**Grobplanung**

Projektziele

Hauptziele

HZ01 – Dezentralisierte Applikation

Um eine dezentralisierte Applikation bereitstellen zu können soll die Entwicklung des Projektes auf der Ethereum Blockchain mittels des Truffle Frameworks erfolgen.

HZ02 – Zugriff auf die Ethereum Blockchain

Um auf die Ethereum Blockchain zugreifen zu können soll die Browsererweiterung MetaMask verwendet werden.

HZ03 – Framework für die Entwicklungsumgebung auf der Ethereum Blockchain

Als Framework für die Entwicklung auf der Ethereum Blockchain soll Truffle verwendet werden.

HZ04– Lokales Testen für die Ethereum Blockchain

Um die Entwicklung lokal zu simulieren soll Ganache verwendet werden.

HZ05– Übermittlung der Daten an die Ethereum Chain mittels JSON

Die Übermittlung der Daten an die Ethereum Chain soll mittels JSON erfolgen.

HZ06 – Programmiersprache, um einen Smart Contract auf der Ethereum Blockchain zu programmieren.

Solidity soll als Programmiersprache gewählt werden, um auf der Ethereum Blockchain zu programmieren.

HZ07 – Speicherung von Bildern in IPFS

Bilder von der Unfallstelle und den beschädigten Fahrzeugen, sollen mittels IPFS (InterPlanetary File System) abgespeichert werden. Hierbei ist zu beachten, dass die Hashwerte der Bilder in der Ethereum Blockchain gespeichert werden. Durch die Angabe eines Pfades werden die Bilder in der Ethereum Blockchain eindeutig referenziert. Aus Datenschutzgründen dürfen die Bilder nicht öffentlich zugänglich gemacht werden, daher sollen die Bilder mittels IPFS kryptographisch verschlüsselt werden, damit sie nur für die Beteiligten Personen ersichtlich sind.

HZ08 – Erfassung der Unfallskizze

Eine Unfallskizze, die grafisch veranschaulicht, wie sich der Unfall ereignete, ist laut europäischem Unfallbericht notwendig. Um diese zu digitalisieren soll die Möglichkeit geboten werden in Google Maps Screenshots von der Straße/Gasse/Kreuzung wo sich der Unfall ereignete zu machen. Anschließend sollen mit Pfeilen auf dem Screenshot gekennzeichnet werden, woher die Beteiligten Fahrer kamen, bevor sich der Unfall ereignete. Screenshots/Bilder von der Unfallskizze sollen anschließend hochgeladen werden können.

HZ09 – Eingabefelder für den Unfallbericht

Folgende Felder sollen zur Benutzereingabe in einem digitalen Formular bereitgestellt werden:

* Tag des Unfalls, Uhrzeit (Textfeld)
* Ort (Gemeinde, Straße, Haus-Nr. bzw. Kilometerstein) (Textfeld)
* Verletzte? (auch leicht) (Checkbox)
* Andere Sachschäden als an den Fahrzeugen A u. B (Checkbox)
* Zeugen (Name, Anschrift, Telefon; Insassen von A und B unterstreichen) (Textfeld)

Folgende Felder sind für jeweils beide Fahrzeuge/Fahrer auszufüllen:

* Versicherungsnehmer
  + Name (Textfeld)
  + Vorname (Textfeld)
  + Anschrift (Textfeld)
  + Telefon (Textfeld)
  + Besteht Berechtigung zum Vorsteuerabzug (Checkbox)
* Fahrzeug
  + Marke, Typ (Textfeld)
  + Amtl. Kennzeichen (Textfeld)
* Versicherer
  + Vers.-Nr (Textfeld)
  + Agent (Textfeld)
  + Nr. der Grünen Karte (Textfeld)
  + Versicherungsausweis oder Grüne Karte, gültig bis (Textfeld)
  + Besteht eine Vollkaskoversicherung (Checkbox)
* Fahrer
  + Name (Textfeld)
  + Vorname (Textfeld)
  + Adresse (Textfeld)
  + Führerschein-Nr (Textfeld)
  + Klasse (Textfeld)
  + Ausgestellt durch (Textfeld)
  + Gültig ab (Textfeld) bis (Textfeld)
* Bezeichnen Sie durch einen Pfeil den Punkt des ersten Anstoßes (Grafik)
* Sichtbare Schäden (Textfeld)
* Bemerkungen (Textfeld)
* Umstände
  + Fahrzeug parkte (auf der Straße) (Checkbox)
  + Fuhr aus der Parkstelle heraus (Checkbox)
  + Fuhr in eine Parkstelle hinein (Checkbox)
  + Fuhr aus einem Parkplatz, aus einem Grundstück oder einem Feldweg/ Privatweg ein (Checkbox)
  + Bog in einen Kreisverkehr ein (Checkbox)
  + Fuhr im Kreisverkehr
  + fuhr heckseitig auf ein anderes Fahrzeug auf bei Fahrt in dieselbe Richtung und auf derselben Fahrspur fuhr in gleicher Richtung, aber in einer anderen Spur (Checkbox)
  + wechselte die Spur (Checkbox)
  + überholte (Checkbox)
  + bog rechts ab (Checkbox)
  + bog links ab (Checkbox)
  + setzt zurück (Checkbox)
  + fuhr in die Gegenfahrbahn (Checkbox)
  + kam von rechts (Checkbox)
  + beachtete Vorfahrtzeichen nicht (Checkbox)
  + Anzahl der angekreuzten Felder (locked, Systemermittelt)
* Unfallskizze (Bildupload)

