

Referenzprofil

IT Systems Developer

Oliver Boehm

Dieses Referenzprofil wurde im Rahmen des bmb+f geförderten Projekts „Arbeitsprozess-orientierte Weiterbildung in der IT-Branche“ erarbeitet von:

Danksagung

Diese Profilbeschreibung entstand auf der Grundlage des Praxisprojekts „Die Teltra Televisite“ der *Gesellschaft für telematische Traumatologie mbH* in Bochum. Wir danken herzlich Herrn Dr. med. Bernhard Clasbrummel und Herrn Dipl.-Ing. Karsten Biskup für das Zustandekommen der Zusammenarbeit sowie Frau Anette Gerboth, Herrn Jörg Orlowski und Herrn Roman Engelbarts für die fruchtbaren Gespräche und hilfreichen Anregungen bei der Ausarbeitung des Profils. Ohne ihre Mithilfe hätte dieses Dokument nicht entstehen können.

Inhalt

1	EINFÜHRUNG: REFERENZPROZESSE ALS CURRICULA	4
1.1	EREIGNIS-PROZESS-KETTEN: SYMBOLIK	4
1.2	REFERENZPROZESS UND TEILPROZESSE	6
2	DAS PROFIL: IT SYSTEMS DEVELOPER (IT-SYSTEMPLANER/IN).....	9
2.1	TÄTIGKEITSBESCHREIBUNG	9
2.2	PROFILTYPISCHE ARBEITSPROZESSE	9
2.3	PROFILPRÄGENDE KOMPETENZFELDER.....	10
2.4	QUALIFIKATIONSERFORDERNISSE.....	11
2.5	EINORDNUNG INS SYSTEM UND KARRIEREPFADE	11
3	REFERENZPROZESS	12
3.1	SYSTEMENTWICKLUNG	12
3.1.1	Referenzprozess: Systementwicklung	14
3.1.2	Das Beispielprojekt: Die Teltra Televisite	16
3.1.3	Prozesskompass: Systementwicklung	17
3.1.4	Teilprozesse: Systementwicklung	17
3.1.4.1	Erstellen grober Systemarchitekturen	18
3.1.4.2	Suchen nach Referenzen	22
3.1.4.3	Begleiten der Prototyperstellung	26
3.1.4.4	Testen der Prototypen	30
3.1.4.5	Auswählen der zu realisierenden Lösungsskizze mit dem Kunden	33
3.1.4.6	Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte	35
3.1.4.7	Prüfen von hauseigenen Fertigprodukten	37
3.1.4.8	Bewerten von existierenden Standards.....	39
3.1.4.9	Auswählen einzusetzender Fertigprodukte	41
3.1.4.10	Erstellen von technischen Lösungsvorschlägen	43
3.1.4.11	Zerlegen in Subsysteme.....	45
3.1.4.12	Zerlegen in SW-/HW-Komponenten	47
3.1.4.13	Erstellen des Systemdesigns	49
3.1.4.14	Abgleichen von Kundenanforderungen und Systemdesign	53
3.1.4.15	Untersuchen der Realisierbarkeit	55
3.1.4.16	Beschreiben der Schnittstellen	59
3.1.4.17	Spezifizieren der Integration und Systemtests	61
3.1.4.18	Definieren von zu erstellenden SW-/HW-Komponenten	65
3.1.4.19	Begleiten der SW-/HW-Komponentenerstellung	68
3.1.4.20	Spezifizieren notwendiger Anpassungen von Fertigprodukten	70
3.1.4.21	Veranlassen der Beschaffung	73
3.1.4.22	Begleiten der Prüfung und Anpassung von Fertigprodukten	75
3.1.4.23	Vorbereiten der Systemintegration.....	78
3.1.4.24	Durchführen der Systemintegration und Systemtests	81
3.1.4.25	Vorbereiten der Installation in der Kundenumgebung	84
3.1.4.26	Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien.....	86
3.1.4.27	Begleiten der Installation und des Abnahmetests	88
3.1.4.28	Übergeben an den IT Systems Administrator	90
3.1.4.29	Unterstützen von Schulungen	92
3.1.4.30	Sammeln von Verbesserungsvorschlägen.....	94

1 Einführung: Referenzprozesse als Curricula

Das Referenzprojekt des IT Systems Developer verdeutlicht paradigmatisch die diesem Tätigkeitsfeld zugrunde liegenden Arbeitsprozesse, die mit ihnen verbundenen Ansprüche sowie die daraus resultierenden Anforderungen an Inhalt und Durchführung einer qualitativ hochwertigen Weiterbildung.

Das Referenzprojekt erfüllt mehrere Funktionen:

Aus der Praxis für die Praxis

Als Abstraktion tatsächlich stattgefundener Projekte und Prozesse bieten die Referenzprozesse eine realistische und leicht nachvollziehbare Abbildung dessen, was die Tätigkeiten eines IT Systems Developer sind.

Prozessorientierung als innovatives „Curriculum“

Als vollständige Darstellung aller wichtigen Arbeitsprozesse sowie der dazugehörigen Qualifikationen, Tätigkeiten und Werkzeuge bieten die Referenzprozesse die Grundlage für die Weiterbildung zum IT Systems Developer. Alle diese Prozesse müssen – entsprechend den Vorgaben – einmal oder mehrmals durchlaufen werden und ermöglichen dadurch den Weiterzubildenden den arbeitsplatznahen, integrativen Erwerb von relevanten Kompetenzen. Durch den Verbleib im Arbeitsprozess wird nicht nur für die Weiterzubildenden eine hohe Motivation (Arbeit an echten Projekten/Aufgaben) und Nachhaltigkeit erreicht, sondern auch – aus Sicht des Unternehmens – die Kontinuität und Qualität der laufenden Arbeiten gesichert (keine Ausfallzeit durch Seminartage, kein mühsamer Transfer).

Qualitätsstandard für die Weiterbildung

Als Referenz bieten insbesondere die Teilprozesse und die mit ihnen verbundenen Tätigkeits- und Qualifikationsziele einen Qualitätsmaßstab für die arbeitsprozessorientierte Weiterbildung und die resultierenden Abschlüsse. Vollständige Transparenz und klare Zielvorgaben ermöglichen die qualitativ hochwertige Absicherung auch komplexer Kompetenzen sowie den systematischen Erwerb des notwendigen Erfahrungswissens.

Transferprozesse

Die Generalisierung des Referenzprojekts aus der Praxis und seine didaktische Anreicherung ermöglichen eine leichte Auswahl angemessener Transferprozesse, deren Bearbeitung die Grundlage der Weiterbildung ist. Transferprozesse sind reale Prozesse, die Referenzprojekte in einer lernförderlichen Umgebung abbilden. Abgeschlossene Transferprozesse auf Basis der hier dargestellten Anforderungen und Qualitätsmaßstäbe sind nicht nur Qualifikationsnachweis des Einzelnen, sondern bilden auch die Basis eines angemesseneren und zielgerichteteren Umgangs mit Geschäfts- und Arbeitsprozessen im Unternehmen.

1.1 Ereignis-Prozess-Ketten: Symbolik

Die Darstellung der Referenzprozesse in Form von Ereignis-Prozess-Ketten¹ ermöglicht einen schnellen Überblick. Vollständigkeit kann leicht überprüft werden, Anpassungen und Modifikationen im Hinblick auf das eigene Unternehmen sind problemlos möglich und Anknüpfungspunkte an andere Prozesse, aber auch zu weiter führenden Informationen ergeben sich automatisch.

Die bei der Darstellung der Referenz- und Teilprozesse verwendete Modellierungssprache stellt eine Anpassung und Weiterentwicklung der klassischen EPK-Modellierung dar:

¹ Vgl. A.-W. Scheer, *Wirtschaftsinformatik*, Springer 1998.

Referenz- wie Teilprozesse sind aus der Sicht des jeweiligen Spezialisten, also als Arbeitsprozesse einer Person dargestellt.
Referenz- wie Teilprozesse stellen in der Regel keinen Geschäftsprozess dar.

Die EPK-Symbole werden hier wie folgt verwendet:

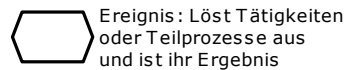
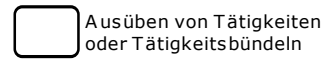
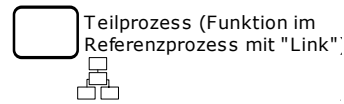


Abbildung 1: Grundlegende Symbole der Referenz- und Teilprozessmodelle.

Die wichtigsten Symbole sind:

- die Tätigkeiten bzw. Tätigkeitsbündel oder Teilprozesse, die mit dem Funktionssymbol dargestellt werden
- die Ereignisse, die Tätigkeiten bzw. Teilprozesse auslösen und Ergebnisse von Teilprozessen sind

Grundsätzlich gilt: Auf ein Ereignis folgt immer ein Teilprozess bzw. eine Tätigkeit. Ergebnisse von Tätigkeiten sind sehr oft Dokumente; diese werden dann zusätzlich durch das Dokumentsymbol dargestellt.

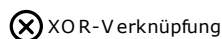
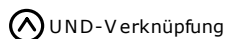


Abbildung 2 : Konnektoren.

Wenn Alternativmöglichkeiten bestehen, werden Ereignisse und Teilprozesse/Tätigkeiten über Konnektoren (AND, OR, XOR) verbunden. Dabei steht AND für ein verbindendes „Und“, OR für ein „Oder“, das alle Möglichkeiten offen lässt, und XOR für ein „ausschließendes Oder“, welches nur einen der angegebenen Pfade ermöglicht.

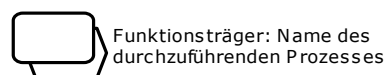


Abbildung 3: Schnittstelle.

Da die Prozesse aus der Sicht des jeweiligen Spezialisten formuliert werden, sind Schnittstellen zu Prozessen anderer Spezialisten oder zu Entscheidungsprozessen auf höherer Ebene notwendig. Dazu wird das Schnittstellensymbol verwendet. Es steht für Prozesse, die der Spezialist nicht selber durchführt, auf deren Durchführung er aber angewiesen ist. Parallel zu jeder Schnittstelle wird die Tätigkeit dargestellt, die der Spezialist selbst in diesem Zusammenhang ausübt, wie „Beraten bei ...“, „Unterstützen bei ...“ oder „Informieren des ...“.

Alle Prozesse werden durch die Verwendung dieser Symbole klar und einfach strukturiert dargestellt und sind offen für die Übertragung in konkrete Transferprozesse.

1.2 Referenzprozess und Teilprozesse

Der hier vorgestellte Referenzprozess und seine Teilprozesse stellen das Curriculum des Spezialistenprofils IT Systems Developer dar.

Der Referenzprozess erhebt nicht den Anspruch eines Vorgehensmodells, sondern bildet beispielhaft den möglichen Arbeitsprozess und Verlauf eines Projekts auf Spezialistenebene ab.

Er bildet die Grundlage für Weiterbildungen und damit einen Qualitäts-, Niveau- und Komplexitätsmaßstab. Die zugehörigen Teilprozesse sind hier beispielhaft modelliert und stellen eine Möglichkeit der Durchführung dar. Einzelheiten zu den unverzichtbaren Prozessen und Kompetenzfeldern sind im Referenzprojekt festgelegt. Die Reihenfolge und die Inhalte der Teilprozesse sind abhängig vom jeweils auszuwählenden Transferprojekt und werden in diesem Zusammenhang festgelegt.

Die Darstellung der Prozesse erfolgt systematisch:

Jeder Prozess wird mithilfe von Ereignis-Prozess-Ketten dargestellt. Einem auslösenden Ereignis folgt eine Funktion, die wiederum ein oder mehrere Ereignisse zum Ergebnis hat. Ereignisse und Funktionen können mit AND, OR oder XOR, den Konnektoren, verbunden sein.

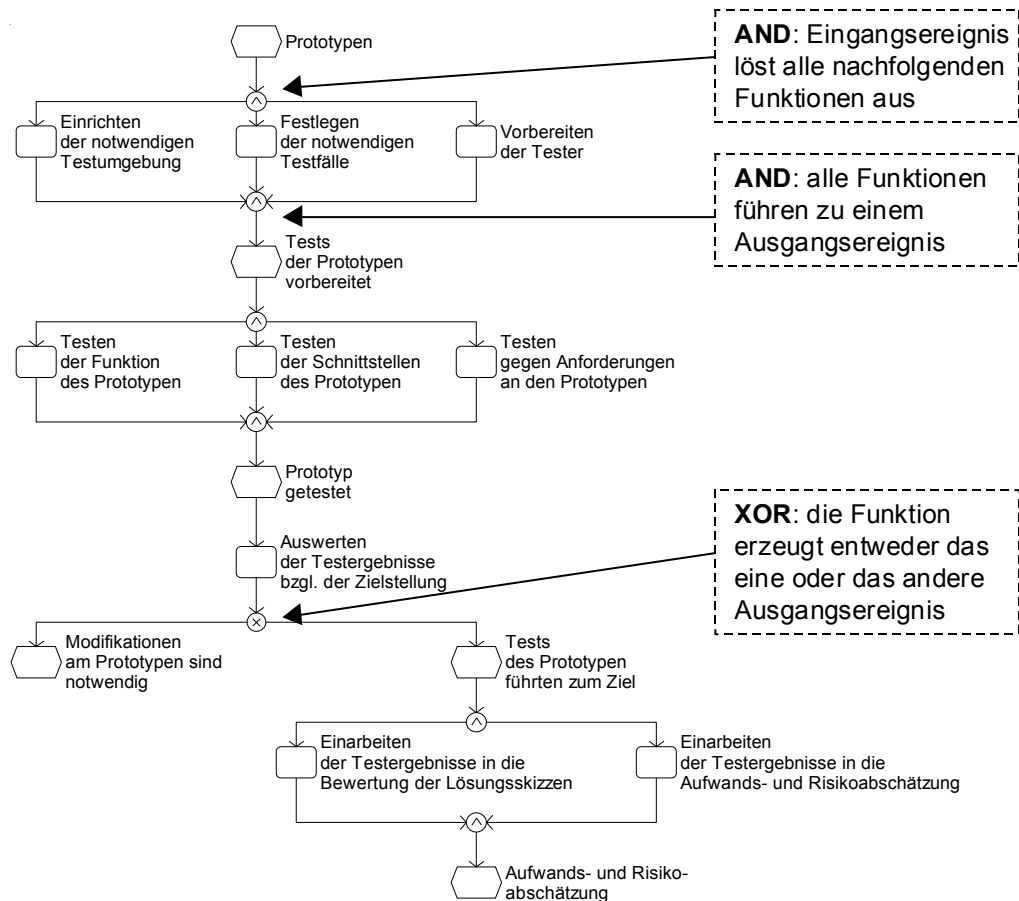


Abbildung 4: Beispielprozess (Teilprozess „Testen der Prototypen“) mit unterschiedlicher Verwendung von Konnektoren.

Die Verbindung von Referenzprozess und Teilprozessen erfolgt über die Funktionen des Referenzprozesses:

Jede Funktion im Referenzprozess steht für einen Teilprozess.

Ereignisse, die dem jeweiligen Teilprozess direkt vor- oder nachgeordnet sind, sind Anfangs- und Endereignisse der jeweiligen Teilprozesse. Damit stellen die Teilprozesse die Funktionen des Referenzprozesses ausführlich dar und ein Hin- und Herbewegen zwischen Referenz- und Teilprozessen ist jederzeit problemlos möglich.

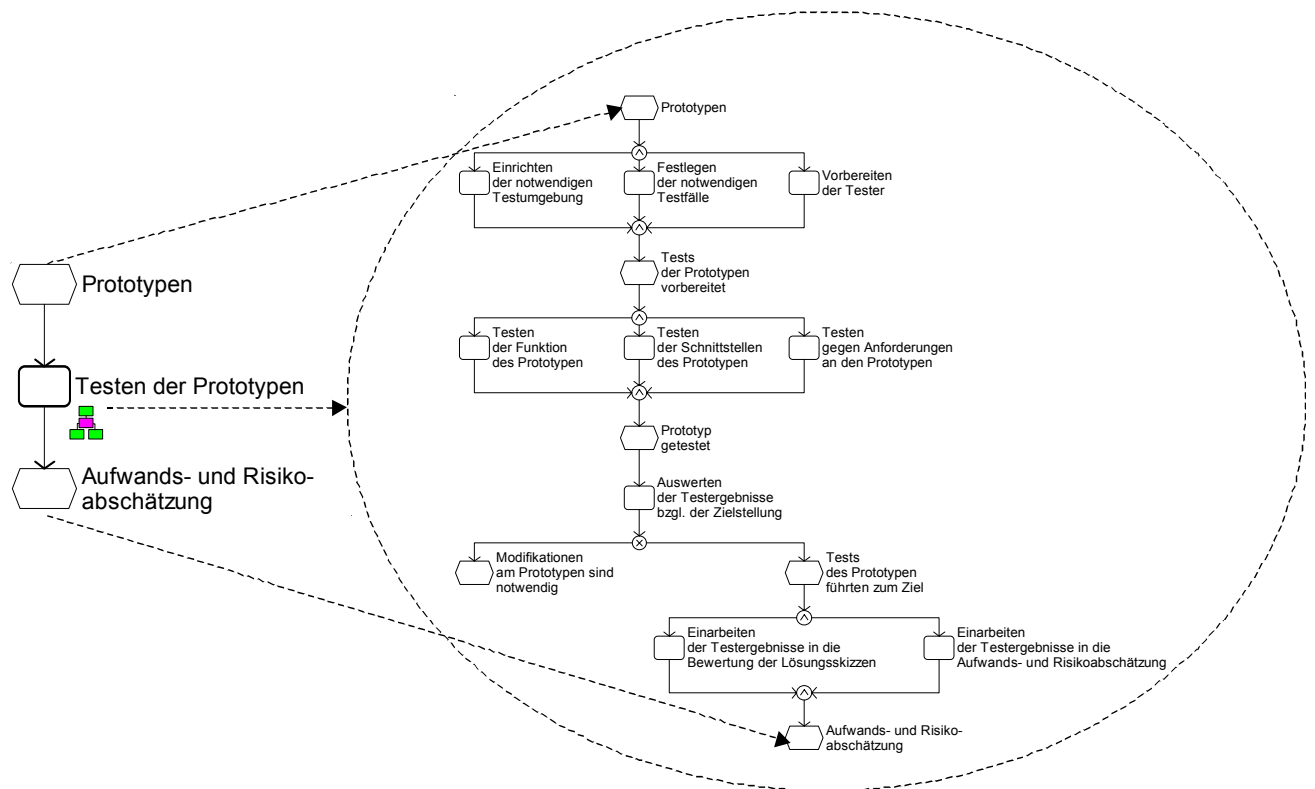


Abbildung 5: Ausschnitt aus dem Referenzprozess und Teilprozess „Testen der Prototypen“.

Die Teilprozesse stellen so die wesentlichen Teile eines Projekts dar und lassen sich entsprechend auf Transferprojekte übertragen. Den Teilprozessen sind die jeweils wesentlichen Tätigkeiten und Kompetenzfelder zugeordnet.

2 Das Profil: IT Systems Developer (IT-Systemplaner/in)

IT Systems Developer² planen IT-Systeme, beschreiben sie durch ein formalisiertes Systemdesign und begleiten ihre Umsetzung.

2.1 Tätigkeitsbeschreibung

IT Systems Developer planen auf der Basis existierender Anforderungsmodelle die Neuentwicklung von IT-Systemen bzw. die Weiterentwicklung bestehender IT-Systeme und die Einbindung von IT-Systemen in die Infrastruktur des Kunden. Dies schließt die Konzeption der Systemarchitekturen unter Berücksichtigung der Funktionalitäten und Schnittstellen der benötigten Komponenten mit ein. Sie arbeiten Nutzungsszenarien aus und analysieren potenzielle Schwachstellen.

Die zentrale Tätigkeit des IT Systems Developer ist die Aufbereitung von IT-Systemen in Form von (eher statischen) Komponentendiagrammen und (dynamischen) Ablaufszenarien unter Zuhilfenahme von formalen Notationen. IT Systems Developer wirken darüber hinaus bei der Beschaffung bzw. Erstellung der benötigten Komponenten und deren Integration mit.

2.2 Profiltypische Arbeitsprozesse

Die im Folgenden beschriebenen Teilprozesse dokumentieren den gesamten profiltypischen Arbeitsprozess des IT Systems Developer. Die Beherrschung dieses Arbeitsprozesses in Verbindung mit den Kompetenzen in den jeweiligen Kompetenzfeldern und der Berufserfahrung bildet die Grundlage für die berufliche Handlungskompetenz.

1. Erarbeiten von bedarfsgerechten Lösungskonzepten auf der Basis von Anforderungsmodellen in Zusammenarbeit mit IT-Systemanalytikern und Kunden
2. Erstellen von groben Systemarchitekturen zur Sicherstellung der Realisierbarkeit und zur Abschätzung der zu erwartenden Aufwände; Unterstützen der Angebotserstellung durch die Identifikation von möglichen Schwachstellen und Risiken, z. B. durch die Erstellung von „Explorations“-Prototypen
3. Erarbeiten von Lösungsvorschlägen für die Integration und technische Anbindung vorhandener Systeme – z. B. Warenwirtschaftssysteme oder Datenbank-Server; Analysieren und Testen bestehender Schnittstellen unter Berücksichtigung von Performance und Sicherheit
4. Erstellen von Systemdesign-Dokumenten zur formalisierten Spezifikation des statischen Aufbaus und des dynamischen Verhaltens von Systemarchitekturen und der vorzunehmenden Systemerweiterungen; Dekomposition von IT-Systemen in interagierende Komponenten
5. Spezifizieren der softwaretechnischen Anforderungen an die Datenhaltungskomponenten des Systems und die einzusetzenden Kommunikationsprotokolle als Basis für die interne Softwareentwicklung und für extern zu beziehende Komponenten
6. Prüfen und Bewerten von Angeboten für extern zu beziehende Komponenten; Bewerten von existierenden Standards und existierenden Softwarekomponenten bezüglich ihrer Eignung

² Kapitel 2: „Das Profil: Industrial IT Systems Developer (IT-Systemplaner/in)“ gibt – mit Ausnahme des Abschnitts 2.5 „Einordnung in das System und Karrierepfade“ – den offiziellen Text der „Vereinbarung über die Spezialistenprofile im Rahmen des Verfahrens zur Ordnung der IT-Weiterbildung“ vom 25.05.2002 (Bundesanzeiger 105, ausgegeben am 12.06.2002) wieder.

7. Planen und Initiieren der beim Kunden für die Installation des Systems durchzuführenden Vorarbeiten (technische Arbeiten wie Datenkonvertierungen, organisatorische Arbeiten wie Mitarbeiterschulungen)
8. Mitarbeiten im Change Control Board zur Beurteilung der technischen Auswirkungen vorgeschlagener Änderungen
9. Begleiten der Umsetzung; Unterstützen der Systemintegration und Systemtests bzw. bei kleineren Projekten; Durchführen der Systemintegration mit Unterstützung der am Projekt beteiligten Entwickler
10. Übergeben des Systems an den Administrator
11. Mitarbeiten bei der Erstellung von Handbüchern, Installationsanleitungen und Trainingsmaterialien sowie Durchführen von Schulungen für das zuständige Personal (Betreuung und Wartung des Systems) beim Kunden
12. Sammeln und Bewerten von Vorschlägen für Erweiterungen und Verbesserungen der Systeme für die Konzeption der nächsten Releases

2.3 Profilprägende Kompetenzfelder

Die Beherrschung der profiltypischen Arbeitsprozesse setzt Kompetenzen unterschiedlicher Reichweite in den nachstehend aufgeführten beruflichen Kompetenzfeldern³ voraus. Den Kompetenzfeldern sind Wissen und Fähigkeiten sowie typische Methoden und Werkzeuge unterschiedlicher Breite und Tiefe zugeordnet.

Grundlegend zu beherrschende, gemeinsame Kompetenzfelder⁴:

- Unternehmensziele und Kundeninteressen
- Problemanalyse, -lösung
- Kommunikation, Präsentation
- Konflikterkennung, -lösung
- fremdsprachliche Kommunikation (englisch)
- Projektorganisation, -kooperation
- Zeitmanagement, Aufgabenplanung und -priorisierung
- wirtschaftliches Handeln
- Selbstlernen, Lernorganisation
- Innovationspotenziale
- Datenschutz, -sicherheit
- Dokumentation, -standards
- Qualitätssicherung

Fundiert zu beherrschende, gruppenspezifische Kompetenzfelder:

- Methoden und Werkzeuge der Softwareentwicklung
- Engineering-Prozesse
- System-Analyse

³ Die Kompetenzfelder werden in der nachfolgenden Auflistung jeweils durch ein zusammenfassendes Stichwort benannt. Da die Weiterbildung zum Spezialisten auf die erfolgreiche Bewältigung zunehmend offener beruflicher Handlungssituationen sowie ganzheitlichen Kompetenzerwerb abzielt, bildet der Kompetenzerwerb einen integralen Bestandteil der Arbeits- und Weiterbildungsprozesse und lässt sich nur im Zusammenhang mit diesen operationalisieren.

