

Alexander Blehm - 2712024

Simon Lehmann - 2761826

Oleg Mryga - 2690599

# **Projektplan**

## **Kollisions Informations Werkzeug**

### **(KIWe)**

Version: 1.02

Verfasser: Alexander Blehm, Simon Lehmann, Oleg Mryga

erstellt am: -24.10.2012-

letzte Änderung: -31.10.2012-

# Inhaltsverzeichnis

- 1. Einleitung**
  - 1.1 Zweck und Nutzen des Projektplans
  - 1.2 Projektbeschreibung "KIWe"
  - 1.3 Motivation
- 2. Formale Grundlagen**
  - 2.1 Anforderungen an die Projektdurchführung
  - 2.2 Anforderungen an das Produkt
- 3. Leistungen**
  - 3.1 Lieferumfang, Arbeitsumfang
  - 3.2 Externe Meilensteine
  - 3.3 Leistungen des Auftraggebers
- 4. Entwicklungsprozess**
  - 4.1 Phasen der Entwicklung
  - 4.2 Review und Testtermine
- 5. Risiken**
  - 5.1 Risiken und Risikobewertung
  - 5.2 Risikoreduktion und Gegenmaßnahmen
- 6. Richtlinien für die Entwicklung**
  - 6.1 Design- und Programmierrichtlinien
  - 6.2 Einsatz von Werkzeugen
- 7. Schlüsselpersonen des Projekts**
- 8. Entwicklungsplan**
  - 8.1 Gantt-Diagramm
  - 8.2 Termindrift-Diagramm
- 9. Versionshistorie**

# 1. Einleitung

## 1.1 Zweck und Nutzen des Projektplans

Der Hauptzweck des Projektplans ist die Dokumentation des gesamten Projektes. Die dokumentierten Teilaspekte des Projektes sind unter anderem: Die festgelegten Projektanforderungen (durch den Kunden), eine übersichtliche und realistische Zeitplanung und alle weiteren, für dieses Projekt relevanten, Nebenbedingungen.

Dieses Dokument wird im Hinblick auf mögliche, eintretende Risiken und ebenso mögliche Zeitabweichungen aktuell gehalten und erweitert werden.

## 1.2 Projektbeschreibung "KIWe"

"KIWe" steht für ein Kollisions- und Informationswerkzeug, das zur Berechnung von möglichen Kollisionen mit Objekten im Weltraum (im Orbit der Erde) dienen soll. Es ist vorgesehen, dass es automatisch, vor jeder Berechnung, die aktuellsten Daten aus Onlinequellen bezieht, diese mit Hilfe von Algorithmen anhand von bestimmten Kriterien auswertet, und schließlich grafisch und tabellarisch ausgibt, so wie in einer Datenbank speichert.

## 1.3 Motivation

Das Projekt soll den kompletten Ablauf eines Softwareprojektes simulieren. Eine hohe Qualität soll durch genaue Dokumentation und ständige Kontrolle gesichert sein.

Die Beiträge für eine hohe Qualität sind:

- dieser Projektplan, der laufend aktualisiert wird
- geplante Reviews
- ein sinnvoll kommentierter Quellcode
- eine Projektspezifikation
- eine ausführliche Anforderungsliste

## **2. Formale Grundlagen**

### **2.1 Anforderungen an die Projektdurchführung**

Das Projekt soll konsistent in einer Sprache (Deutsch oder Englisch) aufgebaut sein. Es soll bei Benutzung externer Plugins besonders auf eine Kompatibilität mit der Lizenz geachtet werden.

### **2.2 Anforderungen an das Produkt**

Das Produkt soll mit Hilfe einer grafischen Benutzeroberfläche Kollisionen mit Weltraumobjekten berechnen können. Die Quellen für die Berechnung sind durch TLE Datensätze aus dem Internet gegeben. Der Berechnungsalgorithmus wird vom Kunden bereitgestellt.

Es soll zudem jede Handlung im Projekt ausführlich dokumentiert werden und mitgeliefert werden.

Für weitere und detailliertere Anforderungen wird an dieser Stelle auf die Spezifikation verwiesen.

