



Open Education Platform
for Management Schools

Publikationstyp: Lehrmaterialien

Requirements Engineering bei der Firma Alarmitis

Version Nr. 1, 19. Dezember 2022

Tremp, Hansruedi

OST - Ostschweizer Fachhochschule

Publiziert auf: www.oepms.org

Unter doi: 10.25938/oepms.338



Open Education Platform
for Management Schools

Requirements Engineering bei der Firma Alarmitis

Version Nr. 1, 19. Dezember 2022

Tremp, Hansruedi

OST - Ostschweizer Fachhochschule

Publikationsform: Fallstudie

Institution: OST - Ostschweizer Fachhochschule

Schlüsselbegriffe: Anforderungsanalyse; Requirements Engineering;
UML Modellierung

Einsatzbereich: Bachelorstudierende

Lizenz:



Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung 4.0 International Lizenz](https://creativecommons.org/licenses/by/4.0/).

Zitierweise nach APA:

Tremp, H. (2022). Requirements Engineering bei der Firma Alarmitis. *Open Education Platform*. DOI: 10.25938/oepms.338

Didaktische Reflexion:

Requirements Engineering bei der Firma Alarmitis Aufgabenstellung

Hansruedi Tresp

OST – Ostschweizer Fachhochschule, Oberseestrasse 10, 8640 Rapperswil, hansruedi.tresp@ost.ch

Abstract. Requirements Engineering bildet im Rahmen des Bachelor Wirtschaftsinformatik ein zentrales Modul, da bedeutsame Kompetenzen in der Schnittstelle der Fachabteilungen und der ICT entwickelt werden. In der Fallstudie «Alarmitis» geht es um die objektorientierte Anforderungsanalyse im Rahmen einer Software für ein Alarmsystem. Die aktuelle Fallstudie ist seit ca. 6 Jahren im Modul zum Thema Requirements Engineering im Hauptstudium (3./4. Semester) des B.Sc. Wirtschaftsinformatik im Einsatz und wurde dabei fortlaufend erweitert bzw. aktualisiert. Die Studierenden schätzen den Praxisbezug und die Möglichkeit, die in den Vorlesungen dargelegte Theorie und unterschiedlichen Methoden an einem konkreten Fall anzuwenden.

Inhaltsverzeichnis

1.1	Einführung	2
1.2	Lernziele	2
1.3	Lehrplan	2
1.4	Allgemeine Bewertungskriterien	3

Didaktischer Reflexionsbericht

1.1 Einführung

Requirements Engineering bildet im Rahmen des B.Sc. Wirtschaftsinformatik ein zentrales Modul, da bedeutsame Kompetenzen in der Schnittstelle der Fachabteilungen und der ICT entwickelt werden. In der Fallstudie «Alarmitis» geht es um die objektorientierte Anforderungsanalyse im Rahmen einer Software für ein Alarmsystem. Die aktuelle Fallstudie ist seit ca. 6 Jahren im Modul zum Thema Requirements Engineering im Hauptstudium (3./4. Semester) des B.Sc. Wirtschaftsinformatik im Einsatz und wurde dabei fortlaufend erweitert bzw. aktualisiert.

Die Fallstudie gibt den Studierenden einen authentischen Situationsbezug zur Arbeit als Requirements Engineer. Sie ermöglicht die Anwendung der erarbeiteten Konzepte, Verfahren, Methoden und Techniken im praxisorientierten Kontext mit Schwerpunkt auf die klassische Anforderungsanalyse. Die Aufgabenstellungen fördern die berufsorientierten Kompetenzen der Informationsermittlung, strukturierten Dokumentation, objektorientierten Analyse mittels UML Modellierung, Demonstrationsprototypenerstellung und der Sicherung der Qualität der erstellten Artefakte im Requirements Engineering Prozess. Elemente der agilen Vorgehensweise wie die User Story sind ergänzend aufgenommen.

1.2 Lernziele

Die Zielgruppe bilden Bachelor-Studierende der Informatik bzw. Wirtschaftsinformatik im Hauptstudium. Die Einführung in die Programmierung sowie die Grundlagen der Betriebswirtschaftslehre bilden die Voraussetzung.

Die Studierenden können

- eine Softwareproduktevision mit Visionstext, Kontextabgrenzung und SMART-Zielen verständlich formulieren.
- für funktionale und nicht-funktionale Anforderungen die passenden Informationsquellen eruieren und mittels geeigneter Erhebungstechniken vollständig ermitteln.
- die Anforderungsspezifikation verständlich und attributbasiert formulieren.
- eine Anforderungsanalyse mittels der UML (Unified Modeling Language) vornehmen.
- mit einem Demonstrationsprototypen (UI-Wireframe) für ausgewählte Anwendungsfälle die Anforderungsdokumentation ergänzen.
- die Qualität der Requirements Engineering Artefakte sicherstellen.

1.3 Lehrplan

Die Fallstudie wird im Rahmen des Moduls «Requirements Engineering» eingesetzt. Sie begleitet den Unterricht über mehrere Unterrichtswochen. Dabei sind folgende Zeiten vorzusehen:

- Vorstellung und Einführung der Fallstudie im Unterricht ca. 30 Min.
- Die Aufgabenstellung wird im Rahmen einer Gruppe von 3 - 5 Studierenden gelöst.
- Für die Lösung der Aufgaben ist im Durchschnitt mit ca. 30 bis 90 Min. Aufwand je Aufgabe in der Gruppe zu rechnen.

- Für die Besprechung im Unterricht und einem primären Feedback sind ca. 15-20 min. je Gruppe und Aufgabe einzuplanen.
- Für die Präsentation der erstellten Artefakte eignet sich ein E-Portfolio, wie es z.B. SWITCH-portfolio (siehe www.switch.ch/de/services/portfolio) anbietet.
- Zur Förderung der Beurteilungskompetenz können Peer-Reviews vorgesehen werden, wobei jede Gruppe jeweils die Arbeiten von 2 anderen Gruppen gemäss den gegebenen beobachtbaren Kriterien bewertet.
- Für die Korrektur und allfällige Bewertung der Gruppenarbeit hat der Dozierende mit ca. 45 min. je Gruppe zu rechnen.

1.4 Allgemeine Bewertungskriterien

Bei den Lösungen der einzelnen Aufgaben sind nebst den dort aufgeführten Beurteilungskriterien auch die nachfolgenden zu berücksichtigen:

- Vollständigkeit:
 - Alle relevanten Aspekte der Ausgangslage sind adäquat berücksichtigt.
- Zweckmässigkeit:
 - Die vorgeschlagene Lösung ist praktikabel und zielführend.
- Einsatz von Methoden und Techniken
 - Die Methoden bzw. Techniken sind korrekt angewendet.
- Argumentation:
 - Die Begründungen sind in sich schlüssig und zeigen einen klaren Bezug sowohl zur Aufgabenstellung als auch zur Theorie.



Fallstudie:

Requirements Engineering bei der Firma Alarmitis Aufgabenstellung

Hansruedi Tresp

OST – Ostschweizer Fachhochschule, Oberseestrasse 10, 8640 Rapperswil, hansruedi.tresp@ost.ch

Abstract. Requirements Engineering bildet im Rahmen des Bachelor Wirtschaftsinformatik ein zentrales Modul, da bedeutsame Kompetenzen in der Schnittstelle der Fachabteilungen und der ICT entwickelt werden. In der Fallstudie «Alarmitis» geht es um die objektorientierte Anforderungsanalyse im Rahmen einer Software für ein Alarmsystem. Die aktuelle Fallstudie ist seit ca. 6 Jahren im Modul zum Thema Requirements Engineering im Hauptstudium (3./4. Semester) des B.Sc. Wirtschaftsinformatik im Einsatz und wurde dabei fortlaufend erweitert bzw. aktualisiert. Die Studierenden schätzen den Praxisbezug und die Möglichkeit, die in den Vorlesungen dargelegte Theorie und unterschiedlichen Methoden an einem konkreten Fall anzuwenden.

Inhaltsverzeichnis

Abkürzungsverzeichnis	3
1. Einführung	4
2. ICT-Strategie	4
3. Produktidee	4
4. Aufgaben.....	5
4.1 Die Softwareproduktevision erstellen.....	5
4.2 Die Stakeholderanalyse vornehmen.....	6
4.3 Die Ermittlung (Erhebung) planen.....	6
4.4 Die nicht-funktionalen Anforderungen (NFA) spezifizieren.....	6
4.5 Die funktionalen Anforderungen spezifizieren.....	7
4.6 Das Anwendungsfallmodell erstellen	7
4.7 User Stories formulieren.....	7
4.8 Einen Demonstrationsprototypen (UI-Wireframe) erstellen.....	8
4.9 Die objektorientierte Informationsanalyse vornehmen.....	8
4.10 Die Objektinteraktion modellieren	9
4.11 Die Funktionsanalyse vornehmen.....	9
4.12 Die Verhaltensanalyse vornehmen	9
4.13 Den Prüfplan für die Artefakte des RE (Requirements Engineering) erstellen	10
Literaturverzeichnis.....	11

Abkürzungsverzeichnis

BSc	Bachelor of Science
ERP	Enterprise Resource Planning
ICT	Information and Communication Technology
IT	Information Technology
LDS	Lightweight Directory Service
MA	Mitarbeitende
OData	Open Data Protocol
RE	Requirements Engineering
REST	REpresentational State Transfer
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SPA	Single Page App
UI	User Interface
UML	Unified Modeling Language

Fallstudie

1. Einführung

Die Alarmitis GmbH bietet mit ca. 400 Mitarbeitende seit mehr als 30 Jahren einer internationalen Kundschaft hochstehende Alarmsysteme an. Aufgrund von neuen Systemkomponenten und mehreren Schwachstellen des aktuellen Softwaresystems hat sich die Geschäftsleitung entschieden, eine neue Software mit dem Produktnamen «Alarmsteuerung+» zu entwickeln. Sie erteilt der ICT-Abteilung den Auftrag, in einer ersten Phase eine solide Anforderungsanalyse vorzunehmen und die Resultate zusammen mit einem machbaren Realisierungsplan innert 3 Monaten zu präsentieren.