HZ10 – Führerschein und Zulassungsschein hochladen

Als Identifikationsnachweis soll zusätzlich die Option geboten werden, die Führerscheine und die Zulassungsscheine der Beteiligten Personen hochzuladen.

Optionale Ziele

OZ01 – Sprachauswahl einstellen

Um eine benutzerfreundliche Applikation bereitstellen zu können, soll am Anfang des Formulars die Sprache eingestellt werden können. Je nach Sprache sollen sich die Labels der einzelnen Formularelemente anpassen. Dies soll durch das Verwenden von JSON Dateien ermöglicht werden.

OZ02 – Führerschein einscannen

Um die Eingabefelder automatisch zu befüllen, soll die Option geboten werden, den Führerschein einzuscannen und die Informationen aus dem Führerschein zu entnehmen.

Nicht Ziele

NZ01 – Bilder sollen öffentlich zugänglich sein

Von den Benutzern hochgeladene Bilder sollen der Öffentlichkeit zugänglich gemacht werden und jeder soll auf die Bilder zugreifen können.

**Aufgabenverteilung**

Aksakal Ozan

|  |  |
| --- | --- |
| Themenschwerpunkt: | Bereitstellung des Formulars und Validierung von Benutzereingaben |
| Auflistung der einzelnen Ziele  und Anforderungen | HZ01, HZ02, HZ03, HZ04, HZ05, HZ06, HZ09 |

Bedöcs Julian

|  |  |
| --- | --- |
| Themenschwerpunkt: | Bereitstellung des Front-Ends und Übermittlung der Formulardaten |
| Auflistung der einzelnen Ziele  und Anforderungen | HZ01, HZ02, HZ03, HZ04, HZ05, HZ06, OZ01, OZ02 |

Konca Burak

|  |  |
| --- | --- |
| Themenschwerpunkt: | Hochladen und abspeichern von Bildern |
| Auflistung der einzelnen Ziele  und Anforderungen | HZ01, HZ02, HZ03, HZ04, HZ05, HZ06, HZ07, HZ08, HZ10 |

Mousawi Maral

|  |  |
| --- | --- |
| Themenschwerpunkt: | Hochladen und abspeichern von Bildern |
| Auflistung der einzelnen Ziele  und Anforderungen | HZ01, HZ02, HZ03, HZ04, HZ05, HZ06, HZ07, HZ08, HZ10 |

