Alexander Blehm - 2712024, Simon Lehmann - 2761826, Oleg Mryga - 2690599

Update: 31.10.2012

**Ist-Analyse:**

\* Bis jetzt wurde das Berechnungsprogramm manuell genutzt, also Eingabedaten wurden mit der Hand eingetippt, Ergebnisse manuell gespeichert und archiviert.

\* Es gab ein kleines Pearl-Skript, das diesen "Händischen" Prozess unterstützen sollte.

\* Es gibt zudem kein vergleichbares Programm, das den beschriebenen Anforderungen gerecht wird.

\* Es gibt wenige Menschen, die das Berechnungsprogramm momentan nutzen (Zitat des Kunden: "ein überschaubarer Kreis der User"). Folglich landet auch der Wunsch nach einem Mehrbenutzerbetrieb in der niederen Prioritätskategorie: "Nice-to-Haves".

**Soll-Analyse:**

**Design und Struktur:**

\* Es soll eine grafische Benutzeroberfläche geben (also nicht nur Kommandozeileneingabe).

-> Explizite Design-Vorgaben (oder Strukturvorgaben eines Menüs) gibt es nicht. Es wurde aber darauf hingewiesen, dass es sinnvoll aufgebaut sein sollte. Es gebe auch deshalb keine genauen Strukturellen Design-Vorgaben, da es nicht viele Eingabedaten gibt. Einzige Vorgabe ist: Datenfelder sollen 10 existieren (damit man 1 bis 10 Referenzobjekte eingeben kann) - diese sollen alle auf einer Seite angezeigt werden (keine Tabs/neue Fenster erwünscht).

\* Die Endanwenderstruktur sollte eine Website sein! Keine Desktop-Anwendung.

-> Unterstützte MUST-Have Browser sind: Internet Explorer (eventuell Firefox, wobei sowieso offiziell nur der Internet Explorer zugelassen ist beim DLR).

\* Die Ergebnisse (der Berechnung) sollen schön, grafisch aufgearbeitet werden (aber auch als Text-Datei oder Tabelle für wissenschaftliche Zwecke).

**Technische Anforderungen:**

\* Das Ziel-Betriebssystem, auf dem das Programm läuft, wird Windows sein. (Win 7)

\* Die Datenbasis soll sich vor jeder Rechnung aktualisieren (da Daten sich alle 3 Stunden online ändern). -> Hier wurde explizit erwähnt, dass man keinen Wert auf einen schnellen Download der Celestrak Daten setze.

\* Es sollen Standartports (über http(s)) verwendet werden.

\* Es sollen auch eigene Daten eingegeben werden können (die Referenzobjekte).

-> Gemeint sind damit mehrere TLE Datensätze.

-> Es soll eine Prüfung geben, ob irgendwelche eingelesenen TLE Datensätze korrupt sind (also eine Art TLE-Datensatz -Verifikation. Richtlinien könne man aus Wikipedia entnehmen).

-> Man soll eigene Beschränkungen eingeben sollen (z.B.: Zeitraum, Kollisionsabstände von kleiner 3 Km, 5Km oder 10 Km). -ABER: schlechte Zeiträume sollen Warnmeldungen bringen (Bedingung waren Zeiträume, die größer als eine Woche sind, da die aktuellen Datensätze bis dahin sehr ungenau sein könnten.).

\* Es soll das Berechnungsprogramm angebunden werden.

-> Das heißt, über das fertige Programm, soll das externe Zweitprogramm ("Berechnungsprogramm", das den Algorithmus enthält) bedienen können.

\* Die Ergebnisse sollen speicherbar sein

-> in einer Datenbank (auch die Datenbank "sollte" wählbar sein).

-> Lokal (auch ohne Eintrag in eine Datenbank).

\* Es sollen die Fehlermeldungen des Berechnungsalgorithmus nebenher in einer Log-Datei gesichert werden.

\* Es wird eine Lauffähigkeit mit Java 1.7 gefordert.

**"SOLL" Anforderungen:**

**Technische Anforderungen:**

\* Ein geodätisches Referenzsystem soll zu Beginn bestimmt werden können (z.b.: WGS84, WGS72,... etc.).

\* Falls Celestrak "Down" ist, soll eine alternative Datenquelle gewählt werden können. Das nächste ist "AGI". Weitere Quellen wurden nicht genannt.

\* Angezeigte Ergebnisse, sollten nach sinnvollen Kriterien (alphabetisch) sortierbar sein ("should-have").

\* Eine Fortschrittsanzeige sollte vorhanden sein für eine Berechnung.

**Dokumentationsanforderungen:**

\* Eine anständige Dokumentation (Handbuch) in Form einer .pdf und .doc (Word) Datei, die erklärt, was das Programm überhaupt macht sollte im Lieferumfang dabei sein.

\* Auch JavaDoc ist gefordert.

**"NICE-TO-HAVE" - Anforderungen:**

**Technische Anforderungen:**

\* Name, Beschreibung einzelner Berechnungen sollen individuell speicherbar sein.

-> auch hier in Grafischer Form.

\* Information über Ende der Simulation (vllt Sound-Ausgabe! Oder E-Mail/SMS).

\* Ein Mehrbenutzerbetrieb wäre ein Wunsch.  
\* dazu gehört eine User-Authentifizierung.

\* Log Datei (FÜR das Programm, eventuell auch mit Namen des Benutzers).

\* Abfrage der Datenbankeinträge sollte nach Kriterien möglich sein.

-> Kriterien: Satellitenname, Maximale Entfernung, Zeitraum.

-> Direkter Export dieser Abfrageergebnisse soll möglich sein

Anwenderrechte-System:

\* Auf Serverseite - Administrationsrechte (wird vom Herrn Calaminus administriert)

\* Auf Desktop-Seite (also normale Userrechte)

-> Diese Unterteilung kommt aber unter der Voraussetzung eine Art Benutzersystem anzulegen.

\* Backups des Programmes, sofern eingebaut, sollten Nachts laufen. ABER WICHTIG: auch konfigurierbar sein, falls es anspruchsvolle Projektzeiten gibt und viel gearbeitet werden muss.

**Nicht Kunden- / Projektspezifische Aussagen:**

\* Plugins sollen durch Betreuer genehmigt werden. ACHTUNG: können nur benutzt werden, wenn auch die Lizenz mit diesem Softwareprojekt (KIWe) kompatibel und legal ist.

\* Das Wort "Satellit" soll nicht nur aktive, nützliche Satelliten beschreiben, sondern alle Weltraumobjekte, die größer als 10cm sind! (Auch "Weltraumschrott")

**Weitere, nicht Funktionale Anforderungen:**

\* Mehrsprachigkeit (Englisch / Deutsch / erwünscht sind auch andere Sprachen) für das laufende System.

-> WICHTIG ist hier: Man soll sich bei der Implementierung auf eine einheitliche Sprache (konsistent pro Team) einigen. Die Dokumentation darf allerdings abweichen. Gewünscht ist:

Englischer Quellcode und JavaDoc. Deutsches Handbuch, Projektplan, Spezifikation.