## **3. Leistungen**

### **3.1 Lieferumfang, Arbeitsumfang**

Im Lieferumfang sollen enthalten sein, eine CD die folgendes enthält:

- Das ausführbare, fertige Programm
- Den Quellcode
- Die Dokumentation zum Produkt (als .pdf und .doc (Word) Datei)

Der Arbeitsumfang ist im folgenden durch externe Meilensteine definiert und als "Umfang: " bezeichnet.

### **3.2 Externe Meilensteine**

M2 (29.10.2012): Das Release einer ersten Version des Projektplans. Zudem soll eine Anforderungsliste abgegeben werden und es soll allgemein ein "Usability"-Ingenieur bestimmt werden.  
[Umfang: Analyse und Projektplan] - (Verantwortlicher: Oleg Mryga)

- M3 (15.11.2012): Es soll eine erste Spezifikation erscheinen und auch eine Vorlage für das User-Interface (Prototyp). Die Spezifikation soll unter anderem alle Use-Cases des Systems und ein Begriffslexikon enthalten.  
[Umfang: Spezifikation, User-Interface Prototyp] -  
(Verantwortlicher: Alexander Blehm)
- M4 (29.11.2012): Es muss eine korrigierte Spezifikation abgegeben werden (Korrektur soll vor allem auf Grund der Spezifikationsreviews erfolgen). Zudem soll eine Zeitabrechnung abgegeben werden, die einen Zwischenstand des Projektfortschritts beschreiben soll.  
[Umfang: Review, korr. Spezifikation] -  
(Verantwortlicher: Simon Lehmann)
- M5 (13.12.2012): Abgabe eines Entwurfs mit anschließender Präsentation. Der Entwurf soll mit einer Übersichtssgrafik die geplante Architektur beschreiben.  
[Umfang: Entwurf] - (Verantwortlicher: Oleg Mryga)
- M6 (31.12.2012): Es soll ein Zwischenstand der Implementierung abgegeben werden (Alpha Version). Ein Systemtestplan sollte auch dabei sein, der vor allem alle Use-Cases der Spezifikation prüfen soll.  
[Umfang: Implementierung (Alpha), Systemtests planen] -  
(Verantwortlicher: Alexander Blehm)
- M7 (10.01.2013): Hier soll eine lauffähige Implementierung (Beta) abgegeben werden. Dazu gehören auch fertige Modultests.  
[Umfang: Implementierung (Beta), Modultests] -  
(Verantwortlicher: Simon Lehmann)
- M8 (24.01.2013): Eine ausführbare JAR-Datei vom Programm soll abgegeben werden. Es gilt, dass hier alle bekannten Fehler behoben werden sollen und das Programm einwandfrei lauffähig sein sollte ("prinzipiell fertig").  
[Umfang: Implementierung als JAR, lauffähig, Integration der Komponenten] - (Verantwortlicher: Oleg Mryga)
- M9 (31.01.2013): Abnahme durch den Kunden. Das Fertige Programm wird auf einer CD abgegeben. Es folgt ein Abnahmeszenario und eine kurze Befragung zum Projekt.  
[Umfang: Abnahme (als Test), CD erstellen] -  
(Verantwortlicher: Alexander Blehm)

### 3.3 Leistungen des Auftraggebers

Der Auftraggeber stellt das Berechnungsprogramm und einen Server, den er selbst administrieren wird. Zudem wurde ein Skript gestellt, das den bisherigen Ablauf des Prozesses betrieben hat.

## 4. Entwicklungsprozess

### 4.1 Phasen der Entwicklung

Aus dem Aufbau der Meilensteine, ergeben sich folgende Phasen, die mit der Angabe der Meilensteine notiert werden:

- M1 bis M2: Analyse (grobe Anforderungsliste)
- M2 bis M4: Spezifikation (und dazugehörige Reviews)
- M4 bis M5: Entwurf
- M5 bis M8: Codierung und Tests
- M5 bis M6: Systemtests (Grundlage ist ein fertiger Systemtestplan)
- M6 bis M7: Modultests
- M7 bis M8: Integration
- M9: Abnahme durch den Kunden (Abnahmeszenario, kurze Befragung)

Man beachte, dass die Planung vorsieht diverse Tests neben der Codierung zeitlich ablaufen zu lassen (siehe Gesamtphase M5 bis M8).