2. ICT-Strategie

Alarmitis hat sich ihre ICT-Strategie vor Jahren erarbeitet und überprüft diese in einem jährlichen Strategie-Workshop. In einer ersten Sichtung hat das Requirements Engineering Team die nachfolgenden Punkte notiert. Die konsequente Umsetzung von SOA (Service Oriented Architecture) steht ganz zuvorderst. Zu entwickelnde Applikationen sollen konsequent auf die Bedürfnisse einer internationalen Kundschaft ausgerichtet sein. Für die optimale Integration mit den Umsystemen sind offene standardisierte Services, vorzugsweise REST, vorzusehen. Clientseitig strebt Alarmitis die Entwicklung von Mobile-First Web Apps an, welche sich an ISO 9241, der internationalen Richtlinien der Mensch-Computer-Interaktion, orientiert. Eine intern erarbeitete, verpflichtende UI (User Interface) Guideline steht der Softwareentwicklung zur Verfügung. Die Software hat sich grundsätzlich an die Compliance-Richtlinien der Unternehmung zu halten. Die Security-Anforderungen sind gemäss den Vorgaben von ISO 27001, der internationalen Richtlinien für Informationssicherheit, abzuleiten. Alle Anlagen sind bei der VdS Schadenverhütung GmbH, Köln (siehe www.vds.de) zu zertifizieren. Die Datenschutzgesetze sind sowohl auf EU- als auch Schweizer-Ebene einzuhalten. Die Benutzerverwaltung erfolgt auf dem zentralen Identity-System und wird als Cloud-Service allen Kunden zur Verfügung gestellt. Die ICT verfolgt eine Clean-Code-Strategie und bietet deshalb für die Softwareentwicklung ein verbindliches Programmierhandbuch mit Code-Konventionen und einzuhaltende Prinzipien.

3. Produktidee

Das Requirements Engineering Team hat in einem ersten Schritt vorhandene Dokumente über die Produktidee zusammengetragen:

Das Alarmsystem besteht aus mehreren frei zusammenstellbaren Komponenten:

- Zentrale Einheit mit Touchscreen
- Türschliesssysteme
- Bewegungsmelder
- Sirenen
- Notstromversorgung
- Kommunikationseinheit: LAN, WLAN, WAN, 4G, 5G

Die Benutzenden können das System an der zentralen Einheit, über eine Web-App oder eine Mobile-App bedienen.

Folgende Daten-Schnittstellen sollen als RESTful oder SOAP (Simple Object Access Protocol) Service konfigurierbar sein:

- Externe Meldestellen, wie z.B. private Sicherheitsfirmen oder Polizei.
- Kommunikationsserver, welcher die Alarmitis den Kunden als Cloud-Service zur Verfügung stellt, um Short-Messages oder E-Mails zu versenden.
- Software-Update-Server, bei welchem die Steuerungssoftware die Softwareupdates automatisch abholt und danach installiert.

Weitere individuelle Schnittstellenerweiterungen sollen möglich sein. Das Alarmsystem verfügt im eingeschalteten Zustand über zwei Grund-Modi: «inaktiv» und «aktiv». Autorisierte Benutzende können das System aktivieren bzw. deaktivieren.

Ist das System im Modus «aktiv», so wird, sobald ein Bewegungsmelder jemanden erkannt hat, folgender Ablauf ausgelöst:

- Vorstufe Alarm: während x Sek. (gem. Parametereinstellung) ertönt ein entsprechender Pieps-ton und die Benutzenden können mittels eines Codes den Alarm deaktivieren.
- Auslösung Alarm: die Sirenen werden aktiviert, entsprechende Meldungen werden an alle Bezugspersonen versandt und eine Meldung wird an die Meldestelle übermittelt, welche diese bestätigt.
- Deaktivieren Alarm: der einmal ausgelöste Alarm kann z.B. über die Meldestelle mittels der korrekten Meldung deaktiviert werden. Es sollen aber auch andere Kommunikationswege einstellbar sein.

Das System soll täglich um eine frei festlegbare Zeit (sofern das System inaktiv ist) auf dem Software-Update-Server nach einer neuen Version prüfen und (sofern vorhanden) diese installieren. Die registrierten Bezugspersonen erhalten eine entsprechende Mitteilung. Die Administrierenden können die Parameter der Anlage mittels der Web-App verwalten.

4. Aufgaben

Alle nachfolgenden Aufgaben zu dieser Case Study müssen die oben beschriebene Ausgangslage adäquat berücksichtigen und die im Requirements Engineering anerkannten Methoden und Techniken korrekt anwenden. Konsultieren Sie zu den einzelnen Themen die in Kapitel 4 aufgeführte Fachliteratur.

4.1 Die Softwareproduktevision erstellen

Erstellen Sie eine Produktevision für die Steuerungssoftware mit folgenden Elementen, damit die beteiligten Stakeholder eine präzisere Vorstellung erhalten:

- Produktebeschrieb mit Kontextabgrenzung
- Motivation und Nutzen
- Ziele

Visualisieren Sie den Scope der Software mittels eines Kontextdiagramms, in welchem die Umsysteme sowie die relevanten Benutzergruppen ersichtlich sind.

Beurteilungskriterien (siehe auch 2.4):

- Der Produktebeschrieb von ca. 3 - 4 ganzen Sätzen ist gut verständlich.
- Das Kontextdiagramm ist korrekt und mit einer gängigen Notation modelliert.
- Die Motivation und der Nutzen, in ganzen Sätzen formuliert, sind nachvollziehbar aus der Ausgangslage abgeleitet und passend.
- Mindestens 3 Ziele sind SMART formuliert und der Bezug zur Ausgangslage ist gegeben.

4.2 Die Stakeholderanalyse vornehmen

Identifizieren Sie die zehn wichtigsten Stakeholder dieses Projektes. Berücksichtigen Sie dabei sowohl die organisationsinternen als auch -externen Stakeholder des gesamten Projektumfeldes. Erstellen Sie eine **Stakeholder Tabelle** mit folgenden Attributen:

- Stakeholder-Bezeichnung
- Funktion im Projekt (Stichworte)
- Einschätzung des Wissens in Bezug zu den Geschäftsprozessen (von 0 - 10)
- Einschätzung des Wissens in Bezug zur Technologie (von 0 - 10)
- Einschätzung des Einflusses auf den Projekterfolg (Macht) (von 0 - 10)
- Einschätzung des Nutzens aus dem Projekt (Interesse) (von 0 - 10)
- Als Requirements-Provider zu berücksichtigen (ja/nein)
- Stichwortartige Begründung mit Hinweis auf die Ausgangslage oder das anzunehmende Projektumfeld

Beurteilungskriterien (siehe auch 2.4):

- Ca. 10 der wichtigen, relevanten Stakeholder sind identifiziert.
- Die Einschätzungen sind stimmig.
- Sinnvolle Requirements Provider sind festgelegt.
- Die Begründung ist in sich schlüssig und nachvollziehbar.

4.3 Die Ermittlung (Erhebung) planen

Planen Sie im nächsten Schritt die Ermittlung (Erhebung) der Anforderungen. Wählen Sie dazu projektadäquate Ermittlungstechniken aus und erstellen Sie eine **Planungsliste** mit folgendem Inhalt:

- Reihenfolge-Nr.
- Ermittlungstechnik
- Ermittlungsinhalt (Stichworte)
- Beteiligte Stakeholder

Berücksichtigen Sie alle in der Aufgabe 3.4.2 festgelegten Requirements-Provider.

Beurteilungskriterien (siehe auch 2.4):

- Ca. 5 unterschiedliche für den dargelegten Projektkontext sinnvolle Ermittlungstechniken sind vorgeschlagen.
- Die Reihenfolge ist schlüssig und zielführend.
- Der Ermittlungsinhalt ist passend und insgesamt vollständig.
- Die beteiligten Stakeholder sind als Requirements-Provider markiert.

