User Stories und Tasks  
Patrice Keusch, Severin Müller

Bemerkungen zum Dokument

* Das Prinzip ist so angedacht, dass die folgenden Seiten jeweils Vorder und Rückseite darstellen. D.h. auf der 1. folgenden Seite ist die User Story dargestellt, auf der nächsten Seite dann die Tasks
* Es sind nur diejenigen User Stories mit Tasks ausgearbeitet, welche effektiv in der Fallstudie innerhalb des Studiums, Fach „Methoden der Programmierung“, realisiert werden.
* Die Velocity wurde in der ersten Iteration mit 0.6 angenommen (das Lehrbuch sieht eine Velocity von 0.7 vor für die erste Schätzung, wir möchten hier aber auf der sicheren Seite sein). Für die folgenden Iterationen wird dann die effektive Velocity der Iteration 1 herangezogen (“Every day predict that tomorrow's weather will be exactly the same as today's” – Zitat aus Buch “Planning Extreme Programming”, Seite 19).
* In der Phase 1 wird daher für jede User Story und jeder Task die Velocity von 0.6 angelegt, erst danach wird genauer kalkuliert auf Basis der Erfahrungen von Iteration 1
* Die Velocity wird für jeden Task und jede User Story angegeben. Zuerst steht der geschätzte Wert, dann ein 🡺, dann die Velocity

|  |  |
| --- | --- |
| **Hintergrundgrafik** | Est: 2.5 Tage 🡺 4 Tage |
| Die Hintergrundgrafik soll einen Rasen mit einem sonnigen Hintergrund mit einigen Wolken zeigen. | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task6 Implementieren der Start-Seite gemäss Mock-Up | | Task7 Implementieren der Detailansicht gemäss Mock-Up | |
| Zust: OA | Est: 5h 🡺 1 Tag | Zust: OA | Est: 10h 🡺 2 Tage |

|  |  |  |  |
| --- | --- | --- | --- |
| Task8 Testen der Start-Seite (Dummy-Links, Visuelle Darstellung) | | Task9 Testen der Detailansicht (Dummy-Links, Visuelle Darstellung) | |
| Zust: SM | Est: 2.5 h 🡺 0.5 Tage | Zust: SM | Est: 2.5h 🡺 0.5Tage |

|  |  |
| --- | --- |
| **Fallschirmspringer** | Est: 8 h 🡺 1 Tag |
| Der Fallschirmspringer soll einfach gehalten werden, d.h. ein Strichmännchen genügt. | |
| Prio: 1 | |

|  |  |
| --- | --- |
| Task4 Mock-Ups der Start-Seite erstellen | |
| Zust: PK | Est: 2.5 h 🡺 0.5d |

|  |  |
| --- | --- |
| Task5 Mock-Ups der Detailansicht erstellen | |
| Zust: PK | Est: 2.5 h 🡺 0.5d |

|  |  |
| --- | --- |
| **Effekte: Wind** | Est: 10h 🡺2 Tage |
| Mann soll die Windstärke – und Richtung angeben können. | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task15 Suchfeld in Startseite integrieren (inkl. Button) | | Task16 Datenbank-Query realisieren | |
| Zust: PK | Est: 2.5h 🡺 0.5 d | Zust: SM | Est: 2.5h 🡺 0.5 d |

|  |  |  |  |
| --- | --- | --- | --- |
| Task17 Darstellung der Resultateliste realisieren | | Task18 Testen der Suchfunktion (Verschiedene Dummy-Filme sind implementiert vom Lehrling) | |
| Zust: PK | Est: 5h 🡺 1 d | Zust: RK | Est: 2.5h 🡺 0.5 d |

|  |  |
| --- | --- |
| **Effekte: Gravitation** | Est: 10h 🡺2 Tage |
| Die Erdgravitation muss berücksichtigt werden. | |
| Prio: 2 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task22 Filesystem in Kategorien umwandeln (auslesen und in Baum Parsen) | | Task23 Implementation des Baums in Startseite | |
| Zust: RK | Est: 5h 🡺 1d | Zust: PK | Est: 2.5h 🡺 0.5 d |

