

Titel 

Robert Bruckbauer

Version 2.4.1, 2022/05/29: cards+ asciidoc book

Inhalt

Vorwort	2
Hinweise des Autors	4
Über mich	4
Über cards+	4
Fallbeispiel	5
Einleitung	7
Dokumentierte Informationen	7
Digitale Transformation	8
Agile Software-Entwicklung	8
Strategisches Design	8
Taktisches Design	8
Gemeinsame Sprache	9
Qualitätspolitik	9
Jetzt hab ich's	10
1. Von der Idee zur Vision	11
2. Entscheidung für cards+	13
3. Domänensprache	15
Endlich geht es los!	17
4. Kapitel	18
So viel Neues... ..	19
5. Kapitel	20
Es läuft!	21
6. Kapitel	22
Fertig!	23
7. Kapitel	24
Alles auf Anfang	25
8. Kapitel	26
8.1. Gantt diagramm	26
8.2. Mindmap	26
8.3. Wireframe	27
tbd	28
9. Kapitel	29
tbd	30
Danksagung	31
Anhang A: Bla bla	32
Anhang B: Bla blubb	33
Stichwortverzeichnis	34
Literaturverzeichnis	35

Powered by

GitHub

Vorwort

Die Software-Industrie ist im Wandel. Die Herausforderungen durch *Cloud*, *Big Data* und das *Internet* der Dinge erfordern eine Agilität und Transparenz bei der Entwicklung von Software. Im Kern der digitalen Transformation geht es darum, dass Unternehmen und Organisationen ihre vorhandenen Prozesse und Daten digital verfügbar machen. Die Konsequenz ist, dass alle Unternehmen zu Softwareentwicklern werden. Die Entwicklung von Software ist nicht mehr beschränkt auf die IT-Branche. Bei Software wird es künftig häufiger um individuelle Lösungen statt um Produkte gehen. Geschwindigkeit ist wichtig. Kosten und Qualität bilden ein Spannungsfeld, in dem sich eine ganze Reihe von Ideen und Konzepten etabliert haben. Einige davon haben direkten Einfluss auf die Dokumentation.

Clean Code [ccd]

tbd Praktiken und Prinzipien, Spielregeln

Domain-Driven Design [ddd]

tbd gemeinsame Sprache, strategisches und taktisches Design

Agilität

tbd Manifest agiler Software-Entwicklung, Vision, Backlog, Story

Docs-as-Code [dac]

tbd AsciiDoc, Wiki

Dokumentation ist ein Thema in **jedem** Entwicklungsprojekt. Information ist das Ziel, Dokumentation ist der Weg. Abhängig von der Organisation und vom Entwicklungsprozess gibt es Pflichtenhefte, Studien, Grob- und Fachfeinkonzepte, fachliche und technische Spezifikationen, Richtlinien, Anleitungen, Entscheidungen, Anweisungen, Skizzen, Handbücher, um nur die bekanntesten Dokumenttypen zu nennen. Sie unterscheiden sich

- › im Zeitpunkt der Erstellung.
- › in der Lebensdauer.
- › in der Zielgruppe.
- › in der Art des Inhaltes.
- › in der Komplexität.

Dokumentation ist wie die Software ein fixer Bestandteil des Produktes. Ändert sich die Software, muss auch ein Teil der Dokumentation angepasst werden oder wird ungültig. In einem agilen Umfeld muss daher die Tätigkeit des Dokumentierens eine Aufgabe für das ganze Team sein. Genau wie beim Programmieren von Software gibt es beim Dokumentieren den Anspruch, Ergebnisse mit guter Qualität zu produzieren.

Ein im Jahr 2014 gestartetes agiles Entwicklungsprojekt hatte das Ziel, das IT-System der Transportleitungen der Deutschen Bahn zu modernisieren. Meine dort gewonnenen Erkenntnisse führten zum ersten Entwurf von **cards+** als Ansatz für inkrementelles Dokumentieren. Die Prinzipien und Praktiken von **cards+** halfen uns damals, Wissen langfristig und in guter Qualität zu erhalten. Mit »uns« ist dabei die gesamte Projektorganisation und die interessierten Parteien des Auftraggebers gemeint.

Wir wollten eine Produktdokumentation, auf die wir uns verlassen können. Neben dem Quelltext und den Testplänen des Produktes wurde von Anfang an auch die Dokumentation für das Produkt inkrementell erarbeitet und mit dem Fortschritt der Entwicklung kontinuierlich verbessert. Dokumente betrachteten wir wie Quelltext und versioniert sie. Wir klassifizierten sie nach folgenden Kriterien.

- › Dokumente müssen strukturiert sein.
- › Sie müssen prüfbar sein.
- › Ihre Qualität muss messbar sein.

Wir nutzen sehr viele Praktiken der Entwickler auch für die Autoren der Dokumentation: **KISS**, **DRY**, **INVEST**, die Pfadfinderregel. Die Autoren nutzten die gleichen Strategien, die Entwickler schon sehr lange anwenden, um die Komplexität von Software in den Griff zu kriegen.

- › Sie wählen Strukturen, die besser testbar sind als andere.
- › Sie modularisieren.
- › Sie delegieren.
- › Sie kapseln.

Mit **cards+** gab es ein Konzept, das solche Strategien auch für eine Produktdokumentation bietet. **cards+** machte es möglich, die Ergebnisse aus der Analyse (z.B. mit Domain-Driven Design) direkt in ein Wiki zu übernehmen.

