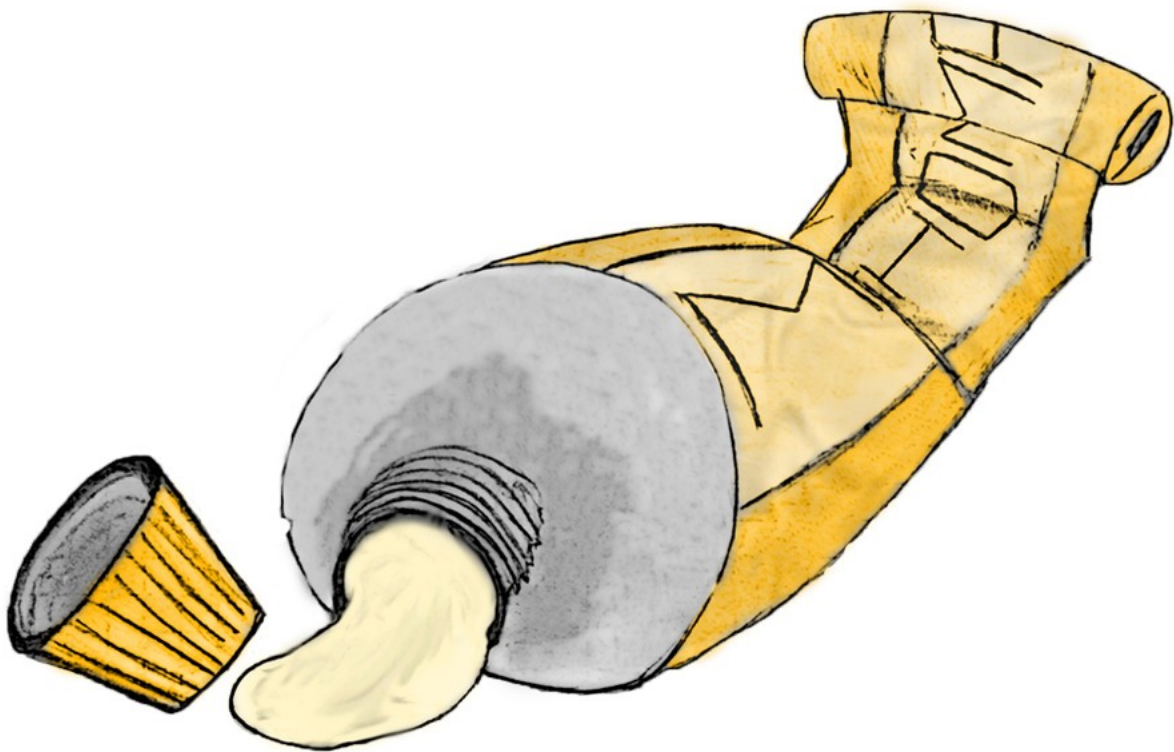


Pflichtenheft

„MathoMadaja“



MATHOMADAJA

| | |
|--|-----------|
| 1 Das Projekt..... | 3 |
| 2 Der Auftraggeber, die Kunden, und andere Projektbeteiligte..... | 3 |
| 2.1 Der Auftraggeber..... | 3 |
| 2.2 Die Kunden..... | 3 |
| 2.3 Andere Projektbeteiligte..... | 3 |
| 3 Anwender des Produkts..... | 3 |
| 4 Randbedingungen..... | 4 |
| 4.1 Lösungsbedingungen..... | 4 |
| 4.2 Partner- oder Schnittstellenprogramme..... | 4 |
| 4.3 Terminbedingungen..... | 4 |
| 5 Namenskonventionen und Definitionen..... | 6 |
| 5.1 Definition aller im Projekt benutzter Namen..... | 6 |
| 5.2 Datenkatalog..... | 6 |
| 6 Relevante Fakten..... | 7 |
| 7 Umfang des Projekts..... | 7 |
| 7.1 Ist-Zustand..... | 7 |
| 7.2 Kontextdiagramm..... | 8 |
| 7.3 Ereignisliste..... | 9 |
| 8 Funktionale Anforderungen..... | 10 |
| 9 Erscheinungs- und Bedienungsanforderungen..... | 13 |
| 10 Leistungsanforderungen..... | 14 |
| 10.1 Sicherheitskritische Anforderungen..... | 14 |
| 10.2 Verfügbarkeitsanforderungen..... | 14 |
| 10.3 Fehler-Toleranzanforderungen..... | 14 |
| 11 Produktveröffentlichung..... | 14 |