4.4 Die nicht-funktionalen Anforderungen (NFA) spezifizieren

Eruieren Sie basierend auf dem Software Produkte Qualitätsmodell ISO/IEC 25010 ca. 10 der wichtigsten NFA. Notieren Sie das Qualitätshauptmerkmal als Titel. Formulieren Sie die Anforderungen unter Anwendung einer adäquaten Satzschablone, wie z.B. in SOPHISTen (2016) dargelegt. Geben Sie in Bezug auf die gewählten Anforderungen eine Beurteilung in Form einer Rangfolge (1 am wichtigsten) bezüglich der Wichtigkeit in Bezug auf die anderen NFA ab

Beurteilungskriterien (siehe auch 2.4):

- Ca. 10 der relevanten nicht-funktionalen Anforderungen sind identifiziert.
- Der Bezug zu ISO/IEC 25010 ist dargelegt.
- Die Formulierung ist sinnvoll und mit einer Satzschablone vorgenommen.
- Eine Beurteilung in Bezug auf Wichtigkeit mit Rangangabe (1-6) ist vorgenommen

4.5 Die funktionalen Anforderungen spezifizieren

Eruieren Sie basierend auf der Ausgangslage ca. 15 wichtige funktionale Anforderungen. Erstellen Sie eine Tabelle mit den nachfolgenden Attributen:

- Id-Nr.
- Bezeichnung
- Kurzbeschreibung (1 - 2 Sätze)
- Begründung mit Bezug zur Ausgangslage
- Owner (Stakeholder-Bezeichnung)
- Verbindlichkeit (must, can, nice-to-have)
- Kritikalität (0-5)

Beurteilungskriterien (siehe auch 2.4):

- Ca. 15 der wichtigen funktionalen Anforderungen sind identifiziert.
- Die erstellte Tabelle weist alle Attribute korrekt auf.
- Jede Anforderung ist komplett und stimmig beschrieben.

4.6 Das Anwendungsfallmodell erstellen

Analysieren Sie die Ausgangslage und die bisher erstellten Artefakte. Führen Sie eine Anwendungsfallmodellierung durch. Zeichnen Sie ein Use Case Diagramm nach UML 2.x auf. Verwenden Sie nach Möglichkeit ein professionelles Modellierungstool wie z.B. Visual Paradigm (siehe www.visual-paradigm.com).

Beurteilungskriterien (siehe auch 2.4):

- Das Alarmsystem ist als System modelliert.
- Alle Umsysteme und Benutzenden als Akteure sind stimmig modelliert.
- Alle relevanten Anwendungsfälle sind fachgemäss modelliert.
- Alle Beziehungen sind korrekt vorhanden.
- Die Notation von UML 2.x ist eingehalten.
- Jeder Anwendungsfall ist mit 1-2 Sätzen beschrieben und verweist zu dessen Begründung auf die Ausgangslage.

4.7 User Stories formulieren

In Agilen Projekten ist der Ansatz, Anforderungen über Epics und User Stories anstatt der Use Cases zu spezifizieren. Im Sinne einer Verfeinerung der Use Cases und szenarienorientiertem Vorgehen soll diese Aufgabe als Ergänzung dienen.

Wählen Sie aus der Aufgabe 3.4.6 einen oder zwei Use Cases aus und schreiben Sie insgesamt 2 User Stories. Verwenden Sie dabei eine für User Stories übliche Satzschablone. Geben Sie zu jeder User

Story ein Refinement, d.h. ergänzende Informationen, welche sie zu Beginn des Sprints geklärt haben wollen.

Beurteilungskriterien (siehe auch 2.4):

- 2 relevante User Stories sind basierend auf einer Satzschablone formuliert.
- Die User Stories sind verständlich und lassen sich von der Ausgangslage ableiten.
- Die jeweiligen Refinements sind adäquat.

4.8 Einen Demonstrationsprototypen (UI-Wireframe) erstellen

Erstellen Sie basierend auf der Ausgangslage ein UI-Wireframe für die Administrations-App, welche auf einem Tablet (Android oder iPad) ausführbar sein soll. Folgende User-Stories sollen abgedeckt werden:

- Die Administrierenden müssen sich mit einer Zwei-Faktor-Authentifikation anmelden können.
- Die Administrierenden müssen ein Alarmsystem auswählen können.
- Die Administrierenden müssen auf einem Blick den Status des Alarmsystems sehen können.
- Die Administrierenden sehen auf der Karte den Standort und können mit einem Klick in die Google-Map-App wechseln und die Route anzeigen lassen.
- Die Administrierenden müssen das Alarmsystem ein- bzw. ausschalten können.
- Die Administrierenden müssen das Alarmsystem aktivieren bzw. in allen unterschiedlichen Zuständen deaktivieren können.
- Die Administrierenden müssen die Systemereignisse einsehen und adäquat filtern können.
- Die Administrierenden müssen beliebigen Benutzern eine Mitteilung (mit Hinzufügen eines Fotos) senden können.

Die Administrierenden können Mitarbeitende oder aber Personen von den Meldestellen (Polizei, Securitas usw.) sein.

Beurteilungskriterien (siehe auch 2.4):

- Inhaltlich sind die Anforderungen aus den angegebenen User-Stories abgedeckt.
- Das Visual-Design der einzelnen Bildschirmseiten ist ansprechend und verständlich angeordnet.
- Die Gestaltungselemente sind auf den einzelnen Bildschirmseiten adäquat eingesetzt.

4.9 Die objektorientierte Informationsanalyse vornehmen

Führen Sie eine Informationsanalyse mittels der OO-Methode durch. Identifizieren Sie alle vorkommenden Geschäftsobjekte und deren Beziehungen.

Zeichnen Sie in einem ersten Schritt ein Business Object Diagram (Geschäftsobjektdiagramm) und verwenden Sie für die einzelnen Geschäftsobjekte ansprechende Icons.

Analysieren Sie die Geschäftsobjekte eingehender und legen Sie die Klassen mit deren Eigenschaften und wichtigstem Verhalten fest. Prüfen Sie das Potenzial für Generalisierungen und führen Sie diese allenfalls durch. Modellieren Sie das Fachklassendiagramm. Je Klasse sind die aus der Ausgangslage ableitbaren Attribute und Operationen aufzuführen.

Beurteilungskriterien (siehe auch 2.4):

- Alle relevanten Objekte sind identifiziert.
- Das Geschäftsobjektdiagramm ist ansprechend, korrekt und übersichtlich.
- Das Fachklassendiagramm ist korrekt und vollständig:
 - Alle relevanten Klassen sind mit korrekter Namensgebung (Substantiv, Einzahl) vorhanden.
 - Mögliche Vererbungsstrukturen sind erkannt und entsprechend modelliert.
 - Alle notwendigen Attribute und Operationen sind dargelegt.
 - Die Assoziationen (Beziehungen) sind korrekt mit den Multiplizitäten und allenfalls weiteren Elementen, wie z.B. Beschriftung, Lese Pfeil, Rollenbezeichnung usw., modelliert.
 - UML ist korrekt nach 2.x angewendet.

4.10 Die Objektinteraktion modellieren

Modellieren Sie das nachfolgend beschriebene Szenario mittels eines UML 2.x Sequenzdiagramms:

- Der Bewegungsmelder löst beim Alarmsystem+ einen Alarm aus.
- Das Alarmsystem+ wartet 10 sec.
- Danach führt das Alarmsystem+ folgende 3 Aktivitäten parallel aus:
 - Es aktiviert die Sirene.
 - Es meldet die Alarmauslösung dem Meldesystem des Sicherheitspartners und wartet die Antwort ab.
 - Es sendet der verantwortlichen Person eine Short Message.

Beurteilungskriterien (siehe auch 2.4):

5. Alle relevanten Lebenslinien sind identifiziert korrekt modelliert.
6. Der Nachrichtenablauf ist korrekt dargelegt.
7. UML ist korrekt nach 2.x angewendet.

4.11 Die Funktionsanalyse vornehmen

Nehmen Sie die Funktionsanalyse für das Alarmsystem+ vor. Gehen Sie davon aus, dass das System die Prüfung für das Update der Software jeweils um 01:00 Uhr vornimmt. Erstellen Sie ein UML Aktivitätsdiagramm, in welchem die Funktionalität als Prozessablauf modelliert ist.

Beurteilungskriterien (siehe auch 2.4):

4. Alle relevanten Aktionen und Ereignisse sind identifiziert korrekt modelliert.
5. Der Kontrollfluss ist vollständig und fachgemäss modelliert.
6. UML ist korrekt nach 2.x angewendet.

4.12 Die Verhaltensanalyse vornehmen

Führen Sie für das Objekt «Alarm» eine Verhaltensanalyse durch. Identifizieren Sie alle möglichen Zustände sowie Ereignisse auf Basis der bisherigen Analyseergebnisse und der Ausgangslage. Modellieren Sie das komplette Verhalten des Alarm-Objektes mittels eines UML-Zustandsdiagramms.

Beurteilungskriterien (siehe auch 2.4):

- 7 Alle relevanten Zustände sind identifiziert und korrekt modelliert.
- 8 Alle relevanten Ereignisse sind identifiziert und die jeweiligen Zustandsübergänge adäquat modelliert.
- 9 UML ist korrekt nach 2.x angewendet.

