Projektantrag

Niklaus Hofer, Lukas Knöpfel, Kaleb Tschabold

May 1, 2011

Status	In Arbeit/In Prüfung/**Abgeschlossen**
Projektname	Projektexplorer
Projektleiter	Niklaus Hofer
Auftraggeber	M. Frieden, GIBB
Autoren	Niklaus Hofer
Verteiler	Lukas Knöpfel, Kaleb Tschabolt, Niklaus Hofer

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
0.1	08.02.2011	Gesammelten Text einfügen	Kaleb Tschabold
0.9	08.02.2011	Abgabebereit	Kaleb Tschabold
0.99	May 1, 2011	Transfer nach LaTeX	Niklaus Hofer
1.0	May 1, 2011	Korrekturen	Niklaus Hofer

Definitionen und Abkürzungen

Begriff/ Abkürzung	Bedeutung
CLI	Command Line Interface

References

[1] Markus Studer Martin Frieden. Aufgaben- und terminliste. `http://www2.gibb.ch/iet/module/dokumente/modul306/01_organisation/306_01_Terminliste.doc`, Dec 2008.

[2] Lukas Knoepfel Kaleb Tschabold Niklaus Hofer. Projektplan. Teil der Abgabedokumente, 2011.

Contents

1 Zweck des Dokuments

Projektfreigabe und die erforderlichen Mittel (personell, materiell, finanziell) zu erhalten.

2 Ausgangslage

2.1 Problemstellung

Die heutigen Mglichkeiten Dateien, die organisatorisch eine Einheit bilden, aus strukturellen Gründen aber über verschiedene Ordner auf dem Dateisystem verteilt sind, zu verwalten sind, sehr begrenzt.
Vor Problem wird man auch durch heutige Systeme zur Versionierung von Dokumenten gestellt.

2.2 Anlass und Begründung des Projekts

Wir wollen eine einfache Möglichkeit bieten mehrere Dateien zu gruppieren (z.B. zu einem Projekt) und zu versionieren.

2.3 Rahmenbedingungen

Das Projekt muss bis Ende Mai abgeschlossen sein. Während des Berufsschulunterrichts an der GIBB haben wir jede Woche 4 Lektionen Zeit an dem Projekt zu arbeiten.

2.4 Situationsanalyse

Angesichts der Tatache, dass die Programmiersprache für 2/3 der Beteiligten Neuland darstellt und das Projekt doch recht aufwändig ist, muüste man von eher unguten Bedingungen ausgehen.
Das Team ist aber hochmotiviert und bereit einiges an Zeit zu investieren.

2.5 Erbrachte Vorleistunge

Keine.

3 Ziele und Lösungen

3.1 Zielvorstellungen (kurz- und langfristig)

Ziel des Projektes ist es, eine alternative zu herkömmlichen Dateimanagern zu erstellen, die neue Funktionen zur Verwaltung von Daten (insbesondere im Zusammenhang mit Projekten) bietet.
Folgende Funktionalität soll die Lösung bieten:

- Das Programm bietet drei Masken die den wichtigsten Aufgaben entsprechen:
  - Dateien können mit Tags versehen werden.
  - Dateien lassen sich nach Tags geordnet werden.
  - Im Programm können Projekte definiert werden. Ein Projekt kann mehrere Dateien enthalten, die aber nicht zwingend alle in demselben Ordner liegen müssen.
  - Mit dem Programm können Dateien auch auf herkömmliche Weise in der gewohnten Baumstruktur verwaltet werden.
- Das Programm kann auf Wunsch alte Versionen einer Datei beibehalten. Dabei legt es im betreffenden Ordner ein neues Unterverzeichnis (z.B. _version) an, in dem es alte Versionen ablegt. (Von der Bedienung her vergleichbar mit Apple's Timemachine)

3.2 Mögliche Lösungen

Das Programm soll primär als GUI verfügbar sein. Eine Bedienung über die Kommandozeile sollte aber zwecks Automatisierbarkeit nicht fehlen.
Das Speichern der Daten kann über eine Datei in jedem Ordner oder eine Datenbank realisiert werden. Wir tendieren stark zur Datenbank.
Sowohl Mac OS X als auch moderne Linux Systeme bieten die Möglichkeit, bei Dateisystem-Operationen ein Event an Programme zu senden. Das könnte insbesondere beim erkennen von Dateiverschiebungen die nicht mit dem Programm geschehen hilfreich sein.

Als Datenbank wird SQLite zum Einsatz kommen. Die Datenbank wird als einzelne Datei gespeichert und wird auch von sehr populaeren Projekten (z.B. Firefox) verwendet.
Als Programmiersprache wird Python zum Einsatz kommen.

3.3 Bewertung der Sicherheits- und Datenschutzaspekte

Da das Programm lediglich lokal läuft und für die abgelegten Dateien weiterhin die lokalen Rechte des Dateisystems gelten ist das Programm Datenschutztechnisch kein Problem.
Für die Konsistenz der Dateien wäre es vorteilhaft, wenn das Programm nicht unerwartet alle Files löschen würde.

4 Mittelbedarf

4.1 Sachmittel

Sachmittel werden, neben den zur Verfügung stehenden Computern, keine benötigt.

