

GUI-Grafikgenerator für mobiles Erfassungssystem (PDA)

Bachelorarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2010

Autor(en): Michel Rothmund
Betreuer: Prof. Eduard Glatz
Projektpartner: Dr. Jürgen Held ETH Spin-off Systemergonomie, Zürich, ZH
Experte: Robert Pajetta, Credit Suisse AG Zürich, ZH
Gegenleser: Prof. Dr. Peter Heinzmann

Inhaltsverzeichnis

1. Aufgabenstellung	1–5
2. Erklärung	2–9
3. Abstract	3–10
4. Management Summary	4–11
4.1 Ausgangslage	4–11
4.1.1 Einführung ins FIT – System	4–11
4.1.2 Projektziel	4–12
4.1.3 Vergleichbare Produkte / Lösungen	4–12
4.2 Vorgehen	4–13
4.2.1 Entwicklungsprozessmodell	4–13
4.2.2 Realisierung	4–13
4.2.3 Involvierte Personen	4–13
4.3 Ergebnisse	4–14
4.3.1 Resultat	4–14
4.3.2 Zielerreichung	4–14
4.3.3 Erkenntnisse	4–15
4.4 Ausblick	4–15
4.4.1 Einsatzfähigkeit der Software	4–15
4.4.2 Verbleibende Probleme	4–15
4.4.3 Ausbau- und Weiterentwicklungsmöglichkeiten	4–15
5. Technischer Bericht	5–16
5.1 Einleitung und Übersicht	5–16
5.1.1 Ausgangslage	5–16
5.1.2 Aufgabenstellung	5–16
5.2 Ergebnisse	5–17
5.2.1 Zeichnen	5–17
5.2.2 Erfassen	5–17
5.2.3 Löschen	5–18
5.3 Ursachen nicht erreichter Funktionen	5–18
5.3.1 Dateiformat	5–18
5.3.2 Signieren	5–18
5.4 Funktionen nach Anforderung	5–19
5.5 Probleme	5–20
5.5.1 Gezeichnete Figur nicht zusammenhängend	5–20
5.5.2 Überschreiben der Zeichnung	5–20
5.5.3 Wenn auf „PictureBox“ gezeichnet wird, wird alles schwarz	5–20
5.5.4 Glättung nicht so wie vorgestellt	5–21
5.5.5 Komplexe Mouse Event Logik	5–21
5.5.6 Rubber Problem	5–21
5.5.7 Letzte erfasste Koordinate anzeigen	5–22
5.5.8 “Out of Memory Exception” auf iPAC	5–22
5.5.9 Datenfile Dateiformat	5–22
5.5.10 Auflösung Emulator und iPAC unterschiedlich	5–23
5.6 Schlussfolgerungen	5–24
5.6.1 Bewertung der Ergebnisse	5–24

5.6.2	Alternative Lösungen	5-24
5.6.3	Was noch zu tun ist.....	5-24
5.6.4	Fazit.....	5-24
6.	Anforderungsspezifikation.....	6-25
6.1	Einleitung.....	6-25
6.2	Funktionale Anforderungen	6-26
6.2.1	Übersicht.....	6-26
6.2.2	Zeichnungsfunktionen	6-27
6.2.3	Erfassungsfunktionen.....	6-29
6.2.4	Löschfunktionen.....	6-30
6.2.5	Signierfunktionen	6-31
6.2.6	FK_1: Kopierschutz.....	6-32
6.3	Nichtfunktionale Anforderungen	6-33
6.3.1	Benutzbarkeit.....	6-33
6.3.2	Zuverlässigkeit.....	6-33
6.3.3	Mengenanforderungen.....	6-33
6.3.4	Anforderungen an die Schnittstellen.....	6-33
6.3.5	Erweiterbarkeit.....	6-34
6.3.6	Wartbarkeit.....	6-34
6.3.7	Anforderungen zur Entwicklungsumgebung	6-34
6.3.8	Sprachen	6-34
6.3.9	Anforderungen der System - Tests.....	6-34
6.4	Use Cases	6-35
6.4.1	UC – Diagramm	6-35
6.4.2	UC1: Neues Layout erstellen	6-36
6.4.3	UC2: Layout bearbeiten	6-37
6.4.4	UC3: Aufnahme ohne Layout	6-38
6.4.5	UC4: Aufnahme mit Layout	6-39
6.4.6	UC5: Datenfile signieren mit layout	6-40
6.4.7	UC6: Datenfile Signieren ohne Layout	6-41
6.4.8	UC7: Datenfile löschen	6-42
6.4.9	UC8: Layout löschen.....	6-42
7.	Analyse	7-43
7.1	Einleitung.....	7-43
7.2	Domain Model.....	7-43
7.2.1	Domain Object Forms	7-43
7.2.2	Domain Objects PD.....	7-44
7.3	Zustandsdiagramme und Externes Design.....	7-44
7.4	Glättung (Bezier)	7-45
7.4.1	Implementation	7-45
8.	Internes Design	8-46
8.1	Einleitung.....	8-46
8.2	Software - Architektur.....	8-46
8.2.1	Forms.....	8-46
8.2.2	PD - Package.....	8-47
8.3	Dateiaufbau.....	8-47
8.3.1	Klassen - Diagramm.....	8-48
8.3.2	Form Klassen.....	8-49
8.3.3	PD Dateien.....	8-53
8.4	Sequenzdiagramme	8-56

8.4.1	SD1: Erstellen eines neuen Layouts	8-56
8.4.2	SD2: Bearbeiten eines Layouts	8-57
8.4.3	SD3: Aufnehmen eines Datenfiles mit Layout	8-58
8.4.4	SD4: Löschen eines Datenfiles	8-59
8.4.5	SD5: Löschen eines Layouts.....	8-60
8.5	Design – Entscheidungen	8-61
8.5.1	Warum wurde mit C# Programmiert.	8-61
8.5.2	Logik in Form Klasse.....	8-61
9.	Tests	9-62
9.1	Einleitung.....	9-62
9.2	Unit Tests.....	9-62
9.3	Systemtests	9-63
9.3.1	IPAC-214	9-63
9.3.2	Emulator Windows Mobile 6.0.....	9-65
9.3.3	Bug List.....	9-66
10.	Projekt Management Plan	10-67
10.1	Einleitung.....	10-67
10.2	Software Development Plan	10-67
10.2.1	Projektübersicht	10-67
10.2.2	Lieferumfang	10-67
10.2.3	Projektorganisation.....	10-68
10.2.4	Projektmanagement	10-70
10.2.5	Risk List	10-72
10.2.6	Personalplanung	10-72
10.2.7	Entwicklungsprozess.....	10-73
10.3	Iteration Plan.....	10-74
10.3.1	Übersicht Iterationen	10-74
10.3.2	Arbeitspakete	10-75
10.3.3	Meilensteine	10-94
10.3.4	Zeitplan	10-94
10.4	Iteration Assesment.....	10-95
10.4.1	Soll- und Ist- Aufwand der Iterationen und Arbeitstakte	10-95
10.4.2	Inception	10-96
10.4.3	Draw	10-96
10.4.4	Record	10-96
10.4.5	Extend.....	10-96
10.4.6	Sign.....	10-97
10.4.7	Transition	10-97
10.4.8	Hilfspakete	10-97
10.4.9	Fazit	10-97
11.	Users Manual.....	11-98
11.1	Introduction	11-98
11.2	Requirements	11-98
11.3	Start Applikation.....	11-98
11.4	Main Form	11-98
11.5	Record.....	11-99
11.6	Draw	11-100
11.7	Delete	11-101
12.	Installationsanleitung	12-102

12.1	Einrichten Entwicklungsumgebung	12-102
12.2	Installation des FIT-Systems auf PDA	12-102
12.2.1	Erstellen eines CAB Projektes.....	12-102
13.	Persönliche Berichte	13-103
13.1	Projektarbeit	13-103
13.2	Gelerntes.....	13-103
13.3	Negative Erfahrungen	13-103
13.4	Positive Erfahrungen	13-103
14.	Anhang	14-104
14.1	Sitzungsprotokolle	14-104
14.2	Meilensteinrapporte	14-119
15.	Glossar	15-125
16.	Referenzen	16-128

1. Aufgabenstellung

Aufgabenstellung zur Bachelorarbeit FS 2010

„GUI- Graphikgenerator für mobiles Erfassungssystem (PDA)“

Gruppe: Michel Rothmund

Ausgangssituation

Das FIT-System (flexible Interfacetechnik) dient für die mobile Erfassung beobachtbarer Ereignisse in Industrie und Wissenschaft und besitzt derzeit weltweit circa 80 Lizenznehmer. Es besitzt eine patentierte Eingabetechnik durch GUIs die vom Benutzer direkt erstellt werden. Die bestehende Applikation läuft auf Palm PDAs und ist in C geschrieben [1].

Aufgabe

Die GUI-Graphikfunktionalität soll für neue mobile Geräte neu entwickelt werden, welche das Windows Mobile Betriebssystem besitzen. Die Interaktionen für das Zeichnen eines GUI (Symbole, Beschriftungen, Farben) sollen möglichst intuitiv realisiert werden. Die Eingaben des Benutzers sollten dabei automatisch angepasst werden. D.h. geometrische Elemente (Linien, Kreise, etc.) sowie eingegebene Beschriftungen müssen von der Software erkannt und automatisch korrigiert (geglättet) werden. Die so optimierten GUIs sind im PDA für die spätere PCAuswertung als Bitmap zu verwalten. Mobile Geräte (HP iPAQ Pocket PC) und ggf. Entwicklungssysteme werden vom Auftraggeber zur Verfügung gestellt. Für die Programmiersprache der Neuentwicklung soll eine begründete Wahl getroffen werden, d.h. es gibt keine Vorgaben, solange die geforderte Gesamtfunktionalität realisiert werden kann.

Erwartete Resultate

Als Resultat dieser Arbeit soll eine demonstrierbare Software auf Stufe „stabiler Prototyp“ vorliegen, ergänzt mit einer Dokumentation, die wichtige Überlegungen der Analyse, des Entwurfs und der Implementierung festhält.

Die entwickelte Software muss auf einem Testsystem installiert sowie bei Verwendung der dokumentierten Entwicklungs-Tools auch generiert werden können. Es ist, so weit nötig, eine Installations- bzw. Konfigurationsbeschreibung bereitzustellen.

Ergänzend soll ein nach den unten stehenden Anforderungen aufgebauter Bericht entstehen. Abzugeben sind drei CD-ROMs (MS Windows kompatibel), die die Software und den Bericht in elektronischer Form enthalten (Gesamtbericht als PDF-Datei und Originaldateien des benutzten Textsystems). Der Bericht ist zusätzlich zweimal in gedruckter Form mit Ringbindung (kein Ordner!) bereitzustellen. Basis für die Bewertung der Arbeit stellt der gedruckte Bericht sowie die auf der CD bereitgestellte Software dar.

Geforderte Berichtsinhalte:

1. *Resultatdokumentation:*

Sie beschreibt alle Resultate der Arbeit und soll projektbegleitend gemäss dem für die Arbeit gewählten Vorgehensmodell erstellt werden. Sie soll alle wichtigen Informationen enthalten, die ein Ingenieur benötigt, der die Arbeitsresultate ohne vorgängige Kenntnis des Projekts weiterverwenden möchte.

2. *Projektdokumentation:*

Sie umfasst jegliche Dokumentation, die sich auf die Durchführung der Studienarbeit bezieht. Dazu gehören auch ein *nachgeführter Projektplan* (Arbeitspakete, Zeitplan mit Meilensteinen, Plan- und Ist-Aufwände), *Kurzprotokolle aller Besprechungen* und ein *Projektschlussbericht* (was wurde erreicht bzw. nicht erreicht, was ist in der Durchführung gut/schlecht gelaufen, Schlussfolgerungen, Ausblick auf mögliche Zusatz- und Nachfolgearbeiten). Im Bericht ist die Aufteilung der Arbeit innerhalb der Gruppe auszuweisen (thematisch und in Anzahl Arbeitsstunden).

Im weiteren soll die Gliederung und der Inhalt des Berichts den Regelungen der Abt. Informatik entsprechen (siehe Richtlinien unter den Web-Seiten der Abt. Informatik [2]). Teildokumente sollen derart in den Bericht integriert werden, dass ein hierarchisches Gesamt-Inhaltsverzeichnis mit durchgängiger Seitennummerierung ermöglicht wird. Alle obligatorischen Berichtsinhalte müssen im gedruckten Bericht vorhanden sein (nicht nur auf Abgabe-CD). Datenblätter, Handbücher und allfällige Kopien anderer Dokumente, die nicht selbst erstellt wurden, aber für weitere Arbeiten mit den Systemen nützlich sind, sollen *separat* zum Gesamtbericht abgegeben werden (Abgabeform frei; evtl. auch nur auf Abgabe-CD). Berichtsinhalte, die nicht selbst erarbeitet wurden [3], sind mit ihrer Quelle zu bezeichnen [4] (gilt auch für Bilder!).

[1] Bericht der Studienarbeit SS04: Matthias Thut, Normen Scheiber
„Draw – FIT – System“

[2] URL: <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html>
(dort Link „Anleitung“ anklicken)

[3] URL: http://www.plagiarism.org/learning_center/what_is_plagiarism.html

[4] URL: http://www.plagiarism.org/learning_center/preventing_writing.html#write

Termine

22. Februar 2010	Arbeitsbeginn
18 Juni 2010	Abgabe des Berichts an den Betreuer (bis 12:00)

Weitere Termine siehe Terminangaben auf dem HSR-Web (intern).

Betreuung

Betreuer: Prof. Eduard Glatz, Email: eglatz@hsr.ch Während der Durchführung der Arbeit findet nach Möglichkeit regelmässig jede Woche eine Besprechung mit dem Betreuer statt. Dazu werden entsprechende Termine bei Arbeitsbeginn festgelegt.

Kontaktperson Firmenpartner:

PD Dr. Jürgen Held, ETH Spin-off Systemergonomie

Technoparkstr. 1, 8005 Zürich, Tel.: 044 632 26 16, Email: jheld@ethz.ch

Rapperswil, den 20. Feb 2010



Prof. Eduard Glatz

GUI-Graphikgenerator für mobiles Erfassungssystem (PDA)



Hintergrund:

Das FIT-System (flexible Interfacetechnik) dient für die mobile Erfassung beobachtbarer Ereignisse in Industrie und Wissenschaft und besitzt derzeit weltweit circa 80 Lizenznehmer. Es besitzt eine patentierte Eingabetechnik durch GUIs die vom Benutzer direkt erstellt werden.

Aufgabe:

Die GUI-Graphikfunktionalität soll für neue mobile Geräte neu entwickelt werden, welche das Windows Mobile Betriebssystem besitzen. Die Interaktionen für das Zeichnen eines GUI (Symbole, Beschriftungen, Farben) sollen möglichst intuitiv realisiert werden. Die Eingaben des Benutzers sollten dabei automatisch angepasst werden. D.h. geometrische Elemente (Linien, Kreise, etc.) sowie eingegebene Beschriftungen müssen von der Software erkannt und automatisch korrigiert (geglättet) werden. Die so optimierten GUIs sind im PDA für die spätere PC-Auswertung als Bitmap zu verwalten. Mobile Geräte (HP iPAQ Pocket PC) und ggf. Entwicklungssysteme werden vom Auftraggeber zur Verfügung gestellt.

Auftraggeber:

Jürgen Held - ETH Spin-off Systemergonomie
Technoparkstr. 1 - 8005 Zürich - 044 632 26 16 - jheld@ethz.ch

2. Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt sind oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.

Ort, Datum:

Name, Unterschrift:

3. Abstract

Das neue FIT-System ist eine Software für einen Windows Mobile 6.0 Handheld PC. Für die Entwicklung und den Test der Software wurde ein HP iPAC-214 verwendet. Ältere Versionen der Software waren für den Apple Newton und den PalmPilot entwickelt. Der Benutzer kann mit dieser Software Daten erfassen, die dazu dienen, Arbeitsschritte zu analysieren und zu optimieren. Um dies zu tun, kann der Benutzer eine von ihm erstellte Zeichnung verwenden oder eine Schablone auflegen. Der Benutzer kann für den zu analysierenden Arbeitsablauf seine eigene Benutzeroberfläche zeichnen. Beim Zeichnen wird er durch eine optionale Glättungsfunktion, sowie der Möglichkeit bestimmte Figuren (Box, Circle, Polygon) zu zeichnen, von der Software unterstützt. Will der Benutzer Daten erfassen, kann er die entsprechende Zeichnung laden. Der Benutzer tippt mit dem Stift auf die Symbole der Zeichnung (oder Schablone). Die Koordinaten des angetippten „Punktes“ werden von der Software inklusive der aktuellen Uhrzeit abgespeichert. Zur Auswertung der aufgenommenen Daten kann das Textfile auf einen Desktop Computer geladen werden. Im Benutzer Handbuch¹ zum FIT-System wird die genaue Funktionsweise der Software beschrieben.

¹ Kapitel 11 „User Manual“ dieses Dokumentes

4. Management Summary

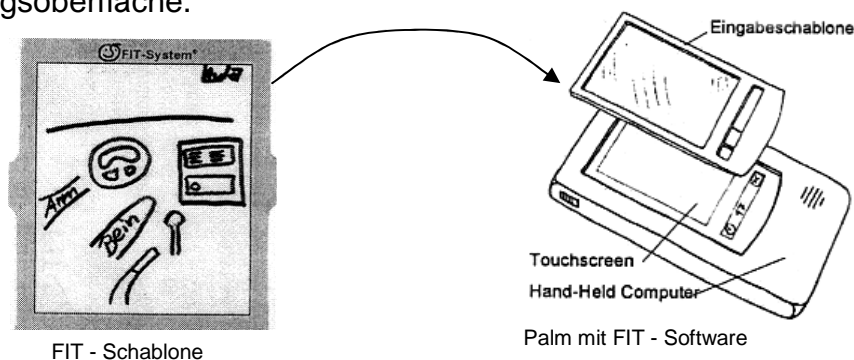
4.1 Ausgangslage

4.1.1 Einführung ins FIT – System²

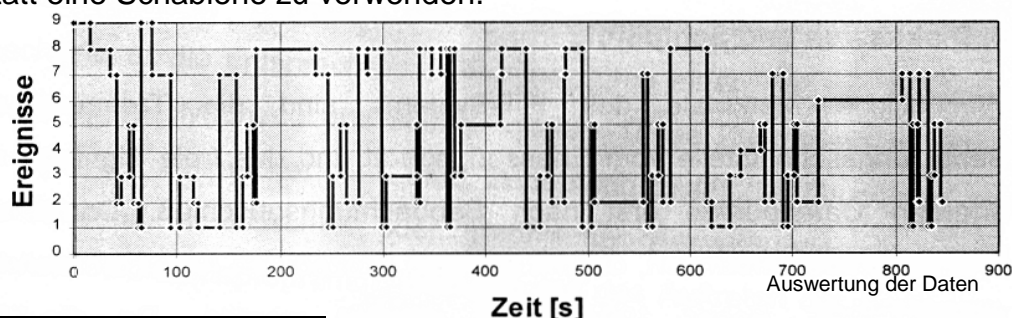
Unter dem Namen FIT- System (Flexible Interface Technik) hat die Arbeitsgruppe für Systemergonomie des Instituts für Hygiene und Arbeitsphysiologie der ETH Zürich ein System zur Erfassung und Auswertung von Prozessereignissen entwickelt. Bei diesem Verfahren werden mittels eines Handheld - Computers bestimmte Abläufe aufgezeichnet und später am PC ausgewertet und analysiert. Ziel des FIT – Systems ist die Analyse gewisser Arbeitsschritte und deren Optimierung.

Anwendungsverfahren

Der Anwender zeichnet eine seinen Bedürfnissen entsprechende Schablone (durchsichtige Folie), deren Symbole einer Tätigkeit, Bewegung, Haltung, Form, Anordnung usw. entsprechen. Er gestaltet sich so seine individuelle Benutzungsoberfläche.



Diese Schablone wird nun auf einem (mit dem FIT – Programm ausgestatteten) Handheld - Computer angebracht, welcher zudem über einen „Touchscreen“ verfügt. Das Erfassen der Ereignisse erfolgt nunmehr durch das Antippen der entsprechenden Bereiche auf der Schablone. Die FIT – Software protokolliert zu jedem Eintrag den Zeitpunkt und die Koordinaten des angetippten Punktes. Ist die Erfassung aller Ereignisse des gesamten Prozesses abgeschlossen, so können die aufgenommenen Daten auf den PC übertragen und dort ausgewertet werden. Es besteht auch die Möglichkeit, eine Zeichnung direkt mit dem FIT-System zu erstellen anstatt eine Schablone zu verwenden.



² Siehe Studienarbeit SS04 Draw-FIT-System Kapitel 4.1.1. Dieses Kapitel wurde komplett, abgesehen vom nicht mehr funktionierenden Link zu SmileDesing, übernommen.

4.1.2 Projektziel

Ziel dieser Arbeit ist es nun das gegenwärtige FIT - System so weiter zu entwickeln, dass es auf modernen Handheld Computern mit Windows Mobile 6.0 läuft. Dem Benutzer sollen nun auch Zeichnungshilfen wie Glättung der Zeichnung, sowie das Erkennen von Figuren (Kreis, Rechteck, etc) angeboten werden. Ausserdem sollten die Möglichkeiten von modernen Betriebssystemen genutzt werden, die dem Benutzer ermöglichen, in verschiedenen Farben zu Zeichnen. Falls genügend Zeit vorhanden ist, wäre es noch schön die Signierfunktion vom Desktop PC auf den Handheld zu verlagern.

4.1.3 Vergleichbare Produkte / Lösungen

Das Verfahren des FIT – Systems auf dem Handheld – Computer ist gegenwärtig einmalig. Deshalb hat es sein Erfinder „Jürgen Held“ patentieren lassen. Für das Erstellen von Zeichnungen gibt es auf Handheld Computer bereits verschiedene Programme.

PocketPictures³

- 6 verschiedene Stiftdicken, beliebige Farben
- Kopier-, Ausschneid-, Verschiebungsfunktionen
- Zeichnen von Figuren (Linien, Rechtecke, Ellipsen)
- Keine Glättungsfunktion und keine „Figuren“ (Polygon, Text)
- Speichern (PNG, JPG, GIF, BMP)

Vspainter

- Kostenpflichtige⁴- und Freeware⁵- Version (VsPainter LE) mit verringertem Leistungsumfang.
- Freeware (VsPainter LE) keine fortgeschrittene Bildbearbeitung. (Appetizer)
- Meiner Meinung nach für Hobby-Künstler geeignet
- Keine Glättung
- Viele für FIT-Applikation ungeeignete Figuren (Herz, Noten, Sterne)

PDACraft Paint⁶

- Funktionalität wie Paint für Windows
- Vorschau beim Zeichnen von Figuren schneller als beim FIT-System
- Keine Glättung, Kein zeichnen von Polygonen
- Kann nur im bmp Format speichern.

³ <http://code.google.com/p/pocketpicture/downloads/list>

⁴ <http://www.vetasoft.com/index.php?index=12&language=en>

⁵ <http://vspainter-le.softonic.de/pocketpc>

⁶ <http://pdacraft.com/paint.php>

4.2 Vorgehen

4.2.1 Entwicklungsprozessmodell

Für die Realisierung des Projektes wurde die agile Version des iterative Entwicklungsprozess RUP (Rational Unified Process) angewendet. „Agile RUP“⁷ ist eine dem Projekt angepasst abgespeckte Version des RUP die sich auf wenige Artefakte konzentriert. Der Prozess führt über stets erweiterte, funktionstüchtige Prototypen zum Produkt. Insgesamt ist das Projekt in 6 Iterationen aufgeteilt.

4.2.2 Realisierung

Das Projekt wurde nach dem obigen Prozessmodell verwirklicht, dazu wurden Etappenweise folgende Teilschritte durchgeführt und ausgearbeitet.

Iteration 1		Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	
1.	2.	3.	4.	5.	6.	7.	8.

1. Planung & Technologiestudium
2. Anforderungsspezifikationen & Analyse
3. Zeichnungsfunktionen implementieren
4. Aufnahmefunktion implementieren und mit Zeichnungsfunktion kombinieren
- 5- Löschfunktionen für Dateien implementieren
- 6 Zeichnungsfunktion um Glättung und Farben erweitern.
- 7 Systemtest- & Integrationstests durchführen
- 8 Dokumentationsabschluss

4.2.3 Involvierte Personen

Die Projektarbeit wurde von dem Erfinder des FIT – Systems Jürgen Held in Auftrag gegeben. Alle Entscheide bezüglich Funktionalität und Design wurden von Ihm gefällt oder bestätigt. Als Betreuer wirkte Eduard Glatz als kompetenter-, technischer- und allgemeiner- Berater. Michel Rothmund war für die Lösungsfindung und deren Realisierung verantwortlich.

⁷ Agil RUP Siehe Vorlesung SE3 von L. Müller AgileMethoden_Folien_2_proSeite2.pdf

4.3 Ergebnisse

4.3.1 Resultat

Aus der Projektarbeit resultiert eine funktionsfähige FIT-Software für Handheld – Computers mit folgendem Funktionsumfang:

- **Zeichnungs – Modus;** In welchem eine neue Zeichnung erstellt oder eine schon bestehende Zeichnung geladen werden kann. Zum zeichnen bieten sich die Funktionen: Stift (dünn, mittel, dick), Radiergummi, „Alles Löschen“ „Undo“ Funktion, Glättungsfunktion (Bezier) sowie Shape (Box, Circle, Polygon, Text) an. Beim fertig stellen kann die virtuelle Zeichnung als .jpg File abgespeichert werden. Es kann auch ein beliebiges .jpg File geladen und verändert werden.
- **Aufnahme – Modus;** In diesem Bereich kann zuerst gewählt werden, ob mit einer physischen Schablone (Template) oder virtuellen Folie (Layout) gearbeitet werden soll. Danach wird eine Aufnahme datei erstellt und man kann beliebig lang die gewünschten Ereignisse aufnehmen. Der letzte auf dem Display angetippte Punkt, ist mit einem roten Kreuz dargestellt. Die Anzahl bereits erfasster Ereignisse wird immer angezeigt. Wird ein Layout verwendet, so kann dieses während der Aufnahme mit dem Zeichnungsmodus editiert werden. Ist die Aufnahme zu Ende, kann Sie abgespeichert werden.
- **Lösch – Modus,** in diesem Modus können die einzelnen virtuellen Zeichnungen oder aufgenommenen Aufnahme dateien wieder gelöscht werden.

Die erstellte Software läuft auf Windows Mobile 6.0 Handheld PCs von HP. Um es auf anderen Handhelds laufen zu lassen muss der Kopierschutz umprogrammiert oder entfernt werden. Es sollte auch auf älteren Windows Mobile Versionen lauffähig sein insofern das .net compact 2.0 framework auf dem Handheld installiert ist.

4.3.2 Zielerreichung

Das erstellte Produkt erfüllt alle in der Anforderungsspezifikation definierten zwingenden Anforderungen. Zudem wurden die meisten „Wunsch“ – Funktionen implementiert. Es wurde ein gewisser Grad der Glättung erreicht. Ausserdem stellt die Applikation Figuren (Rechteck, Kreis, Polygon, Text) zur Verfügung wo die Glättung nicht ganz ausreicht.

Eine optimale Glättung die, die „Figuren“ überflüssig gemacht hätte, wurde wegen der entsprechenden Komplexität und des Zeitlimits nicht erreicht. Die „Wunsch“ - Funktion „Signieren“ der Aufnahme dateien wurde mit Einverständnis des Kunden weggelassen⁸, um die Zeichnungshilfe mit Glättung und Figuren möglichst gut zu optimieren.