1 Das Projekt

Dieses Pflichtenheft behandelt die Software „MathoMadaja“, die im Rahmen der Projektarbeit im 2. Semester Informatik erstellt werden soll. Der Anwender soll später mit der fertig entwickelten Software wiederkehrende Tätigkeiten wie z.B. Verzeichnisse umbenennen, löschen oder auch kopieren ausführen können. Mit diesem Automatismus von „MathoMadaja“ ermöglicht man dem Benutzer sein System ordentlich und überschaubar zu halten. Außerdem muss dieser sich dann nicht mehr an solche manuellen Tätigkeiten erinnern und diese selber ausführen.

2 Der Auftraggeber, die Kunden, und andere Projektbeteiligte

2.1 Der Auftraggeber

Der Auftraggeber ist Frau Prof.,Dipl.-Inform. Beck, Fakultät Informationstechnik, Hochschule Esslingen.

2.2 Die Kunden

Da das Projekt ein Open-Source-Projekt wird und somit allen Interessierten offen steht sind alle Interessierten Kunden des Programms MathoMadaja.

Speziell zur Erfüllung des Projekts stellt Frau Prof.,Dipl.-Inform. Beck, neben Auftraggeberin, auch Kundin der Software dar.

2.3 Andere Projektbeteiligte

Ausgewählte Testpersonen:

Es werden von anderen Fakultäten diverse Testpersonen ausgewählt. Diese Personen sollen jedoch nicht direkt in der Projektumsetzung beteiligt sein, sondern nur die Funktionalitäten und die grafische Oberfläche der Anwendung bewerten. Dabei ist es wichtig Personen mit keinem Kenntnisstand des Projekts auszusuchen. Somit ist eine neutrale Bewertung der Software und der größtmögliche Effekt beim Verbessern der Software möglich.

3 Anwender des Produkts

Wie bereits oben beschrieben sind die Kunden gleichzeitig auch die späteren Anwender des Produkts.

Key-User ist Frau Prof.,Dipl.-Inform. Beck.

Secondary-Users sind alle Interessenten der Software.

Da „MathoMadaja“ ein Open-Source-Projekt sein wird, können sich auch alle Interessenten später am Projekt beteiligen und dieses gleichzeitig verbessern, was auch dem Wunsch des Projektteams entspricht.

4 Randbedingungen

4.1 Lösungsbedingungen

Das Produkt sollte unter gängigen Windows 32 Bit Systemen laufen.

Die meisten PC-Nutzer haben gängige Windows Systeme und würden nicht für das Produkt wechseln

Beurteilungsmaßstab: QS-Manager testet das Produkt auf gängigen Windows-Systemen.

4.2 Partner- oder Schnittstellenprogramme

Anwendung: Autostart von Windows

Begründung: Damit das Programm automatisch mit Windows startet und ausgeführt wird.

- Nur so können dann auch die eingegeben Termine später ausgeführt werden.

Anwendung: Systemzeit von Windows

Begründung: Das Programm greift zur Ausführung bestimmter Termineingaben auf die Systemzeit zu und überprüft das aktuelle Datum

- Führt erst dann das entsprechende Programm aus
- D.h. kein Schutz vor Manipulation der Systemzeit.

Anwendung: Installierte Programme auf Windowssystem

Begründung: Das Programm führt zu bestimmten Zeitpunkten Dateien oder Anwendungen aus. Die Anwendungen müssen dazu natürlich installiert sein und was noch wichtiger ist, auszuführende Dateien müssen mit einer Anwendung verknüpft sein.