4.2 Personal

Mindestens zwei der Programmierer müessen sich Kenntnisse und Fähigkeiten im Umgang mit Python im Selbststudium aneignen.
Zudem muss der Umgang mit SQLite erlernt werden.

4.3 Dienstleistungen

Zur Zusammenarbeit und Recherche muss bei jedem Entwickler eine funktionierende Internetverbindung vorhanden sein.
Zudem werden zur Kommunikation und Zusammenarbeit die Officedienste und das Codeverwaltungstool von Google in Anspruch genommen.

5 Planung und Organisation

5.1 Projektorganisation

s Team besteht aus zwei Programmierern und einem Projektmanager (PM).
Die Aufgabe des PMs wird im Verlaufe des Projektes weitergegeben.
Sollte der PM mit seiner Tätigkeit nicht ausgelastet sein, so wird er die Programmierer unterstützen.
Wahrscheinlich werden sich auch einige Spezialisierungen herausbilden. Es ist zum Beispiel denkbar, dass eine Person für das Datenbankdesign zuständig sein wird. Auch die Anpassung/Optimierung an die verschiedenen Betriebssysteme wird wohl jeweils einer einzelnen Person zufallen.

5.2 Anwenderorganisation

Die Anwender sind Einzelpersonen, die an einem lokalen Rechner arbeiten. Mehrere Personen könnten über freigegebene Ordner auf die Dateien zugreifen.

5.3 termine

Die Termine sind von der Lehrkraft vorgegeben und im Terminplan [1] eingetragen.
Zudem haben wir den genauen Zeitplan im projektplan [2] festgehalten.

5.4 Prioritäten

Priorität haen die Schaffung einer GUI Version, die Dokumentation so wie die Überschaubarkeit des Codes.

6 Wirtschaftlichkeit

Zur Zeit gibt es keine Pläne, das Programm kommerziell zu vertreiben. Mindestens ein Beteiligter des Teams spricht sich dagegen aus.
Das Projekt wird unter der Apache 2.0 Lizenz entwickelt und ist somit open source und kann von jedem frei verwendet werden.

Natürlich steht es aber einzelnen Beteiligten aus dem Projekt, oder gar Drittanbietern frei, kommerziellen Support für das Programm anzubieten.
Es kann davon ausgegangen werden, dass das Endprodukt bei der Verwendung einige Vorteile gegenüber herkömmlichen Dateimanagern bieten wird, wodurch Zeit eingespart werden kann.
In welchem Umfang das geschieht und damit verbunden die zu erwartenden Einsparungen sind stark vom Einsatzzweck abhängig und können daher nicht vorhergesagt werden.
Für uns bietet diese Projekt einen hohen Lernwert. Da wir selbst zu der Zielgruppe des Projekts gehören und es auch an unsere Bedürfnisse anpassen können, können wir das Program gut einsetzen.

7 Konsequenzen

7.1 Auswirkungen (organisatorisch, personell, baulich, Vorschriften/Weisungen)

Es stehen neue Wege zum Organisieren und Finden von Dateien zur Verfgung.
Das Organisieren von Projekten die sich über mehrere Ordner verteilen wird einfacher.
Die Beteiligten sammeln Erfahrungen mit den im Projekt eingesetzten Technologien.

7.2 Bei Nichtrealisierung

- Die versprochenen Funktionen stehen nicht zur Verfügung.
- Es gibt verschiedene Lösungen die jeweils teile dessen erfüllen was von unserem Program zu erwarten ist, jedoch keine, die alle Funktionen implementiert. Zudem sind die Lösungen zumeist an eine einzelne Platform gebunden.
- Schlechte Note

7.3 Bei verspäteter Realisierung (gegenüber Wunschtermin)

Da das Program nicht unmitelbar benötigt wird, hätte eine Verspätete Fertigstellung höchstens eine schlechte Note zur Folge.

7.4 Auf Schnittstellen zu anderen Systemen

Da das Rrogram unter anderen ein CLI zur Verfügung stellen soll, kann es auf einfache Weise in scripts oder gar andere Programme integriert werden.

7.5 Qualitätsverbesserungen

Einfacherer Umgang mit Dateien. Mehr Möglichkeiten zur Verwaltung von Dateien auf dem lokalen System.

7.6 Risikobeurteilung

Da das Team viel neues lernen muss und das Projekt recht viele einzelne Teile beinhaltet ist das Risiko, dass es nicht gänzlich fertiggestellt werden kann als beachtlich einzustufen.
Wenn das Programm keine Vorteile gegenüber anderen Dateiverwaltungstools beinhaltet, gerät das Programm in Vergessenheit.

7.7 Ausweichmöglichkeiten

Herkoemmliche Dateimanager (explorer,bridge von adobe,ls,cp,rsync ,mv,rm,find,locate).

8 Antrag

8.1 Bisherige Entscheide

Als Programmiersprache wurde Python gewählt.

8.2 Formulierung des Projektantrags

Nach unserer Erkenntnis könnten wir mit der Voranalyse beginnen. Wir bitten Sie den Antrag für die neue Phase zu genehmigen.