**Geplanter Projektablauf**

Meilensteine

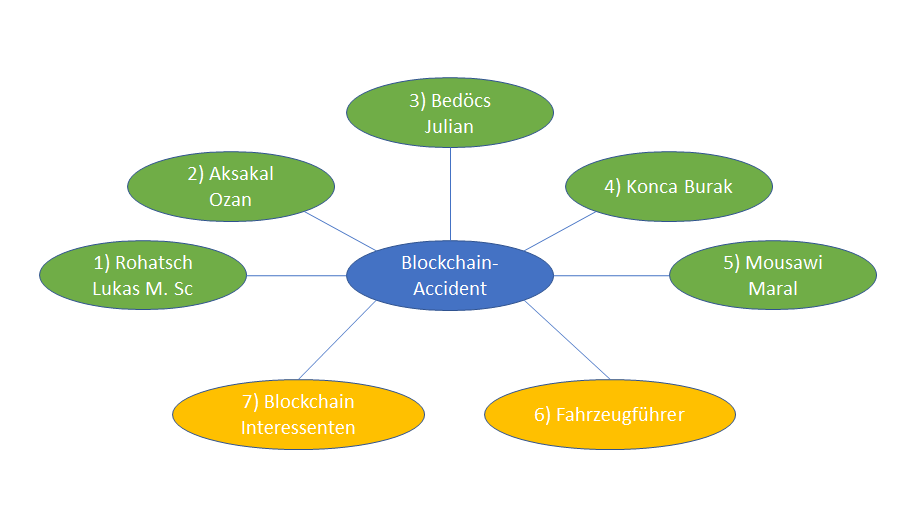
|  |  |  |  |
| --- | --- | --- | --- |
| # | KW | Datum | Beschreibung |
| 1 | 40 | 04.10.2020 | Projektstart |
| 2 | 45 | 08.11.2020 | Projektforschung fertiggestellt |
| 3 | 46 | 15.11.2020 | Fertigstellung der Grobplanung |
| 4 | 48 | 29.11.2020 | Formular bereitstellen |
| 5 | 04 | 31.01.2021 | Benutzereingaben validieren |
| 6 | 08 | 28.02.2021 | Bilder/Unfallskizze in die Chain beziehen |
| 7 | 10 | 14.03.2021 | Upload der Daten auf die Chain |
| 8 | 12 | 28.03.2021 | Frontend fertiggestellt |
| 9 | 16 | 25.04.2021 | Projektabschlussdokument fertiggestellt |
| 10 | 24 | 20.06.2021 | Testing abgeschlossen |
| 11 | 27 | 04.07.2021 | Blockchain Accident gelangt in den Echteinsatz |

Sprints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beginn | Ende | Hinweis | Meilensteine |
| 0 | 04.10.2020 | 04.10.2020 | Projektstart | 1 |
| 1 | 05.10.2020 | 18.10.2020 |  |  |
| 2 | 19.10.2020 | 01.11.2020 |  |  |
| 3 | 02.11.2020 | 15.11.2020 |  | 2,3 |
| 4 | 16.11.2020 | 29.11.2020 |  | 4 |
| 5 | 30.11.2020 | 13.12.2020 |  |  |
| 6 | 14.12.2020 | 27.12.2020 |  |  |
| 7 | 28.12.2020 | 10.01.2021 |  |  |
| 8 | 11.01.2021 | 24.01.2021 |  |  |
| 9 | 25.01.2021 | 07.02.2021 |  | 5 |
| 10 | 08.02.2021 | 21.02.2021 |  |  |
| 11 | 22.02.2021 | 07.03.2021 |  | 6 |
| 12 | 08.03.2021 | 21.03.2021 |  | 7 |
| 13 | 22.03.2021 | 04.04.2021 |  | 8 |
| 14 | 05.04.2021 | 18.04.2021 |  |  |
| 15 | 19.04.2021 | 02.05.2021 |  | 9 |
| 16 | 03.05.2021 | 16.05.2021 |  |  |
| 17 | 17.05.2021 | 30.05.2021 |  |  |
| 18 | 31.05.2021 | 13.06.2021 |  |  |
| 19 | 14.06.2021 | 27.06.2021 |  | 10 |
| 20 | 28.06.2021 | 11.07.2021 | Projektabschluss | 11 |

**Projektumfeldanalyse**

Grafische Darstellung



Beschreibung der wichtigsten Umfelder

|  |  |  |  |
| --- | --- | --- | --- |
| # | Bezeichnung | Beschreibung | Bewertung |
| 1 | Rohatsch Lukas M. Sc. | Projektbetreuer | + |
| 2 | Aksakal Ozan | Projektmitarbeiter | + |
| 3 | Bedöcs Julian | Projektmitarbeiter | + |
| 4 | Konca Burak | Projektleiter | + |
| 5 | Mousawi Maral | Projektmitarbeiter | + |
| 6 | Fahrzeugführer | Zielgruppe | ~ |
| 7 | Blockchain-Interessenten | Potentielle Kunden | ~ |