⁸ Siehe Sitzungsprotokoll vom 7.5.2010 im Anhang dieses Berichtes Kapitel 14.1

4.3.3 Erkenntnisse

Die wichtigste Erkenntnis war, dass gute Dokumentation sehr wichtig ist, damit jemand anders die Arbeit übernehmen kann. Das .fit Dateiformat war nirgends beschrieben. Ich hab zwar das Format vom Auftraggeber bekommen, aber die Daten waren nicht im Klartext abgespeichert, sondern irgendwie serialisiert. Es konnte kein Rückschluss erlangt werden wie diese Daten gespeichert oder gelesen wurden. Dadurch musste das Dateiformat einfach auf ASCII Klartext geändert werden.

4.4 Ausblick

4.4.1 Einsatzfähigkeit der Software

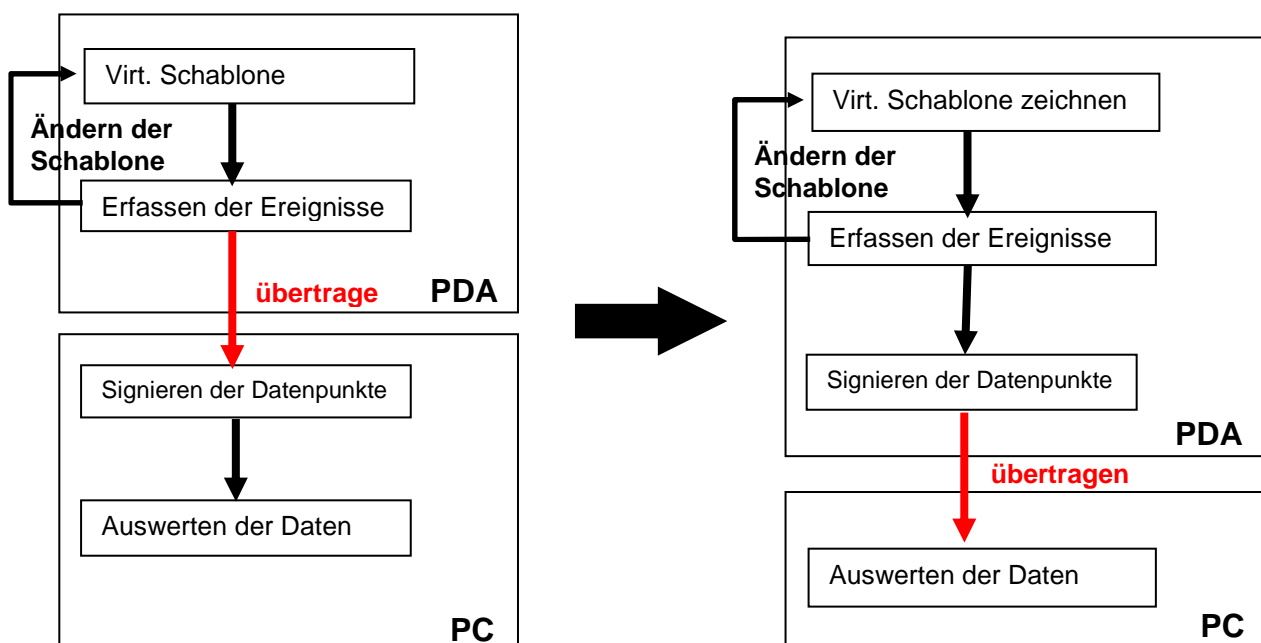
Die Software ist vollständig einsetzbar, alle implementierten Funktionen arbeiten fehlerlos. Die „Vorschau Funktion“ beim Zeichnen von Figuren (Rechteck, Kreis, etc.) ist etwas langsam. Der Code der Software benötigt wenig Speicherplatz (28KB). Die gesamte Signierfunktion, um die Aufnahme Dateien später auswerten zu können, muss immer noch auf dem PC durchgeführt werden. Außerdem muss der FIT-Manager angepasst werden um das neue .fitdb Dateiformat (ASCII Klartext) lesen zu können.

4.4.2 Verbleibende Probleme

- FIT-Manager muss angepasst werden um .fitdb Dateiformat lesen zu können.
- Der Kopierschutz macht die Software spezifisch für HP Handheld auf denen die iPAQUtil.dll installiert ist.
- Bei Geräten, die eine andere Auflösung haben als der iPAC-214, könnte es zu Problemen kommen. Z.B. Korrektur bei den Polygonpunkten bei ähnlichen x oder y Koordinaten, oder Auswirkung der Glättungsfunktion.

4.4.3 Ausbau- und Weiterentwicklungsmöglichkeiten

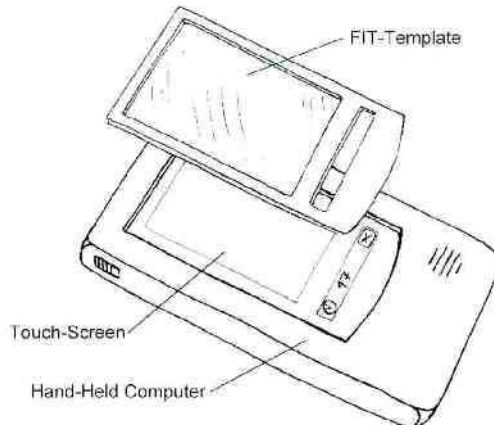
Die Glättungsfunktion könnte weiter verbessert und optimiert werden. Weiter könnte die Software mit der gewünschten, jedoch nicht realisierten „Signierfunktion“ ausgebaut werden. (siehe Bild)



Technischer Bericht

5.1 Einleitung und Übersicht

5.1.1 Ausgangslage



Der Benutzer kann seine eigene Eingabeoberfläche gestalten, indem er eine von Hand gezeichnete Schablone auf den PDA legt, oder indem er mit Hilfe des FIT-Systems ein Layout auf dem Handheld zeichnet. Die gezeichneten Symbole können vom Benutzer selbst beschriftet und definiert werden. Falls beim Erfassen die Option mit „Template“ (Schablone) gewählt wurde, wird die Schablone auf den PDA Screen aufgelegt. Bei der Option Layout wird die entsprechende Zeichnung geladen und angezeigt. Wird ein Symbol angetippt, werden die Koordinaten des Punktes mit der aktuellen Uhrzeit gespeichert. Nachdem die Daten auf den Desktop übertragen und die Datenpunkte dort „Signiert“ (In eine Gruppe Zusammengefasst, der eine selbst definierte Bedeutung(Ereignis) zugeteilt wird). Nun können die Daten aufgrund der Auftrittshäufigkeit und Zeitunkte der „Ereignissen“ ausgewertet werden.

5.1.2 Aufgabenstellung

Die alte Software läuft auf veralteten PDAs wie der PalmPilot. Das Ziel ist es nun die Software auf moderne PDAs mit dem Betriebssystem Windows Mobile 6.0 zu portieren. Die neue Applikation soll mindestens die gleiche Funktionalität wie die alte zur Verfügung stellen. Ausserdem ist es dem Kunden sehr wichtig, dass der Benutzer beim Zeichnen unterstützt wird. Dies soll durch eine Glättung der Zeichnung erreicht werden. Der Benutzer soll gerade Linien, Rechtecke und Kreise zeichnen können, die schön dargestellt werden. Ausserdem sollte man auch Farben integrieren.

Die Aufgabe gliedert sich in folgende Tätigkeiten:

- Einarbeitung in die PDA Programmierung
- Wahl der Programmiersprache
- Studie Glättung
- Entwicklung
- Dokumentation der Arbeit

5.2 Ergebnisse

5.2.1 Zeichnen

Als erstes wurde die Zeichnungsfunktion implementiert mit der gleichen Funktionalität wie in der alten Applikation. Dies beinhaltet im wesentlichen Eingabe und Überprüfung des einzigartigen Zeichnungsnamens, Zeichnen mit drei verschiedenen Stiftsticken, Radiergummi und „Alles löschen“ Funktion für Korrekturen und Speicherung der Zeichnung. In der neuen Applikation wird die Zeichnung als .jpg File gespeichert. Erst gegen den Schluss der Arbeit wurde diese Funktionalität erweitert. Der Benutzer kann nun die Farbe (Blau, Gelb, Grün, Rot, Schwarz) des Zeichnungsstiftes verändern. Es wurde eine einstufige „Undo“ Funktion eingebaut, in welcher das zuletzt Gezeichnete wieder gelöscht wird. Eine optionale Glättungsfunktion basierend auf Bezier Kurven wurde in vereinfachter Weise implementiert. Diese Funktion reichte nicht aus, um perfekte Rechtecke oder Kreise zu zeichnen. Deshalb wurde noch ein Funktion „Shape“(Box, Circle, Polygon, Text) erstellt, die es ermöglicht schöne Figuren auf einfache Weise zu zeichnen. Der Benutzer wählt Figur aus und berührt mit dem Stift den „Screen“, um den Startpunkt der Figur festzulegen. Bewegt er den Stift wird ihm eine Vorschau der gewünschten Figur angezeigt. Entfernt er den Stift vom „Screen“, wird die Figur, anhand des letzten Berührungspunktes des Stiftes gezeichnet.

5.2.2 Erfassen

Beim Erfassen von Daten muss zuerst ein Name fürs Datenfile angegeben werden, der auf Einzigartigkeit überprüft wird. Ist dies nicht der Fall, wird der Benutzer aufgefordert diesen zu ändern. Sobald dies geschehen ist, wird ein Textfile mit dem entsprechenden Namen und der Endung .fitdb im Ordner „MyDocuments/fit“ angelegt. Danach muss der Benutzer entscheiden, ob er ein „Template“ (Schablone) oder ein Layout(Zeichnung als .jpg) verwenden möchte. Verwendet er eine Schablone, muss er diese nun auflegen, andernfalls wird ihm die Zeichnung angezeigt. Der Benutzer kann nun ein Ereignis erfassen, indem er mit dem Stift auf ein Symbol tippt. Nun werden die Koordinaten und der Zeitpunkt dieses Ereignisses aufgezeichnet und ins Datenfile geschrieben. Verwendet der Benutzer ein Layout(Zeichnung), kann er jederzeit in den Zeichnungsmodus wechseln, um Änderungen vorzunehmen. Sobald er den Zeichnungsmodus wieder verlässt, kann er in den Erfassungsmodus zurück und kann seine Arbeit fortsetzen. Der zuletzt erfasste Punkt wird mit einem roten Kreuz markiert. Bevor der Benutzer die Erfassung beendet, wird er noch gefragt, ob er das wirklich will. Unten wird noch das .fitdb Dateiformat für Datenfiles beschrieben.

Dateiformat .fitdb

Besteht aus ASCII Zeichen die im Klartext abgespeichert sind und in folgendem Format aufgebaut sind:

Indikator<TAB>Name<TAB>Grösse<TAB>Eintrag1 <TAB>...EintragN<TAB>!

<TAB>	Tabulator
Indikator	Identifiziert den Gerätetyp von welchem die Daten kommen. Für den iPAC-214 ist dies „@Windows480536@“. Als erstes das Betriebssystem und dann die Auflösung des Gerätes. Der Indikator für den Palm Pilot ist: „@PalmPilot@“
Name	Name des Files
Grösse	Anzahl Einträge des Files
Eintrag	Format: „xKoordinate-yKoordinate-Sekunden“ Sekunden = Anzahl Sekunden seit der Erstellung des Files.
!	Definiert Ende des Files

5.2.3 Löschen

Der Benutzer hat die Möglichkeit, die durch ihn erstellten Zeichnungen oder Datenfiles wieder zu löschen. Er kann dies auch mit dem Windows Explorer tun. Die Files befinden sich alle im Verzeichnis „MyDocuments\fit“. Der Benutzer muss angeben, ob er ein Datenfile oder ein Layout(Zeichnung) löschen will. Nun wird dem Benutzer eine Liste mit den Dateien angezeigt. Wenn er eine Datei auswählt, werden ihm einige Angaben zur Datei angezeigt und er hat die Möglichkeit es zu Löschen. Bevor die Datei gelöscht wird, wird der Benutzer noch angefragt ob er die Datei wirklich löschen will.

5.3 Ursachen nicht erreichter Funktionen

5.3.1 Dateiformat

Das Dateiformat der Datenfiles wurde so implementiert wie vom Kunden vorgegeben. Es wurde zwar vom FIT-Manager erkannt aber nicht korrekt gelesen. Die Datenfiles wurden von der alten Applikation irgendwie serialisiert abgespeichert. Wie das genau ablief, konnte nicht herausgefunden werden, da das Dateiformat nicht dokumentiert war und aus dem Code nicht genügend Rückschlüsse gezogen werden konnten. Deshalb wurde in Übereinkunft mit dem Kunden⁹ das Dateiformat .fitdb erstellt.

5.3.2 Signieren

Am Anfang des Projekts war geplant, die bis anhin auf einem Desktop PC durchgeführte wurde, auf den PDA zu verlagern, falls noch genug Zeit zur Verfügung stehen würde. Da durch die Probleme beim Dateiformat die Glättungsimplementierung verzögert wurde, wollte der Kunde das die Signier-Funktion ausgelassen wird so, dass mehr Zeit für die Implementierung der Glättung zur Verfügung steht.

⁹ Siehe Sitzungsprotokoll vom 7.5.2010 im Anhang Kapitel 14.1

5.4 Funktionen nach Anforderung

Funktion	Muss/Wunsch	erfüllt	nicht erfüllt
Batteriezustand anzeigen	E ¹⁰	X	
Anzeigen des Zeichnungsnamens	E	X	
Erstellen einer neuen Zeichnung	E	X	
Speichern einer Zeichnung (mit Datum)	E	X	
Laden einer bestehenden Zeichnung	E	X	
Stift-Funktion dünn	E	X	
Stift-Funktion mittel	E	X	
Stift-Funktion dick	E	X	
Radiergummi-Funktion	E	X	
"Alles löschen"-Funktion	E	X	
Glättung der Zeichnung	S ¹¹	X	
Farbauswahl	S	X	
Undo-Funktion	S	X	
Shape Box Zeichnen	S	X	
Shape Circle Zeichnen	S	X	
Shape Polygon Zeichnen	S	X	
Shape Text Zeichnen	W	X	
Wechseln zwischen erfassen/editieren	E	X	
Anzeigen Namen Datenfile und Anzahl Punkte	E	X	
Speichern eines Datenfile	E	X	
Anzeigen der Option FE1 oder das Folie verwendet wird	E	X	
Zeichnungsfunktion Im Modus Erfassen integrieren	E	X	
Letzter Punkt anzeigen	W ¹²	X	
Löschen eines Datenfiles	E	X	
Löschen eines Zeichnungsfiles	E	X	
Bestätigungsfenster anzeigen	E	X	
Auswahl des Datenfile	W		X
Auswahl des Zeichnungsfile oder Folie	W		X
Anzeigen der Punkte aus Datenfile	W		X
Polygon um Symbol zeichnen	W		X
Polygon „Benennung ändern“	W		X
Polygon „Löschen“	W		X
Polygon „verschieben“	W		X
Aufforderung um Namen des Icons einzugeben	W		X
Speichern des Datenfiles mit Iconnamen	W		X
Kopierschutz	S	X	

¹⁰ Essenzielle Anforderung

¹¹ Sehr wichtige Anforderung

¹² Wunsch Anforderung

5.5 Probleme

Während des Projektverlaufs sind einige unvorhergesehene Probleme aufgetreten, welche im nachfolgenden kurz beschrieben werden. (Begriffe in „“ sind .net Klassen)

5.5.1 Gezeichnete Figur nicht zusammenhängend

Zuerst wurde bei jedem „Mouse down“ oder „Mouse move“ Ereignis ein Punkt an der entsprechenden Koordinate gezeichnet. Da diese Ereignisse jedoch nicht so häufig auftraten wie ich mir vorgestellt hatte, standen alle diese Punkte etwa in einem Abstand von 2mm und waren nicht miteinander verbunden.

Lösungsansatz 1

Ich vergrößerte einfach die gezeichneten Punkte. Dies löste das Problem vorübergehend. Wenn der Stift jedoch sehr schnell bewegt wurde, waren die Punkte nicht mehr miteinander verbunden.

Lösungsansatz 2

Die Applikation merkt sich die Koordinate des letzten Maus Ereignisses und verbindet diesen mit den Koordinaten des neuen Ereignisses. Dieser Ansatz erwies sich als sehr gut und wird fürs Zeichnen ohne Glättung verwendet.

5.5.2 Überschreiben der Zeichnung

Zuerst wurde direkt aufs Graphics Objekt der Form gezeichnet. Dies hatte zur Folge, dass jedes Mal wenn ein Popup die Zeichnung verdeckte, die Zeichnung überschrieben wurde.

Lösung

Eine „PictureBox“ wurde eingefügt, welche in C# CF 2.0¹³ leider kein „Graphics“ Objekt zum Zeichnen hat. Jedoch kann ihr ein „Image“ übergeben werden. Dazu wurde eine „Bitmap“ benutzt, die von „Image“ abgeleitet ist und ein „Graphics“ Objekt zum Zeichnen besitzt. Dadurch wird das Gezeichnete immer in der „Bitmap“ der „PictureBox“ gespeichert und wird nicht mehr überschrieben.

5.5.3 Wenn auf „PictureBox“ gezeichnet wird, wird alles schwarz

Sobald ich auf die „Bitmap“ der „PictureBox“ gezeichnet habe anstatt auf die „Form“ wurde der ganze Inhalt der „PictureBox“ einfach schwarz.

Lösung

Bei einem Versuch habe ich die Farbe des Zeichnungsstiftes geändert. Als ich anfang zu Zeichnen wurde der Hintergrund zwar noch schwarz, jedoch zeichnete der Stift mit der geänderten Farbe auf die schwarze Fläche. Nun erkannte ich was das Problem war. Die „Bitmap“ war vor dem Zeichnungsstart „null“. Als angefangen wurde zu Zeichnen, wird sie initialisiert und jeder Pixel auf Schwarz gesetzt. Dieses Problem konnte gelöst werden indem man auf dem „Graphics“ Objekt der „Bitmap“ die „Clear“ Methode aufruft und die Farbe „Color.Transparent“ übergibt.

¹³ .net Compact Framework -> Abgespeckte Version des .net Frameworks für Software Entwicklung auf mobilen Geräten

5.5.4 Glättung nicht so wie vorgestellt

Das .net Framework stellt GDI+ Zeichnungsfunktionen im Namespace „System.Drawing“ bereit. Diese beinhalten auch Glättungsfunktionen. Unter anderem gibt es eine „Smoothing“ Option und auch die Möglichkeit Bezierkurven zu zeichnen. Im .net Compact Framework waren jedoch beide dieser Möglichkeiten nicht vorhanden. Ich stiess auf eine vereinfachte „Bezier“ Implementierung im Internet und anhand dieser erstellte ich meine Version. Diese glättete nicht so, wie ich mir erhofft hatte. Es bestand kaum ein Unterschied zum normalen Zeichnen.

Lösung

Das Problem war, dass die Punkte die für das Zeichnen der Bezierkurve nötig waren zu nahe beieinander lagen. Das ganze wurde gestreckt indem nur jede vierte „Mouse Move Event“ Koordinate zum Zeichnen von Bezierkurven berücksichtigt wird. Dies führte bereits zu recht guten Resultaten. Damit war es jedoch nicht möglich schöne Kreise oder Rechtecke zu zeichnen. Deshalb wurde die Funktion „Shapes“ (Box, Circle, Polygon, Text), eingeführt um solche Figuren zu zeichnen.

5.5.5 Komplexe Mouse Event Logik

Je nachdem ob die Glättungsoption aktiviert ist oder nicht muss bei einem Mouse Event anders reagiert werden. Und nun kommen noch die „Shapes“ (Box, Circle, Polygon, Text) vier weitere verschiedene Vorgänge hinzu.

Lösung

Bei den „Shapes“ wurde Polymorphismus angewendet um das Ganze nicht allzu zu komplizieren. Je nachdem welche Figur es ist wird die entsprechende Logik aufgerufen um sie zu Zeichnen. Die Mouse Event Logik ist jedoch immer noch recht komplex und könnte wahrscheinlich noch weiter verbessert werden.

5.5.6 Rubber Problem

Dieses Problem ist aus der Komplexen Mouse Event Logik entstanden. Der Rubber funktioniert wie ein normaler Stift (Pen) mit der Ausnahme, dass seine Farbe transparent ist. Wenn nun ein „Shape“ oder die „Bezier“ Glättungsoption ausgewählt war und danach der Rubber ausgewählt wurde, wurde einfach ein transparentes „Shape“ oder eine transparente geglättete Zeichnung gezeichnet.

Lösung

Die Mouse Event Logik musste durch boolesche Prüfungen weiter ausgebaut werden.

5.5.7 Letzte erfasste Koordinate anzeigen

Beim Erfassen von Ereignissen sollen die Koordinaten des zuletzt erfassten Ereignisses mittels eines Kreuzes angezeigt werden. Wenn ich ein Kreuz auf eine „Bitmap“ zeichne dann ist dieses dort gespeichert und ich bringe es nicht mehr weg. Nun soll aber wenn ich ein neues Ereignis erfasse dieses angezeigt werden und das letzte Kreuz soll verschwinden.

Lösung

Dieses Problem wurde gelöst indem eine Temporäre „Bitmap“ mit dem Inhalt des Originals erstellt wurde auf die das Kreuz gezeichnet wird. Bei jedem Ereignis wird eine Kopie der originalen „Bitmap“ erstellt und das Kreuz auf diese gezeichnet und diese temporäre „Bitmap“ dann der „PictureBox“ zugewiesen. Diese Technik wurde oft wiederverwendet um ähnliche Probleme zu lösen. Folgende Probleme wurden damit gelöst:

- Anzeigen der Zeichnung und danach der geglätteten Version
- Vorschau beim zeichnen der Figuren(Shapes)
- Anzeige des eingegebenen Textes beim „Shape“ „Text“.

Sie ist jedoch recht Speicherintensiv und hat auch zum nächsten Problem geführt.

5.5.8 “Out of Memory Exception” auf iPAC

Als ich das Erfassen von Ereignissen zum ersten Mal auf dem iPAC testete, stürzte das Programm nach dem 24sten Punkt, mit einer „Out of Memory Exception“ ab.

Lösung

Überall wo temporäre Kopien von „Bitmaps“ erstellt wurden musste der „Garbage Collector“ aufgerufen werden, um den Speicher aufzuräumen.

5.5.9 Dateifile Dateiformat

Das Dateiformat der Datenfiles war in der Dokumentation der Vorgängerapplikation nicht beschrieben. Vom Auftraggeber habe ich Informationen erhalten, wie das Format aufgebaut ist. Ich habe dies übernommen und die Daten in diesem Format im Klartext abgespeichert. Die FIT-Manager Applikation konnte die Datei zwar erkennen und lesen, jedoch wurden die Datenpunkte an einem völlig falschen Ort dargestellt.

Lösungsversuch

Ich konnte erkenne, dass die Daten der alten Applikation nicht im Klartext abgespeichert wurden. Aus dem Code der Alten Applikation und des Fit-Managers konnte ich erkennen, dass das Datenfile als .pdb Datei abgespeichert wurde und im FIT-Manager mittels einer Klasse „CArchive“ von MFC gelesen wurde. Ich konnte herausfinden das „CArchive“ Objekte auf die Festplatte serialisiert und, dass dessen Serialisierungsmechanismus nicht mit demjenigen von C# kompatibel ist. Da der Kunde und ich dieses Problem nicht lösen konnten, kamen wir überein, dass die neue Applikation die Daten wie bis anhin speichern soll und dass der FIT-Manager später angepasst wird.

5.5.10 Auflösung Emulator und iPAC unterschiedlich

Auf Grund der doppelt so grossen Auflösung des iPAC gegenüber dem Emulator entstanden verschiedene Probleme, so dass sich das Programm auf den unterschiedlichen Plattform anders oder unerwartet verhielt. (z.B. Glättung, Korrektur beim zeichnen von Polygonen)

Lösung

Gewisse Grössen mussten angepasst werden, so dass es auf dem iPAC so funktioniert wie der Benutzer erwartet. Dass es dann auf dem Emulator nicht mehr so gut läuft kann hingenommen werden.

5.6 Schlussfolgerungen

5.6.1 Bewertung der Ergebnisse

5.6.1.1 Allgemein

Die Arbeit kann als gut gelungen betrachtet werden. Alle Essenziellen und sehr wichtigen Anforderungen für den Kunden wurden erfüllt. Einzig ist schade, dass das Datenfile nicht so abgespeichert werden konnte, wie in der alten Applikation. Durch Kommunikation mit dem Kunden konnten viele keine Änderungen und Wünsche des Kunden übernommen werden. Auch war der Kunde sehr entgegenkommend wenn etwas nicht so ging wie er es am liebsten hätte. Es ist ein Produkt entstanden, dass dem Kunden einen wirklichen Mehrwert zur Vorgängerapplikation liefert.

5.6.1.2 Source Code

Beim erstellen des Source Codes wurde darauf geachtet, dass er einheitlich und sauber dargestellt ist. Weiter wurde darauf geachtet die Variablen gut zu benennen, was beim generierten Teil des Source Code, jedoch nicht immer der Fall ist. Wo es erforderlich war wurde er auch ausreichend dokumentieren um es einem unabhängiger Entwickler einfacher zu machen, sich in den Code einzuarbeiten und diesen zu verstehen.

5.6.1.3 Dokumentation

Sie sollte für jeden Entwickler klar und verständlich sein.

5.6.2 Alternative Lösungen

Es wäre auch Möglich gewesen diese Applikation in C++ oder in Java zu erstellen. Auch wäre es möglich gewesen andere Glättungsalgorithmen zu verwenden um die Glättung zu realisieren.

5.6.3 Was noch zu tun ist

In einem weiteren Projekt könnte sicherlich die Signierfunktion vom Desktop PC auf den PDA übertragen werden. Auch könnte die Software angepasst werden für PDAs die nicht von HP sind oder Geräte mit anderen Auflösungen. Die Funktion „Glättung“ könnte man auch noch verbessern.

5.6.4 Fazit

Es war toll diese Applikation in C# zu. Man wurde wirklich gut durchs Visual Studio unterstützt. Es ist eine wirklich gute Applikation entstanden die den Benutzer beim Zeichnen auf verschiedene Arten gut unterstützt.

6. Anforderungsspezifikation

6.1 Einleitung

In diesem Dokument werden alle funktionalen und nichtfunktionalen Anforderungen für das Projekt Draw –FIT festgelegt. Die Anforderungen wurden aufgrund der Sitzung mit dem Auftraggeber, der Aufgabenstellung und der alten Draw-FIT Applikation ausgearbeitet. Die Anforderungen wurden weitgehend von der Anforderungsspezifikation der Vorgängerapplikation übernommen. Nach Wunsch des Auftraggebers wurden einige wenige neue Anforderungen hinzugefügt, neu priorisiert und einige Formulierungsänderungen durchgeführt. Die Anforderungen werden in die drei Kategorien unterteilt:

Essentielle Anforderungen (E)

Anforderung muss erfüllt sein um die gleiche Funktionalität wie im Vorgängerprojekt zur Verfügung zu stellen.

Sehr wichtige Anforderungen(S)

Verbessert Mängel der Vorgängerapplikation die dem Kunden wichtig sind. Diese Anforderungen werden nur im Notfall nicht erfüllt und werden nur mit Einverständnis des Auftraggebers und anhand eines wichtigen Grundes weggelassen.

Wunsch Anforderungen (W)

Wunschanforderungen werden nach Möglichkeit ebenfalls im Rahmen des Projektes durchgeführt. Bleibt jedoch nicht genügend Zeit dazu können sie mit einer entsprechenden Begründung weggelassen werden.

Die Funktionen müssen so formuliert sein, dass man am Ende des Projekts überprüfen kann, ob die Anforderungen erfüllt wurden oder nicht.