### 4.2 Review und Testtermine

Die folgenden Termine sind festgelegt und werden bei Änderungen aktuell gehalten:

- *21.11.2012 bis 23.11.2012* - In dem Zeitraum sollen Spezifikationsreviews abgehalten werden. (Genauere Angaben folgen mit der Zeit)
- *bis 31.12.2012* - Bis zu diesem Datum soll ein Systemtestplan festgelegt sein. Getestet wird mit Unit Tests.
- *bis 10.01.2013* - In diesem Zeitraum sollen Modultests fertiggestellt sein.
- *bis 24.01.2012* - Abgabe eines fertigen Systemtestprotokolls.
- *31.01.2012* - Abnahme durch den Kunden. Befragung durch den Kunden.

## 5. Risiken

### 5.1 Risiken und Risikobewertung

- 1.) Ein Risiko besteht darin, dass der erste Implementierungsfortschritt bis zu den Weihnachtsfeiertagen nicht vorangetrieben werden könnte. (Da die Vorlesungszeit bis knapp an die Feiertage reicht). Schaden und Eintrittswahrscheinlichkeit halten sich hier noch in Grenzen.
- 2.) Ein weiteres Risiko ist die verhältnismäßig kurze Codierungszeit nach dem Neujahr (1 Monat), sofern das erste Risiko eintritt, ist der entstehende Schaden hier sehr hoch (eine Eintrittswahrscheinlichkeit ist wegen des kurzen Zeitraums auch relativ hoch).
- 3.) Aus den Risiken 1.) und 2.) folgt dass man wichtige Details des Softwareprojekts vernachlässigt (die Dokumentation, die Testpläne). Auch hier gilt, dass die Eintrittswahrscheinlichkeit und der resultierende Schaden von den Risiken 1.) und 2.) abhängig sind.
- 4.) Unabhängig davon besteht das Risiko eines Ausfalls eines Projektmitarbeiters. Auf Grund einer infizierenden, grippalen Erkrankung ist die Eintrittswahrscheinlichkeit zur Winterzeit hoch, auch der resultierende Schaden ist hoch, da eine komplette Funktionalität ausfällt.
- 5.) Es ist immer mit einer Anforderungsänderung zu rechnen. Über die Eintrittswahrscheinlichkeit kann hier keine genaue Annahme getroffen werden. Der Schaden berechnet sich mit dem Eintreffen der Anforderungsänderung. Wird eine Anforderungsänderung früh angesprochen, so bleibt der Schaden gering. Wird er spät angesprochen, ist mit hohem Schaden zu rechnen.

### 5.2 Risikoreduktion und Gegenmaßnahmen

Zur Reduktion des Risikos (zeitliches Risiko) soll ein fester, wöchentlicher Termin innerhalb der Projektgruppe festgelegt werden, um sich einen Überblick über den Stand der Dinge zu verschaffen. Er dient auch der Diskussion und dem Abgleich bestimmter Ergebnisse aus Einzelarbeiten. Aus dem festen wöchentlichen Termin soll entschieden werden, wann es Bedarf für Zusatztermine gibt. Es wird entschieden über die Art und die Anzahl der Zusatztermine.

Zur Reduktion des Schadens vom Risiko (Grippaler Infekt) soll im Falle des Eintretens ein Mitarbeiter vom erkrankten Mitarbeiter eine kleine Schulung bekommen (per Dokumentation oder Selbststudium), sodass dieser dann die fehlenden Aufgaben (z.B. Bedienen des GTD-Managers, etc...) komplett übernehmen kann.

Zur frühen Erkennung eines Risikos (Typ 5.)) soll täglich das Kundenforum untersucht werden. Man soll versuchen so gut wie möglich mit dem Kunden in Kontakt zu stehen und eine mögliche Anforderungsänderung noch am selben Tag erfahren. Sollte dann eine Anforderungsänderung erfasst werden, kann der Schaden reduziert werden, durch Rücksprache mit dem Kunden und dem Abgleich der neuen / alten Anforderungen (wenn z.b. durch neue Anforderungen eventuell alte wegfallen. Oder neue Anforderungen doch nicht gewollt werden oder sogar nicht realisierbar sind).