**Risikoanalyse**

P… Eintrittswahrscheinlichkeit

A… Schadensausmaß bei Eintritt des Risikos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Bezeichnung | Beschreibung | P | A | RPZ |
| 10 | Manipulationsrisiko, bei der Nutzung von Daten außerhalb der Blockchain | Aufgrund der global abgespeicherten Bilder entsteht eine Sicherheitslücke, wodurch diese manipuliert werden können | 30 | 60 | 1800 |
| 6 | Mangelhafte Projektforschung | Aufgrund mangelhafter Projektforschung, die von den Projektmitgliedern betrieben wurde, kommt es zu einer Verzögerung des Projektes | 40 | 30 | 1200 |
| 1 | Verspätung der Frontend Fertigstellung | Aufgrund der verspäteten Fertigstellung des Frontends wird das Projekt nicht rechtzeitig bis zum 15.07.2021 fertig | 10 | 100 | 1000 |
| 4 | Unzureichendes Projektcontrolling | Aufgrund von unzureichendem Projektcontrolling, kommt es zu einer Verspätung der Fertigstellung des Projektes | 20 | 50 | 1000 |
| 7 | Verspätung der Validierung der Benutzereingaben | Aufgrund verschiedener Faktoren wird die Benutzereingaben Validierung nicht bis zum Ende des 3. Semesters fertiggestellt | 20 | 50 | 1000 |
| 3 | Mangelhafte Dokumentation | Wenn die Richtlinien nicht eingehalten werden, kann es zu einer nicht überschaubaren Dokumentation kommen | 30 | 30 | 900 |
| 5 | Inkompatible Software | Aufgrund von Kompatibilitätsschwierigkeiten der verwendeten Software, muss neue Software bestimmt werden | 20 | 40 | 800 |
| 9 | Zugriff der Smart Contracts auf externe Daten in der Blockchain nicht möglich | Aufgrund der fehlerhaften Einbindung externer Daten, kann auf die hochgeladenen Bilder, die außerhalb des definierten Blockchain-Netzwerks abgespeichert sind, nicht zugegriffen werden. | 20 | 40 | 800 |
| 8 | Bugs in der fertigen Software | Aufgrund unzureichendem Testing befinden sich Bugs in der fertigen Software | 30 | 20 | 600 |
| 2 | Unzureichende Projektplanung | Aufgrund von unzureichender Projektplanung, kommt es zu einer Verspätung der Fertigstellung des Projektes | 10 | 50 | 500 |

RPZ… errechnete Risikoprioritätszahl

**Projektressourcen**

Projektressourcen: Soll – Ist Vergleich

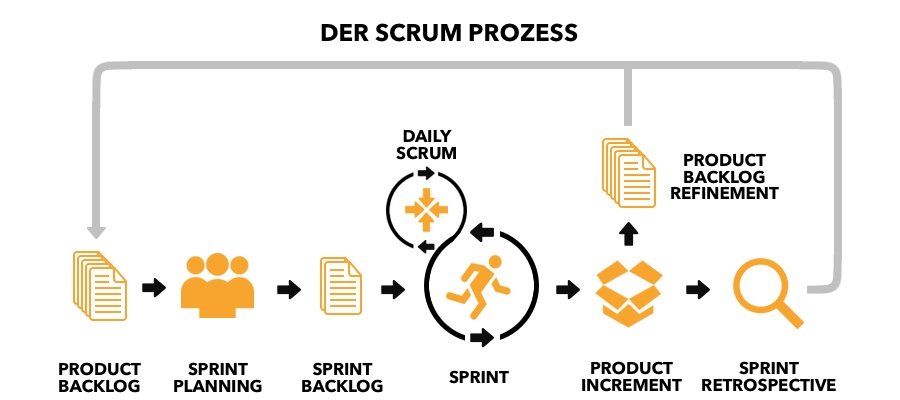
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Soll | | Ist | Risiko(X) | PSP(X) |
| KNOW-HOW im Bereich: Entwicklung auf der Blockchain |  | | X |  |
| KNOW-HOW im Bereich: Softwareentwicklung in Solidity |  | | X |  |

Personelle Ressourcen (geplant)

|  |  |
| --- | --- |
| Projektmitglied | Personenstunden |
| Aksakal | 150 |
| Bedöcs | 150 |
| Konca | 150 |
| Mousawi | 150 |

**Projektmanagementprozess**

Als Grundlage für unseren internen Projektmanagementprozess wurde Scrum für agile Softwareentwicklung gewählt. Die Grundidee und Struktur von Scrum wurde übernommen und an unsere Timeboxes und individuellen Bedürfnissen angepasst. Der Managementprozess, welcher für das Projekt Smart Mirror definiert wurde, wird im Folgenden beschrieben.

**

**WEEKLY**

Product Backlog

Der Product Backlog enthält alle Anforderungen an das Produkt, wie beispielsweise in der Softwareentwicklung die Anforderungen an eine zu erstellende Software. Die Qualität des Product Backlogs ist für die erfolgreiche Abwicklung eines Projekts ausschlaggebend.

Zuständig für den Product Backlog ist der Product Owner, der diesen ständig weiterentwickelt und pflegt. Zusätzlich ordnet und priorisiert er die enthaltenen Anforderungen. Da der Product Owner die Liste der Anforderungen kontinuierlich weiterentwickelt, handelt es sich beim Product Backlog nicht um ein Lastenheft im traditionellen Sinn, sondern um eine dynamische Liste.