6.2 Funktionale Anforderungen¹⁴

6.2.1 Übersicht

ID	Funktion	Meilenstein	Priorität
FZ_1	Batteriezustand anzeigen	M2	E
FZ_2	Anzeigen des Zeichnungsnamens	M2	E
FZ_3	Erstellen einer neuen Zeichnung	M2	E
FZ_4	Speichern einer Zeichnung (mit Datum)	M2	E
FZ_5	Laden einer bestehenden Zeichnung	M2	E
FZ_6	Stift-Funktion dünn	M2	E
FZ_7	Stift-Funktion mittel	M2	E
FZ_8	Stift-Funktion dick	M2	E
FZ_9	Radiergummi Funktion	M2	E
FZ_10	„Alles Löschen“ Funktion	M2	E
FZ_11	Glättung der Zeichnung	M5	S
FZ_12	Farbauswahl	M5	S
FZ_13	„Undo“ Funktion	M5	S
FZ_14	Shape Box Zeichnen	M5	S
FZ_15	Shape Circle Zeichnen	M5	S
FZ_16	Shape Polygoy Zeichnen	M5	S
FZ_17	Shape Text Zeichnen	M5	W
FE_1	Wechseln zwischen Erfassen/Editieren	M3	E
FE_2	Anzeigen des Datenfilenamens und Anzahl Punkte	M3	E
FE_3	Speichern eines Datenfile	M3	E
FE_4	Anzeigen der Option FE1 oder das Folie verwendet	M3	E
FE_5	Zeichnungsfunktion im Modus Erfassen integrieren	M3	E
FE_6	Letzer Punkt markiert	M3	W
FL_1	Löschen eines Datenfiles	M4	E
FL_2	Löschen eines Zeichnungsfiles	M4	E
FL_3	Bestätigungsfenster anzeigen	M4	E
FS_1	Auswahl des Datenfile	M6	W
FS_2	Auswahl des Zeichnungsfile oder Folie	M6	W
FS_3	Anzeigen der Punkte aus Datenfile	M6	W
FS_4	Polygon um Symbol zeichnen	M6	W
FS_4.1	Polygon Benennung ändern	M6	W
FS_4.2	Polygon Löschen	M6	W
FS_4.3	Polygon verschieben	M6	W
FS_5	Aufforderung um Namen des Icons einzugeben	M6	W
FS_6	Speichern des signierten Datenfiles mit Iconnamen	M6	W
FK_1	Kopierschutz	M7	S

¹⁴ Anforderungen die gleich sind wie in der Vorgängerapplikation wurden von dort übernommen. Siehe DrawFIT[1].pdf Kapitel 6.2

6.2.2 Zeichnungsfunktionen

Wie in der Vorgänger Version des FIT-Systems, wird eine Zeichnungsfunktion angeboten. Der Benutzer hat somit weiterhin die Möglichkeit, eine Zeichnung direkt auf dem IPAQ zu erstellen, anstatt eine Folie zu verwenden.

6.2.2.1 FZ1: Batteriezustand anzeigen M

Während dem sich der Benutzer im Zeichnungsmodus befindet, wird ihm der aktuelle Ladungszustand der Batterie bzw. des Akkus angezeigt.

6.2.2.2 FZ2: Anzeigen des Zeichnungsnamens M

Damit der Benutzer weiss, in welcher Zeichnung er sich momentan befindet, wird der von ihm gewählte Zeichnungsname (ohne Dateierweiterung) angezeigt.

6.2.2.3 FZ3: Erstellen einer neuen Zeichnung M

Der Benutzer hat die Möglichkeit eine neue Zeichnung zu erstellen. Dazu wird er aufgefordert einen Zeichnungsname einzugeben. Existiert bereits eine Zeichnung mit dem vom Benutzer gewählten Namen, erscheint eine entsprechende Fehlermeldung und der Benutzer muss einen neuen Zeichnungsname eingeben.

6.2.2.4 FZ4: Speichern einer Zeichnung M

Hat der Benutzer eine neue Zeichnung erstellt und er verlässt den Zeichnungsmodus, wird die vom Benutzer erstellte Zeichnung mit dem aktuellen Datum gespeichert.

6.2.2.5 FZ5: Laden einer bestehenden Zeichnung M

Der Benutzer hat die Möglichkeit, eine bestehende Zeichnung zu laden und diese zu ändern. Dazu wird ihm eine Liste aller vorhandenen Zeichnungen angezeigt, von welcher er diejenige wählen kann, die er editieren möchte. Zu jedem Eintrag wird noch das Erstellungsdatum der entsprechenden Zeichnung angezeigt.

6.2.2.6 FZ6: Stift-Funktion dünn M

Zum zeichnen der Symbole steht dem Benutzer ein Stift zur Verfügung. Dieser Stift erlaubt es dem Benutzer, eine Freihandzeichnung zu erstellen.

6.2.2.7 FZ7: Stift-Funktion mittel M

Diese Funktion erfüllt dieselben Anforderungen wie die Funktion FZ6. Der Unterschied ist nur, dass die durch den Stift erzeugte Linie mittlere Dicke besitzt.

6.2.2.8 FZ8: Stift-Funktion dick M

Diese Funktion erfüllt dieselben Anforderungen wie die Funktion FZ6. Der Unterschied ist nur, dass die durch den Stift erzeugte Linie dicker ist.

6.2.2.9 FZ9: Radiergummi-Funktion M

Diese Funktion dient dazu, bereits gezeichnete Linien wieder zu löschen.

6.2.2.10 FZ10: "Alles Löschen"-Funktion M

Durch wählen dieser Funktion kann der Benutzer die ganze Zeichnung wieder löschen. Bevor die Zeichnung jedoch gelöscht wird, wird nach einer Bestätigung für den Löschvorgang gefragt.

6.2.2.11 FZ_11: Glättung der Zeichnung S

Wenn eine Line, Dreieck oder Kreis gezeichnet wird, sollen diese geglättet werden so, dass es dem Benutzer möglichst einfach gemacht wird, die gewünschten Symbole zu zeichnen.

6.2.2.12 FZ_12: Farbauswahl S

Mit dieser Funktion kann der Benutzer die Farbe des Stiftes verändern, so dass er seine Zeichnung noch flexibler gestalten kann.

6.2.2.13 FZ_13 „Undo“ Funktion

Mit dieser Funktion kann der Benutzer den letzten Zeichnungsschritt rückgängig machen. (nur einmal)

6.2.2.14 FZ_14 „Shape Box“ Zeichnen

Der Benutzer hat mit dieser Funktion die Möglichkeit Rechtecke zu Zeichnen. Dabei soll dem Benutzer eine Vorschau angezeigt werden.

6.2.2.15 FZ_15 „Shape Circle“ Zeichnen

Mit dieser Funktion kann der Benutzer einen Kreis (Ellipse) zeichnen. Indem er den Stift bewegt wird ihm eine Vorschau angezeigt.

6.2.2.16 FZ_16 „Shape Polygon“ Zeichnen

Ermöglicht dem Benutzer ein Polygon zu Zeichnen, indem er die Punkte angibt und beim letzten Punkt einen Doppelklick macht (zwei aufeinanderfolgende Klicks nahe beieinander), um die Eingabe zu beenden. Dabei soll eine Korrektur durchgeführt, so dass Punkte die nahezu auf der Selben y / x Achse liegen auf der gleichen Achse zu liegen kommen.

6.2.2.17 FZ_17 „Shape Text“ Zeichnen

Der Benutzer hat mit dieser Funktion die Möglichkeit Text zu Zeichnen. Er soll den gewünschten Text über die SIP-Tastatur eingeben können. Dabei wird ihm angezeigt, welche Buchstaben er eingetippt hat. Drückt er den Stift nun auf den Screen, soll ihm eine Vorschau des Textes angezeigt werden. Er soll nun den Stift bewegen können und den Text dort platzieren wo er möchte. Wenn er den Stift abhebt soll der Text gezeichnet werden.

6.2.3 Erfassungsfunktionen

Damit die durch den Benutzer erstellten Zeichnungen integriert werden können, müssen beim erfassen der Daten einige Änderungen vorgenommen werden.

6.2.3.1 FE1: Wechseln zwischen erfassen und editieren M

Befindet sich der Benutzer im Modus um Daten zu erfassen, hat er die Möglichkeit, den Modus zu wechseln um die Aktuelle Zeichnung zu ändern (sofern er nicht eine Folie verwendet). Während dem der Benutzer die Zeichnung editiert, läuft die Zeit zur Erfassung weiter. Hat der Benutzer die Zeichnung nach seinen Bedürfnissen geändert, kehrt er wieder in den Erfassungsmodus zurück und die Zeichnung wird mit den entsprechenden Änderungen gespeichert.

6.2.3.2 FE2: Anzeigen des Datenfilenamens und Anzahl Punkte M

Während der Erfassung der Daten, wird dem Benutzer der Name des Datenfiles angezeigt und er sieht, wie viele Datenpunkte sich bereits in dem Datenfile befinden.

6.2.3.3 FE3: Speichern eines Datenfile M

Beendet der Benutzer die Datenerfassung, wird er nach einer Bestätigung gefragt, um den Modus zu verlassen. Bestätigt der Benutzer diese, wird das Datenfile mit allen Punkten abgespeichert.

6.2.3.4 FE4: Anzeigen der Option FE1 oder das Folie verwendet wird M

Damit die neue Version des FIT-Systems zur älteren Versionen Kompatibel bleibt, soll die Möglichkeit bestehen, anstatt einer digitalen Zeichnung, wie bisher eine von Hand angefertigte Folie zu verwenden. Damit der Benutzer weiss, in welchem Modus er sich befindet, wird ihm angezeigt, ob eine Folie verwendet wird. Falls dies der Fall ist, hat der Benutzer logischerweise keine Möglichkeit, die Zeichnung zu editieren. Verwendet der Benutzer jedoch eine digitale Zeichnung wird ihm die Möglichkeit geboten, vom Modus Datenerfassen in den Modus Zeichnung editieren zu wechseln. Diese Möglichkeit impliziert die Tatsache, dass keine Folie verwendet wird.

6.2.3.5 FE5: Zeichnungsfunktion Im Modus Erfassen integrieren M

Wenn der Benutzer in den Modus Daten erfassen wechselt wird er neu aufgefordert, wie er die Datenerfassung haben möchte. Er kann wählen, ob er eine Folie verwenden will oder eine bestehende Zeichnung laden, um die Daten anschliessend zu erfassen. Will er eine bereits vorhandene Zeichnung verwenden wird ihm eine Liste mit allen zur Verfügung stehenden Zeichnungen eingeblendet. Will der Benutzer eine Folie verwenden gelangt er direkt in den Erfassungsmodus.

6.2.3.6 FE6: Letzter Punkt blinkend W

Im Modus Daten erfassen soll der letzte Datenpunkt jeweils markiert sein, damit der Benutzer sieht, wo genau er den Punkt gesetzt hat.

6.2.4 Löschfunktionen

Dem Benutzer wird die Möglichkeit angeboten, bestehende Dateien zu löschen. Er kann auswählen, ob er ein Datenfile oder ein Zeichnungsfile löschen will.

6.2.4.1 FL1: Löschen eines Datenfiles M

Will der Benutzer ein Datenfile löschen, wird ihm eine Liste mit allen vorhandenen Datenfiles angezeigt. Ist ein Datenfile angewählt, wird angezeigt, wann die Datei kreiert wurde und die Anzahl Datenpunkte die es enthält.

6.2.4.2 FL2: Löschen eines Zeichnungsfiles M

Will der Benutzer ein Zeichnungsfile löschen, wird ihm eine Liste mit allen vorhandenen Zeichnungsfiles angezeigt. Ist ein Zeichnungsfile angewählt, wird angezeigt, wann die Datei kreiert wurde.

6.2.4.3 FL3: Bestätigungsfenster anzeigen M

Bevor der Benutzer ein entsprechendes File löschen kann, wird eine Bestätigungsmeldung angezeigt, welche durch den Benutzer quittiert werden muss.

6.2.5 Signierfunktionen

Um die Datenpunkte unterscheiden zu können, müssen diese durch den Benutzer signiert und mit einem Namen versehen werden. Momentan geschieht diese Signierung auf dem PC. Der Wunsch ist es nun, dass man diese Signierung auch auf dem Palm durchführen könnte.

6.2.5.1 FS1: Auswahl des Datenfile W

Will der Benutzer ein Datenfile signieren wird ihm die Liste mit den Datenfiles angezeigt, von welcher er das zu signierende File auswählen muss.

6.2.5.2 FS2: Auswahl des Zeichnungsfile oder Folie W

Will der Benutzer ein Datenfile signieren, muss er die zum Datenfile dazugehörige Zeichnung auswählen. Dazu wird ihm eine Liste mit allen vorhandenen Zeichnungen angezeigt. Hat der Benutzer eine Folie verwendet, hat er die Möglichkeit in der Liste den Eintrag "None" auszuwählen. Es werden dann nur die Punkte (ohne die Zeichnung) angezeigt.

6.2.5.3 FS3: Anzeigen der Punkte aus Datenfile W

Hat der Benutzer ein Datenfile und eine Zeichnung ausgewählt, werden alle Punkte angezeigt. Hat der Benutzer zusätzlich eine Zeichnung ausgewählt, erscheint diese (nach Möglichkeit in einem helleren Grauton) ebenfalls auf dem Display. Wurde eine Folie verwendet werden nur die Punkte angezeigt.

6.2.5.4 FS4: Polygon um Symbol zeichnen W

Um die Datenpunkte für ein Symbol zu signieren müssen diese durch ein Polygon umrandet werden.

6.2.5.5 FS_4.1: Polygon Benennung ändern

Es muss möglich sein den gegebenen Namen des Polygons zu ändern. (z.B. Auswahl + Doppelklick)

6.2.5.6 FS_4.2: Polygon Löschen

Es muss möglich sein ein Erfasstes Polygon zu Löschen

6.2.5.7 FS_4.3: Polygon verschieben

Ein Gezeichnetes Polygon muss verschoben werden können

6.2.5.8 FS5: Aufforderung um Namen des Icons einzugeben W

Hat der Benutzer ein Polygon um die Punkte gezogen, wird er aufgefordert, einen Namen zur Kennung dieser Datenpunkte einzugeben.

6.2.5.9 FS6: Speichern des Datenfiles mit Iconnamen W

Verlässt der Benutzer den Modus signieren, werden die entsprechenden Änderungen im Datenfile gespeichert.

6.2.6 FK_1: Kopierschutz

Es soll ein Kopierschutz erstellt werden, der verhindert, dass ein Lizenznehmer die Software auf mehr als einem Gerät laufen und einsetzen kann.

6.3 Nichtfunktionale Anforderungen

6.3.1 Benutzbarkeit

Die neue FIT Manager Applikation soll vom Aufbau her gleich gestaltet sein, wie die Vorgängersoftware des FIT Systems. Ein Benutzer der alten FIT Applikation soll sich in der neuen Software sofort wieder zurechtfinden. Ausserdem kann die Applikation Farblich verschönert werden.

6.3.2 Zuverlässigkeit

Die Applikation muss zu jederzeit auf Benutzereingaben sofort reagieren. Wartezeiten über 1 Sekunden sind inakzeptabel.

6.3.3 Mengenanforderungen

- Die Anzahl der zu erstellenden Layouts (Zeichnungen) wird durch den Hauptspeicher des Gerätes beschränkt
- Die Anzahl der aufzunehmender Datenfiles wird durch den Hauptspeicher des Gerätes beschränkt
- Die Anzahl der abgespeicherten Datenfiles bzw. Layouts wird durch den Speicher des Gerätes beschränkt.

6.3.4 Anforderungen an die Schnittstellen

6.3.4.1 Benutzer – Schnittstelle

Der Benutzer muss das FIT System über den Touchscreen und dem „Graphical User Interface“ (GUI) bedienen können. Während die Applikation läuft, müssen die Funktionstasten des IPAQ's ausgeschaltet sein, mit Ausnahme des Ein- / Ausschalters.

6.3.4.2 System- / Hardware – Schnittstelle

- Die Applikation muss auf dem IPAQ unter Windows Mobile 6.0 lauffähig sein
- Nach Möglichkeit sollte beim Zeichnen und Erfassen der ganze Bildschirm verwendet werden können. (Auf jeden Fall ein möglichst grosser Teil)

6.3.4.3 Übertragungs-Schnittstelle

Bei Windows XP wird ActiveSync 4.5 und bei Vista und Windows 7 wird Windows Mobile Device Center benutzt. Der HP IPAQ 214 verfügt über folgende Übertragungsschnittstellen:

- USB 2.0
- Bluetooth v2.0
- Wi-Fi (IEEE 802.11b/g) mit WPA-2 Unterstützung

6.3.4.4 Software – Schnittstelle

Die Software läuft auf dem Betriebssystem Windows Mobile 6. Falls das .net Compact Framework heruntergeladen wird, ist die Software sehr wahrscheinlich auch auf älteren Betriebssystemversionen lauffähig

6.3.4.5 Datenhaltungs – Schnittstelle

Zeichnungen sollen im JPEG Format abgelegt werden, damit die Zeichnungen auch auf dem Desktop angeschaut werden können.

6.3.5 Erweiterbarkeit

Das Software-Design soll so konzipiert sein, dass es sich problemlos um zusätzliche Funktionen erweitern lässt.

6.3.6 Wartbarkeit

Die Applikation muss so entwickelt werden, dass keine Wartungen bei der Software mehr nötig sind.

6.3.7 Anforderungen zur Entwicklungsumgebung

Es wurde Entschieden mit folgenden Entwicklungswerkzeugen zu arbeiten:

- Windows XP/ Vista / Windows 7
- Visual Studio 2008
- Windows Mobile 6 Professional SDK
- .net CF 2.0 SP2 (falls nötig oder vorteilhaft .net CF 3.5)

6.3.8 Sprachen

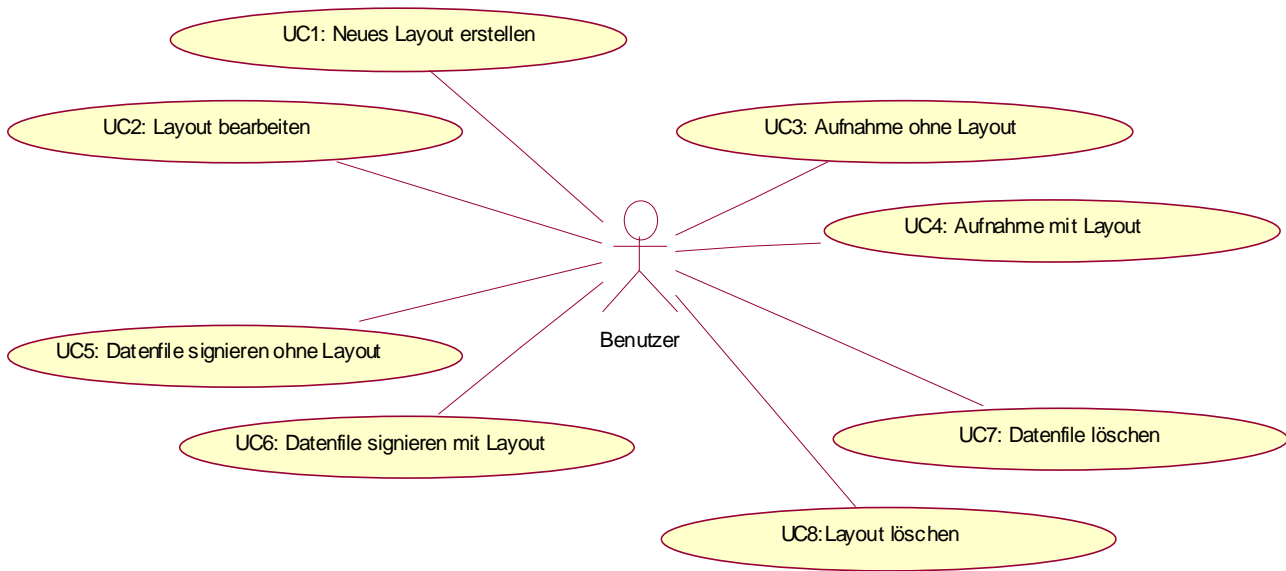
Das Fit System soll in der englischen Sprache vorliegen

6.3.9 Anforderungen der System - Tests

Die Integrations- und Systemtests sollen auf dem IPAQ durchgeführt werden

6.4 Use Cases¹⁵

6.4.1 UC – Diagramm



¹⁵ Use Cases sind gleich wie in der Vorgängerapplikation und wurden von dort übernommen. Siehe DrawFIT.pdf Kapitel 6.4

6.4.2 UC1: Neues Layout erstellen

Preconditions:	Benutzer befindet sich im Haupt-Form des FIT-System.
Postconditions:	Das durch den Benutzer neu erstellte Layout hat einen Namen und ist im System abgespeichert. Der Benutzer befindet sich wieder im Haupt-Form.
Primary Actor:	PDA Benutzer
Any Time:	Beendigung des Vorgangs
Beschreibung:	Der Benutzer kann ein neues Layout erstellen, welches er mit einem Namen versieht. Er kann in diesem Layout mit einem Stift beliebige Figuren zeichnen und eine "Radiergummifunktion" sowie eine "Lösche alles" Funktion verwenden. Der Stift kann verschiedene Dicken annehmen. Auch kann die Zeichen Farbe verändert werden. Ausserdem werden die Zeichnungen geätzt, falls die Glättungsoption eingeschaltet ist.

Aktor	System
	1. Stellt die Haupt-Form dar(Anfangszustand)
2. Will ein neues Layout erstellen	
	3. Fordert den Benutzer auf, einen Namen für das neu zu erstellende Layout einzugeben.
4. Gibt einen Namen für das neu zu erstellende Layout ein und bestätigt die Eingabe.	
	5. - Zeigt den Zustand der Batterie an. - Zeigt den Namen des Layouts an. - System bietet Zeichnungsfunktionen an.
6. Zeichnet beliebig viele Figuren.	
7. Will das neu erstellte Layout speichern	
	8. System fragt nach einer Bestätigung um den Zeichnungsmodus zu verlassen
9. Benutzer bestätigt die Absicht den Zeichnungsmodus zu beenden.	
	10. Speichert das neu erstellte Layout und kehrt zum Anfangszustand zurück

6.4.3 UC2: Layout bearbeiten

Preconditions:	Es ist bereits ein Layout vorhanden.
Postconditions:	Die durch den Benutzer vorgenommenen Änderungen sind im Layout gespeichert.
Primary Actor:	PDA Benutzer
Any Time:	Beendigung des Vorgangs
Beschreibung:	Der Benutzer kann ein bereits vorhandenes Layout laden und nach belieben mit den Zeichnungsfunktionen Stift, Radiergummi bzw. "lösche alles" das Layout abändern. Ist das Layout nach den Bedürfnissen des Benutzers geändert, wird es in der neu vorliegenden Form gespeichert, wenn der Benutzer den Zeichnungsmodus verlässt.

Aktor	System
	1. Stellt die Haupt-Form dar (Anfangszustand)
2. Will ein vorhandenes Layout abändern	
	3. Fordert den Benutzer auf, das zu ändernde Layout auszuwählen.
4. Wählt das zu ändernde Layout aus.	
	5. - Zeigt den Zustand der Batterie an. - Zeigt den Namen des Layouts an. - System bietet Zeichnungsfunktionen an.
6. Ändert das Layout beliebig ab.	
7. Will das geänderte Layout speichern	
	8. System fragt nach einer Bestätigung um den Zeichnungsmodus zu verlassen
9. Benutzer bestätigt die Absicht den Zeichnungsmodus zu beenden.	
	10. Speichert das geänderte Layout und kehrt zum Anfangszustand zurück

6.4.4 UC3: Aufnahme ohne Layout

Preconditions: Benutzer befindet sich im Haupt-Form des FIT-System.
Postconditions: Die durch den Benutzer eingegebenen Datenpunkte sind mit dem Zeitstempel und den Koordinaten in einem Datenfile im System gespeichert.
Primary Actor: PDA Benutzer
Any Time: Beendigung bzw. Unterbrechung des Vorgangs Daten erfassen
Beschreibung: Der Benutzer besitzt ein Template um die Datenpunkte zu erfassen.

Aktor	System
	1. Stellt die Haupt-Form dar (Anfangszustand)
2. Will Datenpunkte aufnehmen	
	3. Fordert den Benutzer auf, einen Namen für das Datenfile anzugeben.
4. Gibt einen Namen für das Datenfile ein.	
	5. - Zeigt den Zustand der Batterie an. - Zeigt den Namen des Datenfiles an. - Zeigt die Uhrzeit an. - System bietet Erfassungsfunktion an.
6. Erfasst beliebig viele Datenpunkte.	
7. Will die erfassten Datenpunkte speichern	
	8. System fragt nach einer Bestätigung um den Erfassungsmodus zu verlassen
9. Benutzer bestätigt die Absicht den Erfassungsmodus zu beenden.	
	10. Speichert das Datenpunktefile und kehrt zum Anfangszustand zurück

6.4.5 UC4: Aufnahme mit Layout

- Preconditions:** Benutzer befindet sich im Haupt-Form des FIT-System. Es liegt ein durch den Benutzer erstelltes Layout vor.
- Postconditions:** Die durch den Benutzer eingegebenen Datenpunkte sind mit dem Zeitstempel und den Koordinaten in einem Datenfile im System gespeichert.
- Primary Actor:** PDA Benutzer
- Any Time:** Beendigung bzw. Unterbrechung des Vorgangs Daten erfassen
- Beschreibung:** Der Benutzer hat ein Layout erstellt um die Datenpunkte zu erfassen. Dazu wird das Layout geladen und der Benutzer kann anschliessend die Datenpunkte erfassen.

Aktor	System
	1. Stellt die Haupt-Form dar (Anfangszustand)
2. Will Datenpunkte aufnehmen	
	3. Fordert den Benutzer auf, einen Namen für das Datenfile anzugeben.
4. Gibt einen Namen für das Datenfile ein.	
	5. Fordert den Benutzer auf, das gewünschte Layout zu laden.
6. Wählt ein Layout aus.	
	7. - Zeigt den Zustand der Batterie an. - Zeigt den Namen des Datenfiles an. - Stellt das Layout als Eingabefeld dar - Zeigt die Uhrzeit an. - System bietet Erfassungsfunktion an.
8. Erfasst beliebig viele Datenpunkte.	
9. Will die erfassten Datenpunkte speichern	
	10. System fragt nach einer Bestätigung um den Erfassungsmodus zu verlassen
11. Benutzer bestätigt die Absicht den Erfassungsmodus zu beenden.	
	12. Speichert das Datenpunktefile und kehrt zum Anfangszustand zurück

6.4.6 UC5: Datenfile signieren mit layout

Preconditions: Der Benutzer hat Datenpunkte mit einem von ihm erstellten Layout erfasst und das entsprechende Datenfile ist im System gespeichert.
Postconditions: Die Datenpunkte des Datenfiles sind mit einem Namen signiert.
Primary Actor: PDA Benutzer
Any Time: Beenden des Vorgangs
Beschreibung: Der Benutzer will die Datenpunkte eines Datenfiles signieren, bedeutet, er will den Datenpunkten Namen zuordnen. Als Zeichnung wird ein Layout verwendet

Aktor	System
	1. Stellt die Haupt-Form dar (Anfangszustand)
2. Will Datenpunkte signieren	
	3. Fordert den Benutzer auf, das gewünschte Datenfile anzugeben, welches signiert werden soll
4. Gibt zu signierendes Datenfile an	
	5. Fordert den Benutzer auf, das dazugehörige Layout anzugeben.
6. Gibt das zum Datenfile zugehörige Layout an	
	7. - Zeigt den Zustand der Batterie an. - Zeigt den Namen des Datenfiles an. - Stellt das Layout als Eingabefeld dar - Zeigt die Uhrzeit an. - Zeigt alle vorhandenen Punkte des Datenfiles an - System bietet Signierfunktion an
8. Signiert alle seine Datenpunkte.	
9. Will alle signierten Datenpunkte in dem Datenfile abspeichern	
	10. System fragt nach einer Bestätigung um den Signiermodus zu verlassen
11. Benutzer bestätigt die Absicht den Signiermodus zu beenden.	
	12. Speichert das signierte Datenpunktefile und kehrt zum Anfangszustand zurück

6.4.7 UC6: Datenfile Signieren ohne Layout

- Preconditions:** Der Benutzer hat Datenpunkte erfasst und dazu eine Folie verwendet. Das entsprechende Datenfile ist im System gespeichert.
Postconditions: Die Datenpunkte des Datenfiles sind mit einem Namen signiert
Primary Actor: PDA Benutzer
Any Time: Beenden des Vorgangs
Beschreibung: Der Benutzer will die Datenpunkte eines Datenfiles signieren, bedeutet, er will den Datenpunkten Namen zuordnen. Als Zeichnung wird eine physische Folie verwendet

Aktor	System
	1. Stellt die Haupt-Form dar (Anfangszustand)
2. Will Datenpunkte signieren	
	3. Fordert den Benutzer auf, das gewünschte Datenfile anzugeben, welches signiert werden soll
4. Gibt zu signierendes Datenfile an	
6. Legt die selbst gezeichnete physische Folie auf	
	7. - Zeigt den Zustand der Batterie an. - Zeigt den Namen des Datenfiles an. - Stellt das Layout als Eingabefeld dar - Zeigt die Uhrzeit an. - Zeigt alle vorhandenen Punkte des Datenfiles an - System bietet Signierfunktion an
8. Signiert alle seine Datenpunkte.	
9. Will alle signierten Datenpunkte in dem Datenfile abspeichern	
	10. System fragt nach einer Bestätigung um den Signiermodus zu verlassen
11. Benutzer bestätigt die Absicht den Signiermodus zu beenden.	
	12. Speichert das signierte Datenpunktefile und kehrt zum Anfangszustand zurück

6.4.8 UC7: Datenfile löschen

- Preconditions:** Benutzer befindet sich im Haupt-Form des FIT-System. Es liegen durch den Benutzer erstellte Datenfiles vor.
- Postconditions:** Das durch den Benutzer ausgewählte Datenfile ist gelöscht und im System nicht mehr vorhanden.
- Primary Actor:** PDA Benutzer
- Beschreibung:** Der Benutzer kann ein erstelltes Datenfile wieder löschen.

