnung „erfolgreich“ noch nicht möglich, jedoch ergab die Umfrage, dass es dort nur geringer Verbesserungen bedarf, um ebenfalls, die Ziele zu erreichen. Lediglich die Steuerung wurde als nicht ausreichend bewertet. Dies kommt daher, dass sie sehr elementar bei der Oberflächengestaltung zu sein scheint und dennoch nicht vollständig bei der Entwicklung einbezogen wurde und zudem die einbezogenen Aspekte nicht ausreichten.

Kritisch betrachtet, befasst sich diese Arbeit jedoch nur mit einem Ausschnitt der Umsetzung der Gebrauchstauglichkeit. Es gibt weitere Methoden, um die Teilziele zu erreichen. Ebenso gibt es weitere Richtlinien, welche zusätzliche Teilziele definieren. Es wurde sich zudem nicht tiefergehend mit der Prüfung der Gebrauchstauglichkeit auseinandergesetzt, sondern nur eine einzelne Methode mit wenigen Anpassungen angewendet. Daraus ergibt sich, dass sowohl weitere, konzeptionelle Schritte für die Planung der Benutzeroberfläche durchgeführt werden müssen, sowie zusätzliche Prüfungen für die

Erreichung der Gebrauchstauglichkeit. Mögliche weitere Methoden sowohl zur Konzeption als auch zur Überprüfung könnten zum Beispiel während der Umfrage oder der Nutzung der App erfolgen. Diese passiven Verfahren laufen automatisch im Hintergrund und der Nutzer ist nicht gezwungen aktiv die Methode anzuwenden, wie das Beantworten eines Fragebogens. Eine von N. A. Hamacher (siehe [Ham06]) vorgestellte Methode befasst sich mit der Wahrscheinlichkeitsmessung der Navigationspfade. Dabei wird gemessen, welche Pfade der Nutzer von App-Seite A zu App-Seite B beschreitet. Diese Methode dient dazu die Pfade zu optimieren, um dem Nutzer Klicks und Zeit zu sparen. Dies wird erreicht durch eine andere Anordnung der Navigations-Funktionen. Genauso schlägt N. A. Hamacher eine Methode zur Messung der Fehl-Navigations vor. Dies gibt Aufschluss darüber, ob Funktionen irreführend beschriftet oder ungeeignet platziert sind. Können Fehl-Navigations mittels der Zurück-Funktion aufgezeichnet werden. Passive Methoden können jedoch erst dann zur Umsetzung der Gebrauchstauglichkeit angewendet werden, wenn die Anwendung bereits in der Benutzung ist. Eine Methode, welche bereits bei der ersten Konzeption verwendet werden kann, wird ebenfalls von N. A. Hamacher vorgestellt und dient der exakten Untergliederung einer jeden App-Seite. Diese wurde bereits teilweise angewendet, jedoch nur grob für Informationen, Funktionen und die Strukturkarten. Hamacher betreibt dies noch kleinschrittiger und zerlegt nicht nur die Oberfläche in ihre Elemente, sondern die Oberflächenelemente selbst in ihre Attribute. Dadurch können beispielsweise automatisiert Kontraste evaluiert werden. Ein weiterer Vorschlag von N. A. Hamacher bezieht sich auf Metriken. So können zum Beispiel die Anzahl der Oberflächenelemente pro App-Seite gemessen und miteinander verglichen werden. Doch hierfür müssen zu den Metriken Vergleichswerte aus anderen Anwendungen verfügbar sein. Generell sollten weitere Vergleiche mit bereits existierenden Planungstools angestrebt werden. Als letzten Vorschlag zur Prüfung und Erreichung der Gebrauchstauglichkeit ist noch die Website <https://blog.seibert-media.net/> zu nennen, welche 99 Teilziele definiert. Die Erreichung der Teilziele lässt sich durch die 99 Ja-/Nein-Fragen prüfen. Wichtig bei allen Methoden ist jedoch, dass die mit Methode A ermittelte Benutzeroberfläche nicht mit Methode A überprüft wird, sondern mit Methode B, da sonst die Ergebnisse redundant sind. Zur Einarbeitung der weiteren Methoden, sowie der in der Masterarbeit reduzierten Strukturen und Funktionen sollte zur Nutzung der Anwendung in Zukunft weitere Zeit eingeplant werden.

