

Titel des Dokuments: Anhang Entwicklungsbericht “Digitalisierung”

Anforderungsmanagement für Softwareentwicklung – Anforderungserhebung & Projektmanagement für die Digitalisierung auf kommunaler Ebene

Hinweis auf die Nummer des B4R-Arbeitspaketes: WP3.1

Gliederung

1. Einleitung – Problembeschreibung
2. Partizipation & Design – verstehen was wirklich gebraucht wird
 - 2.1. Methoden für die erste Bedarfserfassung
 - 2.1.1. Einzelinterview
 - 2.1.2. Fokusgruppen
 - 2.1.3. User Story Mapping
 - 2.2. Methoden zur Visualisierung
 - 2.2.1. Personas
 - 2.2.2. Wireframes
 - 2.2.3. Mockups
 - 2.3 Prototyping mit Klickdummies
 - 2.4 Erstellen des Lastenheftes
3. Arbeits- und Projektmanagement
 - 3.1. Projektorganisation mit Scrum
 - 3.1.1. Sprints
 - 3.1.2. Rollen
 - 3.1.3. Artefakte
 - 3.1.4. Ereignisse (Events)
 - 3.1.5. Scrum als Vorgehensweise in der Regionalentwicklung
 - 3.2. Arbeitsorganisation mit Kanban
4. Fazit

Autor: Alexander Mäule (FH Kiel)

Institution: FH Kiel

Projekt: Benefit4Regions

Datum: 13.03.2019

Fördermittelhinweis: Dieses Projekt wird gefördert mit Mitteln des Europäischen Fonds für regionale Entwicklung. | Dette projekt finansieres af midler fra Den Europæiske Fond for Regionaludvikling

Website: www.benefit4region.eu

Anzahl der Zeichen inkl. Leerzeichen (ganzes Dokument): 35.591

1. Einleitung – Problembeschreibung

Die fortschreitende Digitalisierung durchdringt mit zunehmender Geschwindigkeit nahezu alle Arbeits- und Lebensbereiche. Diesen technologischen Fortschritt gilt es zu nutzen um Prozesse und Arbeitsaufgaben zu vereinfachen, zu beschleunigen die damit verbundenen Effizienzgewinne in sinnvollere Tätigkeiten zu investieren. Bisher ist dieser technologische Fortschritt hauptsächlich von Unternehmen genutzt worden, um damit die Werthaltigkeit für den Kunden zu steigern. Im öffentlichen Sektor in Deutschland geht viel Potential durch veraltete, langsame und oft noch manuelle Prozesse verloren. Dänemark ist insbesondere im Rahmen von E-Government wesentlich weiter.

Behördengänge wie die Ummeldung des Wohnorts machen es notwendig, dass Bürger mit händisch ausgefüllten Formularen zu den regulären Arbeitszeiten in den jeweiligen Verwaltungen vorstellig werden müssen, um dort anschließend auf die manuelle Bearbeitung durch einen Sachbearbeiter zu warten. Diese Situation steht exemplarisch für die Möglichkeiten wie Digitalisierung dazu beitragen könnte Zeit, Nerven und Ressourcen zu sparen, in wichtigere Aufgaben zu investieren und den drohenden Fachkräftemangel im öffentlichen Sektor abzufedern. Darüber hinaus gibt es außerhalb von E-Government diverse öffentliche Bereiche, in denen die Digitalisierung immer wichtiger wird, wie z.B. bei der lokalen Kommunikation, lokalen Bereitstellung von Informationen und der Mobilität.

Jedoch gibt es bei der Digitalisierung von solchen Prozessen oder der Entwicklung neuer Softwareanwendungen einige Fallstricke, die es zu beachten gilt. Ein häufig auftretendes Phänomen in diesem Bereich ist beispielsweise, dass an dem tatsächlichen Bedarf der beteiligten Stakeholder/ Nutzern wie den Fachkräften und den BürgerInnen vorbei entwickelt wird.

Dieses Dokument stellt einen Leitfaden dar, wie die zielgenaue und bedarfsgerechte Entwicklung von digitalen Lösungen in Kommunen in Deutschland und Dänemark vorangetrieben werden kann. Dabei werden moderne Methoden der Softwareentwicklung und des Projektmanagements vorgestellt und erklärt.

2. Partizipation & Design – verstehen was wirklich gebraucht wird

Der mit Abstand wichtigste Schritt bei der Umsetzung von Softwareprojekten ist es ein möglichst genaues Verständnis dessen zu erlangen, was tatsächlich von den Nutzern gebraucht wird. In der Praxis der Softwareentwicklung hat sich dabei ein schrittweises Vorgehen mit mehreren Partizipationsschritten als besonders effektiv erwiesen. Dieses Vorgehen umfasst alle Schritte von der ersten Bedarfsanalyse in Fokusgruppen und Interviews, der Erstellung von einfachen visuellen Vorlagen bis hin zu der Umsetzung der Software als Klickdummy. Am Ende dieses Prozesses besteht ein gemeinsamer Konsens über die genaue Ausgestaltung und Anforderungen an die zu entwickelnde Lösung, die in Form eines Lastenhefts festgehalten werden kann. Die folgende Darstellung stellt einen solchen Prozess beispielhaft dar. Die einzelnen Schritte werden anschließend erläutert.

Partizipationsschritte aus Sicht der Softwareentwicklung

(Quelle: www.familienapp.de (20.03.2019))



2.1. Methoden für die erste Bedarfserfassung

Ziel der Bedarfserhebung ist es ein erstes Verständnis für Erwartungen und Wünsche der Stakeholder zu erlangen. Zudem können besonders wichtige (kritische) Stakeholder frühzeitig über das Projekt informiert und in die Umsetzung eingebunden werden. Diese Phase der Bedarfserhebung spielt deshalb eine besonders wichtige Rolle, um einen möglichst breiten Konsens herzustellen. Beispiele für Stakeholder im kommunalen Umfeld die bei der Bedarfserhebung berücksichtigt werden sollten sind Vorgesetzte/Entscheider, Fachkräfte (Experten/Sachbearbeiter), diverse Fachabteilungen sowie die Endanwender – typischerweise die BürgerInnen, die eine zu entwickelnde Anwendung am Ende benutzen sollen. Je nach Zielgruppe gibt es dabei verschiedene Vorgehensweisen die sich bewährt haben:

2.1.1 Einzelinterviews

In Einzelinterviews werden einzelne Beteiligte anhand eines Interviewleitfadens befragt. Einzelinterviews richten sich insbesondere an Experten wie beispielsweise Fachkräfte, Berater oder andere Stakeholder. Typische Inhalte eines Einzelinterviews sind eine erste Information über das geplante Projekt und dessen Inhalte sowie Fragen nach Prozessen, Erfahrungen, Wünschen und Erwartungen an das Projekt. Die dabei gewonnenen Erkenntnisse werden anschließend dokumentiert und mit anderen Stakeholdern ggf. gegengeprüft.