|  |  |
| --- | --- |
| Task24 Testen der korrekten Abbildung des Filesystems | |
| Zust: SM | Est: 2.5h 🡺 0.5d |

|  |  |
| --- | --- |
| **Funktionalität: Schirm** | Est: 1 Tage |
| Der Benutzer muss durch einen Mausklick den Fallschirm öffnen können. | |
| Prio: 3 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task22 Filesystem in Kategorien umwandeln (auslesen und in Baum Parsen) | | Task23 Implementation des Baums in Startseite | |
| Zust: RK | Est: 5h 🡺 1d | Zust: PK | Est: 2.5h 🡺 0.5 d |

|  |  |
| --- | --- |
| Task24 Testen der korrekten Abbildung des Filesystems | |
| Zust: SM | Est: 2.5h 🡺 0.5d |

|  |  |  |
| --- | --- | --- |
|  | |  |
| **Funktionalität: Landung** | Est: 1 Tage | |
| Der Zeitpunkt der Schirmöffnung muss für die Landung berücksichtigt werden (Absturz oder sanfte Landung) | | |
| Prio: 3 | | |

|  |  |
| --- | --- |
| **Detailansicht: Titelbild\*** | Est: 3 Tage |
| Dem Benutzer soll bei jedem Film ein Titelbild des Films angezeigt werden (Titelbild) | |
| Prio: 2 | |

|  |  |
| --- | --- |
| **Detailansicht: Titelbild selbst geladen** | Est: 2 Tage |
| Der User kann ein eigenes Titelbild in die Detailansicht laden | |
| Prio: 3 | |

|  |  |
| --- | --- |
| **Detailansicht: Titelbild aus Film generiert** | Est: 6 Tage |
| Wenn der User kein eigenes Titelbild in die Detailansicht lädt, soll ein Titelbild aus dem Film automatisch generiert werden. (Est vorsichtig geschätzt, da Know-How noch nicht vorhanden) | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Detailansicht: Beschreibung** | Est: 5h 🡺 1 Tag |
| Der Benutzer soll eine Beschreibung des Filminhalts sehen können (Text) | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task19 Beschreibungs-Ausgabe implementieren | | Task20 Beschreibung aus DB holen (Datenbank query erstellen | |
| Zust: OA | Est:2.5 h🡺 0.5 d | Zust: SM | Est: 2.5 h🡺 0.5 d |

|  |  |
| --- | --- |
| Task21 Testen der Beschreibungs-Anzeige | |
| Zust: PK | Est: <2.5 h 🡺 <0.5d |

|  |  |
| --- | --- |
| **Detailansicht: Bewertung ansehen** | Est: 1 Tag |
| Der Benutzer soll für jeden Film eine Bewertung ansehen können. | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Detailansicht: Bewertung abgeben** | Est: 2 Tage |
| Der User soll zu jedem Film eine Bewertung abgeben können (1 Bewertung pro User) | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Detailansicht: Bewertung ändern** | Est: 1 Tag |
| Der User soll seine abgegebene Wertung nachträglich ändern können. | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: User-Login** | Est: 3 Tage |
| Der Benutzer soll nach dem Klick auf „Login“ auf diese Login-Seite kommen, um sich anzumelden (danach sind die Editierungsfunktionen in den Ansichts-Seiten verfügbar) | |
| Prio: 3 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Neue Filme hinzufügen** | Est: 3 Tage 🡺5 Tage |
| Der Benutzer (Verwaltung) soll einen neuen Film hinzufügen können über ein Dateiverwaltungstool (Filme liegen auf Server) | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task10 Design für Eingabemaske erstellen und implementieren | | Task12 Zusammenbringen der Eingabemaske mit den Datenbank-verknüpfungen (Speicher-mechanismus) | |
| Zust: OA | Est: 5h 🡺 1 d | Zust: RK | Est: 10h 🡺 2d |