Mit **cards+** hatten wir die »besseren Karten«.

cards+ ist ein in sich geschlossenes Ökosystem von Bausteinen. Jeder Baustein hat einen klaren Zweck, eine prüfbare Struktur und einen für den Inhalt verantwortlichen Autor. Die Bausteine sind aufeinander abgestimmt mit dem Ziel, jede Information nur einmal zu erfassen. Im Wiki ist jeder Baustein eine Seite. Bausteine bilden Hierarchien (z.B. Topic, Epic und Case) oder werden miteinander verknüpft (z.B. Begriffe aus dem Glossar, die Lösung in einem Case).

Bei **cards+** gibt es ein einfaches Kriterium, das beim Aussortieren von Information hilft. Eine Information, für die es einen passenden Baustein in **cards+** gibt, wird dauerhaft erfasst und inkrementell gepflegt. Alle andere Informationen, die zu einem bestimmten Zeitpunkt zwar wichtig und notwendig sind, später aber nicht mehr gebraucht werden, gehören zur Projektdokumentation. Dazu zählen beispielsweise User-Stories in einem Backlog, Studien oder Analysen.

Mit **cards+** werden Entwickler zu Autoren, ohne sich extra anstrengen zu müssen. Sie nutzen Formate wie **AsciiDoc** für Spezifikationen, die sich wie Code anfühlen. Sie nutzen Formate wie **JSON**, **XML** oder **YAML** für Konfigurationen, die durch Kommentare lesbar sind. Sie generieren Dokumentation direkt aus dem kommentierten Code. **cards+** als Werkzeug macht dokumentierte Informationen im Quelltext und in Testplänen für die Dokumentation im Wiki nutzbar.

Wird das Konzept **cards+** von Anfang an in der Projektorganisation gelebt, dann ist eine hochwertige Produktdokumentation keine Illusion mehr. Der Aufwand für die Herstellung der Dokumentation mit den Prinzipien und Praktiken von **cards+** unterscheidet sich nicht von dem Aufwand, Dokumente "irgendwie" zu schreiben. Aber die Chancen, die sich durch eine strukturierte Produktdokumentation in einem Wiki ergeben, sind nur durch die Kreativität der Personen beschränkt, die sie lesen und pflegen.

Hinweise des Autors

Über mich

Seit dem Jahr 1990 bin ich leidenschaftlicher Software-Entwickler. Der Schwerpunkt meiner Tätigkeiten in den Projekten ist der Entwurf und die Umsetzung komplexer IT-Systeme. Meine Rollen waren Entwickler, Projektleiter, IT-Berater und Produktverantwortlicher. Seitdem immer mehr Auftraggeber auf agile Entwicklungsprozesse setzen, hat sich für mich vieles geändert.

Über cards+

cards+ ist eine Idee, die sich aus meinen positiven und negativen Erfahrungen mit Dokumentation in agilen Entwicklungsprozessen entwickelte.

tbd Bausteine, Spielregeln im Wiki

tbd Prinzipien und Praktiken

tbd Integration mit `htmltool`



Begleitende Informationen zu cards+ gibt es auf meiner persönlichen Website [\[rb1\]](#).



Wer Fragen hat oder einfach nur Feedback zu cards+ geben möchte, kann das sehr gerne in der von mir moderierten Gruppe "Agil dokumentieren" [\[rb2\]](#) im sozialen Netzwerk Xing tun.

cards+ als Name hat sich aus der ursprünglichen Idee der Karten (engl. *cards*) von XP entwickelt. Das Plus-Zeichen steht für die Weiterentwicklung der Idee der Karten.

User stories are written on cards. The card does not contain all the information that makes up the requirement. Instead, the card has just enough text to identify the requirement, and to remind everyone what the story is. The card is a token representing the requirement. It's used in planning. Notes are written on it, reflecting priority and cost. It's often handed to the programmers when the story is scheduled to be implemented, and given back to the customer when the story is complete.

— Ron Jeffries

Die Karte ist eine Metapher für eine dokumentierte Information. Jede Karte hat einen Wert. Jede Karte hat eine Bedeutung. Die Verwendung einer Karte unterliegt bekannten Spielregeln.

Das Kartenhaus ist eine Metapher für die Qualität. Es ist eine wesentliche Eigenschaft von cards+, dass die Bausteine wie die Karten eines Kartenhauses (engl. *house of cards*) aufeinander aufbauen. Jede Karte muss in einer bestimmten Reihenfolge in einer festgelegten Art und Weise aufgestellt werden, um am Ende ein vollständiges Kartenhaus zu bilden. Es ist nur dann stabil und vollständig, wenn alle Karten vorhanden sind. Fehlt eine Karte, dann stürzt das Kartenhaus in sich zusammen. Die Karten des Kartenhauses bilden eine Pyramide aus aufeinander aufbauenden Ebenen. Das Kartenhaus steht darum für die Idee der Vermittlung von Wissen vom Groben ins Feine (engl. *top down*).

Fallbeispiel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Der Reise-Butler ist eine intelligente Smartphone-App, die einen Fahrgast auf seiner Reise individuell begleitet. Die App versorgt ihn entlang der Fahrtroute von Verkehrsmitteln des öffentlichen Personenverkehrs mit Wissen aus dem Internet über vorbeiziehende Gebäude, Landmarken und andere gut sichtbare Sehenswürdigkeiten. Er bekommt zielgerichtet Informationen über Stationen, an denen er einsteigt, in ein anderes Verkehrsmittel umsteigt oder an seinem Reiseziel aussteigt. Die genaue Kenntnis des Fahrtverlaufes der geplanten Reise versetzt die App in die Lage, den Fahrgast aktiv auf Verspätungen oder andere Abweichungen vom Fahrplan hinzuweisen.