Anhang

A Große Abbildungen

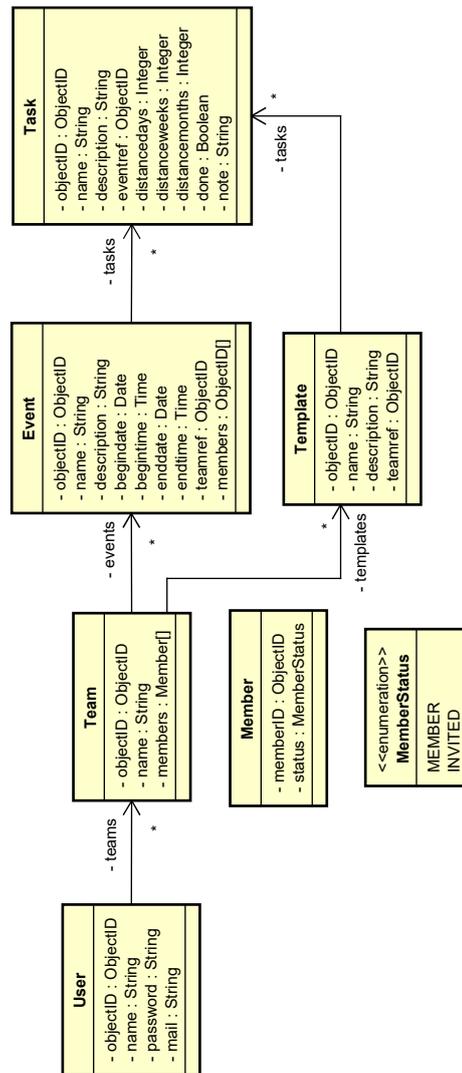


Abbildung 24: Implementierte Struktur.

