

Wissensicherung

Asha Schwegler

18. Mai 2022

Inhaltsverzeichnis

1	LE01	2
1.1	Was ist Software Engineering?	2
1.2	Was für Prozesse bzw. Disziplinen können im Software Engineering unterschieden werden?	2
1.3	Was sind die Charakteristiken eines iterativ-inkrementellen Softwareentwicklungsprozesses?	2
1.4	Warum wird im Software Engineering modelliert und was für Modelle werden erstellt?	3
1.5	Welche Artefakte werden in der Anforderungsanalyse erstellt und wozu werden sie gebraucht?	3
2	LE02	3
2.1	Was ist Usability und Usability-Engineering?	3
2.2	Was ist Usability-Engineering und was sind seine Ziele?	3
2.3	Welche 7 Usability-Aspekte sind gemäss ISO EN 9241-110 wichtig und was fordern sie?	4
2.4	Was sind die wichtigsten Artefakte aus dem UCD-Prozess und was beschreiben sie?	4
3	LE03	5
3.1	Fully-dressed UCs	5
3.2	Weitere Anforderungen	7
3.3	Systemsequenzdiagramm	9
3.4	Contracts	9
4	LE04	9
4.1	Domänenmodell Erweiterungen	9

1 LE01

1.1 Was ist Software Engineering?

- Herstellung oder Entwicklung von Software, Organisation und Modellierung der zugehörigen Datenstrukturen und dem Betrieb von Softwaresystemen.
- Anhand eines strukturierten (Projekt-)Planes. (Schritte, Phasen, Meilensteine)
- Schritte während Entw.Prozess eng miteinander verzahnt.

1.2 Was für Prozesse bzw. Disziplinen können im Software Engineering unterschieden werden?

Kernprozesse

- Anforderungserhebung
- Systemdesign/technische Konzeption
- Implementierung
- Softwaretest
- Softwareeinführung
- Wartung/Pflege

Unterstützungsprozesse

- Projektmanagement
- Qualitätsmanagement
- Risikomanagement

1.3 Was sind die Charakteristiken eines iterativ-inkrementellen Softwareentwicklungsprozesses?

- Abwicklung in Iterationen
- Inkrement = In jeder Iteration ein Stück SW entwickelt
- Ziele sind Risiko-getrieben
- Iterationsreviews mit Learnings für nächste Iteration

1.4 Warum wird im Software Engineering modelliert und was für Modelle werden erstellt?

Analyse- und Designentwürfe : diskutieren, abstimmen, dokumentieren und kommunizieren.

- Verstehen eines Gebildes
- Kommunizieren
- Gedankliches Hilfsmittel
- Kritisieren
- Experimentieren
- Aufstellen und Prüfen von Hypothesen
- in OOP:
 - Statische Modelle:
 - * Klassen und Assoziationen
 - Dynamische Modelle:
 - * Abläufe und Verhalten

1.5 Welche Artefakte werden in der Anforderungsanalyse erstellt und wozu werden sie gebraucht?

- Systemabgrenzung und Systemkontextdiagramm
- Use-Case-Modell und UI-Sketches
- Qualitätsanforderungen und Randbedingungen
- Domänenmodell

2 LE02

2.1 Was ist Usability und Usability-Engineering?

Usability: Die Effektivität, Effizienz und Zufriedenheit mit der die adressierten Benutzer ihre Ziele erreichen in ihren spezifischen Kontexten.

Usability Engineering: Software entwickeln, die die drei Anforderungen von Usability erfüllen.

2.2 Was ist Usability-Engineering und was sind seine Ziele?

- Usability-Engineering = Software-Ergonomie
- Ziel: SW-Produkte entwickeln, die effektiv, effizient und zufriedenstellend sind.

2.3 Welche 7 Usability-Aspekte sind gemäss ISO EN 9241-110 wichtig und was fordern sie?

1. Aufgabenangemessenheit
 - Aufwand im Vergleich zu Aufgaben und Ziele sollte angemessen sein.
2. Selbstbeschreibungsfähigkeit
 - Wissen wo in der SW man ist und was man tun muss/kann und was das System tut.
3. Kontrolle
 - Kontrolle über Interaktion mit System haben.
4. Erwartungskonformität
 - Funktionalität
 - Interaktion
 - Design
 - Struktur
 - Ansprechen der Komplexität
5. Fehlertoleranz
 - Fehler vermeiden
 - Fehler und Ursache erkennen
 - Fehler korrigieren
6. Individualisierbarkeit
 - Anpassbar auf Bedürfnisse (Laien, Experten, Benutzer mit besonderen Bedürfnisse)
7. Lernförderlichkeit
 - Informationen über unterliegende Konzepte, Regeln, Verfahren und neue Funktionalitäten/Interaktionsmöglichkeiten

