

# **Projektführungsbericht**

Niklaus Hofer, Lukas Knöpfel, Kaleb Tschabold

May 30, 2011

<b>Status</b>	In Arbeit/In Prüfung/ <b>Abgeschlossen</b>
<b>Projektname</b>	Projektexplorer
<b>Projektleiter</b>	Lukas Knöpfel
<b>Auftraggeber</b>	M. Frieden, GIBB
<b>Autoren</b>	Kaleb Tschabold, Lukas Knöpfel, Niklaus Hofer
<b>Verteiler</b>	Lukas Knöpfel, Kaleb Tschabold, Niklaus Hofer

### Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
1.0	08.02.2011	Dokument schreiben	Niklaus Hofer

### Definitionen und Abkürzungen

Begriff/ Abkürzung	Bedeutung
CLI	Command Line Interface
GUI	Graphical user interface
DB	Database
KM	Konfigurationsmanagement

### References

- [1] Lukas Knoepfel Kaleb Tschabold Niklaus Hofer. Konzeptbericht. Teil der Abgabedokumente, 2011.
- [2] Lukas Knoepfel Kaleb Tschabold Niklaus Hofer. Realisierungsbericht. Teil der Abgabedokumente, 2011.

## Contents

<b>1</b>	<b>Zweck des Dokuemnts</b>	<b>4</b>
<b>2</b>	<b>Projektplan</b>	<b>4</b>
<b>3</b>	<b>Projektbericht</b>	<b>6</b>
3.1	KW 7	6
3.2	KW 8	6
3.3	KW 9	6
3.4	KW 10	6
3.5	KW 11	7
3.6	KW 12	7
3.7	KW 13	7
3.8	KW 17	8
3.9	KW 18	8
3.10	KW 20	8
3.11	KW 21	8
<b>4</b>	<b>Risikokatalog</b>	<b>9</b>
<b>5</b>	<b>QS-Plan</b>	<b>9</b>
5.1	Vorgehen zur Qualitätssicherung	9
5.2	Qualitätsziele	9
5.3	Prüfplan	11
5.4	Pföfmethoden	11
5.4.1	Review	11
5.4.2	Applikations-Tests	11
5.5	Prüfspezifikationen	12
5.5.1	Checklisten für die Prüfung der Projektdokumente	12
5.5.1.1	Projektantrag	12
5.5.1.2	Projektplan	12
5.5.1.3	Voranalysebericht	12
5.5.1.4	Konzeptbericht	12
5.5.1.5	Programm	12
5.5.1.6	Realisierungsbericht	12
5.5.1.7	Einführungsbericht	12
5.5.1.8	Projektführung	12
5.5.2	Testfalltabellen	12
<b>6</b>	<b>Konfigurationsmanagementplan (KM-Plan)</b>	<b>13</b>
6.1	Aufzubewahrende Projektergebnisse und ggf. sonstige Dokumente	13
6.2	Ablagestruktur	13
6.2.1	documentation	13
6.2.2	Code	14
6.3	Namenskonventionen	14
<b>7</b>	<b>Konfigurationsidentifikation</b>	<b>14</b>
7.1	Dokumentation	14
7.2	Applikation (sourcecode)	14

## **1 Zweck des Dokuments**

Zusammenfassung von Planung und Ergebnissen zu den fünf Projektführungsthemen.

- Projektmanagement
- Risikomanagement
- Qualitätsmanagement
- Konfigurationsmanagement
- Projektmarketing

## **2 Projektplan**

Aktivität	PL Soll	PL Ist	KW 05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Initialisierung	12	12																					
Voranalyse	36	36																					
Konzept	36	36																					
Realisierung	48	192											Ferien										
Einführung	24	24																					
Reserve	24	-																					
Abschlussphase	12	24																					
Projektleiter																							
Niklaus Hofer																							
Kaleb Tschabold																							
Lukas Knöpfel																							

## 3 Projektbericht

### 3.1 KW 7

**Stand der Arbeit:**

- Nachforschungen zu verschiedenen Technologien.
- Die Arbeiten am Voranalysebericht schreiten gut voran.
- Wir haben bereits ein recht gutes Bild der aktuellen Lage geschaffen und wissen in etwa, welche Produkte vergleichbares liefern.