⁴ Jeder Spezialist muss in den in diesem Abschnitt genannten „weichen“ Kompetenzfeldern wie „Kommunikation, Präsentation“, „Konflikterkennung, -lösung“ usw. ein Niveau erreichen, dass über dem einer Fachkraft liegt. Das heißt, er muss auch in diesen Feldern zu eigenständigem Handeln in der Lage sein und zum Erreichen des Ziels in dem jeweiligen Feld gegebenenfalls über den Rahmen bekannter Verfahren und Lösungen hinausgehen können.

- Entwicklungsstandards (Leistungsfähigkeit, Sicherheit, Verfügbarkeit, Innovation)
- Qualitätsstandards
- Datenbanken, Netzwerke

Routiniert zu beherrschende, profilspezifische Kompetenzfelder:

- Designmethoden, -strategien, -muster
- Systemarchitekturen
- Methoden und Konzepte der Systemintegration und -anpassung
- Schnittstellen
- Projektmanagement

2.4 Qualifikationserfordernisse

Im Regelfall wird ein hinreichendes Qualifikationsniveau auf der Basis einschlägiger Berufsausbildung oder Berufserfahrung vorausgesetzt.

2.5 Einordnung ins System und Karrierepfade

Das neue IT-Weiterbildungssystem gibt auf Basis der vier neuen IT-Ausbildungsberufe drei Ebenen für die Weiterqualifizierung vor: Spezialisten, operative und strategische Professionals.

Der IT Systems Developer ist als Spezialist der Profilgruppe Software Developer zugeordnet.

Verwandte Profile

Der IT Systems Developer ist innerhalb eines Systementwicklungsprozesses mit seinem Tätigkeitsspektrum zwischen den Profilen:

- IT Systems Analyst
- und Software Developer, Database Developer, User Interface Developer, Multimedia Developer

eingebettet.

Der Übergang zwischen diesen Profilen verläuft fließend. Es gibt demnach Tätigkeiten, die in unterschiedlichen Ausprägungen von allen Spezialisten der Profilgruppe Software Developer durchgeführt werden.

Der IT Systems Developer entwirft basierend auf den Ergebnisse der Systemanalyse durch den IT Systems Analyst ein Systemdesign, definiert Anforderungen für Entwicklungsarbeiten der Software Developer, Database Developer, User Interface Developer und Multimedia Developer und integriert Produkte aus deren Tätigkeiten zu einem System.

Die Systementwicklung weist methodisch Parallelen zur Softwareentwicklung auf. Dadurch ist eine Verwandtschaft zu den Profilen Software Developer, Database Developer, User Interface Developer und Multimedia Developer gegeben. Die Konstruktion eines Systems basiert aber ausschließlich auf der Integration von Subsystemen und deren Systemelementen (Software- und Hardwarekomponenten). An der Implementierung der Software- und Hardwarekomponenten ist der IT Systems Developer planend und beratend, aber nicht ausführend beteiligt.

Aufstiegsqualifizierung

Der IT Systems Developer kann sich innerhalb des IT-Weiterbildungssystems zu den operativen Professionals IT Systems Manager oder IT Business Manager weiterbilden, aber auch bei entsprechender Spezialisierung zum IT Business Consultant.

3 Referenzprozess

Der Referenzprozess gibt den gesamten Entwicklungsprozess auf hohem Abstraktionsniveau wieder und ermöglicht so einen Überblick.

Mit den Teilprozessen wird in den Referenzprozess hineingezoomt. Die Teilprozesse entsprechen damit in etwa der Abbildung von Arbeitsprozessen; sie stellen einen konkreten Tätigkeitsverlauf einschließlich auslösendem Ereignis und Ergebnis dar.

Die zur Durchführung der Teilprozesse notwendigen Tätigkeiten und Kompetenzfelder werden jeweils in einem separaten Abschnitt aufgelistet.

Das Praxisprojekt dient als Beispiel zur Konkretisierung und Veranschaulichung. Es ist ein echtes, bereits durchgeführtes Projekt, auf dessen Grundlage die hier dargestellten Referenz- und Teilprozesse entwickelt wurden.

Das Profil „IT Systems Developer“ hat nur einen Referenzprozess: Die Systementwicklung.

3.1 Systementwicklung

Für das Verständnis des Referenzprozesses und seiner Teilprozesse soll an dieser Stelle zunächst die Verwendung einiger Begriffe geklärt werden.

Ein *System* besteht in diesem Kontext aus der zielführenden Komposition von Systemkomponenten. Es bezeichnet das Endprodukt der Systementwicklung. Das System als Ganzes, zusammen mit eventuellen organisatorischen Infrastrukturmaßnahmen erfüllt alle Anforderungen des Kunden. *Systemkomponenten* können Subsysteme oder Systemelemente sein. Ein *Subsystem* ist ebenfalls ein System, bezeichnet aber nicht das Endprodukt, sondern eine komplexe Systemkomponente (Zwischenprodukt) des Systems.

Systemelemente sind Software- oder Hardwarekomponenten (SW-/HW-Komponenten), die aus der Sicht des System nicht weiter zerlegt werden. Diese Komponenten werden konstruiert, d. h. entworfen und implementiert (SW) oder gebaut (HW). Systeme werden ebenfalls entworfen, entstehen aber durch die Integration von Systemkomponenten, d. h. von Subsystemen oder Systemelementen. *Integration* meint die Herstellung einer Einheit aus Differenziertem, *Systemintegration* demnach die Komposition von Systemen aus Subsystemen und/oder Systemelementen.

Eine *Systemarchitektur* entsteht im Ergebnis des technischen Entwurfs des Systems und beschreibt die Struktur der Systemkomponenten und deren Interaktion. Systemkomponenten interagieren über ihre Schnittstellen, d. h. eine Systemkomponente kann Dienste einer anderen nur über deren definierte Schnittstellen nutzen.

Ein System wird in einer *Systemumgebung* betrieben. Das System interagiert über seine Systemschnittstellen mit dieser Umgebung und nutzt deren Dienste. Test- und Integrationsumgebungen simulieren die Systemumgebungen teilweise oder komplett und werden für den jeweiligen Zweck um zusätzliche Bestandteile erweitert.

Anforderungen an ein System können in funktionale und nichtfunktionale Anforderungen unterschieden werden. *Funktionale Anforderungen* beschreiben einen Dienst, den das System erbringen soll. *Nichtfunktionale Anforderungen* beschreiben Eigenschaften bzgl. der Leistung des Systems, seiner Skalierbarkeit oder Verfügbarkeit sowie bzgl. der Sicherheit, Integrierbarkeit, Zukunftssicherheit und Erweiterbarkeit.

Bei der Entwicklung eines Systems wird i. d. R. auf Fertigprodukte zurückgegriffen. *Fertigprodukte* sind Subsysteme oder SW-/HW-Komponenten, deren Entwicklung unabhängig von dem zu entwickelnden System abgeschlossen ist. Sie werden angepasst und in das System integriert. Beispiele für Fertigprodukte sind bzgl. der Hardware existierende Geräte und bzgl. der Software COTS-Komponenten (Commercial-Off-the-Shelf-Softwarekomponenten), aber auch komplette Systeme, d. h. bereits integrierte Hardware und Software.

Prototypen realisieren immer nur einen Ausschnitt eines Systems. Dieser Ausschnitt kann vertikal oder horizontal durch das System gezogen werden. Der gewählte Ausschnitt bestimmt umzusetzende Anforderungen und relevante Systemkomponenten.

Der Referenzprozess unterteilt sich in folgende Arbeitsfelder, die durch die Teilprozesse näher spezifiziert werden:

1. Erstellen grober Systemarchitekturen auf Grundlage eines Anforderungsmodells unter Berücksichtigung von Musterarchitekturen und Fertigprodukten am Markt
2. Begleiten der Erstellung und Testen von Prototypen
3. Beschaffen und Prüfen von Fertigprodukten
4. Bewerten und Auswählen von Standards
5. Erstellen eines anforderungsgerechten Systemdesigns und Prüfen der Realisierbarkeit
6. Spezifizieren, Vorbereiten und Durchführen der Systemintegration und Systemtests
7. Spezifizieren von Anforderungen an die Erstellung von SW-/HW-Komponenten und die Anpassung von Fertigprodukten und Begleiten der Realisierungsprozesse
8. Vorbereiten und Begleiten der Installation des Systems, der Abnahmetests und der Übergabe
9. Mitarbeiten bei der Erstellung von Dokumentationen und Schulungsmaterialien sowie bei der Durchführung von Schulungen
10. Aufnehmen, Verwalten und Einarbeiten von Änderungsanforderungen an das System

3.1.1 Referenzprozess: Systementwicklung

Abbildung 6, Abbildung 7 auf Seite 15 und Abbildung 8 auf Seite 16 zeigen den gesamten Referenzprozess Systementwicklung mit seinen Teilprozessen.

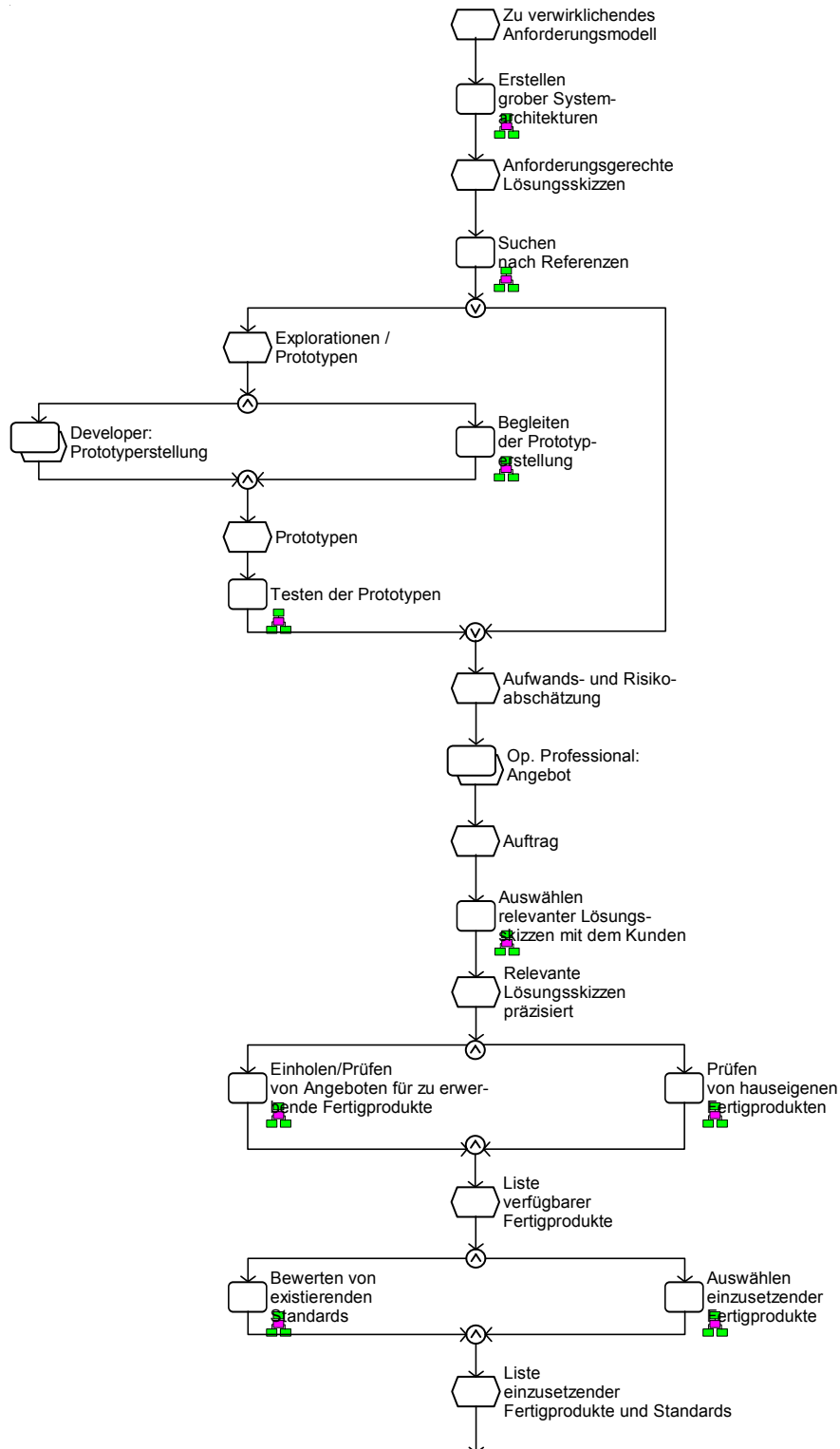


Abbildung 6: Referenzprozess Systementwicklung (Teil 1).

Abbildung 7: Referenzprozess Systementwicklung (Teil 2).

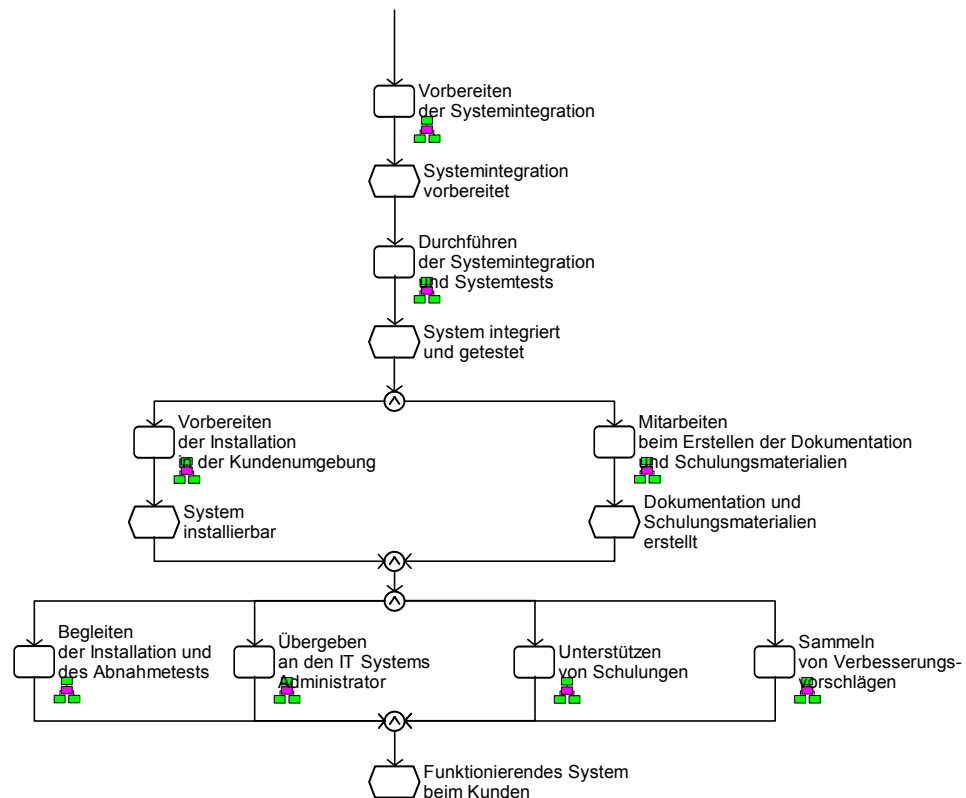


Abbildung 8: Referenzprozess Systementwicklung (Teil 3).

3.1.2 Das Beispielprojekt: Die Teltra Televisite

Das Beispielprojekt „Die Teltra Televisite“, welches diesem Referenzprofil zugrunde liegt, wurde von der Gesellschaft für telematische Traumatologie mbH (TELTRA) in Bochum durchgeführt. Ziel des Projekts war die Entwicklung eines telemedizinischen Systems (telematische Anwendungen in der Medizin), das eine herkömmliche Visite ergänzen und zum Teil ersetzen kann. Durch den Einsatz des Systems soll z. B. die Verweildauer von Patienten im Krankenhaus nach unfallchirurgischen Eingriffen verkürzt und damit erhebliche Einsparungen bei den Behandlungskosten und Krankengeldern erreicht werden.

Das System ermöglicht eine poststationäre Betreuung der Patienten in ihrem häuslichen Umfeld auf qualitativ hohem Niveau. Arzt und Patient können mithilfe eines Arzt- bzw. Patientenrechners über die Servicestation des Systems synchron und asynchron kommunizieren. Verschiedene geschützte Kommunikationskanäle ermöglichen den Austausch von Sprach- oder Textnachrichten und Aufnahmen einer Digitalkamera (Foto, Video) sowie eine Videokonferenz zwischen Arzt und Patient. Die Kommunikation zwischen beiden kann räumlich und zeitlich entkoppelt werden, was einen reibungslosen Informationsfluss sicherstellt.

Die telematische Übertragung der Aufnahmen von Wundenbereichen des Patienten, ergänzt um ein korrigierendes Farbmanagement ermöglicht dem Arzt die wirklichkeitsgetreue und reproduzierbare Darstellung dieser Wunden und deren visuelle Beurteilung. Zusammen mit den Sprach- oder Textnachrichten des Patienten (Antworten auf Fragen des Arztes) schätzt der Arzt den Behandlungsverlauf des Patienten ein. Patientendaten und Behandlungsverläufe werden auf einem zentralen Datenbankserver verwaltet.

3.1.3 Prozesskompass: Systementwicklung

1. Erstellen grober Systemarchitekturen
2. Suchen nach Referenzen
3. Begleiten der Prototyperstellung
4. Testen der Prototypen
5. Auswählen relevanter Lösungsskizzen mit dem Kunden
6. Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte
7. Prüfen von hauseigenen Fertigprodukten
8. Bewerten von existierenden Standards
9. Auswählen einzusetzender Standards
10. Erstellen von technischen Lösungsvorschlägen
11. Zerlegen in Subsysteme
12. Zerlegen in SW-/HW-Komponenten
13. Erstellen des Systemdesigns
14. Abgleichen von Kundenanforderungen und Systemdesign
15. Untersuchen auf Realisierbarkeit
16. Beschreiben der Schnittstellen
17. Spezifizieren der Integration und Systemtests
18. Definieren von zu erstellenden SW-/HW-Komponenten
19. Spezifizieren notwendiger Anpassungen an Fertigprodukten
20. Veranlassen der Beschaffung
21. Begleiten der SW-/HW-Komponentenerstellung
22. Begleiten der Prüfung und Anpassung von Fertigprodukten
23. Vorbereiten der Systemintegration
24. Durchführen der Systemintegration und Systemtests
25. Vorbereiten der Installation in der Kundenumgebung
26. Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien
27. Begleiten der Installation und Abnahmetests
28. Übergeben an den IT Systems Administrator
29. Unterstützen von Schulungen
30. Sammeln von Verbesserungsvorschlägen

3.1.4 Teilprozesse: Systementwicklung

Die Teilprozesse geben im Folgenden die Tätigkeiten des IT Systems Developer bei der Systementwicklung in einer beispielhaft sinnvollen Reihenfolge detailliert wieder. Sie basieren auf dem oben beschriebenen Beispielprojekt. Dieses Beispielprojekt wird im Weiteren auch zur Veranschaulichung der modellierten Prozesse dienen.

Nicht alle Teilprozesse werden in jedem Projekt im hier dargestellten Umfang vorkommen. Einem IT Systems Developer müssen sie jedoch auf Spezialistenebene vertraut sein.

3.1.4.1 Erstellen grober Systemarchitekturen

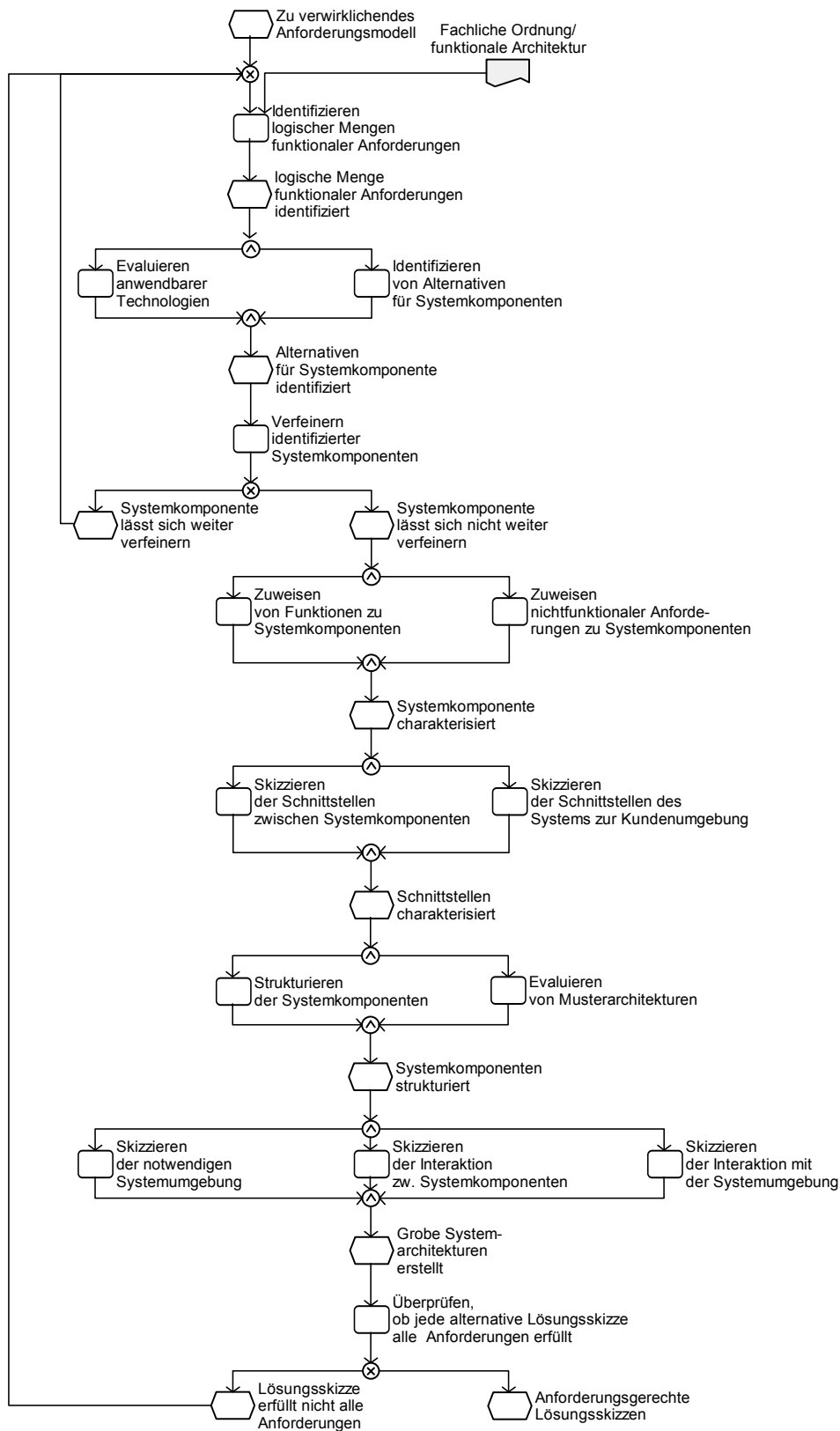


Abbildung 9: Erstellen grober Systemarchitekturen.

Der Teilprozess „Erstellen grober Systemarchitekturen“ hat zum Ziel, basierend auf den Kundenanforderungen und der fachlichen Ordnung des Systems erste Ideen zum technischen Entwurf grob zu skizzieren. Dabei können mehrere Lösungsvarianten erarbeitet werden.

Zunächst werden die funktionalen Kundenanforderungen entsprechend ihrem funktional-logischen Zusammenhang gruppiert. Anhand dieser Gruppierung lassen sich Komponenten des Systems (Subsysteme oder Systemelemente) und ihre Funktion identifizieren. Parallel dazu können erste technologische Überlegungen für Standardfunktionalitäten wie z. B. einen Web-Server einfließen. Sind verschiedene alternative Systemkomponenten zur Erfüllung der funktionalen Anforderungen möglich, sollten diese beschreiben werden.

Lässt sich die Funktion einer Systemkomponente weiter verfeinern, werden die verfeinerten funktionalen Anforderungen erneut gruppiert und spezifischere Systemkomponenten identifiziert. Die Verfeinerung sollte in diesem Stadium nicht über eine Skizze hinausgehen.

Wurden alle Systemkomponenten identifiziert, werden ihnen funktionale und nichtfunktionale Anforderungen zugewiesen. Diese Zuweisung dient der Vorbereitung der abschließenden Kontrolle der vollen Abdeckung aller Kundenanforderungen durch das skizzierte System und der weiteren Spezifikation einzelner Systemkomponenten.

Zuvor werden die Schnittstellen zwischen den Systemkomponenten und zur Systemumgebung des Kunden skizziert. Danach werden die Systemkomponenten grob strukturiert. Hierfür sollten etablierte Musterarchitekturen herangezogen werden.

Resultierende Anforderungen an die Systemumgebung, die Interaktion zwischen den Systemkomponenten und die Interaktion zwischen dem System und der Systemumgebung werden im Folgenden grob festgehalten.

Zum Abschluss erfolgt eine Überprüfung, ob alle erarbeiteten Lösungsskizzen zum System alle Kundenanforderungen erfüllen. Ist dies nicht der Fall, muss der Teilprozess erneut durchlaufen werden.

