

# Scrumiverse Teilnehmer Handbuch

## Marketing

Rel. Userstory ID: US002  
Version: V003

### Inhaltsverzeichnis

<b>1</b>	<b>VERSIONSGESCHICHTE</b>	<b>0</b>
<b>2</b>	<b>TOOLSBASELINE</b>	<b>2</b>
2.1	GITHUB DESKTOP (GUI FÜR GIT)	2
2.2	ACCOUNT AUF DER SEITE GITHUB	2
2.3	ACCOUNT AUF DER SEITE SCRUMDESK	2
2.4	MICROSOFT OFFICE	2
2.5	PROJECTLIBRE (GANTT DIAGRAMME)	2
<b>3</b>	<b>VEREINBARUNGEN</b>	<b>2</b>
3.1	GENERELL	2
3.2	FORMATIERUNGEN	2
<b>4</b>	<b>WORKFLOW</b>	<b>2</b>
4.1	GENERELL	2
4.2	OFFICE DOKUMENTE	3
4.3	ABNAHME VON TASKS ODER USER STORIES	3
<b>5</b>	<b>TAGGING ÜBERSICHT</b>	<b>3</b>
5.1	COMMIT TAGS	3
5.2	SCRUM TAGS	4
<b>6</b>	<b>INFOS ZU GIT</b>	<b>4</b>

## 1 Versionsgeschichte

Version	Datum	Änderungsbeschreibung	Bearbeiter
001	03.01.2016	Dokumenterstellung	Jolitz
002	03.01.2016	Tagging & Workflow erklärt, erste Vereinbarungen und Tools notiert	Jolitz
003	07.01.2016	Toolsbaseline überarbeitet und Abnahme von User Stories & Tasks eingefügt.	Jolitz



## 2 Toolsbaseline

### 2.1 GitHub Desktop (GUI für Git)

**Version:** v3.0.11

**Link:** <https://desktop.github.com/>

**Info:** Versionsverwaltungstool für unser Projekt.

### 2.2 Account auf der Seite GitHub

**Version:** -

**Link:** [https://github.com/Sl0thking/mp\\_anwen\\_scrumiverse](https://github.com/Sl0thking/mp_anwen_scrumiverse)

**Info:** Bereitstellung unseres Repositories, Rechteverwaltung

### 2.3 Account auf der Seite ScrumDesk

**Version:** -

**Link:** <http://app.scrumdesk.com/>

**Info:** Unser Projektplanungstool, enthält die Planung und Aufgaben der Teilnehmer.

### 2.4 Microsoft Office

**Version:** 2010 oder höher

**Link:** [http://www.microsoftstore.com/store/msde/de\\_DE/cat/Office/categoryID.64873300](http://www.microsoftstore.com/store/msde/de_DE/cat/Office/categoryID.64873300)

**Info:** Die Projektinterne Office Software

### 2.5 ProjectLibre (Gantt Diagramme)

**Version:** 1.6

**Link:** <http://www.projectlibre.de/>

**Info:** Software zur Erstellung von Gantt Diagrammen. Notwendig um die aktuelle zeitliche Planung anzusehen (übersichtlicher als ScrumDesk).

## 3 Vereinbarungen

### 3.1 Generell

- Änderungen werden nur dann hochgeladen, wenn diese nicht zu einer Beeinträchtigung des Systems führen. Das heißt der vorhandene Quellcode muss kompilieren und ausführbar sein.
- Nehmt die Zeit Notation sehr ernst. Notiert so genau wie möglich (im Rahmen der Möglichkeiten unserer gewählten Toolsbaseline).

### 3.2 Formatierungen

- Benutzt für Dokumente die Formatvorlagen.

## 4 Workflow

### 4.1 Generell

1. Führt in GitHub Desktop ein Sync durch.
2. Notiert euch eure Start Arbeitszeit an diesem Tag.
3. Führt eure Änderungen durch. Bei Office Dokumenten editiert das Dokument mit euren Namen
4. Errechnet eure Gesamtarbeitszeit für diesen Tag (achtet auf vorherige Arbeitszeiten wenn ihr über mehrere Tage arbeitet).
5. Bewertet die Änderungen, ist ein Commit mit anschließendem Sync sinnvoll oder nicht? (Siehe Vereinbarungen).