2.1.2 Fokusgruppen

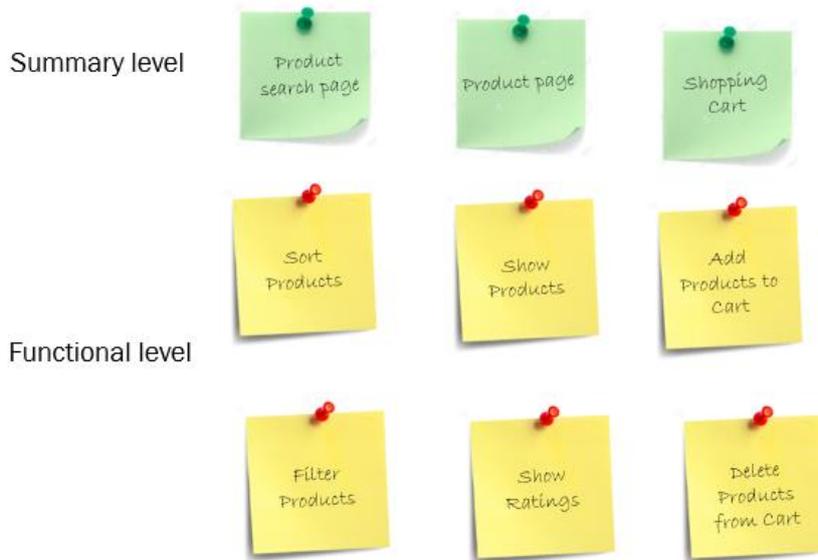
Fokusgruppen sind eine besonders effektive Methode um Ideen zu sammeln, zu diskutieren und zu reflektieren. Idealerweise sollte eine Fokusgruppe dabei aus 4-8 Teilnehmern bestehen um eine möglichst gute Moderation zu gewährleisten. Analog zu den Einzelinterviews ist es ebenfalls sinnvoll für Fokusgruppen einen Leitfaden mit den wichtigsten Fragen vorzubereiten. Prinzipiell können Fokusgruppen sowohl mit Fachkräften als auch mit BürgerInnen durchgeführt werden. Je nach Art des Projektes empfiehlt es sich allerdings die unterschiedlichen Gruppen von Stakeholdern zu trennen. Das ist insbesondere dann der Fall, wenn die Anwesenheit von bestimmten Stakeholdern z.B. Fachkräften / Experten sich negativ auf die Bereitschaft zur Teilnahme an Diskussionen auswirken kann. Erfahrungsgemäß eignen sich Fokusgruppen besonders gut, wenn bereits erste Ergebnisse auf Einzelinterviews vorliegen. Diese Erfahrungen können als Inspiration für neue Ideen in den Leitfaden aufgenommen werden. Die Vorschläge und Ideen sollten dabei idealerweise z.B. mit Hilfe von Metaplanwänden visualisiert und ggf. konkretisiert werden.

2.1.3 User Story Mapping

User Story Mapping ist eine Methode die aus dem modernen, agilen Softwareprojektmanagement stammt. Ziel dieser Methode ist es sogenannte „User-Task“ also einzelne Aufgaben, die der Benutzer in einer zu erstellenden Software ausführen können soll, zu identifizieren und einzuordnen. Klassischerweise werden dazu verschiedenfarbige Post-Its und eine möglichst große freie Wand verwendet. Das User-Story Mapping kann dabei sowohl in Fokusgruppen als auch im Anschluss an die Bedarfserhebung durchgeführt werden. Wichtig für die Durchführung eines User Story Mappings ist ein möglichst gutes Verständnis dessen was Bestandteil des umzusetzenden Produktes ist.

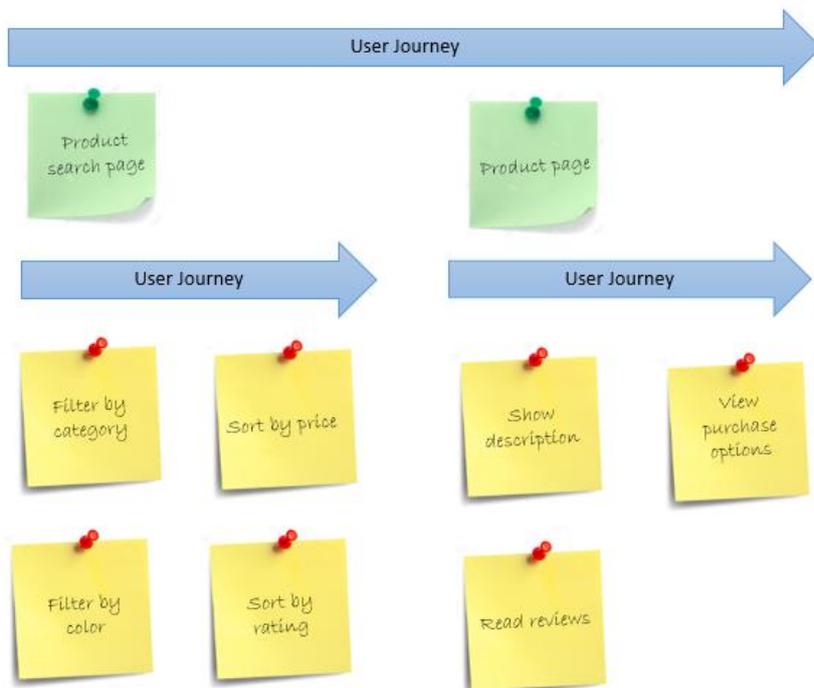
Im ersten Schritt werden die gesammelten Ideen und Anforderungen in User Tasks überführt. Dabei bietet es sich an im Vorwege eine erste grobe Kategorisierung z.B. nach Ansichten (siehe Abbildung nächste Seite) vorzunehmen.

User Story Mapping mit User Task (gelb) sortiert nach Ansichten (grün)
 (Quelle: www.familienapp.de)



Im zweiten Schritt werden die einzelnen Kategorien und die dazugehörigen User-Tasks in einem sogenannten User-Journey angeordnet. Dieser User-Journey bildet eine mögliche zeitliche/prozesshafte Abfolge in der Verwendung der zu entwickelnden Software dar.