Der Product Backlog wird während der gesamten Projektlaufzeit gepflegt. Die Anforderungen werden Schritt für Schritt in Sprints bearbeitet.

Sprint Planning

Vor jedem Sprint (Arbeitszyklus) findet ein Sprint Planning statt. Dieses sollte nicht länger als zwei Stunden dauern. Hierbei werden die zu erledigenden Teilaufgaben für den nächsten Sprint in Form von User-Stories beschrieben bzw. aus dem Backlog übernommen. Geplant wird der Sprint innerhalb des gesamten Teams, festgelegt werden die Teilaufgaben schließlich vom Product Owner. Die detaillierte Beschreibung einer User-Story erfolgt hingegen von jenem Teammitglied, dem die Teilaufgabe zugewiesen ist. Welche Aufgaben als nächstes zu erledigen sind wird von unserem Projektstrukturplan und den festgelegten Meilensteinen eingegrenzt und dementsprechend priorisiert.

Sprint Backlog

In jedem Sprint soll ein funktionsfähiges Zwischenprodukt entwickelt werden. Deshalb wird bereits vorab im Sprint Planning Meeting entschieden, welche Anforderungen aus dem Product Backlog im nächsten Sprint bearbeitet werden sollen. Das Sprint Planning Meeting findet zu Beginn jedes Sprints zur Planung dieser Projektetappe statt.

Nachdem die Anforderungen ausgewählt wurden, werden sie im jeweiligen Sprint Backlog festgehalten. Letztendlich entscheidet der Product Owner allein, welche Anforderungen im nächsten Sprint bearbeitet werden und damit im Sprint Backlog landen. Der Product Owner nimmt eine der drei zentralen Rollen im Scrum (Product Owner, Scrum Master, Scrum Team) ein. Er ist für die Erstellung einer konkreten Produktversion zuständig. Hierfür stellt er fachliche Anforderungen an das Projekt, priorisiert diese und steht in regelmäßigem Kontakt mit dem Auftraggeber.

Sprint

Im Sprint werden die im Sprint Planning festgelegten Stories abgearbeitet. Ein Sprint ist immer festgelegt auf zwei Wochen, wobei dieser immer an einem Montag beginnt und am Sonntag endet. In dieser Zeit sind die Mitglieder hauptsächlich auf sich allein gestellt und für die Abarbeitung der individuellen Aufgaben selbst verantwortlich. Bei Problemen kann jedoch immer eine spontane Besprechung zwischen mehreren Teammitgliedern stattfinden, welche jedoch so kompakt wie möglich gehalten werden sollte. Die Ergebnisse dieser Besprechungen werden auf den Story-Cards dokumentiert.

Weekly Scrum

Das Weekly Scrum ist eine kurze Besprechung innerhalb des Entwicklerteams, bei dem jedes Mitglied das Team über seinen aktuellen Fortschrittsstatus informiert. Dabei sollen eventuell aufgetretene Probleme aufgedeckt werden. Das Meeting findet wöchentlich, sprich immer montags, statt. Für das Meeting sind maximal 15 Minuten vorgesehen.

Sprint Review

Das Sprint Review ist ein Meeting innerhalb des Projektteams, welches auf jeden Fall am Ende jedes Sprints, in sinnvollen Fällen auch mehrmals innerhalb eines Sprints durchgeführt wird, um die neuesten Ergebnisse vorzustellen und kritisch zu betrachten. Jedoch ist es nicht zwingend erforderlich, dass das gesamte Projektteam Feedback zu jedem einzelnen Ergebnis gibt. Erforderlich ist ein solches Review von Ergebnissen mit mindestens einem Teamkollegen. Im Idealfall sollten jedoch zumindest all jene Teammitglieder mit einbezogen werden, welche direkt oder indirekt von diesem Ergebnis betroffen sind.

Definition of Done (DoD)

Fertige Stories (Arbeitspakete), welche in der Retrospektive vorgestellt wurden und keine Verbesserungsvorschläge mehr erhalten haben werden durch die DoD (Definition of Done) Kriterien überprüft. Erst sobald alle folgenden Kriterien von einem Arbeitspaket erfüllt sind, gilt es als fertig.

* Die Story ist gemäß der teaminternen Vorlage vollständig beschrieben
* Alle Ergebnisse wurden im Projekt-Repository bereitgestellt
* Dokumentation (Projekt- und Produktdokumentation, Sourcecode) sind fertiggestellt
* Ergebnisse wurden besprochen bzw. vorgestellt - Keine Verbesserungen mehr erforderlich

**Projektmanagement Tools**

Für die Planung und Verwaltung von Arbeitspaketen sowie das Management und die Sicherung von Ergebnissen waren für uns zwei Tools essenziell, welche im Folgenden überblicksmäßig beschrieben werden.