Der Reise-Butler ist eine Geschichte. Die Smartphone-App gibt es nicht, die Dokumentation im Wiki schon. Die Geschichte zeigt den Einsatz von **cards+** in einem agilen Entwicklungsprojekt. Die Handlung wird von folgenden Personen getragen (in der Reihenfolge ihres Erscheinens):



Moritz

Moritz ist Berater bei der xxx GmbH. Er besitzt fundierte Kenntnisse über verteilte event-basierte Systeme und Technologien zur Verarbeitung großer Datenmengen. Der Schwerpunkt seiner Tätigkeiten in Projekten ist der Entwurf und die Umsetzung komplexer Software-Systeme und das Coaching von Entwicklerteams.



Verena

Verena ist Mitgründerin und Geschäftsführerin der xxx GmbH. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.



Anne

Anne ist jüngstes Kind einer großen Familie. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.



Georg

Georg zeigte schon seit seinen Kindheitstagen großes Interesse für alles, was mit Computern zu tun hat. Einige Jahre lang betreute er nebenberuflich die IT-Landschaft eines mittelständischen Bauunternehmens. At vero eos et accusam et justo duo dolores et ea rebum.



Julius

Julius hat die Welt der Computer seit Jugendtagen fasziniert. Seine erste Webseite hat er bereits in sehr jungen Jahren programmiert - über sein Hobby, das Modellfliegen. Er liebt es zu tüfteln und Probleme zu lösen. Je schwieriger die Aufgabe desto besser. At vero eos et accusam et justo duo dolores et ea rebum.



Richard

Richard konnte sich scho in der Realschulzeit für Software begeistern. Er hatte immer einen Computer. Anfangs ein C64, dann diverse PCs. Jetzt liebt er seinen Mac. At vero eos et accusam et justo duo dolores et ea rebum.



Tim

Tim ist in Oxford (England) und Lüneburg aufgewachsen. Angefangen hat er vor vielen Jahren mit kleineren Projekten auf dem Einplatinencomputer Raspberry Pi, wie zum Beispiel eine selbst gebaute Überwachungskamera, ein NAS Server oder kleinere Webseiten. At vero eos et accusam et justo duo dolores et ea rebum.

In verschiedenen Situationen während der Analyse der Anforderungen, beim Entwurf des IT-Systems und bei der Realisierung der Software verwenden sie Bausteine von **cards+**. Sie besprechen typische Probleme und finden angemessene Lösungen. Dabei folgen sie den Prinzipien von **cards+** und nutzen die vorgeschlagenen Praktiken.



Personen und Handlung in der Geschichte sind frei erfunden. Etwaige Ähnlichkeiten mit tatsächlichen Begebenheiten, lebender oder verstorbener Personen wäre reiner Zufall. Fühlt sich jemand im positiven Sinne angesprochen, freut mich das.

Einleitung

Dokumentierte Informationen

Unternehmen brauchen das Wissen ihrer Beschäftigten. Wissen muss bewahrt und geteilt werden. Wissensinseln sind ein Risiko in jeder Organisation. Durch Fluktuation oder durch Änderungen in der Organisation entsteht ein regelmäßiger Wechsel beim Personal und bei anderen interessierten Parteien. Damit ist immer die Gefahr verbunden, dass Wissen verloren geht. Viel Wissen steckt auch in Software. Der Verlust von Wissen in Entwicklungsprojekten ist gefährlich. Die Produktvision geht verloren. Qualität und Produktivität sinkt. Anpassungen und Erweiterungen der Software werden schwieriger. Es ist nicht klar, was die Software jetzt schon leistet. Die Innovationskraft der Organisation sinkt. Agile Software-Entwicklung ist darum Teil der digitalen Transformation in vielen Unternehmen. Dabei kommt es aber nicht nur darauf an, agil vorzugehen, sondern generell flexibel zu sein. Agilität wird häufig in einem Atemzug mit Geschwindigkeit oder Effizienz genannt. Aber über Dokumentation wird nicht so häufig gesprochen. Es gibt die einen, die auf Dokumentation verzichten wollen.

»Code und Testpläne reichen!« sagen sie.

Sie sagen, auf Dokumentation kann in agilen Projekten sogar verzichtet werden und berufen sich dabei auf Punkt 2 im Manifest für agile Software-Entwicklung:

»Wir schätzen funktionierende Software mehr als umfassende Dokumentation.«

Sie vergessen dabei, dass es im Manifest ganz klar heißt, dass beide Werte wichtig sind, wir aber den Wert auf der linken Seite höher einschätzen. Andere wollen wiederum alles ganz genau aufschreiben.

»Wissen darf nicht verloren gehen!« fordern sie.

Dokumentation ist wichtig und (eigentlich) Pflicht. Abhängig von der Projektorganisation und dem gewählten Vorgehen - phasen-orientiert, iterativ oder agil - schreiben Projektmitarbeiter verschiedener technischer Disziplinen im Laufe der Zeit eine ganze Reihe dokumentierter Informationen:

- › Die Nutzer der Software benötigen Handbücher, Trainingsunterlagen und Online-Hilfen, um die Software verwenden zu können.
- › Die Betriebsorganisation braucht Anleitungen für Einführung und Betrieb der Software und Handlungsanweisungen für die Behebung von Störungen.
- › Produktverantwortliche braucht eine Systembeschreibung für sein Anforderungsmanagement. In einem agilen Entwicklungsprozess schreiben sie *User Stories* in einem *Product Backlog*.
- › Produktverantwortliche muss den Architekturentwurf für das System kennen. Er muss Kosten für den Betrieb und den Einsatz von Technologien gegenüber dem Auftraggeber begründen können. Selbst wenn ein Auftraggeber akzeptiert, dass eine Software-Architektur »einfach so« entsteht, sollten die fundamentalen Entscheidungen und grundlegenden Konzepte dann doch dokumentiert werden.
- › Entwickler kommentieren ihren Code. Sie schreiben Spezifikationen. Oft nutzen sie UML-Diagramme, um Zusammenhänge darzustellen. Sie tun das in der Regel für sich und andere Entwickler im Team.