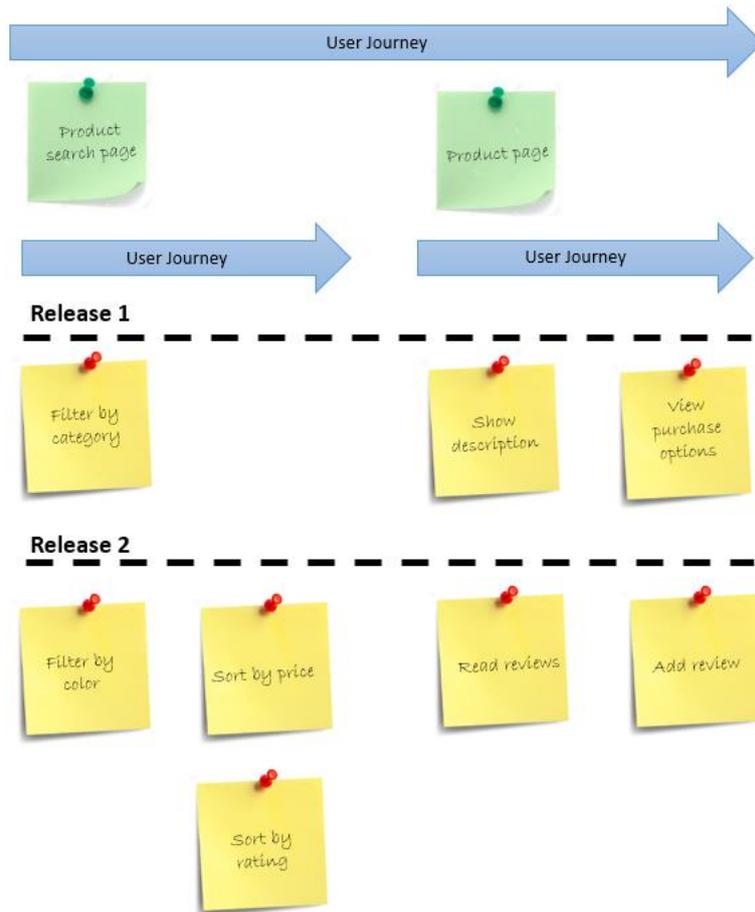
User Story Mapping mit möglicher zeitlicher Abfolge (User Journey) einzelner Tasks
 (Quelle: www.familienapp.de (20.03.2019))



Im dritten und letzten Schritt können die einzelnen User Tasks innerhalb ihrer Kategorien nochmals geordnet werden. Klassischerweise wird hier in sogenannten Sprints oder Releases geplant (mehr dazu im Kapitel über Scrum) möglich ist es aber auch eine Sortierung nach Benutzergruppen (Eltern, Kinder, Teenager) oder nach Wichtigkeit der Usertasks.

User Story Mapping mit Sortierung nach Release Version

(Quelle: www.familienapp.de (20.09.2019))



Die User-Story Map bildet damit am Ende eine Art Landkarte, die einen strukturierten Überblick über die gewünschte Funktionalität bildet. Diese User Story Map wird im Laufe des Beteiligungsprozesses laufend angepasst und erweitert. Die User Story Map bildet die Grundlage für die umzusetzende Funktionalität und die folgenden Partizipationsschritte.

2.2. Methoden der Visualisierung

Die Visualisierung der Ergebnisse aus den Fokusgruppen und Einzelinterviews ist essentiell um die oft noch sehr vagen Vorstellungen der Stakeholder zu konkretisieren. Ziel dieser Maßnahme ist es, einen ersten groben Überblick über die verschiedenen Benutzer und deren Erwartungen zu erhalten. Außerdem wird eine erste Darstellung der Software mit den darin enthaltenen Funktionen entwickelt. Diese erste grobe Darstellung der zu entwickelnden Software gilt es in Einzelinterviews mit den Erwartungen und Vorstellungen einzelner Personen aus den jeweiligen Zielgruppen abzugleichen. Dabei wird ein besonderes Augenmerk darauf gelegt, dass die Anforderungen dieser Zielgruppen korrekt verstanden wurden und wie Funktionalität und Design weiter verbessert werden können. Drei Methoden für die Visualisierung werden im Folgenden vorgestellt:

2.2.1 Personas

Diese Methode kommt nach der Durchführung der Einzelinterviews und der Fokusgruppen zum Einsatz. Dazu werden zunächst die verschiedenen Nutzergruppen identifiziert. Anschließend werden fiktive Profile einzelner Benutzer, die repräsentativ für diese Nutzergruppen stehen, erstellt. Diese fiktiven Profile sind die eigentlichen Personas. Ziel dieser Personas ist es einen besseren Überblick über die verschiedenen Nutzergruppen, deren Wünsche und Ansprüche zu entwickeln und sich in

diese Personas hineinversetzen zu können. Zu diesem Zweck werden Personas mit einem Namen, einem Gesicht, einer Funktion, einer kurzen Historie, Vorlieben, und Erwartungen versehen. Die dabei erfassten Informationen können je nach Art des Projektes unterschiedlich sein und richten sich nach der Nützlichkeit dieser Informationen. Wichtig ist es keine wirkliche Person aus einer der Nutzergruppen zu beschreiben, sondern diese Werte anhand des durchschnittlichen Nutzers zu charakterisieren. Ein typisches Beispiel in Deutschland für eine solche fiktive Persona ist „Max Mustermann“. Je nach Zielgruppen der zu entwickelnden Software werden verschiedene Personas beispielsweise für Fachkräfte, BürgerInnen und externe Experten entwickelt. Dabei kann es in jeder Nutzergruppe mehrere Personas geben um geschlechterspezifische, altersspezifische oder situationsspezifische Erwartungen einzelner Untergruppen abzubilden und zu berücksichtigen. In der folgenden Abbildung ist eine solche Persona, die in einem Projekt mit der Landeshauptstadt Kiel verwendet wurde, abgebildet.

Persona aus der Entwicklung des Kiel-Kompass die eine jugendliche, weibliche Neuzugewanderte darstellt. (Quelle: www.familienapp.de (20.03.2019))

Nutzerin: Rehan
Xtensio



"Ich möchte in meiner Freizeit mehr mit meiner Familie machen"

Alter: 16
Familie: Eltern und 3 Geschwister
Herkunft: Irak
Sprache: Kurdisch
Charakter: Verantwortungsbewusste große Schwester

Persönlichkeit

Introvertiert	Extrovertiert
Denken	Fühlen
Abhängig	Selbstständig
Abwägen	Intuition

Hintergrund

Rehan ist mit ihren Eltern und drei Geschwistern vor 4 Jahren nach Deutschland gekommen. Sie geht zur Schule und kann schon sehr gut deutsch. Zuhause kümmert sie sich oft um ihre jüngeren Geschwister, würde aber gern mehr unternehmen oder ein neues Hobby anfangen. Ihren Eltern hilft sie dabei, ihre Sprachkenntnisse zu verbessern. In ihrer Freizeit trifft Rehan sich gern mit Freundinnen, schickt Bilder über Snapchat oder ist bei Instagram. In der Schule hat sie von der App erfahren und hofft, dort Inspirationen für ein neues Hobby zu finden. Es sollte zeitlich mit der Schule vereinbar sein und nicht zu weit weg, da sie nicht so weit fahren möchte. Für ihre kleine Schwester schaut sie dann auch nach Möglichkeiten, Nachhilfe zu bekommen.