Trello

Trello ist eine web-basierte Projektmanagementsoftware, welche zum Planen und Verwalten von Arbeitspaketen eines Projektes entworfen und somit optimal geeignet ist. Hierbei ist es möglich, benannte Listen zu erstellen und diese mit sogenannten Karten zu füllen. Die Karten stellen für uns Stories (Arbeitspakete) dar. Diese Stories können einem Teammitglied zugeordnet, einer oder mehreren Kategorien hinzugefügt, geschätzt sowie mit einer Frist versehen und detailliert beschrieben werden.

Das Format zur Beschreibung solcher Stories war innerhalb des Teams geregelt. Trotz der Orientierung an Scrum wurden die Story Beschreibungen sinnvoll an unsere Bedürfnisse angepasst. Neben dem Ziel und dem Zweck des Arbeitspaketes sowie Akzeptanzkriterien wurde auch eine direkte Verlinkung zum Ergebnis des Arbeitspaketes (Link zu unserer Versionskontrollsoftware) integriert.

Github

Für unser Projekt waren außerdem ein zentrales Datenverwaltungssystem bzw. eine Versionskontrolle für unsere Ergebnisse essenziell. Hierfür haben wir uns für die freie Software Github entschieden.

Die Benennung und Ablage von Ergebnissen am Server war durch unsere intern festgelegten Richtlinien geregelt, was das allgemeine Datenmanagement, vor allem aber das Wiederfinden von Dokumenten erleichtern hat.

Das Github Repository ist unter folgendem Link aufrufbar: <https://github.com/Avokaci/BlockchainAccident.git>

**Best-Practices**

**Chrome Plugins**

Erweiterungen für den Web Browser Google Chrome, welche im Chrome Web Store heruntergeladen werden können. Die von den Erweiterungen beinhalteten Funktionen sind wichtig, um den Scrum-Prozess beim Arbeiten mit Trello richtig umsetzen zu können. Des Weiteren sollen sie zur erleichterten und übersichtlicheren Benutzung beitragen und damit den Management-Prozess optimieren.

Wichtige Plugins

Scrum for Trello

Tool zum Hinzufügen von Scrum Points zu einer Story (Schätzen des Zeitaufwandes)

**Download:**   
<https://chrome.google.com/webstore/detail/scrum-fortrello/jdbcdblgjdpmfninkoogcfpnkjmndgje?utm_source=chrome-app-launcher-info-dialog>

**Verwendung:**   
Ist eine Story geöffnet, so klickt man auf den Titel. Dabei öffnet sich die Auswahl zum Schätzen einer Story. Nach vollenden der Story kann auf die gleiche Weise ausgewählt werden, wie lange man tatsächlich gebraucht hat.

Parent/Child Management

Tool zum Festlegen von Parent-Child Beziehungen zwischen Stories. Damit lassen sich Beziehungen und Abhängigkeiten zwischen Stories beschreiben.

**Download:**   
<https://chrome.google.com/webstore/detail/parentchild-managementfo/flnpbgmiploomjgagfbcjlikpiehclld?utm_source=chrome-app-launcher-info-dialog>

**Verwendung:**   
Ist eine Story geöffnet, so hat man rechts unter "Hinzufügen" die Auswahlen "Parent" sowie "Child". Klickt man eines davon an, so kann man den Namen der Story eingeben, die als Parent oder Child festgelegt werden soll.

Trello Card Numbers

Plugin, welches bewirkt, dass im Titel jeder Trello Karte eine individuelle ID angezeigt wird.

**Download:**  
<https://chrome.google.com/webstore/detail/trello-cardnumbers/kadpkdielickimifpinkknemjdipghaf>

**Verwendung:**   
Sobald das Plugin aktiviert ist, braucht der Benutzer keine weiteren Eingaben mehr zu tätigen. Die jeweilige ID wird automatisch bei jeder Trello Karte angezeigt. Seite 3 von 12 Best Practices 03.04.2018

Optionale Plugins

Trelabels

Tool zum Anpassen der Label Styles. Beispielsweise kann eingestellt werden, dass die Bezeichnung eines Labels immer angezeigt wird.