Aktor	System
	1. Stellt die Haupt-Form dar(Anfangszustand)
2. Will ein Datenfile löschen	
	3. Fordert den Benutzer auf, das zu löschende Datenfile auszuwählen.
4. Wählt das zu löschende Datenfile. *	
	5. Löscht das ausgewählte Datenfile.
6. Will den Modus Datenfiles löschen beenden.	
	7. Kehrt zum Haupt-Form zurück.

* Schritte 4 und 5 können beliebig oft wiederholt werden.

6.4.9 UC8: Layout löschen

- Preconditions:** Benutzer befindet sich im Haupt-Form des FIT-System. Es liegen durch den Benutzer erstellte Layouts vor.
- Postconditions:** Das durch den Benutzer ausgewählte Layout ist gelöscht und im System nicht mehr vorhanden.
- Primary Actor:** PDA Benutzer
- Beschreibung:** Der Benutzer kann ein erstelltes Layout wieder löschen.

Aktor	System
	1. Stellt die Haupt-Form dar(Anfangszustand)
2. Will ein Layout löschen	
	3. Fordert den Benutzer auf, das zu löschende Layout auszuwählen.
4. Wählt das zu löschende Layout. *	
	5. Löscht das ausgewählte Layout.
6. Will den Modus Layout löschen beenden.	
	7. Kehrt zum Haupt-Form zurück.

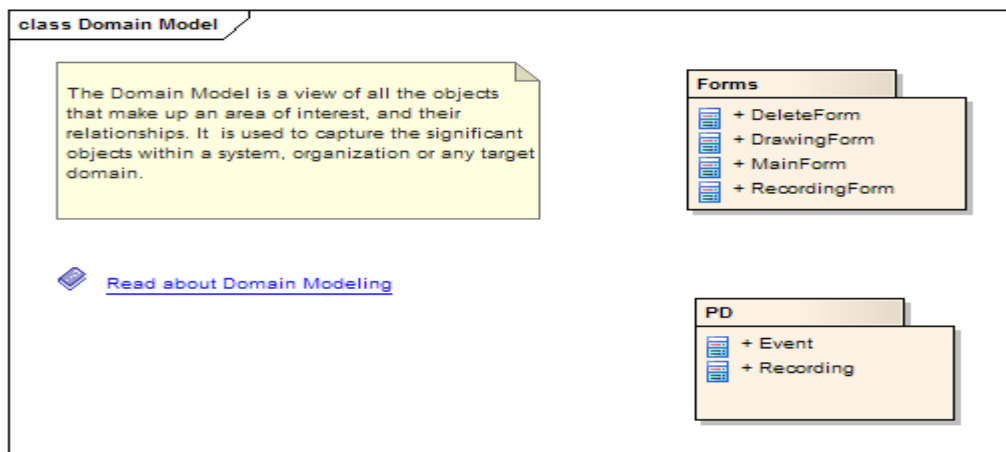
* Schritte 4 und 5 können beliebig oft wiederholt werden.

7. Analyse

7.1 Einleitung

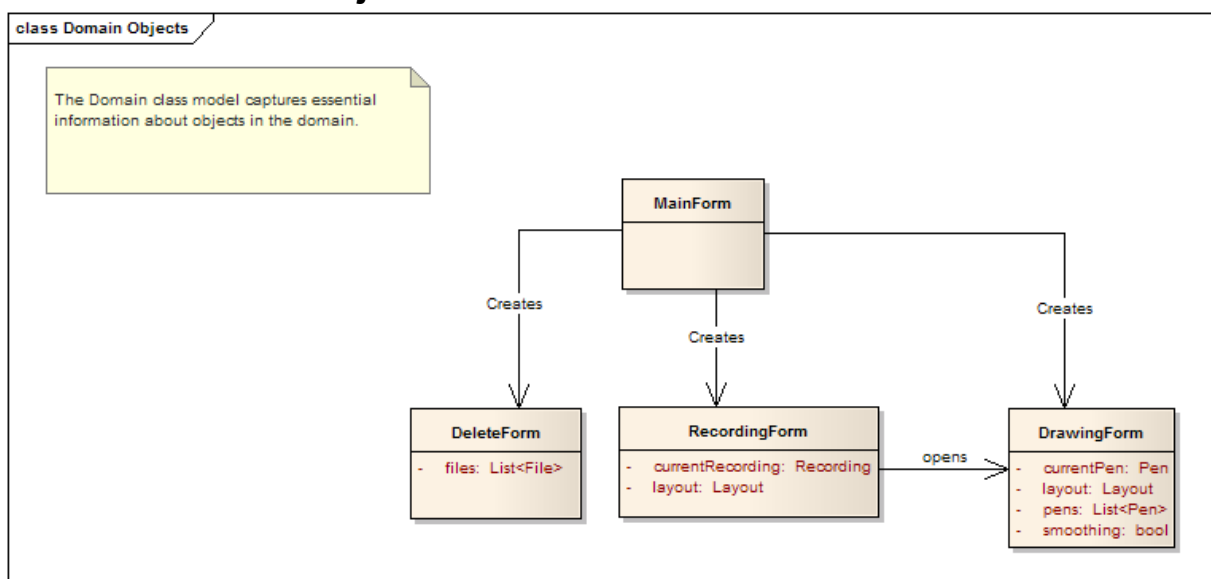
In diesem Dokument werden die Anforderungen an das FIT-Systems analysiert und in einem DomainModel dargestellt. Die Zustandsdiagramme und das Externe Design wurden weitgehend von der Vorgängerapplikation übernommen.

7.2 Domain Model



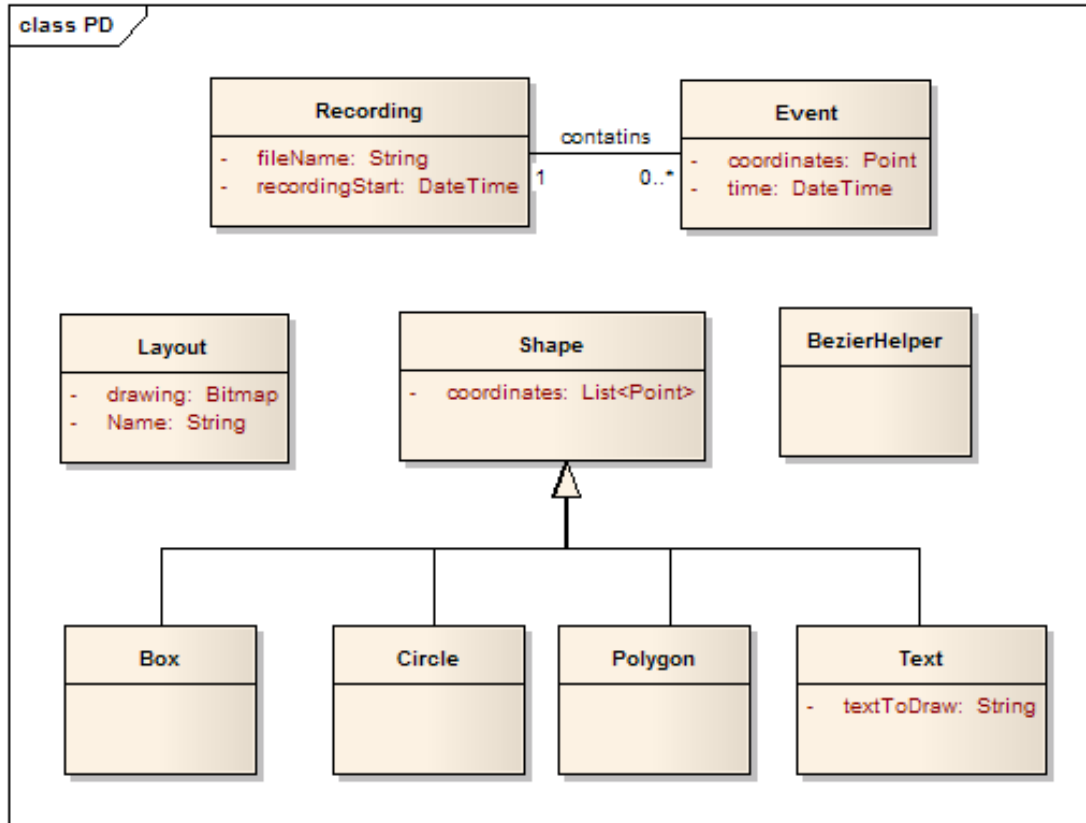
Die Applikation wird in zwei Bereiche aufgeteilt nämlich in die Form in denen Die Benutzeroberfläche definiert ist und die PD (Problem Domain) in denen Funktionen abgehandelt werden die aus den Forms ausgelagert werden konnten.

7.2.1 Domain Object Forms



Für alle drei Hauptfunktionalitäten wurde je eine Form angedacht.

7.2.2 Domain Objects PD



Konzept Recording & Event

Diese beiden Elemente abstrahieren die Aufzeichnung von Ereignissen der FIT-System Applikation.

Konzept Layout

Stellt die vom Benutzer mit dem FIT-System gezeichnete Oberfläche dar die er für seine Beobachtung benötigt um die gewünschten Arbeitsschritte zu Optimieren.

Konzept Shape

Um den Zeichnungsmechanismus zu vereinfachen und teilweise aus der „DrawingForm“ auszulagern wurde dieser Mechanismus entworfen, der mittels Polymorphismus die Figuren zeichnet.

7.3 Zustandsdiagramme und Externes Design

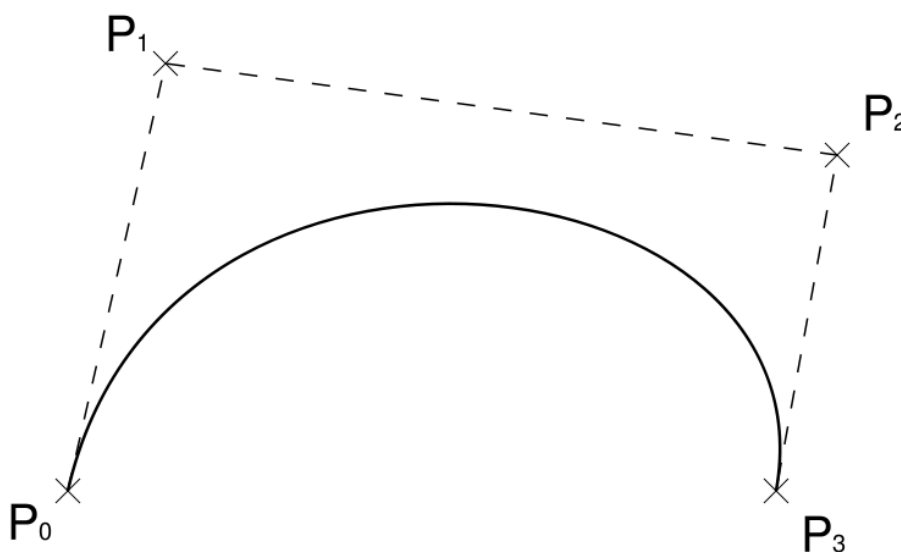
Siehe Dokumentation Vorgängerapplikation¹⁶

¹⁶ DrawFIT[1].pff Kapitel 7 Analyse & Externes Design

7.4 Glättung (Bezier)

Die Bezierkurve ist in der numerischen Mathematik eine parametrische modellierte Kurve, die ein wichtiges Werkzeug für Vektorgrafiken darstellt. Sie wurde Anfang der 1960er Jahre entwickelt.¹⁷

Für die Entwicklung der Glättungsfunktion wurden Kubische Bezierkurven verwendet. Bei Kubischen Bezierkurven bestimmen Vier Punkte (P0, P1, P2, P3) wie die Kurve gezeichnet wird. Die Kurve beginnt bei P0 und geht in Richtung P1 und dann aus Richtung P2 zu P3. Normalerweise geht die Kurve nicht durch P1 und P2, da diese Punkte nur der Richtung dienen. Für Weitere Informationen Siehe Wikipedia¹⁸.



7.4.1 Implementation

Die Implementation wurde basierend auf eine Quelle¹⁹ im Internet erstellt. Um Kurven miteinander zu verbinden, wird einfach der Punkt P3 der alten Kurve, zum P0 der neuen Kurve. Um verbundene Bezierkurven zu zeichnen, braucht man eine gewisse Anzahl von Punkten, welche folgende Bedingung erfüllt: $((p-4) \% 3 == 0)$ falls die Anzahl von Punkten nicht mit der benötigten Menge übereinstimmt, wird der letzte Punkt einfach so oft wie nötig kopiert. Liegt der erste- und der letzte Punkte nicht weit voneinander entfernt (10Pixel), wird der letzte Punkt durch den ersten ersetzt, um die Zeichnung zu verbinden. Zuerst wurde für jedes „Mause Ereignis“ ein Punkt zum Zeichnen der Kurve erfasst. Dies führte jedoch kaum zu einer sichtbaren Glättungswirkung. Das ganze wurde um den Faktor 4 gestreckt, was bedeutet, dass jetzt nur bei jedem vierten „Mause Ereignis“ ein Punkt für das Zeichnen der Kurve erfasst wird. Dies führt auf dem iPAC zu einer sehr guten Glättung, auf dem Emulator ist sie so jedoch etwas stark.

¹⁷ <http://de.wikipedia.org/wiki/B%C3%A9zierkurve>

¹⁸ <http://de.wikipedia.org/wiki/B%C3%A9zierkurve>

¹⁹ http://www.therobotgeek.net/articles/CF_Bezier_Curves.aspx (Falls Link nicht funktioniert einfach Hauptseite öffnen und Artikel suchen)

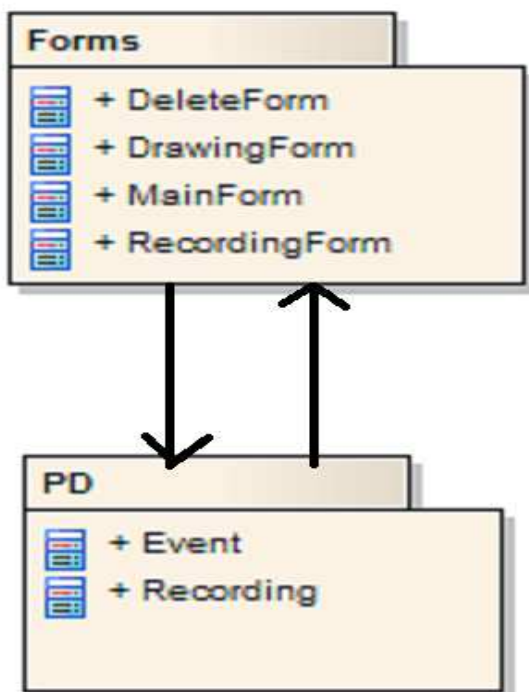
8. Internes Design

8.1 Einleitung

Das interne Design beschreibt den Aufbau und die Architektur der FIT-System Software. Das Programm wurde in der Programmiersprache C# entwickelt und ist somit Objektorientiert. Dies war bei der Vorgängerapplikation nicht der Fall.

8.2 Software - Architektur

Der Source Code der Applikation wurde in zwei verschiedene Pakete aufgeteilt (Siehe Bild).



8.2.1 Forms

In diesem Packet befinden sich alle Forms (Formulare) mit denen der Benutzer der Software interagiert. Es ist auch ein grosser Teil der Applikationslogik drin vorhanden. Jede dieser Form Klassen ist in Zwei Dateien aufgeteilt. Die eine Datei wird um den Namenszusatz .Designer ergänz und beinhaltet alle durch den „GUI Bilder“ generierten Daten unter anderem alle GUI Steuerelemente (Button, TextBox, MenuItem). Die andere Datei enthält die Logik der Form.

8.2.2 PD - Package

Im Package PD befinden sich Logik der Applikation, die aus den Forms ausgelagert wurde. Logik die gut ausgelagert werden konnte und zusammengehörte wurde in einer Klasse zusammengefasst. Dies beinhaltet das „Layout“ auf dem gezeichnet wird, das „Recording“ mit den dazugehörigen „Events“, die die aufgezeichneten Ereignisse repräsentieren und die „BezierHelper“ Klasse die das Zeichnen der Bezierkurven übernimmt. Auch dazu gehören die Klassen „Shape“, „Box“, „Circle“, „Polygon“ und „Text“, die ihre eigene Zeichnungslogik enthalten.

8.3 Dateiaufbau

Jede Klasse befindet sich in ihrer eigenen Datei, mit Ausnahme der Form Klassen, die in je zwei Dateien aufgeteilt sind und in einen vom GUI Generator generierten Teil und in den Logik Teil aufgespalten sind.

8.3.1 Klassen - Diagramm



8.3.2 Form Klassen

8.3.2.1 MainForm.cs



Diese Klasse stellt das Anfangsfenster der Applikation dar. Wo der Benutzer auswählen kann welche Aktion er mit dem FIT-System ausführen möchte. Er Aktionen die der Benutzer durchführen kann sind:

- Record Um Daten zu erfassen
- Draw Um ein Layout für die Datenerfassung zu zeichnen.
- Delete Hiermit kann er entweder ein Layout oder en Datenfile Löschen

Von diesem Fenster aus kann er auch die Applikation verlassen indem er auf das Kreuz oben links klickt.

8.3.2.2 EnterRecordNameForm.cs



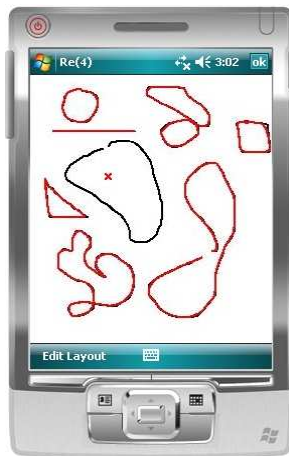
Diese Form wird angezeigt, nachdem der „Record“ Button in der „MainForm“ gedrückt wurde. Das SIP Keyboard wird eingeblendet, damit der Benutzer einen Namen eingeben kann. Die Klasse ist außerdem dafür verantwortlich, die Einzigartigkeit des Namens sicherzustellen. Nachdem „Ok“ gedrückt wurde, wird eine „.fitdb“ Textdatei, mit dem angegebenen Namen erstellt und die „RecordLayoutOrTemplate“ Form wird angezeigt. Falls der Name nicht einzigartig oder ungültig ist, wird eine Fehlermeldung angezeigt.

8.3.2.3 RecordTempalteOrTemplate.cs



Beinhaltet die Funktionalität, um das links dargestellte Formular anzuzeigen und die Funktionalität hinter den einzelnen Buttons. Wählt der Benutzer „Load layout“, dann wird im der „LoadLayout“ Dialog angezeigt in dem er das gewünschte Layout auswählen kann. Hat er dies getan, wird die „RecordForm“ mit dem ausgewählten Layout als Parameter geöffnet. Bei „Use Template“ geschieht dasselbe, jedoch mit leerem Parameter. Der „Cancel Button“ schliesst die Form und man kehrt zur „MainForm“ zurück.

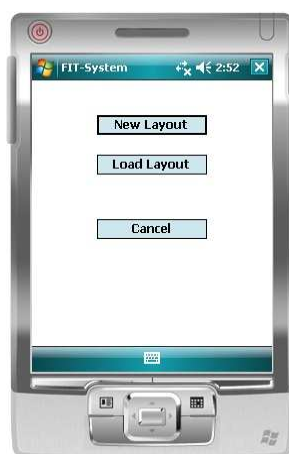
8.3.2.4 RecordForm



Diese Form beinhaltet die wichtigste Funktionalität der Applikation, nämlich das aufzeichnen von Ereignissen. Im oberen Windowsbalken wird der Name des Datenfiles, gefolgt von der Anzahl bereits erfasster Ereignisse (in Klammern) angezeigt. Falls der Benutzer ein Layout verwendet wird dieses angezeigt und er hat die Möglichkeit über die „Edit Layout“ Menüoption in die Drawing Form zu wechseln und das Layout zu verändern. Jedes Mal wenn der Benutzer ein Symbol des Layouts (oder der aufgelegten Schablone) mit dem Stift antippt wird dies registriert und der „Event“ (Ereignis) mit den dazugehörigen Koordinaten und Zeitstempel erfasst. Der Ort des jeweils zuletzt erfassten „Events“, wird durch ein rotes Kreuz angezeigt. Wird oben rechts auf „Ok“ gedrückt,

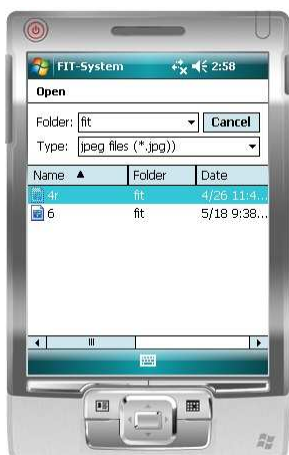
wird der Benutzer gefragt, ob er die Aufzeichnung wirklich beenden möchte. Ist dies der Fall werden alle Events in der „.fitdb“ Datei abgespeichert, die Datenaufzeichnung beendet und die Applikation kehrt zur „MainForm“ zurück.

8.3.2.5 DrawingSelectionForm.cs



Diese Form wird angezeigt, nachdem in der „MainForm“ die Option „Draw“ gewählt wurde. Durch den „Cancel Button“ wird diese Form geschlossen und der Benutzer kehrt zur „MainForm“ zurück. Durch „New Layout“ wird die „EnterLayoutNameForm“ angezeigt. Der „Load Layout“ Button öffnet den „Load Layout“ Dialog.

8.3.2.6 LoadLayoutDialog



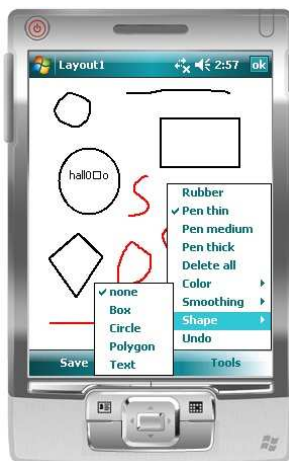
Dieser Dialog zeigt alle Layouts (.jpg Files) der FIT-System Applikation an. Der Benutzer kann jedoch auch beliebige andere .jpg Files laden (sie sollten jedoch die korrekte Grösse haben). Wird ein File ausgewählt, wird dieses geladen und je nach Dialog Ort die „DrawingForm“ oder „RecordForm“ mit dem ausgewählten Layout angezeigt.

8.3.2.7 EnterLayoutNameForm.cs



Falls „New Layout“ ausgewählt wurde, wird die diese Form angezeigt. Die SIP Tastatur wird eingeblendet, so dass der Benutzer den Namen des Layouts eingeben kann. Durch „Cancel“ kehrt der Benutzer zur „DrawingSelectionForm“ zurück. Bei „OK“ wird der angegebene Name auf Gültigkeit und Einzigartigkeit überprüft. Falls dies ein Problem ist wird dies dem Benutzer mittels „Alert“ mitgeteilt, sonst wird die Drawing Form geöffnet.

8.3.2.8 Drawing Form.cs



Diese Form enthält zwar nicht die wichtigste Funktionalität der Applikation, jedoch stecken am meisten Funktionalitäten und Aufwand und Neuerungen gegenüber der Vorgängerapplikation dahinter. Diese Form enthält fast die gesamte Zeichnungslogik. Nur das Zeichnen der Bezierkurven und der Shapes wurde in eigene Klassen ausgelagert. Der Layout Name wird im Windowsbalken angezeigt. Um den Batteriestatus anzuzeigen muss man auf die Uhrzeit drücken. Alle Zeichnungsereignisse werden in dieser Form behandelt. Der Benutzer hat die Möglichkeit zwischen drei Stiftgrößen zu wählen. Dies beeinflusst nicht nur die Zeichnungen, sondern auch die Grösse des „Shape“ „Text“. Das Zeichnen wird über die „Mouse down“, „Mouse move“, und „Mouse up“ Events abgehandelt. Je nachdem ob die „Smoothing“ Option „Bezier“, ein „Shape“ oder keines ausgewählt ist verhält sich die Applikation anders beim Auftreten dieser „Events“. Die „Mouse Events“ werden durch berühren des Touchscreens aufgerufen. Die Logik der „Rubber“- , der „Undo“- und der „Delete All“ Funktion befinden sich auch in dieser Form.

8.3.2.9 DeleteSelectionForm.cs



Diese Form wird angezeigt, nachdem der Benutzer die „Delete“ Option in der „Main Form“ betätigt hat. Der „Cancel Button“ schliesst diese Form und die Applikation kehrt zur „MainForm“ zurück. „Delete Datafile“ öffnet die „DeleteRecordForm“ und „Delete Layout“ die „DeleteLayoutForm“.

8.3.2.10 DeleteRecordForm.cs



Diese Form Zeigt alle gespeicherten „Data Files“ (.fitdb Dateien) in einer List an. Wird eine ausgewählt, so zeigt die Form das Erstellungsdatum sowie die Anzahl der Einträge an. Mit der Menü Option „Cancel“ kehrt man zur „DeleteSelectionForm“ zurück. Die „Delete“ Menü Option ist nur aktiviert, falls eine Datei selektiert ist. Wird sie betätigt erscheint ein „Alert“ mit der Nachfrage ob die Datei wirklich gelöscht werden soll. Wird dies bestätigt wird die Datei vom Dateisystem des PDA entfernt.

8.3.2.11 DeleteLayoutForm.cs



Diese Form Zeigt alle gespeicherten „Layout Files“ (.jpg Dateien) in einer List an. Wird eine ausgewählt, so zeigt die Form das Erstellungsdatum an. Mit der Menü Option „Cancel“ kehrt man zur „DeleteSelectionForm“ zurück. Die „Delete“ Menü Option ist nur aktiviert falls eine Datei selektiert ist. Wird sie betätigt erscheint ein „Alert“ mit der Nachfrage ob die Datei wirklich gelöscht werden soll. Wird dies bestätigt wird die Datei vom Dateisystem des PDA entfernt.