Sollten im allgemeinen Risiken auftreten, die das Softwareprojekt behindern könnten, so werden die Anforderungen priorisiert werden müssen. Eine Priorisierung der Anforderungen ist teilweise schon durch den Kunden gegeben (angesprochene Must-Haves sollten zuerst entwickelt werden, dann Should-Haves, dann Nice-to-Haves). Die Anforderungen unter diesen Punkten werden dann erneut eine innere Priorität erhalten.

## **6. Richtlinien für die Entwicklung**

### **6.1 Design- und Programmierrichtlinien**

Der Code soll vollständig mit Kommentaren in einer konsistenten Sprache (Orientierung an der Programmiersprache selbst) versehen werden.

Es ist festgelegt worden (im Team), dass für den Code die Java-Code-Conventions von Oracle als Richtlinie angesehen werden. Siehe auch:

<http://www.oracle.com/technetwork/java/codeconv-138413.html>

Ansonsten wurden explizit keine weiteren Design-Richtlinien gefordert im System.

### **6.2 Einsatz von Werkzeugen**

Bis zu diesem Zeitpunkt (Version und Änderungsdatum des Projektplans) sind folgende Werkzeuge im Einsatz:

- GTD - Manager (siehe Gantt und Termindrift-Diagramm)
- Microsoft Word (Projektplan und weitere, schriftliche Dokumente)
- Java als Programmiersprache
- Eclipse IDE (Juno Version) für die Implementierung
- zu Eclipse: Java Server Faces als Plugin
- zu Eclipse: Als Applikationsserver das Tomcat Plugin und Glassfish.
- RevAger für zukünftige Reviews
- CodeCover für GassBox-Tests (Messung einer Code-Überdeckung)