Motivation zur Nutzung der App

Freizeitgestaltung	<div style="width: 80%; background-color: #f00; height: 10px;"></div>
Bildung/Arbeit	<div style="width: 60%; background-color: #f00; height: 10px;"></div>
Sprache	<div style="width: 30%; background-color: #f00; height: 10px;"></div>
Unabhängigkeit	<div style="width: 10%; background-color: #f00; height: 10px;"></div>
Unterstützung finden	<div style="width: 40%; background-color: #f00; height: 10px;"></div>
Kinder	<div style="width: 70%; background-color: #f00; height: 10px;"></div>
Integration	<div style="width: 50%; background-color: #f00; height: 10px;"></div>

Ziele

- Ein neues Hobby finden
- Neue Freunde finden
- Den kleinen Geschwistern helfen
- Die Eltern unterstützen beim Deutsch lernen

Erwartungen

- Angebote speziell für Kinder/ Jugendliche
- Inspirationen für Freizeitaktivitäten finden
- Moderne App, die Spaß macht
- Nicht lange suchen müssen, um etwas Passendes zu finden

Frustration

- Werbung
- Wenn wichtige Informationen versteckt sind oder die App/ Webseite unübersichtlich ist
- Wenn die Seite zu lange lädt
- Wenn es nur "langweilige" Angebote für Erwachsene gibt

Mithilfe von Personas kann eine nutzerorientierte Entwicklung von Software betrieben werden, um damit möglichst viele Erwartungen der Nutzergruppen zu erfassen und abzubilden. Über die Softwareentwicklung hinaus werden die erstellten Personas auch dazu verwendet die fertige Software in den jeweiligen Zielgruppen mit zielgruppenspezifischen Werbekampagnen zu bewerben.

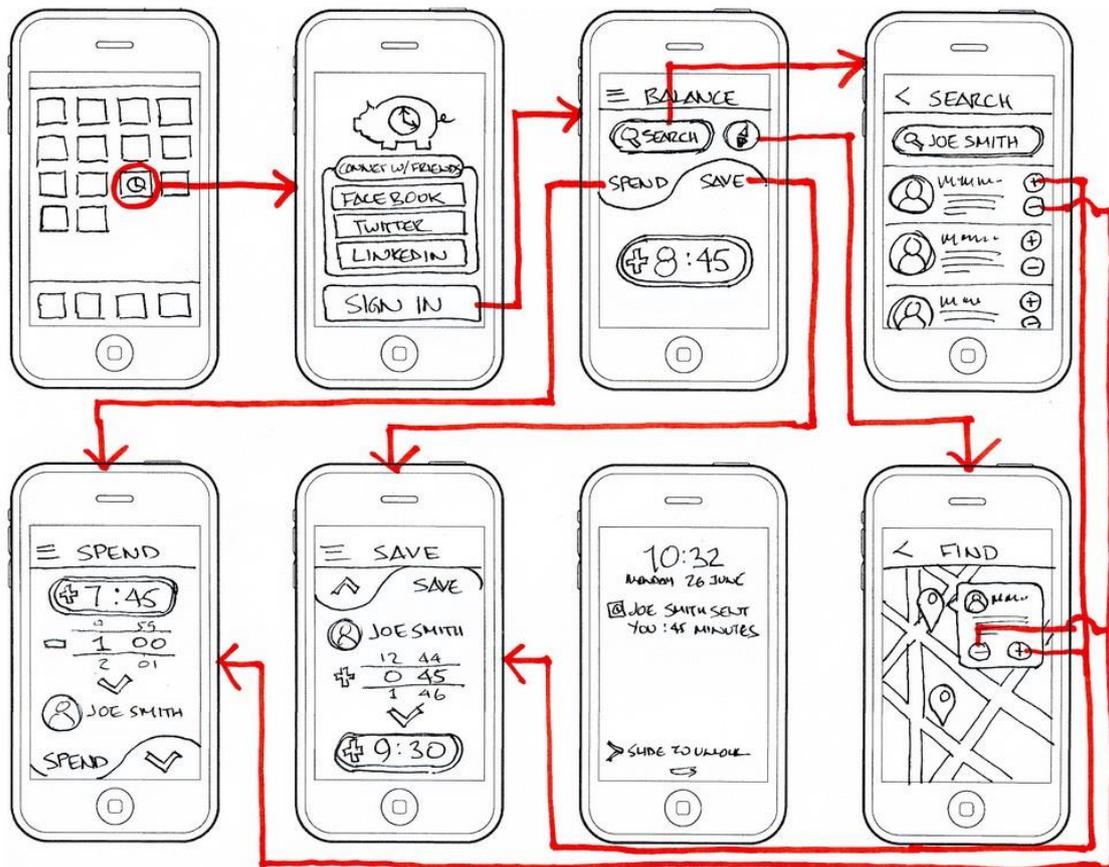
2.2.1 Wireframes

Hinter einem Wireframe verbirgt sich eine erste, grobe Skizze mit der eine Software konzeptionell dargestellt werden soll. Bei Wireframes geht es nicht um das eigentliche Design der Software, sondern um die Struktur, Inhalte, Benutzerführung und einer ersten Idee für die Benutzeroberfläche. Aufgrund des konzeptionellen Charakters von Wireframes werden einfachheitshalber häufig Skizzen auf Papier verwendet. Je nach Endgerät gibt es für das jeweilige Format im Internet auch Vordrucke zu finden, die eine visuelle Anordnung der verschiedenen Funktionen erleichtern. Dieser Prozess wird auch „Paper-Prototyping“ genannt. Ein Beispiel für ein solches Wireframing ist in der Abbildung auf

der nächsten Seite zu sehen. Alternativ gibt es für die Erstellung von Wireframes auch diverse Softwarepakete die verwendet werden können.

Beispiel für Wireframes auf Basis einer Vorlage

(Quelle: <https://blog.prototypr.io/why-you-shouldnt-skip-your-wireframing-1f7a70d5c125> (20.03.2019))

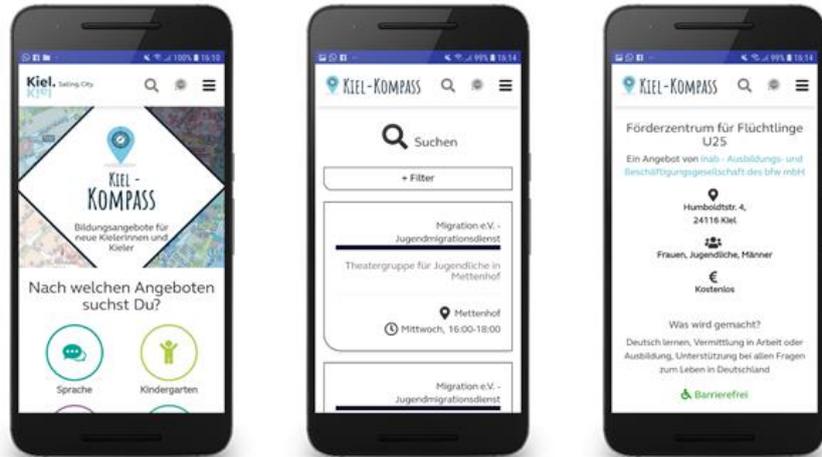


Mithilfe dieser Wireframes wird anschließend Feedback über die geplante Funktionalität und Benutzerführung gesammelt. Dazu werden wiederum Einzelinterviews mit Repräsentanten aus den verschiedenen Benutzergruppen gesammelt und die Wireframes entsprechend überarbeitet.

