



M1 Auftrag Projektskizze

1 Kontext

Sie haben die Idee für ein Software-Produkt, das Sie entwickeln wollen. Um Ihre Idee umsetzen zu können, suchen Sie Team-Mitglieder und Investoren. Sie verfassen deshalb eine Projektskizze, in der die Team-Mitglieder den gemeinsamen Plan festhalten, so dass Sie die Investoren von Ihrer Projekt-Idee damit überzeugen können. Wenn Ihre Idee die Investoren überzeugt, erhalten Sie eine erste Finanzierung, um eine erste Anwendung zu entwickeln, die einerseits die Machbarkeit Ihrer Idee darlegt und andererseits als Basis für die Akquisition weiterer Investoren dienen soll.

2 Inhalt einer Projekt-Skizze

In PM2 haben Sie bereits gelernt, wie eine Projektskizze geschrieben wird. In PM3 wenden Sie Ihr Wissen an.

2.1 Zweck

Eine Projekt-Skizze enthält in kompakter Form die wichtigsten Informationen, um Investoren von Ihrer Projektidee so zu überzeugen, dass sie bereit sind, Geld in das Projekt und in Ihr Start-Up zu investieren. Dazu gehören typischerweise die folgenden Elemente:

- Idee/Ziel: Beschreibung des Projektziels, in unserem Fall ein Software-Produkt. Das Hauptziel soll so detailliert beschrieben werden, dass **jemand ohne spezifisches Wissen** dieses verstehen kann.
- Kundennutzen: Wer sind die Hauptbenutzer des Software-Produkts und was ist der Nutzen, den diese von Ihrem Softwareprodukt erhalten?
- Abgrenzung zu Bestehendem: Was existiert bereits und was ist neu/anders an Ihrer Idee?
- Der Weg zum Ziel: Dazu gehören eine Lösungsidee, die wichtigsten Ressourcen und eine zeitliche Grobplanung.
- Risiken auf dem Weg zum Ziel: Welche wichtigsten Risiken sehen Sie aus heutiger Sicht?

2.2 Aufbau für ein Software-Projekt

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

2.2.1 Ausgangslage

Die Ausgangslage beschreibt die heute Situation möglichst neutral in wenigen prägnanten Sätzen.

2.2.2 Idee

Die grundlegende Idee des Projekts soll mit ein paar Sätzen möglichst klar erläutert werden. Dies umfasst insbesondere die Problemstellung des Kunden, die mit dem Software-Produkt gelöst werden soll, sowie den Kern-Nutzen für den Kunden bezüglich dieses Problems.



2.2.3 Kundennutzen

In diesem Kapitel beschreiben Sie den Kundennutzen im Detail. Warum soll der Kunde das Produkt, das entwickelt werden soll, später einmal kaufen? Kann er damit eine bisherige Tätigkeit schneller oder effizienter durchführen? Kann er damit neue Tätigkeiten ausüben, die für ihn einen Nutzen bringen?

2.2.4 Stand der Technik / Konkurrenzanalyse

In einem kurzen Überblick soll die Konkurrenzsituation bezüglich der Problemstellung (vgl. Kapitel 2.2.1) dargelegt werden: Welche Lösungen gibt es bereits? Wie grenzt sich das geplante Software-Produkt von diesen ab? Die gemachten Aussagen sollen mit korrekt formulierten Literaturreferenzen (gemäss Zitierleitfaden) abgestützt werden.

2.2.5 Hauptablauf (Kontextszenario)

Auf einer halben Seite wird der Hauptanwendungsfall der fertigen Software in schriftsprachlicher 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. Diese Beschreibung sollte möglichst lösungsneutral beschrieben sein.

2.2.6 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.

2.2.7 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 das Know-how, welches sich nicht im Projektteam oder in der Organisation befindet, muss aufgelistet werden. Eine grobe Aufwandsschätzung für die Entwicklung des gesamten Softwareprodukts darf natürlich auch nicht fehlen.

2.2.8 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.

2.2.9 Grobplanung

Für die Erstellung einer ersten Version müssen die funktionalen Anforderungen mit Use Cases aufgelistet sowie eine Grobplanung mit den vorgegebenen Iterationen und Meilensteinen gemacht werden. Dazu gehört auch das Ziel jeder Iteration festzulegen.