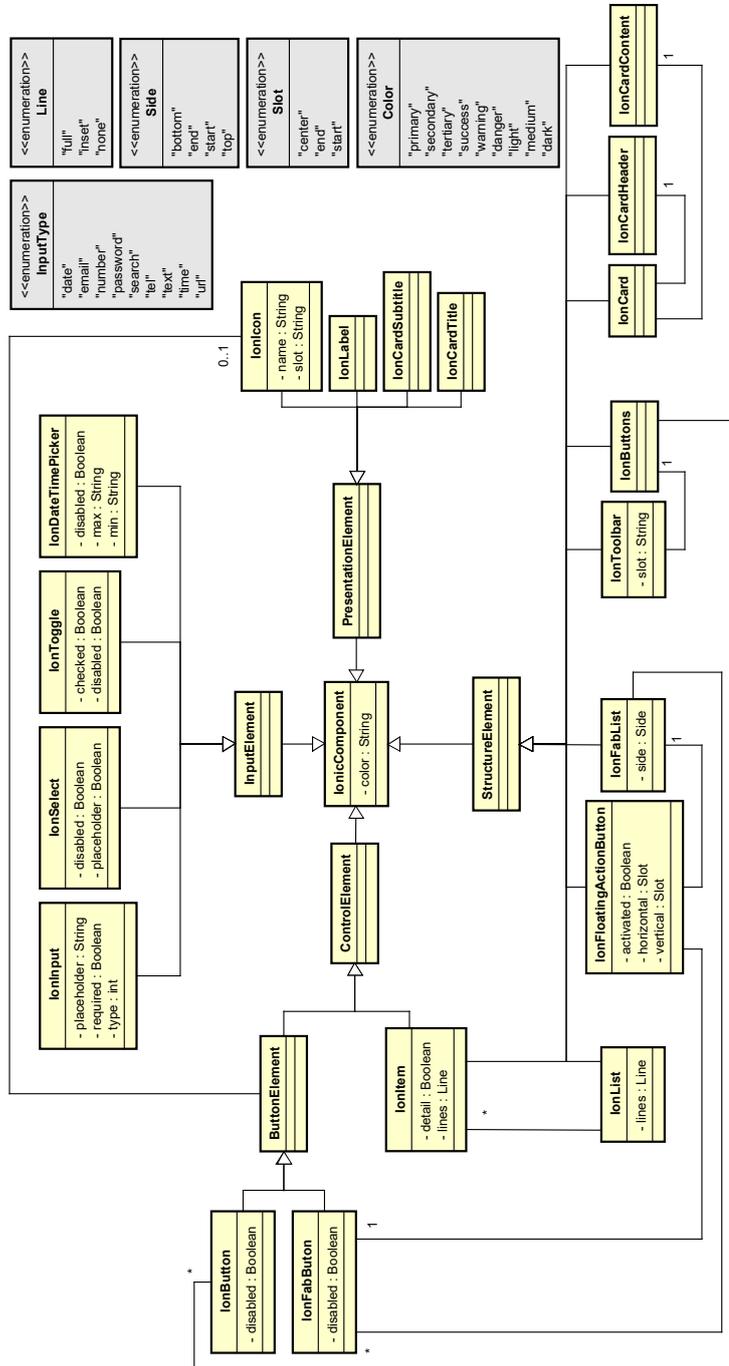


Abbildung 25: Übersicht über alle Ionic Komponenten zur Gestaltung des IonContents