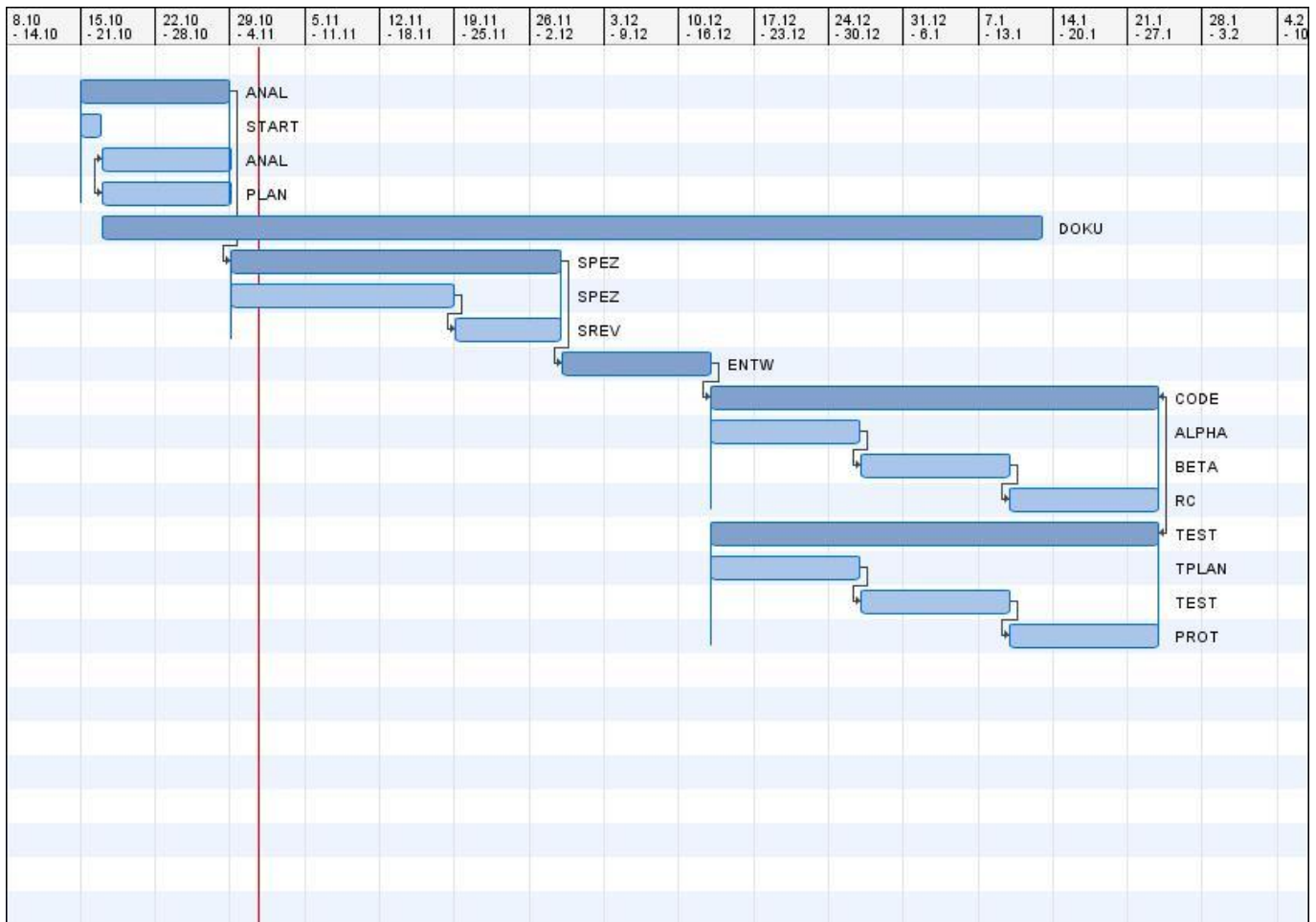


## 7. Schlüsselpersonen des Projekts

Kunde extern:	Bastian Calaminus E-Mail: Bastian.Calaminus@dlr.de
Kunde intern:	Jan-Peter Ostberg Telefon: +49 711 685-88555 E-Mail: Jan-Peter.Ostberg@informatik.uni-stuttgart.de
Betreuer:	Jasmin Ramadani Telefon: +49 711 685-88306 E-Mail: Jasmin.Ramadani@informatik.uni-stuttgart.de
Projektmitarbeiter 1:	Alexander Blehm (Usability-Ingenieur) E-Mail: blehmar@stud.informatik.uni-stuttgart.de
Projektmitarbeiter 2:	Simon Lehmann E-Mail: lehmannsn@stud.informatik.uni-stuttgart.de
Projektmitarbeiter 3:	Oleg Mryga E-Mail: mrygaog@stud.informatik.uni-stuttgart.de