2.2.2 Mockups

Mockups stellen eine Weiterentwicklung von Wireframes dar. Die Erstellung von Mockups ist dementsprechend anspruchsvoller und beinhaltet neben den Ergebnissen der Benutzertests mit Wireframes auch eine detaillierte Darstellung der Benutzeroberflächen mit entsprechender Farbgestaltung, Typographie und einem Design. Häufig empfiehlt es sich zu diesem Schritt eine Designerin oder einen Designer hinzuzuziehen.

Mockups für den Kiel-Kompass nach mehrmaliger Korrektur mit dem fertigen Design und Benutzerführung (Quelle: www.familienapp.de (20.03.2019))



Wie bereits bei den Wireframes ist es notwendig das Design, die Benutzerführung und die verschiedenen Funktionen mit Repräsentanten der Zielgruppen auf die einfache Benutzbarkeit hin (bevorzugt in Einzelinterviews) zu testen.

2.3. Prototyping mit Klickdummies

Aufbauend auf den ersten Mockups kann die Erstellung eines Klickdummy erheblich zur Verbesserung der Benutzbarkeit (Usability) einer Software beitragen. Im Gegensatz zu Mockups und Wireframes handelt es sich bei einem Klickdummy um eine teilweise interaktionsfähige Demo die einzelnen Interaktionen der fertigen Software simuliert. Der Klickdummy kommt dabei in der Gestaltung und der Benutzerführung der zu erstellenden Software schon sehr nahe. Allerdings stellt ein Klickdummy nur eine Simulation der Software dar und verfügt noch nicht über die tatsächliche Funktionalität. Für die Erstellung eines Klickdummy wird keinerlei Erfahrung in der Programmierung benötigt, da es sich lediglich um Grafiken handelt. Die folgende Abbildung zeigt den Klickdummy für den „Kiel-Kompass“ der mit der Software von proto.io erstellt wurde:

Klickdummy für die Betrachtung auf einem Smartphon. Mit den türkisen umrandeten Elementen kann interagiert werden – die anderen Elemente sind statisch

(Quelle: www.familienapp.de (20.03.2019))



Der Klickdummy in diesem Beispiel ist für Smartphones optimiert und wurde mit einzelnen Personen aus den Nutzergruppen getestet. Der Vorteil von Klickdummies ist, dass diese unmittelbar auf den dafür vorgesehenen Endgeräten ausgeführt werden können. Damit können die Testpersonen die Funktionsweise unmittelbar und auf eigenen Geräten testen und nachvollziehen. Die dabei gewonnenen Erfahrungen werden nochmals dazu genutzt die Funktionalität, das Design und die Benutzerführung zu verbessern und anzupassen.

2.4. Erstellen des Lastenheftes

Nachdem alle vorherigen Schritte von der ersten Präzisierung mit Einzelinterviews und Fokusgruppen bis hin zum Klickdummy abgeschlossen sind, geht es an die eigentliche Erstellung der Software. Als Grundlage für die Durchführung der Softwareentwicklung und für die damit verbundene Ausschreibung wird ein sogenanntes Lastenheft erstellt. Das Lastenheft beschreibt dabei die gesamte Funktionalität, die von der zu entwickelnden Software erfüllt werden soll. Damit stellt das Lastenheft einen Anforderungskatalog dar, der als Grundlage für die eigentliche Umsetzung dient. Diese Anforderungen werden unmittelbar aus dem vorangegangenen Partizipationsprozess abgeleitet. Durch diesen Partizipationsprozess und intensiven Beteiligung von Nutzern kann sichergestellt werden, dass eine genaue Vorstellung existiert was letztendlich Entwickelt werden soll. Grundsätzlich gilt: je genauer die Anforderungen bereits feststehen und durch eine Partizipationsphase getestet sind, desto höher ist die Nützlichkeit und damit die Akzeptanz des fertigen Produkts. Eine beispielhafte Gliederung (In Anlehnung an: Balzert, Helmut; Lehrbuch der Software-Technik; 1998) für ein Lastenheft besteht aus den folgenden Punkten:

- 1. Zielbestimmung:** Es wird beschrieben, welche Ziele durch den Einsatz des Produkts erreicht werden sollen.
- 2. Produkteinsatz:** Es wird festgelegt, für welche Anforderungsbereiche und für welche Zielgruppen das Produkt vorgesehen ist. Sinnvoll ist es hier die erstellten Personas zu verwenden.
- 3. Produktfunktionen:** Hier werden die Funktionen die sich aus den Einzelinterviews und Fokusgruppen ergeben haben detailliert aufgelistet und durch Beispiele, Wireframes und Mockups verdeutlicht.
- 4. Produktleistungen:** Betrifft Anforderungen bezüglich Zeit, Datenumfang oder Genauigkeit in Bezug auf einzelne Hauptfunktionen und Hauptdaten.
- 5. Qualitätsanforderungen:** Die wichtigsten Qualitätsanforderungen sollten aufgeführt werden. Dazu gehören Zuverlässigkeit, Benutzbarkeit und Effizienz.
- 6. Ergänzungen:** Hier werden Ergänzungen oder spezielle Anforderungen beschrieben, z.B. außergewöhnliche Anforderungen an die Benutzerschnittstelle.

3. Arbeits- & Projektmanagement

Im Bereich des Arbeits- und Projektmanagements haben sich im Bereich der Softwareentwicklung in den letzten Jahren vermehrt sogenannte agile Arbeits- und Vorgehensweisen durchgesetzt. Diese Vorgehensweisen verfolgen ein Prinzip, dass in den letzten 15 Jahren auch vermehrt im Fokus der Regionalentwicklung in Deutschland und Dänemark steht und wächst: die direkte partizipative Einbindung von BürgerInnen in Projekte. Der Begriff des agilen Projektmanagements hat sich in den 2000er Jahren gebildet nachdem immer häufiger große und komplexe IT-Projekte spektakulär gescheitert sind. Bei der Analyse der Ursachen für das Scheitern dieser Projekte wurde immer wieder ein Aspekt als Hauptursache identifiziert – mangelnde Kommunikation und Austausch zwischen Kunden, Softwareentwicklern, Management und Auftraggebern. Um dieser Entwicklung entgegenzuwirken hat sich im Jahr 2001 eine Gruppe aus Softwareentwicklern zusammengeschlossen und das „**Agile Software Development Manifesto**“ entwickelt. Dieses

Manifest sollte primär als Grundlage für eine bessere Vorgehensweise bei der Durchführung von Projekten dienen. Dieses Manifest umfasst primär die folgenden Prinzipien:

- **Individuen und Interaktionen** haben Vorrang vor Prozessen und Werkzeugen.
- **Funktionsfähige Produkte** haben Vorrang vor ausgedehnter Dokumentation.
- **Zusammenarbeit mit dem Kunden** hat Vorrang vor Vertragsverhandlungen.
- Das **Eingehen auf Änderungen** hat Vorrang vor strikter Planverfolgung.

Zentraler Bestandteil des agilen Softwaremanagements ist damit die Fokussierung auf Interaktion/Partizipation und auf tatsächliche Arbeitsergebnisse. Diese Prinzipien stehen damit diametral/entgegengesetzt zu den Vorgehensweisen im klassischen Projektmanagement bei dem die Vorgehensweise und die Dokumentation im Zentrum der Anstrengungen liegen. Zu den Urhebern dieses agile Manifestos gehören unter anderem Ken Schwaber, Mike Beedle und Jeff Sutherland und damit die Begründer des modernen Projektmanagements mit Scrum.