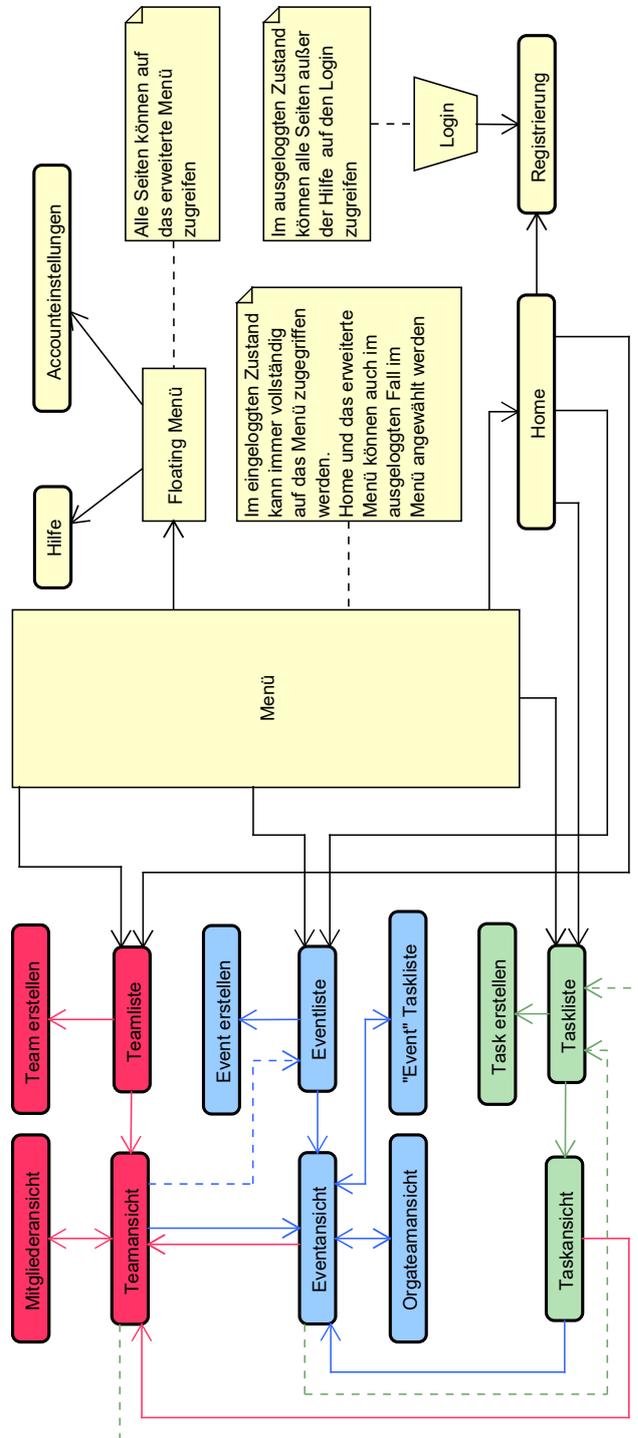


Abbildung 26: Übersicht über alle Verbindungen zwischen den Pages

B Interview-Ergebnisse

Ämterliste

Exklusive Ämter der Diözesanebene

Titel	Verantwortlichkeit
Anhänger-Verwaltung/ Buswart	Verleih des Anhängers/Bus und Pflege des Anhängers/Bus
Chronik-Beauftragter	Erstellung, Pflege und Prüfung der Chronik (Homepage)
e.V. Vorstand	Vorstand des offiziellen Vereins; Eigentümer aller Finanzmittel
Gruppenleitung	Gestaltung und Durchführung der Gruppenstunden
Haus und Hof Team/ Hausmeister	Pflege und Planung der Pflege der Räumlichkeiten für Gruppenstunden und andere Treffen
Hausvermietung	Vermietung der Räumlichkeiten
Homepage-Administration	Weiterentwicklung der Homepage; Erstellen von Sicherheitskopien;
Kalender-Beauftragter	Veröffentlichen der Termine im Homepage-Kalender
Kassenwart	Prüfen der Kasse des e.V.
Materialwart (inkl. Zeltwart)	Pflege aller Gegenstände, wie z. B. Werkzeug oder Zelte
Mitgliederverwaltungs-Beauftragter	Aktualisierung der Mitgliederdaten in dem Tool NaMi
Presse-Beauftragter	Schreiben von News-Artikeln über vergangene und zukünftige Aktionen und Veröffentlichung dieser auf der Homepage-Startseite, Kirchenseite und verschiedenen sozialen Netzwerken
Rüsthauswart/ Rüsthaus-Sammelbesteller	Verkauf von Rüsthaus-Waren an Mitglieder
Vorstand (inkl. Kuratenamt)	Leitung des Stammes; Planung von Leiterrunden und ähnlichen Aktionen; Betreuung der Mitglieder; ...
Zeltvermietung	Vermietung von Zelten
Zuschuss-Beauftragter	Beantragung und Verwaltung der Zuschusslisten

Veranstaltungsliste

Aktionen, welche in allen Stämmen vorkommen:

Titel	Wiederholung/ Zeitraum	Beschreibung
Aufstieg und Versprechen	jährlich / Abend, Tag	Aufstufen der Jugendlichen in die nächste Stufe und stammesinterne Feierlichkeit
Elternabend	unregelmäßig / Abend	Austausch von Informationen mit den Eltern
Gruppenstunde	wöchentlich / Abend	Betreuung der Jugendlichen
Jahresplanung	jährlich / Abend, Tag, Wochenende	Planung der Aktionen und Verteilung der Ämter
Leiterrunde/ Stammesleitung	monatlich / Abend	Besprechung aller anliegenden Aktionen und Aufgaben
Stammesversammlung (inkl. Elternabend)	jährlich / Abend, Tag	Entlastung des Vorstandes; Wahl neuer Vorstände; Kindermitbestimmung
Stammesvorstandstreffen	monatlich, unregelmäßig / Abend	Besprechung von Amts-Aufgaben
(Stufen-) Lager	jährlich (und häufiger) / Mehrere Tage	Betreuung der Jugendlichen

Aktionen, welche nur in einzelnen Stämmen durchgeführt werden

Titel	Wiederholung/ Zeitraum	Beschreibung
72h Aktion	unregelmäßig / Wochenende	Erledigung einer gemeinnützigen, sozialen oder naturschützenden Aufgabe innerhalb der 72 Stunden
Abschlusswochenende	jährlich / Wochenende	Abschlusslager für alle Gruppen
Adventswanderung	unregelmäßig / Tag	Gemütliche Wanderung
Altpapiersammlung	jährlich / Tag	Sammlung des Altpapiers in Wohngebieten
Blumentepich	jährlich / Abend + Vormittag	Legen eines Blumenbildes für die Fronleichnams-Prozession
e.V. Bildungsreise/ Fördervereinslager/ Rovertour	zweijährlich / Mehrere Tage	Gemütlicher, freundschaftlicher Urlaub/ Lager für die Mitarbeiter
Fest der kath. Jugend	jährlich / Abend	Gemeinsamer Jugendgottestdienst mit anschließender Feier mit Messdienern, Chor und Pfadfindern
Gemeindefest	unregelmäßig / Tag, Wochenende	Teilnahme an einem Gemeindefest
Glühweinparty	jährlich / Abend	Rover veranstalten Party
Halbjahresreflexion	jährlich / Abend	Gemütlicher Abend und Zwischenreflexion des letzten Halbjahres
Jahresabschlussfeier	jährlich / Abend	Gemütlicher Abend als Dankeschön für alle Mitarbeiter
Jugendsammelwoche	jährlich / Woche	Sammeln von Geld für sozialen Zweck
Leiterfunwochenende/ Leiterwochenende/ Motivationswochenende	jährlich / Wochenende	Gemütliches Wochenende zur Motivation der Mitarbeiter
MüllMoritz	zweimal im Jahr / Tag	Pflege des Pfadiheims und Gartens
Nikolaus	jährlich / Mehrere Abende	Besuch von Familien im Nikolauskostüm
Stadtfest	unregelmäßig / Tag, Wochenende	Teilnahme mit Attraktion oder Stand an einem Stadtfest
Stammestag	jährlich / Tag	Ausflugstag für den ganzen Stamm