An dokumentierten Informationen mangelt es in den meisten Projekten daher nicht.

Digitale Transformation

Die Digitale Transformation braucht Wissensmanagement. Unternehmen brauchen das Wissen ihrer Beschäftigten. Wissen muss bewahrt und geteilt werden. Wissensinseln sind ein Risiko in jeder Organisation. Viel Wissen steckt auch in Software. Agile Software-Entwicklung ist darum Teil der digitalen Transformation in vielen Unternehmen. Dabei kommt es aber nicht nur darauf an, agil vorzugehen, sondern generell flexibel zu sein. Agilität wird häufig in einem Atemzug mit Geschwindigkeit oder Effizienz genannt. Aber über Dokumentation wird nicht so häufig gesprochen. Es gibt die einen, die auf Dokumentation verzichten wollen.

»Code und Testpläne reichen!« sagen sie.

Andere wollen wiederum alles ganz genau aufschreiben.

»Wissen darf nicht verloren gehen!« fordern sie.

In diesem Spannungsfeld bewegt sich **cards+**. Eine agile Organisation schätzt funktionierende Software. Langfristig betrachtet braucht sie auch eine angemessene Dokumentation für die Software. Mit **cards+** sind beide Ziele von Anfang an erreichbar. Software und Dokumentation wird Hand in Hand inkrementell entwickelt.

Agile Software-Entwicklung

Der Ansatz agiler Software-Entwicklung ist empirisch, inkrementell und iterativ. Schritt für Schritt wird ihr Team besser und schneller. Selbstorganisiert. Maximal Motiviert. Der Fokus liegt auf Qualität und Exzellenz. In kurzen Zyklen entstehen kundenwirksame Inkremente.

Mit dem Einsatz von **cards+** ist jede agile Projektorganisation in der Lage, zielgerichtet und iterativ eine Produktdokumentation zu entwickeln. Zielgerichtet bedeutet, dass die erstellten Dokumente einen Wert für die Organisation hat. Iterativ bedeutet fortlaufende Wiederholung, verbunden mit der Chance, sich Schritt für Schritt zu verbessern. Dokumentieren wird zur Routine. Mit dem Fortschritt in der Entwicklung der Software wächst auch die Produktdokumentation, als integraler Bestandteil des Produktinkrements. Dokumentieren wird zur leistbaren Aufgabe des gesamten Teams. Die Dokumentation wird prüfbar, die Qualität messbar.

Strategisches Design

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Taktisches Design

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum

dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Gemeinsame Sprache

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Qualitätspolitik

Qualitätspolitik kennen wir aus dem Sprachgebrauch der [\[ISO-9001\]](#). Mit dem Begriff «Politik» tun sich viele schwer. Im Englischen spricht man von «*Policy*», übersetzt also Strategie oder Leitlinie. Agile Entwicklungsprozesse haben sich aus der Erkenntnis entwickelt, dass sich Anforderungen für ein IT-System nie so exakt spezifizieren lassen, dass kein Spielraum für Interpretationen bleibt. Die Welt dreht sich während der Realisierung im Projekt weiter. Was heute eine gute Idee ist, kann morgen schon ein alter Hut sein. Steuerungsmaßnahmen für die Qualität eines Entwicklungsprozesses, wie sie in der [\[ISO-9001\]](#) von einer Organisation gefordert werden, sind nur mit genauer Kenntnis der Entwicklungsergebnisse möglich.

Eine Dokumentation muss Qualitätsmerkmale wie Vollständigkeit und Richtigkeit aufweisen. Die [\[ISO-9001\]](#) lehrt uns, dass Qualitätsmerkmale messbar sein müssen. Die [\[ISO-25010\]](#) macht Vorschläge für die Kategorisierung von Qualitätsmerkmalen. Eine vollständige und korrekte Dokumentation ist ein wichtiges Qualitätsmerkmal einer Software in Bezug auf die Kategorie Wartbarkeit.

Jetzt hab ich's

Powered by

GitHub

1. Von der Idee zur Vision

Moritz sitzt in der Bahn und sieht aus dem Fenster. Er hat gerade keine Lust, sich mit etwas Sinnvollem zu beschäftigen. Schlafen geht auch nicht, weil er einfach nicht müde genug ist. Außerdem ist es im Großraumwagen zu hell und zu laut. Sein Blick schweift in die Ferne und bleibt an einem seltsamen Gebäude hängen.

«Welchen Zweck hat dieses Gebäude?», denkt er sich.

Einen Moment später ist es durch eine bewaldete Hügelkette verdeckt. Weil er eh gerade nichts Besseres zu tun hat, nimmt er sein Smartphone in die Hand. Er will herausfinden, was das für ein Gebäude ist.

«Welchen Begriff soll ich eingeben?», überlegt er.

Kino, Einkaufszentrum, Museum, er hat ja keine Ahnung, wozu dieses seltsame Gebäude verwendet wird. Die Suche nach "seltsames Gebäude Frankfurt" bringt - wenig überraschend - keine wirklich nützliche Erkenntnis. Er beschließt, Google Maps zu verwenden, um das Gebäude zu finden. Im Reiseplan, der im Zug für Reisende verteilt wird, kann er nachlesen, dass er sich gerade irgendwo zwischen Frankfurt und Fulda befindet. Aber auch das hilft nicht wirklich. Recht schnell verliert er das Interesse an der Suche.