2.4 Was sind die wichtigsten Artefakte aus dem UCD-Prozess und was beschreiben sie?

1. Personas
 - Repräsentiert Benutzergruppe (fiktive Beschreibung)
2. Usage-Szenarien
 - Aktuelle Situation, Beschreibung wie Persona BESTEHENDES System benutzt um Aufgabe zu lösen.
3. Kontextszenario
 - Zukünftige Situation, Beschreibt wie Persona zukünftiges im Idealfall benutzen wird.

4. Storyboard

- Visualisierung vom Kontextszenario

5. Mentales Modell

- Domänenmodell (Vorstellung Benutzers über Problemdomäne)

6. Wireframes (Interaktionsprototypen)

- Demonstration der Interaktion mit dem System

7. Stakeholder Map

- Direkte Akteure und weitere Stakeholder, die interessiert oder betroffen sind vom SW-Produkt

8. Service Blueprint / Geschäftsprozessmodell

- Darstellung logischen Schritte Service kunden, Service-Providers, und über welche Kanäle interagiert wird.

3 LE03

3.1 Fully-dressed UCs

UC01: Teilbestellung erfassen

Umfang: Swift4Restaurants-Anwendung

Ebene: Anwenderziel

Primärakteur: Servierperson

Stakeholder und Interessen:

- Servierperson, anderes Servierpersonal:
 - Schnelle, genaue und flexible Erfassung der Bestellwünsche
- Gast:
 - Schnelle und genaue Erfassung seines Bestellwunsches
 - Nicht mehr bezahlen als konsumiert
- Koch:
 - Genaue Erfassung des Bestellwunsches
- Gastwirt:
 - Schnelle und genaue Erfassung Bestellwunsches

- genaue Abrechnung am Abend
- Steueramt:
 - Genaue und nachvollziehbare Abrechnung der Steuern.

Vorbedingungen:

- Servierperson muss angemeldet sein

Nachbedingungen:

- Teilbestellung ist gespeichert
- Teilbestellung wird an der Theke und auf dem Bildschirm des Kochs als aufgegeben angezeigt.

Standardablauf:

1. Servierperson fragt nach Bestellung.
2. Gast äussert seinen Bestellwunsch.
3. Servierperson erfasst den Bestellwunsch.
4. Das System zeigt Detailinformationen des Bestellwunsches an.
5. Servierperson bestätigt den Bestellwunsch. Loop 1-5 bis alle Bestellwünsche der Teilbestellung erfasst sind.
6. Serviceperson schliesst Teilbestellung ab.
7. System zeigt die Detailinformationen zur Teilbestellung.
8. System zeigt die Teilbestellung am Zentralcomputer als «aufgegeben» an.
9. Bestellpositionen, die Menus betreffen, zeigt System auf dem Bildschirm in der Küche als «aufgegeben» an

Erweiterungen:

- 3a: Bestellposition++, wenn Gast dasselbe bestellt wie vorheriger Gast.
- 3b: Falls 1. Bestellung des Tisches: Neue Bestellung eröffnen
- 3c: Falls Tisch von einer anderen Servierperson betreut wird, laufende Bestellung übernehmen und Servierperson informieren.
- 3d: Korrektur Bestellwunsch: Bestellpos. löschen und neue einfügen.
- 3e: Wenn Bestellpos. nicht existiert-¿ Servierperson trägt von Hand ein.

Jederzeit: wenn Mobilgerät abstürzt

1. Servierperson holt sich neues Mobilgerät
2. Servierperson Meldet sich an und gibt Tischnummer ein
3. Zeigt aktuellen Stand der TEilbestellung
4. Servierperson fährt dort weiter, wo sie unterbrochen worden ist.

Spezielle Anforderungen:

- Touch UI, Stift oder Finger
- 0.1s Antwortzeit
- Bei Absturz, erfassten Bestellwünsche wiederherstellbar
- Automatisch miterfass: Genauer Zeitpunkt Bestellerfassung, Tischnummer, Servierperson.

Liste der Technik- und Datenvariationen

- Mobilgerät soll barchodes einscannen können falls menus mit Barcodes versehen sind.