3.1. Projektorganisation mit Scrum

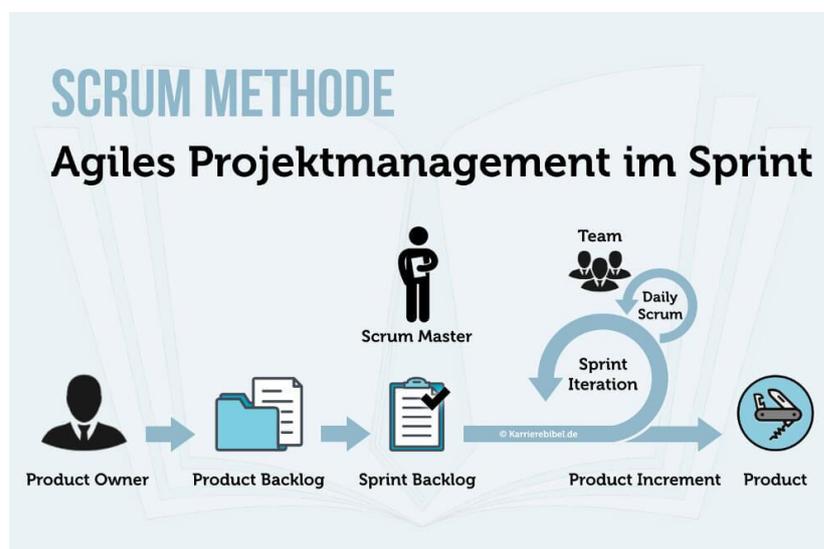
Dementsprechend versteht sich Scrum als ein Gegenentwurf zu klassischen, hierarchischen Organisationen. Scrum setzt stark auf den Informations-, Erfahrungs- und Wissensaustausch zwischen den an Projekten beteiligten Personen. Um dies zu erreichen gibt es innerhalb der Scrum Methode unterschiedliche Rollen, Ereignisse, Artefakte und Anforderungen an die Durchführung. Die eigentliche Arbeit findet dabei in sogenannten Sprints statt.

3.1.1 Sprints

Ein Sprint ist ein Zeitraum in dem das zu erstellende Produkt inkrementell verbessert wird. Zu Beginn eines jeden Sprints werden dabei die in dem Sprint durchzuführenden Tätigkeiten festgelegt. Diese Tätigkeiten dürfen während eines Sprints nicht verändert werden. Ein Sprint endet nach einer festgelegten Zeitspanne in der die vorher definierten Tätigkeiten durchgeführt sein müssen. Diese Zeitspanne beträgt üblicherweise zwischen zwei und vier Wochen. Alle Sprints haben dabei die gleiche Länge und ein Projekt besteht aus mehreren Sprints an deren Ende jeweils ein leicht verbessertes, funktionierendes Produkt steht. Am Ende eines jeden Sprints steht eine Sprint Review oder eine Sprint Retrospektive (Evaluation und Rejustierung).

Die Scrum – Methode im Überblick

(Quelle: <https://karrierebibel.de/wp-content/uploads/2017/03/Scrum-Methode-Beispiel-Ablauf-Grafik.jpg> (20.03.2019))



3.1.2 Rollen

Scrum kennt insgesamt drei verschiedene Rollen: den **Product Owner**, **Scrum Master** und das **Entwicklungsteam**. Zusammengefasst werden diese Rollen als **Scrum Team** bezeichnet. Zusätzlich zu diesen Rollen gibt es noch die Stakeholder. Unter Stakeholdern versteht man im Allgemeinen Personen, die mit dem Projekt mittelbar oder unmittelbar in Berührung kommen. Beispiele für Stakeholder sind Kunden, Anwender oder das Management. Scrum erlaubt es diesen Stakeholdern üblicherweise regelmäßig an sogenannten Sprint Review Meetings teilzunehmen um dort die Software zu testen und Feedback zu geben. Dieses Feedback wird in der weiteren Entwicklung berücksichtigt. Damit unterscheidet sich Scrum von traditionellen Projektmanagementmethoden. Im traditionellen Projektmanagement wird die Funktionalität der entwickelten Software, Anwendern üblicherweise erst am Ende eines Projekts in einer Test- oder Einführungsphase ermöglicht. Dies verhindert in den meisten Fällen eine kostengünstige und schnelle Änderungen der Funktionalität.

Der Product Owner ist im Wesentlichen für die Eigenschaften und den wirtschaftlichen Erfolg einer zu entwickelnden Software verantwortlich. In dieser Position entscheidet der Product Owner über die Eigenschaften einer zu entwickelnden Software und deren Priorität. Zudem beurteilt der Product Owner am Ende eines jeden Sprints, ob eine Funktionalität zufriedenstellend fertig entwickelt wurde oder ob Änderungen notwendig sind. Klassischerweise ist der Product Owner dabei eine einzelne Person und kein Komitee. Der Product Owner kann dabei auch eine vom Auftraggeber ermächtigte Person sein die für die über entsprechende Entscheidungskompetenzen verfügt.

Die Hauptaufgabe des **Scrum Master** ist für die Einhaltung der Scrum-Regeln zu sorgen. Er überprüft die Einhaltung dieser Regeln und ist darüber hinaus für die Behebung von Störungen und Hindernissen zuständig, die den Projektablauf oder die Durchführung eines Sprints beeinflussen. Dazu gehören Kommunikationsprobleme, Konflikte im Entwicklerteam, Verwaltungsaufgaben oder Störungen in der Zusammenarbeit zwischen Product Owner und Entwicklerteam. Deshalb wird der Scrum Master auch als dienende Führungskraft bezeichnet. Er steht in seiner Rolle zwar dem Entwicklerteam vor – allerdings nicht disziplinarisch oder beurteilend außerdem gibt er keine direkten Arbeitsanweisungen.

Das Entwicklerteam

Das Entwicklerteam ist für die Durchführung der im Projekt zu verrichtenden Arbeiten in der vom Product Owner festgelegten Reihenfolge zuständig. Das Entwicklerteam organisiert sich dabei komplett selbst und arbeitet eigenverantwortlich an der Umsetzung der Arbeitsaufgaben, die mit dem Product Owner und dem Scrum Master gemeinsam festgelegt wurden. Das Entwicklerteam sollte dabei möglichst interdisziplinär besetzt sein (z.B. Architekt, Datenbankentwickler, Tester, Front- und Backendentwickler) um Abhängigkeiten von zusätzlichen Spezialisten zu vermeiden. Ein ideales Scrum Team ist deshalb sowohl mit Spezialisten als auch Generalisten besetzt. Von den Mitgliedern eines Entwicklerteams wird gegenseitige Hilfe erwartet – Arbeitsaufgaben und deren Erreichung liegen in der Gesamtverantwortung des gesamten Teams! Ein ideales Entwicklerteam besteht dabei aus drei und höchstens neun Mitgliedern, um den Koordinierungsaufwand so gering wie möglich zu halten.