3.1.4.1.1 Tätigkeiten: Erstellen grober Systemarchitekturen

- Identifizieren logischer Mengen funktionaler Anforderungen
- Evaluieren anwendbarer Technologien
- Identifizieren von Alternativen für Systemkomponenten
- Verfeinern identifizierter Systemkomponenten
- Zuweisen von Funktionen zu Systemkomponenten
- Zuweisen von nichtfunktionalen Anforderungen zu Systemkomponenten
- Skizzieren der Schnittstellen zwischen Systemkomponenten
- Skizzieren der Schnittstellen des Systems zur Kundenumgebung
- Strukturieren der Systemkomponenten
- Evaluieren von Musterarchitekturen
- Skizzieren der notwendigen Systemumgebung
- Skizzieren der Interaktion zwischen Systemkomponenten
- Skizzieren der Interaktion mit der Systemumgebung
- Überprüfen, ob jede alternative Lösungsskizze alle Anforderungen erfüllt

3.1.4.1.2 Kompetenzfelder: Erstellen grober Systemarchitekturen

Fähigkeiten/Fertigkeiten

- verschiedene funktionale Anforderungen entsprechend ihrem funktional-logischen Zusammenhang zu Mengen zusammenfassen und ordnen können
- Alternativen für Systemkomponenten abgrenzen und anhand ihrer funktionalen Eigenschaften den logischen Mengen funktionaler Anforderungen zuordnen können
- aus den Kundenanforderungen und der Anwendungsdomäne Kriterien für die Bewertung von Technologien und Musterarchitekturen herleiten können
- Technologien und Musterarchitekturen entsprechend ihrer Merkmalsausprägung den gewichteten Kriterien zuordnen und daraus eine Beurteilung ableiten können
- ausgehend vom Ziel, der Erstellung einer groben Systemarchitektur, die Systemkomponenten in ihre Einzelfunktionen (Subkomponenten) auflösen und diese entsprechend der Zielerfüllung angemessen detailliert beschreiben können
- den Zusammenhang zwischen Funktionen und nichtfunktionalen Anforderungen und deren Abbildung auf Systemkomponenten erkennen, verstehen und schriftlich deutlich machen können
- die wesentlichen Aspekte der Schnittstellen der Systemkomponenten und des Systems, der Interaktionen sowie der Systemumgebung in angemessener Form zur Veranschaulichung einer groben Systemarchitektur darstellen können
- Systemkomponenten entsprechend ihren Wechselbeziehungen und Abhängigkeiten gliedern und zu einem System ordnen können
- eine grobe Systemarchitektur im Hinblick auf die Erfüllung aller Kundenanforderungen untersuchen können

Wissen

- Technologiekenntnisse, Kriterien und Normen der Anwendungsdomäne
- Technologiekenntnisse verschiedener Aspekte eines Systems (z. B. Kommunikationsprotokolle, Persistenz)
- Musterarchitekturen

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassendiagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenflussdiagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Methoden zur Darstellung und Beschreibung von Varianten (z. B. Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (z. B. gewichtete Entscheidungsmatrix, Nutzwertanalyse)

3.1.4.1.3 **Beispiel: Erstellen grober Systemarchitekturen**

Im Beispielprojekt „Die Teltra Televisite“ erarbeitete und dokumentierte das Entwicklerteam ausgehend von den Anforderungen der zwei Nutzergruppen Patient und Arzt und der fachlichen Ordnung des Systems erste Ideen für eine technische Realisierung. Dabei kristallisierte sich die Idee, das System als Web-Anwendung zu entwerfen, als sehr geeignet heraus und wurde für den Entwurf einer groben Systemarchitektur priorisiert. Dafür wurden zunächst die funktionalen Anforderungen hinsichtlich ihres logischen Zusammenhangs gruppiert.

Beispiele für Gruppen funktionaler Anforderungen sind unter anderen: die visuelle Erfassung der Wundbereiche des Patienten, die Eingabe von Daten durch den Patienten (Fieber, Schmerz, Befinden, Wundzustand), die sprachliche und visuelle Kommunikation zwischen Arzt und Patient, die ortsunabhängige, verschlüsselte Übertragung der Patientendaten, die zentrale Verwaltung der Patientendaten und der Zugriff und die aufbereitete Darstellung der Patientendaten für den Arzt.

Anschließend hat das Entwicklerteam zu den einzelnen funktionalen Gruppen verschiedene Technologien evaluiert. Für die ortsunabhängige, verschlüsselte Übertragung der Patientendaten kommen u. a. Übertragungstechnologien und Standards aus dem Mobilfunkbereich wie z. B. GSM/HSCSD/GPRS/UTMS, für die Kommunikation zwischen Client–Server das HTTP bzw. HTML und für die Verschlüsselung SSL bzw. HTTPS in Frage. Es wurden erste Systemkomponenten identifiziert und Alternativen festgehalten. In Betracht kamen für die Funktionsgruppen „Visuelle Erfassung der Wundbereiche“, „Dateneingabe“, „Sprachliche und visuelle Kommunikation“ und „Ortsunabhängige, verschlüsselte Übertragung“ folgende Systemkomponenten und Alternativen: Digitalkamera in Verbindung mit einem Personalcomputer, mobiler PC, Laptop oder TabletPC mit Soundkarte und eingebautem oder externem Mikrofon bzw. Lautsprechern und einem Modem, einer Mobilfunkkarte oder Handy.

Beispiele für eine Verfeinerung der identifizierten Systemkomponenten sind z. B. Softwaresysteme bzw. -komponenten wie Web-Server, Datenbankverwaltungssystem und Web-Browser für die zentrale Verwaltung der Patientendaten und den Zugriff und die aufbereitete Darstellung der Patientendaten für den Arzt.

Den identifizierten Systemkomponenten wurden die funktionalen und die nichtfunktionalen Anforderungen wie z. B. Sicherheit bzgl. Übertragung, Übertragungsraten für Videokonferenz und Verfügbarkeit der Patientendaten zugewiesen.

Danach wurden die Schnittstellen zwischen den Systemkomponenten erfasst und skizziert. Beispiele dafür sind z. B. die Schnittstellen zwischen Patientenrechner und Digitalkamera (Bluetooth, USB), zwischen Patientenrechner und Server (ins Internet über PCMCIA, Mobilfunkkarte und Mobilfunknetz oder serielle Schnittstelle, Modem und Festnetz), zwischen Arztrechner und Server (Zugriff über Netzwerkkarte und LAN), aber auch die Schnittstelle zwischen Web-Server und Datenbankverwaltungssystem (PHP-Library, JDBC).

Auf Grundlage der identifizierten Systemkomponenten und ihrer Schnittstellen konnte das Entwicklerteam diese strukturieren und grobe, alternative Systemarchitekturen skizzieren. Im Anschluss wurden die Systemarchitekturen hinsichtlich der Erfüllung aller Anforderungen unter Verwendung von Checklisten geprüft.

3.1.4.2 Suchen nach Referenzen

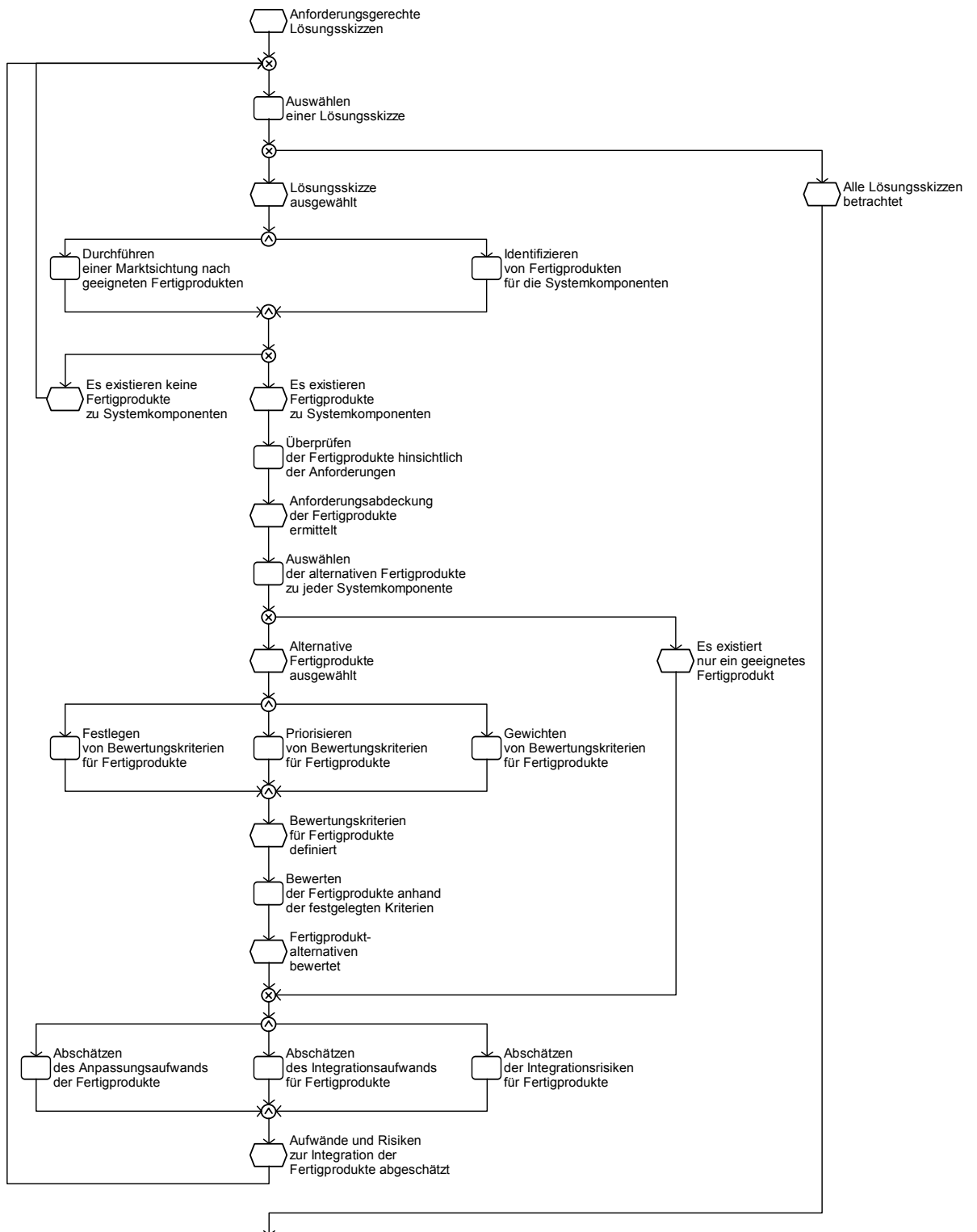


Abbildung 10: Suchen nach Referenzen (Teil 1).

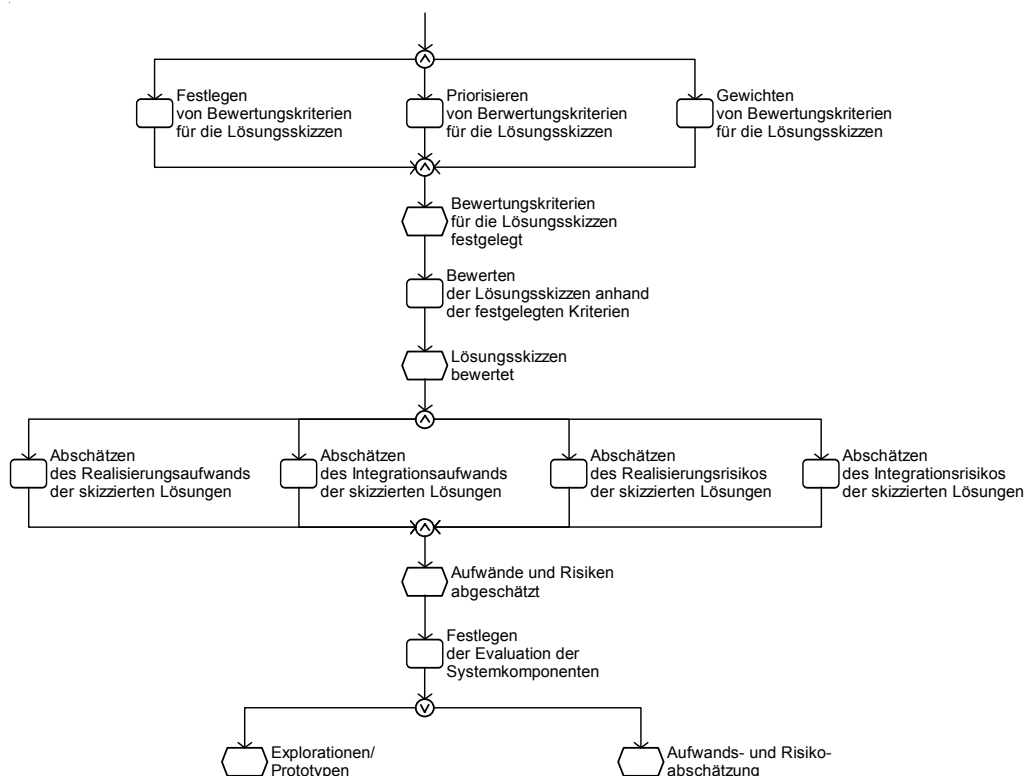


Abbildung 11: Suchen nach Referenzen (Teil 2).

Ziel des Teilprozesses „Suchen nach Referenzen“ ist, ein Angebot für den Kunden und dem Kunden selbst eine Entscheidungsgrundlage bei der Auswahl einer Lösungsvariante vorzubereiten.

Dazu wird zunächst der Einsatz von Fertigprodukten bei der Realisierung einzelner Systemkomponenten aller skizzierten Lösungsvarianten geprüft. Nach einer Marktsichtung werden verfügbare Fertigprodukte identifiziert und festgehalten, in welchem Umfang sie die Anforderung an die jeweilige Systemkomponente abdecken. Wurden im Ergebnis der Marktsichtung verschiedene alternative Fertigprodukte ermittelt, müssen diese anhand zu bestimmender Kriterien bewertet werden. Diese Bewertung ist Voraussetzung für die Auswahl des geeignetsten Fertigprodukts. Danach werden die Aufwände für eventuelle Anpassungen der Fertigprodukte sowie die Aufwände und Risiken bei der Integration der Fertigprodukte in das System abgeschätzt.

Anschließend ist es möglich, die verschiedenen Lösungsvarianten nach festzulegenden Kriterien zu bewerten, ihre Aufwände und Risiken bei der Realisierung sowie der Integration von Systemkomponenten zu dem jeweils skizzierten System abzuschätzen.

3.1.4.2.1 Tätigkeiten: Suchen nach Referenzen

- Auswählen einer Lösungsskizze
- Durchführen einer Marktsichtung nach geeigneten Fertigprodukten
- Identifizieren von Fertigprodukten für die Systemkomponenten
- Überprüfen der Fertigprodukte hinsichtlich der Anforderungen
- Auswählen der alternativen Fertigprodukte zu jeder Systemkomponente
- Festlegen von Bewertungskriterien für Fertigprodukte
- Priorisieren von Bewertungskriterien für Fertigprodukte
- Gewichten von Bewertungskriterien für Fertigprodukte

- Bewerten der Fertigprodukte anhand der festgelegten Kriterien
- Abschätzen des Anpassungsaufwands der Fertigprodukte
- Abschätzen des Integrationsaufwands für Fertigprodukte
- Abschätzen der Integrationsrisiken für Fertigprodukte
- Festlegen der Bewertungskriterien für die Lösungsskizzen
- Priorisieren von Bewertungskriterien für die Lösungsskizzen
- Gewichten von Bewertungskriterien für die Lösungsskizzen
- Bewerten der Lösungsskizzen anhand der festgelegten Kriterien
- Abschätzen des Realisierungsaufwands der skizzierten Lösungen
- Abschätzen des Integrationsaufwands der skizzierten Lösungen
- Abschätzen des Realisierungsrisikos der skizzierten Lösungen
- Abschätzen des Integrationsrisikos der skizzierten Lösungen
- Festlegen der Evaluation der Systemkomponenten

3.1.4.2.2 Kompetenzfelder: Suchen nach Referenzen

Fähigkeiten/Fertigkeiten

- recherchieren und Informationen beschaffen können
- Fertigprodukte abgrenzen und anhand ihrer Funktionalität den Systemkomponenten zuordnen können
- ein Fertigprodukt im Hinblick auf die Erfüllung der Anforderungen untersuchen sowie die Ergebnisse und die Abdeckung der Anforderungen dokumentieren können
- aus den Kundenanforderungen und der Anwendungsdomäne Kriterien für die Bewertung von Fertigprodukten und Lösungsskizzen herleiten können
- Fertigprodukte und Lösungsskizzen entsprechend ihren Merkmalsausprägungen gewichteten Kriterien zuordnen und daraus eine Beurteilung herleiten können
- den Anpassungs- und Integrationsaufwand von Fertigprodukten sowie den Realisierungs- und Integrationsaufwand der Lösungsskizzen unter Verwendung geeigneter Methoden näherungsweise quantitativ bestimmen können
- die Integrationsrisiken von Fertigprodukten sowie die Realisierungs- und Integrationsrisiken von Lösungsskizzen unter Verwendung geeigneter Methoden näherungsweise bestimmen können

Wissen

- Markt- und Produktkenntnisse
- Fachliteratur/-zeitschriften

Werkzeuge/Methoden

- Recherchewerkzeuge (z. B. Internet, Datenbanken)
- Methoden zur Darstellung und Beschreibung von Varianten (z. B. Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (z. B. gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Methoden zur Aufwandsschätzung (z. B. Function-Point-Methode, Constructive Cost Model COCOMO, Vergleichsmethoden)
- Methoden des Risikomanagements (z. B. SWOT-Analyse, Impact/Probability Matrix)

3.1.4.2.3 Beispiel: Suchen nach Referenzen

Auch im Beispielprojekt war es notwendig, zur Vorbereitung einer Entscheidungsgrundlage bei der Auswahl einer Lösungsvariante zunächst für alle Systemkomponenten den Einsatz von Fertigprodukten zu prüfen. Durch eine Marktsichtung wurde für die verschiedenen Systemkomponenten eine Reihe von möglichen Fertigprodukten identifiziert. Zum Beispiel kommen für das Patientensystem Computer (PC, Mobile PC, Laptop) und Digitalkameras verschiedener Hersteller in Betracht. Beim Arztsystem ging es lediglich darum, geeignete Monitore für die wirklichkeitsgetreue Darstellung der Wunden zu finden, da in der Regel ein Stations-PC vorhanden ist. Softwareprodukte wie Betriebssysteme, Web-Server, Web-Browser, Datenbankverwaltungssystem, aber auch Entwicklungsumgebungen wie Java oder PHP wurden ebenfalls berücksichtigt.

Die identifizierten Fertigprodukte wurden auf die Erfüllung der Anforderungen überprüft. Bei der Auswahl möglicher Digitalkameras z. B. musste das Entwicklerteam besonders die Auflösung und Farbwiedergabe und bei den Arztmonitoren die Möglichkeit der Kalibrierung beachten.

Bei den Digitalkameras, Arztmonitoren und Softwareprodukten blieben in der Regel nach der Überprüfung der Anforderungen mehrere Produkte zur Auswahl. Es war also erforderlich, Bewertungskriterien für die Auswahl eines oder einer geringen Auswahl von Produkten zu finden. Bei den Digitalkameras wurden folgende Bewertungskriterien festgelegt: maximale Auflösung, Farbwiedergabe, Handhabung, Preis, Leistung, Desinfizierbarkeit, verfügbare Anschlüsse. Bei den Softwareprodukten spielten eher Kriterien wie Verbreitung, Zuverlässigkeit, Kosten, Sicherheit und leichte Einarbeitung eine Rolle. Diese Bewertungskriterien wurden gewichtet und mithilfe einer Entscheidungsmatrix ein Produkt ausgewählt.

Nach der Auswahl der Fertigprodukte war es möglich, den Aufwand und die Risiken bei ihrer Anpassung und Integration abzuschätzen. Daraufhin konnten die Aufwände und Risiken der verschiedenen Lösungsalternativen abgeschätzt und die Lösungsalternativen selbst nach Kriterien wie z. B. Entwicklungskosten oder -risiko bewertet werden.

Zum Abschluss hat das Entwicklerteam entschieden, die Lösungsskizze durch die Entwicklung eines Prototypen zu überprüfen.

3.1.4.3 Begleiten der Prototyperstellung

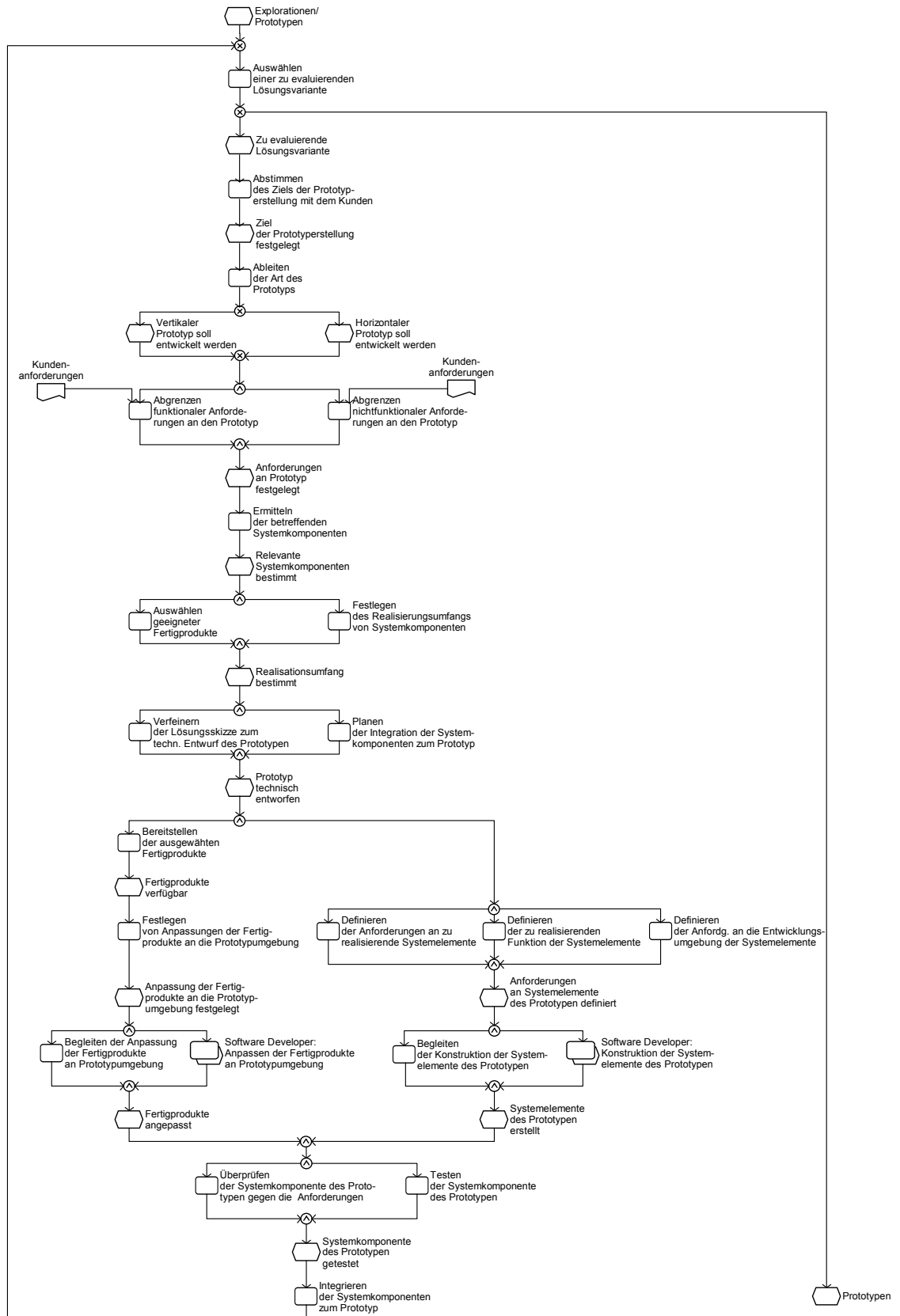


Abbildung 12: Begleiten der Prototyperstellung.

Ziel des Teilprozesses „Begleiten der Prototyperstellung“ ist die Schaffung einer Evaluationsbasis für auszuwählende Systemkomponenten. Die Evaluation dient der Überprüfung und Präzisierung der bisher erstellten Lösungsskizzen und der Kundenanforderungen an das System.

Zu Beginn werden die Lösungsskizzen ausgewählt, die mithilfe eines Prototypen evaluiert werden sollen. Für jeden Prototypen werden Ziel und Art des Prototypen festgelegt. Danach erfolgt ausgehend von den Kundenanforderungen an das System die Abgrenzung der funktionalen und nichtfunktionalen Anforderungen an den Prototypen sowie die Ermittlung der betroffenen Systemkomponenten.

Nach der Auswahl geeigneter Fertigprodukte und der Festlegung des Realisierungsumfangs der betroffenen Systemkomponenten kann jede Lösungsskizze zu einem technischen Entwurf des Prototypen verfeinert und die Integration der Systemkomponenten zum Prototypen geplant werden. Die Anpassung von Fertigprodukten sowie die Konstruktion der Systemkomponenten des Prototypen werden vom IT Systems Developer begleitet. Nach der Erstellung der Systemelemente bzw. Anpassung der Fertigprodukte werden diese hinsichtlich der abgegrenzten Anforderungen geprüft, getestet und zum Prototypen integriert.