3 Fachliche Bewertung (10%)

Alle Kapitel der Projektskizze werden fachlich separat gemäss den folgenden Kriterien bewertet:

- Vollständigkeit gemäss der obigen Beschreibung
- Verständlichkeit und Ausführlichkeit



- Angemessenheit gegenüber Ihrem Projekt

4 Sprachliche Anforderungen

Die Projektskizze muss in einer für die Adressaten verständlichen, präzisen, objektiven und neutralen Sprache (**keine Werbesprache!**) verfasst sein. Der Text soll gemäss obenstehenden Vorgaben strukturiert sein, versehen mit aussagekräftigen Zwischentiteln, in einem ansprechenden Layout, mit Titelblatt, Inhaltsverzeichnis und Quellenangaben. Die verschiedenen Textelemente (Aussagen, Abschnitte) sind kohärent formuliert und nehmen auf Grafiken oder Tabellen ausdrücklich Bezug (Querverweise). Fakten und Aussagen sind belegt (Referenzen, Quellen), die getroffenen Annahmen klar deklariert und die daraus gezogenen Schlüsse gut nachvollziehbar. Die sprachlichen Formulierungen müssen orthografisch, grammatikalisch und inhaltlich korrekt und im Stil an die Adressaten angepasst sein.

4.1 Sprachliche Bewertung (10%)

Konkret wird die Projektskizze sprachlich anhand der folgenden Kriterien bewertet:

- Der Aufbau entspricht der vorgegebenen Struktur.
- Die Inhalte sind relevant und mit qualitativ hochwertigen Quellen belegt.
- Der Stil ist wissenschaftlich, sachlich und einheitlich.
- Sätze und Absätze sind logisch verknüpft.
- Text-Bild-Bezüge sind explizit und explikativ.
- Der Text ist sprachformal korrekt verfasst (Grammatik, Orthografie, Semantik).
- Abbildungs- und Literaturreferenzen sind einheitlich und entsprechen den Vorgaben der SoE.

4.2 Notenrelevante Grundlagen (auf Moodle)

- Der SoE-Standard für formale Texte steht in der «Guideline formales und wissenschaftliches Schreiben».

4.3 Unterstützende Unterlagen (auf Moodle)

- Ordner «Bibliotheksrecherche»

5 Beispiel: Projektskizze für eine Selfscanning-App

5.1 Ausgangslage

In den meisten heutigen Geschäften der Schweiz müssen die Einkäufe nach wie vor an einer bedienten Kasse bezahlt werden, was immer wieder zu längeren Wartezeiten für die Kunden führt. Migros und Coop haben in den letzten Jahren begonnen, Self-Scanning-Systeme einzuführen [1,2], die sich zunehmender Beliebtheit erfreuen. Aber auch dort bilden sich bei den Zahlstationen in Stosszeiten bereits wieder Warteschlangen, da es nur wenige Zahlstationen gibt und die Kunden länger für den Bezahlvorgang benötigen als bei einer bedienten Kasse.

5.2 Idee

Es soll eine mobile App entwickelt werden, mit der die Kunden eines Shops die gekauften Waren einfach mit ihrem Smartphone einscannen können via den aufgedruckten Barcode auf den Produkten. Am Ende des Einkaufs bestätigt der Kunde die Liste seiner Einkäufe einfach auf dem Smartphone und löst die Bezahlung aus, bei der die hinterlegte EC- oder Kreditkarte entsprechend belastet wird.

5.3 Kundennutzen

Die folgenden Punkte können als den wichtigsten Kundennutzen angesehen werden:

- Die Kunden können ihre Einkäufe schneller erledigen, da sie weder an einer Kasse noch an einer Zahlstation oder Scanner-Ausgabestelle anstehen müssen.
- Der Laden kann dem Kunden produktspezifische Informationen direkt nach dem Scannen einblenden (z. B. Aktionen, Herkunft, etc.)
- Der Kunde kann seine Einkäufe schnell und einfach bargeldlos bezahlen. Dabei braucht er seine Kunden-, EC- oder Kreditkarte nicht vorzuweisen.
- Da die Einkäufe personenspezifisch digital erfasst werden, können sowohl der Kunde als auch der Laden statistische Auswertungen über das Kaufverhalten durchführen.
- Die Lagerbuchhaltung ist automatisch auf dem aktuellsten Stand. Dadurch kann das Lager effizienter geführt werden.