Häufigkeit des Auftretens:

- Beinahe laufend

Offene Fragen:

- Wie kann verhindert werden, dass Bestellung für eine falsche Tischnummer erfasst wird?
- Wie kann verhindert werden, dass gleichzeitig mehrere Bestellungen für denselben Tisch aktiv sind?

3.2 Weitere Anforderungen

Funtionality

- Android OS
- Mögliche Bestellpositionen auf dem Mobilgerät müssen automatisch vom Zentralcomputer geladen werden, sobald sie dort angepasst werde
- Das System muss vor Zugriffen nicht autorisierter Personen geschützt werden.

Usability

- Mobilgerät muss von jeder ausgebildeten Servierperson nach einer max. einminütigen Einführung problemlos bedient werden können
- Pro Servierperson und Tag soll es max. 5 Fehleingaben geben
- Eingaben am Mobilgerät sollen problemlos mit Eingabestift machbar sein, notfalls auch mit einem Finger
- Die Mobilgeräte müssen auch bei starker Sonneneinstrahlung und bei praktischer Dunkelheit bedienbar sein.

Reliability

- Ein Mobilgerät soll mit 14h bei normalem Servierbetrieb laufen, ohne aufgeladen werden zu müssen.
- Das Mobilgerät muss ein Sturz von 1m Höhe unbeschadet überstehen.
- Bei einem Absturz des Systems, muss dieses innerhalb von 3' wieder hochgefahren werden können inkl. Wiederherstellung aller Daten

Performance

- Die Eingabe einer Bestellposition darf nicht länger als 0.2s in 90
- Es müssen Bestellungen mit bis zu 20 Mobilgeräten gleichzeitig vorgenommen werden können.
- Das System muss über ein handelsübliches WLAN funktionieren.

Scalability

- HW Ausbau bis auf 100 erhöht werden

+

- Das System muss folgende Sprachen unterstützen (Deutsch, Französisch, Italienisch, Englisch)
- Das System muss Preise in CHF und Euro verarbeiten können.
- Die Mobilgeräte müssen wasserspritzfest sein, so dass sie auch bei Regen bedient werden können
- Der Touchscreen in der Küche muss auch unter den Bedingungen einer Küche problemlos bedient werden und aus bis zu 5m Abstand abgelesen werden

3.3 Systemsequenzdiagramm

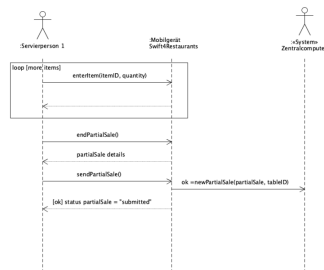


Abbildung 1: SSDAufgabe.

3.4 Contracts

Für die wichtigste System-Operation einen Vertrag:

Vertrag enterItem

- Operation: enterItem(itemID: itemID, quantity: integer)
- Querverweis: UC Teilbestellung erfassen
- Vorbedingung: Eine Teilbestellung ist aktiv für diesen Tisch
- Nachbedingungen:
 - bpi wurde erstellt (Bestellpositionsinstanz)
 - bpi mit aktueller Teilbestellung für aktiven Tisch verknüpft
 - bpi.quantity wurde auf quantity gesetzt
 - bpi wurde anhand übereinstimmenden itemID mit einer ProductDescription verknüpft

4 LE04

4.1 Domänenmodell Erweiterungen

Ursprüngliches Modell:

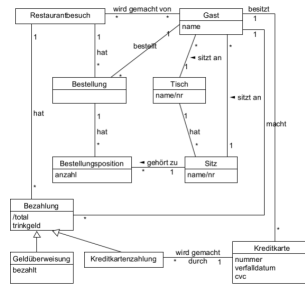


Abbildung 2: DMAufgabeUrsprung.

Verbesserungen:

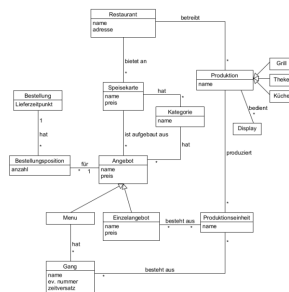


Abbildung 3: DMAufgabeVerbesserung.

- Produktion eingeführt als Generalisierung für mehrere Theken und Küchen
- Kategorien
- Gang (Produktionseinheit und Menu)
- Restaurantbesuch, zeitlich limitiert
- Bestellungen und Bezahlungen spezialisiert
- Konzept Platz eingeführt und mit Tisch verknüpft und mit Gast verbunden und mit Bestellungsposition.