8.3.3 PD Dateien

8.3.3.1 Bezier Helper

Diese Klasse stellt die Funktionalität zur Verfügung, um Bezier Kurven²⁰ zu zeichnen. Die Klasse enthält zwei statische Methoden. Die Methode „DrawBezier“, der vier Punkte bestehend aus (x, y Koordinate) übergeben werden. Anhand des „Bezier Kurve“⁸ Algorithmus wird diese Kurve auf das übergebene „Graphics“ Objekt mit dem übergebenen „Pen“ (Stift) gezeichnet. Die zweite Methode „DrawBeziers“ unterscheidet sich nur darin, dass ein „Point“ „Array“ übergeben wird. Dieses „Array“ muss aus einer Anzahl Punkte bestehen die wenn 4 abgezogen wird durch 3 Teilbar sein muss. Diese Methode zeichnet die Kurven auch mit jeweils 4 Punkten. Wobei der letzte Punkt immer der erste Punkt der nächsten Zeichnung wird. Dieses Vorgehen ist eine vereinfachte wiese der Bezierkurve.

8.3.3.2 Shape

Dies ist eine abstrakte Klasse die zum zeichnen der „Shapes“ (Figuren) wegwendet wird. Durch diese Klasse wird auch der Polymorphismus genutzt, so dass sich jede Figur selbst zeichnet. Sie enthält auch die Logik die für alle Figuren identisch ist.

Properties / Attribute

List<Point> Coordinates	Koordinaten die zum Zeichnen der Figur nötig sind.
Bool Ready	Um festzustellen ob die Figur alle nötigen Daten hat, um sich zu Zeichnen.
Pen PreviewPen	Stift mit dem die Vorschau der Figur gezeichnet wird.

Methoden

addCoordinate(Point)	Fügt Koordinate zu den Cooridnates hinzu.
paint(Graphics, Pen)	Zeichnet die Figur. Muss überschrieben werden.
drawPreview(Graphics, Point)	Zeichnet Vorschau der Figur anhand des übergebenen Punktes. Muss überschrieben werden.

8.3.3.3 Box

Leitet von „Shape“ ab. Zeichnet sich durch die Methode „paint“ als Rechteck. Sobald zwei Koordinaten enthalten sind wird die „Ready“ Property der Superklasse auf „true“ gesetzt.

8.3.3.4 Circle

Leitet von „Shape“ ab. Zeichnet sich als Ellipse selbst, durch die überschriebene „paint“ Methode. Sobald zwei Koordinaten enthalten sind wird die „Ready“ Property der Superklasse auf „true“ gesetzt.

²⁰ http://en.wikipedia.org/wiki/Bezier_curve

8.3.3.5 Polygon

Leitet von „Shape“ ab. Zeichnet sich selbst als Polygon. Falls zwei aufeinanderfolgende Punkte sehr nahe beieinander sind, wird die „Ready“ Property auf „true“ gesetzt. Ausserdem beinhaltet es auch ein Korrekturmechanismus der Koordinaten der einzelnen Punkte korrigiert wenn die x / y Achse nahe beieinander liegen.

Methoden

correctCoordinatesIfNecessary(Point)	Korrigiert zwei aufeinanderfolgende Koordinaten
correctFirstAndLastCoordinatesIfNecessary	Korrigiert die erste und Letzte Koordinaten.

8.3.3.6 Text

Zeichnet den enthaltenen Text auf die Zeichnungsoberfläche. Je nach ausgewähltem Stift („Thin“, „Medium“ „Thick“) unterscheidet sich die Textgösse und der Textstiel.

„Pen thin“	Textgrösse = 8, Textstiel = „Regular“
„Pen medium“	Textgrösse = 12, Textstiel = „Bold“
„Pen thick“	Textgrösse = 20, Textstiel = „Bold“

Properties / Attribute

StringBuilder TextToDraw	Text der gezeichnet werden soll.
--------------------------	----------------------------------

Methoden

getFontStyle(Pen)	Liefert den Textstiel anhand des übergebenen Stiftes.
getFontSize(Pen)	Liefert die Textgrösse anhand des übergebenen Stiftes.

8.3.3.7 Event

Stellt ein mittels der „Record“ Funktion erfasstes Ereignis in der Applikation dar.

Properties / Attribute

Point Coordinates	Koordinaten wo das Ereignis auftrat.
DateTime Time	Auftrittszeitpunkt des Ereignisses.

Methoden

String getEventAsText	Liefert das Ereignis in der Form „x-y-Sekunden“ zurück.
-----------------------	---

8.3.3.8 Layout

Stellt eine mit dem FIT-System gezeichnete Benutzeroberfläche, mit enthaltenen Symbolen dar. Diese wird für Aufzeichnungen mit dem FIT-System benutzt.

Properties

Bitmap Drawing	Zeichnung der Benutzeroberfläche.
String Name	Name des Layouts

Methoden

Save	Speichert die Zeichnung mit Namen als JPEG File ab.
------	---

8.3.3.9 Program

Einstiegspunkt der Applikation, die den Kopierschutz enthält, das Hauptfenster „MainForm“ erstellt und anzeigt. Der Kopierschutz wird durchgeführt indem die Seriennummer des Geräts mit derjenigen verglichen wird, die in der Software eingetragen ist.

Porperties / Attribute

APPLICATION_PATH	Pfad wo die Layouts und Datenfiles abgespeichert sind.
DEVICE_SERIAL_NUMBER	Seriennummer des Geräts für die, die Software ausgeliefert werden soll.

Methoden

CreateFolders	Erstellt das „fit“ Verzeichnis auf dem Gerät, falls es noch nicht vorhanden ist.
isCoordinatesClose(Point, Point)	Prüft ob zwei Koordinaten nahe bei einander liegen.
GetSerialNumber	Liefer die Seriennummer des HP PDA zurück.

8.3.3.10 Recording

Repräsentiert eine Aufzeichnung von Ereignissen, die mittels der „Record“ Funktion der Applikation durchgeführt wird. Ist die Aufzeichnung zu Ende werden die Daten in ein Datenfile im .fitdb Format abgespeichert.

Properties / Attribute

IDENTIFYER	Identifiziert den PDA Typ von dem die Aufzeichnung kommt.
String FileName	Name der Aufzeichnung.
DateTime RecordingStarted	Startzeitpunkt der Aufzeichnung
List<Event> RecordedEvents	Alle erfassten Ereignisse der Aufzeichnung

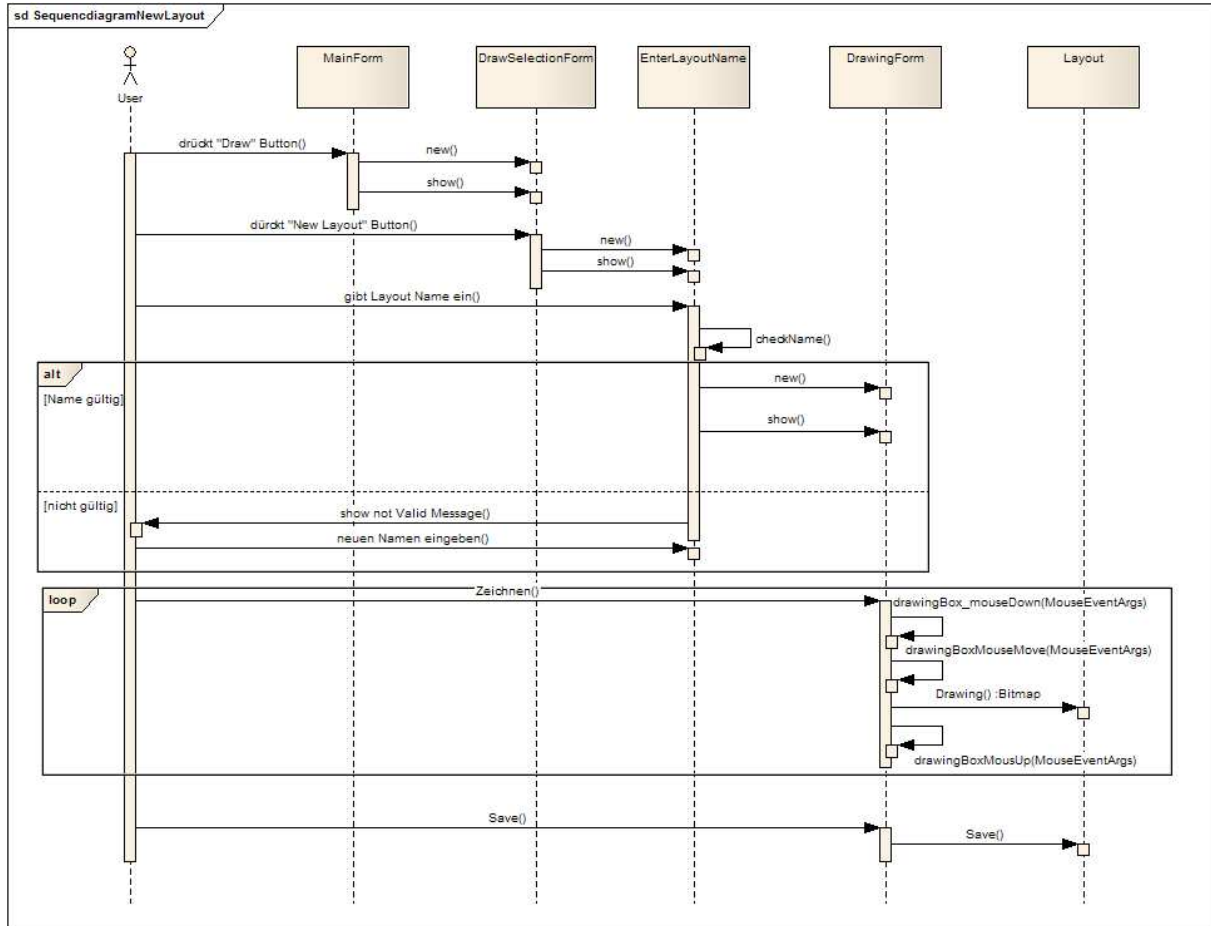
Methoden

Save	Speichert die gesamte Aufzeichnung in die „FileName.fitdb“ Datei.
------	---

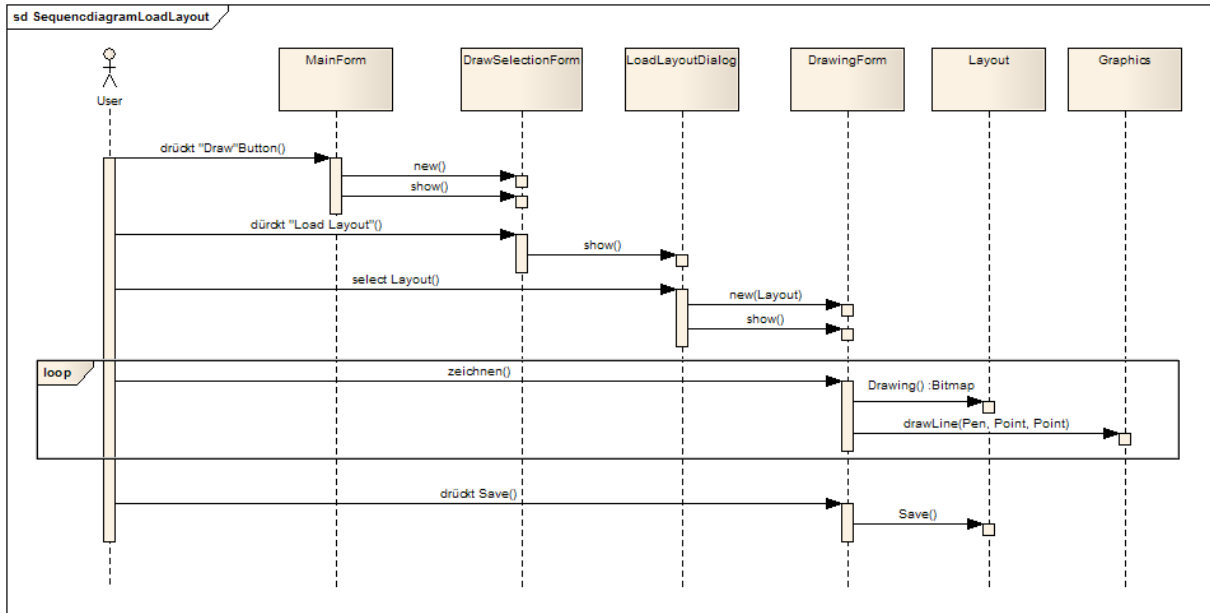
8.4 Sequenzdiagramme

In den folgenden Sequenzdiagrammen werden die wichtigsten Abläufe der Aufrufe zwischen den Dateien dargestellt.

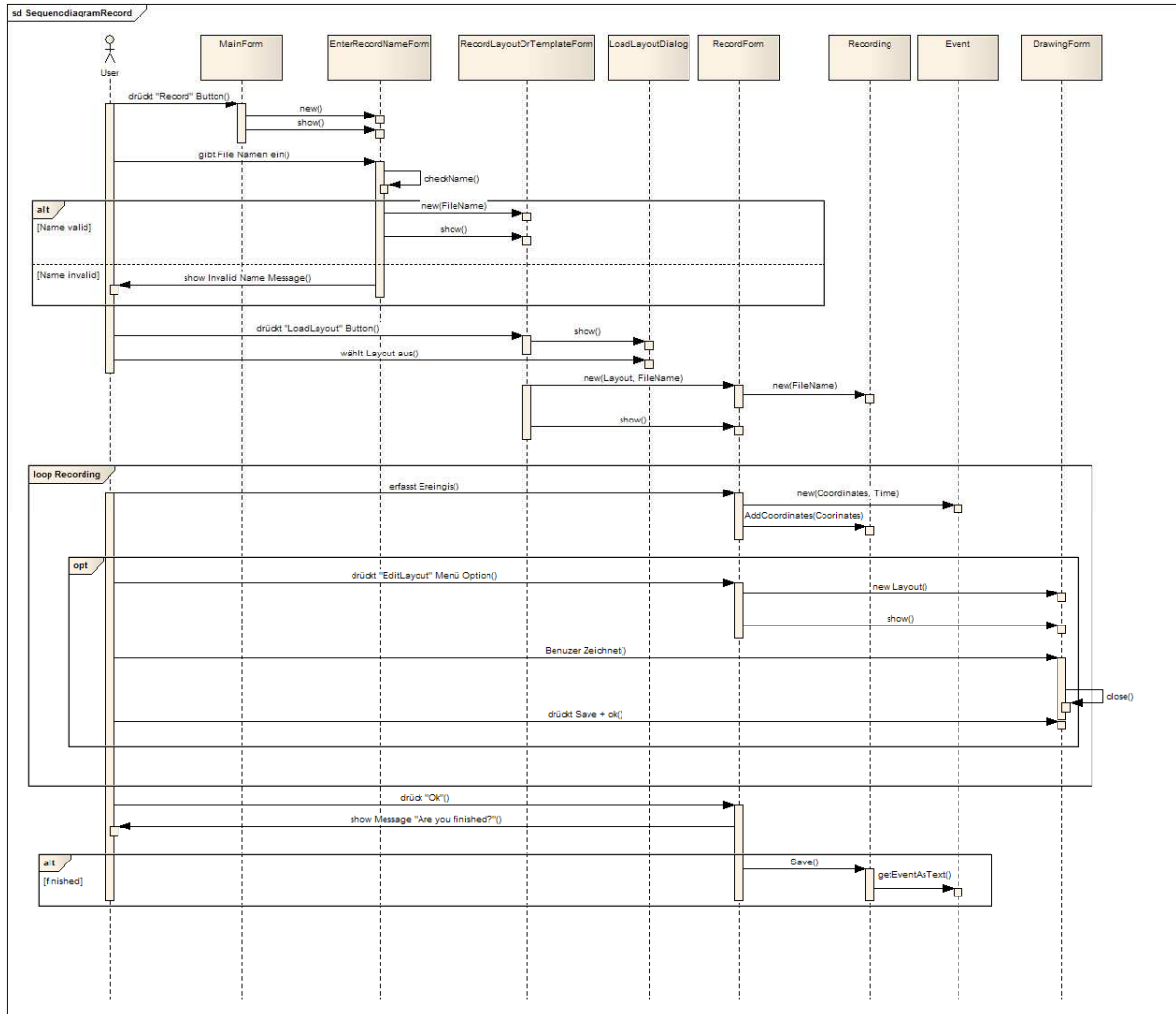
8.4.1 SD1: Erstellen eines neuen Layouts



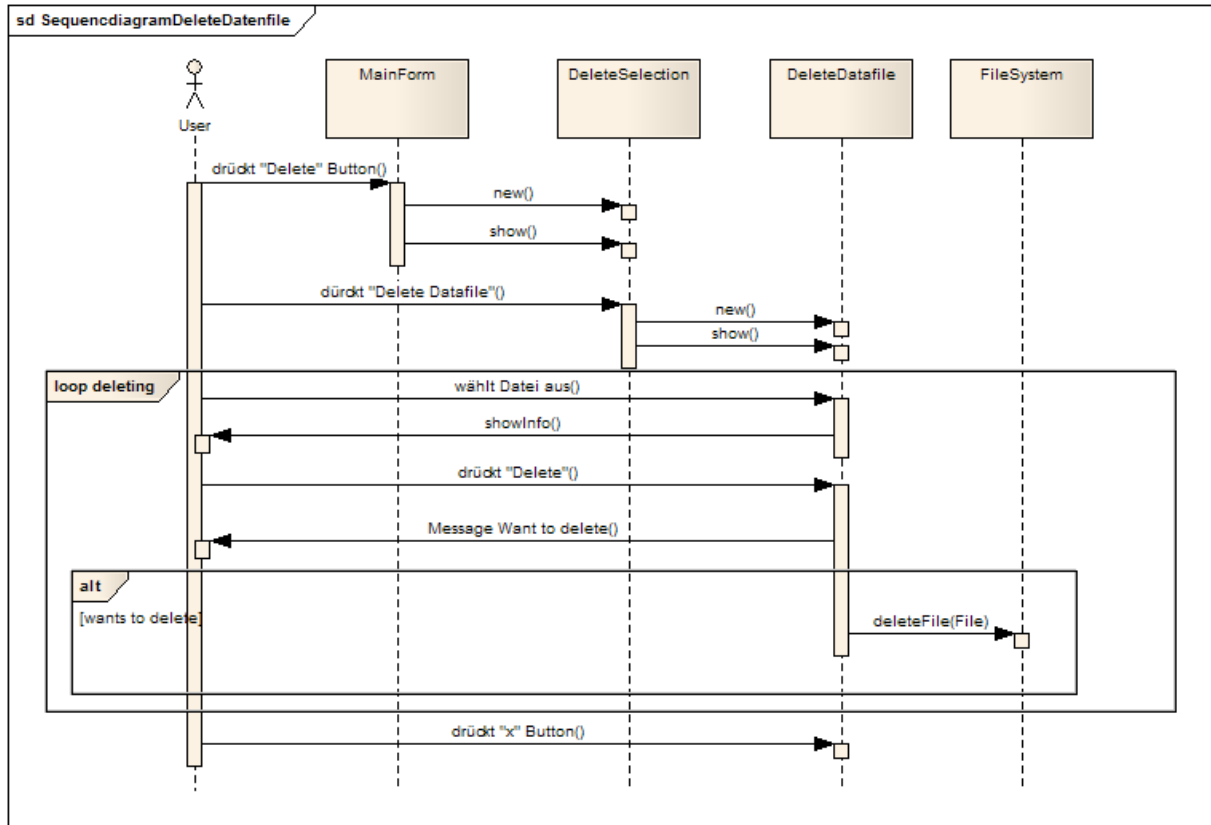
8.4.2 SD2: Bearbeiten eines Layouts



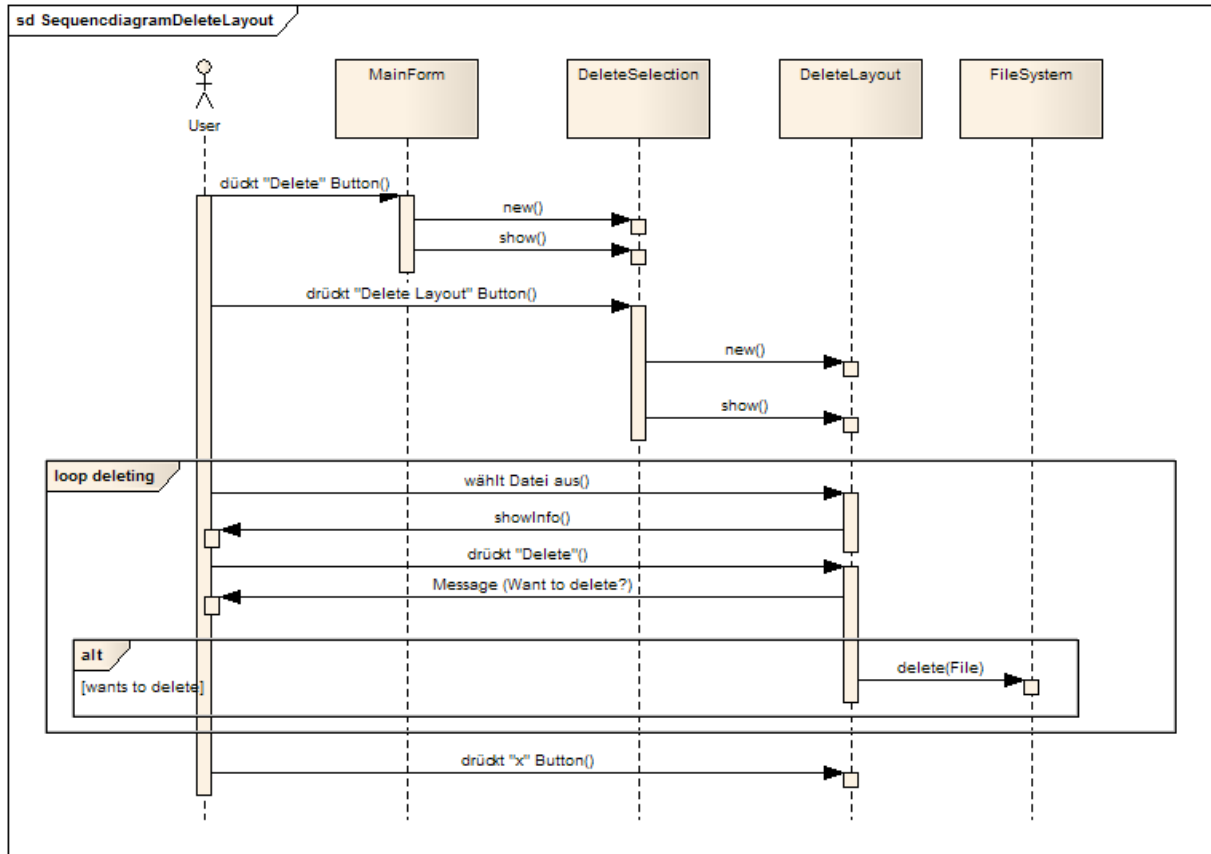
8.4.3 SD3: Aufnehmen eines Datenfiles mit Layout



8.4.4 SD4: Löschen eines Datenfiles



8.4.5 SD5: Löschen eines Layouts



8.5 Design – Entscheidungen

8.5.1 Warum wurde mit C# Programmiert.

Die Alte Applikation war in C geschrieben. Es wäre möglich gewesen dies beizubehalten. Weitere Möglichkeiten wären C++, Java mit J2ME oder .net mit dem Compact Framework gewesen. Am Anfang schien C++ für die Aufgabe am besten geeignet, da es geschwindigkeitsoptierend ist und weniger Speicher benötigt und schon viele Applikation (speziell Grafikapplikationen) für PDAs in dieser Sprache geschrieben wurden. Ich selber programmiere jedoch nicht allzu gern in C oder C++ und wollte die Applikation lieber in Java programmieren. Mit dieser Sprache hatte ich auch viel mehr Erfahrung. Dank moderneren und schnelleren Geräten sollte dies möglich sein. Nach studieren der JAVA ME SDK sah es eher so aus, dass Java sich mehr auf Handy Software Entwicklung konzentriert, als für PDA's. Auch brauchte man eine Spezielle „Java Virtual Machine“ für mobile Geräte. Laut vielen Entwicklern ist die kostenpflichtige J9 JVM von IBM die einzige, die dafür geeignet ist. Auch war es mühsam eine Entwicklungsumgebung für mobile Geräte einzurichten. Viele waren noch in Betaphasen oder kostenpflichtig. Als ich mir C# mit dem .net Compact Framework angeschaut habe, habe ich festgestellt, dass diese bereits auf allen PDAs mit Windows Mobile 6.0 installiert ist. Im Visual Studio 2008 ist die Entwicklung für „Mobile Devices“ für Windows Mobile 5.0 integriert. Für das neuere Betriebssystemversion musste ich nur die Windows Mobile 6.0 SDK herunterladen und installieren. Schon konnte ich einfach ein Projekt erstellen. Dieses konnte einfach und schnell auf dem PDA zum Laufen gebracht werden. Es standen auch einige Zeichnungsfunktionen in dieser Sprache zur Verfügung und man konnte notfalls auch einfach auf native Betriebssystemfunktionen zurückgreifen. Man musste wirklich gar nichts neben der Applikation auf dem PDA installieren um die Applikation lauffähig zu machen. Da dies auch ein Wunsch des Kunden war, mir die Vorteile von C# .net gefielen und ich mich damit eher als mit C++ anfreunden konnte, entschied ich mich die Applikation mit C# zu erstellen.

Zusammenfassende Gründe:

- Einfaches Aufsetzen der Entwicklungsumgebung.
- Objektorientierte gut strukturierbare Programmierung
- Keine zusätzlich Software nötig um auf PDA laufen zu lassen
- Schneller und Speichersparender als Java
- Einfach Software auf PDA zu installieren.
- Übersichtlicher und weniger unvorhergesehene Verhaltensweise als in C#.

8.5.2 Logik in Form Klasse

Die Formklasse die mit Visual Studio erstellt wird, ist in zwei Dateien aufgeteilt. Die .Designer Datei enthält alle mit dem „GUI Designer“ erstellten Komponenten. Die andere Datei ist dazu da, die Logik des GUI zu implementieren. Da dies so vorgesehen ist wurde es auch so getan. Dies funktioniert in fast allen Form Klassen sehr gut und ist dort auch recht übersichtlich. Nur die „DrawingForm“, die ständig erweitert wurde, wurde dadurch so groß, dass die Übersichtlichkeit etwas verloren ging. Im Nachhinein denke ich wäre es besser gewesen, von dieser Klasse noch mehr Logik auszulagern.

9. Tests

9.1 Einleitung

Es wurden Systemtests durchgeführt. Dazu wurde ein Formular erstellt, das alle funktionellen Anforderungen enthält, die implementiert worden sind. Die Unit Tests wurden mit dem SmartDevice UnitTestFramework das im VisualStudio integriert ist durchgeführt.

9.2 Unit Tests

Es wurde ein separates Projekt, nämlich das FIT-SystemTestProject erstellt, das die Unit Tests durchführt. Für jede Klasse der PD, die Methoden haben die zu testen Sinn ergeben, wurde eine Testklasse erstellt die, die alle Methoden der Klasse testet.