**Probleme und Fragen:**

- Zurzeit gibt es von unserer Seite her keine Fragen.

**nächste Schritte:**

- Nächsten Dienstag werden wir die Anforderungen erarbeiten.
- Sobald wir uns zu den verschiedenen Technologien, die für das Projekt in Frage kommen informiert haben (voraussichtlich nächste Woche, hängt von den Vortschritten bei den Specs ab), werden wir die verschiedenen Möglichkeiten analysieren und vergleichen.

### 3.2 KW 8

**Stand der Arbeit:**

- Wir sind zur Zeit mit der Evaluierung der verschiedenen Varianten beschäftigt. Dabei sind die mit Ihnen bereits besprochenen Probleme aufgetaucht, die wir nun als nächsten Schritt lösen werden.
- Die Arbeit am Voranalysebericht schreitet ansonsten gut voran und ist beinahe abgeschlossen.
- Parallel zu dem neuen Dokument arbeiten wir zur Zeit auch an der Korrektur der alten Unterlagen.

**Probleme und Fragen:**

- Zurzeit haben wir keine weiteren Fragen.

**nächste Schritte:**

- Der nächste konkrete Schritt ist das Anpassen der Versionsanalyse.
- Ist die Voranalyse abgeschlossen wird das Dokument aufgeräumt (Rechtschreibung, Punktierung, . . .) und dann mit LaTeX in die Reinverson geschrieben.

### 3.3 KW 9

**Stand der Arbeit:**

- Voranalysebericht fertig gestellt.
- Parallel zu den neuen Dokumenten arbeiten wir zur Zeit auch an der Korrektur der alten Unterlagen.

**Probleme und Fragen:**

- Zurzeit haben wir keine weiteren Fragen.

**nächste Schritte:**

- Erstellen des Konzeptberichtes.

### 3.4 KW 10

**Stand der Arbeit:**

- Ersellung des Konzeptberichts.
- Arbeit am Klassendiagramm.

**Probleme und Fragen:**

- Zurzeit haben wir keine weiteren Fragen.

**nächste Schritte:**

- Die Arbeit am Konzeptbericht wird noch eine Menge Arbeit beanspruchen. Besonders muessen viele Funktionen genau geplant und durchdacht werden.

### 3.5 KW 11

**Stand der Arbeit:**

- Ich(Kaleb) habe nun die Aufgabe des Projektleiters übernommen.
- Wir haben heute zum grossen Teil einige Prototypen erstellt.
- Auch haben wir unser Klassen-Diagramm überarbeitet.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Da wir mit unserem Konzept noch nicht weit sind, müssen wir jetzt vor allem zu Hause das Konzept erarbeiten.
- Besonders das Klassendiagramm wollen wir optimieren.
- Auch die kleinen Korrekturen von den vorherigen Dokumenten müssen wir noch nachführen(Projektantrag und Voranalyse).

### 3.6 KW 12

**Stand der Arbeit:**

- Wir haben gestern das Konzept abgeschlossen. Wir haben gemerkt, dass wir den Zeitplan nicht gut eingeteilt haben und darum sind wir gestern ein bisschen in Zeitdruck gekommen.
- Wir beginnen jetzt mit der Realisierung.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Wir werden nun nächstes Mal, die Aufteilung machen, in der definiert ist, wer welche Programmteile programmiert.
- In den Frühlingsferien haben wir dann Zeit um voll am Projekt zu arbeiten.

### 3.7 KW 13

**Stand der Arbeit:**

- Wir sind gerade dran richtig in die Realisierungsphase zu starten.
- Wir sind gerade dran alle Dateien und etc. zu erstellen, damit wir, dann die Funktionalitäten erstellen können.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Ich werde einen Zeitplan für die Unterrichtsfreizeit machen so, dass wir gegen Ende dieser Phase nicht in Zeitdruck kommen.
- Wir müssen diese Zeit auch nutzen um unsere Dokumente(Voranlyse,...) zu korrigieren.
- Auch die aktuellen Dokumente(Realisierungsbericht und Projektführungsbericht) ergänzen.