4.13 Den Prüfplan für die Artefakte des RE (Requirements Engineering) erstellen

Identifizieren Sie alle RE-Artefakte. Entscheiden Sie, welche dieser Artefakte einer formellen Prüfung unterliegen sollen. Erstellen Sie einen realistischen Prüfplan, indem Sie eine Tabelle mit folgenden Kolonnen anlegen:

- Prüfobjekt mit kurzer Begründung
- Prüfmethode
- Beteiligte Mitarbeitende
- Verantwortliche Person

Beurteilungskriterien (siehe auch 2.4):

- 10 Alle relevanten Prüfobjekte sind identifiziert und kurz begründet.
- 11 Die vorgeschlagene Prüfmethode ist für den Kontext passend.
- 12 Die vorgeschlagenen beteiligten Mitarbeitenden sind sinnvoll.

Literaturverzeichnis

Rupp, C., & Queins, S. (2012). *UML 2 glasklar: Praxiswissen für die UML-Modellierung* (4., aktualisierte und erweiterte Auflage). München: Hanser.

Rupp, C., & die SOPHISTen. (2021). *Requirements-Engineering und -Management. Das Handbuch für Anforderungen in jeder Situation*. (7., aktualisierte und erweiterte Auflage). München: Hanser.

SOPHISTen (2016). *Schablonen für alle Fälle*. SOPHIST GmbH. https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_3-Auflage_interaktiv.pdf

Tremp, H. (2022). *Agile objektorientierte Anforderungsanalyse: Planen – Ermitteln – Analysieren – Modellieren – Dokumentieren – Prüfen*. Wiesbaden: Springer Vieweg

Vernon, V. (2017). *Domain-Driven Design kompakt* (1. Auflage; C. Lilienthal & H. Schwentner, Übers.). Heidelberg: dpunkt.verlag.



Open Education Platform
for Management Schools

Lösungsvorschläge:

Requirements Engineering bei der Firma Alarmitis Aufgabenstellung

Hansruedi Treppe

OST – Ostschweizer Fachhochschule, Oberseestrasse 10, 8640 Rapperswil, hansruedi.treppe@ost.ch

Abstract. Requirements Engineering bildet im Rahmen des Bachelor Wirtschaftsinformatik ein zentrales Modul, da bedeutsame Kompetenzen in der Schnittstelle der Fachabteilungen und der ICT entwickelt werden. In der Fallstudie «Alarmitis» geht es um die objektorientierte Anforderungsanalyse im Rahmen einer Software für ein Alarmsystem. Die aktuelle Fallstudie ist seit ca. 6 Jahren im Modul zum Thema Requirements Engineering im Hauptstudium (3./4. Semester) des B.Sc. Wirtschaftsinformatik im Einsatz und wurde dabei fortlaufend erweitert bzw. aktualisiert. Die Studierenden schätzen den Praxisbezug und die Möglichkeit, die in den Vorlesungen dargelegte Theorie und unterschiedlichen Methoden an einem konkreten Fall anzuwenden.

Inhaltsverzeichnis

Abbildungsverzeichnis	3
Tabellenverzeichnis	3
Lösungsvorschläge zu den Aufgaben 4.1 bis 4.13 aus der Fallstudie.....	4
4.1 Die Softwareproduktevision erstellen.....	4
4.2 Die Stakeholderanalyse vornehmen.....	5
4.3 Die Ermittlung (Erhebung) planen.....	7
4.4 Die nicht-funktionalen Anforderungen spezifizieren	7
4.5 Die funktionalen Anforderungen spezifizieren.....	8
4.6 Das Anwendungsfallmodell erstellen	11
4.7 User Stories formulieren	11
4.8 Einen Demonstrationsprototypen (UI-Wireframe) erstellen.....	12
4.9 Die objektorientierte Informationsanalyse vornehmen.....	14
4.10 Die Objektinteraktion modellieren	16
4.11 Die Funktionsanalyse vornehmen.....	17
4.12 Die Verhaltensanalyse vornehmen	18
4.13 Den Prüfplan für die RE-Artefakte erstellen.....	18

Abbildungsverzeichnis

Abbildung 1: Kontextdiagramm.....	4
Abbildung 2: Use Case Diagramm.....	11
Abbildung 3: UI-Wireframe Einstieg ins System	12
Abbildung 4: UI-Wireframe aktivierter Alarm	13
Abbildung 5: UI-Wireframe Mitteilung senden	13
Abbildung 6: UI-Wireframe Systemereignisse	14
Abbildung 7: Business Object Diagram.....	14
Abbildung 8: Fachklassendiagramm	15
Abbildung 9: Sequenzdiagramm.....	16
Abbildung 10: Aktivitätsdiagramm.....	17
Abbildung 11. Zustandsdiagramm	18

Tabellenverzeichnis

Tabelle 1: Stakeholdertabelle	5
Tabelle 2: Ermittlungstabelle	7
Tabelle 3: Funktionale Anforderungen	8
Tabelle 4: Prüfplan	19

Lösungsvorschläge zu den Aufgaben 4.1 bis 4.13 aus der Fallstudie

Diese Lösungsvorschläge zeigen eine gültige Lösungsvariante für die Aufgabenstellung «Case Study Alarmitis» und geben Hinweise zur Bewertung. Die Natur der Aufgabenstellung lässt oft mehrere gültige Lösungsvarianten zu.

4.1 Die Softwareproduktvision erstellen

Produktebeschreibung

Die "Alarmitis AG" vertreibt Alarmsysteme. Das geplante Softwareprodukt "Alarmsteuerung+" kontrolliert die unterschiedlichen, marktüblichen Komponenten des Alarmsystems. Es ermöglicht die unmittelbare elektronische Kommunikation zwischen allen Beteiligten. Alarmauslösungen und Deaktivierung erfolgt sicher und ohne Zeitverzögerung. Das System notifiziert zeitnah alle registrierten Personen über alle relevanten Events.

Kontextabgrenzung mittels einem Kontextdiagramm

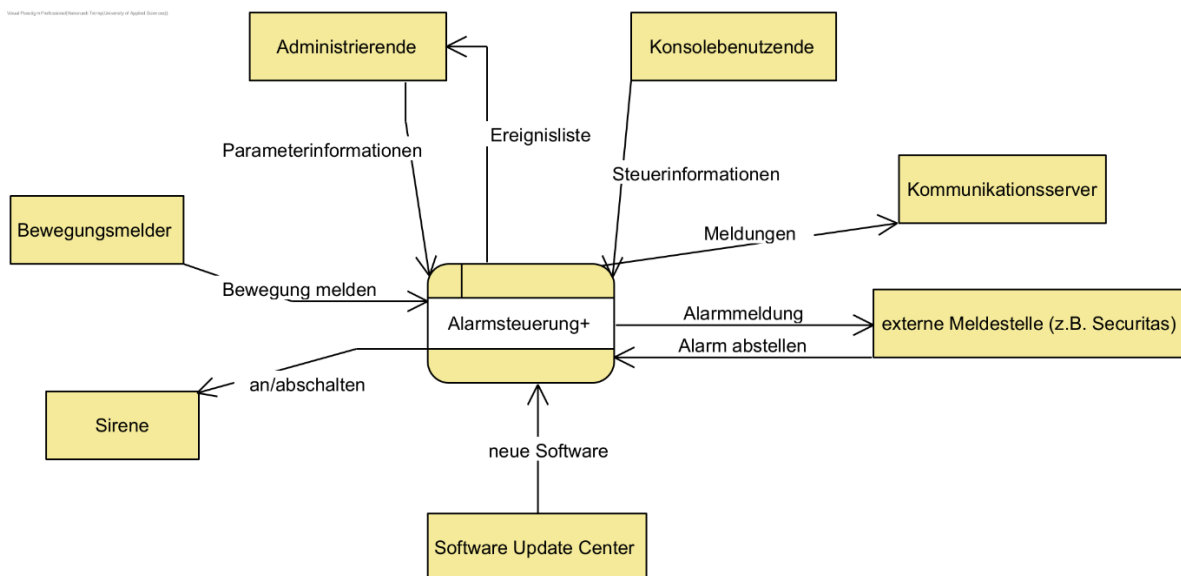


Abbildung 1: Kontextdiagramm

Motivation und Nutzen

Für den Anbieter

- Die neuen Alarmsystemkomponenten lassen sich problemlos betreiben.
- Eklatante Sicherheitslücken sind geschlossen.
- Mit dem neuen Produkt gibt es ein attraktives und wettbewerbsfähiges Angebot auf dem Markt.

Für den Kunden

- Alle handelsüblichen Komponenten eines Alarmsystems lassen sich flexibel kombinieren.
- Die Betreiber können sich auf die Zuverlässigkeit und Sicherheit der Alarmsteuerung verlassen.

Ziele

- Das neue System kann alle neuen Systemkomponenten einwandfrei (mittlere Ausfallzeit < 5 min., max. 1 Ausfall pro Jahr) steuern.