9.3 Systemtests

9.3.1 IPAC-214

Systemtestprotokoll

Datum	11.06.10		
Testdurchführer	Michel Rothmund	Visum	MRO
Windows Mobile	6.0		
Testumgebung	Handheld		
Hardware	iPAC-214	Hardware ID	3CC00802WL

Funktion	ID	erfüllt	nicht erfüllt	Absturz
Batteriezustand anzeigen zeichnen	FZ1	x		
Anzeigen des Zeichnungsamen	FZ2	x		
Erstellen einer neuen Zeichnung	FZ3	x		
Speichern einer Zeichnung (mit Datum)	FZ4	x		
Laden einer bestehenden Zeichnung	FZ5	x		
Stift-Funktion dünn	FZ6	x		
Stift-Funktion mittel	FZ7	x		
Stift-Funktion dick	FZ8	x		
Radiergummi-Funktion	FZ9	x		
"Alles Löschen"-Funktion	FZ10	x		
Glättung der Zeichnung	FZ11	x		
Farbauswahl	FZ12	x		
„Undo“ Funktion	FZ13	x		
„Shape Box“ Zeichnen	FZ14	x		
„Shape Circle“ Zeichnen	FZ15	x		
„Shape Polygon“ Zeichnen	FZ16	x		
„Shape Text“ Zeichnen	FZ17	x		
Wechseln zwischen erfassen/editieren	FE1	x		
Anzeigen Namen Datenfile und Anzahl Punkte	FE2	x		
Speichern eines Datenfile	FE3	x		
Anzeigen der Option FE1 oder das Folie verwendet wird	FE4	x		
Zeichnungsfunktion Im Modus Erfassen integrieren	FE5	x		
Letzter Punkt anzeigen	FE6	x		
Löschen eines Datenfiles	FL1	x		
Löschen eines Zeichnungsfiles	FL2	x		
Bestätigungsfenster anzeigen	FL3	x		
Kopierschutz	FK1	x		

Bemerkungen

Alle Funktionen wurden schon früher auf dem PDA getestet und auftretende Fehler behoben.

Massnahmen zu den gefundenen Mängeln

9.3.2 Emulator Windows Mobile 6.0

Systemtestprotokoll

Datum	11.06.10		
Testdurchführer	Michel Rothmund	Visum	MRO
Windows Mobile	6.0		
Testumgebung	Emulator		
Hardware		Hardware ID	

Funktion	ID	erfüllt	nicht erfüllt	Absturz
Batteriezustand anzeigen zeichnen	FZ1	x		
Anzeigen des Zeichnungsname	FZ2	x		
Erstellen einer neuen Zeichnung	FZ3	x		
Speichern einer Zeichnung (mit Datum)	FZ4	x		
Laden einer bestehenden Zeichnung	FZ5	x		
Stift-Funktion dünn	FZ6	x		
Stift-Funktion mittel	FZ7	x		
Stift-Funktion dick	FZ8	x		
Radiergummi-Funktion	FZ9	x		
"Alles Löschen"-Funktion	FZ10	x		
Glättung der Zeichnung	FZ11	x		
Farbauswahl	FZ12	x		
„Undo“ Funktion	FZ13	x		
„Shape Box“ Zeichnen	FZ14	x		
„Shape Circle“ Zeichnen	FZ15	x		
„Shape Polygon“ Zeichnen	FZ16	x		
„Shape Text“ Zeichnen	FZ17	x		
Wechseln zwischen erfassen/editieren	FE1	x		
Anzeigen Namen Datenfile und Anzahl Punkte	FE2	x		
Speichern eines Datenfile	FE3	x		
Anzeigen der Option FE1 oder das Folie verwendet wird	FE4	x		
Zeichnungsfunktion Im Modus Erfassen integrieren	FE5	x		
Letzter Punkt anzeigen	FE6	x		
Löschen eines Datenfiles	FL1	x		
Löschen eines Zeichnungsfiles	FL2	x		
Bestätigungsfenster anzeigen	FL3	x		
Kopierschutz	FK1			x

Bemerkungen
Der Kopierschutz benötigt die iPAQUtil.dll. Diese ist im Emulator nicht vorhanden deshalb stürzt das Programm ab.
Massnahmen zu den gefundenen Mängeln
Bool Variable definiert die gesetzt werden muss und aussagt ob die Software auf dem Emulator oder gerät läuft.

9.3.3 Bug List

ID	Beschreibung	Priorität	Status
01	Zeichnung wird überschrieben wenn es durch ein Gui Element (Popup) überlagert wird	HIGH	erledigt
02	Wenn man anfängt auf Picture Box zu Zeichnen wir sofort alles Schwarz.	HIGH	erledigt
03	Nicht alle Hardware Buttons werden durch die Applikation ausgeschaltet. Dies ist nicht so schlimm da Applikation im Hintergrund weiterläuft und keine Daten verloren gehen.	LOW	offen
04	Layout image zu klein konnte in einem gewissen Bereich nicht zeichnen und nichts Recorden	HIGH	erledigt
05	Jeder nicht nur der Letzte Erfasste Punkt wird angezeigt	HIGH	erledigt
06	Rubber Problem. Falls ein Shape ausgewählt oder Soothing aktiviert war konnte man nicht radieren. Sondern man versuchte eine Transparentes Shape zu zeichnen oder eine Transparente geglättete Zeichnung zu erstellen.	HIGH	erledigt
07	Dateiformat das anhand der Informationen vom Auftraggeber erstellt wurde konnte von FIT Manager nicht gelesen werden.	MEDIUM	offen.
08	Kopierschutz muss ausgelassen werden wenn man die Software auf dem Emulator ausführt. (Wurde mit bool Variable gelöst)	LOW	erledigt
09	„Undo“ Funktion führt bei „Edit Layout“ Option zu Problemen wenn zurück in die RecordForm gewechselt wird.	MEDIUM	erledigt
10	Absturz wenn kein Datenpunkt erfasst und durch „Edit Layout“ in von „DrawForm“ in die „RecordForm“ zurückgewechselt wird.	HIGH	Offen
11	Bei der Funktion „Shape Text“ kann der eingegebene Text nicht korrigiert werden.	LOW	Offen

10. Projekt Management Plan

10.1 Einleitung

Dieses Kapitel bezieht sich auf die Durchführung der Bachelorarbeit „GUI-Graphikgenerator für mobiles Erfassungssystem (PDA)“ des Frühjahrssemester 2010. Es enthält den Software Entwicklungsplan, applikationsspezifischen Risiken, den Iterationsplan sowie die Auswertung der Iterationen. Das Projektmanagement erfolgte basierend auf dem „Agil mit RUP“²¹ aus der Software Engineering 3 Vorlesung. Ich habe mich für „Agil mit RUP“ entschieden, weil es für kleine Projekte geeignet ist. Es heisst es sei geeignet ab zwei Projektmitarbeiter. Da ich jedoch keine Projektmanagement Methode gefunden habe die nur für eine Person geeignet ist, habe ich diese trotzdem ausgewählt. Ein weiterer Grund war, dass die Vorgängerapplikation mit dem Rational Unified Process (RUP) durchgeführt wurde und ich hoffte dadurch einige Artefakte die gleich bleiben von dort übernehmen zu können.

10.2 Software Development Plan

10.2.1 Projektübersicht

Unter dem Namen FIT- System (Flexible Interface Technik) hat die Arbeitsgruppe für Systemergonomie des Instituts für Hygiene und Arbeitsphysiologie der ETH Zürich ein System zur Erfassung und Auswertung von Prozessereignissen entwickelt. Bei diesem Verfahren werden mittels eines Handheld - Computers bestimmte Abläufe aufgezeichnet und später am PC ausgewertet und analysiert. Der Zweck des FIT – Systems ist die Analyse gewisser Arbeitsschritte und deren Optimierung.²²

Diese Software lief auf veralteten Handheld „Devices“. Ziel dieser Arbeit ist dieselbe Funktionalität auf ein Handheld mit modernem Betriebssystem (Windows Mobile 6.0) zu portieren. Des Weiteren sollte dem Benutzer Unterstützung beim Zeichnen seiner flexiblen Benutzeroberfläche in Form von einer Glättung der Zeichnung gewährleistet werden. Zudem wäre es schön, wenn der Benutzer die Farbe seines Zeichnungsstiftes wechseln könnte. Falls genügend Zeit vorhanden ist, besteht noch die Möglichkeit die „Signierfunktion“, die bis anhin auf dem Desktop PC durchgeführt wurde auf den PDA zu verlagern.

10.2.2 Lieferumfang

Der Lieferumfang des Projektes stellt sich folgendermassen zusammen:

Beschreibung	Abgabeform	Datum
FIT-System Applikation für iPAC-214	elektronisch	18.06.2010
Kompletter Source Code	elektronisch	18.06.2010
Bericht + Poster	Papier	18.06.2010
Abstract für Broschüre	Elektronisch	11.06.2010

²¹ Agil RUP Siehe Vorlesung SE3 von L. Müller AgileMethoden_Folien_2_proSeite2.pdf

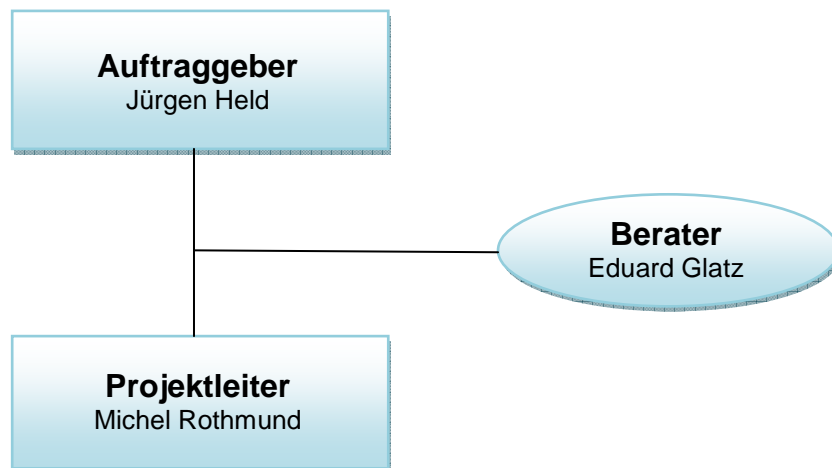
²² Dieser Abschnitt wurde aus dem Bericht FitDraw[1].pdf Kapitel 10.1.1 Projektübersicht übernommen

10.2.3 Projektorganisation

10.2.3.1 Prozessmodell

Als Prozessmodell wird „Agil mit RUP“²³ (Siehe Kapitel 10.1 Einleitung) verwendet. Das Projekt ist in 7 Iterationen aufgeteilt, die je durch einen Meilenstein abgeschlossen werden.

10.2.3.2 Organisationsstruktur



Das gesamte Projekt wird von Michel Rothmund durchgeführt und durch den Betreuer Eduard Glatz überwacht und betreut. Der Auftraggeber der Arbeit ist Jürgen Held.

²³ Agil RUP Siehe Vorlesung SE3 von L. Müller AgileMethoden_Folien_2_proSeite2.pdf

10.2.3.3 Kontaktangaben

Auftraggeber:	Jurgen Held ETH Spin-off Systemergonomie Technoparkstr. 1 8005 Zürich Tel: +41 44 632 26 16 Email: jheld@ethz.ch
Betreuer	Eduard Glatz Hochschule Rapperswil Oberseestr. 10 Postfach 1475 8649 Rapperswil Tel: +41 55 222 49 04 Email: eglatz@hsr.ch
Projektteam	Michel Rothmund Feldhofstr. 24 8600 Dübendorf Tel: +41 44 820 08 24 Email: mrothmun@hsr.ch

10.2.3.4 Sitzungen

Es wurde entschlossen jede Woche eine Sitzung mit dem Betreuer durchzuführen, die normalerweise am Donnerstagmorgen in der Cafeteria stattfinden und ca. 30 Minuten dauern soll. Mit dem Auftraggeber wird je nach Bedarf eine Sitzung abgemacht, die im Technopark Zürich stattfinden und ca. 1 Stunde dauern soll.

10.2.4 Projektmanagement

10.2.4.1 Ziele

Das Projektmanagement soll die Risiken der Projektes minimieren und den Projektverlauf so planen, dass die zeitlich gesetzten Meilensteine erreicht und eingehalten werden können. Bei jedem ausser dem ersten Meilenstein wird ein funktionsfähiger Prototyp erstellt, welcher gewisse Funktionen erfüllen muss. Dieser wird dem Auftraggeber vorgeführt, vorauf hin dieser Verbesserungswünsche einbringen kann.

10.2.4.2 Konfigurationsmanagement

Das Konfigurationsmanagement wird mittels Subversion 1.6.9 durchgeführt. Somit ist die Verwaltung der Erzeugnisse (Dokumente, Code, Grafiken usw.) sichergestellt. Dazu wird ein Subversion Server der HSR benutzt. Die Versionierung und ein kontinuierliches Backup der Dateien erfolgt bei diesem System automatisch.

10.2.4.3 Vergleichbare Produkte Lösungen

Das Verfahren des FIT – Systems auf dem Handheld – Computer ist gegenwärtig einmalig und ist deswegen von dem Erfinder Jürgen Held patentieren lassen worden. Fürs erstellen von Zeichnungen gibt es auf Handheld Computer bereits verschiedene Programme. Ich bin bei meinen Nachforschungen auf drei Verschiedene Zeichnungsprogramme gestossen, die kurz vorgestellt werden.

PocketPictures²⁴

Bietet Zeichnungsfunktion mit 6 verschiedenen Stiftdicken und beliebigen Farben. Das Programm bietet weiter Kopierfunktionen, Ausschneid und Verschiebungsfunktionen an. Des Weiteren ist eine Funktion „Figur“ für (Linien, Rechtecke und Ellipsen (Kreise) enthalten. Es bietet keine Glättung an, auch das Zeichnen von Text und Polygonen werden nicht unterstützt. Zeichnungen können in den Formaten (PNG, JPG, GIF und BMP) abgespeichert werden.



²⁴ <http://code.google.com/p/pocketpicture/downloads/list>

Vspainter

Es gibt eine kostenpflichtige²⁵- und Freeware²⁶- Version (VsPainter LE) mit verringertem Leistungsumfang.

Vspainter LE bietet grundlegende Zeichen- und Malfunktionen für den Pocket PC und dient als Appetitmacher für die Kaufversion der gleichen Software. Die Freeware bietet zwar keine fortgeschrittene Bildbearbeitung, eignet sich aber als Skizzenblock. Die Kaufversion machte, in einem Vorführungsvideo, eher den Eindruck für Künstler geeignet zu sein. In der Trial- Version konnte ich nicht abspeichern. Bietet keine Glättung an. Es hat viele für die FIT-Applikation unnütze Figuren und Funktionen.



PDAcraft Paint²⁷

Enthält etwa die gleiche Funktionalität wie das normale Paint für Windows. Die Vorschau beim Zeichnen von Figuren ist klar schneller als bei meinem Programm. Eine Glättung wird jedoch nicht angeboten. Jedoch können Zeichnungen nur, im viel Speicherplatz verbrauchendem, .bmp Format abgespeichert werden. Keine Unterstützung für das Zeichnen von Polygonen.



²⁵ <http://www.vetasoft.com/index.php?index=12&language=en>

²⁶ <http://vspainter-le.softonic.de/pocketpc>

²⁷ <http://pdacraft.com/paint.php>

10.2.5 Risk List

Risiko	Auswirkung	Kosten	Wahrscheinlichkeit in %	Risiko = K * WK	Massnahme
Programmiersprache für Projekt nicht geeignet (z.B. langsam)	Projekt muss in anderer Programmiersprache neu geschrieben werden	80h	5%	5% * 80h = 4h	Research ob gewünschte Funktionen vorhanden sind.
Glättungsalgorithmus wird nicht verstanden	Kann nicht implementiert werden da nicht verstanden	20h	5%	5% * 20 = 1h	Algorithmus mit Betreuer besprechen.
Glättungsalgorithmus überraschend komplex und schwierig zu implementieren	Viel Zeitverbrauch bei der Implementation	40h	20%	20% * 40 = 8h	Algorithmus Literaturstudium (auf Signierfunktion verzichten)
Gewisse Programmiersprachen Features sind in der Device Version nicht vorhanden	Es muss nach Alternativen gesucht werden Zeitverlust	12h	30%	30% * 12 = 3h	Genug Zeit einplanen
Signierfunktion zu schwierig auf Pocket PC zu realisieren	Signieren muss weiterhin auf dem Desktop durchgeführt werden	0	0	0	Signierfunktion wird weggelassen
Total Risiko				16h	

Um diesem Risiko gerecht zu werden wurde ein Arbeitspaket „Risiko“ definiert. (Siehe Kapitel 10.2.3.23). Ausserdem ist die ganze sechste Iteration (Iteration „Sign“) optional und kann falls mehr Zeit für die Glättung benötigt wird ausgelassen werden.

10.2.6 Personalplanung

Die Arbeit wird als Einzelarbeit durchgeführt. Der Arbeitsaufwand ist auf 364 Stunden geplant. Für die Arbeit stehen 17 Wochen zur Verfügung was eine durchschnittliche Arbeitszeit von 21 Stunden die Woche ergibt. Diese ist jedoch je nach Ferien + Feiertage unterschiedlich. Bei Auftritt eines Risikos werden die geplanten Reservestunden eingesetzt und falls nötig die Signierfunktion gestrichen.

10.2.7 Entwicklungsprozess

10.2.7.1 Einzusetzende Tools

Für die Realisierung des Projektes werden folgende Produkte verwendet:

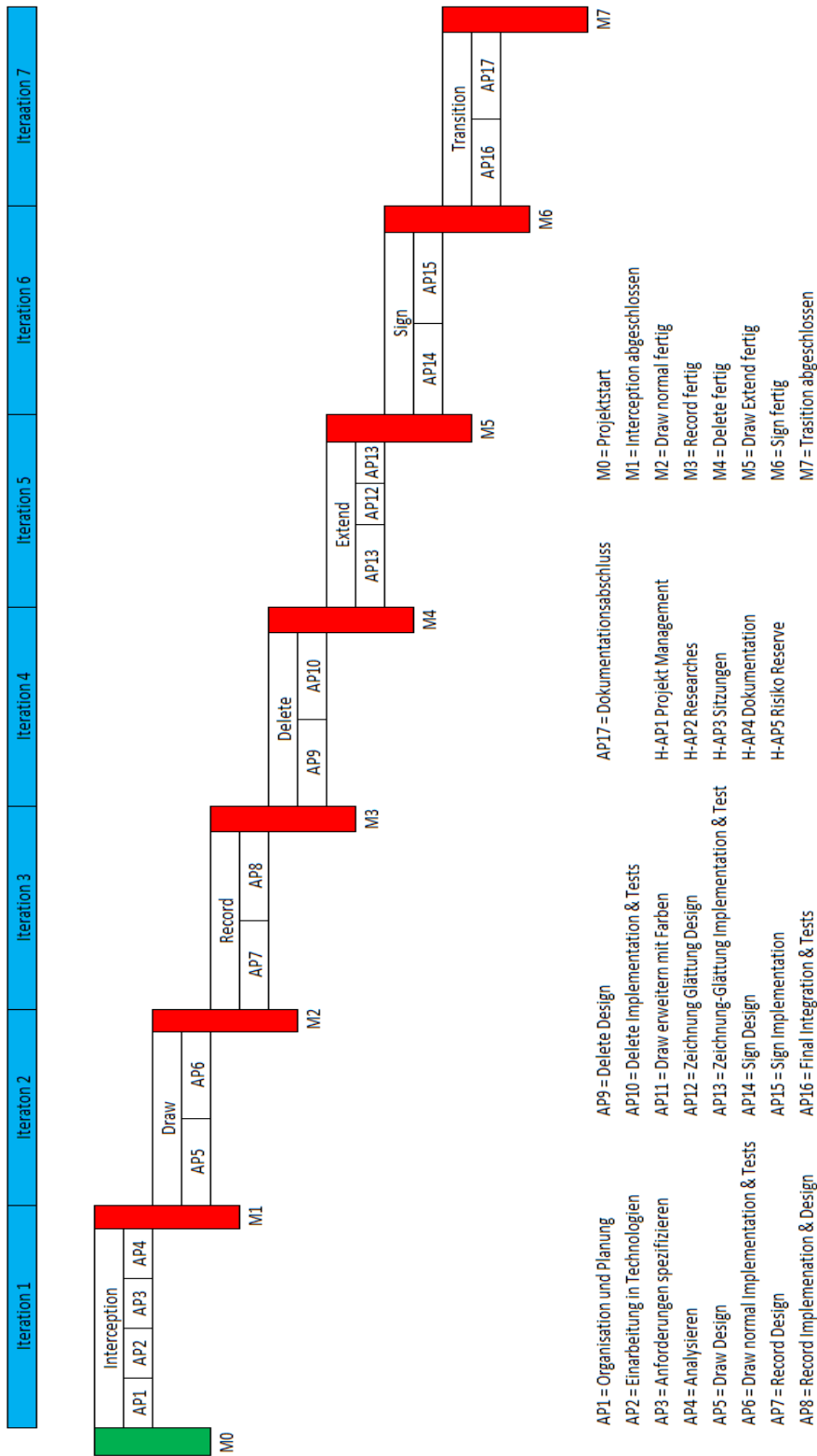
Betriebssystem:	Windows 7 Windows XP Windows Mobile 6.0
Entwicklungsumgebung:	Visual Studio 2008 Windows Mobile 6.0 Professional SDK
Design Tool:	Enterprise Architect
SVN Server:	Subversion 1.6.9
SVN Client:	TortoiseSVN 1.6.7
PDA Datenaustausch Software:	Windows Mobile Device Center (WMDC)
Dokumentation	Microsoft Word 2007 Microsoft Excel 2007 Adobe Acrobat 9
Präsentation	Microsoft Powerpoint 2007

10.2.7.2 Einzusetzende Methoden & Techniken

Programmiersprache	C# .net
Bibliotheken:	.net Compact Framework 2.0

10.3 Iteration Plan

10.3.1 Übersicht Iterationen



10.3.2 Arbeitspakete

Es werden 2 verschiedene Arbeitspakete definiert.

AP Normale Arbeitspakete: sind zeitlich begrenzt und werden nach einer gewissen Anzahl Arbeitsstunden abgeschlossen.

H-AP Hilfsarbeitspakete, welche während des gesamten Projektverlaufs zu bearbeiten sind.

ID	Titel	Aufwand	Priorität
AP1	Organisation und Planung	15	1
AP2	Einarbeitung in Technologien	21	1
AP3	Anforderungen spezifizieren	12	1
AP4	Analysieren	7	1
AP5	Draw Design	6	1
AP6	Draw normal Implementation & Tests	52	1
AP7	Record Design	4	1
AP8	Record Implementation & Tests	30	1
AP9	Delete Design	3	1
AP10	Delete Implementation & Tests	10	1
AP11	Draw erweitern mit Farben	8	2
AP12	Zeichnung Glättung Design	6	1-2
AP13	Zeichnung-Glättung Implementation & Test	28	1-2
AP14	Sign Design	6	3
AP15	Sign Implementation	34	3
AP16	Final Integration & Systemtests	16	1
AP17	Dokumentationsabschluss	22	1
AP18	Kopierschutz	4	1-2
H-AP1	Projekt Management	10	1
H-AP2	Reseraches	16	1
H-AP3	Sitzungen	14	1
H-AP4	Dokumentation	12	1
H-AP5	Risiko Reserve	16	1
Total		360	

Prioritäten:

1. Arbeitspaket, für erfolgreichen Abschluss unverzichtbar.
2. Dem Auftraggeber liegt sehr viel daran dass diese Arbeitspakete abgeschlossen sind.
3. Wäre schön wenn Arbeitspaket abgeschlossen werden könnte.

10.3.2.1 AP1: Organisation & Planung

Als erstes müssen die Unterlagen, Software und weitere Informationen beschafft werden. Zudem müssen administrative Arbeiten erledigt werden.

Arbeitsschritte	Aufwand
Kick off Meeting mit Herrn Glatz	1h
Erste Sitzung mit Jürgen Held	2h
Vorlagen für Dokumente erstellen	3h
Installationen von Software	8h
SVN einrichten und vorbereiten	1h
Totaler Aufwand	<u>15h</u>

Resultate:

- Dokumentvorlagen

Zu bearbeitende Dokumente:

- Projektmanagementplan

10.3.2.2 AP2: Einarbeitung in Technologien

Wegen fehlender Erfahrung, in der Applikationsentwicklung für Pocket PCs, ist eine Einarbeitung nötig, die wie folgt geplant wird:

Arbeitsschritte	Aufwand
Umgang mit iPAQ	1h
Applikationen für Datenaustausch mit iPAQ	1h
Theorie lernen für Programmieren auf iPAQ J2ME + .net CF	12h
Applikationsentwicklung mit eclipse (mtj) visual studio	2h
Programmieren einer eigenen iPAQ - Applikation	2h
FIT System Applikationsdokumentation	3h
Totaler Aufwand	<u>21h</u>

Zu bearbeitende Dokumente:

- Glossar
- Literaturverzeichnis

10.3.2.3 AP3: Anforderungsspezifikation

Die Aufgabenstellung wird analysiert und zerlegt, folglich werden die Ziele und Anforderungen für das Projekt definiert. Anschliessend wird der gesamte Arbeitsumfang zeitlich abgeschätzt (H-AP1).

Arbeitsschritte	Aufwand
Funktionale Anforderungen definieren	4h
Nichtfunktionale Anforderungen definieren	3h
Use Cases erstellen	5h
Totaler Aufwand	<u>12h</u>

Resultate:

- Funktionale Anforderungen
- Nichtfunktionale Anforderungen
- Use Cases

Zu bearbeitende Dokumente:

- Anforderungsspezifikation
- Analyse Dokument

10.3.2.4 AP4: Analysieren

Analyse des Codes der alten Fit Applikation und der Anforderungen

Arbeitsschritte	Aufwand
Anforderungen analysieren	2h
Code der alten Applikation analysieren	5h
Totaler Aufwand	<u>7h</u>

Resultate:

Zu bearbeitende Dokumente:

- Design Model
- Software Architecture Document

10.3.2.5 AP5: Draw Design

In diesem Arbeitspaket werden die Klassen- und Sequenzdiagramme für die Zeichnungs- (Draw) Funktionen der Applikation besprochen und erstellt. Diese Diagramme dienen als Vorlage für die Implementation der Software-Klassen. In diesem Arbeitspaket wird auch die Systemarchitektur festgelegt.

Arbeitsschritte	Aufwand
Festlegen der Systemarchitektur	1h
Erstellen der Klassendiagramme	2h
Erstellen der Sequenzdiagramme	3h
Totaler Aufwand	<u>6h</u>

Resultate:

- Klassendiagramm
- Sequenzdiagramm

Zu bearbeitende Dokumente:

- Software Architecture Document

10.3.2.6 AP6: Draw Implementation

Alle zu implementierenden Zeichnungs- (Draw) Funktionen werden anhand der Design Entscheide programmiert und einzeln getestet. Sind alle Funktionen fertig gestellt, wird ein letzter „Integrations“ Test der Applikation durchgeführt. (Während dem Programmieren soll im Sourcecode auch eine Beschreibung der Klasse als Kommentar erstellt werden)

ID	Draw - Funktionen	Priorität	Aufwand
FZ1	Batteriezustand anzeigen	1	2h
FZ2	Anzeigen des Zeichnungsnamen	1	2h
FZ3	Erstellen einer neuen Zeichnung	1	8h
FZ4	Speichern einer Zeichnung (mit Datum)	1	14h
FZ5	Laden einer bestehenden Zeichnung	1	10h
FZ6	Stift-Funktion dünn	1	2h
FZ7	Stift-Funktion mittel	1	2h
FZ8	Stift-Funktion dick	1	2h
FZ9	Radiergummi-Funktion	1	3h
FZ10	"Alles Löschen" - Funktion	1	2h
Totaler Aufwand:			47h

Arbeitsschritte	Aufwand
Programmieren & Implementieren aller Zeichnungsfunktionen	47h
Final Integration Test	3h
Klassen oder Funktionsbeschreibung	2h
Totaler Aufwand	<u>52h</u>

Resultate:

- Fertige Applikation mit Zeichnungsfunktion
- Modultests der einzelnen Funktionen
- Final Integration Test Protokoll
- Klassen oder Funktionsbeschreibung

Zu bearbeitende Dokumente:

- Design Dokument
- Test Dokument

10.3.2.7 AP7: Record Design

In diesem Arbeitspaket werden die Klassen- und Sequenzdiagramme für die Aufnahme (Record) Funktionen der Applikation besprochen und erstellt. Diese Diagramme dienen als Vorlage für die Implementation der Software-Klassen.