### 3.8 KW 17

**Stand der Arbeit:**

- Wir sind gerade voll dran am System entwickeln.
- Ich habe gerade heute Abend die erste Version zum Setzen und Suchen von Tags programmiert.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Wir müssen besprechen wer, was bis zum nächsten Meilenstein macht.
- Auch den Realisierungsbericht müssen wir anfangen zu erstellen.

### 3.9 KW 18

**Stand der Arbeit:**

- Ich (Lukas) habe nun die Aufgabe des Projektleiters übernommen.
- Wir haben eine experimentelle Version unseres Programmes die schon einige Grundfunktionen integriert hat.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Wir werden hauptsächlich am Programm arbeiten.
- Aus Zeitmangel werden wir uns auf die Linux Version unseres Programmes konzentrieren. Die Windows und Mac Versionen werden je nachdem auf wieviele Probleme wir stossen weniger Funktionalität besitzen als die Linux Version.

### 3.10 KW 20

**Stand der Arbeit:**

- Wir haben eine stabile Version unseres Programmes auf Linux.
- Wir haben uns einen Überblick über das Projekt verschafft. Wir wissen nun welche Funktionen funktionieren und welche eine Überarbeitung benötigen.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Wir werden alle Funktionen in das GUI implementieren, damit wir sie an der Präsentation zeigen können.
- Ausserdem werden wir das Programm noch vollständig auf Windows portieren und eine Versionierung einbauen.
- Wir bereiten uns auch auf die Präsentation vor, da diese nun vorverschoben wurde.

### 3.11 KW 21

**Stand der Arbeit:**

- Wir arbeiten an der Präsentation um diese wie vereinbart nächsten Dienstag Vorzutragen.

**Probleme und Fragen:**

- Zurzeit haben wir keine Fragen.

**nächste Schritte:**

- Wir werden die Präsentation fertigstellen.
- Wir werden uns absprechen um im vorgegebenen Zeitrahmen zu bleiben.
- Wir werden eine dedizierte Maschine aufsetzen um unser Tool zu demonstrieren.



## 4 Risikokatalog

Nr.	Risiko für das Projekt	mögliche Auswirkungen	Massnahmen	Status
1.	Die ausgewählten Technologien sind für die Programmierung der Applikation nicht geeignet, da Funktionen fehlen oder aufwendig zu implementieren sind.	Die Arbeit an den einzelnen Teilen des Programmes wird massiv aufwendiger. Verspätungen können nicht ausgeschlossen werden. Evtl. müssten sogar Teile der Applikation mit einer anderen Technologie neu geschrieben werden.	Genaue Nachforschungen zu den Technologien anstellen. Prototypen und kleine Testapplikationen entwickeln.	erledigt
2.	Die Applikation kann nicht rechtzeitig fertiggestellt werden, da zu viele Funktionen geplant sind, die Implementierung Probleme macht oder Mitglieder des Teams ausfallen.	Das Projekt kann nicht rechtzeitig fertiggestellt werden. Die Qualität der Applikation leidet.	Planen der Ressourcen.	erledigt
3.	Die Kompatibilität mit allen Betriebssystemen kann nicht gewährleistet werden, da manche features auf einer Plattform viel schwieriger zu implementieren sind als auf einer anderen.	Das Programm läuft nicht auf allen Systemen gleich stabil. Möglicherweise sind auch einige Funktionen nicht auf allen Plattformen verfügbar. Zeitaufwendige Implementierung für verschiedene Plattformen könnten auch zu Verspätungen führen.	Verwendung einer Systemunabhängigen Sprache (Python). Nachforschungen anstellen zu den Modulen, die für die Funktionalität entscheidend sind. Im schlimmsten Falle könnten wir auch den Support für eine Plattform fallen lassen.	erledigt
4.	Die Applikation genügt den Ansprüchen in Sachen Stabilität, Nutzerfreundlichkeit oder einem anderen Gebiet nicht.	Die Applikation funktioniert nicht wie gewünscht.	Die einzelnen Komponenten, ihre interoperabilität und integration sowie das fertige Produkt sind zu testen	erledigt

## 5 QS-Plan

### 5.1 Vorgehen zur Qualitätssicherung

Die Qualität der Applikation haben wir durch Tests sichergestellt. Code-reviews haben keine stattgefunden. Zum Testen der Applikation kamen vorallem Modultests zum Einsatz. Genauere Details zur Durchführung der Tests können [2, 4. Systemtests] entnommen werden.