- Das neue System ist gegen aktuelle und zukünftige Cyberangriffe geschützt, bzw. weist eine Architektur auf, mittels welcher heute noch unbekannte Gefahren gelöst werden können.
- Das neue System muss in einem ersten produktiven Release in 9 Monaten auslieferbar sein.

4.2 Die Stakeholderanalyse vornehmen

Stakeholdertabelle

Tabelle 1: Stakeholdertabelle

Stakeholder Bezeichnung	Funktion im Projekt	Wissen in Bezug zu Geschäftsprozess (0-10)	Wissen in Bezug auf Technologie (0-10)	Macht/Interesse (0-10)	Als Requirements-Provider berücksichtigen (ja/nein)
Verantwortliche für Alarmanlage	Anwender	10	1	10/10	ja
Händler	Verkauft und installiert später die Lösungen	8	6	8/10	ja
Sicherheitsexperte	Berater	4	8	8/5	ja
Management	Geldgeber	4	3	7/10	nein
Private externe Meldestelle (Securitas)	Beratend, ev. später Anwender	8	6	8/8	ja
Öffentl. externe Meldestelle (Polizei)	Beratend, ev. später Anwender	8	4	6/8	ja
Entwickler	Implementierer	2	10	10/8	ja
Systembetreibende	Beratend, später techn. Betreiber	0	9	8/10	ja
Mitarbeitende	Anwender	6	0	6/8	ja
Geschäftspartner	ev. Betroffene bei Besuch, Reputation	2	2	8/8	nein
Staat	Compliance-Vorgaben	0	0	5/4	nein
Konkurrenz	Hinweise auf Basis-/Leistungsfeatures	0	10	4/4	nein

Ev. weitere relevante organisationsexterne Stakeholder:

- Sicherheitspartner: private Sicherheitsorganisation (z.B. Securitas) oder Polizei
 - Wie kann die Kommunikation optimal gestaltet werden?

- Wie sieht der Ablauf eines Alarmeinsatzes vor Ort aus?
- Nachbarn:
 - Wie kann auf Lärmemissionen bei der Auslösung eines Alarms vorbereitet und die Akzeptanz erhöht werden?
- Versicherung
 - Welche Massnahmen werden für optimale Bedingungen für die Versicherungspolice gefordert?
- IT-Security-Partner
 - Welche Gesetze müssen eingehalten werden?
 - Welche Normen bzw. Best Practices kommen zum Einsatz?
- Konkurrenz von Alarmanlagenanbieter
 - Was sind erwartete Features (d.h. was ist Commodity)?
 - Was sind aktuelle Trends in der Branche?

4.3 Die Ermittlung (Erhebung) planen

Tabelle 2: Ermittlungstabelle

Nr.	Ermittlungstechnik	Inhalt	Rollen/Stakeholder
1	Feldbeobachtung	ersten Eindruck erhalten, Begriffe klären	Mitarbeitende geben vor Ort Auskunft
2	Pers. Interview	Prozessablauf, Anforderungen aus Business-Sicht	Verantw. Alarmanlage Key Mitarbeitende
3	Workshop	Sicherheits- und Technische Anforderungen	Systembetreiber, Entwickler, IT-Security, Securitas
4	Pers. Interview	Ergänz. Anforderungen aus Sicherheits- und rechtlicher Sicht	Polizei
5	Tel. Interview	Ergänzende Anforderungen aus Sicht des Verkaufs und der Installation	ausgewählter Händler
6	Workshop	Anforderungsabstimmung, -Ergänzung	Verantw. Alarmanlage, Securitas, Software-Entwicklung

Der Requirements Engineer ist immer dabei, er führt die jeweiligen Ermittlungstechniken durch

4.4 Die nicht-funktionalen Anforderungen spezifizieren

Wichtige NFA für das Alarmsystem:

Performance

Die Reaktionszeit des Systems muss in mehr als 90 % der Fälle unter 1 sec. sein.

Rang: 3

Usability

Die Human-Interaction muss auf Anhieb von jedem Mitarbeitenden korrekt bedienbar sein.

Fehlbedienungen dürfen nicht zu einem Systemabsturz führen.

Die Benutzeroberfläche muss internationalen GUI-Standards entsprechen.

Rang: 4

Reliability

Das System muss zu mehr als 98 % verfügbar sein.

Ein Systemausfall darf max. 1 Std. betragen.

Rang: 2

Security

Der Transport der HTTP-Verbindungen muss immer verschlüsselt sein mittels einem Extended Validation SSL-Zertifikat.

Alle Systemaktivitäten müssen geloggt und somit nachvollziehbar sein.

Die persönliche Anmeldung am System muss über eine 2-Phasen-Authentifizierung erfolgen.

Rang: 1

Maintainability

Die Software muss modular und nach OO-Prinzipien entwickelt sein.

Der Code muss vereinbarten Coding-Guidelines entsprechen.

Rang: 6

Portability

Die Software muss auf Windows oder Linux-Server installierbar sein.

Rang: 7

4.5 Die funktionalen Anforderungen spezifizieren

Tabelle 3: Funktionale Anforderungen

Id	Bezeichnung	Kurzbeschreibung	Begründung	Owner	Verbind.	Krit.
1	Kommunikation mit Bewegungsmelder	Die Schnittstelle der am meisten verwendeten Bewegungsmelder muss bedient werden können. Melder an- und abstellen, Bewegungsmeldungen entgegennehmen.	Bewegungsmelder sind integraler Bestandteil jeder Alarmlösung	Verantw. Alarmsystem	m	5
2	Kommunikation mit Sirenen	Die Schnittstelle der am meisten verwendeten Sirenen muss bedient werden können. Sirene an- und abstellen.	Sirenen sind integraler Bestandteil jeder Alarmlösung	Verantw. Alarmsystem	m	5
3	Schnittstelle zu Kommunikations-server	Die Schnittstelle muss die notwendigen Protokolle unterstützen, um SMS und/oder E-Mails zu versenden.	Meldungen an beteiligte Personen gehören zu den Grundfunktionen einer Alarmlösung	Verantw. Alarmsystem	m	4

4	Schnittstelle zu ext. Meldestelle	Die Schnittstelle muss die notwendigen Protokolle unterstützen, um zuverlässig und sicher mit den wichtigsten externen Meldestellen wie z.B. Securitas, Polizei usw. zu kommunizieren. Wichtigste Meldungen: Alarm auslösen, Fehlalarm anzeigen, Alarm abstellen	Die Verbindung mit externen Stellen ist eine der wichtigsten Funktionen jeder Alarmlösung	Verantw. Alarmsystem	m	5
5	Schnittstelle zu Software Update Server	Über diese Schnittstelle kann bei neuen Updates der Server instruiert werden, damit zu einem günstigen Zeitpunkt (wenn der Grundmodus inaktiv ist) das Update automatisiert vorgenommen werden kann.	Die Software muss aus Sicherheitsüberlegungen immer möglichst aktuell sein.	Entwickler	can	2
6	Grundmodus inaktiv	In diesem Modus ist die Alarmanlage nicht aktiviert, kann aber Meldungen empfangen. Über den Touch-Point kann das aktivierte System einfach und sicher in den inaktiven Modus gebracht werden.	Basisfunktion	Verantw. Alarmsystem	m	4
7	Grundmodus aktiviert	Die Alarmanlage muss über den Touch-Point einfach und sicher aktiviert werden können. Im aktiven Modus reagiert das System auf die Bewegungsmelder.	Basisfunktion	Verantw. Alarmsystem	m	5
8	Touch Point Bedienung mit Tastatur	Die Codes zur Aktivierung und Deaktivierung können über die Tastatur vorgenommen werden.	Basisfunktion	Entwickler	m	5
9	Touch Point Bedienung mit Badge	Ein speziell ausgegebener Badge kann hingehalten werden zur Veränderung des Grundmodus.	Badge vereinfacht die Bedienung.	Entwickler	can	3

10	Touch Point Bedienung mit biometrischen Merkmalen	z.B. Fingerprint, Iris, Gesichtserkennung. Es muss eine Lernphase vorgesehen werden.	Vereinfacht und sichert die Bedienung.	Entwickler	nc	2
11	Alarmauslösung	Im aktivierten Modus muss gemäss Prozesseinstellung der Alarm aktiviert werden.	Basisfunktion	Verantw. Alarmsystem	m	5
12	Alarmdeaktivierung	Über Touch Point Eingabe oder Meldung einer externen Meldestelle wird der Alarm deaktiviert.	Basisfunktion	Verantw. Alarmsystem	m	5
13	Meldung wichtiger Ereignisse über SMS	Wichtige Ereignisse (wie z.B. Alarmauslösung, Deaktivierung Alarm) werden via SMS an verantwortliche Personen gesendet.	Unmittelbarer Informationsfluss ist wichtig	Verantw. Alarmsystem	m	3
14	Meldung wichtiger Ereignisse über andere Kanäle wie z.B. E-Mail, WhatsApp	Wichtige Ereignisse (wie z.B. Alarmauslösung, Deaktivierung Alarm) werden via E-Mail an verantwortliche Personen gesendet.	Informationsfluss über beliebig einstellbare Kommunikationskanäle ist wichtig	Verantw. Alarmsystem	nc	2
15	Zugang über Webbrowser für Administratoren	Die Administratoren können über einen Internetbrowser (die aktuell wichtigsten) die Parameter verwalten und alle Events einsehen.	Basisfunktion	Verantw. Alarmsystem	m	4