Ja: Weiter mit Punkt 6

Nein: Zu Punkt 1, mit Arbeitszeit von diesem Tag

6. Tagged den Commit vernünftig (siehe Tagging Übersicht) und führt ein Sync durch.
7. Notiert eure Zeit auf der ScrumDesk Seite. Geht dazu auf den Reiter Work, sucht die betreffende Task und klickt auf diese. Klickt im Task Fenster auf Channel -> LogWork und gebt dort eure Zeit und dieselbe Commit-Message wie bei Git an. Markiert die betroffene Task entsprechend (In Progress / Done).

## 4.2 Office Dokumente

Office Dokumente sind für Git bei einem Konflikt nicht vergleichbar und so nur schwer mergbar. Deshalb stehen für Tasks, die eine schriftliche Ausarbeitung erfordern und an denen mehr als eine Person arbeitet, mehrere Dokumente bereit mit dem jeweiligen Namen des Projektteilnehmers. Sprecht euch ab wer welchen Teil übernimmt und führt das Workflow wie unter Generell beschrieben durch. Fügt euren Teil am Ende in einem finalen Dokument zusammen, ohne Namen der Teilnehmer. Achtet bitte darauf und versucht Konflikte zu vermeiden.

## 4.3 Abnahme von Tasks oder User Stories

Tasks die im Work Reiter auf Done geschoben werden, sind nicht automatisch „Fertig“. Es fehlt noch die Prüfung vom Projektleiter oder eventuell sogar dem ganzen Team.

Tasks ohne Farbe werden zunächst geprüft. In der Kommentar Sektion wird der Projektleiter als Kommentar #Abnahme schreiben mit einem Status, OK oder NOK. Bei einem NOK wird die Task wieder nach „In progress“ verschoben, Rot eingefärbt und über ein #Kommentar der Grund der gescheiterten Abnahme beschrieben. Der zuständige Mitarbeiter muss nun den Fehler beheben und bei Fertigstellung die Task wieder zur Prüfung freigeben, in dem er den roten Task wieder auf „Done“ verschiebt.

Sobald alle Tasks einer User Story abgenommen sind, wird die User Story selbst Grün eingefärbt, als Signal dass sie nun fertig bearbeitet ist.

# 5 Tagging Übersicht

## 5.1 Commit Tags

### #USID XXX

Betreffende User Stor(ies), muss immer als Titel angegeben werden.

### #Datum 12.12.12, ...

Gibt die Daten an, an denen gearbeitet wurde. Optional wenn der Commit nur einen Arbeitstag betrifft, bei mehreren Arbeitstagen aber notwendig.

### #Zeit 2h 20min

Gibt die Gesamtarbeitszeit an diesem Commit an. Immer notwendig.

### #Info

-Changed something

-...

Gibt die wirklichen Änderungen die durch diesen Commit entstehen an. Ist immer notwendig.

### **#Kommentar**

Bla bla bla...

Optional Tag, kann dazu verwendet werden auf Probleme oder Besonderheiten hinzuweisen.

### **#Abnahme [OK | NOK]**

Weißt in Kommentaren auf die Abnahme der Task oder User Story hin. Ok weist auf eine erfolgreiche Abnahme hin, Not OK auf Probleme.

## 5.2 Scrum Tags

### **#Rechtschreibung**

Bedeutet dass bei der betroffenen User Story auf die Rechtschreibung geachtet wird. Bei mehr als 10 Grammatik- und Rechtschreibfehlern wird die User Story nicht akzeptiert und zurückgewiesen.

Wird nur für User Stories verwendet, die als Ergebnis Dokumente liefert.

### **#Fließtext**

Das Ergebnisdokument muss einen sinnvollen Fließtext ergeben. Wenn dieser Tag fehlt, reichen auch Aufzählungen.

### **#DokPfad**

Hilfestellung, gibt an wo im Repo sich die betreffenden Dokumente befinden.

## 6 Infos zu Git

## Git Data Transport Commands

<http://osteele.com>