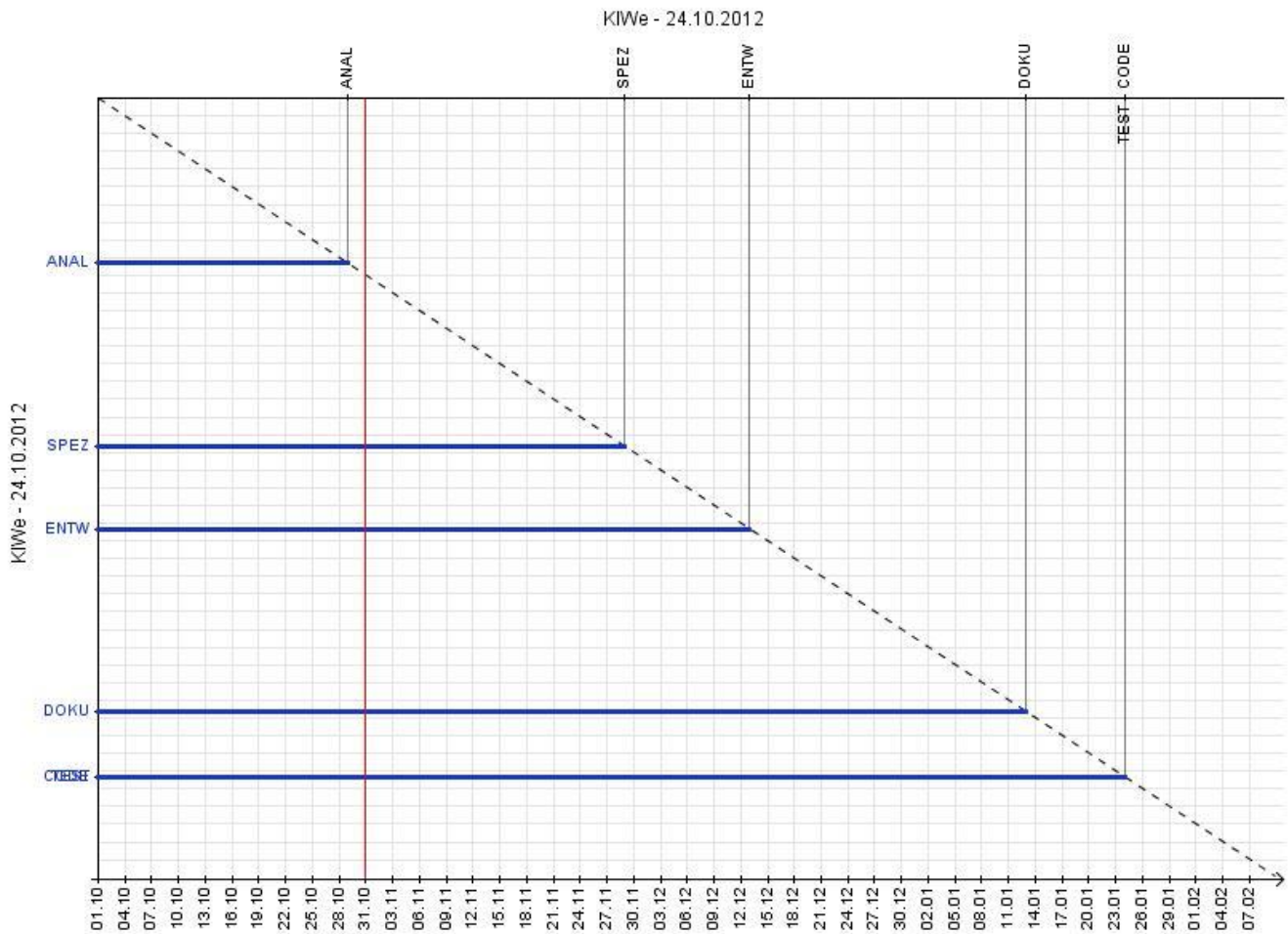
## 8. Entwicklungsplan

### 8.1 Gantt-Diagramm



- **START** - Projektbeginn
- **ANAL** - Analyse, feststellen der Anforderungen
- **PLAN** - Erstellung eines Projektplans
- **SPEZ** - Spezifikation
- **SREV** - Review und Korrektur der Spezifikation
- **ENTW** - Entwurf
- **DOKU** - Allgemeine Dokumentationsphase (Projektplan, Spezifikation, Entwurf, Systemtest und Modultestpläne)
- **CODE** - Implementierung
- **ALPHA** - Erster Zwischenstand der Implementierung
- **BETA** - Zweiter Stand der Implementierung (lauffähig)
- **RC** - Release Candidate (das fertige Programm)
- **TEST** - Allgemeine Tests (untergeordnet als "Modultest" anzusehen, also auch Beta-Test)
- **TPLAN** - Ein Systemtestplan und es folgt das Testen der Alpha Version
- **PROT** - Testen des Release Candidates und erstellen von Systemtestprotokollen (Ergebnisse der Testdurchläufe)

## 8.2 Termindrift-Diagramm



## 9. Versionshistorie

1.00 - erster Prototyp des Projektplans (Projektmitarbeiter 1 - 3)

1.01 - Erstellen der Diagramme (Simon Lehmann), Anpassung der Anforderungen (Alexander Blehm, Oleg Mryga)

1.02 - Anpassung der Anforderungen mit Verweis auf die Spezifikation (Alexander Blehm, Oleg Mryga). Anpassen der Gantt-Diagramme (DOKU-Phase) und der Beschreibung der Testphasenabschnitte (Simon Lehmann). Hinzufügen von möglichen Risiken, genauere Werkzeugbeschreibung genauere Programmierrichtlinien (Alexander Blehm, Oleg Mryga).