3.1.4.3.1 Tätigkeiten: Begleiten der Prototyperstellung

- Auswählen einer zu explorierenden Lösungsvariante
- Abstimmen des Ziels der Prototyperstellung mit dem Kunden
- Ableiten der Art des Prototyps
- Abgrenzen funktionaler Anforderungen an den Prototyp
- Abgrenzen nichtfunktionaler Anforderungen an den Prototyp
- Ermitteln der betreffenden Systemkomponenten
- Auswählen geeigneter Fertigprodukte
- Festlegen des Realisierungsumfangs von Systemkomponenten
- Verfeinern der Lösungsskizze zum technischen Entwurf des Prototypen
- Planen der Integration der Systemkomponenten zum Prototyp
- Bereitstellen der ausgewählten Fertigprodukte
- Festlegen von Anpassungen der Fertigprodukte an die Prototypumgebung
- Definieren der Anforderungen an die zu realisierenden Systemelemente
- Definieren der zu realisierenden Funktion der Systemelemente
- Definieren der Anforderungen an die Entwicklungsumgebung der Systemelemente
- Begleiten der Anpassung der Fertigprodukte an die Prototypumgebung
- Begleiten der Konstruktion der Systemelemente des Prototypen
- Überprüfen der Systemelemente des Prototypen gegen die Anforderungen
- Testen der Systemelemente des Prototypen
- Integrieren der Systemkomponenten zum Prototyp

3.1.4.3.2 Kompetenzfelder: Begleiten der Prototyperstellung

Fähigkeiten/Fertigkeiten

- Kommunikationsfähigkeit
- ausgehend von Ziel und Art des Prototypen dessen funktionale und nichtfunktionale Anforderungen genau bestimmen und von den anderen Kundenanforderungen absetzen können

- feststellen können, welche Systemkomponenten und Fertigprodukte in den Prototypen einbezogen werden müssen
- aus den Anforderungen an den Prototyp den Realisierungsumfang einzelner Systemkomponenten herleiten können
- alle betroffenen Systemkomponenten einer Lösungsskizze entsprechend den Erfordernissen des technischen Entwurfs eines Prototypen in ihre Teile auflösen und detailliert beschreiben können
- basierend auf dem technischen Entwurf Schritte zur Integration der Systemkomponenten zum Prototypen ausarbeiten können
- die Integrationsschritte entsprechend ihren Wechselwirkungen, Randbedingungen und Auswirkungen als logische Abfolge strukturieren können
- mögliche Alternativen in der logischen Abfolge der Integration bewerten und die optimale Alternative auswählen können
- aus der optimalen logischen Abfolge der Integrationsschritte eine zeitliche und organisatorische Abfolge ableiten können, die Aufwandsabschätzungen zur Realisierung und zur Integration der Systemkomponenten sowie die Informationen zur Verfügbarkeit von Fertigprodukten und die Aufwände zu deren Anpassung berücksichtigt
- Aufwände zur Integration des Prototypen abschätzen können
- Fertigprodukte für die Erstellung des Prototypen beschaffen können
- aus den zugewiesenen Anforderungen an eine Systemkomponente des Prototypen und der ermittelten Abdeckung durch ein Fertigprodukt den Anpassungsumfang des Fertigprodukts ableiten können
- basierend auf dem technischen Entwurf Anforderungen an zu realisierende Systemelemente herleiten können
- aus den konkreten Merkmalsausprägungen der Systemelemente Anforderungen an deren Entwicklungsumgebung ableiten können
- realisierte Systemelemente des Prototypen im Hinblick auf die Erfüllung der zugewiesenen funktionalen und nichtfunktionalen Anforderungen an den Prototypen untersuchen können
- Fehler der Systemkomponenten durch Unit-, Modul- oder Komponententests erkennen und protokollieren können
- Systemkomponenten nach den aufgezeigten Integrationsschritten zum Prototypen zusammenstellen können
- Fehler des Prototypen durch Integrationstest und Systemtest erkennen und protokollieren können
- Fehlerprotokolle auswerten und weitere Vorgehensweise bestimmen können

Wissen

- Prototyping
- Verständnis von Funktion, Schnittstellen und Verhalten der Systemkomponenten und Fertigprodukte
- betriebliche Beschaffungsvorgänge

Werkzeuge/Methoden

- Interviewtechniken
- Moderationstechniken
- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)

- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassendiagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenflussdiagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Methoden der Strukturierung, des Ablauf- und Terminmanagements (z. B. Netzplantechnik)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Methoden zur Aufwandsschätzung (z. B. Vergleichsmethoden)
- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (z. B. funktionaler, struktureller Test)

3.1.4.3.3 Beispiel: Begleiten der Prototyperstellung

Ziel der Prototyperstellung im Beispielprojekt war die Überprüfung der Bedienbarkeit, der Bildqualität und der Realisierbarkeit. Das Entwicklerteam entwickelte einen eingeschränkt vertikalen Prototypen, d. h. Patientensystem, Serversystem und Arztsystem wurden durchgehend, aber in stark eingeschränktem Umfang realisiert.

So sollten z. B. das Serversystem ohne Verwaltungsfunktionalität für Patientendaten realisiert, die Sicherheitsanforderungen vernachlässigt und das Arztsystem weitgehend simuliert werden.

Konkrete Fertigprodukte wie Digitalkamera, Monitor und TabletPC hat das Entwicklerteam ausgeliehen, geprüft, entsprechend angepasst, d. h. beispielsweise konfiguriert oder auch Software entwickelt und installiert und zum Prototypen integriert.

3.1.4.4 Testen der Prototypen

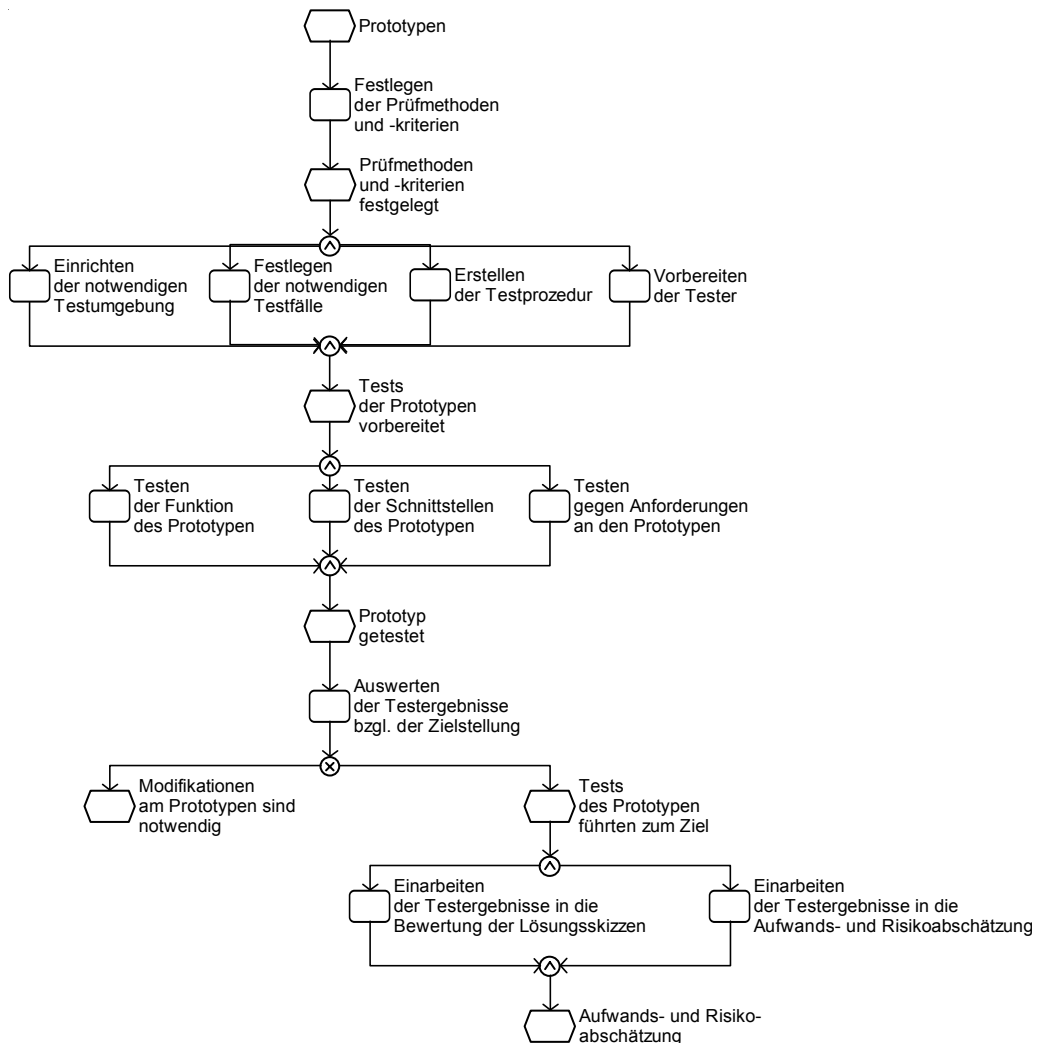


Abbildung 13: Testen der Prototypen.

Ziel des Teilprozesses „Testen der Prototypen“ ist die Evaluation der Lösungsskizzen und Kundenanforderungen anhand des erstellten Prototypen.

Dazu müssen zunächst ausgehend von den Anforderungen an den Prototypen die Anforderungen an dessen Prüfung bzw. Test abgeleitet, Prüfmethode und -kriterien festgelegt werden. Danach werden die notwendigen Testumgebungen für jeden Prototypen eingerichtet, Testfälle festgelegt und die Tester vorbereitet. Im Anschluss lassen sich realisierte Funktionen, Schnittstellen und die Erfüllung von Kundenanforderungen testen. Die Testergebnisse werden hinsichtlich der Zielstellung des Prototypen ausgewertet und in die Bewertung der Lösungsskizzen und in die Aufwand- bzw. Risikoabschätzungen eingearbeitet.

3.1.4.4.1 Tätigkeiten: Testen der Prototypen

- Festlegen der Prüfmethode und -kriterien
- Einrichten der notwendigen Testumgebung
- Festlegen der notwendigen Testfälle
- Erstellen der Testprozedur

- Vorbereiten der Tester
- Testen der Funktion des Prototypen
- Testen der Schnittstellen des Prototypen
- Testen gegen die Anforderungen an den Prototypen
- Auswerten der Testergebnisse bzgl. der Zielstellung
- Einarbeiten der Testergebnisse in die Bewertung der Lösungsskizzen
- Einarbeiten der Testergebnisse in die Aufwands- und Risikoabschätzung

3.1.4.4.2 Kompetenzfelder: Testen der Prototypen

Fähigkeiten/Fertigkeiten

- aus den Anforderungen an den Prototypen Anforderungen an dessen Prüfung/Test und daraus die anzuwendenden Prüfmethoden ableiten können
- Prüfkriterien festlegen können, die eine Bewertung der Prüfung hinsichtlich ihrer erfolgreichen und ausreichenden Durchführung ermöglichen
- aus den konkreten Merkmalsausprägungen des Prototypen, den festgelegten Prüfmethoden und -kriterien Anforderungen an die Testumgebung ableiten können
- Testfälle für z. B. Black-Box- und White-Box-Tests aus Designdokumenten (z. B. UML-Zustandsübergangs-Diagrammen) und Kundenanforderungen ableiten können
- ausgehend von den Testfällen eine genaue Arbeitsanleitung für die Tester (mit Testvorbereitung, -nachbereitung und Durchführung) erarbeiten und dokumentieren können
- Testprozedur in nachvollziehbaren Schritten dem Tester vermitteln können
- Fehler des Prototypen hinsichtlich seiner Funktion, der Schnittstellen und der Erfüllung der Anforderungen erkennen und protokollieren können
- Testergebnisse hinsichtlich der Zielstellung des Prototypen prüfen und als aussagekräftige Entscheidungsgrundlage aufbereiten können
- Auswertungsergebnisse der Tests des Prototypen in die Bewertung der Lösungsskizzen sinnvoll einfügen können

Werkzeuge/Methoden

- Black-Box-Testfallentwurfs-Methoden (Äquivalenzklassenbildung, Grenzwertanalyse, Intuitive Testfallermittlung, Funktionsabdeckung)
- White-Box-Testfallentwurf

3.1.4.4.3 Beispiel: Testen der Prototypen

Ziel des Prototypen im Beispielprojekt war die Überprüfung der Bedienbarkeit, der Bildqualität und der Realisierbarkeit. Hinsichtlich der Bedienbarkeit war es notwendig, die Bedienung der Digitalkamera und die Bedienung der Nutzerschnittstelle des Patientenrechners mit einem Touchscreen durch ältere Patienten testen zu lassen. Digitalkamera und Arztmonitor mussten auf Bildqualität und wirklichkeitsgetreue Farbwiedergabe geprüft werden.

Die Testumgebung (Digitalkamera, Patientenrechner, Server und Arztrechner mit Monitor) wurde in einem Entwicklungslabor der Firma Teltra aufgebaut, da die räumliche Nähe der Systemkomponenten für die Tests besonders wichtig war.

Das Entwicklerteam arbeitete Testfälle für die Wundaufnahme durch den Patienten und die Wundbeurteilung durch den Arzt aus. Mit den Testpersonen (ältere Patienten und Ärzte) wurden Termine für die Durchführung der Tests vereinbart. Vor der Durchführung der Tests wurden die Testpersonen in die Bedienung der Systemkomponenten eingewiesen.

Bezüglich der Funktion wurde z. B. geprüft, an welchen Körperteilen ein älterer Patient Wunden mit einer Digitalkamera aufnehmen und wie er dabei die Digitalkamera bedienen kann. Die Übertragung der Wundbilder von der Digitalkamera auf den Patientenrechner, die Bedienung der Patientenanwendung mit einem Touchscreen und die Übertragung der Daten auf den Server per Mobilfunk wurden ebenfalls getestet.

Auf der Seite des Arztes wurde geprüft, inwieweit es dem Arzt möglich ist, Wunden anhand der vom Patienten übertragenen Wundbilder zu beurteilen. Das Entwicklerteam testete, welche Qualität der Wundbilder für eine korrekte Beurteilung durch den Arzt notwendig ist und inwieweit sich der Weißabgleich der Digitalkamera und die Kalibrierung des Arztmonitors aufeinander abstimmen lassen, damit die Wundbereiche wirklichkeitsgetreu wiedergegeben werden.

Bezüglich der Schnittstellen wurde z. B. geprüft, ob sich für den Anschluss der Digitalkamera an den Patientenrechner ein USB-Kabel oder die Bluetooth-Schnittstelle besser eignet.

Im Ergebnis des Tests des Prototypen kam u. a. heraus, dass die Patientenanwendung zwar grundsätzlich auch durch ältere Patienten bedienbar ist, die Patienten aber geschult werden müssen. Die graphische Nutzerschnittstelle muss auf wenige einfache Schaltflächen beschränkt und zusätzlich mit ausführlichen und verständlichen Fehlermeldungen versehen werden. Bei den Schnittstellen erwies sich der Einsatz der Bluetooth-Schnittstelle als nicht praktikabel.

Für den Patientenrechner (TabletPC) muss ein zusätzlicher Schutz entworfen werden, um die Desinfizierbarkeit zu verbessern, die Fehlbedienung durch Abdeckung nicht verwendeter Anschlüsse zu minimieren und eine bessere Stoßfestigkeit zu erreichen.

Die Ergebnisse der Tests wurden in die verschiedenen Lösungsskizzen zum System eingearbeitet.

3.1.4.5 Auswählen der zu realisierenden Lösungsskizze mit dem Kunden

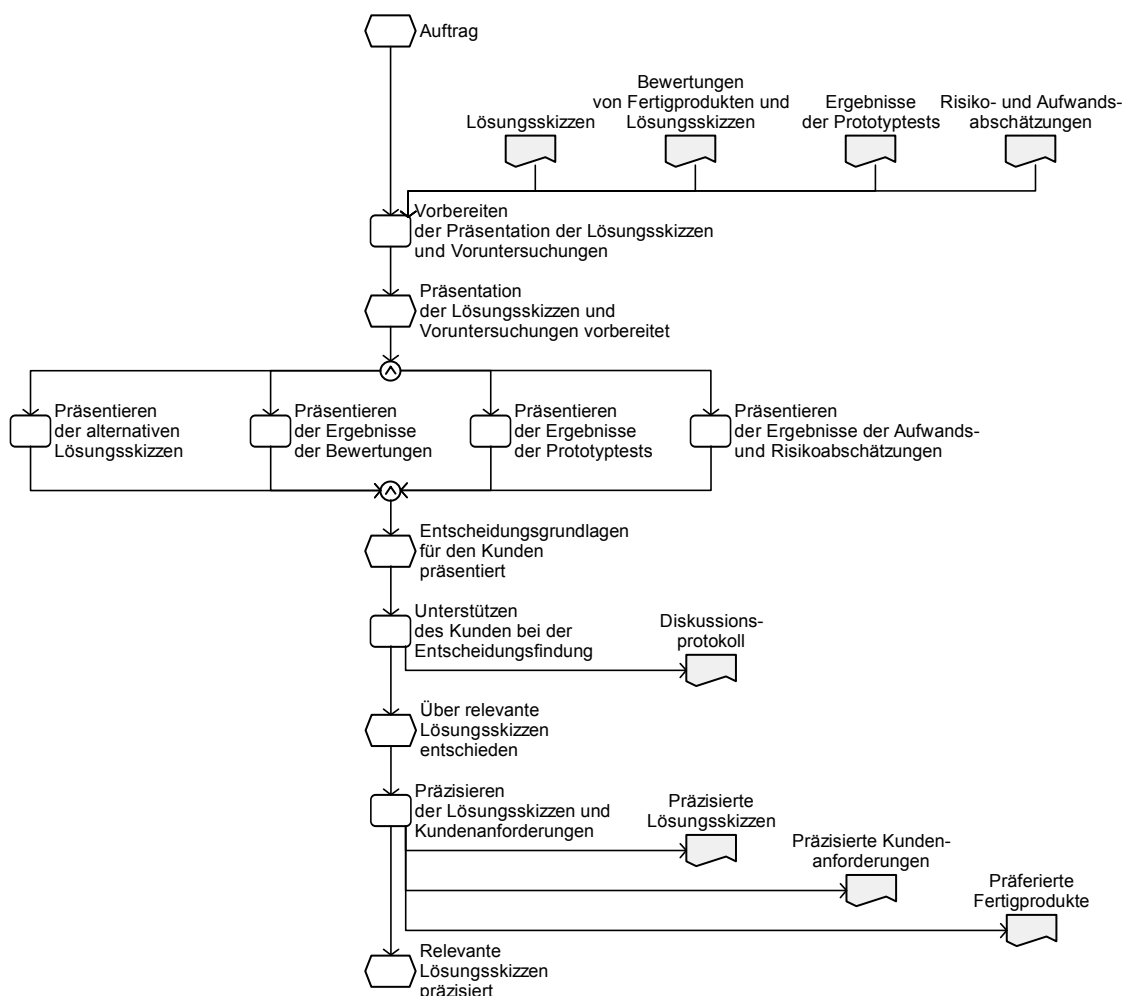


Abbildung 14: Auswählen der zu realisierenden Lösungsskizze mit dem Kunden.

Auf Grundlage der Kundenanforderungen, der erarbeiteten Lösungsskizzen, der Aufwands- und Risikoabschätzungen sowie der Evaluation anhand der Prototypen wurde durch den operativen Professional ein Angebot für den Kunden erstellt. Hat der Kunde über das Angebot positiv entschieden, gilt es im Teilprozess „Auswählen der zu realisierenden Lösungsskizze mit dem Kunden“ relevante Lösungsskizzen auszuwählen und diese einschließlich der Kundenanforderungen mit dem Kunden zusammen zu präzisieren. Dazu muss zunächst die Präsentation aller Lösungsskizzen und ihrer Voruntersuchungen – d. h. Auswahl und Bewertung von Fertigprodukten, die Prototypen und deren Testergebnisse sowie Aufwands- und Risikoabschätzungen – vorbereitet werden. Nach der Durchführung der Präsentation beim Kunden wird dieser in der Entscheidungsfindung unterstützt, welche der alternativen Lösungsskizzen im weiteren Entwicklungsprozess berücksichtigt werden. In der Diskussion mit dem Kunden werden die relevanten Lösungsskizzen und seine Anforderungen weiter präzisiert. Gleichzeitig wird ermittelt, ob der Kunde bestimmte Fertigprodukte für sein System präferiert.

3.1.4.5.1 Tätigkeiten: Auswählen der zu realisierenden Lösungsskizze mit dem Kunden

- Vorbereiten der Präsentation der Lösungsskizzen und Voruntersuchungen
- Präsentieren der alternativen Lösungsskizzen
- Präsentieren der Ergebnisse der Bewertungen
- Präsentieren der Ergebnisse der Prototypentests
- Präsentieren der Ergebnisse der Aufwands- und Risikoabschätzungen
- Unterstützen des Kunden bei der Entscheidungsfindung
- Präzisieren der Lösungsskizzen und Kundenanforderungen

3.1.4.5.2 Kompetenzfelder: Auswählen der zu realisierenden Lösungsskizze mit dem Kunden

Fähigkeiten/Fertigkeiten

- Lösungsskizzen, ihre Bewertung, die Ergebnisse der Prototypentests und die Risiko- und Aufwandsabschätzungen in vergleichbarer Form aufbereiten, strukturieren, visualisieren und beim Kunden vorstellen können
- Kommunikationsfähigkeit
- Anforderungen und Argumentationen des Kunden protokollieren, zusammenfassen und zur Förderung der Entscheidungsfindung in die Diskussion einbringen können
- Lücken in den Kundenanforderungen hinsichtlich der Realisierung der ausgewählten Lösungsskizzen erkennen können
- die Anforderungen und Lösungsskizzen mit dem Kunden vervollständigen und eindeutig beschreiben können

Werkzeuge/Methoden

- Präsentationstechniken
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Moderationstechniken
- Interviewtechniken
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.5.3 Beispiel: Auswählen der zu realisierenden Lösungsskizze mit dem Kunden

Da es im Beispielprojekt „Die Teltra Televisite“ keinen externen Kunden gab, wurden die Präsentation der Ergebnisse und die Entscheidungsfindung mit dem Management der Firma durchgeführt. Die zu realisierende Lösungsskizze wurde zusammen auf Basis einer Kosten-Nutzen-Analyse ausgewählt. Daraufhin präziserte das Entwicklerteam die Lösungsskizze und Anforderungen unter Berücksichtigung der Rückmeldungen von Patienten und Ärzten aus den Tests weiter.

3.1.4.6 Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte

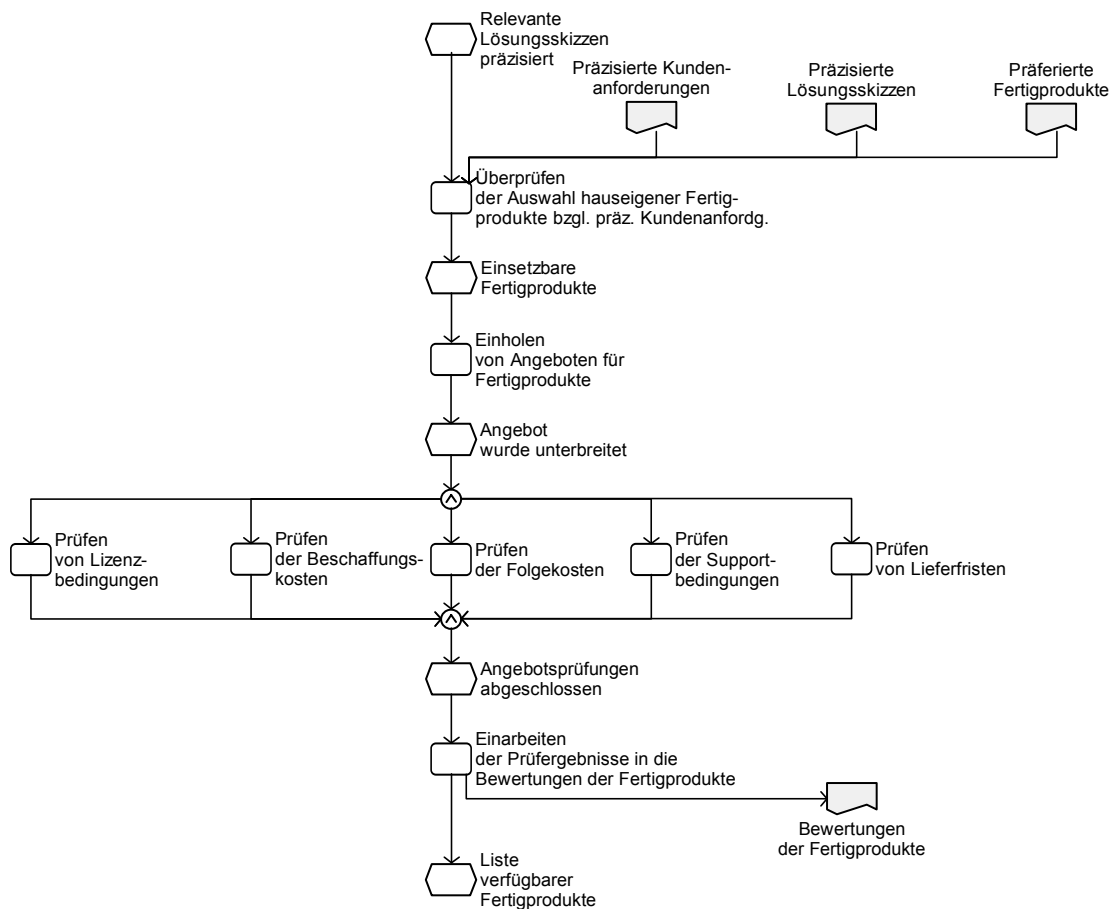


Abbildung 15: Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte.

