

# Effort Driven Scheduling

## Technische Dokumentation

Version	Projektname	Autor	Status	Datum	Kommentar
0.1	Effort Driven Scheduling	Tamás Karli, Carlos Mojentale	in Bearbeitung	19.12.2017	

## 1 Einführung

### 1.1 Zweck

Dieses Dokument soll der Projektleitung als Überblick über die wichtigsten technischen Eigenschaften des Features dienen.

### 1.2 Beziehung zu anderen Dokumenten

Neben diesem Dokument existieren Lastenheft, Pflichtenheft, Glossar und Testplan bezüglich des vorliegenden Projekts.

## 2. Systemübersicht

Das Feature ist eingebettet in die existierende Ganttproject-Software. Um die gewünschte Funktionalität "Effort Driven Scheduling" zu realisieren wurden einige Schlüsselemente der Software erweitert oder abgeändert. Die Dauer von Tasks kann nun entweder wie gehabt statisch oder neu auch ressourcenabhängig definiert werden. Die dafür nötigen Angaben können in neuen Eingabefeldern in Task- und Ressourceneigenschaften gemacht werden.

## 3. Designziele

- Das Design soll sich möglichst unscheinbar in die bestehende grafische Oberfläche einfügen.

- Das Design soll maximale Stabilität gewährleisten.
- Das Design soll die bestehende Funktionalität nicht beeinträchtigen.

## 4. Systemverhalten

Das Feature soll es erlauben Task-Dauer ressourcenabhängig zu definieren. Dazu wird einem Task eine workload und allen Ressourcen ein Pensum zugeordnet. Das Feature berechnet aus diesen Angaben bei dynamischen Tasks nun die Dauer und stellt diese in der Übersichtstabelle dar.

## 5. Logische Sicht

Der grösste Teil des Systems ist die Erweiterung der Task Klasse mit den Variablen *workload* und *isDynamic*. Der Integer *workload* wird intern als Objekt der Klasse *TimeDuration* verwaltet und *isDynamic* ist ein Boolean. Diese Werte können im "Task Properties"-Fenster für jeden Task angepasst werden. Die beiden Variablen besitzen beinahe alle Methoden, welche die anderen, bestehenden Task-Feldvariablen wie *name*, *startDate* oder *duration* bereits hatten. Dies sorgt dafür, dass die meisten Operationen und Manipulationen an Task Objekten wie bisher funktionieren.

Analog zum Task erhält das HumanResource Objekt die Variable *pensum*, ebenfalls mit allen zugehörigen Methoden.

Diese beiden Teilsysteme werden in der TaskImpl Klasse zusammengeführt. Dort wurden in der Ausgangssoftware die Enddaten für die Darstellung in der Übersichtstabelle aus *startDate* und *duration* berechnet. An diesem Punkt wird neu eine Fallunterscheidung zwischen statischen und dynamischen Tasks gemacht. Für dynamische Tasks werden die Enddaten wie folgt berechnet: Es werden zuerst alle Pensum der dem betreffenden Task zugeordneten Ressourcen mit deren jeweiligen prozentualen Zeitaufwand für den Task gewichtet. Danach wird die *workload* durch die aufsummierten Pensum geteilt. Das Resultat wird zur nächsten ganzen Zahl aufgerundet und man erhält die Dauer des Tasks in Tagen. Dieser Wert wird nun wie gehabt vom Programm weiterverarbeitet und korrekt in der Timeline dargestellt.

## 6. Deployment und Installation

Ausser der Ganttproject-Software selbst sind keine weiteren Installationen nötig.

## 7. Weitere Hinweise

- Lessons learned. Was haben sie in diesem Projekt gelernt. Was würden sie jetzt anders machen?

- Dies war unser erster Kontakt mit einem Program dieser Komplexität. Zu Beginn hatten wir Schwierigkeiten uns zurechtzufinden, aber nach Abschluss dieses Projekt würden wir uns mit mehr Selbstvertrauen an eine vergleichbar Aufgabe wagen.
- Das Program zu verstehen hat uns mehr Zeit gekostet als erwartet. Wir sind uns dadurch der Wichtigkeit von guten Kommentaren bewusst geworden, diese hätten uns das Einlesen bedeutend erleichtert.
- Selbsteinschätzung: Wie sehen sie die Qualität des Projekts? Kann das von ihnen implementierte Feature in die Offizielle Kursversion vom Ganttproject integriert werden? Könnte es vielleicht sogar in die offizielle Version integriert werden?
- Grundsätzlich ist unser Feature qualitativ hochwertiger als wir erwartet haben. Jedoch reicht es nicht für einen release auf die offizielle Version des Ganttprojects. Es existieren einige uns bekannte und gravierende Lücken: Beispielsweise werden die neuen Variablen nicht abgespeichert und können somit nicht aus einem erstellten File ausgelesen werden. Zudem ist das Feature kaum auf Randfälle und jegliche Eventualitäten getestet, es ist also gut möglich dass bei normaler Benutzung Inkonsistenzen auftreten können.