



# **Alarm Clock**

Jonathan Hyams  
Pascal Schmalz

14. März 2017

# Inhaltsverzeichnis

0.1	Zweck des Dokument . . . . .	4
0.2	Kurzbeschreibung . . . . .	4
0.3	Projektziele . . . . .	4
0.4	Stakeholders . . . . .	4
0.5	Systemabgrenzung . . . . .	4
0.5.1	Geschäftsprozesse . . . . .	4
0.5.2	Systeme . . . . .	4
0.5.3	Randbedingungen . . . . .	6
0.5.4	Prozessumfeld . . . . .	6
0.5.5	Systemumfeld . . . . .	6
0.5.6	Randbedingungen . . . . .	6
0.6	Anforderungen . . . . .	6
0.6.1	Quellen und Vorgehen . . . . .	7
0.6.2	Technische Anforderungen . . . . .	8
0.6.3	Qualitätsanforderungen . . . . .	8
0.7	Glossar . . . . .	8
0.8	Anhang . . . . .	10
0.8.1	Abstimmung der Anforderungen . . . . .	10
0.8.2	Definition of Ready - Checklist . . . . .	10
0.9	Versionskontrolle . . . . .	10



## 0.1 Zweck des Dokument

TODO this is new text

## 0.2 Kurzbeschreibung

Das Ziel des Projektes ist einen Ersatz zum Programm kAlarm zu entwickeln. Das Programm kAlarm erlaubt es den User Benutzerdefinierte Erinnerungen zu erstellen. Mittels Pop-Up Windows wird der User dann zur gegebenen Zeit daran erinnert. Im gegensatz zu kAlarm soll das zu erstellende Produkt Plattformübergreifend verfügbar sein. Wie kAlarm soll dieses Produkt unter einer Open Source Lizenz entwickelt werden.

## 0.3 Projektziele

Unser Auftraggeber ist kein Unternehmen. Es werden somit auch keine Ziele innerhalb einer Unternehmens verfolgt. Ziele welche ein Benutzer unsere Applikation verfolgen könnte wären Beispielsweise:

- Wiederkehrende Events nicht zu vergessen.
- An den Wäscheta erinnert werden.
- Die Wäsche rechtzeitig aus der Wäscheküche holen.
- Den Kuchen nicht im Backofen verbrennen lassen.
- Der Sekretärin zum Geburtstag Blumen schicken.

## 0.4 Stakeholders

- Auftragsgeber: Prof. Claude Furrer
- Auftragsnehmer: Jonathan Hyams, Pascal Schmalz
- Benutzer: FOSS Community

## 0.5 Systemabgrenzung

### 0.5.1 Geschäftsprozesse

Es existieren keine vor oder nachgelagerten Prozesse. Unsere Lösung lässt sich in beliebige viele Prozesse integrieren.

### 0.5.2 Systeme

Unser System wirkt mit Datenbanken zusammen. Es baut auf Betriebssystemkomponenten auf. Namentlich Cronjob /Schtasks.

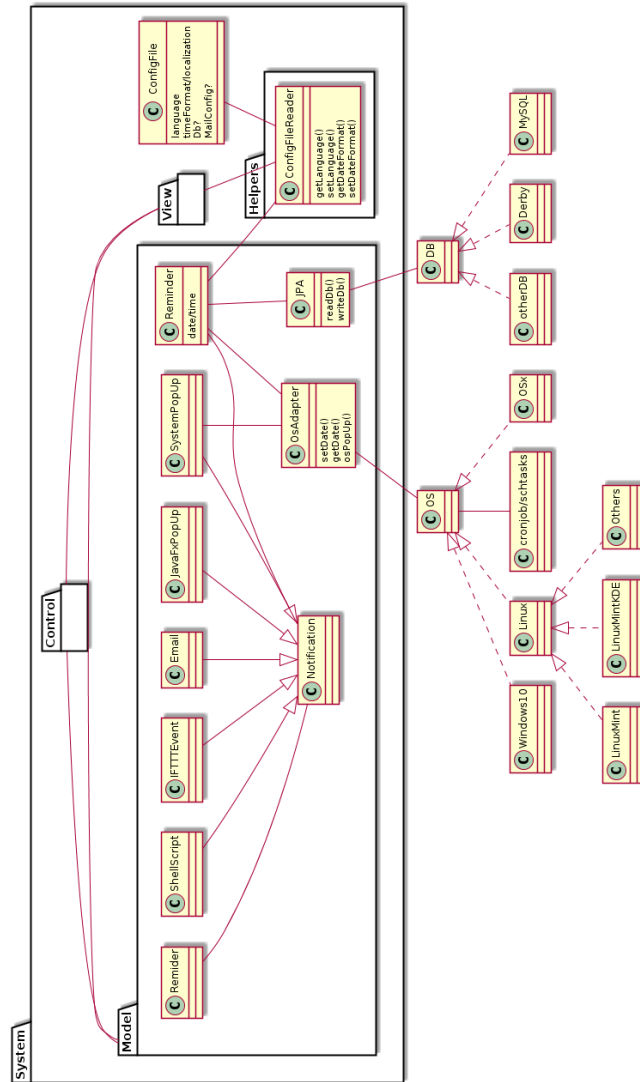


Abbildung 0.1: Systemübersicht und Abgrenzung

Die Systemübersicht und Abgrenzung Abb. 0.1 Seite 5 bedient sich der UML Notation, ohne die UML Spezifikation vollständig einzuhalten. 5