Ziel des Teilprozesses „Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte“ ist, auf Grundlage der präzisierten Lösungsskizzen und Kundenanforderungen sowie der vom Kunden präferierten Fertigprodukte Angebote für diese Fertigprodukte einzuholen, zu bewerten und eine Liste potenziell verfügbarer Fertigprodukte zusammenzustellen. Die eingeholten Angebote werden hinsichtlich ihrer Lizenzbedingungen, der Beschaffungskosten, der Folgekosten, der Support-Bedingungen und der Lieferfristen geprüft. Die Ergebnisse dieser Prüfung werden in die Bewertung der Fertigprodukte eingearbeitet.

3.1.4.6.1 Tätigkeiten: Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte

- Überprüfen der Auswahl von Fertigprodukten bzgl. der präzisierten Kundenanforderungen
- Einholen von Angeboten für Fertigprodukte
- Prüfen von Lizenzbedingungen
- Prüfen der Beschaffungskosten
- Prüfen der Folgekosten
- Prüfen der Support-Bedingungen
- Prüfen von Lieferfristen
- Einarbeiten der Prüfergebnisse in die Bewertung der Fertigprodukte

3.1.4.6.2 Kompetenzfelder: Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte

Fähigkeiten/Fertigkeiten

- die Auswahl der Fertigprodukte im Hinblick auf die Erfüllung der präzisierten Kundenanforderungen untersuchen können
- Angebote für Fertigprodukte recherchieren und anfordern können
- Beschaffungs- und Folgekosten kalkulieren können
- Lizenz- und Support-Bedingungen sowie Beschaffungs- und Folgekosten hinsichtlich der Erfüllung der präzisierten Kundenanforderungen untersuchen können
- Ergebnisse der Untersuchungen sinnvoll in die Bewertung der Fertigprodukte einfügen können

Wissen

- Markt- und Produktkenntnisse
- Fachliteratur/-zeitschriften
- Vertragsrecht
- Kostenkalkulation

Werkzeuge/Methoden

- Recherchewerkzeuge (z. B. Internet, Datenbanken)
- Tabellenkalkulation (z. B. MS Excel)
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.6.3 Beispiel: Einholen/Prüfen von Angeboten für zu erwerbende Fertigprodukte

Nach der Präzisierung der Kundenanforderungen und der Lösungsskizze hat das Entwicklerteam im Beispielprojekt die Auswahl der Fertigprodukte hinsichtlich der Erfüllung dieser Anforderungen überprüft. Die Angebote für die Fertigprodukte holten sie über die Einkaufsabteilung der Berufsgenossenschaftlichen Kliniken Bergmannsheil ein.

Bei der Auswahl der Softwareprodukte hat das Entwicklerteam größtenteils auf frei verfügbare Software gesetzt, um die Lizenzkosten zu minimieren. Dabei spielte hinsichtlich des Supports das Vorhandensein einer beständigen Entwicklergemeinschaft eine sehr große Rolle.

Bei Angeboten für Hardwarekomponenten wurden Beschaffungs-, Folgekosten und Support-Bedingungen geprüft.

Die Ergebnisse der Angebotsprüfung hat das Entwicklerteam in die Bewertung der Fertigprodukte eingearbeitet.

3.1.4.7 Prüfen von hauseigenen Fertigprodukten

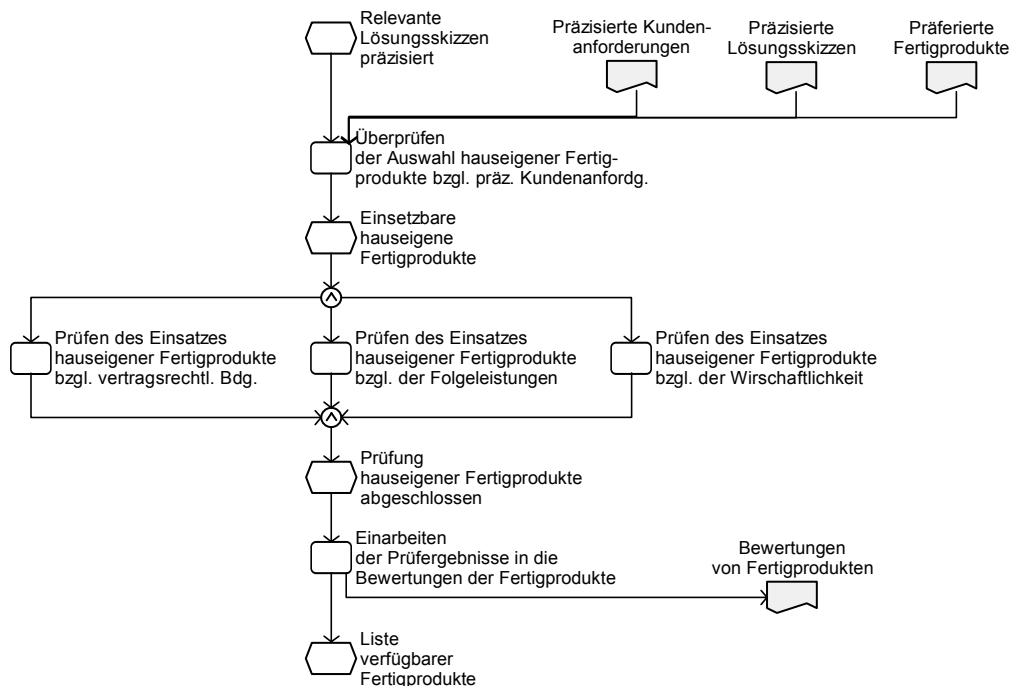


Abbildung 16: Prüfen von hauseigenen Fertigprodukten.

Neben Fertigprodukten anderer Hersteller können natürlich auch Fertigprodukte aus dem eigenen Haus bei der Systementwicklung eingesetzt werden. Ziel des Teilprozesses „Prüfen von hauseigenen Fertigprodukten“ ist, auf Grundlage der präzisierten Lösungsskizzen und Kundenanforderungen sowie der vom Kunden präferierten Fertigprodukte hauseigene Fertigprodukte hinsichtlich ihres Einsatzes zu prüfen. Dabei müssen insbesondere vertragsrechtliche Bedingungen, Folgeleistungen und die Wirtschaftlichkeit des Einsatzes berücksichtigt werden. Die Ergebnisse dieser Prüfung werden ebenfalls in die Bewertung der Fertigprodukte eingearbeitet.

3.1.4.7.1 Tätigkeiten: Prüfen von hauseigenen Fertigprodukten

- Überprüfen der Auswahl hauseigener Fertigprodukte bzgl. präzisierter Kundenanforderungen
- Prüfen des Einsatzes hauseigener Fertigprodukte bzgl. vertragsrechtlicher Bedingungen
- Prüfen des Einsatzes hauseigener Fertigprodukte bzgl. der Folgeleistungen
- Prüfen des Einsatzes hauseigener Fertigprodukte bzgl. der Wirtschaftlichkeit
- Einarbeiten der Prüfergebnisse in die Bewertung der Fertigprodukte

3.1.4.7.2 Kompetenzfelder: Prüfen von hauseigenen Fertigprodukten

Fähigkeiten/Fertigkeiten

- die Auswahl hauseigener Fertigprodukte im Hinblick auf die Erfüllung der präzisierten Kundenanforderungen untersuchen können
- vertragsrechtliche Risiken und Verpflichtungen aus dem Einsatz von hauseigenen Fertigprodukten untersuchen und bewertbar aufbereiten können

- Folgeleistungen, die aus dem Einsatz eines hauseigenen Fertigprodukts entstehen, untersuchen und kalkulieren können
- den Einsatz des hauseigenen Fertigprodukts hinsichtlich der Wirtschaftlichkeit untersuchen können
- Ergebnisse der Untersuchungen sinnvoll in die Bewertung der Fertigprodukte einfügen können

Wissen

- Vertragsrecht
- Kostenkalkulation
- betriebliche Kostenstruktur

Werkzeuge/Methoden

- Tabellenkalkulation (z. B. MS Excel)
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.7.3 Beispiel: Prüfen von hauseigenen Fertigprodukten

Im Beispielprojekt „Die Teltra Televisite“ werden für den Betrieb des Serversystems technische Kapazitäten des Rechenzentrums der Klinik genutzt. Der Betrieb des Serversystems erfolgt in Verantwortung der Firma Teltra. Das Personal des Rechenzentrums führt Maßnahmen nur auf Anweisung von Teltra durch.

In diesem Fall wurde geprüft, inwieweit hauseigene Kapazitäten, also z. B. Rechner für die Entwicklung und den Betrieb des Systems, genutzt werden konnten. Es wurde geprüft, unter welchen Bedingungen das System im Rechenzentrum der Klinik betrieben werden kann. Besonderes Augenmerk fand dabei die Sicherheit und Verfügbarkeit. So erfolgten die Administration des Systems und die Datensicherung über verschlüsselte Leitungen zum Rechenzentrum. Das System selbst darf die Verfügbarkeit anderer klinischer Systeme nicht beeinflussen.

Unter wirtschaftlichem Aspekt musste z. B. berücksichtigt werden, wie die Betriebskosten des Serversystems mit dem Rechenzentrum verrechnet werden.

- Bewerten der Standards anhand der festgelegten Kriterien
- Abstimmen der Standards mit dem Kunden
- Einarbeiten der Bewertungen der Standards in die Bewertungen der Fertigprodukte

3.1.4.8.2 Kompetenzfelder: Bewerten von existierenden Standards

Fähigkeiten/Fertigkeiten

- recherchieren und Informationen über Standards beschaffen können
- aus den Kundenanforderungen und der Anwendungsdomäne Kriterien für die Bewertung von Standards herleiten können
- Standards entsprechend ihrer Merkmalsausprägung den priorisierten und gewichteten Kriterien zuordnen und daraus eine Beurteilung ableiten können
- Kommunikationsfähigkeit
- Ergebnisse der Bewertung der Standards sinnvoll in die Bewertung der Fertigprodukte einfügen können

Wissen

- Standards und Normen der Anwendungsdomäne und der verwendeten Technologien

Werkzeuge/Methoden

- Recherchewerkzeuge (Internet, Datenbanken)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Tabellenkalkulation (z. B. MS Excel)
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.8.3 Beispiel: Bewerten von existierenden Standards

Im Beispielprojekt galt es z. B. verschiedene Übertragungsstandards wie GSM, HSCSD, GPRS und UMTS zu bewerten. Als Bewertungskriterien wurden Verfügbarkeit und Netzabdeckung, Übertragungsraten, Kosten sowie Sicherheit festgelegt. Die Ergebnisse wurden in die Bewertung z. B. der Mobilfunkkarte, aber auch des Netzproviders eingearbeitet.

Ein anderer zu berücksichtigender Aspekt ist eine mögliche Einstufung des Patientensystems als medizinisches Produkt. Für diesen Fall gelten gesetzliche Bestimmungen und Standards für eine Zertifizierung, womit wiederum Auflagen in der Herstellung des Patientensystems verbunden wären.

Standards wie z. B. HTTP, HTML, HTTPS/SSL aber auch Java wurden wegen ihrer Verbreitung in der Bewertung der Fertigprodukte berücksichtigt.

3.1.4.9 Auswählen einzusetzender Fertigprodukte

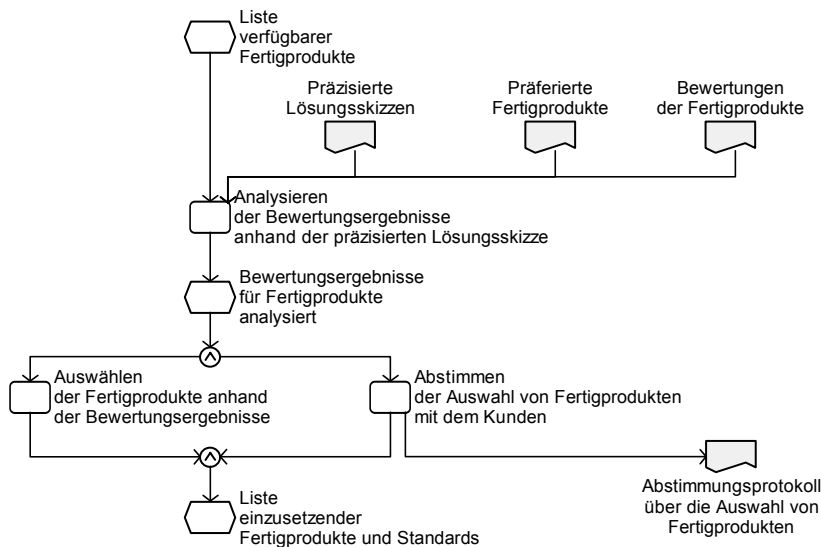


Abbildung 18: Auswählen einzusetzender Fertigprodukte.

Im Teilprozess „Auswählen einzusetzender Fertigprodukte“ werden zunächst die Bewertungsergebnisse der Fertigprodukte zusammen mit den präzisierten Lösungsskizzen analysiert. Anhand der Bewertungsergebnisse erfolgt eine Auswahl einzusetzender Fertigprodukte. Die Auswahl wird abschließend mit dem Kunden abgestimmt.

3.1.4.9.1 Tätigkeiten: Auswählen einzusetzender Fertigprodukte

- Analyse der Bewertungsergebnisse anhand der präzisierten Lösungsskizzen
- Auswahl der Fertigprodukte anhand der Bewertungsergebnisse
- Abstimmung der Auswahl von Fertigprodukten mit dem Kunden

3.1.4.9.2 Kompetenzfelder: Auswählen einzusetzender Fertigprodukte

Fähigkeiten/Fertigkeiten

- ausgehend von den präzisierten Lösungsskizzen die Bewertung der Fertigprodukte hinsichtlich aller Bewertungskriterien und -faktoren systematisch untersuchen können
- basierend auf den geprüften Bewertungsergebnissen Fertigprodukte aussuchen können
- die Auswahl der Fertigprodukte und deren Begründung dokumentieren können
- Kommunikationsfähigkeit

Werkzeuge/Methoden

- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Tabellenkalkulation (z. B. MS Excel)
- Textverarbeitung (z. B. MS Word)

- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.9.3 Beispiel: Auswählen einzusetzender Fertigprodukte

Nach der Bewertung der Standards und deren Einarbeitung in die Bewertung der einzelnen Fertigprodukte wurden im Beispielprojekt die Bewertungsergebnisse anhand der präzisierten Lösungsskizzen analysiert. Dabei geht es hauptsächlich darum, sich widersprechende Bewertungsergebnisse gegeneinander abzuwägen.

Zieht zum Beispiel die Kombination von zwei als optimal bewerteten Fertigprodukten einen höheren Entwicklungsaufwand oder mehr Risiken nach sich, ist abzuwägen, ob nicht der Einsatz einer minder bewerteten Alternative letztendlich günstiger ist. Dies kann beispielsweise auftreten, wenn die als optimal bewerteten Fertigprodukte zueinander nicht schnittstellen- oder funktionskompatibel sind.

Nach der Analyse der Bewertungsergebnisse konnte das Entwicklerteam konkrete Fertigprodukte auswählen. Die Wahl fiel z. B. für das Patientensystem auf die Digitalkamera Canon G3, ein PaceBook Tablet PC von PaceBlade mit SuSe Linux und die Mobilfunkkarte Nokia D 211.

3.1.4.10 Erstellen von technischen Lösungsvorschlägen

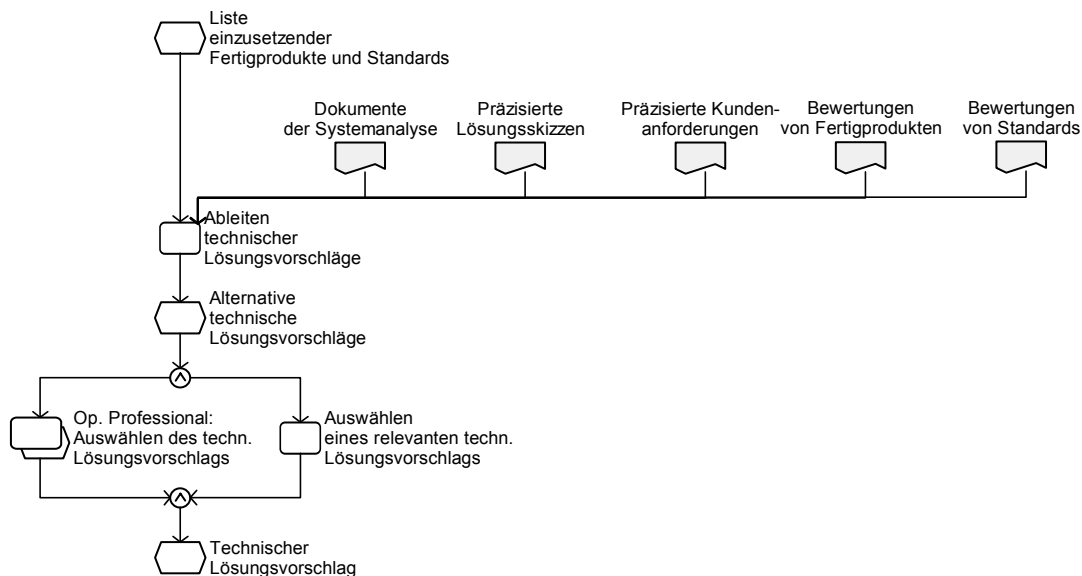


Abbildung 19: Erstellen von technischen Lösungsvorschlägen.

Im Teilprozess „Erstellen von technischen Lösungsvorschlägen“ werden die Lösungsskizzen unter Berücksichtigung der bewerteten Fertigprodukte und Standards hinsichtlich der technischen Realisierung verfeinert. Aus alternativen technischen Lösungsvorschlägen wird zusammen mit dem Projektmanagement derjenige ausgewählt, der bei der weiteren Systementwicklung umgesetzt werden sollen. Sollen bei der Systementwicklung mehrere Lösungsvorschläge berücksichtigt werden, sind die nachfolgenden Teilprozesse für alle ausgewählten Lösungsvorschläge durchzuführen.

3.1.4.10.1 Tätigkeiten: Erstellen von technischen Lösungsvorschlägen

- Ableiten technischer Lösungsvorschläge
- Auswahl eines relevanten technischen Lösungsvorschlags

3.1.4.10.2 Kompetenzfelder: Erstellen von technischen Lösungsvorschlägen

Fähigkeiten/Fertigkeiten

- Lösungsskizzen unter Berücksichtigung der einzusetzenden Fertigprodukte und Standards weiter auflösen und zusammen mit den technischen Aspekten detailliert als technische Lösungsvorschläge beschreiben können
- operative Professionals bei der Prüfung möglicher alternativer technischer Lösungsvorschläge und der Entscheidung über den zu realisierenden Lösungsvorschlag beraten können
- Kommunikationsfähigkeit

Wissen

- Standards
- technische Merkmale der Fertigprodukte

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)

3.1.4.10.3 Beispiel: Erstellen von technischen Lösungsvorschlägen

Im Beispielprojekt hat das Entwicklerteam ausgehend von der Lösungsskizze, den ausgewählten Fertigprodukten und Standards einen technischen Lösungsvorschlag erarbeitet. Er beschreibt u. a. den Aufbau des Serversystems unter Verwendung eines Apache-Web-Servers mit OpenSSL-Erweiterung zur Verschlüsselung der Daten und mit PHP-Modul zur Realisierung der Nutzerschnittstelle des Arztes und für den Zugriff auf die relationale Datenbank PostgreSQL zur Verwaltung der Patientendaten.

3.1.4.11 Zerlegen in Subsysteme

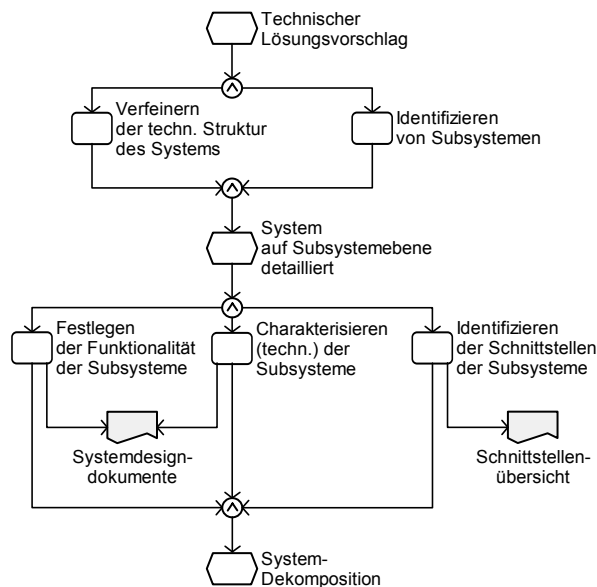


Abbildung 20: Zerlegen in Subsysteme.

Ziel der Teilprozesse „Zerlegen in Subsysteme“ und „Zerlegen in SW-/HW-Komponenten“ (S. 47) ist die funktionale Dekomposition des Systems. Ausgehend vom technischen Lösungsvorschlag wird das System zunächst auf Subsystemebene verfeinert. Es erfolgt die Identifikation der Subsysteme, das Festlegen ihrer Funktionalität, ihre technische Charakterisierung und die Identifikation ihrer Schnittstellen.

3.1.4.11.1 Tätigkeiten: Zerlegen in Subsysteme

- Verfeinern der technischen Struktur des Systems
- Identifizieren von Subsystemen
- Festlegen der Funktionalität der Subsysteme
- Charakterisieren (technisch) der Subsysteme
- Identifizieren der Schnittstellen der Subsysteme

3.1.4.11.2 Kompetenzfelder: Zerlegen in Subsysteme

Fähigkeiten/Fertigkeiten

- Systemkomponenten in weitere funktionale Einheiten auflösen und detailliert beschreiben können
- funktionale Einheiten ordnen können
- Subsysteme abgrenzen und den funktionalen Einheiten zuordnen können
- Funktionalität der identifizierten Subsysteme bestimmen und dokumentieren können
- spezifische technische Merkmale der identifizierten Subsysteme erkennen und dokumentieren können

- Schnittstellen angrenzen und den Subsystemen bzw. deren Funktionalität zuordnen können
- Schnittstellen und Zuordnung dokumentieren können

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.11.3 Beispiel: Zerlegen in Subsysteme

Ausgehend vom technischen Lösungsvorschlag wurde die technische Struktur des Systems im Beispielprojekt „Die Teltra Televisite“ zunächst bis auf die Ebene von Subsystemen verfeinert und beschrieben. Dabei erstreckt sich hier die Ebene der Subsysteme über alle Systemkomponenten, die durch Integration weiterer Systemkomponenten entstehen. Elementare, also aus Sicht des Systems nicht weiter zerlegbare Systemkomponenten werden auf der Komponentenebene näher beschrieben. Ein eindeutiges Kennzeichen zur Identifikation eines Subsystems ist z. B., wenn die Systemkomponente aus der Integration von Hardware- und Softwarekomponenten entsteht.

Im Gesamtsystem konnten zunächst auf oberster Ebene folgende Subsysteme identifiziert werden: Patienten-, Kommunikations-, Server- und Arzt-Subsystem. Das Patienten-Subsystem entsteht z. B. durch die Integration des PateBook Tablet PCs, der Digitalkamera Canon G3, der Mobilfunkkarte, des Betriebssystems und entsprechender Treiber- und Anwendungssoftware. Im Unterschied zum Patienten-Subsystem ist die Digitalkamera elementar. Sie wird aus Sicht des Systems in keine weiteren Systemkomponenten zerlegt.

3.1.4.12 Zerlegen in SW-/HW-Komponenten

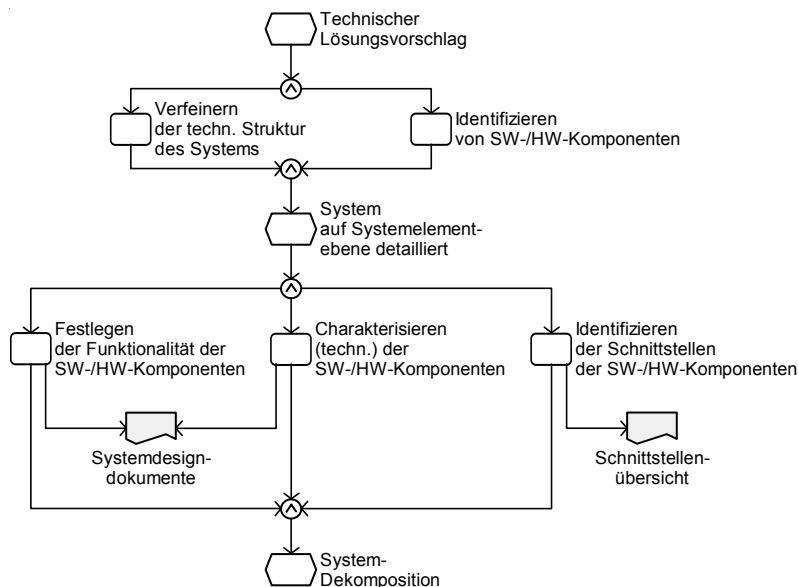


Abbildung 21: Zerlegen in SW-/HW-Komponenten.