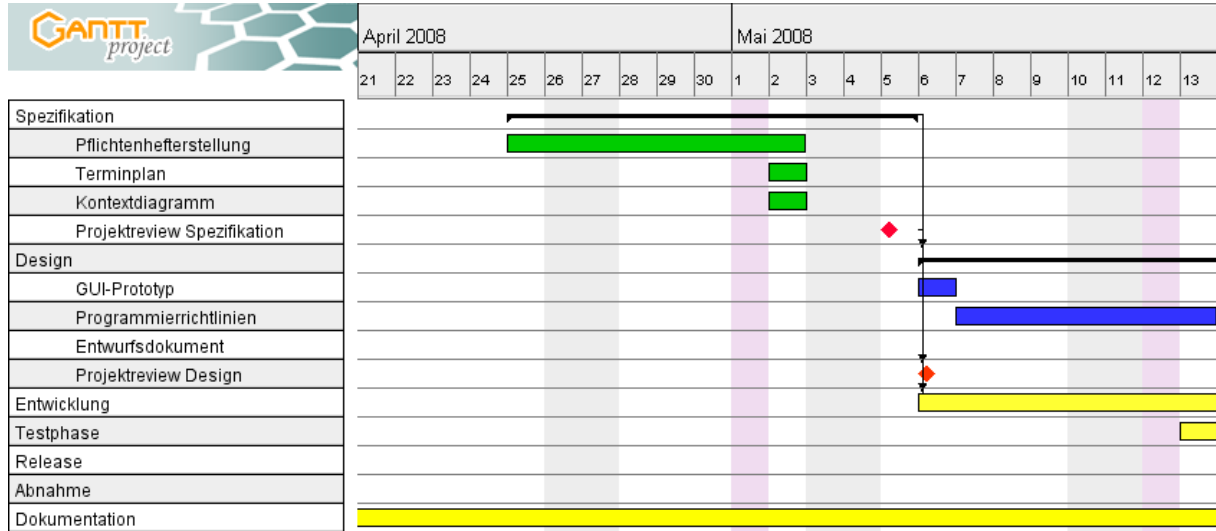
Beispiel: Ausführen von dokument.doc täglich um 12:00 Uhr. Die Endung.doc muss mit Word oder entsprechender Anwendung verknüpft sein.

4.3 Terminbedingungen

Das Produkt muss zur Endabnahme am 10.06.2008 fertig sein.

Zwischenschritte:

Siehe Gantt-Diagramm



5 Namenskonventionen und Definitionen

5.1 Definition aller im Projekt benutzter Namen

MathoMadaja: So heißt das Endprodukt, die Software, die nachher jeder benutzen soll. Der Name entstand aus der Sitzordnung der Projektmitglieder. Markus, Thomas, Marco, David, Jan.

Taskplaner: Der Taskplaner führt zu, vom Benutzer eingestellten Datums-, Uhrzeiten, bestimmte Befehle und Programme aus.

Log-Datei: Sämtliche Vorgänge und Befehle, die das Programm abarbeitet, werden in dieser Datei protokolliert und können zur späteren Überprüfung aufgerufen werden.

Verzeichnis: entspricht dem Windows-Ordner

Backup: Sicherungskopie von Datei, Ordner, Partition oder ganzer Festplatte.

Termin: festgelegte Zeit mit / ohne Datum

Aufgabe: Befehl oder Programmaufruf welchen MathoMadaja abzuarbeiten hat. Auch *Tätigkeit* genannt.

Batch: sequentielle, nicht-interaktive Bearbeitung von Aufgaben.

5.2 Datenkatalog

```
Dpplpkt = „:“ .  
Pkt     = „.“ .  
Pipe    = „|“ .
```

```
Termin = {Stunde + Dpplpkt + Minute + Dpplpkt + Sekunde} +  
        Pipe + {(Tag) + Pkt + (Monat) + Pkt + (Jahr)} .
```

```
Taetigkeit = [Aendern | Loeschen | Kopieren | Ausfuehren] +  
            Pfad + (Datei) .
```

```
Log-Datei-Eintrag = Programmstart + Pipe + Termin + Pipe +  
                  Taetigkeit + Programmende .
```

```
Einstellung-Speichern = Termin + Pipe + Taetigkeit .
```

```
Sitzung-Ende-Speichern = Programmende .
```

6 Relevante Fakten

- Je nach installiertem Windows System kann der zu verwendende Speicherplatz für *Temporary Internet Files* eines Internet Browsers mehrere 100 MB betragen.
- Internet Cookies werden ohne Blocken ständig angelegt, meist ohne Nachfragen.

