

Effort Driven Scheduling

Pflichtenheft

(Vorlage nach Balzert, Lehrbuch der Softwaretechnik (3 Auflage, Kapitel 20.3))

Das Pflichtenheft ist eine Verfeinerung und Präzisierung des Lastenhefts.

Version	Autor	Quelle	Status	Datum	Kommentar
0.1	Tamás Karli, Carlos Mojental, Projektleiter in Bearbeitung 21.11.2017				

1. Visionen und Ziele

Beschreibung der Visionen und Ziele, die durch den Einsatz des Systems erreicht werden sollen.

- /V10/ Nutzern der Ganttproject-Software soll es ermöglicht werden, die benötigte Zeit für Tasks abhängig von den vorhandenen Ressourcen zu definieren.
- /Z10/ Beim Erstellen eines Tasks hat der Nutzer die Option den Task von den vorhandenen Ressourcen abhängig zu machen. Die für einen Task benötigte Zeit soll sich anpassen wenn sich die dafür verantwortlichen Ressourcen verändern.

2. Rahmenbedingungen

Beschreibung der organisatorischen Rahmenbedingungen: Anwendungsbereiche, Zielgruppen, Betriebsbedingungen (Wenn nicht relevant entsprechende Beispielskategorie löschen)

- /R10/ Ganttproject ist eine Open-Source Software die das Zeitmanagement von Projekten erleichtern soll.
- /R20/ Die geplante, zusätzliche Funktionalität richtet sich an Nutzer, die sich eine funktionale Verknüpfung zwischen Ressourcen und Tasks wünschen. Dies ist beispielsweise interessant für Projekte bei welchen sich die Ressourcen häufig ändern oder die Zeitabschätzung in Arbeitsstunden einfacher ist als in absoluten Zeitabschnitten.
- /R50/ Jegliche Funktionalität des Systems erfordert eine Initiative des Nutzers, es ist kein unbeaufsichtigter Betrieb vorgesehen.
- /R60/ Das zu entwickelnde Feature ist komplett in den bestehenden Ganttproject-Code eingebettet und erfordert somit lediglich die Ganttproject-Software selbst.
- /R90/ Die Software wird mithilfe der IntelliJ Community Edition IDE entwickelt.
- /R110/ Das Feature sollte genügend intuitiv bedienbar sein um ohne zusätzliche Materialien verwendet werden zu können.

3. Kontext und Überblick

Festlegung der relevanten Systemumgebung (Kontext) und Überblick über das System:

- /K10/ Das Effort Driven Scheduling fügt sich in den bestehenden Rahmen des Task-Systems ein. Es wird lediglich eine weitere Option beim Erstellen bzw. Bearbeiten eines Tasks hinzugefügt.

4. funktionale Anforderungen

Die Kernfunktionalität des Systems ist aus Auftraggebersicht auf oberster Abstraktionsebene zu beschreiben. Auf Detailbeschreibung ist zu verzichten.

- /F10/ Das Feature muss es Nutzern erlauben ein Taskaufwand als fixen Zeitraum oder ressourcenabhängig zu definieren. - Carlos
- /F20/ Das Feature muss es erlauben einem ressourcenabhängigen Task den Aufwand in Arbeitsstunden zuzuordnen. - Tompi
- /F30/ Das Feature muss es erlauben einer Ressource ein tägliches Arbeitspensum in Stunden zuzuordnen. - Tompi
- /F40/ Das Feature muss die Dauer eines ressourcenabhängigen Tasks anhand der relevanten Variablen berechnen können. Tompi und Carlos
- /F40A/ Die relevanten Variablen sind die zugeteilten Ressourcen, deren Pensum und deren prozentualer Zeitaufwand für diesen Task.
- /F40B/ Das Ergebnis muss in der Übersichtstabelle dargestellt werden können.
- /F50/ Das Feature soll die Ferientage der Ressourcen berücksichtigen. - Carlos
- /F60/ Das Feature soll den Nutzer bei Allokation von Ressourcen über 100% warnen. - Tompi
- /F70/ Das Feature soll Tasks mit Abhängigkeiten bei Änderungen dynamisch anpassen. Das heisst Start- und Enddaten so verändern, dass die chronologische Logik bestehen bleibt. - Carlos
- /F80/ Das System sollte eine Warnung schicken, dass der Zeitaufwand für das Projekt geändert werden muss, falls der Benutzer die Ressourcen während des laufenden Projekts anpassen möchte - Tompi
- /F90/ Das System sollte die vorgenommenen Einstellungen speichern damit diese bei einem Software-Neustart nicht verloren gehen. Carlos und Tompi

Wenn bereits möglich, sind die funktionalen Anforderungen auch Statik, Dynamik und Logik zu gruppieren. Erfolgt die Spezifikation natürlichsprachlich, dann sollten Sprachtemplates verwendet werden.

5. Qualitätsanforderungen

Es sollte anhand einer Tabelle eine Qualitätszielbestimmung für das Systems vorgenommen werden.

Systemqualität sehr gut gut normal nicht relevant

Funktionalität	x	
Zuverlässigkeit		x
Benutzbarkeit	x	
Effizienz		x
Wartbarkeit		x
Portabilität		x

Einzelne Anforderungen können wie folgt festgelegt werden:

- /QB10/ Das Feature sollte sich logisch in das bestehende Funktionsweise des Programms einfügen und von Nutzern somit ohne explizite Instruktion verwendet werden können.
- /QF10/ Das Feature sollte mit möglichst allen bestehenden Funktionalitäten kompatibel sein damit das Effort Driven Scheduling eine brauchbare Alternative zur bisherigen Zeitplanung darstellt.

6. Abnahmekriterien

Abnahmekriterien legen fest, wie Anforderungen bei der Abnahme auf ihre Realisierung überprüft werden können.

- /A10/ Ein Task muss als dynamic markiert und ihm eine workload zugeteilt werden können.

- /A20/ Einer Ressource muss ein tägliches Arbeitspensum zugeteilt werden können.
- /A30/ Das Programm muss aus workload, dem Pensum und dem prozentualen Zeitaufwand der Ressourcen die Dauer des Tasks berechnen und diese in der Übersichtstabelle darstellen.

7. Arbeitsteilung

Wir müssen uns beide erst in den Code einlesen und ein Verständnis für dessen Funktionsweise gewinnen bevor wir mit der Umsetzung beginnen und Aufgaben verteilen können.