Arbeitsschritte	Aufwand
Erstellen der Klassendiagramme	2h
Erstellen der Sequenzdiagramme	2h
Totaler Aufwand	<u>4h</u>

Resultate:

- Klassendiagramm
- Sequenzdiagramm

Zu bearbeitende Dokumente:

- Design Dokument

10.3.2.8 AP8: Record Implementation

Alle zu implementierenden Aufnahme (Record) Funktionen werden anhand der Design Entscheidung programmiert und einzeln getestet. Sind alle Funktionen fertig gestellt, wird ein letzter „Integrations“ Test der Applikation durchgeführt. (Während dem Programmieren soll im Sourcecode auch eine Beschreibung der Klasse als Kommentar erstellt werden)

ID	Record - Funktionen	Priorität	Aufwand
FE1	Wechseln zwischen Erfassen/Editieren	1	5h
FE2	Anzeigen des Datenfilenamens und Anzahl Punkte	1	2h
FE3	Speichern eines Datenfile	1	3h
FE4	Anzeigen der Option FE1 oder das Folie verwendet	1	2h
FE5	Zeichnungsfunktion Im Modus Erfassen integrieren	1	10h
FE6	Letzter Punkt blinkend	3	3h
Totaler Aufwand:			25h

Arbeitsschritte	Aufwand
Programmieren & Implementieren aller Aufnahmefunktionen	28h
Final Integration Test	3h
Klassen oder Funktionsbeschreibung	2h
Totaler Aufwand	<u>30h</u>

Resultate:

- Fertige Applikation mit Aufnahmefunktion
- Modultests der einzelnen Funktionen
- Final Integration Test Protokoll
- Klassen oder Funktionsbeschreibung

Zu bearbeitende Dokumente:

- Design Dokument
- Test Dokument

10.3.2.9 AP9: Delete Design

In diesem Arbeitspaket werden die Klassen- und Sequenzdiagramme für die Löschen (Delete) Funktionen der Applikation besprochen und erstellt. Diese Diagramme dienen als Vorlage für die Implementation der Software-Klassen.

Arbeitsschritte	Aufwand
Erstellen der Klassendiagramme	1h
Erstellen der Sequenzdiagramme	2h
Totaler Aufwand	<u>3h</u>

Resultate:

- Klassendiagramm
- Sequenzdiagramm

Zu bearbeitende Dokumente:

- Design Dokument

10.3.2.10 AP10: Delete Implementation

Alle zu implementierenden Lösch- (Delete) Funktionen werden anhand der Design Entscheide programmiert und einzeln getestet. Sind alle Funktionen fertig gestellt, wird ein letzter „Integrations“ Test der Applikation durchgeführt. (Während dem Programmieren soll im Sourcecode auch eine Beschreibung der Klasse als Kommentar erstellt werden)

ID	Record - Funktionen	Priorität	Aufwan
FL1	Löschen eines Datenfiles	1	3h
FL2	Löschen eines Zeichnungsfiles	1	3h
FL3	Bestätigungsfenster anzeigen	1	1h
Totaler Aufwand:			7h

Arbeitsschritte	Aufwand
Programmieren & Implementieren aller Löschfunktionen	7h
Final Integration Test	2h
Klassen oder Funktionsbeschreibung	1h
Totaler Aufwand	<u>10h</u>

Resultate:

- Fertige Applikation mit Löschfunktion
- Modultests der einzelnen Funktionen
- Final Integration Test Protokoll
- Klassen oder Funktionsbeschreibung

Zu bearbeitende Dokumente:

- Design Dokument
- Test Dokument

10.3.2.11

AP11: Draw erweitern mit Farbe

Dieses optionale Arbeitspaket mit hoher Priorität erweitert die Draw Funktion mit der Möglichkeit die Farbe des Stiftes zu verändern. Sind alle Funktionen fertig gestellt, wird ein letzter „Integrations“ Test der Applikation durchgeführt

Arbeitsschritte	Aufwand
FZ12 Farbauswahl	5h
Final Integration Test	2h
Klassen oder Funktionsbeschreibung	1h
Totaler Aufwand	<u>8h</u>

Resultate:

- Zeichnungsapplikation um Farbauswahl erweitert.
- Modultests der einzelnen Funktionen
- Final Integration Test Protokoll

Zu bearbeitende Dokumente:

- Design Dokument
- Test Dokument

10.3.2.12 AP12: Zeichnung Glättung Design

In diesem Arbeitspaket werden die Klassen- und Sequenzdiagramme für die Zeichnung- Glättung- Funktionen der Applikation besprochen und erstellt. Diese Diagramme dienen als Vorlage für die Implementation der Software-Klassen.

Arbeitsschritte	Aufwand
Erstellen der Klassendiagramme	2h
Erstellen der Sequenzdiagramme	4h
Totaler Aufwand	<u>6h</u>

Resultate:

- Klassendiagramm
- Sequenzdiagramm

Zu bearbeitende Dokumente:

- Design Dokument

10.3.2.13 AP13: Zeichnung-Glättung Implementation

Dieses ist ein optionales Arbeitspaket, mit der höchsten Priorität, beinhaltet die Funktion der Glättung der Zeichnung. Sind alle Funktionen fertig gestellt, wird ein letzter „Integrations“ Test der Applikation durchgeführt. (Während dem Programmieren soll im Source Code auch eine Beschreibung des Algorithmus als Kommentar erstellt werden)

ID	Record - Funktionen	Priorität	Aufwan
FZ11	Glättung der Zeichnung	1-2	13h
Totaler Aufwand:			13h

Arbeitsschritte	Aufwand
Zeichnung Glättung Research	10h
Programmieren & Implementieren aller Löschroutinen	13h
Final Integration Test	3h
Klassen oder Funktionsbeschreibung	2h
Totaler Aufwand	<u>28h</u>

Resultate:

- Fertige Applikation mit Löschroutine
- Modultests der einzelnen Funktionen
- Final Integration Test Protokoll
- Klassen oder Funktionsbeschreibung

Zu bearbeitende Dokumente:

- Design Dokument
- Test Dokument

10.3.2.14 AP14: Sign Design

In diesem Arbeitspaket werden die Klassen- und Sequenzdiagramme für die Signier- (Sign) Funktionen der Applikation besprochen und erstellt. Diese Diagramme dienen als Vorlage für die Implementation der Software-Klassen.

Arbeitsschritte	Aufwand
Erstellen der Klassendiagramme	3h
Erstellen der Sequenzdiagramme	3h
Totaler Aufwand	<u>6h</u>

Resultate:

- Klassendiagramm
- Sequenzdiagramm

Zu bearbeitende Dokumente:

- Design Dokument

10.3.2.15 AP15: Sign Implementation

Alle zu implementierenden Signier- (Sign) Funktionen werden anhand der Design Entscheide programmiert und einzeln getestet. Sind alle Funktionen fertig gestellt, wird ein letzter „Integrations“ Test der Applikation durchgeführt. (Während dem Programmieren soll im Sourcecode auch eine Beschreibung der Klasse als Kommentar erstellt werden)

ID	Record - Funktionen	Priorität	Aufwan
FS1	Auswahl des Datenfiles	3	1h
FS2	Auswahl des Zeichnungsfiles oder der Folie	3	1h
FS3	Anzeigen der Punkte aus Datenfile	3	2h
FS4	Polygon um Symbol zeichnen	3	12h
FS5	Aufforderung um Namen des Icons einzugeben	3	4h
FS6	Speichern des Datenfiles mit Iconnamen	3	9h
Totaler Aufwand:			29h

Arbeitsschritte	Aufwand
Programmieren & Implementieren aller Signierfunktionen	29h
Final Integration Test	3h
Klassen oder Funktionsbeschreibung	2h
Totaler Aufwand	<u>34h</u>

Resultate:

- Fertige Applikation mit Signierfunktion
- Modultests der einzelnen Funktionen
- Final Integration Test Protokoll
- Klassen oder Funktionsbeschreibung

Zu bearbeitende Dokumente:

- Design Dokument
- Test Dokument

10.3.2.16 AP16: Final Integration & Systemtests

Alle bisherigen, erstellten Funktionen werden in die FIT Applikation integriert. Nach der Integration werden Systemtests durchgeführt, bis das FIT System auf dem IPAQ Windows Mobile einwandfrei läuft.

Arbeitsschritte	Aufwand
Integrieren aller erstellten Funktionen	5h
Systemtests	6h
Fehlerbehebungen	5h
Totaler Aufwand	<u>16h</u>

Resultate:

- FIT Applikation

Zu bearbeitende Dokumente:

- Test Dokument

10.3.2.17 AP17: Dokumentationsabschluss

Alle Dokumente werden auf den Schlusstand gebracht. Sie werden vervollständigt, korrigiert und jeweils einem Review unterzogen. Zudem kommen noch alle Abschluss- und Aufräumarbeiten hinzu.

Arbeitsschritte	Aufwand
Dokumente vervollständigen & korrigieren	11h
Dokumente Reviewen	6h
Abschlussarbeiten (Drucken, CD brennen usw.)	5h
Totaler Aufwand	<u>22h</u>

Resultate:

- Vollständig abgeschlossenes Projekt

Zu bearbeitende Dokumente:

- Projekt Dokumentation

10.3.2.18 AP18: Kopierschutz

Dies ist ein optionales Arbeitspaket mit hoher Priorität. Es soll ein Kopierschutz erstellt werden, der dafür sorgt, dass ein Lizenznehmer die Software nur auf einem Gerät laufen lassen kann.

Arbeitsschritte	Aufwand
FK1 Kopierschutz	4h
Totaler Aufwand	<u>4h</u>

Resultate:

- Kopierschutz

Zu bearbeitende Dokumente:

- Projekt Dokumentation

10.3.2.19 H-AP1: Projekt Management

Das Projekt Management ist eine Tätigkeit, welche sich über die gesamte Projektarbeit hinweg erstreckt. Schwerpunkt liegt bei diesem Paket in den Anfängen des Projektes. Jedoch wird während dem gesamten Verlauf weiter am Management gearbeitet.

Arbeitsschritte	Aufwand
Zeit Plan (Iterationen)	4h
Projekt Plan(Arbeitspakete, Meilensteine)	4h
Soll ist Vergleich	2h
Totaler Aufwand	<u>10h</u>

Zu bearbeitende Dokumente:

- Projekt Management Plan
- Zeitplan
- Zeiterfassung

10.3.2.20 H-AP2: Researches

Während des gesamten Projektes kann es vorkommen, dass Technologien (J2ME & .net CF, Glättungsalgorithmus) auftreten, die noch nicht bekannt sind. Somit braucht es wieder eine Einarbeitung (Researches) in diese Technologien.

Arbeitsschritte	Aufwand
Researches J2ME	8h
.net CF	8h
Totaler Aufwand	<u>16h</u>

Zu bearbeitende Dokumente:

- Glossar
- Literaturverzeichnis

10.3.2.21 H-AP3: Sitzungen & Reviews

Es findet jede Woche eine Sitzung mit dem Betreuer statt. Eine Sitzung mit dem Arbeitgeber findet je nach Bedarf statt. (Häufig nachdem ein Meilenstein erreicht wurde.)

Arbeitsschritte	Aufwand
Sitzungen & Reviews	14h
Totaler Aufwand	<u>14h</u>

Zu bearbeitende Dokumente:

- Protokoll Sitzung mit Betreuer / Arbeitgeber

10.3.2.22 H-AP4: Dokumentation

Das Arbeitspaket Dokumentation, steht für weitere Dokumente, welche nicht in den anderen Paketen behandelt werden. Diese Berichte können während der gesamten Projektzeit erledigt werden.

Arbeitsschritte	Aufwand
Abstract	3h
Technischer Bericht	4h
Schlussbericht	4h
Persönliche Berichte	1h
Totaler Aufwand	<u>12h</u>

Resultate:

- Abstract
- Technischer Bericht
- Schlussbericht
- Persönliche Berichte

Zu bearbeitende Dokumente:

- Abstract
- Technischer Bericht
- Schlussbericht
- Persönliche Berichte

10.3.2.23 H-AP5: Risiken

Die im Kapitel „Risk - List“ aufgelisteten Risiken können während des ganzen Projektes auftreten, obwohl je mehr das Ende naht, desto niedriger wird dieses Risiko. (Die genaue Risiken Auflistung ist unter dem Kapitel Risiko – Management aufgestellt)

Arbeitsschritte	Aufwand
Risikos	16h
Totaler Aufwand	<u>16h</u>

Resultate:

- Risk List

Zu bearbeitende Dokumente:

Projekt Management Plan

10.3.3 Meilensteine

Meilenstein	Datum	Ziel
M0	22.02	- Projektstart
M1	14.03	- Projektmanagementplan / Terminplan erstellt - Anforderungsspezifikation erstellt
M2	10.04	- Implementation Zeichnungs-Applikation fertig gestellt - Implementation getestet
M3	25.04	- Implementation Aufnahme-Applikation fertig gestellt - Implementation getestet
M4	02.05	- Implementation Lösch-Applikation fertig gestellt - Implementation getestet
M5	20.05	- Zeichnungs-Applikation Erweiterung Farben implementiert - Implementation getestet - Zeichnungs-Applikation Erweiterung Glättung implementiert - Implementation getestet
M6	06.06	- Implementation Signier-Applikation fertig gestellt - Implementation getestet
M7	18.06	- Systemintegration & Systemtest fertig - FIT Applikation vollständig lauffähig - Dokumentation abgeschlossen - Projektende - Kopierschutz erstellt

10.3.4 Zeitplan

Siehe FITZeitplan.xlsx

10.4 Iteration Assessment

In diesem Kapitel wird der Soll- und Ist- Aufwand der Arbeitspakete miteinander verglichen. Daraus wird ersichtlich, wie viele Stunden geplant waren und wie viele tatsächlich benötigt wurden. Falls es grössere Abweichungen gibt werden diese erläutert.

10.4.1 Soll- und Ist- Aufwand der Iterationen und Arbeitstakte

Soll- und Ist-Aufwände sind in Stunden angegeben.

ID	Arbeitstakt	Soll	IST
I1	Inception	55	59
AP1	Organisation und Planung	15	13
AP2	Einarbeitung in Technologien	21	27.5
AP3	Anforderungen spezifizieren	12	12.5
AP4	Analysieren	7	6
I2	Draw	58	51.5
AP5	Draw Design	6	4
AP6	Draw normal Implementation & Test	52	47.5
I3	Record	34	40.5
AP7	Record Design	4	4
AP8	Record Implementation & Tests	30	36.5
I4	Delete	13	14.5
AP9	Delete Design	3	4
AP10	Delete Implementation & Tests	10	10.5
I5	Exend	44	59
AP11	Draw erweitern mit Farben	8	4
AP12	Zeichnung Glättung Design	8	5.5
AP13	Zeichnung Glättung Implementation und & Tests	28	49.5
I6	Sign	40	0
AP14	Sign Design	6	0
AP15	Sign Implementation & Tests	34	0
I7	Transition	42	64
AP16	Final Integration & Systemtests	16	5
AP17	Dokumentationsabschluss	22	53.5
AP18	Kopierschutz	4	5.5
H-AP1	Projekt Management	10	17
H-AP2	Reseraches	16	23
H-AP3	Sitzungen	14	24.5
H-AP4	Dokumentation	12	18
H-AP5	Risiko	16	13
Gesamt		360	383

10.4.2 Inception

Die Einarbeitung in Technologien dauerte etwas länger als geplant. Ursachen dafür waren, dass vor allem im Java Umfeld das ganze etwas unübersichtlich war und sich die Technologien, für Handheld Devices sehr entwickelt haben. MTJ²⁸ (Mobile Tools for Java), die Eclipse Version für mobile Geräte, ist noch in der Entwicklungsphase. Es war auch aufwendig eine Entwicklungsumgebung einzurichten. Zudem ist die benötigte JVM (Java Virtual Machine) für mobile Geräte nicht frei Verfügbar. Die beste Plattform war die Java™ ME Plattform SDK²⁹ von SUN. Ein Test zeigte, dass sich diese Entwicklungs-Plattform eher für Software Entwicklung auf Handys eignete. Das C# .net compact Framework war für die Entwicklung für PDAs viel besser geeignet. Nur die Installation des Visual Studios 2008 vom ELSM³⁰ Server bereitete auch Probleme, da es oft während des Downloads zu Abstürzen kam.

10.4.3 Draw

Ging schneller und einfacher als erwartet, erst nachdem noch Diverse zusätzliche Kundenwünsche berücksichtigt wurden, wurde der geplante Aufwand fast noch erreicht.

10.4.4 Record

In dieser Iteration kam ich auch schneller Vorwärts als erwartet. Am Ende jedoch kam es zu einem Problem. Das Datenfile, das von der Software erstellt wurde, konnte vom FIT-Manager nicht gelesen werden, obwohl ich beim Dateiformat die Anweisungen des Kunden befolgt habe. Der Versuch dieses Problem zu lösen kostete recht viel Zeit. Weil kein Lösungsweg für mich und den Kunden in Sicht war, haben wir entschieden, das Dateiformat so zu belassen wie es ist.

10.4.5 Extend

Diese Iteration brauchte klar länger als geplant. Dies hatte damit zu tun, dass der Aufwand sehr schwer zu schätzen war, da die Glättung ein sehr komplexes Problem darstellt. Auch wusste man nicht, ob und wie einem das Betriebssystem oder die Programmiersprache dabei unterschützt. Es wurde eine gewisse Glättung erreicht, die jedoch noch nicht alle Bedürfnisse des Kunden befriedigte. Da diese Erweiterung dem Kunden sehr wichtig war, kamen wir zu der Übereinkunft die Iteration „Sign“ zu streichen und die gewonnene Zeit für diese Iteration bereit zu stellen. Es wurde eine zusätzliche Funktion „Shape“ mit der Figuren (Rechteck, Polygon, Kreis, Text) gezeichnet werden konnten. Damit waren die die Bedürfnisse des Kunden ans zeichnen, die nicht von der Glättungsfunktion erledigt werden konnten, abgedeckt. Ausserdem wurde noch eine „Undo“ Funktion auf Wunsch des Auftraggebers eingebaut.

²⁸ <http://www.eclipse.org/dsdp/mtj/>

²⁹ <http://java.sun.com/javame/downloads/sdk30.jsp>

³⁰ http://msdn40.e-academy.com/elms/Storefront/Home.aspx?campus=hhft_info

10.4.6 Sign

Wurde durch Übereinkunft mit dem Kunden gestrichen, um mehr Zeit für die Iteration „Extend“ zur Verfügung zu haben.

10.4.7 Transition

Brauchte klar mehr Zeit, da während des Projektes ein bisschen knapp dokumentiert wurde. Der Dokumentationsaufwand war sowieso grösser als geplant. Da dies der Fall sein könnte, habe ich schon von Anfang an her vermutet und mich darauf vorbereitet, indem ich die letzten beiden Wochen dafür reserviert hatte und dadurch den dazu nötigen Zusatzaufwand einfach erbringen konnte.

10.4.8 Hilfspakete

Der Aufwand für Sitzungen war klar höher als eingeplant, weil ich nicht an die Vorbereitung und an das nachträgliche Sitzungsprotokoll gedacht hatte. Auch fürs Projektmanagement war der Zeitaufwand für eine Person grösser, als ich mir vorgestellt hatte. Ich musste die meisten Tätigkeiten, die in einem Team durchgeführt werden auch tun. Der Research Aufwand war auch eher schwer einzuschätzen und kam dadurch zustande, dass ich beim Studieren der Java Technologien für PDA's oft auf veraltete Informationen stiess, oder ich oft Informationen zu Produkten fand, die man erwerben musste und ich nach etwas anderem Ausschau halten wollte. Auch musste ich mich nach Themen erkundigen, an die ich am Anfang des Projektes nicht gedacht habe.

10.4.9 Fazit

Im Grossen und Ganzen ist der Auftraggeber und ich mit dem erreichten Ergebnis sehr zufrieden. Die Software ist nun auf einem modernen PDA lauffähig, was bei Kunden des Auftraggebers nicht mehr den Eindruck erwecken wird, dass das System veraltet sei, da es auf alten Geräten läuft. Weiterhin wird der Benutzer beim Zeichnen durch viele neue Funktionen, wie das Zeichnen von Formen, durch Glättung und die Möglichkeit Farben zu verwenden, erheblich unterstützt.

11. Users Manual

11.1 Introduction

This Manual describes the use of the FIT-System running on Windows Mobile 6.0 PDA Handhelds.

11.2 Requirements

The FIT-System is designed for the iPAC-214, but should be runnable on all Windows Mobile 6.0 Devices. It may work with older operating systems, if the .NET Compact Framework 2.0 is installed.

11.3 Start Applikation

If you have successfully installed your FIT-Application, you can start it, by tapping on Start->Programs->FIT-System. Now the “Main Form” of the Application will be shown.

11.4 Main Form

The main form contains 3 buttons. In the following sections, you find a description on what happens, when you choose one of the buttons.



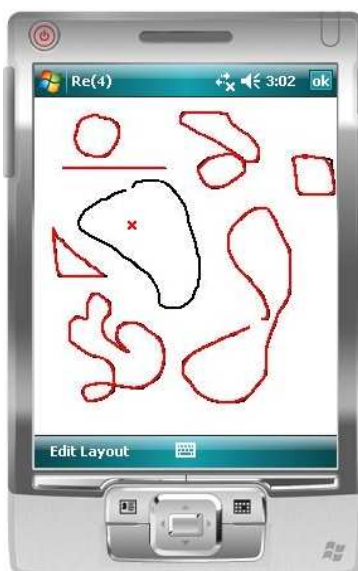
11.5 Record

Select the „Record“-Button in the main form, to record data. You are then asked to enter a data file name. The name has to be unique, otherwise the system won't accept the name. After you have entered the name press the “Ok” button.



To record data, you can either choose to use a “Layout” created with your PDA or you can use the technique of the old system, by choosing to use a “Template”.

If you choose to use a “Layout”, you are prompted with a list of all layout files currently stored in the application folder. Select the “Layout” you want to work with.



To record a data point tab on a Symbol of the “Layout”. The last recorded data point will be marked (Red Cross), so you know that it got recorded. While recording, you have the possibility to edit the “Layout”, through pressing on “Edit Layout”. If you use a “Template” this option is not shown. To finish the recording press “ok” and select “Yes” on the following “Alert”.

11.6 Draw

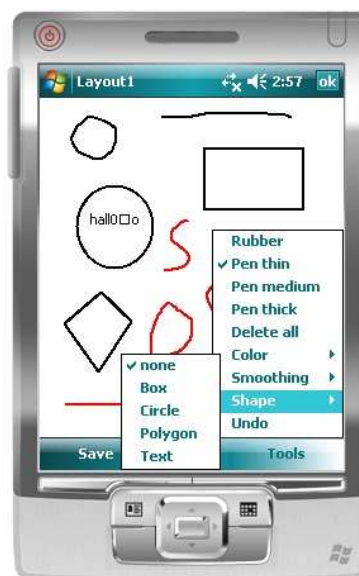
Select the “Draw“-button in the main form, to draw a „Layout“. You can choose now to either create a new „Layout“, or to edit an existing one.



If you chose to create a new “Layout”, you have to enter a layout name. The name has to be unique. When you have entered the name, press the “OK” button.



To draw the “Layout” you can choose the desired pen thickness and between the colors (Black, Blue, Green, Red, Yellow). You can erase part of the drawing with the “Rubber”, undo the last drawing step with the “Undo” option, or delete everything with the “Delete all” option. With the “Shape” function you can draw a “Box”, “Circle” (Ellipse), a Polygon, or Text. You can activate or disable the “Bezier” “Smoothing” function. To draw a box chose the “Box” option and tap on the screen. While you move the pen a preview, of how the box would look like is shown. When you’re content with the look of your box, remove the pen from the screen and the box is drawn. Drawing a circle (ellipse) works the same way. To draw a polygon, choose the “Polygon” option. Then tap on the screen for each point of the polygon and a “double tap” for the last point. (They need to be close together in distance, what may require some practice). With the “Save” option you can save the drawing, then a message with the location of the stored image is shown. To leave the “Drawing mode” press “ok” in the top right corner. If you haven’t saved the resent changes of the drawing, you will be asked, if you want to save them.



11.7 Delete

Select the „Delete“-Button in the main form, to delete a file. You have now the option to of deleting a “Layout” or a “Data” file.



If you chose to delete a “Data” file you see the “Form” on the left. You can select a file and press on “Delete”. If you do this, you will be prompted with an “Alert” to confirm the deletion of the file.



If you chose to delete a “Layout” file, you see the “Form” on your right. To delete the file select it in the list and press on the “Delete” option. You will be prompted with an “Alert” to confirm the deletion of the file. To get back to the former “Form” use the “Cancel” button.



12. Installationsanleitung

12.1 Einrichten Entwicklungsumgebung

1. Installation Visual Studio 2008 (mit C# Bibliotheken)
2. Download und Installation der Windows Mobile 6 Professional SDK³¹
3. Öffnen der Solution:
`\fit\FitDrawCode\FitDraw.NET_CF_2.0\FitDraw.NET_CF_2.0.sln` mit Visual Studio 2008.

12.2 Installation des FIT-Systems auf PDA

1. Öffnen der Solution:
`\fit\FitDrawCode\FitDraw.NET_CF_2.0\FitDraw.NET_CF_2.0.sln` mit Visual Studio 2008.
2. Klasse „Program“ im „FitDraw.NET_CF_2.0“ Projekt öffnen.
3. Variable „runOnEmulator“ auf „false“ setzen, falls nicht der Fall.
4. Bei Konstante „DEVICE_SERIAL_NUMBER“ die Seriennummer des PDA, auf dem die Software installiert werden soll, eintragen.
5. Build auf das „FitDraw.NET_CF_2.0“ Projekt ausführen
6. Build auf das „FitDraw.NET_CF_2.0Cab“ Projekt ausführen
7. Datei „FitDraw.NET_CF_2.0Cab.CAB“ im Verzeichnis
`fit\FitDrawCode\FitDraw.NET_CF_2.0\FitDraw.NET_CF_2.0Cab` auf den PDA kopieren.
8. FitDraw.NET_CF_2.0Cab.CAB auf dem PDA ausführen und Installationsanweisungen befolgen.
9. Programm unter „Program“ „FIT“ starten

12.2.1 Erstellen eines CAB Projektes

<http://msdn.microsoft.com/de-de/library/1b8fd8ss.aspx>

³¹ <http://www.microsoft.com/downloads/details.aspx?familyid=06111a3a-a651-4745-88ef-3d48091a390b&displaylang=en>

13. Persönliche Berichte

13.1 Projektarbeit

Am Anfang dieser Arbeit hatte ich einige Bedenken. Ich wollte diese Arbeit als Teamarbeit durchführen und dieses Thema war der Favorit meines vorgesehenen Partners. Mir gefiel die Arbeit auch recht gut, jedoch hatte ich noch nie für PDAs Software entwickelt und stellte es mir als problematisch und mühsam vor. Mein vorhergesehener Arbeitspartner bestand jedoch die Semesterarbeit nicht, womit er nicht für die Bachelorarbeit zugelassen wurde. Die anfänglichen Bedenken legten sich jedoch mit der Zeit und mir hat das Projekt sehr gut gefallen. Auch das Resultat das in der Arbeit erreicht wurde gefällt mir sehr gut. Leider konnte die „Signier“ Funktion nicht mehr umgesetzt werden.