3.1.3 Artefakte

Artefakte beschreiben schriftlich festgehaltene Ziele und Entwicklungsschritte. Im Vergleich zum traditionellen Projektmanagement ist es jedoch nicht zwingend notwendig diese Ziele und Entwicklungsschritte in Form von langatmigen Dokumenten festzuhalten. In der Regel wird in Scrum entweder mit digitalen Plattformen wie Trello gearbeitet oder analog mit Post-Its um eine bessere Visualisierung und leichtere Erfassung zu erreichen.

Das **Product Backlog** ist eine geordnete Sammlung aller Anforderungen an das Produkt. Typischerweise ergibt sich das Product Backlog am Anfang eines jeden Projektes aus dem Lastenheft des Kunden. Im Gegensatz zu einem klassischen Lastenheft wird das Product Backlog dynamisch

und konstant weiterentwickelt und konkretisiert, um Änderungen der Kundenwünsche Rechnung tragen zu können. Dabei übernimmt der Product Owner die Verantwortung für die Pflege, Reihenfolge und Priorisierung der Einträge im Backlog. Die Anforderungen im Backlog sind dabei nicht primär technischer, sondern fachlicher oder anwenderorientierter Natur. Um Anforderungen möglichst einfach, klar und verständlich zu formulieren, werden User Stories verwendet.

Zusätzlich dazu wird das Product Backlog in regelmäßigen Abständen weiterentwickelt. Dies geschieht im **Product Backlog Refinement**. Dabei arbeitet der Product Owner gemeinsam mit dem Entwicklerteam an der Überarbeitung des Backlogs.

Typische Aufgaben sind:

- Ordnen der Einträge
- Löschen von Einträgen, die nicht mehr wichtig sind
- Hinzufügen von neuen Einträgen
- Detaillieren von Einträgen
- Zusammenfassen von Einträgen
- Schätzen von Einträgen
- Planung von Releases

Dabei ist es ausdrücklich erwünscht, dass Stakeholder weitere Informationen beitragen und beispielsweise erklären wie einzelne Funktionalitäten funktionieren sollen. Deshalb nehmen an Product-Backlog-Treffen auch häufig neben dem Product Owner und dem Entwicklerteam ausgewählte Stakeholder teil.

Zusätzlich zum Product Backlog, der alle Anforderungen an das gesamte Produkt enthält, gibt es noch einen **Sprint Backlog**. Im Sprint Backlog werden alle Aufgaben festgehalten, die in einem Sprint zu erledigen sind. Dabei stellt ein Sprint eine bestimmte, mit dem Product Owner und dem Entwicklerteam definierte, Zeitspanne dar. Diese Zeitspanne ist dabei stets gleich lang – üblicherweise zwischen zwei und vier Wochen. Die Aufgaben, die im Sprint Backlog festgehalten werden, stammen dabei aus dem Product Backlog. Am Ende eines jeden Sprints ergeben die umgesetzten Anforderungen aus dem Sprint Backlog ein sogenanntes Product Increment.

Das **Product Increment** bezeichnet die Summe aller in einem Sprint umgesetzten Backlog Einträge. Am Ende eines jeden Sprints steht dabei zwingend ein neues Inkrement des Produktes in einem nutzbaren Zustand.

3.1.3 Ereignisse (Events)

Scrum hat auch für das abhalten von Meetings einen eigenen Begriff entwickelt: Events. Diese Abgrenzung findet dabei bewusst statt um klarzustellen, dass es sich um Arbeit handelt. Zu diesem Zweck werden die in Scrum stattfindenden Ereignisse auch stets in feste Zeitfenster eingeteilt um möglichst effektiv zu arbeiten. Üblicherweise finden diese Events in wiederkehrenden zeitlichen Abständen statt.

Das **Sprint Planning** findet stets zu Beginn eines neuen Sprints statt. Darin geht es primär darum festzulegen, welche Elemente aus dem Product Backlog innerhalb eines Sprints umgesetzt werden können und wie diese Tätigkeiten umgesetzt werden. Was genau in einem Sprint umgesetzt werden soll wird dabei vom Product Owner aus den Elementen im Backlog ausgewählt und priorisiert. Außerdem einigt sich der Product Owner für jede Tätigkeit mit dem Entwicklerteam über die Kriterien die darüber entscheiden ob eine Tätigkeit erfolgreich fertig gestellt wurde (*Definition of done*). Bei der Auswahl der in einem Sprint umzusetzenden Tätigkeiten entscheidet das Entwicklerteam wie viele Product-Backlog Einträge abhängig von deren Komplexität in einem Sprint umgesetzt werden können.

Zusätzlich zum Sprint Planning gibt es an jedem Arbeitstag ein Meeting innerhalb des Entwicklerteams. Dieses Meeting wird deshalb auch als **Daily Scrum** bezeichnet. Das Daily Scrum sollte dabei nicht mehr als 15 Minuten Zeit in Anspruch nehmen. Um diese Zeitspanne einzuhalten findet das Daily Scrum üblicherweise im Stehen statt. Im Daily Scrum tauschen sich die Mitglieder des

Entwicklerteams über den aktuellen Stand der Tätigkeiten aus, besprechen Hindernisse und mögliche Lösungen. Ziel ist es einen guten Überblick innerhalb eines Entwicklerteams über den Arbeitsfortschritt zu erhalten und ggf. gemeinsam an der Lösung von Problemen zu arbeiten.

Am Ende eines jeden Sprints findet ein **Sprint Review** statt. Innerhalb des Sprint Reviews werden die Ergebnisse eines Sprints evaluiert und getestet. Bei diesem Event sind deshalb die Anwesenheit des Kunden und der Anwender ausdrücklich erwünscht und vorgesehen. Kunden und Anwender sollen das fertige Inkrement testen und Feedback über die Umsetzung geben. Wenn eine Tätigkeit nicht nach den Wünschen des Kunden ausgeführt wurde, kommt diese wieder zur Vorlage im nächsten Sprint. Innerhalb eines Sprint Reviews werden oft auch neue Anforderungen von Kunden und Anwendern definiert, die vom Product Owner ggf. in das Product Backlog im nächsten Product Backlog Refinement aufgenommen werden. Durch diesen Mechanismus entsteht ein ständiges Feedback und es wird überprüft, ob eine Funktionalität tatsächlich den Anforderungen des Kunden entspricht.

Innerhalb des Entwicklerteams findet zudem noch eine **Sprint Retrospektive** statt. In der Sprint Retrospektive wird überprüft wie der Sprint ausgeführt wurde, welche Verbesserungen vorgenommen werden können und wie Probleme effektiv angegangen werden.