5.4 Stand der Technik/Konkurrenzanalyse

Verschiedene Einkaufshäuser wie z.B. Migros oder Coop kennen bereits Selfscanning-Systeme. Diese unter den Namen „Subito“ [1] bzw. „Passabene“ [2] geführten Systeme basieren jedoch auf stationären Self-Scanning-Kassen [2] oder speziellen mobilen Scannern [1], die an die Kunden ausgegeben und wieder zurückgenommen werden müssen. Beide Systeme sind mit erheblichen Investitionen sowie Betriebs- und Wartungskosten verbunden. Zudem besteht die Gefahr, dass Wartezeiten bei der Aus- und Rückgabe der Scanner oder den Zahlstationen entstehen, falls diese nicht grosszügig genug ausgestattet sind.

Die vorgeschlagene App benötigt keinerlei stationäre Scanner oder zusätzliche Hardware, die an die Kunden ausgeliehen werden muss, und der Kunde kann den Einkauf einfach von seinem Smartphone aus bezahlen, ohne eine zentrale Zahlstation aufsuchen zu müssen.

Mit TWINT [3] und Paymit [4] wurden in der Schweiz zwei Mobile-Payment-Lösungen entwickelt, die das einfache mobile Bezahlen ermöglichen sollen. Die beiden Systeme haben sich im Mai 2016 zusammengeschlossen [5], um gegen das internationale Bezahlssystem von Apple, ApplePay [6], anzutreten.

Diese mobilen Bezahlssysteme weisen keinerlei Funktionalität bezüglich des Produktescannens auf. Eine Kombination der vorgeschlagenen App mit diesen mobilen Bezahlssystemen ist jedoch eine attraktive Option für die Zukunft.

5.5 Hauptablauf / Szenario

Der Hauptanwendungsfall ist der Kunde, der eine Menge von Artikeln kaufen will:

Ein Kunde betritt den Laden und startet die Einkaufs-App auf seinem Smartphone. Die App hat er bereits zu Hause heruntergeladen und eingerichtet. Der Kunde startet den Einkauf auf der App. Er scannt alle Artikel, die er in den Einkaufswagen legt, mit der Smartphone-Kamera. Bei jedem eingescannten Artikel zeigt die App die wichtigsten Informationen und auch den Preis des entsprechenden Produkts an. Die Mengenangabe gibt der Kunde von Hand ein oder scannt denselben Artikel entsprechend oft. Am Ende des Einkaufs schliesst der Kunde den Einkauf auf seiner App ab. Die App zeigt ihm nun eine Übersicht über die eingekauften Artikel sowie die Gesamtkaufsumme. Nachdem der Kunde den gesamten Einkauf bestätigt hat, wählt er als Zahlungsmethode seine hinterlegte EC-Karte. Die App belastet den Kaufbetrag der ausgewählten EC-Karte und zeigt eine Bestätigung der erfolgreichen Zahlung. Er erhält umgehend einen Quittungsbeleg auf der App und auch per Mail, auf dem alle eingekauften Artikel aufgeführt sind. Nun kann der Kunde den Laden mit den eingekauften Artikeln verlassen.

5.6 Weitere Anforderungen

- Die App soll sich beim Aufstarten automatisch mit dem zentralen Einkaufssystem des Ladens verbinden.
- Ein Einkauf muss unterbrochen und später fortgesetzt werden können, ohne Daten zu verlieren.
- Die App muss auf der gesamten Einkaufsfläche des Ladens funktionieren.
 - 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, an die alle Einkäufe übermittelt werden. Diese können dort archiviert und ausgewertet werden.
 - Folgende weiteren Anforderungen sind einzuhalten:
 - Die Antwortzeit der App beim Einscannen eines Artikels soll kleiner als 0.5 Sekunden sein.
 - Ein Einkauf über die App muss in 99.9% der Fälle erfolgreich durchführbar sein.
 - Sicherheit: In Bezug auf unbefugten Zugriff, Datenmanipulation und sichere Datenübertragung sind die branchenüblichen Sicherheitsstandards für mobiles Bezahlen einzuhalten.
- Erstes Release der App nur für Android, ab Release 2 auch für iOS.

5.7 Ressourcen

Für die Erstellung der Software wird ein Kernteam von ca. 5 Personen benötigt. Alle müssen über praktische Erfahrungen mit OOP verfügen, wobei mindestens eine Person Erfahrungen in der Entwicklung von Mobilapplikationen haben sollte.

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. 40 «Personen-Monate» geschätzt.