Eine weitere Dekomposition des Systems erfolgt im Teilprozess „Zerlegen in SW-/HW-Komponenten“. Äquivalent zur Subsystemebene wird auf der Ebene der Systemelemente die Funktionalität der SW-/HW-Komponenten festgelegt; sie werden technisch charakterisiert und ihre Schnittstellen werden identifiziert.

3.1.4.12.1 Tätigkeiten: Zerlegen in SW-/HW-Komponenten

- Verfeinern der technischen Struktur des Systems
- Identifizieren von SW-/HW-Komponenten
- Festlegen der Funktionalität der SW-/HW-Komponenten
- Charakterisieren (technisch) der SW-/HW-Komponenten
- Identifizieren der Schnittstellen der SW-/HW-Komponenten

3.1.4.12.2 Kompetenzfelder: Zerlegen in SW-/HW-Komponenten

Fähigkeiten/Fertigkeiten

- Systemkomponenten in weitere funktionale Einheiten auflösen und detailliert beschreiben können
- funktionale Einheiten ordnen können
- SW-/HW-Komponenten abgrenzen und den funktionalen Einheiten zuordnen können
- die Funktionalität der identifizierten SW-/HW-Komponenten bestimmen und dokumentieren können
- spezifische technische Merkmale der identifizierten SW-/HW-Komponenten erkennen und dokumentieren können

- Schnittstellen angrenzen und den SW-/HW-Komponenten bzw. deren Funktionalität zuordnen können
- Schnittstellen und Zuordnung dokumentieren können

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.12.3 Beispiel: Zerlegen in SW-/HW-Komponenten

Ähnlich wie im Teilprozess „Zerlegen in Subsysteme“ erfolgte im Beispielprojekt die weitere Verfeinerung der identifizierten Subsysteme in Software- bzw. Hardwarekomponenten. Wie bereits beschrieben kann z. B. die Digitalkamera aus Systemsicht als elementare Hardwarekomponente betrachtet werden. Ihre Funktion, technischen Charakteristika und Schnittstellen werden demnach auf der Komponentenebene beschrieben.

Die Anwendungssoftware des Patientensystems kann z. B. hinsichtlich ihrer graphischen Nutzerschnittstelle weiter in Softwarekomponenten wie Maske und Dialogelemente zerlegt werden. Diese Verfeinerung kann notwendig werden, wenn Kundenanforderungen an das System in entsprechende Anforderungen an diese Softwarekomponenten abgebildet und beschrieben werden müssen.

3.1.4.13 Erstellen des Systemdesigns

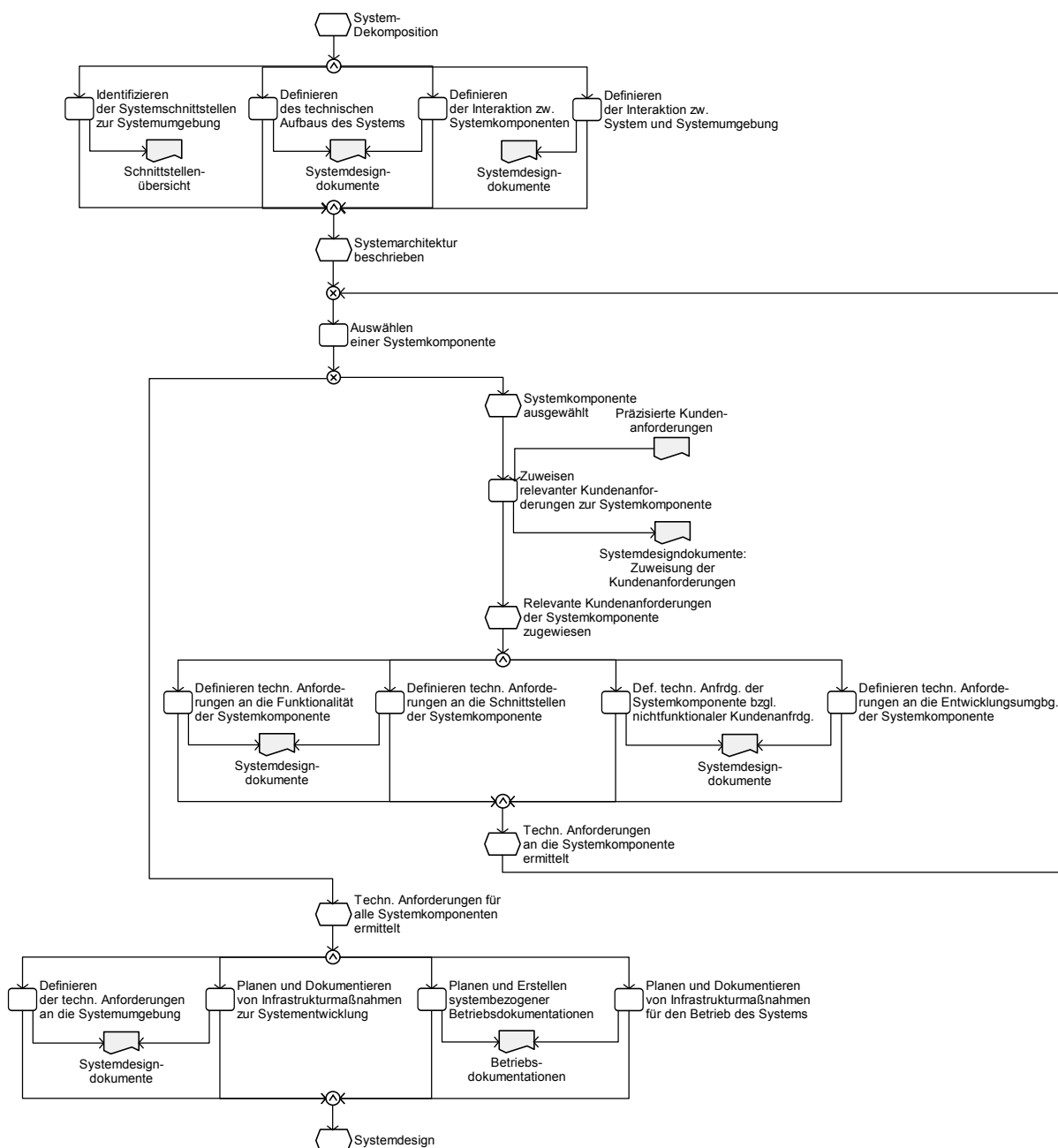


Abbildung 22: Erstellen des Systemdesigns.

Ziel des Teilprozesses „Erstellen des Systemdesigns“ ist die detaillierte Beschreibung der Systemarchitektur, die Definition der technischen Anforderungen an die Systemkomponenten und Systemumgebung, das Planen von Infrastrukturmaßnahmen für Entwicklung und Betrieb des Systems sowie das Planen und Erstellen systembezogener Dokumentationen. Zur Beschreibung der Systemarchitektur werden zunächst die Systemschnittstellen zur Systemumgebung identifiziert. Ausgehend von den relevanten technischen Lösungsvorschlägen und den festgelegten Funktionen der Systemkomponenten als Ergebnis der funktionalen Dekomposition (Abschnitt 3.1.4.11 und 3.1.4.12) erfolgt die Definition des technischen Aufbaus des Systems. Die Systemkomponenten werden zielführend strukturiert. Gleichzeitig werden die Interaktionen zwischen Systemkomponenten und die Interaktion zwischen System und Systemumgebung definiert.

Ausgehend von der Systemarchitektur werden den einzelnen Systemkomponenten relevante Kundenanforderungen zugewiesen. Die zugewiesenen Kundenanforderungen dienen der Ermittlung technischer Anforderungen an die Systemkomponente. Dabei sind insbesondere die Funktionalität, die Schnittstellen der Systemkomponente, die zugewiesenen nichtfunktionalen Kundenanforderungen und die notwendige Entwicklungsumgebung der Systemkomponente zu berücksichtigen.

Abschließend werden die technischen Anforderungen an die Systemumgebung definiert, die Betriebsdokumentation des Systems geplant und mit ihrer Erstellung begonnen sowie Infrastrukturmaßnahmen für die Systementwicklung und den Systembetrieb geplant und dokumentiert.

3.1.4.13.1 Tätigkeiten: Erstellen des Systemdesigns

- Identifizieren der Schnittstellen zur Systemumgebung
- Definieren des technischen Aufbaus des Systems
- Definieren der Interaktion zwischen Systemkomponenten
- Definieren der Interaktion zwischen System und Systemumgebung
- Auswählen einer Systemkomponente
- Zuweisen relevanter Kundenanforderungen zur Systemkomponente
- Definieren technischer Anforderungen an die Funktionalität der Systemkomponente
- Definieren technischer Anforderungen an die Schnittstellen der Systemkomponente
- Definieren technischer Anforderungen der Systemkomponente bzgl. nichtfunktionaler Kundenanforderungen
- Definieren technischer Anforderungen an die Entwicklungsumgebung der Systemkomponente
- Definieren der technischen Anforderungen an die Systemumgebung
- Planen und Dokumentieren von Infrastrukturmaßnahmen zur Systementwicklung
- Planen und Erstellen systembezogener Betriebsdokumentationen
- Planen und Dokumentieren von Infrastrukturmaßnahmen für den Betrieb des Systems

3.1.4.13.2 Kompetenzfelder: Erstellen des Systemdesigns

Fähigkeiten/Fertigkeiten

- Schnittstellen der Systemumgebung abgrenzen und ihren Diensten/Funktionen zuordnen können
- Schnittstellen und Zuordnung dokumentieren können
- den statischen Aufbau des Systems als vernetzte Struktur von Systemkomponenten und die Interaktionen zwischen diesen als auch zwischen dem System und der Systemumgebung genau bestimmen und dokumentieren können
- die Zuordnung von Kundenanforderungen zu Systemkomponenten nachweisbar dokumentieren können
- ausgehend von den zugewiesenen Kundenanforderungen und der technischen Charakterisierung einer Systemkomponente technische Anforderungen bzgl. der Funktionalität, der Schnittstellen und der nichtfunktionalen Anforderungen ableiten und dokumentieren können
- die technischen Anforderungen an die Entwicklungsumgebung einer Systemkomponenten basierend auf ihren Merkmalsausprägungen genau bestimmen und dokumentieren können

- die technischen Anforderungen an die Systemumgebung basierend auf den geforderten Diensten und ihrer Güte sowie den Merkmalsausprägungen des Systems genau bestimmen und dokumentieren können
- aus den Anforderungen an die Entwicklungsumgebung der Systemkomponenten notwendige Infrastrukturmaßnahmen für die Systementwicklung ableiten und diese entsprechend der Systemarchitektur, ihren Wechselwirkungen, Randbedingungen und Auswirkungen als logische Abfolge strukturieren können
- aus den Anforderungen des Systems an die Systemumgebung mögliche Infrastrukturmaßnahmen für den Betrieb des Systems ableiten und diese entsprechend ihren Wechselwirkungen, Randbedingungen und Auswirkungen als logische Abfolge strukturieren können
- mögliche Alternativen in der logischen Abfolge der Infrastrukturmaßnahmen bewerten und die optimale Abfolge auswählen können
- aus der optimalen logischen Abfolge von Infrastrukturmaßnahmen eine zeitliche und organisatorische Abfolge ableiten können, die z. B. die Dauer der Maßnahmen, ihre Durchführbarkeit etc. berücksichtigt
- den Aufwand der Infrastrukturmaßnahmen abschätzen können
- ausgehend von der Systemarchitektur des Systems Struktur und Schritte zur Erstellung einer Betriebsdokumentation für das System ableiten können
- die Dokumentationsschritte logisch und zeitlich strukturieren können
- den Aufwand der Erstellung der Betriebsdokumentation abschätzen können

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Methoden der Strukturierung, des Ablauf- und Terminmanagements (z. B. Netzplantechnik)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Methoden zur Aufwandsschätzung (z. B. Vergleichsmethoden)
- Textverarbeitung/Tabellenkalkulation
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.13.3 Beispiel: Erstellen des Systemdesigns

Im Beispielprojekt „Die Teltra Televisite“ wurde zur Erstellung des Systemdesigns zunächst die Systemarchitektur beschrieben. Ausgehend von der funktionalen Dekomposition des Systems wurden alle Schnittstellen zur Systemumgebung identifiziert. Im Beispielprojekt sind

das zum einen die Nutzerschnittstellen des Patienten- und Arzt-Subsystems, zum anderen die notwendigen Schnittstellen zur Einbettung des Serversystems in die Betriebsumgebung und zur Nutzung fremder Dienste. Danach wurde der technische Aufbau der Subsysteme wie Patienten-, Arzt-, Server- und Kommunikations-Subsystem detailliert beschrieben. Im Anschluss wurden die Interaktionen zwischen Systemkomponenten sowie zwischen System und Systemumgebung beschrieben. Hierzu zählt z. B. die Interaktion zwischen Digitalkamera und Tablet PC des Patienten, die Interaktion Patienten- und Server-Subsystem bzw. Arzt- und Server-Subsystem oder auch die Interaktion zwischen Web-Server und Datenbank über PHP-Skripte.

Beispiele für die Zuweisung von Kundenanforderungen zu Systemkomponenten sind die Bedienbarkeit des Patienten-Subsystems durch ältere Menschen oder dessen Einsatz im medizinischen Umfeld. Die Bedienbarkeit betrifft z. B. die Digitalkamera, den Tablet PC und die Nutzerschnittstelle der Patientenanwendung. Die Einsetzbarkeit im medizinischen Umfeld bezieht sich auf die Hardwarekomponenten wie Digitalkamera und Tablet PC.

Als Beispiel für die Ableitung von technischen Anforderungen an die Systemkomponenten aus den zugewiesenen Kundenanforderungen soll wiederum die Bedienbarkeit durch ältere Patienten dienen. Daraus resultiert die einfache und sichere Bedienung der Digitalkamera und Patientenanwendung.

Die Digitalkamera muss in einem Standardmodus immer korrekt funktionieren. Hat der Patient die Kamera versehentlich verstellt, muss er diesen Standardmodus einfach wiederherstellen können. Die Patientenanwendung muss erkennen können, dass die Digitalkamera verstellt wurde und den Patienten entsprechend darauf hinweisen. Dies sind technische Anforderungen an die Funktionalität von Digitalkamera und Patientenanwendung, die aus der allgemeinen Kundenanforderung „Bedienung durch ältere Patienten“ resultieren.

Resultierende technische Anforderungen an die Schnittstellen sind z. B. deren Robustheit und Benutzbarkeit. Ein älterer Patient muss Digitalkamera und Tablet PC mit einem USB-Kabel verbinden. Das heißt, die Steckverbindungen müssen für ihn erkennbar, fassbar und entsprechend robust sein. Ebenso müssen Schaltflächen der Nutzerschnittstelle der Patientenanwendung so groß sein, dass sie über einen Touchscreen bedient werden können. Ihre Schrift muss entsprechend groß und Fehlermeldungen verständlich sein.

Hinsichtlich der nichtfunktionalen Anforderungen müssen die eingesetzten Hardwarekomponenten z. B. ausreichend resistent gegen Erschütterungen sein. Aus dem Einsatz im medizinischen Umfeld resultiert die Forderung nach der Beständigkeit der Hardwarekomponenten gegenüber Wischdesinfektionen.

Technische Anforderungen an die Entwicklungsumgebung bei der Entwicklung der Patientenanwendung sind z. B. die Verfügbarkeit entsprechender Entwicklungshard- und -software, das Vorhandensein eines Mobilfunknetzes, eines Netzzugangs sowie der Anbindung zum Serversystem.

Bei der Definition der technischen Anforderungen an die Systemumgebung muss ebenfalls zwischen Patienten-, Server- und Arzt-Subsystem unterschieden werden. Beispiele für technische Anforderungen an die Systemumgebung des Patienten-Subsystems sind Stromversorgung und Netzabdeckung für Mobilfunk bzw. Zugang zum Festnetz.

Nach der Definition der technischen Anforderungen für alle Systemkomponenten wurden Infrastrukturmaßnahmen für die Systementwicklung geplant. Dazu gehören z. B. der Aufbau und die Einrichtung von speziellen Messgeräten zur Hardwarekalibrierung bei der Entwicklung des Farbmanagements oder auch die Schulung von Entwicklern.

Systembezogene Betriebsdokumentationen, z. B. Anwenderhandbücher für Arzt und Patient sowie das Betriebshandbuch für den Systemadministrator, wurden geplant und es wurde mit der Erstellung begonnen. Ebenso wurden Infrastrukturmaßnahmen z. B. zur Sicherung der geforderten Verfügbarkeit des Serversystems geplant und dokumentiert.

3.1.4.14 Abgleichen von Kundenanforderungen und Systemdesign

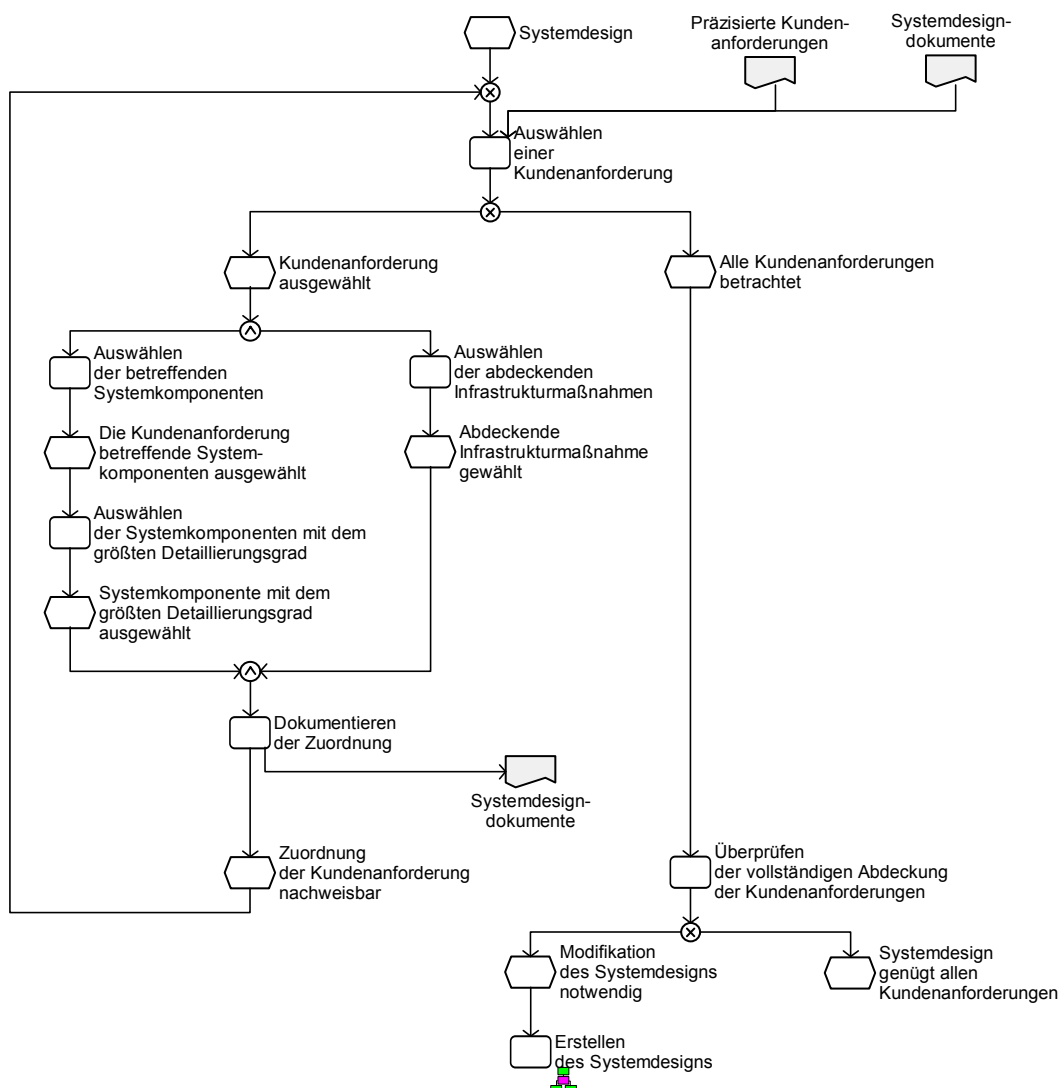


Abbildung 23: Abgleichen von Kundenanforderungen und Systemdesign.

Nach der Erstellung des Systemdesigns wird dieses mit den Kundenanforderungen abgeglichen. Dazu werden im Teilprozess „Abgleichen von Kundenanforderungen und Systemdesign“ für alle Kundenanforderungen die Systemkomponenten mit dem höchsten Detaillierungsgrad bzw. entsprechende Infrastrukturmaßnahmen ausgewählt. Die Zuordnung wird zum Nachweis dokumentiert. Abschließend wird überprüft, ob alle Kundenanforderungen vollständig abgedeckt wurden. Ist dies nicht der Fall, muss eine Modifikation des Systemdesigns erfolgen.

3.1.4.14.1 Tätigkeiten: Abgleichen von Kundenanforderungen und Systemdesign

- Auswählen einer Kundenanforderung
- Auswählen der betreffenden Systemkomponenten
- Auswählen der Systemkomponenten mit dem größten Detaillierungsgrad
- Auswählen der abdeckenden Infrastrukturmaßnahmen
- Dokumentieren der Zuordnung

- Überprüfen der vollständigen Abdeckung der Kundenanforderungen

3.1.4.14.2 Kompetenzfelder: Abgleichen von Kundenanforderungen und Systemdesign

Fähigkeiten/Fertigkeiten

- einer Kundenanforderung zugeordnete Systemkomponenten und Infrastrukturmaßnahmen aus den Dokumenten des Systemdesigns ermitteln können
- relevante Systemkomponenten entsprechend ihrem Detaillierungsgrad ordnen und die Systemkomponenten mit dem größten Detaillierungsgrad separieren können
- Systemkomponenten und Infrastrukturmaßnahmen im Hinblick auf die Erfüllung der Kundenanforderung untersuchen und den Grad der Erfüllung der Kundenanforderung nachweisbar dokumentieren können
- die Erfüllung der Kundenanforderung durch das Systemdesign im Hinblick auf Vollständigkeit untersuchen und dokumentieren können
- eventuell nicht vollständig abgedeckte Kundenanforderungen explizit dokumentieren können

Wissen

- Verständnis der Dokumente der Systemanalyse und des Systemdesigns

Werkzeuge/Methoden

- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.14.3 Beispiel: Abgleichen von Kundenanforderungen und Systemdesign

Dieser Teilprozess dient auch im Beispielprojekt zur Überprüfung des Systemdesigns. Ziel ist nachweislich festzustellen, ob alle Kundenanforderungen vom Systemdesign erfüllt werden.

So kann z. B. ermittelt werden, ob wirklich alle Hardwarekomponenten des Patienten-Subsystems mit denen der Patient in Kontakt kommt, beständig gegen Wischdesinfektionen sind.

Durch die dokumentierte Zuordnung der Kundenanforderungen wird deutlich, wenn Kundenanforderungen vom Systemdesign nicht berücksichtigt wurden.

3.1.4.15 Untersuchen der Realisierbarkeit

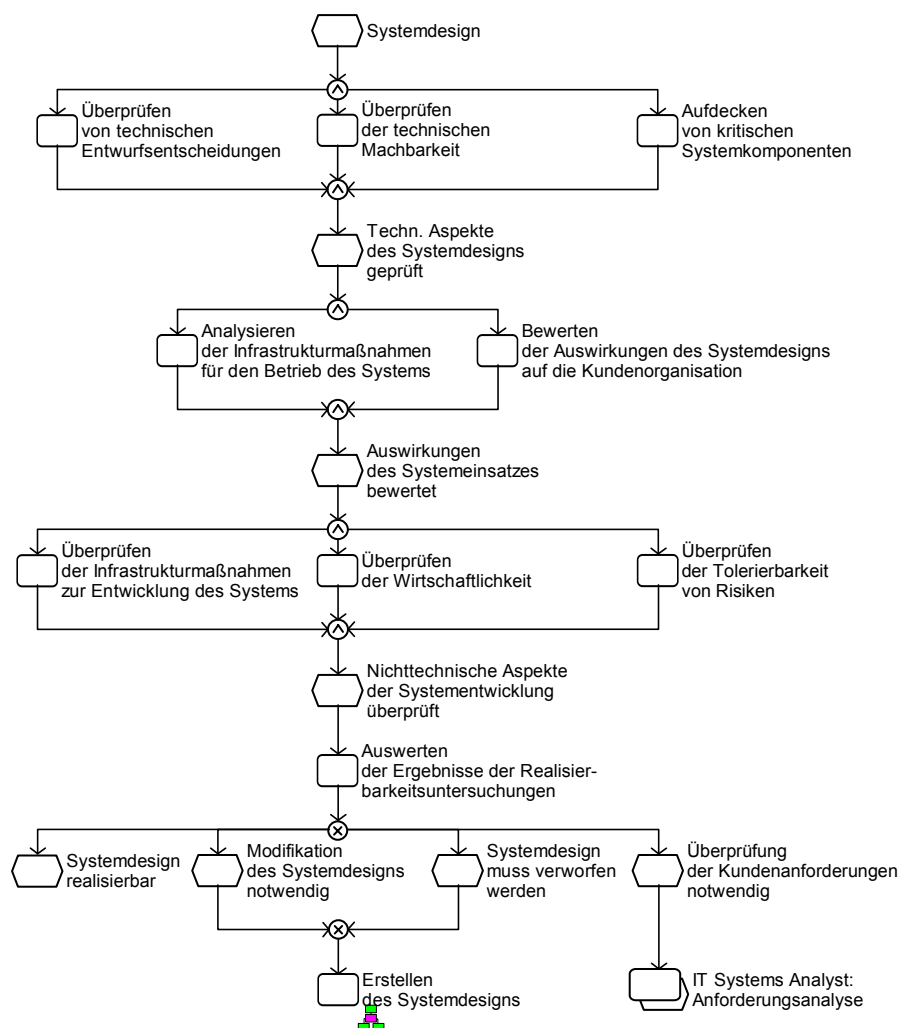


Abbildung 24: Untersuchen der Realisierbarkeit.