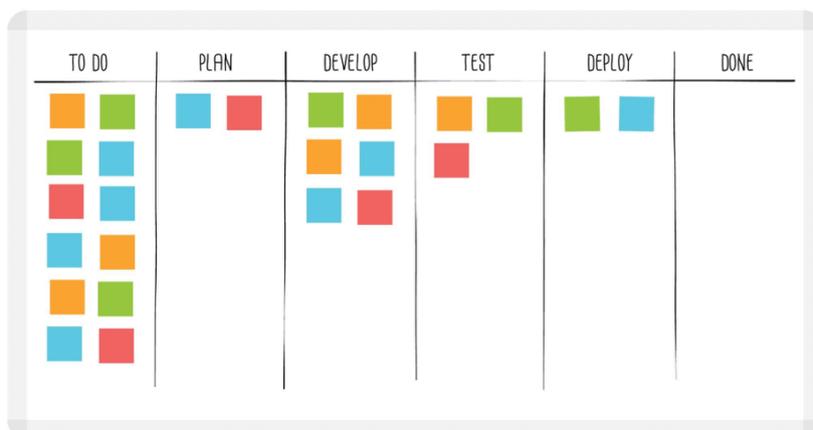
3.1.4 Scrum als Vorgehensweise in der Regionalentwicklung

Insgesamt bietet Scrum damit eine sehr kunden- und nutzerzentrierte Vorgehensweise, die durch ein inkrementelles Vorgehen und direkte Einbindung von Kunden und Anwendern passgenaue Produkte entwickelt, um den Kundennutzen zu maximieren. Dieses Vorgehen kommt dem Partizipationsgedanken in der Regionalentwicklung schon sehr nahe – stellt darüber hinaus in Abhängigkeit des Partizipationsgrades ein strukturiertes und dynamisches Vorgehensmodell dar, um BürgerInnen noch stärker an Entscheidungen und Entwicklungen auf regionaler Ebene beteiligt zu können.

3.2. Arbeitsorganisation mit Kanban

Kanban ist ähnlich wie Scrum eine Methode, die auf eine schrittweise Verbesserung setzt. Allerdings ist Kanban keine Projektmanagementmethode, sondern beschäftigt sich mit der Durchführung der eigentlichen Arbeit. Kanban wird deshalb auch als Workflow Methode bezeichnet. Kanban selbst stammt ursprünglich aus Japan und ist eng mit der Idee des Kaizens verbunden. Kaizen umfasst dabei die Philosophie der kontinuierlichen Verbesserung. Primäres Element von Kanban ist dabei das Kanban Board. Üblicherweise besteht das Kanban Board aus einem Whiteboard mit Haftnotizen oder Karteikarten. Auf jeder Haftnotiz oder Karteikarte steht dabei genau eine Arbeitsaufgabe.

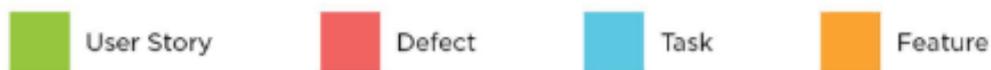
Beispiel für ein Kanban Board (Quelle: <https://hive.com/what-is-a-kanban-board/> (20.03.2019))



Dieses Kanban Board sollte dabei für alle Mitglieder eines Teams deutlich sichtbar platziert sein um den derzeitigen Arbeitsfortschritt, etwaige Engpässe und die Zuständigkeiten zu visualisieren.

Für die Arbeitsaufgaben können dabei verschiedenfarbige Karten verwendet werden um unterschiedliche Kategorien oder Prioritäten abzubilden.

Arbeitsaufgaben können verschiedene Farben erhalten um Prioritäten oder Kategorien zu Symbolisieren (Quelle: <https://hive.com/what-is-a-kanban-board/> (20.03.2019))



Die Arbeitsaufgaben selbst sollten dabei möglichst konkret formuliert werden und jeder Arbeitsaufgabe für sich innerhalb maximal eines einzelnen Arbeitstages durchführbar sein. Außerdem sollte für jede Arbeitsaufgabe eine klare Zuständigkeit vergeben werden.

Beispiel für eine mögliche Unterteilung einer Karte für eine Arbeitsaufgabe

(Quelle: <https://hive.com/what-is-a-kanban-board/> (20.03.2019))



Auf dem Kanban Board gibt es üblicherweise verschiedene Schritte. Diese Schritte sind dabei von dem Workflow abhängig der abgebildet werden soll. Für das Beispiel in der Abbildung S. 14 aus dem Bereich der Softwareentwicklung besteht dieser Workflow aus Spalten für ToDo's (oft auch als Backlog bezeichnet) Plan, Develop, Test, Deploy und Done. Kanban sieht dabei vor, dass in jeder der Spalten zu jedem Zeitpunkt nur eine vom Team vorher festgelegte Anzahl an Arbeitsaufgaben stehen darf. Diese Maßnahme dient dazu Multitasking und Engpässe zu vermeiden. Wenn ein Engpass oder eine Blockade auftritt sieht Kanban eine kritische Betrachtung dieses Engpasses vor, um daraus Maßnahmen abzuleiten um derartige Unterbrechungen zu vermeiden.

Außerdem legt Kanban fest, dass Arbeitsaufgaben jeweils von Spalte zu Spalte wandern müssen und dabei keinen Arbeitsschritt überspringen dürfen. Konkret bedeutet dies, dass eine einzelne Arbeitsaufgabe zuerst geplant, dann umgesetzt, getestet, produktiv gestellt und dann erst als tatsächlich abgeschlossen gilt. Diese Regel vermeidet damit die Abgabe von unfertigen oder ungetesteten Arbeitsaufgaben, um eine gewisse Qualität zu gewährleisten.

Kanban kommt häufig auch in Verbindung mit Scrum (auch Scrumban) zum Einsatz, um Arbeitsfortschritte innerhalb von Sprints nachzuvollziehen, sowie um die allgemeine Produktivität und Transparenz zu verbessern.

4. Fazit

Insgesamt hat die Case Study Digitalisierung viele Ähnlichkeiten zwischen deutschen und dänischen Kommunen was den Themenbereich der Digitalisierung betrifft hervorgebracht. Ein besonders wichtiges Ergebnis dieser Diskussionen und des Erfahrungsaustausches ist der Grundgedanke der Nutzerpartizipation. Die verschiedenen Ideen und Auffassungen zum Thema Bürgerpartizipation wurden dabei gemeinsam von den beteiligten Akteuren aufgenommen und mit Methoden der Softwareentwicklung verglichen. Zu diesem Zweck wurden die in diesem Dokument beschriebenen Vorgehensweisen und Methoden erläutert, anhand von Workshops und Diskussionen z.T. getestet sowie auf den Kontext der Digitalisierung in der Regionalentwicklung übertragen.

Im Rahmen der Case Study Digitalisierung konnte so ein reger Ideen- und Erfahrungsaustausch initiiert werden und die beschriebenen Partizipations- und Arbeitsmethoden abgeglichen (z.B. Interview) oder erlernt (z.B. User Story Mapping) werden. Die Vermittlung der Grundlagen für agiles Arbeits- und Projektmanagement ermöglichen es den beteiligten Projektpartnern zudem, einen Einblick zu erhalten, um Projekte im Kontext „Digitalisierung und Regionalentwicklung“ und möglicherweise darüber hinaus effektiver und effizienter zu gestalten.