Aktionen, welche von der Diözesanebene für die Stammesebene organisiert werden:

Titel	Wiederholung/ Zeitraum	Beschreibung
Ausbildung und Seminare	mehrmals im Jahr / Tag, Wochenende	Schulungen für alle Mitarbeiter
Diözesanversammlung	jährlich / Wochenende	Entlastung und Wahl des Diözesanvorstandes; Besprechung aller Diözesanweiten Fragen; ...
Friedenslicht	jährlich / Tag	Übersendung des Friedenslichtes über die Pfadfinder in die verschiedenen Gemeinden
Stammesvorständekonferenz	jährlich / Wochenende	Besprechung der Stammesvorstände in der Diözese untereinander
Stufenlager	jährlich / Wochenende	Diözesanweites Lager für die jeweiligen Stufen
Stufenkonferenz	jährlich / Wochenende	Planungstreffen der Stufenvertreter und Referenten

C Fragebogen

Seite 1

Lieber Teilnehmer, diese Umfrage ist Teil der Masterarbeit „Konzeption und Entwicklung einer hybriden App zur Organisationsverwaltung“ von Lisa Werner an der Universität Koblenz-Landau. Das im Rahmen der Arbeit entwickelte Tool unterscheidet sich dahingehend von anderen Tools, dass die vordefinierten Strukturen explizit für Teams (z.B. Vereine) und wiederkehrende Events entwickelt wurde.

Ziel der Umfrage ist nun die Evaluation der Usability des Organisationsverwaltungstools. Die Dauer der Umfrage beträgt ungefähr 20 Minuten.

Vielen Dank für deine Teilnahme!

1. Datenschutz: Diese Umfrage erfolgt anonym, sodass alle Angaben nicht zurückverfolgbar sind. Die Angaben selbst sind alle nicht personenbezogene Daten und unterliegen somit nicht der DSGVO.

Dennoch der Hinweis: Diese Daten werden ausschließlich für die Forschung innerhalb der Masterarbeit verwendet und im Rahmen der Auswertung innerhalb der Masterarbeit veröffentlicht, welche zukünftig auch online verfügbar sein wird.

Seite 2

Bitte starte mit allgemeinen Fragen zu dir als Person.

2. Wie alt bist du?

3. Nenne dein Geschlecht:

Männlich | Weiblich | Divers

4. Welche Tools hast du bereits am Handy oder Tablet verwendet um Events zu planen?

Trello | ClickUp | Asana | ToDoIst | Confluence (Atlassian) |

Wunderlist | InLoox | Sonstiges _____ |

Ich habe noch keine Erfahrung mit solchen Tools

5. Welche Tools hast du bereits am Computer verwendet, um Events zu planen?

Trello | ClickUp | Asana | ToDoIst | Confluence (Atlassian) |
Wunderlist | InLoox | Sonstiges _____ |
Ich habe noch keine Erfahrung mit solchen Tools

6. Wofür würdest du ein Tool zum Planen von Events nutzen wollen?

Vereinsarbeit | Beruf | Für mich als Einzelperson |
In privaten Gruppen (z.B. Familie, Freundeskreis) |
Ich weis es noch nicht

Seite 3

Zur Beantwortung der nachfolgenden Fragen ist zuvor eine Nutzung des Tools notwendig. Dieses lädt automatisch innerhalb dieser Umfrage im Anschluss der Aufgabenstellung und es muss keine andere Website besucht oder etwas installiert werden! Deshalb erledige bitte folgende Aufgaben der Reihe nach:

1. Registriere dich in dem Tool (Siehe weiter unten)

Hinweis: Die Registrierungsdaten dürfen frei erfunden sein. Du erhältst auch bei der Verwendung deiner richtigen Mail-Adresse keine E-Mails. Du solltest dir die Daten jedoch für den Fall eines Logouts merken, damit du dich wieder einloggen kannst. Alle erstellten Accounts werden nach Beendigung der Masterarbeit gelöscht.