Je nach gewünschter Notification Möglichkeiten kann mit E-Mails, SMS und Systemnotifications gearbeitet werden. Dies Bedingut, dass entsprechende Services / Server eingerichtet und über eine definierte Schnittstelle erreichbar sind.

### 0.5.3 Randbedingungen

Die Entwicklung soll mittels Java und JavaFX erfolgen. Das Prgoramm soll Betriebssystem und Datenbank agnostisch sein. Das Projekt muss unter einer Opensource Lizenz veröffentlicht werden. Die Dokumentation muss in  $\text{\LaTeX}$  erstellt werden.

### 0.5.4 Prozessumfeld

### 0.5.5 Systemumfeld

### 0.5.6 Randbedingungen

## 0.6 Anforderungen

Basisfaktoren:

- Es soll ein Timer erstellt werden, welcher auf den 3 grossen Computerbetriebssystemen (Windows, OSX, Linux) läuft. (Aleinstellungsmerkmal)
- Das Programm soll Datenbankagnostisch sein. (Aleinstellungsmerkmal)
- Einzelevents sollen definiert werden können.
- Wiederkehrende Events sollen definiert werden können.
- Die Verwaltung der Events soll über eine GUI vorgenommen werden können.
- Auf die Events soll mittels einer Notification hingewiesen werden
- Das Programm soll durch den Benutzer mittels eines externen Configfile an seine Bedürfnisse angepasst werden können.

Leistungsfaktoren:

- Events sollen kategorisiert werden können.
- SMS Notification.
- Email Notification.
- Script wird bei Notificatin ausgeführt.
- GUI soll Benutzerzentriert und Ergonomisch sein.

Begeisterungsfaktoren:

- IFTTT aktionen mittels Events auslösen. (Alleinstellungsmerkmal)
- Eine Lauffähige Version auf Android.
- Events Geräteübergreifend verwalten.
- Notification mit Betriebssystem eigenen Notifications anstelle von JavaFX Notification
- GUI an die jeweiligen Betriebssystem Design Richtlinien angepasst.

### 0.6.1 Quellen und Vorgehen

Unsere wichtigste Quelle ist unser Auftraggeber Prof. Furrer. Um die Fragen, welche bei der Erstellung der Anforderungen aufgetaucht sind zu klären, werden wir uns mit ihm treffen. Besondere folgende Fragen gilt es abzuklären:

- Entspricht die bisherige Ausarbeitung der Vorstellung des Auftraggebers.
- vollständigkeit der Anforderungen.
- Kategorisierung der Anforderungen.
- Priorisierung der Anforderungen.
- Was soll mit der Eventkategorisierung erreicht werden, welche weiteren Anforderungen ergeben sich daraus.
- Abgrenzung des Programmes gegenüber eines Kalenders und gegenüber des Cron-jobs.

Als weitere Diskussionsgrundlage sind wir daran neben diesem Dokument einen Prototyp des GUI zu erstellen.

Die Angegebenen Informationen,insbesonder die abzuklärenden Punkte, sind dementsprechend lediglich eine Work in Progress.

Als weitere Quellen können wir die Dokumentation und Code Basis von kAlarm nutzen. Da wir selber der potentiellen Nutzergruppe des Programmes angehören, könnten wir auch uns selber als Quelle nutzen, als Entwickler hat man jedoch oft einen anderen Blickwinkel als ein normaler Benutzer.

Im sinne einer Agilen Entwicklung wollen wir möglichst rasch präsentierbare Artefakte generieren, damit der Auftraggeber frühstmöglich Einfluss nehmen kann und Fehlentwicklungen vermieden werden. Treffen werden bei bedarf mit dem Auftraggeber ausgemacht.

### **0.6.2 Technische Anforderungen**

### **0.6.3 Qualitätsanforderungen**

## **0.7 Glossar**



# Abbildungsverzeichnis

0.1 Systemübersicht und Abgrenzung . . . . .	5
--	---

# Tabellenverzeichnis

## **0.8 Anhang**

### **0.8.1 Abstimmung der Anforderungen**

### **0.8.2 Definition of Ready - Checklist**

## **0.9 Versionskontrolle**

Manuelle Version: 0.0.2

Automatische Versionierung:

Last compiled: Tue Mar 14 16:28:28 CET 2017

Git HEAD Version: 22