**Download:**   
<https://chrome.google.com/webstore/detail/trelabels-fortrello/annjdmkbhchmobehkcfilecnlhibedbj?utm_source=chrome-app-launcher-info-dialog>

**Verwendung:**   
Am linken oberen Bildschirmrand ist rechts neben dem Button "Boards" ein Button mit einem Label-Symbol darauf. Klickt man auf diesen Button, so hat man mehrere Auswahlmöglichkeiten, den Stil von Labels zu ändern.

Filter by Trello Label

Funktion zum Filtern der Trello Karten nach einem bestimmten Label

**Download:**   
<https://chrome.google.com/webstore/detail/filter-by-trellolabel/mabebanbpngnkpchhjjaeekopclakabk>

**Verwendung:**   
Rechts oben in der Trello Menüleiste wird durch das Plugin der Button „Show Label“ hinzugefügt. Klick man auf diesen, so bekommt man eine Auswahl der Labels, die von mindestens einer Trello Karte verwendet wird.

Card Counter   
Plugin, welches bewirkt, dass die Anzahl der Trello Karten jeder Liste sowie die Anzahl aller Karten im Trello Board angezeigt wird.

**Download**:   
<https://chrome.google.com/webstore/detail/cardcounter-for-trellowi/jbdmgabkegidhagemkfdogmplgelaoab>

**Verwendung:**   
Sobald das Plugin aktiviert ist, braucht der Benutzer keine weiteren Eingaben mehr zu tätigen. Die Anzahl der Karten einer Liste wird rechts neben sowie direkt unter dem Listentitel angezeigt.

**Trello Listen**

Richtlinien zur Verwendung der Listen in Trello (Meetingroom, Product Backlog, etc.) mit Erklärungen, welche Trello Karten mit welchen Eigenschaften in welcher Liste sein sollen.

Product Backlog

Hier abgelegte Karten müssen bereits soweit zerlegte Arbeiten sein, dass sie als Story beschrieben werden können. Stories ohne Beschreibung, Schätzung oder Zuweisung dürfen in der Liste bestehen, müssen aber hier beschrieben werden.

Sprint Backlog

Stories, die im nächsten Sprint abzuarbeiten sind. Auch Stories, an denen gerade gearbeitet wird sind in dieser Liste. Die Karten müssen beschrieben, geschätzt und zu einem oder mehreren Team-Mitgliedern zugewiesen sein.

Doing

In die Doing Liste kommen Stories aus dem Sprint Backlog, an ddenen gerade gearbeitet wird. Ist eine Story am Ende einer Arbeitssession noch nicht abgearbeitet so bleibt sie bis zu Fertigstellung oder dem Ende des Sprints in dieser Liste.  
In dieser Liste darf nur eine Story pro Teammitglied bestehen (kein paralleles Arbeiten an zwei Stories).

Sprint Review

Hier befinden sich Stories, die vorerst abgearbeitet sind. Das heißt die Akzeptanzkriterien sind alle erfüllt. Die Stories sind damit bereit gemeinsam besprochen und vorgestellt zu werden. Wenn bei einer Diskussion herauskommt, dass die Story komplett fertig ist, kann sie in die Liste „Done“ verschoben werden, wenn noch Fehler oder Lücken auffallen wird eine Childstory im Product Backlog erstellt, die die Probleme beschreibt.

Done (DoD)

Damit eine Story hier hingeschoben werden darf, müssen folgende Kriterien erfüllt sein:

* Die Story ist gemäß der Vorlage vollständig beschrieben
* Alle Akzeptanzkriterien sind erfüllt
* Jegliche Ergebnisse sind hochgeladen
* Dokumentationen sind fertiggestellt und hochgeladen
* Ergebnisse besprochen bzw. vorgestellt
* Keine Verbesserungen mehr erforderlich

**Story erstellen**

Beschreiben einer Story

In der Trello Liste „Best Practices” ist unter anderem eine leere Vorlage für eine Story Beschreibung. Das heißt die auszufüllenden Felder bzw. Tags sind bereits vorgegeben. Wie genau die Felder auszufüllen sind wird im Folgenden beschrieben.

Card

**Kurze Übersicht, durch welche der Inhalt, Nutzen und das Ziel der jeweiligen Story ersichtlich werden soll.**

Role

Stichwort, wer den Endnutzen dieser Funktionalität trägt.

|  |  |
| --- | --- |
| Entwicklerteam | z.B. Dokumente, die Richtlinien zum Arbeiten innerhalb des Teams beinhalten |
| Kunde | z.B. Funktionalitäten, die den Benutzer beim verwenden unseres Produktes unterstützen |

Goal

Kurzer Satz der Antworten auf die Frage „Was ist das Ziel dieser Story?“ gibt. Welches Feature soll durch diese Story bereitgestellt werden?