2. Erstelle ein Team / oder nimm die Teameinladung an

3.1 Erstelle für das Team, welches du eben aufgemacht hast, ein neues Event (Beispiel: Zeltlager 2019)

3.2 Erstelle für das Event drei Tasks (Beispiel: Zelte prüfen; Einladung versenden; Essen planen; Programm erstellen; Platz buchen)

4. Setze den Status von zwei Tasks des ersten Events auf „Erledigt“

5.1 Erstelle aus einem der Events ein Template (Beispiel: Zeltlager)

5.2 Erstelle ein Event aus dem Template (Beispiel: Zeltlager 2020)

5.3 Erledige einen der Task von diesem neuen Event

Hier erscheint nun die Testoberfläche: Achtung: Das Laden der App könnte in Einzelfällen einige Zeit in Anspruch nehmen!

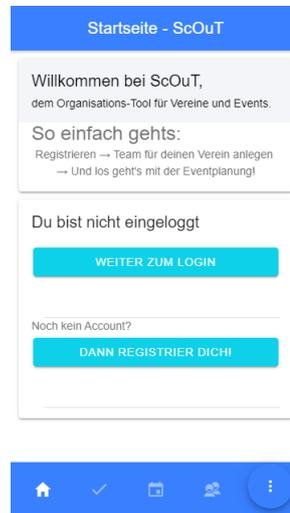


Abbildung 27: Startseite als Platzhalter für IFrame im Fragebogen

Seite 4

Nach der Erledigung der Aufgabe, bitte ich dich, jetzt noch schnell den Fragebogen auszufüllen. Die weitere Dauer sollte jetzt nur noch 10 Minuten dauern.

7. Wie vollständig hast du die Aufgabe erledigt?

Vollständig erledigt | Teilweise erledigt | Komplett unerledigt

Seite 5

8. Spezifiziere, welche Aufgabenteile du unerledigt gelassen hast:

Matrix Spalten-Namen: Erledigt | Vielleicht erledigt | Unerledigt

Matrix Zeilen-Namen: Erstellung eines Teams /Beitritt in ein Team |

Erstellung der Events (Ohne Template) | Erstellung der Events

(Mit Template) | Erstellung der Tasks | Erledigung der Tasks

9. Welche Gründe hängen deiner Meinung nach damit zusammen, dass die Erledigung der Aufgabe länger dauerte oder ganz unerledigt blieb?

Matrix Spalten-Namen: Funktion nicht gefunden | Funktion nicht

verstanden | Funktion hat nicht funktioniert | Aufgabe nicht

verstanden | Sonstiges | Nicht relevant (weil erledigt)

Matrix Zeilen-Namen: Erstellung eines Teams /Beitritt in ein Team |

Erstellung der Events (Ohne Template) | Erstellung der Events

(Mit Template) | Erstellung der Tasks | Erledigung der Tasks

Seite 6

10. Weißt du, wo du welche Informationen finden kannst?

Betrachte für die Bearbeitung der Aufgabe das nachfolgende Bild

Ist der Task bereits erledigt? Ja | Nein

Ist die Erinnerung für die Erledigung des Task bereits aktiv? Ja | Nein

Kann man den Task jetzt erledigen? Ja | Nein

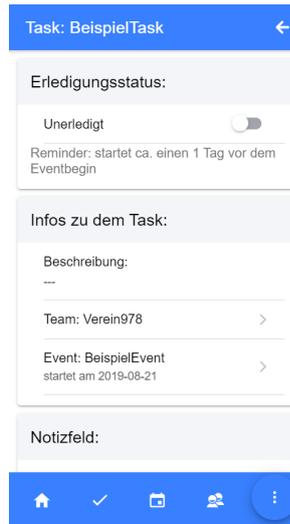


Abbildung 28: Fragebogen-Inhalt: Beispiel Task angezeigt in seiner eigenen Task-Übersichts-Seite.