Verbindlichkeit: m = must; can; nc = nice to have

4.6 Das Anwendungsfallmodell erstellen

UML Use Case Diagramm

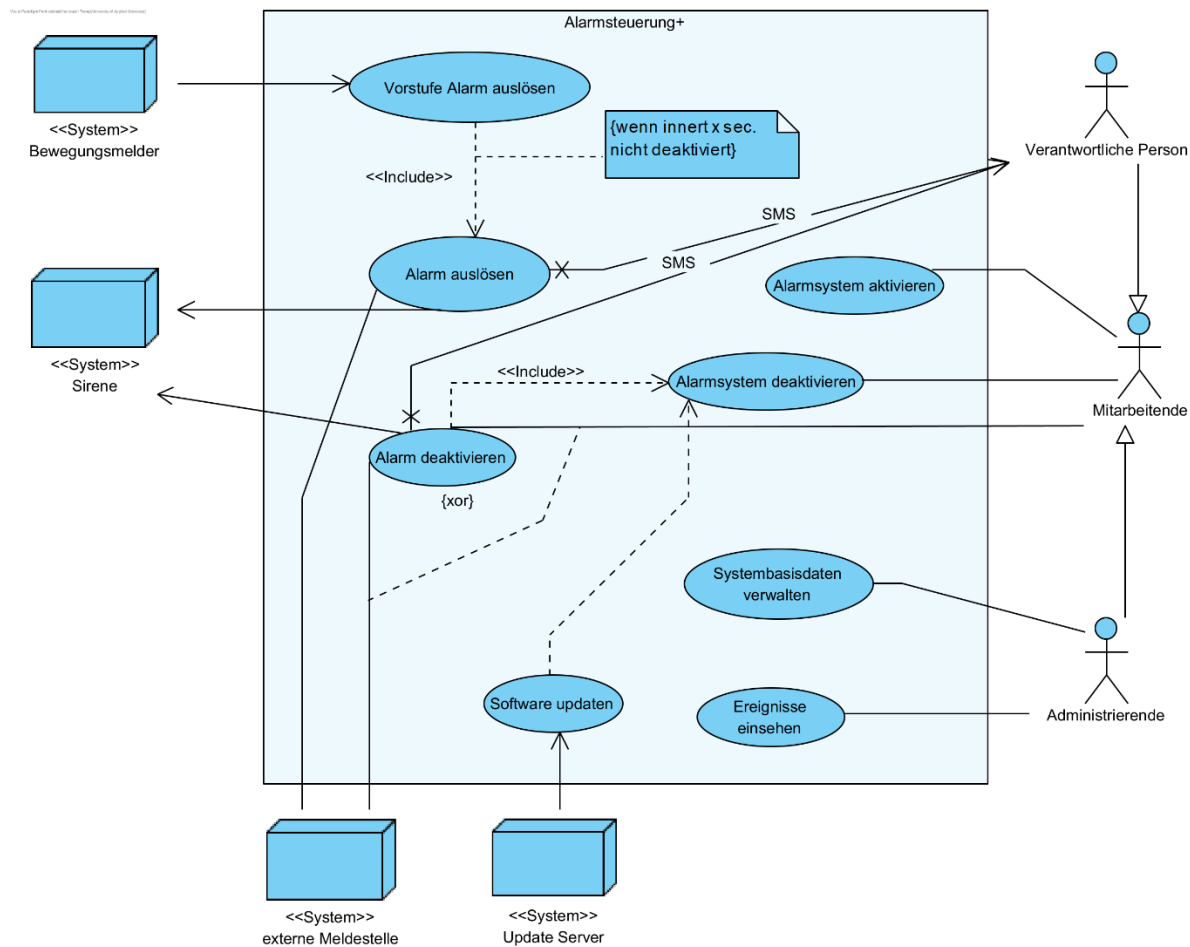


Abbildung 2: Use Case Diagramm

4.7 User Stories formulieren

User Story Nr. 1

Jemand unbekannter löst durch den Bewegungsmelder einen Alarm aus, welcher nach der vordefinierten Zeit nicht deaktiviert wird, damit ein Einbruch verhindert wird.

Refinement:

Das System führt nacheinander bzw. wo möglich parallel folgende Aktivitäten aus:

- Das System löst die Vorstufe des Alarms aus
- Nach einer gemäss Einstellung vorgegebenen Zeit löst das System den Alarm aus
 - sofern keine Deaktivierung der Vorstufe erfolgt ist
- Das System aktiviert nun alle angeschlossenen Sirenen damit die Umgebung alarmiert ist.
- Gemäss Einstellungen kommuniziert das System mit der externen Meldestelle und löst dort den Alarm aus, damit die Securitas/Polizei aktiv werden kann.
- Gemäss Einstellungen sendet das System SMS (andere Meldungen) an die verantwortlichen Personen, damit die entsprechenden Personen gem. Notfallplan handeln können.

User Story Nr. 2

Der Mitarbeitende der externen Meldestelle deaktiviert einen ausgelösten Alarm über die dafür vorgesehene Schnittstelle, damit der aktivierte Alarm kontrolliert beendet wird.

Refinement:

Das System führt nacheinander oder parallel folgende Aktivitäten aus:

- Das System deaktiviert alle angeschlossenen Sirenen
- Gemäss Einstellungen sendet das System SMS (andere Meldungen) an die verantwortlichen Personen, damit diese informiert sind.
- Das System deaktiviert das Alarmsystem

4.8 Einen Demonstrationsprototypen (UI-Wireframe) erstellen

UI-Wireframe erstellt mit balsamiq (<https://balsamiq.com/>):