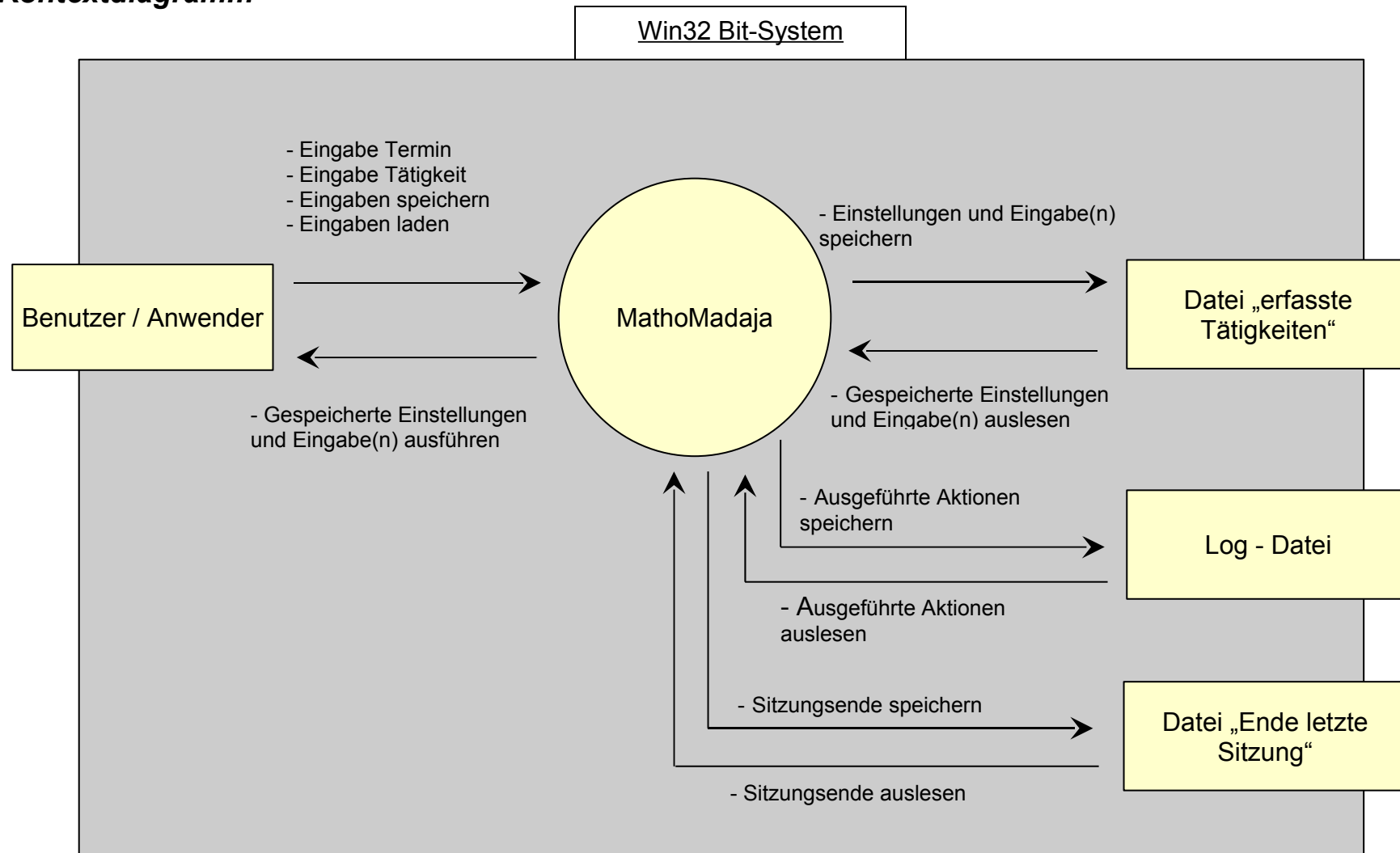
7 Umfang des Projekts

7.1 Ist-Zustand

Standardmäßig ist auf allen neu installierten Windowssystemen der Microsoft Taskplaner installiert. Dieser Taskplaner hat bereits Termin und Aufgabenverwaltung integriert, ist allerdings ziemlich unkomfortabel in der Handhabung. Außerdem können mit dem derzeitigen Taskplaner von Windows nur Programmaufrufe oder Dateiaufrufe ausgeführt werden. Reine Dateioperationen, wie Dateien neu erstellen, Dateien ändern, löschen oder umbenennen ist nur mit anderen Programmen möglich.