5.8 Risiken

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

5.9 Grobplanung

Die Grobplanung sieht für die Grundfunktionalität einer ersten Version eine Entwicklungszeit von 12 Wochen vor. Die Entwicklung erfolgt iterativ-inkrementell und anwendungsfallorientiert. In einer ersten Analyse wurden folgende Use-Cases und Risiken identifiziert:

5.9.1 Use-Cases

1. Mit Einkaufssystem verbinden
2. Artikel einscannen
3. Artikel einkaufen
4. Einkauf abschliessen
5. Einkauf bezahlen mit EC-Karte
6. Artikel löschen

5.9.2 Risiken

1. Keine Scan-Software für Barcodes verfügbar für Smartphones
2. Verbindungsmöglichkeiten mit Einkaufssystem unbekannt

5.9.3 Grober Projektplan

Für die Entwicklung einer ersten Version wird ein Aufwand von 600 Personenstunden veranschlagt. Dieser soll in zweiwöchigen Iterationen in einem Zeitraum von 12 Wochen entwickelt werden gemäss folgendem Gesamtprojektplan mit Meilensteinen:

Meilenstein	Iteration	Start/Dauer [Wo]	Geplant (h)	Ziele
	#1	1 / 2	100	Projektskizze erstellt, Entwicklungsumgebung aufgesetzt, alle UCs für Rel. 1 identifiziert, Weitere Anforderungen erhoben. UC2 u. R1–2 analysiert, Architektur skizziert
Projektskizze	M1	Ende Wo 2		Vision, Ziele und Anforderungen vereinbart
	#2	3 / 2	100	UC 1–3 detailliert ausformuliert, Rest identifiziert und priorisiert, 1. Entwurf Domänenmodell, UI-Papierprototyp vorh.
	#3	5 / 2	100	UC 4-6 detailliert ausformuliert, Domänenmodell fertig, Architektur stabil und als PoC verifiziert (anhand UC2)
Lösungsarchitektur	M2	Ende Wo 6		Architektur verifiziert
	#4	7 / 2	100	UC2 realisiert und getestet, UI Prototyp implementiert



	#5	9 / 2	100	UC1 und UC3 realisiert und getestet, UI aufdatiert
	#6	11 / 2	100	UC4, UC5 und UC6 realisiert und getestet, UI aufdatiert
Beta-Release	M3	Ende Wo 12		Erste Version fertig implementiert, inkl. Systemtests und dokumentiert

5.10 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 Gesamtaufwand von ca. 0.8 Mio. CHF führt. Bei einem Deckungsbeitrag von 50'000 CHF pro verkauftem Selfscanning-System müssen demnach 16 Systeme verkauft werden, um die Auslagen der Grundentwicklung zu decken. Eine vorsichtige Schätzung rechnet mit 0.5% der Filialen der grösseren Detailhandelsketten, die pro Jahr mit diesem System ausgerüstet werden können. Migros und Coop alleine haben ca. 3000 Filialen, was 15 verkauften Systemen pro Jahr entspricht. Der Return on Investment sollte deshalb in etwa einem Jahr erreichbar sein.

5.11 Quellen

- [1] MIGROS. (2014). *Subito – einfach und schnell einkaufen* [Online]. URL: <http://www.migros.ch/de/supermarkt/subito.html> [Stand: 5.9.2014]
- [2] COOP. (2014). *Passabene – Self-Checkout* [Online]. URL: <http://www.coop.ch/pb/site/retail/node/80441021/Lde/index.html> [Stand: 5.9.2014]
- [3] TWINT. (2016). *TWINT Dein digitales Portmonnaie* [Online]. <https://www.twint.ch/> [Stand: 19.9.2016]
- [4] UBS. *UBS Paymit* (2016) [Online]. https://microsites.ubs.com/paymit/de/?campID=SEM-PAYMIT-CH-DEU-GOOGLE-PAYMIT_EXAKT-ANY-paymit-e-c [Stand: 19.9.2016]
- [5] TWINT. (2016). *Gemeinsame Mobile-Payment-Lösung für die Schweiz beschlossen* [Online]. URL: <https://www.twint.ch/gemeinsame-mobile-payment-loesung/> [Stand: 19.9.2016]
- [6] Apple: *Sicher einkaufen, im Geschäft in Apps und jetzt auch im Internet* [Online]. <http://www.apple.com/chde/apple-pay/> [Stand: 19.9.2016]