Ziel des Teilprozesses „Untersuchen der Realisierbarkeit“ ist, das erstellte Systemdesign hinsichtlich seiner Realisierbarkeit zu überprüfen und im Ergebnis zu entscheiden, ob das Systemdesign umsetzbar ist, modifiziert oder verworfen werden muss oder ob womöglich eine Überprüfung der Kundenanforderungen notwendig ist.

Dazu werden zunächst die technischen Aspekte des Systemdesigns überprüft. Dazu gehören insbesondere die technischen Entwurfsentscheidungen und die technische Machbarkeit des Systemdesigns. Kritische Systemkomponenten sollten dabei aufgedeckt und ihr Risiko beschrieben werden.

Danach werden die Auswirkungen des Systemeinsatzes bewertet. Dazu erfolgt eine Analyse der notwendigen Infrastrukturmaßnahmen für den Betrieb des Systems und die Bewertung der Auswirkungen des Systemdesigns auf die Kundenorganisation.

Im Weiteren werden die notwendigen Infrastrukturmaßnahmen zur Entwicklung des Systems, die Wirtschaftlichkeit der Systementwicklung und die Tolerierbarkeit aufgezeigter Risiken überprüft.

Abschließend werden die Ergebnisse der Realisierbarkeitsuntersuchungen ausgewertet und über das weitere Vorgehen entschieden.

3.1.4.15.1 Tätigkeiten: Untersuchen der Realisierbarkeit

- Überprüfen von technischen Entwurfsentscheidungen

- Überprüfen der technischen Machbarkeit
- Aufdecken von kritischen Systemkomponenten
- Analyse der Infrastrukturmaßnahmen für den Betrieb des Systems
- Bewerten der Auswirkungen des Systemdesigns auf die Kundenorganisation
- Überprüfen der Infrastrukturmaßnahmen zur Entwicklung des Systems
- Überprüfen der Wirtschaftlichkeit
- Überprüfen der Tolerierbarkeit von Risiken
- Auswerten der Ergebnisse der Realisierbarkeitsuntersuchungen

3.1.4.15.2 Kompetenzfelder: Untersuchen der Realisierbarkeit

Fähigkeiten/Fertigkeiten

- technische Entwurfsentscheidungen hinsichtlich der Erfüllung der Kundenanforderungen untersuchen können
- einzusetzende Fertigprodukte des Systemdesigns hinsichtlich der geforderten technischen Eigenschaften und Schnittstellen sowie ihrer Anpassbarkeit untersuchen können
- technische Anforderungen an zu entwickelnde Systemkomponenten des Systemdesigns hinsichtlich ihrer Realisierbarkeit unter Berücksichtigung der Entwicklungs- und Betriebsumgebung untersuchen können
- Risiken bei der Anpassung und Integration von Fertigprodukten bzw. bei der Neuentwicklung von Systemkomponenten identifizieren, bewerten und dokumentieren können
- Risiken beim Betrieb des Systems und seiner Systemkomponenten in der Kundenumgebung identifizieren, bewerten und dokumentieren können
- geforderte Infrastrukturmaßnahmen beim Betrieb des Systems hinsichtlich der Abwendung aufgezeigter Risiken, ihrer Durchsetzbarkeit und Auswirkungen auf die Kundenorganisation untersuchen, bewerten und die Ergebnisse dokumentieren können
- Auswirkungen der geforderten Systemumgebung auf die Kundenorganisation identifizieren, bewerten und dokumentieren können
- geforderte Infrastrukturmaßnahmen bei der Entwicklung des Systems hinsichtlich der Abwendung aufgezeigter Risiken, ihrer Durchsetzbarkeit und Auswirkungen auf die Entwicklungsorganisation untersuchen, bewerten und die Ergebnisse dokumentieren können
- alle anfallenden Kosten bei Entwicklung und Einsatz des Systems identifizieren und monetär bewerten können
- monetär bewertete Kosten dem Kostenrahmen für Entwicklung und Betrieb gegenüberstellen sowie daraus eine Beurteilung der Wirtschaftlichkeit ableiten und dokumentieren können
- aufgezeigte Risiken unter Berücksichtigung der festgelegten Maßnahmen zu deren Abwendung hinsichtlich des Verbleibs und der Tolerierbarkeit von Restrisiken untersuchen können
- Ergebnisse aus den verschiedenen Untersuchungen zur Realisierbarkeit des Systemdesigns zu einer Beurteilung zusammenfassen und eventuelle Maßnahmen zur Modifikation des Systemdesigns ableiten und dokumentieren können

Wissen

- Technologiekenntnisse, Kriterien und Normen der Anwendungsdomäne
- Technologiekenntnisse verschiedener Aspekte eines Systems (z. B. Kommunikationsprotokolle, Persistenz, Standards)
- Musterarchitekturen
- einzusetzende Fertigprodukte, deren Lizenz-/Support-Bedingungen und Gewährleistungen der Hersteller

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Methoden des Risikomanagements (z. B. SWOT-Analyse, Impact/Probability Matrix)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Methoden zur Aufwandsschätzung (Function-Point-Methode, Constructive Cost Model COCOMO, Vergleichsmethoden)
- Textverarbeitung/Tabellenkalkulation
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.15.3 Beispiel: Untersuchen der Realisierbarkeit

Um den Tablet PC des Patienten-Subsystems im Beispielprojekt „Die Teltra Televisite“ besser gegen Erschütterungen zu schützen, Fehlbedienungen zu verhindern und die Wirksamkeit der Wischdesinfektion zu verbessern, hat das Entwicklerteam im Systemdesign eine zusätzliche Plastikvorrichtung für den Tablet PC entworfen. Diese schützt den Tablet PC vor Erschütterungen, deckt unnötige Anschlüsse ab, ermöglicht eine bessere Beschriftung und bietet eine geschlossene, glatte Oberfläche. Diese Entwurfsentscheidung wurde zusammen mit Spezialisten hinsichtlich möglicher Alternativen und der technischen Machbarkeit überprüft.

Als kritische Systemkomponenten sind beispielsweise alle Hardwarekomponenten einzuordnen, mit denen der genesende Patient in Kontakt kommt. Diese Hardwarekomponenten stellen bei Patienten mit offenen Wunden ein potenzielles Infektionsrisiko dar. Ebenfalls kann eine Fehlfunktion der Digitalkamera zur Verfälschung der Aufnahmen der Wundbereiche führen, was u. U. Fehldiagnosen des Arztes nach sich zieht. Hier müssen die Maßnahmen zur Reduzierung dieser Risiken überprüft werden. Auch stellt sich die Frage, in welchen Krankheitssituationen die Anwendung des Systems mit allen damit verbundenen Maßnahmen dem Patienten überhaupt zumutbar ist.

Hinsichtlich der Überprüfung der Wirtschaftlichkeit des Systems wurde untersucht, ob die Anschaffung und der Betrieb des Systems, also die häusliche Pflege des Patienten unter Anwendung des Systems gegenüber der stationären Pflege Kostenvorteile verschafft.

Mögliche Risiken wie z. B. der Ausfall des Systems oder das Verschlechtern des Allgemeinzustands des Patienten sind durch die Verfügbarkeit der klassischen ärztlichen Versorgung wie z. B. durch den Hausarzt oder durch Pflegestationen tolerierbar.

3.1.4.16 Beschreiben der Schnittstellen

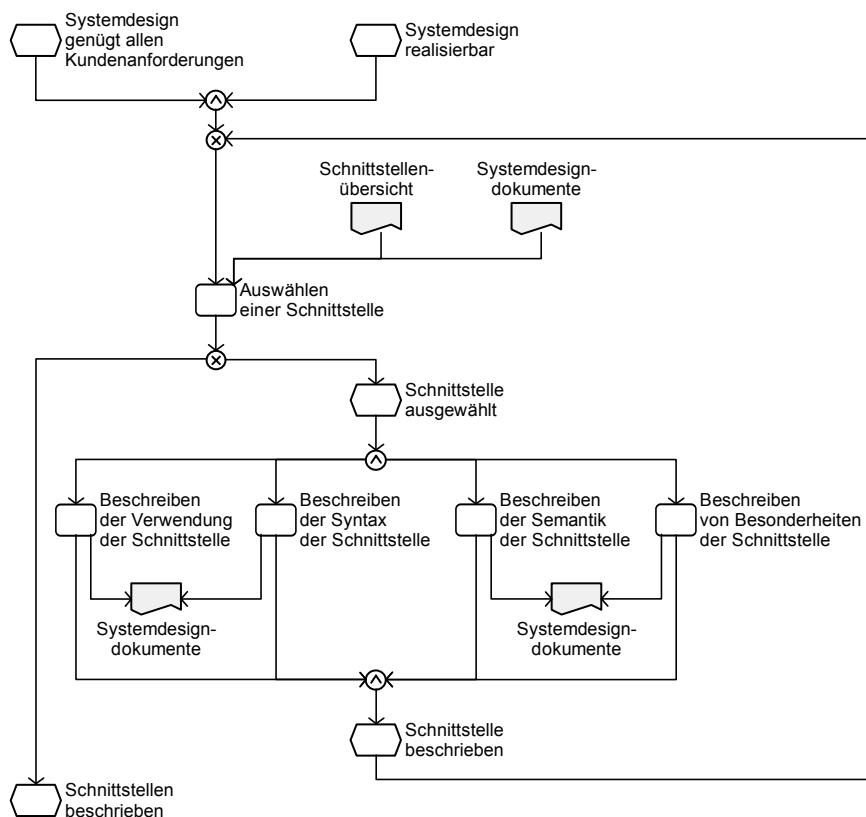


Abbildung 25: Beschreiben der Schnittstellen.

Nachdem ermittelt wurde, dass das erstellte Systemdesign allen Kundenanforderungen genügt und realisierbar ist, wird es im Teilprozess „Beschreiben der Schnittstellen“ um detaillierte Schnittstellenbeschreibungen ergänzt.

Für jede identifizierte Schnittstelle werden die Verwendung, die Syntax, die Semantik und mögliche Besonderheiten detailliert beschrieben.

3.1.4.16.1 Tätigkeiten: Beschreiben der Schnittstellen

- Auswählen einer Schnittstelle
- Beschreiben der Verwendung der Schnittstelle
- Beschreiben der Syntax der Schnittstelle
- Beschreiben der Semantik der Schnittstelle
- Beschreiben von Besonderheiten der Schnittstelle

3.1.4.16.2 Kompetenzfelder: Beschreiben der Schnittstellen

Fähigkeiten/Fertigkeiten

- Syntax, Semantik und Verwendung von Schnittstellen entsprechend den Vorgaben für das Systemdesign formal, semiformal bzw. informal dokumentieren können
- Besonderheiten einer Schnittstelle eindeutig dokumentieren können

Werkzeuge/Methoden

- formale, semiformale, informale Spezifikationsverfahren (z. B. Z, UML, Statecharts)
- Textverarbeitung/Tabellenkalkulation
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.16.3 Beispiel: Beschreiben der Schnittstellen

Ein Beispiel für eine detailliert zu beschreibende Schnittstelle aus dem Projekt „Die Teltra Televisite“ ist die Web-Schnittstelle zwischen Server- und Arzt-Subsystem. Hier ist besonders wichtig, die Syntax und Semantik der Parametrisierung der HTTP-Requests der Web-Oberfläche zu dokumentieren. Diese Schnittstellen und deren Aufrufe werden in der Regel bei der Entwicklung nicht durch einen Compiler oder Ähnliches auf Gültigkeit, Kompatibilität und Konsistenz geprüft, sondern müssen zur Laufzeit durch die PHP-Skripte validiert werden. Manipulationen an den URLs und damit fehlerhafte Aufrufe lassen sich nicht ausschließen.

3.1.4.17 Spezifizieren der Integration und Systemtests

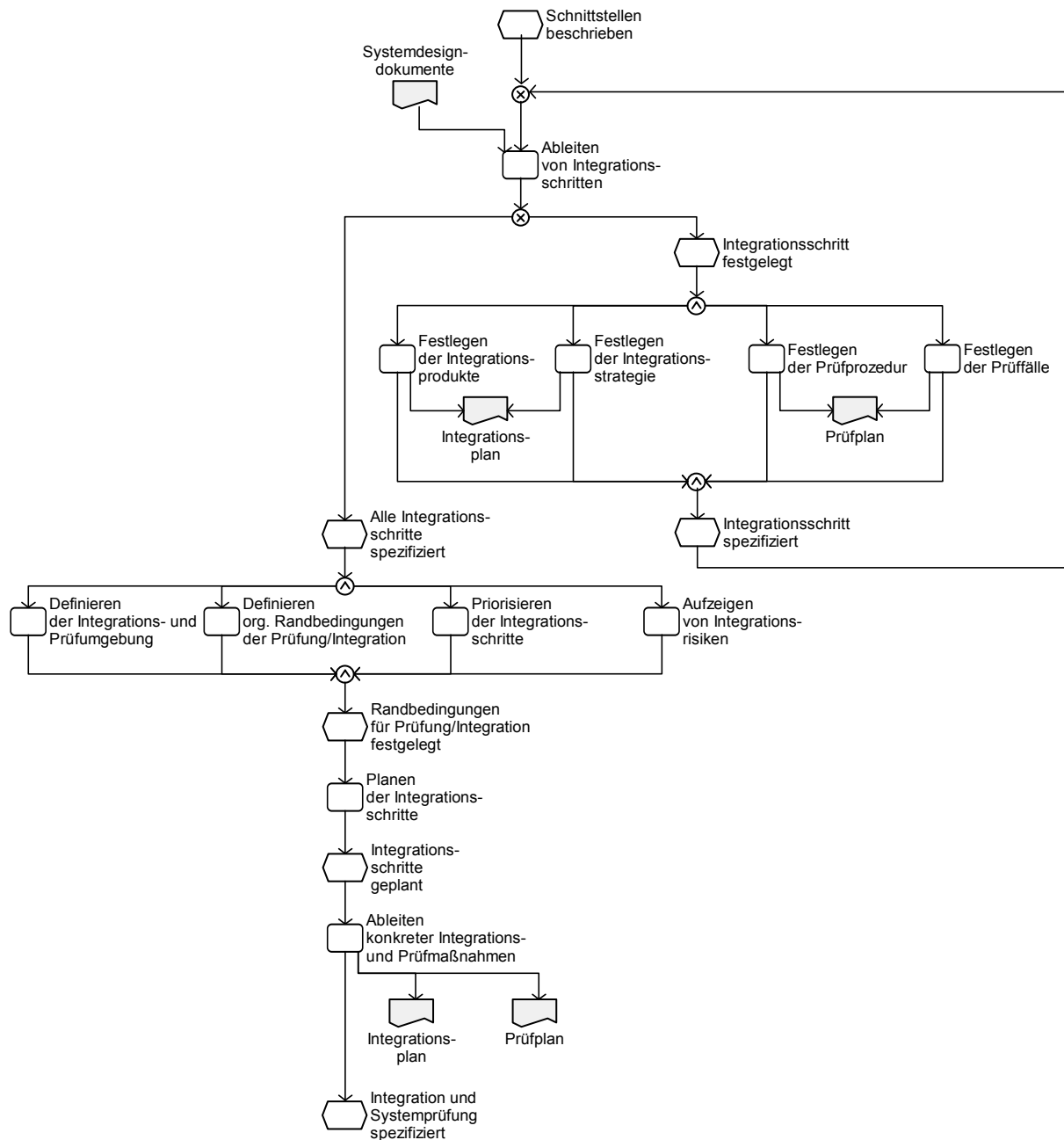


Abbildung 26: Spezifizieren der Integration und Systemtests.

Ausgehend von der Systemarchitektur, die im Ergebnis der funktionalen Dekomposition des Systems definiert wurde, wird in diesem Teilprozess die Komposition des Systems spezifiziert, d. h. die Integration von Systemkomponenten und Überprüfung des Ergebnisses.

Zunächst werden aus der Systemarchitektur die notwendigen Integrationsschritte zur Komposition des Systems abgeleitet. Für jeden Integrationsschritt werden die Integrationsprodukte und die Integrationsstrategie sowie die Prüfprozedur und Prüffälle für die Prüfung der Integrationsprodukte und des Ergebnisses des Integrationsschritts festgelegt.

Nach der Spezifikation aller Integrationsschritte werden die Integrations- und Prüfumgebungen sowie die organisatorischen Randbedingungen der Integration und

Prüfung definiert. Die Integrationsschritte werden priorisiert und mögliche Integrationsrisiken aufgezeigt.

Danach können die Integrationsschritte geplant, konkrete Integrations- und Prüfmaßnahmen abgeleitet und im Integrations- bzw. Prüfplan dokumentiert werden.

3.1.4.17.1 Tätigkeiten: Spezifizieren der Integration und Systemtests

- Ableiten von Integrationsschritten
- Festlegen der Integrationsprodukte
- Festlegen der Integrationsstrategie
- Festlegen der Prüfprozedur
- Festlegen der Prüffälle
- Definieren der Integrations- und Prüfumgebung
- Definieren organisatorischer Randbedingungen der Integration/Prüfung
- Priorisieren der Integrationsschritte
- Aufzeigen von Integrationsrisiken
- Planen der Integrationsschritte
- Ableiten konkreter Integrations- und Prüfmaßnahmen

3.1.4.17.2 Kompetenzfelder: Spezifizieren der Integration und Systemtests

Fähigkeiten/Fertigkeiten

- ausgehend vom Systemdesign schrittweise Integration der Systemkomponenten entwickeln und dokumentieren können
- Produkt und Strategie der Integrationsschritte dokumentieren können
- auf Grundlage des Systemdesigns und der aufgezeigten Integrationsschritte Prüfprozeduren und Prüffälle für zu integrierende Systemkomponenten und die Integrationsprodukte entwickeln und dokumentieren können
- basierend auf den konkreten Merkmalsausprägungen der Systemkomponenten und Integrationsprodukte sowie den Erfordernissen der Integrationsstrategie und Prüfprozedur die Anforderungen an die Integrations- und Prüfumgebung sowie organisatorische Maßnahmen und Randbedingungen genau bestimmen und dokumentieren können
- die Integrationsschritte und Prüfprozeduren entsprechend ihren Wechselwirkungen, Randbedingungen und Auswirkungen ordnen und als logische Abfolge strukturieren können
- Risiken bei der Integration von Systemkomponenten identifizieren, bewerten und dokumentieren können
- mögliche Alternativen in der logischen Abfolge bewerten und die optimale Alternative auswählen können
- Aufwände zur Integration und Prüfung von Systemkomponenten abschätzen können
- aus der logischen Abfolge der Integration und Prüfung eine zeitliche und organisatorische Abfolge ableiten können, die die Aufwandsabschätzungen zur Integration und Prüfung der Systemkomponenten, Informationen über die Erfüllung aufgezeigter organisatorischer Randbedingungen und die Aufwände zur Durchführung entsprechender Maßnahmen berücksichtigt
- logische, zeitliche und organisatorische Abfolge der Integration und Prüfung in geeigneter Form dokumentieren können

- auf Grundlage der Integrationsschritte, der Integrationsstrategie, der Prüfprozedur und der Prüffälle konkrete Anleitungen bzw. Vorschriften zum Zusammenbau von Systemkomponenten und des Systems ausarbeiten und dokumentieren können

Wissen

- Betriebsorganisation, -strukturen

Werkzeuge/Methoden

- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (z. B. funktionaler und struktureller Test)
- Black-Box-Testfallentwurfs-Methoden (z. B. Äquivalenzklassenbildung, Grenzwertanalyse, Intuitive Testfallermittlung, Funktionsabdeckung)
- White-Box-Testfallentwurf
- Methoden der Strukturierung, des Ablauf- und Terminmanagements (z. B. Netzplantechnik)
- Methoden des Risikomanagements (z. B. SWOT-Analyse, Impact/Probability Matrix)
- Methoden zur Darstellung und Beschreibung von Varianten (z. B. Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (z. B. gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Methoden zur Aufwandsschätzung (z. B. Vergleichsmethoden)
- Textverarbeitung/Tabellenkalkulation (z. B. MS Word, MS Excel)
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.17.3 Beispiel: Spezifizieren der Integration und Systemtests

Im Beispielprojekt „Die Teltra Televisite“ wurden auf der Grundlage der Systemarchitektur Integrationsprodukte, -schritte und -strategien nach dem Bottom-Up-Verfahren festgelegt.

Ein Integrationsprodukt des Systems bzw. des Patienten-Subsystems ist z. B. der betriebsbereite Tablet PC. Dazu muss auf dem Tablet PC zunächst das Betriebssystem Linux installiert werden. Danach werden alle notwendigen Betriebssystemfunktionen getestet.

Ist der Tablet PC betriebsbereit, können die Mobilfunkkarte angeschlossen und die zugehörigen Treiber installiert werden. Zur Prüfung der Übertragung von Daten über ein Mobilfunknetz müssen entsprechende Prüffälle und -prozeduren festgelegt und Prüfdaten erzeugt werden. Außerdem muss eine entsprechende Prüfumgebung geplant werden, da die Daten zur Prüfung in irgendeiner Weise wieder empfangen werden müssen.

An einen betriebsbereiten Tablet PC kann ebenfalls die Digitalkamera angeschlossen und die zugehörige Software zum Übertragen der Bilder installiert werden. Es müssen Prüffälle und -prozeduren festgelegt werden, um das Aufnehmen von Bildern und Übertragen auf den Tablet PC zu testen.

Ist die Hardware des Patienten-Subsystems integriert, kann die Anwendungssoftware installiert und getestet werden. Auch hierfür werden Prüffälle und -prozeduren festgelegt.

Ist das Patienten-Subsystem vollständig integriert und getestet, kann es mit dem Server-Subsystem integriert werden und so weiter. Der letzte Integrationsschritt liefert als Integrationsprodukt das Gesamtsystem. Auch für den Systemtest wurden Prüffälle und Prüfprozeduren festgelegt.

Sind alle Integrationsschritte mit ihren Abhängigkeiten spezifiziert, werden die notwendigen Integrations- und Prüfumgebungen definiert. Im Beispielprojekt legt z. B. eine spezielle Software für das Farbmanagement ein Farbprofil für jede Digitalkamera und jeden Arztmonitor an. Zum Testen dieser Software muss die entsprechende Hardware betriebsbereit zur Verfügung stehen.

Organisatorische Rahmenbedingungen sind bei der Integration der Serversystems in die Betriebsumgebung des Rechenzentrums der Klinik zu definieren. Hierbei müssen auch die

Integrationsrisiken und mögliche Gegenmaßnahmen beschrieben werden, um eine Störung anderer Systeme der Klinik zu vermeiden.

Im Anschluss wurden die Integrationsschritte priorisiert und in ihrem zeitlichen und organisatorischen Ablauf geplant. Dabei mussten verschiedene Einflussgrößen wie die Aufwände für die Integration und Prüfung, die Verfügbarkeit des notwendigen Personals, die Dauer der Durchführung notwendiger organisatorischer Maßnahmen etc. berücksichtigt werden.

Auf Grundlage der Spezifikation der Integrationsschritte und ihrer Planung können konkrete Anweisungen zur Durchführung der Integration und Prüfung im Integrations- bzw. Prüfplan dokumentiert werden.