Conversation

**Freier Raum für Detailbeschreibungen des Problems**

In diesem Bereich gibt es keine vorgegebenen Tags. Hier sollte im benötigten Detaillierungsgrad eine genaue Beschreibung des Problems und den Anforderungen dafür dokumentiert werden.

Außerdem soll hier auch die geplante bzw. durchgeführte Umsetzung dokumentiert werden.

Schlussendlich sollten hier auch noch Probleme, welche eventuell aufgetreten sind, niedergeschrieben und auch ein Lösungsweg für diese Probleme verfasst werden.

Results

**Hier sind jegliche Ergebnisse, welche bei der Abarbeitung der Story entstanden sind zu verlinken.**

Jegliche Ergebnisse der Story müssen auf Github commitet werden. In der Results Beschreibung sollten anschließend diese Ergebnisse verlinkt (nicht angehängt) werden. Falls mehrere Ergebnisse entstanden sind ist auch vor den Tags kurz in Worte zu fassen, welche Ergebnisse es gibt und wo diese zu finden sind.

*Github Link*

Hier wird der Link zum Ergebnis auf Github eingefügt.

*Commit Comment*

Der Commit-Kommentar hat bei jeder User Story im Result zu erfolgen. Dabei muss die für die User Story investierte Zeit angeschrieben sein und noch ein Kommentar vom Benutzer angegeben werden.

**Trello Labels**

**Trello Labels dienen einer schnellen Übersicht, welchem Aufgabenbereich eine Story zuzuordnen ist. Im Folgenden sind die möglichen Labels aufgelistet sowie beschrieben wann diese zu verwenden sind.**

Sie sind auf jeden Fall zu einer Trello Karte hinzuzufügen, bevor diese ins Sprint Backlog verschoben wird. Es können auch mehrere Labels kombiniert werden.

Planing

* Für die Erstellung und Bearbeitung von Dokumenten, die für den Projektmanagement Prozess benötigt werden.

Discussion

* Kennzeichnet, dass bei dieser Story noch ausführlich diskutiert werden muss, um Missverständnisse zu klären.

Alert

* Diese Karte muss so schnell wie möglich fertiggestellt werden.
* Diese Karte beinhaltet wichtige Informationen, die für den Prozess der Story wichtig sind.

Testing

* Testen der Software oder Hardware.

Document

* Dokumentation eines gewissen Themas.

Programming

* Für jegliche Aufgaben bei denen Programmiert wird (C#, C).

Hardware

* Für Arbeiten direkt an der Hardware.

Sprint #..

* Für jeden Sprint wurde ein eigenes Label erstellt. Eine Story bekommt ein bestimmtes Sprint Label, wenn in dem jeweiligen Sprint an dieser Aufgabe gearbeitet wurde. Eine Story kann somit auch mehrere Sprint Labels haben, wenn über mehrere Wochen an dieser Karte gearbeitet wurde.

**Commit-Kommentar**

Der Commit-Kommentar hat bei jeder User Story im Result zu erfolgen. Dabei muss die für die User Story investierte Zeit angeschrieben sein und noch ein Kommentar vom Benutzer angegeben werden.

Comment

Was im Kommentar beschrieben werden muss:

* Welche Features wurden hinzugefügt?
* Welche vorhandenen Features wurden geändert + was wurde an diesen Features geändert?
* Welche begonnenen Features sind noch nicht fertiggestellt?

Duration

Zeit in Stunden, welche für die Aufgabe bis zum Commit-Zeitpunkt benötigt wurde. Diese wird lediglich als Zahl angegeben (NICHT z.B. 2h). Wurde weniger als eine Stunde gebraucht, so wird die Zeit als Gleitkommazahl mit einem Punkt für das Komma angegeben.

#duration außerdem 0 angegeben sein.

**Dokumentenbenennung**

Dokumente sind nach dem folgenden Muster zu benennen:

BA. <Typ>. <Dokumentenbezeichnung>. <Dateiformat>

Typ

**Der Typ gibt einen schnellen Hinweis auf die Art des Dokuments**

Als Typ ist je nach Art des Dokuments eine der folgenden Bezeichnungen in Lower Case zu verwenden:

|  |  |
| --- | --- |
| protocol | Dokumente, die Arbeitsprozesse dokumentieren |
| research | Dokumente, die durch Recherchearbeiten entstehen |
| datasheet | Eigene Datenblätter von Hardware Komponenten |
| literature | Dokumente und Bücher, die zum Erlangen von Fachwissen für das Team gedacht sind |
| planing | Planungsdokumente |
| presentation | Präsentationsdokumente |