«Pah, echt umständlich und anstrengend, diese Sucherei im Internet.», ärgert er sich und legt das Smartphone zur Seite.

Völlig vergessen kann Moritz diesen Vorfall nicht. Es ist ja auch nicht zum ersten Mal, dass er sich während einer längeren Fahrt mit dem Zug oder Auto fragt, welche spannenden Dinge da an ihm vorbeiziehen.

Eines Abends sitzt er allein in seinem Hotelzimmer. Da fällt ihm die Geschichte mit dem seltsamen Gebäude wieder ein. Er schnappt sich sein Tablet und gibt "android app berg kirche poi erkennen".

Bsp. 1. Suchergebnis bei Google (2017)

tbd Bild

Am Ende von Moritz' Recherche ist klar, dass es offensichtlich noch keine App gibt, die sein konkretes Problem löst. Er bekommt aber den Eindruck, dass es die technischen Möglichkeiten wie GPS und weitere Sensoren im Smartphone und die notwendigen Daten bei Google, Wikipedia oder Facebook gibt, um eine App zu realisieren. Die Idee zum Reise-Butler ist geboren.

Moritz beschließt, die Idee des Reise-Butlers aufzugreifen und eine erste Vision einer App zu entwickeln. Dazu stellt er sich folgende Fragen:

1. Wer sind die Nutzer für das Produkt?
2. Warum sollten sie das Produkt haben wollen?
3. Was macht das Produkt einzigartig?

4. Welche Fähigkeiten muss das Produkt haben?

Frage 1 ist leicht zu beantworten: Reisende in Zügen, Bussen, Beifahrer im Auto. Die App ist für jene Reisende, die sich für die vorbeiziehende Welt interessieren. Was gleichzeitig die Antwort auf Frage 2 ist. Zur Beantwortung von Frage 3 will er sich Zeit für eine Recherche im Internet und in den App-Stores von Google und Apple nehmen. Dabei legt er sich gleichzeitig fest, nur auf die beiden Smartphone-Plattformen von Google (Android) und Apple (iOS) zu setzen. Die Antwort auf Frage 4 ergibt sich aus der ursprünglichen Idee. Die App muss einen Reisenden in die Lage versetzen, schnell und sicher vorbeiziehende Sehenswürdigkeiten (also Gebäude, Landmarken oder Berggipfel) zu erkennen und ihm Zusatzinformationen aus dem Internet bereitzustellen. Sie muss leicht zu bedienen und schnell einsatzbereit sein, weil das vorbeiziehende Objekt nicht sehr lange sichtbar ist, besonders bei schnell fahrenden Zügen. Aus diesem Grund reicht es nicht, Objekte nur durch die Kamera zu identifizieren. Die App muss Kenntnis über den Verlauf der Reise haben, z.B. Start- und Zielort, Umstiege oder interessante Orte in der Umgebung der Strecke.

«Ich würde so eine App verwenden.», denkt er sich und zeigt sich mit dieser ersten Vision schon recht zufrieden.

Wenn die App jedoch Realität werden soll, muss Moritz daraus ein Projekt machen. Er muss eine Grundlage schaffen, um Unterstützer zu finden. Einen Namen für das Projekt hat er aber: Der Reise-Butler.

2. Entscheidung für cards+

Eine Idee ist die eine Sache, ihre Umsetzung eine andere. Moritz weiß, dass es ein langer Weg ist, aus der Idee des Reise-Butlers ein Produkt zu machen. Er will es aber probieren. In den agilen Methoden der Software-Entwicklung sieht er seine Chance. Scrum begleitet Moritz schon seit vielen Jahren. Er war Teil eines Scrum-Teams sowohl als Entwickler als auch als Produktverantwortlicher. Die Wirksamkeit von Scrum als Werkzeug steht für ihn außer Frage, er vertraut der agilen Methode. Ob Scrum die richtige Wahl ist, hängt nach seiner Erfahrung ganz stark von den Menschen im Projekt ab. Scrum ist einfach, erfordert aber ein großes Maß an Disziplin. Für die erfolgreiche Einführung und Durchführung von Scrum benötigt eine Projektorganisation eine ganze Menge Vertrauen der interessierten Parteien. Innerhalb der Projektorganisation müssen die Rollen richtig besetzt werden. Jeder Projektmitarbeiter muss seine Möglichkeiten im Rahmen von Scrum kennen, muss in seiner Rolle Grenzen respektieren und Pflichten akzeptieren.

Das Thema Dokumentation bei agiler Software-Entwicklung durfte Moritz schon mehrfach in Scrum-Teams kontrovers diskutieren. Das Spektrum ist breit.

«Lesbarer Code ist besser als Dokumentation.», sagen die Einen.

«Das Backlog ist genug Dokumentation.», behaupten Andere.

«Funktionierende Software ist wichtiger als umfassende Dokumentation.», ist einer der Werte des agilen Manifests.

Dokumentation muss angemessen sein. Das bedeutet für Moritz, dass sie gebraucht wird, also Leser hat, und hilfreich, vollständig und korrekt ist. Vermutlich sogar in dieser Reihenfolge. Dokumentation als Alibi möchte er verhindern. Aber in vielen großen Unternehmen wird der Umfang der Dokumentation durch Vorgaben und Richtlinien bestimmt. Moritz setzt beim Thema Dokumentation auf die Methode cards+. Die Methode kennt er schon. Er weiß, er kann sie sofort einsetzen, noch bevor es ein Projekt gibt. Sie wird ihm helfen, seine Dokumentation in einem Wiki inkrementell zu erstellen.