11. Kannst du die Zusammenhänge zwischen den Begriffen reproduzieren?
Wähle für die mit den [Klammern] gekennzeichneten Lücken den passenden Begriff aus:

Matrix Spalten-Namen: Team(s) | Event(s) | Task(s)

Matrix Zeilen-Namen:

Teams können gemeinsam verschiedene [AAA] organisieren. |

Evnts werden durch viele einzelne [BBB] geplant. |

Events werden immer von genau einem/einer [CCC] durchgeführt.

Seite 7

12. Hast du die Hilfe-Funktion verwendet?

Ja | Nein

13. Wurden Fehlermeldungen angezeigt?

Ja | Nein

Seite 8 und 9

14. Wofür hast du die Hilfefunktion verwendet?

Starttutorial | Sitemap | Begriffsklärung

15. War die Verwendung der Hilfe-Funktion hilfreich?

Ja | Nein _ _ _ _

17. Warum hast du die Hilfe-Funktion nicht genutzt?

Brauchte sie nicht | Habe sie nicht gefunden

16. / 18. Bitte gib weitere Angaben dazu an, dass dir Fehlermeldungen erschienen sind:

Haben die Fehlermeldungen dir gesagt, wie du den Fehler beheben kannst?

Ja | Nein

War die Fehlerbehebung erfolgreich? Ja | Nein

Seite 10

19. Wie empfandest du den Unterschied vom manuellen Anlegen eines Events im Vergleich zum Anlegen mit einem Template?

Ich war schneller beim Erstellen eines Events mit: Template | ... | Manuell

Ich fand das Erstellen eines Events einfacher mit: Template | ... | Manuell

20. Wie vollständig erscheint dir der Funktionsumfang des Tools zum Organisieren von Events?

Unerwartet unvollständig (Gib bitte Funktionen an, welche dir fehlen) _ _ _ _

Erwartungskonform vollständig

Überfüllt/ Komplex (Gib bitte Funktionen an, welche dir dieses Gefühl vermittelt haben) _ _ _ _

21. Wie zufrieden warst du bei der Verwendung der App?

Sehr zufrieden 1 | 2 | 3 | 4 | 5 | 6 Unzufrieden

22. Wie wahrscheinlich ist es, dass du das Tool in Zukunft für die Planung von Events nutzt?

Sehr zufrieden 1 | 2 | 3 | 4 | 5 | 6 Unzufrieden

23. Wie wahrscheinlich ist es, dass du das Tool weiter empfiehlst?

Sehr zufrieden 1 | 2 | 3 | 4 | 5 | 6 Unzufrieden

Literatur

- [Ass07] The Scout Association. *100 Jahre Pfadfinder. Die offizielle Geschichte der weltweiten Bewegung.* Spurbuchverlag, Baunach, 2007.
- [Aut19] Verschiedene Autoren. Scout Wiki. Website, 2019. Online erhältlich unter <http://scout-o-wiki.de>; zuletzt abgerufen am 16. September 2019.
- [BN13] Boris Böttcher and Markus Nüttgens. Überprüfung der Gebrauchstauglichkeit von Anwendungssoftware. *HMD Praxis der Wirtschaftsinformatik*, 50(6):16–25, 2013.
- [Bor19] Johannes Borchard. Usability Report. Website, 2019. Online erhältlich unter <http://.usabilityreport.de>; zuletzt abgerufen am 16. September 2019.
- [Car18] Stuart K Card. *The psychology of human-computer interaction.* CRC Press, 2018.
- [DINa] ENISO DIN. 25000 (2014) System und Software-Engineering - Qualitätskriterien und Bewertung von System- und Softwareprodukten (SQuaRE) - Leitfaden für SQuaRE. *Deutsches Institut für Normung, Beuth, Berlin.*
- [DINb] ENISO DIN. 26800 (2011) Ergonomie–Genereller Ansatz, Prinzipien und Konzepte. *Deutsches Institut für Normung, Beuth, Berlin.*
- [DIN97] ENISO DIN. DIN EN ISO 9241-8:1998-04 Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 8: Anforderungen an Farbdarstellungen (ISO 9241-8:1997); Deutsche Fassung EN ISO 9241-8:1997. *Berlin: Beuth, 1997.*
- [DIN98] ENISO DIN. 9241-11: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten–Teil 11: Anforderungen an die Gebrauchstauglichkeit; Leitsätze (ISO 9241-11: 1998). *Berlin: Beuth, 1998.*
- [DIN02] ENISO DIN. 14915: Software-Ergonomie für Multimedia-Benutzungsschnittstellen–Teil 1-3. *Berlin: Beuth, 2003, 2002.*
- [DIN08] ENISO DIN. 9241-11: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten–Teil 110: Grundsätze der Dialoggestaltung; Leitsätze (ISO 9241-110: 2006). *Berlin: Beuth, 2008.*