Die Dokumente wurden jeweils nach dem Erstellen durchgelesen. Die Struktur der Vorlagedokumente gab jeweils vor, welche Informationen enthalten sein mussten.

### 5.2 Qualitätsziele

Nr.	Qualitätsmerkmal	Qualitätsziel	besondere QS-Massnahmen um das Ziel zu erreichen
<b>aus Benutzersicht</b>			
1	Funktionserfüllung	Die Applikation soll die im Konzeptbericht [1] festgelegten Ziele erfüllen.	-
2	Effizienz	Die Applikation soll zügig reagieren, die Benutzung soll sich durchgehen schnell anfühlen.	Evtl. müssen einzelne Teil des Codees auf Geschwindigkeit optimiert werden.
3	Zuverlässigkeit	Die Applikation soll auch bei falschen Eingaben oder schnellen Eingabe-Abfolgen zuverlässig funktionieren und darf nicht abstürzen.	Die Tests sehen auch ungewöhnliche Szenarien vor, die die Applikation stark belasten und in dieser Form im Alltag nicht zu erwarten sind.
4	Benutzbarkeit	Die Oberfläche des Programmes soll intuitiv sein. Wo das nicht direkt möglich ist, soll sich die Applikation an gewohnte Bedienungs-Paradigmen halten.	-
5	Sicherheit	Die Applikation soll die Sicherheit und integrität der Benutzerdaten nicht gefährden.	
<b>aus Entwicklersicht</b>			
6	Erweiterbarkeit	Die Applikation wird objektorientiert entwickelt. Die Klassen werden systematisch und eindeutig bennt. Zudem sind die Funktionen fehlertolerant aufgebaut. Funktionen die von aussen genutzt werden können enthalten eine Erläuterung.	-
7	Wartbarkeit	Die Applikation ist ausreichend dokumentiert. Abläufe sind direkt im Code dokumentiert, komplexe Vorgänge sind in der Dokumentation erklärt.	-
8	Übertragbarkeit	-	-
9	Wiederverwendbarkeit	Die Komponenten der Applikation hänge nicht fest voneinander ab. Einzelne Komponenten (z.B. das Backend) können auch in anderen Projekten wiederverwendet werden.	-
<b>Projektführung</b>			
10	Kommunikation der Beteiligten	Die Entwickler sprechen sich untereinander ab, damit nicht aneinander vorbeiprogrammiert ist. Treten Fragen zum Code eine anderen Mitgliedes des Teams auf, so können diese gestellt werden.	-
12	Termineinhaltung	Die Termine aus dem Projektplan (siehe oben) werde eingehalten.	-
13	Projektdokumentation	Die Vorgegebenen Dokumente werden pünktlich eingereicht. Mails die zwischen den Entwicklern ausgetauscht werden werden aufgehoben.	-

### 5.3 Prüfplan

Prüfobjekt	Termin	Prüfer	Prüfmethode	Prüfkriterien	Bemerkungen
Projektantrag		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung	
Projektplan		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung, Tabelle ausgefüllt	
Voranalysebericht		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung, Diagramme vorhanden, Auswertungen vorhanden	
Konzeptbericht		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung, Diagramme vorhanden, Anwendungsfälle beschrieben, Klassendiagramm vorhanden, DB-Schema vorhanden	
Programm		Lehrperson	Tests	siehe [2, 4. Systemtests]	
Realisierungsbericht		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung, Klassendiagramm aktuell, DB-Schema aktuell, Benutzerhandbuch vorhanden, Supporthandbuch vorhanden, Tests erläutert, Code im Dokument	
Einführungsbericht		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung	
Projektführung		Lehrperson	Review	Alle Punkte behandelt (nach Vorlage), Rechtschreibung	

### 5.4 Prüfmethoden

#### 5.4.1 Review

Die Dokumente wurden jeweils durchgelesen von mindestens einem Teammitglied. Zudem wurde beim Erstellen eine Rechtschreibkorrektur eingesetzt.

