

SEPS Projektskizze

1 Inhalt einer Projektskizze

1.1 Zweck

Eine Projektskizze enthält in kompakter Form die wichtigsten Informationen, um über den Start eines Projekts zu entscheiden. Dazu gehören die folgenden Elemente:

- Beschreibung des Ziels. Das Hauptziel soll so detailliert beschrieben werden, dass jemand ohne spezifisches Wissen dieses verstehen kann.
- Der Weg zum Ziel. Dazu gehören die wichtigsten Ressourcen und eine zeitliche Grobplanung.
- Risiken auf dem Weg zum Ziel.
- Wirtschaftlichkeit. Wird das Projekt innerhalb einer gewinnorientierten Organisation durchgeführt, sind Überlegungen zur Wirtschaftlichkeit letztendlich massgebend für den Durchführungsentscheid.

1.2 Aufbau für ein Software Projekt

In den folgenden Kapiteln werden die einzelnen Komponenten einer Projektskizze für ein Software-Projekt beschrieben.

1.2.1 Idee

Die grundlegende Idee, die zu diesem Projekt führte, soll mit ein paar Sätzen beschrieben werden. Dadurch wird der Kontext für das Verständnis der weiteren Kapitel klarer.

1.2.2 Hauptanwendungsfall

Auf einer halben Seite wird der Hauptanwendungsfall der fertigen Software in normaler Prosa beschrieben. Dazu versetzt sich der Autor in die Rolle des neutralen Beobachters und beschreibt Schritt für Schritt, wie der/die Anwender und die fertige Software miteinander interagieren, um das vom Anwender gewünschte Resultat zu erreichen.

1.2.3 Weitere Anforderungen

Hier werden alle weiteren wichtigen Funktionen aufgelistet. Ebenfalls sollten alle nicht-funktionalen Anforderungen wie z.B. minimale Antwortzeiten oder spezielle Sicherheitsaspekte beschrieben werden. Zuletzt sollten auch ein paar weiterführende Ideen aufgeführt werden, in welche Richtung sich die Anwendung noch entwickeln kann.

1.2.4 Ressourcen

In diesem Kapitel werden die notwendigen Ressourcen grob beschrieben. Dazu gehören die Fähigkeiten der Projektmitarbeiter, eventuell mit bereits vorgesehenen oder möglichen Teammitgliedern. Aber auch Know-How, das sich nicht im Projektteam oder in der Organisation befindet, muss aufgelistet werden. Eine grobe Aufwandsschätzung für die Hauptfunktionalität darf natürlich auch nicht fehlen.

1.2.5 Risiken

Alle über das "übliche" Mass hinausgehenden Risiken müssen offen und klar genannt werden. Das können neue, unbekannte Technologien sein, besonders kritische Anforderungen oder einfach fehlendes Wissen.

1.2.6 Grobplanung

Die Aufwandsschätzung muss noch mit den zu erwartenden Ressourcen zu einer Grobplanung mit Meilensteinen, möglichen Arbeitspaketen und einem Endtermin verarbeitet werden. Dazu gehört eine erste Planung der einzelnen Phasen mit ihrer Dauer und der Anzahl Iterationen.

1.2.7 Kundennutzen

Mit diesem Kapitel wird die interne Sichtweise des Projekts erweitert, indem der Kundennutzen beschrieben wird. Wieso soll ein Kunde überhaupt unser Produkt später einmal kaufen? Kann er damit eine bisherige Tätigkeit schneller oder effizienter durchführen? Kann er damit neue Tätigkeit ausüben, die für ihn einen Nutzen bringen?

1.2.8 Wirtschaftlichkeit

Bei jeder Investition – und ein Software-Projekt ist eine Investition – stellt sich schliesslich die Frage nach der Rentabilität. Diese ist die zinsbereinigte Differenz zwischen den Aufwendungen und den Erträgen einer Investition. Bei einem Software-Produkt sind solche Überlegungen noch schwieriger anzustellen als bei anderen Wirtschaftszweigen, die sich auf eine langjährige Tradition und Erfahrung abstützen können. Dennoch sollten zumindest ein paar mögliche Szenarien durchgerechnet werden und mit existierenden Projekten/Produkten verglichen werden.

2 Beispiel: Projektskizze für eine elektronische Kasse

2.1 Idee

Die Funktionen einer konventionellen, rein mechanischen Kasse sollen ergänzt werden mit den Möglichkeiten der modernen Computer-Technologie. Dazu gehören das automatische Berechnen des Rückgabebetrags und eine vereinfachte Behandlung von Kreditkarten-Belastungen. Zugleich können die Daten aller Kassen sofort und automatisch verarbeitet werden und ein möglicher Fehlbetrag rasch entdeckt werden.

2.2 Hauptanwendungsfall

Der Hauptanwendungsfall ist der Kunde, der eine Menge von Artikeln kaufen will und diese bar bezahlt.