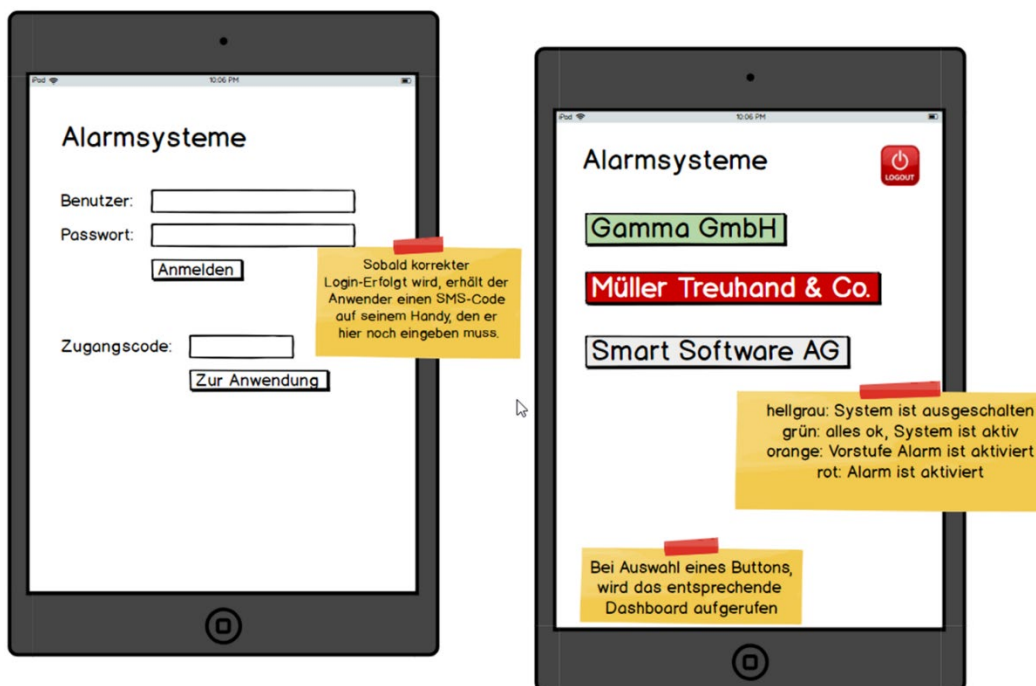


Abbildung 3: UI-Wireframe Einstieg ins System



Abbildung 4: UI-Wireframe aktivierter Alarm

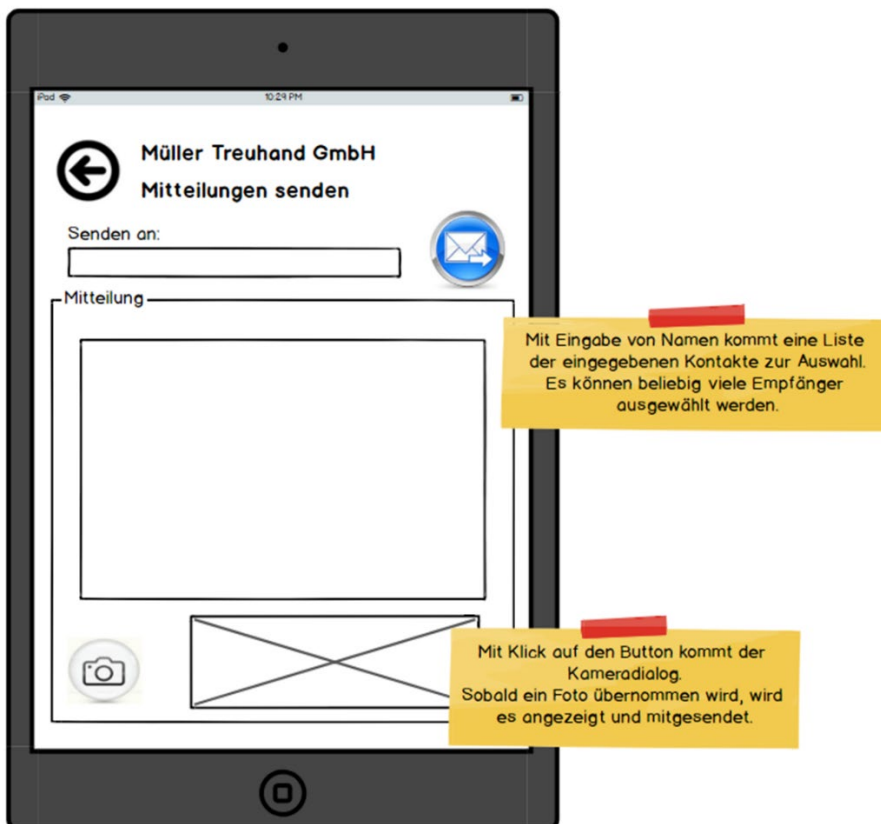


Abbildung 5: UI-Wireframe Mitteilung senden

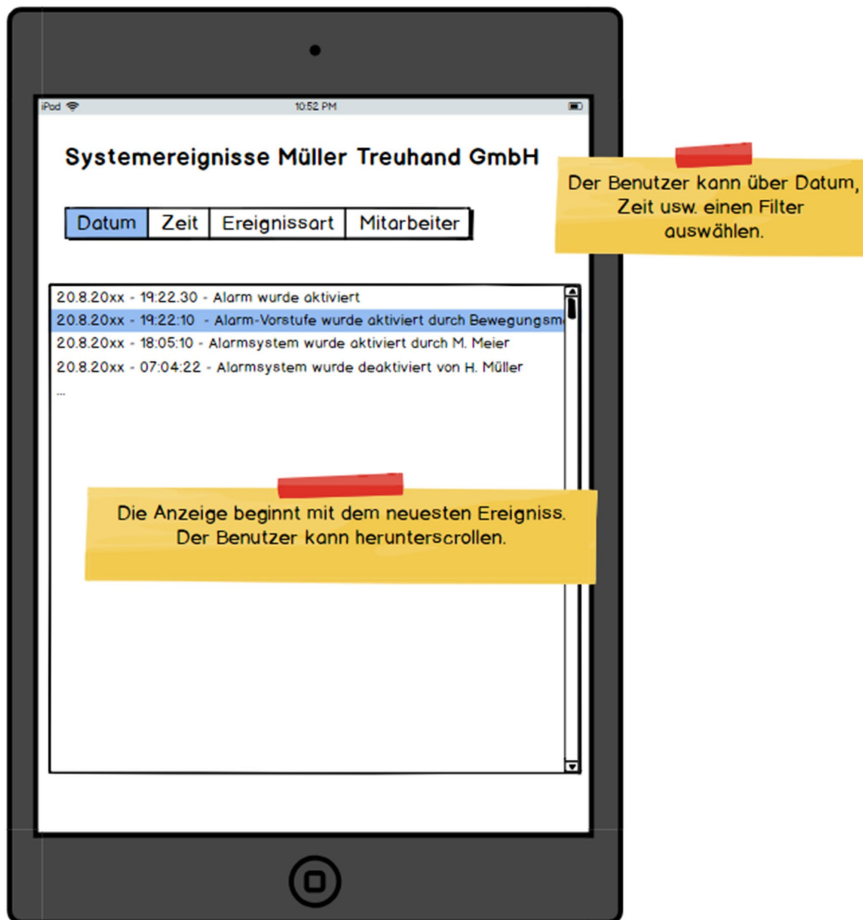


Abbildung 6: UI-Wireframe Systemereignisse

4.9 Die objektorientierte Informationsanalyse vornehmen

Business Object Diagram

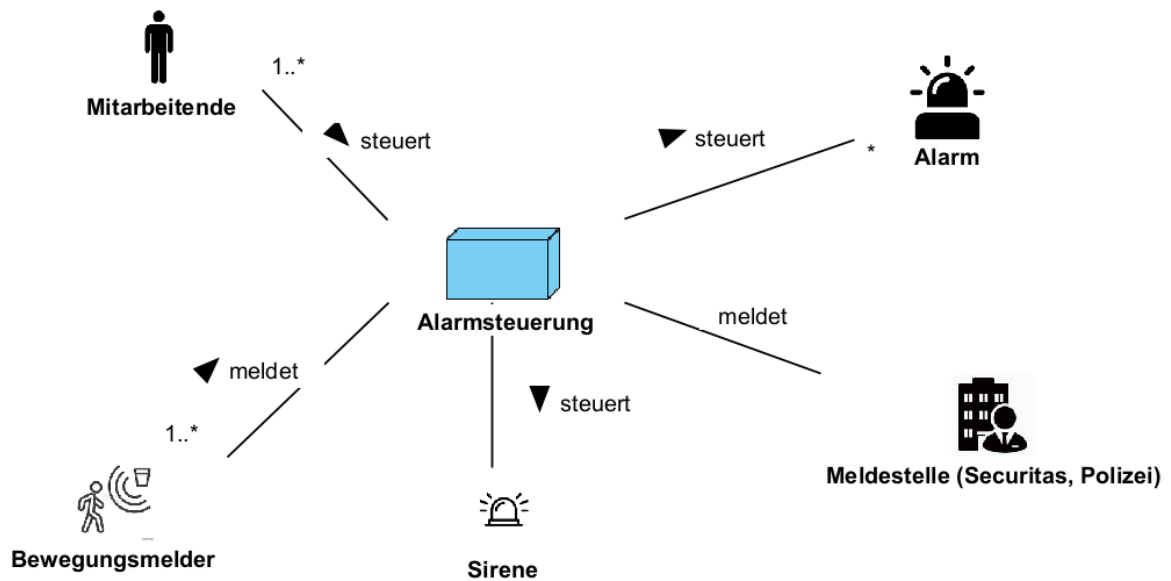


Abbildung 7: Business Object Diagram

4.10 Die Objektinteraktion modellieren

UML Sequenzdiagramm

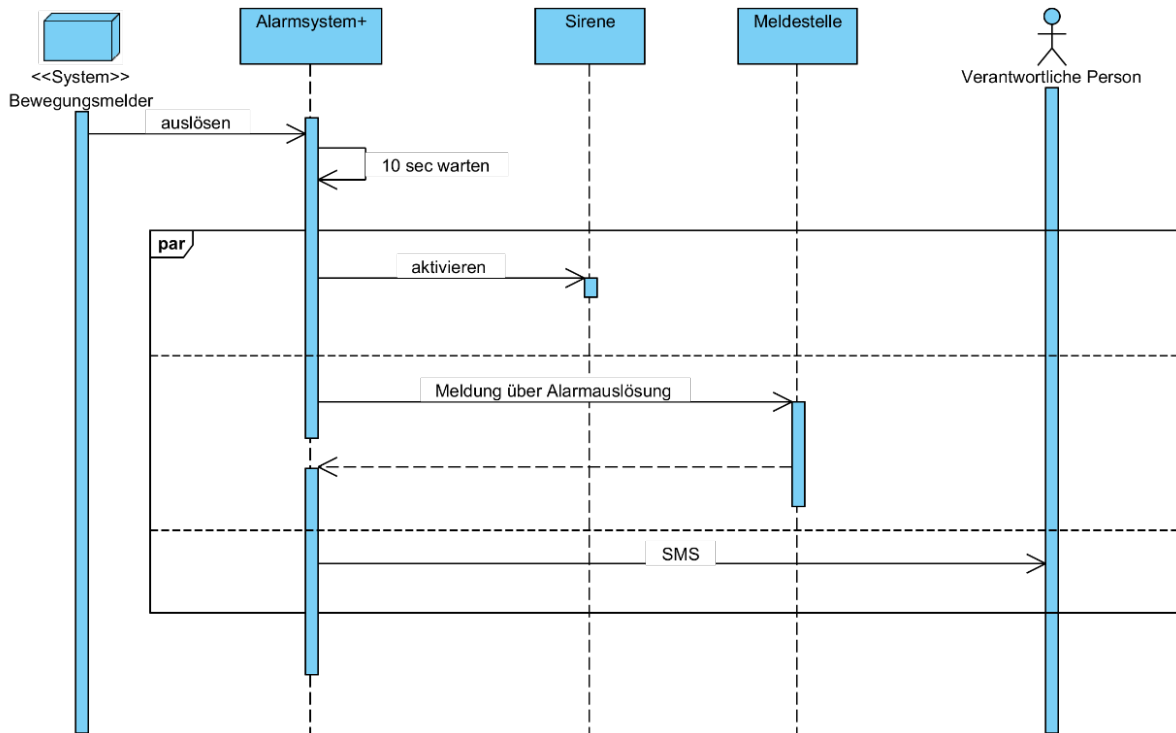


Abbildung 9: Sequenzdiagramm

4.11 Die Funktionsanalyse vornehmen

UML-Aktivitätsdiagramm

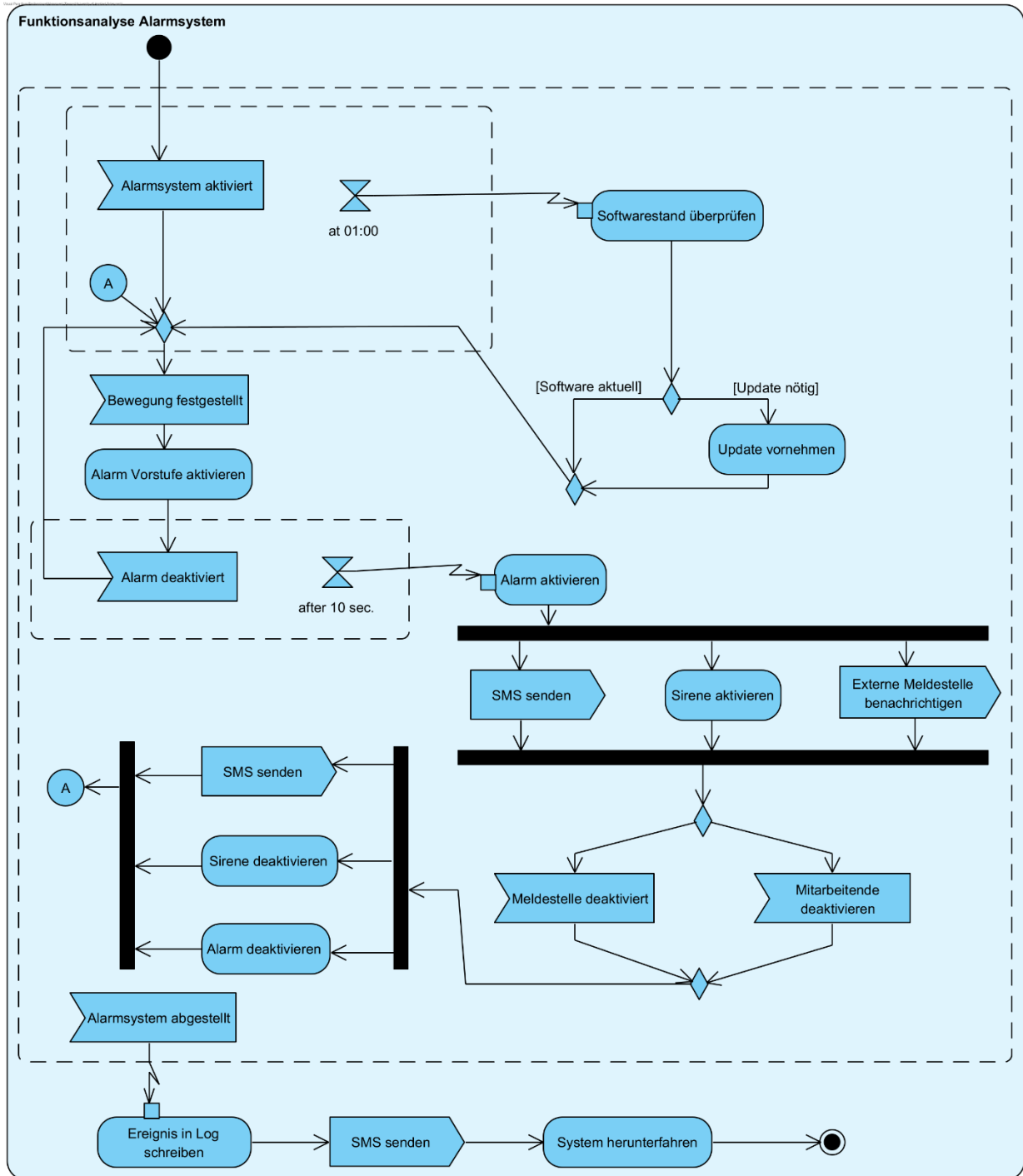


Abbildung 10: Aktivitätsdiagramm

4.12 Die Verhaltensanalyse vornehmen

UML-Zustandsdiagramm

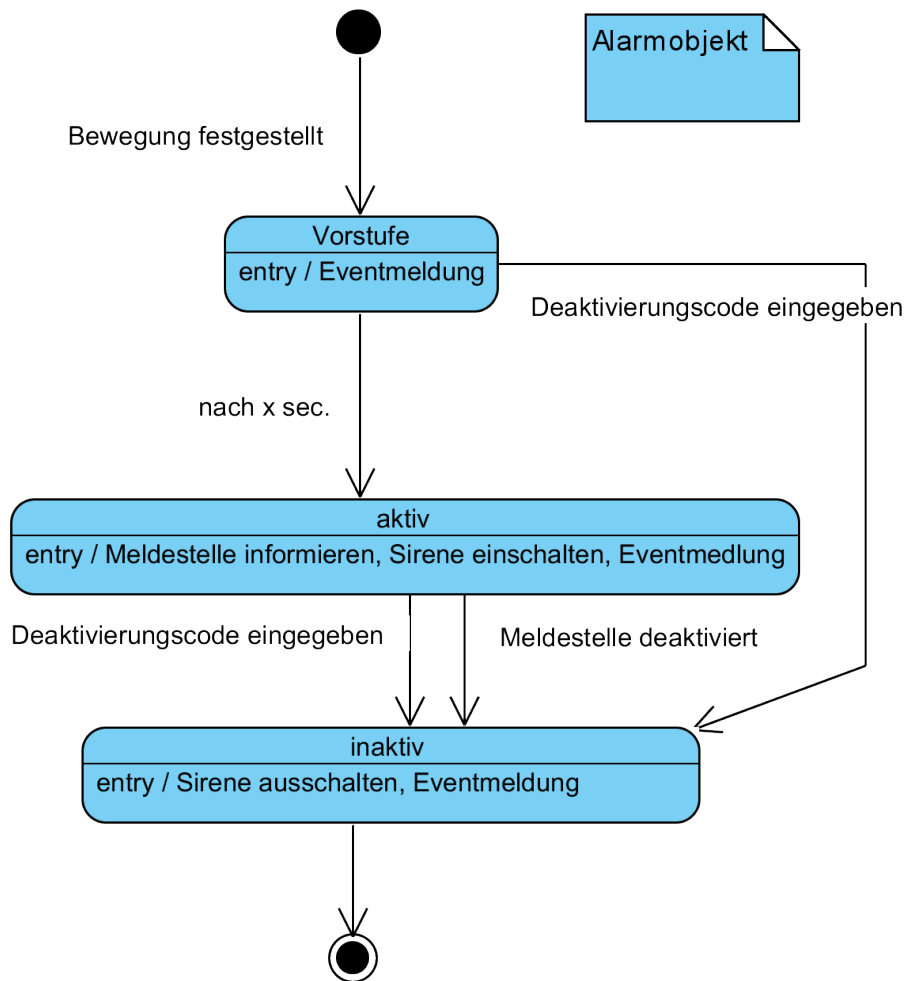


Abbildung 11. Zustandsdiagramm

4.13 Den Prüfplan für die RE-Artefakte erstellen

Prüfobjekte

- Anforderungsspezifikationen:
 - Funktionale Anforderungen, Use Cases, Epics, User-Stories