Die Methode cards+ ist ein agiler Ansatz, um Wissensmanagement in einem Software-Entwicklungsprojekt richtig zu organisieren und zielgerichtet und inkrementell zu dokumentieren.

Mit dem Einsatz von cards+ ist jede agile Projektorganisation in der Lage, zielgerichtet und iterativ eine Produktdokumentation zu entwickeln. Zielgerichtet bedeutet, dass die erstellten Dokumente einen Wert für die Organisation haben. Iterativ bedeutet fortlaufende Wiederholung, verbunden mit der Chance, sich Schritt für Schritt zu verbessern. Mit dem Fortschritt in der Entwicklung der Software wächst auch die Produktdokumentation, als integraler Bestandteil des Produktinkrements. Dokumentieren wird zur leistbaren Aufgabe des gesamten Teams. Die Dokumentation wird prüfbar, die Qualität messbar.

cards+ unterstützt eine Projektorganisation, Wissensmanagement in einem agilen Entwicklungsprojekt richtig zu organisieren. Der Produktverantwortliche setzt cards+ ein, um zielgerichtet und inkrementell sein Produkt angemessen zu dokumentieren, unterstützt durch ein selbstorganisiertes Team.

cards+ tritt mit dem Versprechen an, die Erstellung und Pflege einer qualitativ hochwertigen Produktdokumentation in einem Wiki zu ermöglichen. Viele Prinzipien und Praktiken der Entwickler gelten auch für Autoren in einem Wiki, in den meisten Fällen in nur leicht abgewandelter Form.

cards+ ist kein geschlossenes Vorgehensmodell, kein isolierter Prozess. Aus diesem Grund kann cards+ sehr gut in bestehende Prozesse integriert werden.

Das Wiki ist für Moritz ein sehr wichtiges Werkzeug. Er kann und will jederzeit an den Inhalten arbeiten, auch unterwegs. Moritz entscheidet sich für *Confluence*. Sein Arbeitgeber betreibt eine eigene Installation. Dort kann er einen eigenen Bereich für die Produktdokumentation des Reise-Butlers einrichten.

***Confluence* vom australischen Unternehmen Atlassian ist ein leistungsfähiges Wiki und sehr gut geeignet für den Aufbau einer Produktdokumentation. *Confluence* hat alle Funktionen, die notwendig sind für eine erfolgreiche Umsetzung der Methode cards+.**

Ohne ein Wiki wird die Methode cards+ nicht gelingen! *Confluence* ist mit Sicherheit eine gute Wahl. Durch die große Zahl an Add-Ons gibt es praktisch für jedes Problem im Zusammenhang mit Dokumentation eine brauchbare Lösung. Die Kombination von *Confluence* und cards+ ist in der Praxis mehrfach erprobt. Sie versetzt jedes agile Entwicklungsprojekt in die Lage, Produktwissen schnell und in hoher Qualität zu entwickeln und ohne Hürden allen interessierten Parteien zur Verfügung zu stellen.

Das Wiki bildet das zentrale Verzeichnis für dokumentierte Informationen in anderen Werkzeugen, z.B. Spezifikationen in einem Git-Repository, Testpläne in einem Testmanagementsystem oder Normen, Richtlinien, Verordnungen, die für das Produkt relevant sind, in einem Dokumentenmanagementsystem.

Den Baustein [\[Topic\]](#) will Moritz nutzen, um die Vision des Reise-Butlers zu strukturieren. Wichtige Randbedingungen wie die Beschränkung auf Android und iOS erfasst er mit dem Baustein [\[Decision\]](#). Ganz wichtig ist ihm ein Glossar. Er will sich so ganz früh auf eindeutige Begriffe festlegen, die ihm wichtig sind.

Der Baustein Decision wird verwendet, um konsequent alle fundamentalen technischen Entscheidungen oder vom Auftraggeber geforderten Qualitätsmerkmale, deren Rücknahme bzw. Veränderung große Auswirkung auf das Produkt haben, zu dokumentieren.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3. Domänensprache

Nach der Einrichtung des Bereiches für die Produktdokumentation des Reise-Butlers in *Confluence* beginnt Moritz mit dem Glossar, einem wichtigen Qualitätsmerkmal von **cards+**. Moritz fängt mit den of-fensichtlichen Begriffen an. Beim Reise-Butler geht es um eine Reise. Klar. Für jeden Begriff legt er eine Seite im Glossar an. Als Vorlage nutzt er den Baustein [\[Term\]](#). Er schreibt sofort mindestens einen Satz dazu. Um die Beschreibungen kompakt zu halten, verknüpft er die Begriffe miteinander. Dadurch kann er die einzelnen Beschreibungen auf das Wesentliche beschränken. Er braucht nicht jeden Begriff, den er erwähnt, immer wieder aufs Neue erklären. Mit diesem Vorgehen vermeidet er mehrfache Beschreibungen eines Begriffes. Er bleibt konsistent in seiner Beschreibung.

*«Ist der Nutzer der App ein Reisender, Fahrgast oder Passagier?»,
fragt er sich.*

Moritz entscheidet sich für Fahrgast. Er lässt sich besser vom Begriff Reise unterscheiden, hat einen Bezug zu Fahrplan bzw. Fahrt und ist nach seinem Empfinden nicht so sperrig wie Passagier.