- [DIS09] ISO DIS. 9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems. *International Standardization Organization (ISO). Switzerland*, 2009.
- [FT00] Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Doctoral dissertation, 2000.
- [Gro19] Nielsen Norman Group. NN/g Nielsen Norman Group. Website, 2019. Online erhältlich unter <https://nngroup.com/>; zuletzt abgerufen am 16. September 2019.
- [Ham06] Nicolas Alexander Hamacher. *Automatische kriterienorientierte Bewertung der Gebrauchstauglichkeit interaktiver Systeme*. Hut, 2006.
- [JABH15] Norleyza Jailani, Zuraidah Abdullah, Marini Abu Bakar, and Harni Rohaida Haron. Usability guidelines for developing mobile application in the construction industry. In *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 411–416. IEEE, 2015.
- [Mit19] DPSG Mitglieder. DPSG Bundes-Website. Website, 2019. Online erhältlich unter <http://dpsg.de>; zuletzt abgerufen am 16. September 2019.
- [Nie94] Jakob Nielsen. *Usability engineering*. Elsevier, 1994.
- [Ram19] Dipl. Inf. (FH) Hanjörg Rampl. Handbuch Usability. Website, 2019. Online erhältlich unter <http://handbuch-usability.de>; zuletzt abgerufen am 16. September 2019.
- [Rau92] GWM Rauterberg. Lässt sich die Gebrauchstauglichkeit interaktiver Software messen? *Ergonomie & Informatik*, 16:3–18, 1992.
- [Red] Marc Redepenning. Institut für Geographie Lehrstuhl für Geographie I (Kulturgeographie) Prof. Dr. Marc Redepenning.
- [Ric97] Michael Richter. Kriterien der Benutzerfreundlichkeit. *Philosophische Fakultät der Universität Zürich*, http://www.michaelrichter.ch/literat_97.pdf, 1997.
- [Rut02] Marcia A Ruthford. Mix and match usability methods: picking the pieces for our project. In *Proceedings. IEEE International Professional Communication Conference*, pages 343–351. IEEE, 2002.

- [Sch19] Ulf Schubert. User Experience Blog. Website, 2019. Online erhältlich unter <https://www.user-experience-blog.de/2014/11/anwender-befragen-tipps-und-tricks-fuer-das-interview-im-usability-test/>; zuletzt abgerufen am 16. September 2019.
- [SR91] Brian Shackel and Simon J Richardson. *Human factors for informatics usability*. Cambridge university press, 1991.
- [Ten19] F. Tenzer. Marktanteile von Android und iOS am Absatz von Smartphones in Deutschland von Januar 2012 bis März 2019. Website, 2019. Online erhältlich unter <https://de.statista.com/statistik/daten/studie/256790/umfrage/marktanteile-von-android-und-ios-am-smartphone-absatz-in-deutschland/>; zuletzt abgerufen am 16. September 2019.
- [Usa19] KM Usability. Steckbrief-Sammlung: Usability Engineering-Methoden. Website, 2019. Online erhältlich unter <https://kmu-usability.de>; zuletzt abgerufen am 16. September 2019.
- [VAK⁺08] Maiju Vuolle, Anne Aula, Minna Kulju, Teija Vainio, and Heli Wigelius. Identifying usability and productivity dimensions for measuring the success of mobile business services. *Advances in Human-Computer Interaction*, 2008, 2008.