 - NFA
- UML-Modelle:
 - Use-Case Diagramm
 - Fachklassendiagramm
 - Aktivitätsdiagramm der gesamten Funktionalität
 - Zustandsdiagramm des Alarms
- Demonstrationsprototyp (UI-Wireframe)

Zeitplan

Der Requirements-Engineer ist in allen Prüfungen beteiligt.

Tabelle 4: Prüfplan

Prüfobjekt	Prüfmethode	Beteiligte Mitarbeitende X = verantwortliche Person	Termin
Epics, User Stories	Review	PO (Product Owner) X Fachpersonen	...
NFA	Review	PO X IT-Betrieb SW-Entwicklung SW-Architektur IT-Security	
Use-Case Diagramm	Walkthrough	PO Requirements Engineer X SW-Entwicklung	
Fachklassendiagramm	Walkthrough	SW-Entwicklung System Analyst X	
Aktivitätsdiagramm	Walkthrough	PO SW-Entwicklung Senior Softwareentwickler X	
Zustandsdiagramm	Walkthrough	PO SW-Entwicklung Senior Softwareentwickler X	
UI-Wireframe	Usability Review	PO GUI-Engineer UX Designer X	

Die Begründungen können sich stichwortartig auf die Ausgangslage oder das allgemeine Verständnis im Requirements Engineering beziehen.