Bsp. 2. Seite Fahrgast

tbd Bild

Die Erklärung besteht gerade mal aus zwei Sätzen. In diesen zwei Sätzen stecken aber eine ganze Reihe weiterer Begriffe: Fahrzeug, Verkehrsmittel, Personal, Verkehrsunternehmen. Beim Schreiben stellt er fest, dass es gar nicht so einfach ist, jeden Begriff prägnant und korrekt zu beschreiben. Viele Begriffe haben mehr als eine Bedeutung. Für den Begriff Fahrgast gibt es zwei weitere Begriffe: Passagier und Reisender. Moritz will aber eindeutige Begriffe. Für die Begriffe Passagier und Reisender legt er deshalb keine Seite an. Stattdessen trägt er die Begriffe zusätzlich zum Hauptbegriff Fahrgast als Stichworte in der Seite ein.

Ein wichtiges Ziel des Glossars ist es, Klarheit in die Begriffe zu bringen. Das Glossar baut er so auf, dass es für jeden Buchstaben im Alphabet eine eigene Seite gibt. Die beiden Buchstaben X und Y fasst er zusammen. Begriffe, die mit einer Ziffer beginnen, fasst er ebenfalls zusammen. Jeder dieser Übersichtsseiten zeigt eine Liste aller Begriffe, die diesen Anfangsbuchstaben haben. Damit findet jeder Leser schnell heraus, ob es überhaupt den Begriff im Glossar gibt, den er sucht.

Schnell stellt Moritz fest, dass *Confluence* bei der Eingabe von Links so gewisse Tücken hat. Durch *Copy&Paste* aus der Leseansicht heraus werden Verknüpfungen zu externen Web-Links, um nur ein Problem zu nennen. Das ist nicht gut. Zum Abschluss der Bearbeitung einer Seite wirft Moritz darum immer einen kurzen Blick auf die Seiteninformation. Er prüft, ob alle Links in dieser Seite plausibel sind.

Bsp. 3. Seiteninformation Fahrgast

tbd Bild

Diese kleine Aufmerksamkeit kostet ihn nur wenig Zeit. Sie hilft aber, die Qualität der Seite zu erhöhen. Die Startseite des Glossars enthält darum eine schrittweisen Anleitung für das Anlegen neuer Begriffe und weitere wichtige Hinweise für die Pflege der Seiten. Außerdem bietet die Startseite eine Suche beschränkt auf diesen Bereich. Die Aktivitäten im Glossar werden durch die

Anzeige der zuletzt geänderten Begriffe sichtbar.

Bsp. 4. Startseite für das Glossar

tbd Bild

Suchen ist bereits Teil unserer Kultur. Nicht ohne Grund ist «googeln» der Begriff für eine Suche oder Recherche im Internet. Was liegt also näher, als diese Idee auch im Wiki umzusetzen.

Bsp. 5. Volltextsuche im Glossar

tbd Bild

Die Suche nach Stichworten ist wesentlich exakter als die Volltextsuche. Stichworte lassen sich kombinieren. So lässt sich schrittweise das Suchergebnis verfeinern.

Bsp. 6. Stichwortsuche im Glossar

tbd Bild

Im Verlauf der weiteren Analyse wird Moritz die Begriffe noch weiter verfeinern, manche sogar in ihrer Bedeutung schärfen. Er ist überzeugt, dass eindeutige Begriffe wichtig sind. Sie führen zu einer gemeinsamen Domänensprache. Konsequenterweise hilft das Glossar, Widersprüche in den Bausteinen der Systembeschreibung zu erkennen. Das ist gut für die Qualität der Dokumentation.

Der Baustein Term realisiert das Konzept für ein Glossar. Mit dem Baustein wird ein wichtiger Begriff der Anwendungsdomäne, in der die Software zum Einsatz kommt, zentral und verlinkbar im Glossar erfasst.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Reise-Butler 05: Ziele und Grenzen

Endlich geht es los!

Powered by

GitHub

4. Kapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Reise-Butler 06: Hochschulprojekt

tbd Reise-Butler 07: Erste Entscheidungen

tbd Der Auftrag ist da, wir starten, Forming-Phase im Team

tbd Reise-Butler 08: Kickoff an der TU

tbd Reise-Butler 09: Der Baustein Epic

tbd Reise-Butler 10: Anforderungsanalyse

tbd Reise-Butler 11: Architekturentwurf

So viel Neues..

Powered by

GitHub

5. Kapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Reise-Butler 12: Komponentenüberblick

tbd Reise-Butler 13: Fachklassenmodell

tbd Reise-Butler 14: Gemeinsame Sprache

tbd Wir sind kreativ, Storming-Phase im Team

tbd Beispiel Stationswissen (RISK & ZHV) zeigt, welche Probleme entstehen, wenn zu viele Details "handgeschrieben" und nicht versioniert im Wiki geschrieben werden. Im Beispiel zeige, wie die fachliche Beschreibung aus dem Case in AsciiDoc übertragen wird. Aus den Cases "Auswertung *I HG-Zeile", "Befüllung des Ausgangstopics", "Ermittlung IFOPT" und "Erzeugung UUID" wird ein einziger Case, "Haltestellen für ÖV in Deutschland liefern".

tbd Beispiel Wendeermittlung zeigt, wie eine komplexe Logik in den Cases "Wende erzeugen", "Wende ungültig machen" und "Wendeparameter konfigurieren" sehr abstrakt dargestellt wird, im Detail aber nur in der Spezifikation erklärt.

tbd Beispiel Fahrplanerstellung zeigt, wie viel zu viele Informationen im Wiki entstehen, weil es natürlich leichter ist.

Es läuft!

Powered by

GitHub

6. Kapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Reise-Butler 15: Angemessene Dokumentation

tbd Reise-Butler 16: Programmieren und Dokumentieren

tbd Reise-Butler 17: Spielregeln

tbd Wir sind im Fluss, Performing-Phase im Team

tbd Reise-Butler 18: Erstes Produktinkrement

tbd Reise-Butler 19: Der Baustein Case

tbd Reise-Butler 20: Projektumfang

tbd Reise-Butler 21: Durchstich

tbd Regelung aus PZ, dass im Wiki **immer** der Stand der Produktion gezeigt wird.