Der Vorteil am Windows Taskplaner ist, dass dieser direkt im Windowssystem verankert ist und daher keine speziellen Aufrufrouinen wie MathoMadaja benötigt.

7.2 Kontextdiagramm



Ereignisliste

| Ereignis | Eingabe und Ausgabe | Beschreibung |
|---|---|---|
| Benutzer / Anwender legt Termin und Aktion fest | Termin und Aktion (in) | Termin und Aktion werden gespeichert |
| Benutzer / Anwender speichert Einstellungen | Schreiben erfolgreich (out) <i>„Datei wurde erfolgreich gesichert“</i> | Die Datei wird zum Schreiben der Einstellungen geöffnet, abgespeichert und anschließend geschlossen. |
| Benutzer / Anwender beendet Programm | Sitzungs- und Log-Datei (in) | Sitzungsende wird gespeichert. Textausgabe mit Ende in Log-Datei geschrieben. |
| Benutzer / Anwender startet Programm | Log-Datei (in) Gesicherte Einstellungen (in) | Sitzungsstart wird protokolliert. Auslesen der gesicherten Einstellungen und Tätigkeiten und verarbeiten dieser. |
| Programm soll ausführen | Lese Fehler (out) <i>„Aufruf nicht erfolgreich“</i> Lesen Erfolgreich (out) <i>„Aufruf erfolgreich“</i> | Datei / Verzeichnis soll aufgerufen werden. Protokollierung Log-Datei |
| Programm soll löschen | Lese Fehler (out) <i>„Löschen nicht erfolgreich“</i> Lesen Erfolgreich (out) <i>„Löschen erfolgreich“</i> | Datei / Verzeichnis soll gelöscht werden. Protokollierung Log-Datei |
| Programm soll kopieren | Lese Fehler (out) <i>„Kopieren nicht erfolgreich“</i> Aktion erfolgreich (out) <i>„Kopieren erfolgreich“</i> | Datei / Verzeichnis soll Von-Ort Nach-Ort kopiert werden. Protokollierung Log-Datei |
| Programm soll ändern | Lese Fehler (out) <i>„Änderung nicht erfolgreich“</i> Aktion erfolgreich (out) <i>„Änderung erfolgreich“</i> | Datei / Verzeichnis soll nach Einstellung umbenannt werden. Protokollierung Log-Datei |

8 Funktionale Anforderungen

| Anforderung #: 1 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Ausführung |
|--|--|--|
| Beschreibung: Tätigkeiten müssen zu den genau erfassten (gespeicherten) Terminen ausgeführt werden. | | |
| Begründung: Der Anwender soll sich darauf verlassen können, dass das Programm auch das ausführt, was dieser möchte. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: <ul style="list-style-type: none"> • Tätigkeit erfassen und Zeit simulieren • Tätigkeit erfassen und manuell starten | | |
| Kundenzufriedenheit # (1-5): 5 | Kunden-UN-zufriedenheit # (1-5): 5 | |
| Priorität: 1 | Konflikte: Keine | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

| Anforderung #: 2 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Speichern |
|--|--|---|
| Beschreibung: Die Anwendung soll vom Benutzer eingegebene Termine, Einstellungen erfassen und in einer Datenbank bzw. Datei speichern. | | |
| Begründung: Bei Aufruf der Anwendung sollen wiederkehrende Ereignisse ausgeführt werden, somit müssen Einstellungen gespeichert werden und wieder abgerufen werden können. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: Einstellungen vornehmen → Speichern → Anwendung schließen → Anwendung starten und Einstellungen müssen wieder vorhanden sein und ausgeführt werden. | | |
| Kundenzufriedenheit # (1-5): 5 | Kunden-UN-zufriedenheit # (1-5): 5 | |
| Priorität: 2 | Konflikte: Keine | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