- Der Kunde legt alle seine Artikel dem Kassier vor.
- Der Kassier tippt der Reihe nach die Menge und den Preise jedes Artikels ins System ein
- Der Kassier schliesst die Eingabe ab und der Kunde sieht den Gesamtpreis.
- Der Kunde gibt dem Kassier einen Betrag, der mindestens so gross wie der Gesamtpreis ist.
- Der Kassier gibt den Kundenbetrag ins System ein und dieses berechnet den Rückgeldbetrag.
- Der Kassier gibt dem Kunden den vom System berechneten Rückgeldbetrag.

2.3 Weitere Anforderungen

Folgende Funktionen sollten in der Grundauführung ebenfalls vorhanden sein:

- Der Kunde soll mit einer EC- oder Kreditkarte bezahlen können. Dabei soll der Betrag automatisch validiert werden.
- Das System soll eine Schnittstelle zu einem zentralen Computer aufweisen, über die alle Einkäufe übermittelt werden. Diese können dort archiviert und ausgewertet werden.
- Folgende weiteren Anforderungen sind einzuhalten:
 - Die Antwortzeit des Systems soll kleiner als 0.5 Sekunden sein.
 - Die Verfügbarkeit des Systems muss > 99.9% sein
 - Sicherheit: In Bezug auf unbefugten Zugriff, Datenmanipulation und sichere Datenübertragung sind die branchenüblichen Sicherheitsstandards einzuhalten.

2.4 Ressourcen

Für die Erstellung der Software wird ein Kernteam von ca. 4 Personen benötigt. Alle müssen über praktische Erfahrungen mit OOP haben, wobei mindest eine Person als Systemarchitekt qualifiziert sein muss.

Für die EC-Karten-Validierung muss externes Wissen eingekauft werden.

Für die fachlichen Belange ist Wissen im Bereich Kassen-Funktionen, Buchhaltung und Statistik notwendig.

Der Gesamtaufwand für das Software Produkt wird auf ca. 36 "Mann-Monate" geschätzt.

2.5 Risiken

Eingebettete Echtzeitsoftware ist für sich betrachtet wegen den erhöhten Anforderungen und den Limiten der Hardware-Umgebung bereits ein erhöhtes Risiko. Dazu fehlt das Wissen im Bereich EC-Karten / Kreditkarten-Validierung.

2.6 Grobplanung

Die Gesamtdauer der Projekts wird auf 10 Monate geschätzt. Die Grobplanung sieht für die Grundfunktionalität (1. Release) eine Entwicklungszeit von 16 Wochen vor. Die Planung erfolgt gemäss UP mit einer Iterationsdauer von 2 Wochen:

Phase	Iteration	Ziele
Inception	1	Projektskizze erstellt, Architektur skizziert
Elaboration	2	50 % der Anwendungsfälle detailliert ausformuliert, Rest identifiziert und priorisiert, 1. Entwurf Domänenmodell
	3	80 % der Anwendungsfälle detailliert ausformuliert, Domänenmodell fertig und Architektur stabil
Construction	4	20 % der UCs realisiert und getestet, UI Prototyp
	5	40 % der UCs realisiert und getestet
	6	70 % der UCs realisiert und getestet
	7	100 % der UCs realisiert und getestet, UI fertig
Transition	8	Systemtest und Dokumentation

2.7 Kundennutzen

Wir sehen die folgenden Punkte als die wichtigsten Kundennutzen an :

- Das Rückgabegeld wird automatisch und damit fehlerfrei berechnet
- Die Validierung von Kreditkarten geschieht ebenfalls automatisch und ist damit nicht nur schneller, sondern auch sicherer.
- Die Tagesumsätze sind sofort berechnet und die Daten können automatisch in die Finanz- und Lagerbuchhaltung überführt werden.
- Die Lagerbuchhaltung ist automatisch auf dem aktuellsten Stand. Dadurch kann das Lager effizienter geführt werden.
- Es kann über Statistiken das Konsumverhalten besser erklärt und vorausgesagt werden.

2.8 Wirtschaftlichkeit

Der geschätzte Aufwand beträgt insgesamt 3 Mann-Jahre zu je 200'000 CHF. Dazu kommen die externen Leistungen von 200'000 CHF, was zu einem Budget von ca. 0.8 Mio. CHF führt. Bei einem Deckungsbeitrag von 200 CHF pro verkaufte Kasse müssen demnach 4000 Kassen verkauft werden, um die Auslagen der Grundentwicklung zu decken. Eine vorsichtige Abschätzung geht deshalb von 8000 Kassen aus, die verkauft werden müssen, bis diese Investition anfängt zu rentieren. Dieses Verkaufsvolumen sollte nach Meinung des Marketings innerhalb von 3 Jahren zu erreichen sein.