#### 5.4.2 Applikations-Tests

siehe [2, 4. Systemtests]

## 5.5 Prüfspezifikationen

### 5.5.1 Checklisten für die Prüfung der Projektdokumente

#### 5.5.1.1 Projektantrag

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	ist nicht erfolgt
2	Alle Punkte behandelt	erfüllt

#### 5.5.1.2 Projektplan

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	ist nicht erfolgt
2	Alle Punkte behandelt	erfüllt
3	Tabelle ausgefüllt	erfüllt

#### 5.5.1.3 Voranalysebericht

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	erfüllt
2	Alle Punkte behandelt	erfüllt
3	Diagramme vorhanden	erfüllt
4	Auswertungen vorhanden	erfüllt

#### 5.5.1.4 Konzeptbericht

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	erfüllt
2	Alle Punkte behandelt	erfüllt
3	Diagramme vorhanden	erfüllt
4	Anwendungsfälle beschrieben	erfüllt
5	Klassendiagramm vorhanden	erfüllt
6	DB-Schema vorhanden	erfüllt

#### 5.5.1.5 Programm siehe [2, 4. Systemtests]

#### 5.5.1.6 Realisierungsbericht

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	hat nicht stattgefunden
2	Alle Punkte behandelt	erfüllt
3	Klassendiagramm aktuell	erfüllt
4	DB-Schema aktuell	erfüllt
5	Benutzerhandbuch vorhanden	erfüllt
6	Supporthandbuch vorhanden	erfüllt
7	Tests erläutert	erfüllt
8	Code im Dokument	erfüllt

#### 5.5.1.7 Einführungsbericht

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	erfüllt
2	Alle Punkte behandelt	erfüllt

#### 5.5.1.8 Projektführung

Nr.	Prüfkriterium	Prüfergebnis, Bemerkungen
1	Rechtschreibung	erfüllt
2	Alle Punkte behandelt	erfüllt

### 5.5.2 Testfalltabellen

Testfälle wurden nur für den Code erstellt, nicht aber für die Dokumente. Die vertige Tabelle der Programmtests zusammen mit den Ergebnissen kann im Realisierungsbericht [2]eingesehen werden.

Die Testcases der Modultests (Unittests) können im Code nachgesehen werden. Diese haben zum Teil komplexe Abhängigkeiten und prüfen einige Funktionen mehrmals oder auf überflüssige Weise.

Nr.	Anwendungsfall	Bemerkungen
1	Tag zu einer Datei hinzufügen.	Wahlweise eines das noch nicht oder schon vorhanden ist.
2	Programm starten	
3	Tag mit Sonderzeichen erstellen	Hat zu gewissen Punkten in der Entwicklung zu Problemen geführt.
4	CLI starten und testen.	Wird nicht erfolgen, da nicht umgesetzt.
5	Tags zu einer bestimmten Datei anzeigen.	Geschieht durch das Selektieren der Datei.
6	Verzeichnis das eine Datei mit Sonderzeichen im Namen enthält wird geöffnet und der Datei ein Tag hinzugefügt.	War vor den Änderungen am DB-Modul nicht möglich.
7	Alle Dateien eines Tags archivieren/versionieren.	
8	Ansicht zwischen hierarchisch und tag-view wechseln.	Dazu gibt es zwei Möglichkeiten.
9	Backup einer Datei anlegen.	
10	Datei Löschen und die Reaktion vom Filesystem-Listener prüfen.	der FS-Listener sollte eine entsprechende Änderung in der Datenbank machen.
11	Datei anlegen und die Reaktion vom Filesystem-Listener prüfen.	

## 6 Konfigurationsmanagementplan (KM-Plan)

### 6.1 Aufzubewahrende Projektergebnisse und ggf. sonstige Dokumente

Alle Ergebnisse können auf Google Code abgerufen werden.