| | | |
|--|--|--|
| Anforderung #: 3 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Protokollierung |
| Beschreibung: Jede ausgeführte Tätigkeit soll genauso mitprotokolliert werden, wie und wann diese ausgeführt wurde. | | |
| Begründung: Der Anwender muss genau nachvollziehen können, was das Programm macht. Somit kann er bei evtl. Fehlern diese dem Projektteam mit einem aussagekräftigen Protokoll melden oder aber die Tätigkeit anpassen. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: Einstellungen vornehmen → Speichern → Anwendung schließen → Anwendung starten und Einstellungen müssen wieder vorhanden sein und ausgeführt werden. | | |
| Kundenzufriedenheit # (1-5): 4 | Kunden-UN-zufriedenheit # (1-5): 3 | |
| Priorität: 3 | Konflikte: Keine | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

| | | |
|---|--|---|
| Anforderung #: 4 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Kopieren |
| Beschreibung: Eine elementar ausführbare Tätigkeit stellt das Kopieren dar. Dateien bzw. Verzeichnisse sollen Von-Ort Nach-Ort kopierbar sein. Die Einstellungen der Orte nimmt der Benutzer im Programm vor. | | |
| Begründung: Das Kopieren von Dateien / Verzeichnissen stellt ein Backup-Funktion dar. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: Kopier-Orte einstellen und Programmlauf starten. In Ordnern nachvollziehen, ob alles wie gewünscht geschehen. Auf Protokollierung muss geachtet werden | | |
| Kundenzufriedenheit # (1-5): 5 | Kunden-UN-zufriedenheit # (1-5): 5 | |
| Priorität: 2 | Konflikte: Keine | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

| | | |
|---|--|--|
| Anforderung #: 5 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Löschen |
| Beschreibung: Eine elementar ausführbare Tätigkeit stellt das Löschen dar. Datei bzw. Verzeichnis, das nicht schreibgeschützt ist, soll löscher sein. Die Einstellungen der Datei bzw. des Verzeichnisses nimmt der Benutzer im Programm vor. | | |
| Begründung: Das Löschen einer Datei / eines Verzeichnisses stellt eine Säuberungs-Funktion dar. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: Lösch-Ort einstellen und Programmlauf starten. In Ordnern nachvollziehen, ob alles wie gewünscht geschehen. Auf Protokollierung muss geachtet werden | | |
| Kundenzufriedenheit # (1-5): 5 | Kunden-UN-zufriedenheit # (1-5): 5 | |
| Priorität: 2 | Konflikte: Schreibgeschützte Dateien / Verzeichnisse | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

| | | |
|---|--|---|
| Anforderung #: 6 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Ändern |
| Beschreibung: Eine elementar ausführbare Tätigkeit stellt das Ändern dar. Dateien bzw. Verzeichnisse, die nicht schreibgeschützt sind, sollen änderbar sein. Die Einstellungen der Datei bzw. des Verzeichnisses nimmt der Benutzer im Programm vor. Das Ändern betrifft hier nicht den Dateiinhalte selber, sondern nur die Benennung. | | |
| Begründung: Die Änderungsfunktion soll helfen Dateien bzw. Verzeichnisse nach Einstellung des Benutzers komfortabel umzubenennen und somit dessen System überschaubar und ordentlich zu halten. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: Änderungsort, sowie Änderungseigenschaften einstellen und Programmlauf starten. In Ordnern nachvollziehen, ob alles wie gewünscht geschehen. Auf Protokollierung muss geachtet werden | | |
| Kundenzufriedenheit # (1-5): 5 | Kunden-UN-zufriedenheit # (1-5): 5 | |
| Priorität: 2 | Konflikte: Schreibgeschützte Dateien / Verzeichnisse | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

| | | |
|--|--|---|
| Anforderung #: 7 | Anforderungstyp: Funktionale Anforderung | Ereignis/Anwendungsfall #: Aufrufen |
| Beschreibung: Eine ausführbare Tätigkeit stellt das Aufrufen von Programmen, Dateien oder sogar Verzeichnissen dar. Die Einstellungen was aufgerufen werden soll, nimmt der Benutzer im Programm vor. | | |
| Begründung: Zu bestimmten Terminen kann es erforderlich sein diverse Aufrufe zu starten. | | |
| Auftraggeber: Projekt-Team | | |
| Beurteilungsmaßstab: Datei, Verzeichnis oder Programm welches aufgerufen werden soll einstellen und Programmlauf starten. Nach Erscheinen des entsprechenden Fensters funktioniert die Tätigkeit. Auf Protokollierung muss geachtet werden | | |
| Kundenzufriedenheit # (1-5): 3 | Kunden-UN-zufriedenheit # (1-5): 1 | |
| Priorität: 4 | Konflikte: Dateierweiterungen die mit keinem Programm verknüpft sind | |
| Referenz zur Erklärung: Anwenderdokumentation | | |