13.2 Gelerntes

In diesem Projekt habe ich gelernt, wie man mit der Programmierung von „Handheld“ Computern umgeht. Auch habe ich die Programmiersprache C# .net besser kennen gelernt und festgestellt, dass ich es wirklich mag damit zu programmieren. Ich habe auch viel Erfahrungen gesammelt, wie man grössere Projekte durchführt und darüber berichtet. Da ich die Arbeit als Einzelarbeit durchführen musste, eröffnete dies mir die Möglichkeit in allen Bereichen des Projektes Erfahrungen zu sammeln. Dies hatte zur Folge, dass ich Tätigkeiten besser kennen lernte, die mir unangenehm waren und ich in früheren Projekten häufig auf andre Teammitglieder abschob.

13.3 Negative Erfahrungen

Eine negative Erfahrung war der Anfang des Projektes um die Bewerbung für die Arbeit. Mein Teamkollege wurde kurz vor der Arbeitsvergabe, nicht für die Bachelorarbeit zugelassen. Aus diesem Grund musste ich nun plötzlich die Arbeit als Einzelarbeit durchführen. Dies hatte zur Folge, dass ich mich zusätzlich unter Druck fühlte. Eine schlechte Erfahrung während des Projektverlaufs war die fehlende Dokumentation des Dateiformates (.fit) für Datenfiles. Dies verhinderte, dass die Datenfiles so gespeichert werden konnten, wie in der alten Applikation. Darum kann der FIT-Manager das Datenfile nicht korrekt lesen und muss angepasst werden.

13.4 Positive Erfahrungen

Ich habe festgestellt, dass es mir sehr gefällt mit C# .net zu programmieren. Auch haben mir die Arbeit und ihre Herausforderungen sehr gut gefallen. Die Art wie ich von Herrn Glatz betreut wurde und wie die Zusammenarbeit mit dem Auftraggeber verlief, fand ich hervorragend.

14. Anhang

14.1 Sitzungsprotokolle

Sitzungsprotokoll vom 25.02.2010

Datum: 25.02.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 11.00 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

Administrative Beschlüsse

- Nächste Sitzung 5.03.2010 mit Herrn Held
- Standardsitzungstermin ist am Donnerstag um 10 Uhr

Bis zu nächsten Sitzung soll folgendes erledigt werden

- Anschauen der erhaltenen CD's
 - Draw FIT System Version40 (oberflächlich)
 - Computer Game on .NET Compact Framework (oberflächlich)
 - PIM Suite in J2ME (oberflächlich)
- Installation Visual Studio
- Einrichten SVN
- Windows CE Emulator herunterladen und installieren (falls auffindbar und möglich)
- Hello World Beispielapplikation in J2ME und .NET Compact
- Weitere notwendige Programminstallationen
- Funktionalität der Programmiersprachen überprüfen
- (falls genügend Zeit) Programm das Grafikfunktionalität und Geschwindigkeit von J2ME und .NET Compact testet.

Sitzungsprotokoll

Datum: 05.03.2010
Ort: HSR Rapperswil
Zeit: 17.00 – 18.20 Uhr

Teilnehmer: Jürgen Held
Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Es gibt eine Zwischenpräsentation
 - Der Glättungsalgorithmus beim Zeichnen soll gut Dokumentiert werden
 - C# .net wird als Programmiersprache verwendet.
 - Zusätzliche Anforderungen geordnet nach Priorität
 - Grundfunktionalität der alten Applikation beibehalten.
 - Applikation soll auf Windows Mobile 6 laufen
 - Glättungsfunktion beim Zeichnen
 - Möglichst grosser Teil des Displays muss fürs Zeichnen verwendet werden können.
 - Farben sollen verwendet werden können
 - Gezeichnete Layouts auf Desktop Seite anzeigen
 - Signier Funktion von Desktop zu PDA Seite Verschieben

- Zu Erledigen
 - Grob Zeitplan
 - Anforderungsspezifikation
 - “Hello World” Applikation auf IPAC ausführen

Sitzungsprotokoll

Datum: 11.03.2010
Ort: HSR Rapperswil
Zeit: 10.00 – 10.45 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Meilenstein sagt aus wann welche Anforderungen erfüllt sind
 - Sitzungen mit Herrn Held nach bedarf
 - Zentraldokument Word
 - Aufgabenstellung mit Umfeld verbinden
 - Zusätzlich Anforderungen selber erkennen und vorschlagen.

- Zu Erledigen
 - Grafikfunktionen von C# ausprobieren
 - Anforderungsspezifikation verbessern ergänzen
 - “Hello World” Applikation auf IPAC ausführen
 - Arbeitstakte erweitern
 - Meilenstein M1 abschliessen
 - C Code der Zeichnungsfunktionen der Fit Draw Applikation analysieren
 - Design der Draw Funktionalität erstellen.

Sitzungsprotokoll

Datum: 18.03.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 10.45 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Nur Projektspezifische Risiken
 - Gerätspezifisch
 - Anforderungsspezifisch
 - Überprüfen ob auf Zeitpunkt der Mausereignisse Zugriffen werden kann
 - Abgerundetes Rechteck statt Rechteck zum Zeichnen verwenden
 - Lineare Interpolation (einfacher Glättungsalgorithmus)
 - Literaturstudie Kurvenglättung soll durchgeführt werden
 - Signier Funktion hat geringste Priorität

- Zu Erledigen
 - Design der Draw Funktionalität erstellen.
 - Batteriezustand anzeigen
 - Anzeige Zeichnungsamen
 - Erstellen einer neuen Zeichnung
 - Stif Funktion Dünn (oder Mittel)
 - Risiko Liste

Sitzungsprotokoll

Datum: 25.03.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 10.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Sitzung mit Herrn Held vereinbaren
 - Ohne Herrn Glatz
 - Herrn Glatz über Sitzung mit Herrn Held informieren
 - Protokoll
 - Besprechen an nächster Sitzung mit Herrn Glatz
 - Nächste Sitzung (mit Herrn Glatz) am 1 April um 10 Uhr.
- Zu Erledigen
 - Sitzung mit Herrn Held vereinbaren
 - Was bis jetzt erreicht
 - Was aussteht
 - Was anders ist als beim Palm
 - Glättung Newton
 - Lade Funktion
 - Dokumentation nachführen
 - Analyse Record Implementation
 - (gegebenfalls refactoring Zeichnungsfunktion)
 - Anfangen mit der Record (Datenpunkt erfassen) Implementation

Sitzungsprotokoll

Datum: 29.03.2010
Ort: Technopark Zurück
Zeit: 15.00 – 16.30 Uhr

Teilnehmer: Jürgen Held
Michel Rothmund

- Beschlüsse / Besprochenes
 - Falls Möglich Buttons abrunden.
 - Cancel Button zusätzlich zum Kreuz Button in die meisten Fenster hinzufügen
 - Bestätigung Dialog bevor der Delete All Funktion
 - „Wollen sie Speichern“ Dialog bevor Zeichnungsmodus durch betätigen des Kreuz Buttons (x) verlassen wird.
 - Ob der Save Button links oder rechts sein soll wurde noch nicht festgelegt.
 - Pen soll man so lassen wie er im Moment ist (kann aber noch ändern)
 - Alle Tasten ausser „Swicht“ Off sollen disabled werden
 - Beim Enter „Layout name“ Fenster
 - Muss es einen Cancel Button haben
 - name soll kleingeschrieben sein
 - Die Tatstatur soll eingeblendet werden.
 - Zeichnungsarea wenn möglich vergrössern
 - Einblendbarer Batteriestatus genügt
 - Name der Zeichnung soll im Zeichnungsfernster dort stehen wo jetzt der Applikationsname ist.
 - Pen Grösse ComboBox soll ins Menu hinunter falls damit Platz eingespart werden kann.
 - Angabe der möglichst grossen Zeichnungsarea in (x : y Pixelformat) an Herrn Held senden.
 - Bei Fehler Dialogen soll es Alert statt Error heissen.
 - Herr Held sieht nach ob er noch Information zum fit Dateiformat findet.
 - Herr hat mir auch noch den Apple Newton mitgegeben (testen Glättung)

Sitzungsprotokoll

Datum: 01.04.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 10.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Zwischenpräsentation findet im April statt.
 - Länge zwischen 15-20 Min danach 10-15 Min für Fragen
 - Aufbau
 - Problemstellung
 - Was Erreicht
 - Was Ausstehend
 - Was kann wahrscheinlich nicht erreicht werden
- Zu Erledigen
 - Implementation Wünsche von Herrn Held
 - Übrige Hardware Buttons (wird nach hinten geschoben)
 - Zeichnungsarea versuchen zu vergrössern
 - Zwischenpräsentation vorbereiten
 - Dokumentation nachführen
 - Analyse Record Implementation

Anfangen mit der Record (Datenpunkt erfassen) Implementation

Sitzungsprotokoll

Datum: 22.04.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 10.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Buttons Bitmap in Zeichnungsprogramm (Paint) erstellen und in überschriebener (Paint) Methode des Buttons zeichnen
 - Ausschalten der anderen Hardware Buttons nach hinten geschoben
 - Nächste Sitzung am Freitag 30.04.2010 um 12 Uhr

- Zu Erledigen
 - Problem mit der zu kleinen Zeichnungsarea aus der Vorführung lösen
 - Implementation Erfassen
 - Abklärung fit Dateiformat
 - Speichern eines Datenfiles im fit Format (Notfalls txt)
 - Analyse der Delete Implementation (Code anschauen)
 - Implementation Delete
 - Löschen eines Zeichnungsfiles
 - Löschen eines Datenfiles
 - Bestätigungsdiallog anzeigen (vor löschen)

Sitzungsprotokoll

Datum: 30.04.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 10.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Problem Dateiformat
 - Könnte Big Endian Little Endian von (Palm, Windows) sein
 - Nächste Sitzung am Donnerstag 06.05.2010 um 16.00 Uhr

- Zu Erledigen
 - Implementation Farben
 - Studie Glättung
 - Erste Glättungsversuche
 - Treffen mit Herrn Held vereinbaren
 - Verbesserungswünsche abklären
 - Fragen Dateiformat
 - Dateiformat
 - Big Endian + Little Endian testen

Sitzungsprotokoll

Datum: 06.05.2010

Ort: HSR Rapperswil

Zeit: 10.00 – 10.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Glättung
 - Lineare, Quadratische Interpolation
 - Bezier Kurven (2 Ableitung)
 - Teilglättungen (Rechteck, Linie)
 - Nächste Sitzung am Donnerstag 13.05.2010 um 10.00 Uhr
 - Meeting mit Herrn Held am 7.05.2010
 - Dateiformat
 - Binärdateien, Big-Little Endian

- Zu Erledigen
 - Weitere Glättungsstudien
 - Erste Glättungsversuche
 - Vereinfachte Bezier Version implementieren (keine 2 Ableitung)
 - Treffen mit Herrn Held
 - Verbesserungswünsche durchführen
 - Fragen Dateiformat lösen
 - Protokoll erstellen

Sitzungsprotokoll

Datum: 07.05.2010
Ort: Technopark Zurück
Zeit: 14.00 – 15.30 Uhr

Teilnehmer: Jürgen Held
Michel Rothmund

- Beschlüsse / Besprochenes
 - Verlorener IPAC
 - 1. Suchen Zuhause
 - 2 Mail an alle in der Schule mit iPac Seriennummer
 - 3. Hat die Schule eine (Diebstahl-)Versicherung
 - 4. Herr Held überlegt sich was nun geschieht.
 - Dateiformat
 - Dateien werden weiterhin im Klartext abgespeichert. FIT-Manager wird später angepasst.
 - Endung soll .fitdb sein falls möglich. Sonst Absprache mit Herrn Held
 - Record und Delete Funktionen sind so in Ordnung wie sie sind.
 - Koordinaten und Grösse der Buttons sollen bei allen Forms gleich sein
 - Farben
 - Farbwechsel soll bei wechsel der Stiftgrösse Beibehalten werden.
 - Erster Buchstabe der Farbe im Menü in der jeweiligen Farbe.
 - Glättung
 - Allgemeine Methode verbessern Bezier
 - Einzelne Optionen wie:
 - Circle
 - Polygon
 - Box
 - (vielleicht noch weitere wie z.B. Text)
 - Signieren wird weggelassen (Zeit für Glättung benutzen)
 - Frage warum kein Beliebige Zeichnungsprogramm verwenden
 - Nachschauen was für Zeichnungsprogramme es für den iPAC gibt
 - Feste Pixelgrösse
 - Während der Aufzeichnung Layout verändern

Sitzungsprotokoll

Datum: 12.05.2010

Ort: HSR Rapperswil

Zeit: 13.00 – 13.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Glättung
 - Effekt der Glättung muss vergleichbar sein
 - Falls Gerade dies als solche darstellen.
 - Nächste Sitzung am Donnerstag 20.05.2010 um 16.00 Uhr
 - Ergebnisse der Sitzung mit Herrn Held
 - Anzeige erstellen um über Versicherung Ersatz für iPac erhalten.

- Zu Erledigen
 - Mit und Ohne Glättung nebeneinander Darstellen
 - Zuerst normal Zeichnen dann Geglättete Figur darstellen
 - Zeichnen von Formen über Menü
 - Circle
 - Polygon (kein Doppelklick sonder Zeit(5 Sekunden) Trigger
 - Box

Sitzungsprotokoll

Datum: 20.05.2010

Ort: HSR Rapperswil

Zeit: 16.00 – 16.30 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Glättung
 - Rechteck, Ellipse, Polygon mit Vorschau darstellen
 - Falls Gerade dies als solche darstellen.
 - Nächste Sitzung am Donnerstag 3.06.2010 um 11.00 Uhr
 - Sitzung mit Herrn Held vereinbaren (durchführen)
 - Abklären, Verschiedene Textgrößen, Ellipse
 - Tool für Speicheranalysen für Windows Mobile suchen

- Zu Erledigen
 - Abstract
 - Nachschauen welche Zeichnungsprogramme es schon für Windows Mobile gibt
 - Glättung
 - Bezier Methode Verbessern
 - Gerade Linie erkennen
 - Shapes
 - Polygon, Box, Circle, mit Vorschau

Sitzungsprotokoll

Datum: 31.05.2010
Ort: Technopark Zurück
Zeit: 8.30 – 9.30 Uhr

Teilnehmer: Jürgen Held
Michel Rothmund

- Beschlüsse / Besprochenes
 - Verlorener IPAC
 - Nicht über Versicherung meines Vaters
 - Herr Held hat neuen iPAC bestellt der in ca. 2 Tagen bei Michel eintreffen wird
 - Mail an Herr Held bei Eintreffen der iPAC Lieferung (mit Angabe über Betriebssystemsprache)
 - Dateiformat (fitdb)
 - Muss in Doku beschrieben werden um Ähnliche Probleme in Zukunft zu vermeiden.
 - Shapes mit Vorschau
 - Sind mehrheitlich gut so wie sie sind
 - Korrektur bei Polygon falls fast auf gleicher horiz./vertikaler Ebene
 - (Diese Korrektur soll beim normalen zeichnen nicht durchgeführt werden)
 - Bei Text wäre es gut wenn man den Text sieht den man schreibt. (Priorität niedrig)
 - Glättung
 - Bezier wird so beibehalten wie er ist, mit der Ausnahme, dass die ungeglättete Variante nach heben des Stiftes nicht durch die geglättete ersetzt wird.
 - Farben
 - Im Menü werden die Farben im Klartext beibehalten und nicht in der jeweiligen Farbe gefärbt.
 - Rubber: Fehler behoben wenn Shape ausgewählt (höchste Priorität)
 - Undo Funktion für die letzte Zeichnungsaktion (mittlere Priorität)
 - Kopierschutz mit Seriennummer falls möglich (hohe Priorität)
 - 21 oder 25 Juni Treffen mit Herrn Held für Rückgabe Palm, Norton etc.

Sitzungsprotokoll

Datum: 02.06.2010
Ort: HSR Rapperswil
Zeit: 11.00 – 11.45 Uhr

Teilnehmer: Eduard Glatz
Michel Rothmund

- Beschlüsse / Besprochenes
 - Dokumentation
 - Doxygen für Code
 - Inhaltsverzeichnis erstellen und Herrn Glatz schicken
 - Entwurf, Implementationsbeschreibung (Klassen-, Sequenzdiagramme)

- Zu Erledigen
 - Kopierschutz (hohe Pirotität)
 - Glättung nicht mehr nebeneinander
 - Anzeigen des geschriebenen Textes bei Text
 - Dokumentation
 - Inhaltsverzeichnis
 - Problembereicht
 - Beschreibung der Implementation

14.2 Meilensteinrapporte

Meilensteinrapport M1		Nr. 01
Meilenstein	M1 (Iteration Inception Abschluss)	
Datum:	12. März 2010	
Inhalt:		
<ul style="list-style-type: none"> - Ausgeführte Arbeiten - Pendente Arbeiten - Zeitaufwand / Status - Weiteres Vorgehen 		
Ausgeführte Arbeiten		
<ul style="list-style-type: none"> - Organisation & Planung - Einarbeitung in Technologien - Anforderungsspezifikationen - Analyse Source Code - Researches 		
Pendente Arbeiten		
<ul style="list-style-type: none"> - Keine 		
Zeitaufwand / Status		
<ul style="list-style-type: none"> - Der Meilenstein wurde vollständig erreicht. Alle zu erledigende Arbeiten sind abgeschlossen. Der Zeitplan wurde somit bisher eingehalten - Der effektive Zeitaufwand liegt ganz knapp über der geplanten Zeit. 		
Weiteres Vorgehen		
<ul style="list-style-type: none"> - Beginnen mit der Iteration 2 (Iteration Draw) 		

Meilensteinreport M2		Nr. 02
Meilenstein	M2 (Iteration Draw Abschluss)	
Datum:	1. April 2010	
Inhalt:		
	<ul style="list-style-type: none">- Ausgeführte Arbeiten- Pendente Arbeiten- Zeitaufwand / Status- Weiteres Vorgehen	
Ausgeführte Arbeiten		
	<ul style="list-style-type: none">- Festlegen der Systemarchitektur- Dateien - Diagramm für Draw Funktion- Sequenzdiagramm für Draw Funktion- Draw Implementation der Funktionen FZ1 – FZ9- Draw Integrationstests der Funktionen FZ1– FZ9	
Pendente Arbeiten		
	<ul style="list-style-type: none">- Keine	
Zeitaufwand / Status		
	<ul style="list-style-type: none">- Der Meilenstein wurde ein bisschen früher erreicht als geplant mit ein bisschen weniger Aufwand als eingeplant.	
Weiteres Vorgehen		
	<ul style="list-style-type: none">- Beginnen mit der Iteration 3 (Record)	

Meilensteinreport M3

Nr. 03

Meilenstein M3 (Iteration Record Abschluss)

Datum: 25. April 2010

Inhalt:

- Ausgeführte Arbeiten
- Pendente Arbeiten
- Zeitaufwand / Status
- Weiteres Vorgehen

Ausgeführte Arbeiten

- Dateien – Diagramm mit der Record Funktionen erweitern
- Sequenzdiagramm für Record Funktionen
- Record Implementation von FE1 bis FE6
- Record Integrationstests von FE1 bis FE6

Pendente Arbeiten

- Dateiformat nicht genau wie in der Vorgängerapplikation

Zeitaufwand / Status

- Der Meilenstein wurde erreicht. Alle zu erledigende Arbeiten abgeschlossen sind, jedoch kann der FIT-Manager die Datei nicht lesen. Im Moment wird der Zeitplan noch eingehalten, jedoch kann das aufgetretene Problem zu Verzögerungen führen

Weiteres Vorgehen

- Wie geplant: Mit beginnen der Iteration Delete
- Treffen mit Auftraggeber bezüglich des Dateiformates

Meilensteinreport M4		Nr. 04
Meilenstein	M4 (Iteration Delete Abschluss)	
Datum:	1. Mai 2010	
Inhalt:		
<ul style="list-style-type: none">- Ausgeführte Arbeiten- Pendente Arbeiten- Zeitaufwand / Status- Weiteres Vorgehen		
Ausgeführte Arbeiten		
<ul style="list-style-type: none">- Design des Delete Modus- Implementieren Delete Modus- Integrationstests des Delete Modus		
Pendente Arbeiten		
<ul style="list-style-type: none">- Dateiformat kann immer noch nicht von FIT-Manager gelesen werden- Treffen mit Auftraggeber noch ausstehend		
Zeitaufwand / Status		
<ul style="list-style-type: none">- Der Meilenstein wurde sicher erreicht. Jedoch konnte das Dateiformatproblem immer noch nicht gelöst werden und der Auftraggeber hatte noch keine Zeit.- Die bisherige Investierte Zeit, stimmt sehr gut mit der des Projektplanes überein		
Weiteres Vorgehen		
<ul style="list-style-type: none">- Treffen mit Auftraggeber- Weiter mit Iteration Extend		

Meilensteinreport M5

Nr. 05

Meilenstein M5 (Iteration Palm OS Abschluss)

Datum: 3. Juni 2010

Inhalt:

- Ausgeführte Arbeiten
- Pendente Arbeiten
- Zeitaufwand / Status
- Weiteres Vorgehen

Ausgeführte Arbeiten

- Zeichnungsfunktionen FZ11- FZ17
- Design
- Testbericht erstellt

Pendente Arbeiten

- Die gesamte Signier Funktion
- Neues Dateiformat kann von FIT-Manager nicht gelesen werden.

Zeitaufwand / Status

- Das Dateiformat konnte nicht so erstellt werden wie in der Vorgängerapplikation. Der FIT-Manager muss angepasst werden.
- Die Signier Funktion wird definitiv ausgelassen, da der Auftraggeber wollte dass mehr Zeit in die Funktion „Glättung“ investiert wird um diese zu verbessern
- Der effektive Zeitaufwand stimmt immer noch gut mit dem geplanten Aufwand überein. Jedoch wurde der ganze Aufwand der für die „Signier“ Funktion geplant war für die „Glättung“ Funktion eingesetzt.

Weiteres Vorgehen

- Nach Zeitplan mit der Abschluss Iteration „Transition“

Meilensteinreport M6

Nr. 06

Meilenstein M6 (Iteration Transition Abschluss)

Datum: 16. Juni 2010

Inhalt:

- Ausgeführte Arbeiten
- Pendente Arbeiten
- Zeitaufwand / Status
- Abstract
- Technischer Bericht

Ausgeführte Arbeiten

- Dokumente vervollständigt
- Review der Dokumente
- Systemtests
- Fehlerbehebungen
- Persönliche Berichte
- Projektmanagementplan fertig

Pendente Arbeiten

- keine

Zeitaufwand / Status

- Der Meilenstein wurde etwas erreicht, jedoch entstand ein Mehraufwand so dass der geplante Aufwand um einige Stunden überschritten wurde.
- Status: Projekt vollständig abgeschlossen

15. Glossar

.bmp

(Windows Bitmap) Zweidimensionales Rastergrafikformat. Dateiformat für Bilder.

.NET

Bezeichnet eine von Microsoft entwickelte Software-Plattform, bestehend aus Laufzeitumgebung, Klassenbibliotheken, Programmierschnittstellen und Dienstprogrammen.

.NET CF 2.0

(.NET Compact Framework 2.0) Teil des .NET Frameworks der für die Nutzung auf mobilen Geräten geeignet ist.

.jpg

Dateiformat für komprimierte Bilder.

.pdb

(Palm Database) Dateiformat für Palm Betriebssysteme.

Alert

Ein Alert ist eine Warnung oder Informationsdialog welcher durch den Benutzer zum verschwinden gebracht werden muss.

Bezierkurven

Pragmatisch modellierte Kurve in der numerischen Mathematik³².

C

Programmiersprache „C“. Wurde für die Entwicklung der alten Applikation verwendet

C#

Programmiersprache von Microsoft im Rahmen der .NET Strategie entwickelt.

Datafile

Das Datafile ist ein Textfile, welches während dem Record –Modus alle Daten der angetippten Punkte aufnimmt.

Emulator

Im Emulator kann man eine entwickelte Software testen. Es besteht die Möglichkeit, eine Hardware und- oder deren Betriebssystem zu simulieren auf der die Software getestet werden soll. Der Vorteil des Emulators gegenüber dem Simulator ist, dass der Emulator im Debug-Modus betrieben werden kann.

³² <http://de.wikipedia.org/wiki/B%C3%A9zierkurve>

Event

Ein Event ist ein Ereignis, welches während dem Programm immer wieder auftritt. Es gibt sehr viele verschiedene Arten von Events.

FIT

(Flexible Interface Technik) Abkürzung der durch Jürgen Held patentierten Technologie.

FIT-Manager

Desktop PC Applikation mit der die „Signier“ Funktion bis anhin durchgeführt wird.

Form

Ein Applikations- Fenster, welches normalerweise den ganzen Screen bedeckt. Eine Form kann optional Buttons, Textfelder und Menus enthalten.

Garbage Collector

Verantwortlich für automatische Speicherbereinigung in der Softwareentwicklung. Soll den Speicheranspruch eines Prozesses möglichst minimieren.

GUI

(Graphical User Interface) Benutzeroberfläche einer Applikation mit der, der Benutzer interagiert.

Layout

Layout ist die Bezeichnung für eine Zeichnung, welche der Benutzer direkt auf dem PDA erstellt und abspeichert. Dieses Layout wird der Benutzer später laden, damit er Datenpunkte aufnehmen kann.

PDA

(Personal Digital Assistent) Kleiner tragbarer Computer

PictureBox

Form Element Klasse in C# die ein Bild enthalten kann.

Record

Ein Record ist eine Datenstruktur welche durch eine ID identifiziert wird. Typischerweise speichern Applikationen Daten in Record-Datenbanken.

Screen

Mit dem Begriff Screen wird die Bildschirmoberfläche des PDAs gemeint. Auf den Screen kann mit Zeichnungsfunktionen der Software gezeichnet werden.

Signieren

Erfasste Datenpunkte gruppieren und den Punktgruppen eine Bedeutung (Ereignis) zuweisen.

Template

Das Template ist eine Folie auf welche der Benutzer Symbole mit einem Stift Zeichnen kann. Dieses Template legt der Benutzer schliesslich auf den Palm und Datenpunkte zu erfassen.

16. Referenzen

- [1] DrawFit[1].pdf Bericht des Vorgängerprojektes von Matthias Thut und Normen Schreiber Frühjahr 2004
- [2] <http://code.google.com/p/pocketpicture/downloads/list> letzter Zugriff 15.6.10
Download PocketPicture Zeichnungsprogramm
- [3] <http://www.vetasoft.com/index.php?index=12&language=en> letzter Zugriff 15.6.2010. VSPainter Komerzielle Software mit Vorstellungsvideo
- [4] <http://vspainter-le.softonic.de/pocketpc> letzter Zugriff 15.6.10
Trial-Version von VSPainter mit Beschreibung
- [5] <Http://pdacraft.com/paint.php> letzter Zugriff 15.6.10
Beschreibung des Paint ähnlichen Programms PDACraft Paint für PDAs
- [6] AgileMethoden_Folien_2_proSeite2.pdf aus der Vorlesung Software Engineering 3 von Lothar Müller
- [7] <http://de.wikipedia.org/wiki/B%C3%A9zierkurve> letzter Zugriff 15.6.10
Erklärt Funktionsweise der Bezierkurven. (Speziell Kubische Bezierkurven)
- [8] http://www.therobotgeek.net/articles/CF_Bezier_Curves.aspx
Erläuterung der Bezierkurven und Beispielimplementation der im Standard .net enthalten „Bezier“ Zeichnungsfunktion für das .net CF Framework.
- [9] <http://www.eclipse.org/dsdp/mtj/> letzter Zugriff 14.6.10
Erläuterung und Download der Entwicklungsumgebung für mobile Geräte von IBM.
- [10] <http://java.sun.com/javame/downloads/sdk30.jsp> letzter Zugriff 14.6.10
Erläuterung und Download der Java ME SDK
- [11] <http://www.microsoft.com/downloads/details.aspx?familyid=06111a3a-a651-4745-88ef-3d48091a390b&displaylang=en>
Erklärung und Download des Windows Mobile 6 Toolkits