<https://code.google.com/p/project-browser/source/browse/#svn>

### 6.2 Ablagestruktur

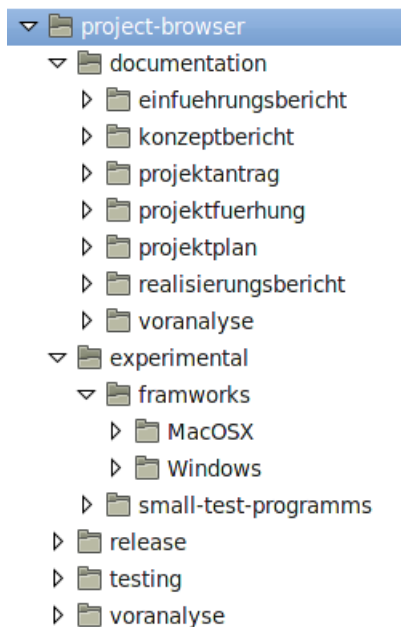


Figure 1: Ablagestruktur

#### 6.2.1 documentation

Alle Elemente der Dokumentation sind im Unterordner "documentation" abgelegt. Dort ist für jedes der Dokumente ein eigener Unterordner abgelegt, der jeweils den Namen des Dokumentes trägt. In diesem Unterordner liegen alle

Dateien, die zum Kompilieren des  $\text{\LaTeX}$ -Dokuments nach PDF notwendig sind, sowie eine kompilierte Version als PDF.

### 6.2.2 Code

Der Code ist im Unterordner experimental abgelegt. Im weiteren Verlauf war die Verschiebung nach "testing" und später nach "release" vorgesehen. In der Praxis hat das aber nie stattgefunden.

Die Python-files liegen direkt im experimental Ordner.

Dieser enthält zudem einen Unterordner "frameworks", in dem weitere Installationsdateien liegen, die zur Benutzung des Programmes unter Windows oder MAC OS X benötigt werden.

## 6.3 Namenskonventionen

Die Elemente der Dokumentation wurden jeweils gleich benannt wie in den verfügbaren Vorlagen, also nach der aktuellen Projektphase.

Die Code-Files wurden nach dem Klassennamen benannt. Diese wiederum wurden nach den Aufgaben der Klassen vergeben.

## 7 Konfigurationsidentifikation

### 7.1 Dokumentation

Ergebniss	Status	Bemerkungen
Projektantrag	fertig	PDF File
Projektplan	fertig	PDF File
Konzeptbericht	fertig	PDF File
Voranalysebericht	fertig	PDF File
Realisierungsbericht	fertig	PDF File
Einführungsbericht	fertig	PDF File
Projektführungsbericht	fertig	PDF File

### 7.2 Applikation (sourcecode)

Ergebniss	Status	Bemerkungen
CLI.py	in Arbeit/nicht fertig	Command line interface für das Programm.
Constant.py	fertig	Enthält Werte die zur Laufzeit vom Programm benötigt werden, sich aber nicht verändern.
DB.py	fertig	Dient als Schnittstelle zur Datenbank
File.py	fertig	Repräsentiert eine Datei auf dem Dateisystem und wird innerhalb der Applikation zum Informationsaustausch verwendet.
FileManager.py	fertig	Dateimanager Funktionen.
FileProperties.py	fertig	Dient zum Setzen von Tags.
FileSystemListener.py	fertig	Mutterklasse der Dateisystem-Listener.
FileSystemListener_Linux.py	fertig	Filesystem-Listener-variante für Linux.
FileSystemListener_Mac.py	in Arbeit/nicht fertig	Filesystem-Listener-variante für MAC OS X.
FileSystemListener_Windows.py	in Arbeit/nicht fertig	Filesystem-Listener-variante für Windows.
GUI.py	fertig	Stellt das GUI dar.
HierarchicalView.py	fertig	Hierarchische Darstellung der Dateien. Erbt von View.
Main.py	fertig	Startpunkt der Applikation.
RepeatTimer.py	in Arbeit/nicht fertig	Backups in regelmässigen Zeitabständen wiederholen.
TagManager.py	fertig	Handeln der Tags.
TagView.py	fertig	Darstellung der Dateien nach Tags. Erbt von View.
Utility.py	fertig	Enthält kleine Funktionen die an verschiedenen Stellen wiederverwendet werden.
View.py	fertig	Mutterklasse für die verschiedenen Views.
gui.glade	fertig	XML Datei. Beschreibt die GTK Oberfläche
fileproperties.glade	fertig	XML Datei.