|  |  |  |  |
| --- | --- | --- | --- |
| Task13 Verknüpfen des Dateimanagers mit Eingabemaske | | Task14 Testen: Erfassen eines Mediums (oder mehrerer) | |
| Zust: RK | Est: 5h 🡺 1 d | Zust: PK | Est: 2.5h 🡺 0.5d |

|  |  |
| --- | --- |
| **Verwaltungsseite: Bestehenden Film bearbeiten** | Est: 3 Tage |
| Der Benutzer soll in jedem Feld, welches auf der Ansichts-Seite verfügbar ist, eine Editiermöglichkeit haben (z.B. über ein kleines „Bearbeitungs-Symbol“ Rechts oben, definitive Platzierung muss noch besprochen werden, Zeitproblem) | |
| Prio: 3 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Film aus Tool löschen** | Est: 2 Tage |
| Der User soll über einen Link den Film einfach aus dem Tool löschen können | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Film aus Tool und Filesystem löschen** | Est: 1 Tage |
| Der Benutzer soll über einen Link den Film einfach aus dem Filesystem löschen können (Wird daher auch nicht mehr in der Filmdatenbank gehalten) | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Ansichts-Seite: Film mittels VLC abspielen** | Est: 1 Tag |
| Der Benutzer soll über einen Link den Film über VLC abspielen können. | |
| Prio: 2 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Userverwaltung** | Est: 3 Tage |
| Der Benutzer soll über ein einfaches GUI Benutzer verwalten können (anlegen, löschen, Passwörter, Zugriffsstufen etc.) | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Film wurde verschoben** **(Anzeige)** | Est: 4 Tage |
| Der Benutzer soll beim Login in die Verwaltung mit einem einfachen Fenster darauf hingewiesen werden, dass der Film nicht mehr am Ursprünglichen Ort liegt. | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Film wurde aus Filesystem gelöscht (Anzeige)** | Est: 2 Tage |
| Der Benutzer soll beim Login in die Verwaltung mit einem einfachen Fenster darauf hingewiesen werden, dass der Film im Filesystem nicht mehr vorhanden ist. | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Behandeln von gelöschten Filmen** | Est: 1 Tag |
| Der Benutzer soll auf gelöschte Filme reagieren können (löschen aus Filmdatenbank mittels einfachem Klick) | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Verwaltungsseite: Behandeln von verschobenen Filmen** | Est: 2 Tage |
| Der Benutzer soll akkurat auf verschobene Filme reagieren können (mittels Klick über Dateiverwaltungstool den Dateipfad neu setzen) | |
| Prio: 4 | |

|  |  |
| --- | --- |
| **Datenbankdesign: Medienverwaltung** | Est: 2 Tage 🡺  3 Tage |
| Alle Medien werden in einer DB verwaltet, welche alle relevanten Daten der Filme beinhaltet (Name, Dauer, Bewertung, …), Kategorien werden aus dem Filesystem generiert | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task1 Datenbank-Schema für die Medienverwaltung designen | | Task2 Implementation des Schemas | |
| Zust: RK | Est: 5h 🡺 1d | Zust: SM | Est: 2.5h 🡺 0.5d |

|  |  |  |  |
| --- | --- | --- | --- |
| Task3 Tests der Implementation (Review) | | Task11 Datenbank-verknüpfungen erstellen (MySQL Parolen fürs einfügen in DB) | |
| Zust: RK | Est: 2.5h 🡺 0.5d | Zust: SM | Est: 5h 🡺 1d |

|  |  |
| --- | --- |
| **Datenbankdesign: Benutzerverwaltung** | Est: 4 Tage |
| Alle Benutzer werden in einer DB verwaltet, diese beinhaltet (Login-Name, Passwort [Cipher], Rechte, Film-Bewertungen) | |
| Prio: 4 | |