Fertig!

Powered by

GitHub

7. Kapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Reise-Butler 22: Arbeitsgruppen

tbd Reise-Butler 23: Der Baustein Event

tbd Reise-Butler 24: Architekturwand

tbd Reise-Butler 25: Der Baustein Entity

tbd Reise-Butler 26: Randbedingungen

tbd Reise-Butler 27: Weniger ist mehr

tbd Reise-Butler 28: Umgebungsübersicht

tbd Reise-Butler 29: Arbeitsdokumente

tbd Wir haben das Ziel erreicht, Team löst ich auf

Alles auf Anfang

Powered by

GitHub

8. Kapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

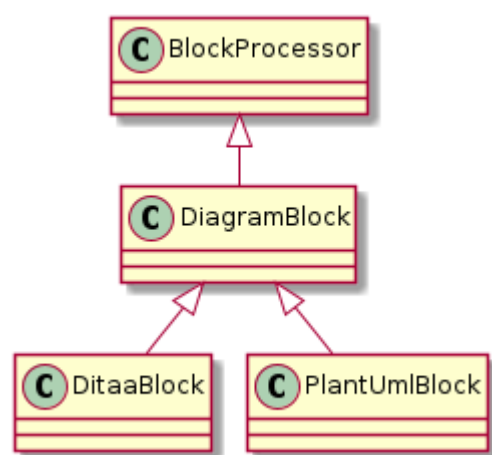


Abb. 1. PlantUML

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

8.1. Gantt diagramm

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

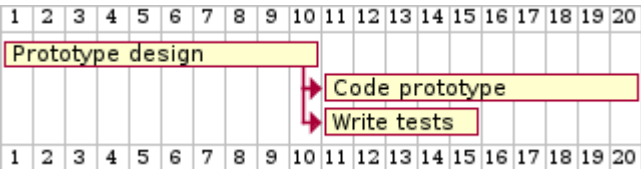


Abb. 2. PlantUML

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

8.2. Mindmap

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

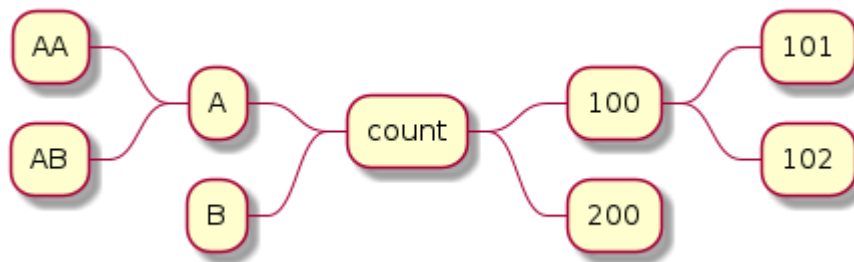


Abb. 3. PlantUML

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

8.3. Wireframe

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

Just plain text

☐ Unchecked radio

☒ Checked radio

☐ Unchecked box

☒ Checked box

Abb. 4. PlantUML

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Neuer Auftrag, neues Team

tbd Reise-Butler 30: Pimp my Wiki (mit neuem Team)

tbd Reise-Butler 31: Perspektiven

tbd Reise-Butler 32: Qualitätsmerkmale

tbd Reise-Butler 33: Minimum Viable Case/Product

tbd Beispiel wie bei PZ der Austausch der Implementierung für den UIC-Adapter wegen neuer Technologie. Hervorheben, dass die Produktdokumentation weitestgehend stabil bleibt, nur die Spezifikation ändert sich natürlich.



Powered by

GitHub

9. Kapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

tbd Wartung, kleines Team

tbd Vision Kontinuierliche Produktentwicklung

tbd Vision Produktinnovation

tbd Vision Produktkrise

tbd Lessons learned, Fazit, Kontakt



Powered by

GitHub

Danksagung

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

Anhang A: Bla bla

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

Anhang B: Bla blubb

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

Stichwortverzeichnis

Anleitung

Lorem ipsum dolor sit amet.

Anweisung

Lorem ipsum dolor sit amet.

Architekturentwurf

Lorem ipsum dolor sit amet.

Asciidoc

Lorem ipsum dolor sit amet.

Asciidoctor

Lorem ipsum dolor sit amet.

Case

Lorem ipsum dolor sit amet.

Decision

Lorem ipsum dolor sit amet.

Epic

Lorem ipsum dolor sit amet.

Handbuch

Lorem ipsum dolor sit amet.

ISO 25010

Lorem ipsum dolor sit amet.

ISO 9001

Lorem ipsum dolor sit amet.

Systembeschreibung

Lorem ipsum dolor sit amet.

Systemstruktur

Lorem ipsum dolor sit amet.

Term

Lorem ipsum bla bla bla

Topic

Lorem ipsum dolor sit amet.

Wiki

Lorem ipsum dolor sit amet.

Literaturverzeichnis

- ☒ [ccd] Clean Code Developer
- ☒ [dac] Docs-as-Code
- ☒ [ddd] Domain-Driven Design
- ☒ [gof] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1994.
- ☒ [pp] Andy Hunt & Dave Thomas. The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley. 1999.
- ☒ [rb1] Robert Bruckbauer, Agil dokumentieren mit cards+. Wordpress. 2020.
<https://cardsplus.info>
- ☒ [rb2] Robert Bruckbauer, Agil dokumentieren mit cards+. Xing. 2020.
<https://www.xing.com/communities/groups/agil-dokumentieren-1084817>