9 Erscheinungs- und Bedienungsanforderungen

Es soll eine einfache und intuitiv bedienbare grafische Oberfläche erstellt werden. Dabei soll diese an Windows-Standards angelehnt werden, das bedeutet der Benutzer soll sich nicht in der Bedienung umstellen müssen und Dialoge sollen wieder erkennbar sein.

Außerdem soll das Programm „sprechend“ sein. Das heißt der Benutzer soll direkt erkennen können, welche Funktionalität hinter welchem Element in der Software steckt. Auch soll der Anwender die erzeugte Protokolldatei direkt verstehen und auswerten können, ohne lange in Dokumentationen nach Bedeutungen suchen zu müssen.

Der Benutzer soll wiederkehrende Bedienungen im Programm selber automatisieren können – wie z.B. das Speichern direkt beim Einstellen einer Tätigkeit oder eben über Aufruf der Speicher-Funktionalität.

Beurteilungsmaßstab: Zur Beurteilung der grafischen Oberfläche und deren Funktionalitäten wäre es sinnvoll, sich gezielt Testpersonen auszusuchen, die mit dem Projekt nichts zu tun haben und diese dann einen vorbereiteten Fragebogen bzw. ein vorbereitetes Testprotokoll ausfüllen zu lassen.

10 Leistungsanforderungen

10.1 Sicherheitskritische Anforderungen

Die Anwendung führt diverse Systemoperationen wie Kopieren, Ändern und Löschen aus. Aus diesem Grund muss gewährleistet werden, dass nicht unbeabsichtigt Dateien oder Verzeichnisse zerstört werden. Die aussagekräftige Log-Datei soll veranschaulichen was das Programm getan hat und noch wichtiger was eigentlich eingestellt wurde. So können ungewünschte Operationen wieder rückgängig gemacht oder abgeändert werden. Dem Anwender muss bewusst sein, dass mit der Anwendung systemkritische Operationen ausgeführt werden können.

Beurteilungsmaßstab: Bevor das Programm überhaupt veröffentlicht wird, müssen ausreichend viele Tests vorgenommen werden. Die Funktionalitäten müssen in allen Varianten ausprobiert werden und es muss festgestellt werden, ob keine Beschädigungen am Testsystem vorkamen.

10.2 Verfügbarkeitsanforderungen

Das Programm soll 24 Stunden, 365 Tage im Jahr nutzbar sein. Sinnvoll ist es auf Rechnern zu installieren, die auch die ganze Zeit über laufen, damit alle erfassten Tätigkeiten zu den gespeicherten Terminen abgearbeitet werden, aber die Anwendung soll genauso auf lokalen Rechner verfügbar sein.

10.3 Fehler-Toleranzanforderungen

Wird ein Termin verpasst soll beim nächsten Aufruf der Anwendung ein Hinweis ausgegeben werden, der beim Benutzer nachfragt, ob die erfasste Tätigkeit nachgeholt werden soll und somit direkt beim Programmstart abgearbeitet werden soll.

11 Produktveröffentlichung

Die Software wird online als herunterladbares gepacktes Archiv im ZIP-Format verfügbar sein. Abhängigkeiten des Programms im Zusammenhang mit DLLs werden so gelöst, dass die entsprechenden DLLs dem Archiv beigelegt werden. Als Referenz zur Installation und den Dokumentationen wird eine ReadMe-Datei beigelegt. Die Dokumentationen werden, wie das Programm selber, auch im Archiv enthalten sein. Das Ziel ist also, dass es einem Anfänger möglich ist, ohne Vorkenntnisse die Software direkt in Betrieb zu nehmen.