

Effort Driven Scheduling

Testplan

Version	Projektname	Autor	Status	Datum	Kommentar
0.1	Effort Driven Scheduling	Tamás Karli, Carlos Mojental	in Bearbeitung	10.12.2017	

1. Einführung

Das Ziel des Projekts ist es die Ganttproject-Software um ein Effort Driven Scheduling Feature zu erweitern. Dieses soll es Usern erlauben ihre Tasks resourcesabhängig zu definieren und ihren Projektplan dynamisch zu gestalten.

1.1 Zweck

Dieses Dokument dient zur Referenz für die Projektleitung. Es soll einen Überblick über das geplante Testverfahren bieten.

1.2 Beziehung zu anderen Dokumenten

Dieser Testplan beruht auf den Anforderungen welche im Lasten- und Pflichtenheft dieses Projekts definiert sind.

2. Systemübersicht

Das Feature ist vollständig in die bestehende Code-Basis des Ganttprojects eingebettet. Die neue Funktionalität fügt sich in die bestehenden Komponenten wie Tasks und Ressourcen ein und erweitert die Benutzeroberfläche um die nötigen Felder.

3. Merkmale

3.1 Zu testende Merkmale (Features / Funktionen)

3.1.1 Funktionale Anforderungen

/F10/, /F20/, /F30/ und /F40/ des Pflichtenhefts werden getestet.

3.2 Nicht zu testende Merkmale (Features / Funktionen)

Für die optionalen Erweiterungen des Features, nämlich /F50/, /F60/ und /F70/, sind keine Tests vorgesehen.

4. Vorgehensweise

4.1 Komponenten und Integrationstests

Geeignete Funktionen werden durch Unit Tests getestet, alle anderen manuell im GUI geprüft.

4.2 Funktionstest

Falls es die zeitlichen Umstände erlauben, werden für die Funktionen /F10/, /F20/ und /F30/ Unit Tests geschrieben um zu prüfen, ob die vom User eingegebenen Werte korrekt abgebildet werden. /F40/ wird direkt anhand des Outputs auf dem GUI getestet. Dieser Test dient ebenfalls zur oberflächlichen Prüfung der anderen zu testenden Anforderungen, da die korrekte Darstellung in der Übersichtstabelle nur durch deren Funktionalität möglich ist.

5. Hardware und Softwareanforderungen

Ausser einer lauffähigen Version der Ganttproject-Software mit dem zu integrierenden Feature wird keine spezielle Hardware oder Software benötigt.

6. Testfälle

6.1 Modultests

Testfall: /TM10/

Testziel: Überprüfung der korrekten Darstellung der Taskdauer in der Übersichtstabelle bei Änderung der relevanten Parameter.

Voraussetzung: Existenz eines dynamischen Tasks mit workload und verantwortlichen Ressourcen mit deklariertem Pensum.

Eingabe: Änderung einer oder mehrerer Parameter die einen Einfluss auf die Dauer des Tasks haben.

Erwartete Ausgabe: Anzeige der neu berechneten Dauer in der Übersichtstabelle.

Abhängigkeiten: Setzt die Funktionalität aller anderen spezifizierten Anforderungen voraus.

6.2 Funktionstests

Testfall: /TF10/

Testziel: Überprüfung der korrekten Verarbeitung bei Deklaration eines dynamischen Tasks.

Voraussetzung: Existenz eines bestehenden Tasks der zu einem dynamischen Task gemacht werden kann.

Eingabe: Nicht dynamischer Task.

Erwartete Ausgabe: Task welcher als dynamisch deklariert ist und einen Standardwert für seine workload enthält.

Abhängigkeiten: keine.

Testfall: /TF20/

Testziel: Überprüfung der korrekten Verarbeitung bei Änderung der workload eines dynamischen Tasks.

Voraussetzung: Existenz eines dynamischen Tasks.

Eingabe: Dynamischer Task und ein Wert zu welchem dessen workload geändert werden soll.

Erwartete Ausgabe: Task mit dem gegebenen Wert als workload, falls die Eingabe gültig war.

Abhängigkeiten: keine.

Testfall: /TF30/

Testziel: Überprüfung der korrekten Verarbeitung bei der Änderung des Pensums einer Ressource.

Voraussetzung: Existenz einer Ressource.

Eingabe: Ressource und ein Wert zu welchem dessen Pensum geändert werden soll.

Erwartete Ausgabe: Ressource mit dem gegebenen Wert als Pensum, falls die Eingabe gültig war.

Abhängigkeiten: keine.