3.1.4.18 Definieren von zu erstellenden SW-/HW-Komponenten

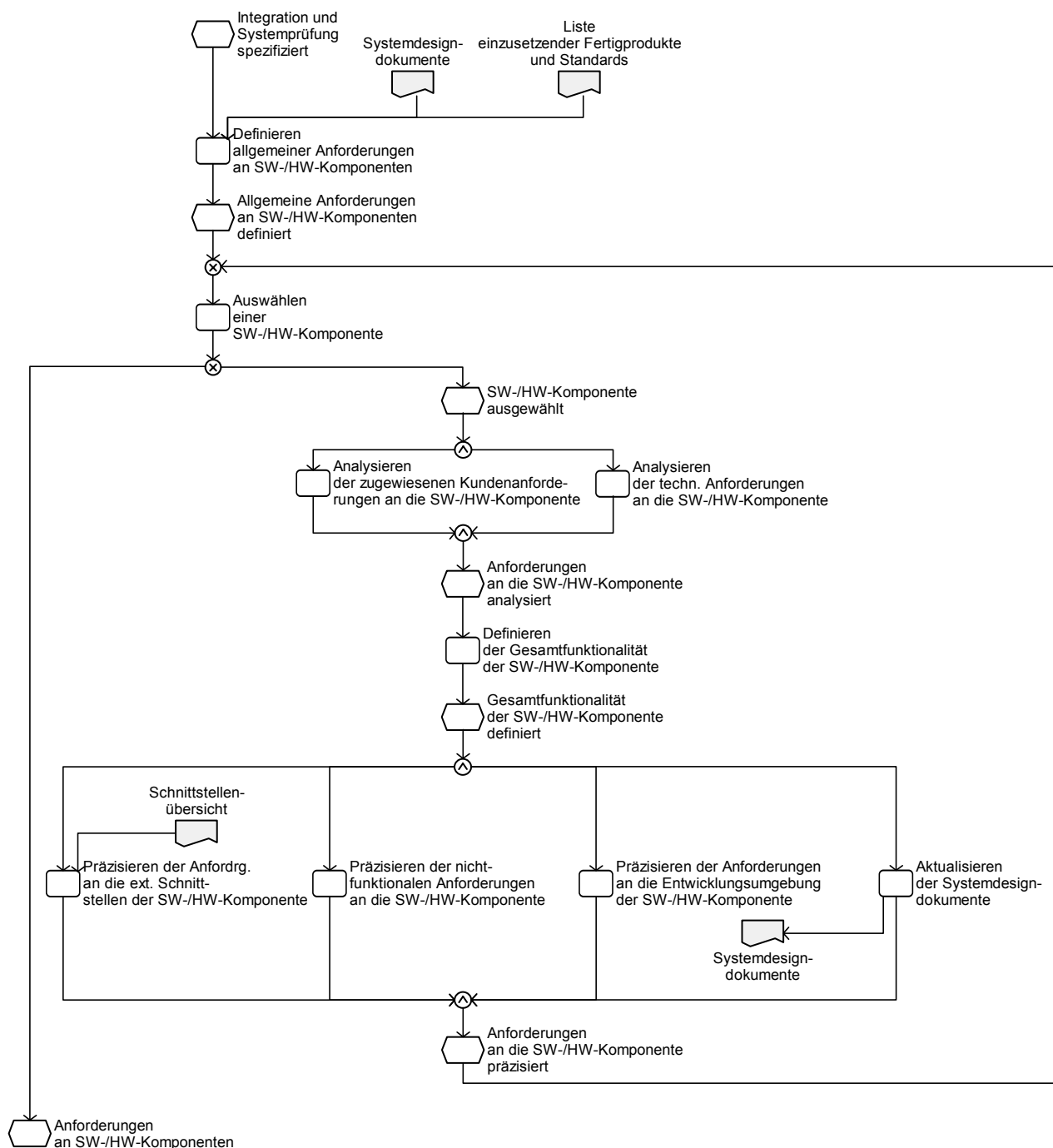


Abbildung 27: Definieren von zu erstellenden SW-/HW-Komponenten.

Ziel des Teilprozesses „Definieren von zu erstellenden SW-/HW-Komponenten“ ist das Definieren und Präzisieren aller Anforderungen, die für die Entwicklung bzw. Konstruktion der SW-/HW-Komponenten des Systems erforderlich sind.

Zunächst werden allgemeine Anforderungen wie z. B. zu berücksichtigende Standards an die SW-/HW-Komponenten definiert.

Ausgehend von den zugewiesenen Kunden- und technischen Anforderungen sowie der festgelegten Anwendungsfunktionalität wird danach die Funktionalität jeder SW-/HW-Komponente in ihrer Gesamtheit (also z. B. technische Funktionalität, Betriebsarten, Reaktionen im Fehlerfall etc.) definiert.

Im Anschluss werden für jede SW-/HW-Komponente die Anforderungen an die externen Schnittstellen, die nichtfunktionalen Anforderungen und die Anforderungen an die Entwicklungsumgebung präzisiert.

Alle Definitionen und Präzisierungen werden in die Systemdesign-Dokumente eingearbeitet.

3.1.4.18.1 Tätigkeiten: Definieren von zu erstellenden SW-/HW-Komponenten

- Definieren allgemeiner Anforderungen an SW-/HW-Komponenten
- Auswählen einer SW-/HW-Komponente
- Analysieren der zugewiesenen Kundenanforderungen an die SW-/HW-Komponente
- Analysieren der technischen Anforderungen an die SW-/HW-Komponente
- Definieren der Gesamtfunktionalität der SW-/HW-Komponente
- Präzisieren der Anforderungen an die externen Schnittstellen der SW-/HW-Komponente
- Präzisieren der nichtfunktionalen Anforderungen an die SW-/HW-Komponente
- Präzisieren der Anforderungen an die Entwicklungsumgebung der SW-/HW-Komponente
- Aktualisieren der Systemdesign-Dokumente

3.1.4.18.2 Kompetenzfelder: Definieren von zu erstellenden SW-/HW-Komponenten

Fähigkeiten/Fertigkeiten

- einer ausgewählten Systemkomponente zugewiesene Anforderungen (Kundenanforderungen, allgemeine und technische Anforderungen) aus dem Systemdesign hinsichtlich der Implementierung bzw. Konstruktion der Systemkomponenten untersuchen und strukturieren können
- ausgehend von den allgemeinen Anforderungen, den zugewiesenen Kundenanforderungen und den technischen Anforderungen konkrete funktionale und nichtfunktionale Anforderungen an die SW-/HW-Systemkomponente und ihre externen Schnittstellen ermitteln bzw. ableiten, modellieren und dokumentieren können
- Anforderungen an die SW-/HW-Entwicklungsumgebung entsprechend der konkreten Merkmalsausprägung der Systemkomponenten genau bestimmen und dokumentieren können

Wissen

- Standards und Technologien der Anwendungsdomäne und der technischen Realisierung

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Hardware-Beschreibungssprachen (z. B. VHDL)

- Textverarbeitung/Tabellenkalkulation
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.18.3 Beispiel: Definieren von zu erstellenden SW-/HW-Komponenten

Beispiele für allgemeine Anforderungen aus dem Beispielprojekt sind die Verwendung von Java für die Patientenapplication und von PHP bei der Implementierung des Serversystems.

Danach werden die zugewiesenen Kundenanforderungen und technischen Anforderungen an die SW-/HW-Komponente analysiert, ihre Gesamtfunktionalität definiert und die Anforderungen an die externen Schnittstellen präzisiert.

Aus der Kundenanforderung „Älterer Patient soll Arzt Nachrichten senden können“ kann man beispielsweise eine Dialogkomponente mit folgender Gesamtfunktionalität ableiten: Die Dialogkomponenten müssen die Eingabe des Texts der Nachricht, das Ändern von Text und das Löschen von Text realisieren. Außerdem muss eine Schaltfläche zum Versenden der Nachricht und zum Abbruch des Vorgangs vorhanden sein. Die Dialogkomponenten müssen den Patient über den Fortschritt des Versands und mögliche Fehler geeignet informieren. Zusätzlich ist der Zugang zur Hilfe notwendig.

Weiterhin lassen sich die Anforderungen an die externen Schnittstellen der Dialogkomponente, wozu auch die Nutzerschnittstelle gehört, präzisieren. Die Maße der Darstellung der Dialogkomponenten sind z. B. abhängig von der Größe und Auflösung der Displays des Tablet PCs. Dabei ist eine Schrift ohne Serifen mit einer Fontgröße von mindestens N Pixel zu verwenden, um die Lesbarkeit für ältere Patienten zu gewährleisten. Schaltflächen müssen mindestens N x N Pixel groß sein, damit sie sicher mit dem Finger über einen Touchscreen bedient werden können.

Nichtfunktionale Anforderungen beziehen sich z. B. auf die zu berücksichtigende Wiederverwendbarkeit von Dialogelementen.

Anforderungen an die Entwicklungsumgebung von SW-/HW-Komponenten lassen sich auf Grundlage der allgemeinen Anforderungen wie z. B. die zu verwendende Programmiersprache, der funktionalen und nichtfunktionalen Anforderungen sowie der Anforderungen an die externen Schnittstellen präzisieren.

Alle Anforderungen an die SW-/HW-Komponenten werden in die Systemdesign-Dokumente eingearbeitet.

3.1.4.19 Begleiten der SW-/HW-Komponentenerstellung

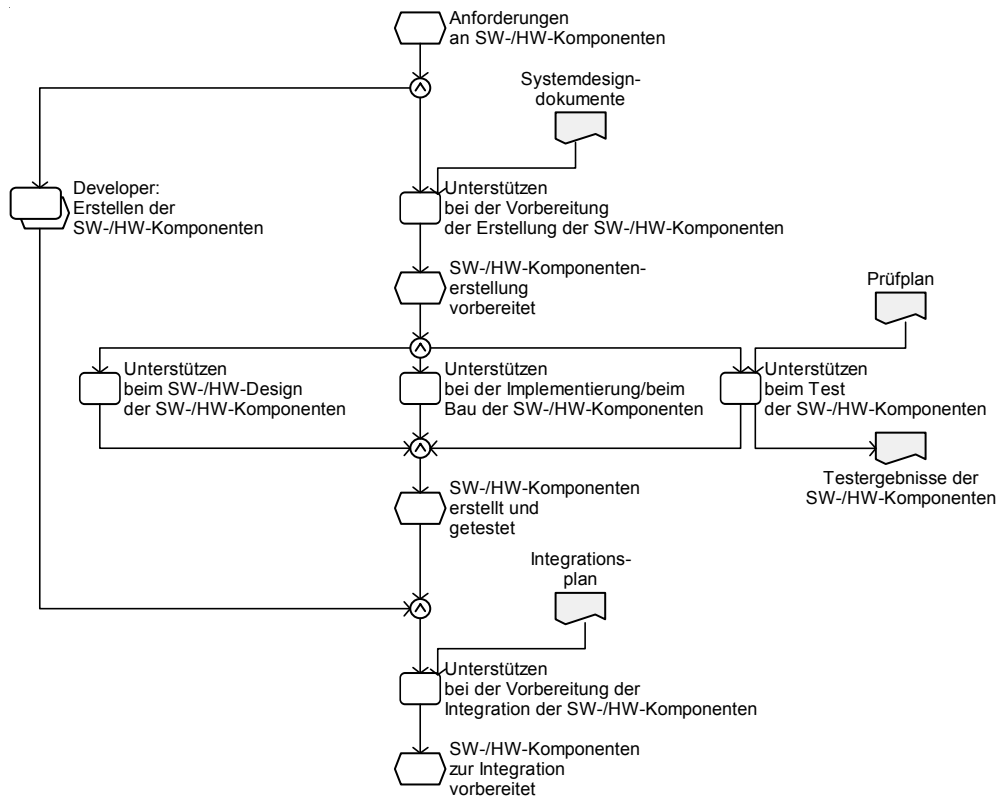


Abbildung 28: Begleiten der SW-/HW-Komponentenerstellung.

Der Teilprozess „Begleiten der SW-/HW-Komponentenerstellung“ beschreibt die Tätigkeiten des IT Systems Developer während der Erstellung der zuvor definierten SW-/HW-Komponenten. Im Wesentlichen unterstützt und berät der IT Systems Developer bei der Vorbereitung der Erstellung von SW-/HW-Komponenten, bei deren Entwurf, Implementierung bzw. Bau und Test. Durch die aktive Unterstützung bei der Vorbereitung der SW-/HW-Komponente zur Integration können Reibungsverluste bei der Systemintegration und -prüfung reduziert werden.

3.1.4.19.1 Tätigkeiten: Begleiten der SW-/HW-Komponentenerstellung

- Unterstützen bei der Vorbereitung der Erstellung der SW-/HW-Komponenten
- Unterstützen beim Design der SW-/HW-Komponenten
- Unterstützen bei der Implementierung/beim Bau der SW-/HW-Komponenten
- Unterstützen beim Test der SW-/HW-Komponenten
- Unterstützen bei der Vorbereitung der Integration der SW-/HW-Komponenten

3.1.4.19.2 Kompetenzfelder: Begleiten der SW-/HW-Komponentenerstellung

Fähigkeiten/Fertigkeiten

- Systemdesign und spezifizierte Anforderungen an die SW-/HW-Komponente sowie ihre Entwicklungsumgebung verständlich und konsistent vermitteln können

- Kommunikationsfähigkeit
- Konflikte konstruktiv lösen können
- resultierende Änderungsanforderungen am Systemdesign einarbeiten und kommunizieren können
- Realisierungsprozess hinsichtlich der beschriebenen Risiken bei der Realisierung einer Systemkomponente beobachten und ggf. eingreifen können
- ausgehend vom Integrations- und Prüfplan die Anforderungen an die Unit-Tests der Systemkomponenten vermitteln, Prüffälle und deren Ergebnisse kontrollieren können
- notwendige Maßnahmen zur Vorbereitung der Integration der Systemkomponenten koordinieren und kontrollieren können

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Hardwarebeschreibungssprachen (z. B. VHDL)
- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (funktionaler und struktureller Test)
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.19.3 Beispiel: Begleiten der SW-/HW-Komponentenerstellung

Im Beispielprojekt „Die Teltra Televisite“ spiegelt sich der Teilprozess in den folgenden Tätigkeiten wieder. Der IT Systems Developer unterstützt z. B. bei der Auswahl und beim Einrichten der Entwicklungsumgebung für SW-/HW-Komponenten. Er bringt seine Systemsicht in den Entwurf von Softwarekomponenten wie z. B. die Patienten-anwendung, aber auch in den Entwurf z. B. der Schutzvorrichtung für den Tablet PC ein. Er steht während des Realisierungsprozesses zur Klärung von Fragen zum Systemdesign zur Verfügung. Er verarbeitet Rückmeldungen aus den Tests der SW-/HW-Komponenten, modifiziert u. U. das Systemdesign, passt Anforderungen an die SW-/HW-Komponenten an und hilft bei der Überarbeitung ihrer Entwürfe. Zusätzlich berät er bei der Vorbereitung der SW-/HW-Komponenten zur Integration ins System.

3.1.4.20 Spezifizieren notwendiger Anpassungen von Fertigprodukten

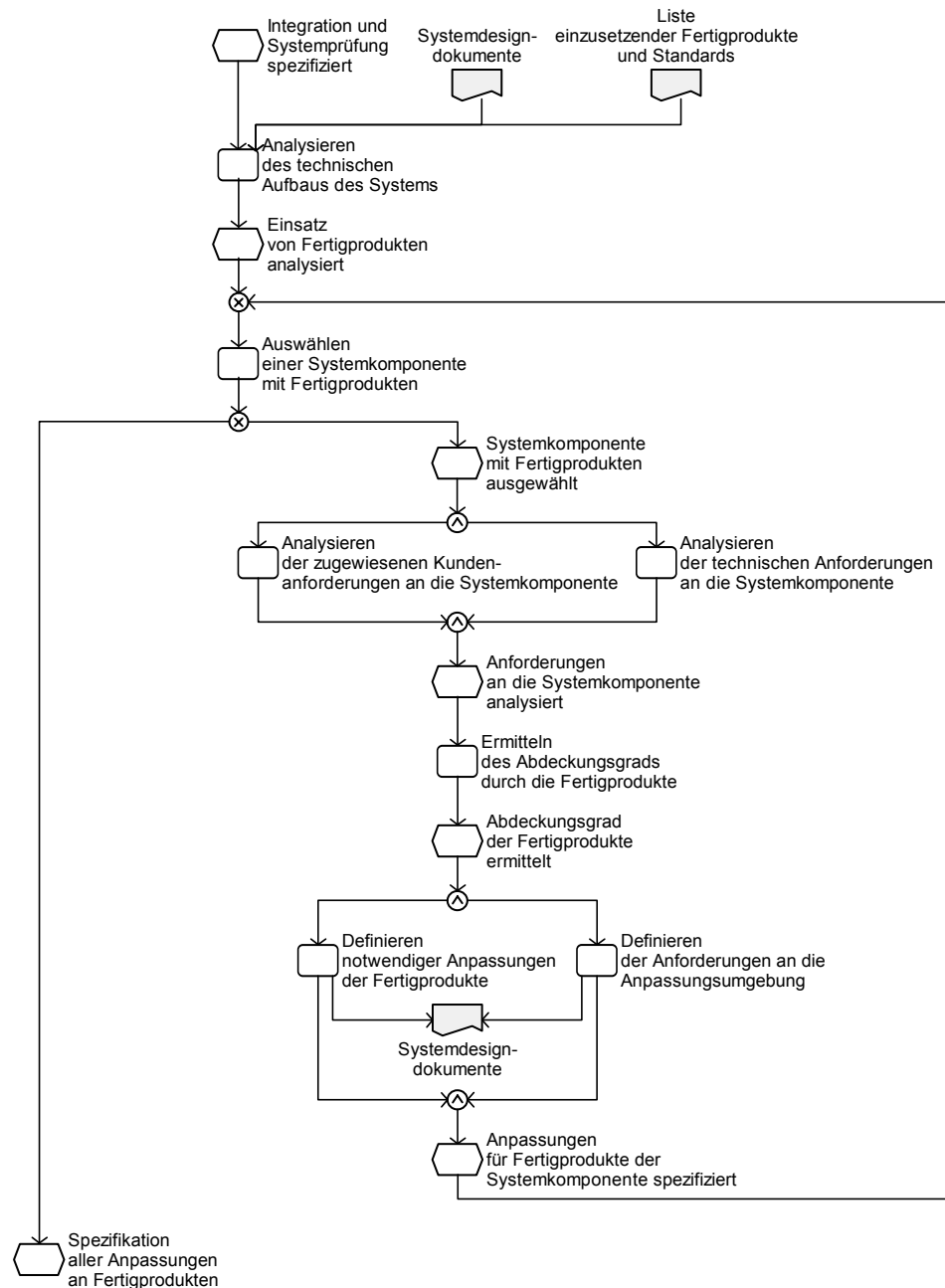


Abbildung 29: Spezifizieren notwendiger Anpassungen von Fertigprodukten.

Im Teilprozess „Spezifizieren notwendiger Anpassungen von Fertigprodukten“ werden ausgehend von der Systemarchitektur und den einzusetzenden Fertigprodukten notwendige Anpassungen an den Fertigprodukten spezifiziert.

Dazu wird für jede Systemkomponente überprüft, inwieweit die einzusetzenden Fertigprodukte zugewiesene Kundenanforderungen und technische Anforderungen abdecken. Auf Grundlage der ermittelten Abdeckung wird definiert, welche Anpassungen an den Fertigprodukten notwendig sind und welche Anforderungen daraus an die Anpassungsumgebung resultieren.

3.1.4.20.1 Tätigkeiten: Spezifizieren notwendiger Anpassungen von Fertigprodukten

- Analysieren des technischen Aufbaus des Systems
- Auswählen einer Systemkomponente mit Fertigprodukten
- Analysieren der zugewiesenen Kundenanforderungen an die Systemkomponente
- Analysieren der technischen Anforderungen an die Systemkomponente
- Ermitteln des Abdeckungsgrades durch die Fertigprodukte
- Definieren notwendiger Anpassungen der Fertigprodukte
- Definieren der Anforderungen an die Anpassungsumgebung

3.1.4.20.2 Kompetenzfelder: Spezifizieren notwendiger Anpassungen von Fertigprodukten

Fähigkeiten/Fertigkeiten

- einer ausgewählten Systemkomponente zugewiesene Anforderungen (Kundenanforderungen, allgemeine und technische Anforderungen) aus dem Systemdesign hinsichtlich der Abdeckung durch Fertigprodukte und ihrer Anpassung untersuchen und strukturieren können
- genau bestimmen können, inwieweit die einzusetzenden Fertigprodukte die funktionalen, nichtfunktionalen und technischen Anforderungen abdecken
- die Abdeckung dokumentieren können
- aus dem ermittelten Abdeckungsgrad notwendige Anpassungen der Fertigprodukte ableiten und dokumentieren können
- ausgehend von den notwendigen Anpassungen an den Fertigprodukten die Anforderungen an die Anpassungsumgebung genau bestimmen und dokumentieren können

Wissen

- Standards, Technologien und Fertigprodukte der Anwendungsdomäne und der technischen Realisierung

Werkzeuge/Methoden

- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Hardware-Beschreibungssprachen (z. B. VHDL)
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.20.3 Beispiel: Spezifizieren notwendiger Anpassungen von Fertigprodukten

Im Projekt „Die Teltra Televisite“ wurde eine Reihe von Fertigprodukten eingesetzt. Als Beispiel für die Spezifikation notwendiger Anpassungen soll die Digitalkamera dienen. Eine der zugewiesenen Kundenanforderungen ist die wirklichkeitsgetreue Wiedergabe der Wundbereiche des Patienten. Hierzu hat das Entwicklerteam zunächst ermittelt, inwieweit die Canon G3 diese Anforderung erfüllen kann. Es stellte fest, dass der automatische Weißabgleich der Kamera auf Urlaubsbilder (Sonne, blauer Himmel, grüne Wiese etc.) optimiert ist und sich zunächst weniger für die Aufnahmen der Wundbereiche eignet. Die daraus resultierenden Anpassungen bestehen in einem manuellen Weißabgleich mit genormten Farbtafeln. Dazu vergleichbar ist auch eine Kalibrierung der Arztmonitore mit speziellen Messinstrumenten notwendig.

3.1.4.21 Veranlassen der Beschaffung

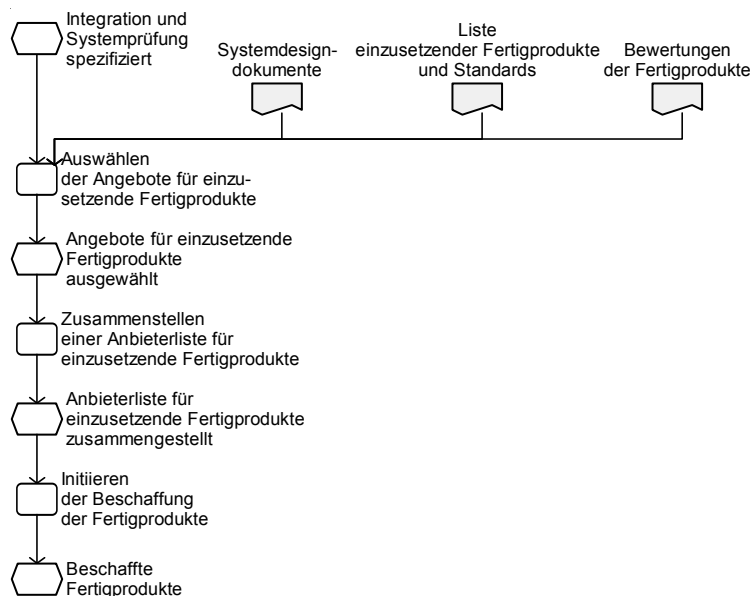


Abbildung 30: Veranlassen der Beschaffung.

Bevor Fertigprodukte angepasst und in das System integriert werden können, müssen sie beschafft werden. Die Beschaffung der Fertigprodukte ist Ziel des Teilprozesses „Veranlassen der Beschaffung“. Zunächst werden anhand der Bewertungen der Fertigprodukte relevante Angebote für die einzusetzenden Fertigprodukte ausgewählt. Aus den Angeboten werden die Anbieter in einer Anbieterliste zusammengestellt und die Beschaffung der Fertigprodukte wird initiiert.

3.1.4.21.1 Tätigkeiten: Veranlassen der Beschaffung

- Auswählen der Angebote für einzusetzende Fertigprodukte
- Zusammenstellen einer Anbieterliste für einzusetzende Fertigprodukte
- Initiieren der Beschaffung der Fertigprodukte

3.1.4.21.2 Kompetenzfelder: Veranlassen der Beschaffung

Fähigkeiten/Fertigkeiten

- ausgehend von der Bewertung der einzusetzenden Fertigprodukte und unter Berücksichtigung einer kostengünstigen Beschaffung eine Liste der zu beschaffenden Fertigprodukte und ihrer Anbieter zusammenstellen können
- die Beschaffung der Fertigprodukte entsprechend den betrieblichen Verfahren auslösen können

Wissen

- betriebliche Verfahren zur Beschaffung

Werkzeuge/Methoden

- betriebliche Anwendungen (z. B. Beschaffungssysteme)

3.1.4.21.3 Beispiel: Veranlassen der Beschaffung

Im Beispielprojekt wurden die eingeholten und geprüften Angebote für die einzusetzenden Fertigprodukte wie z. B. die Digitalkamera oder den Tablet PC zusammengestellt. Diese Angebote wurden an die Einkaufsabteilung der Berufsgenossenschaftlichen Kliniken Bergmannsheil weitergeleitet und von dieser entsprechend den gültigen Richtlinien beschafft.

3.1.4.22 Begleiten der Prüfung und Anpassung von Fertigprodukten

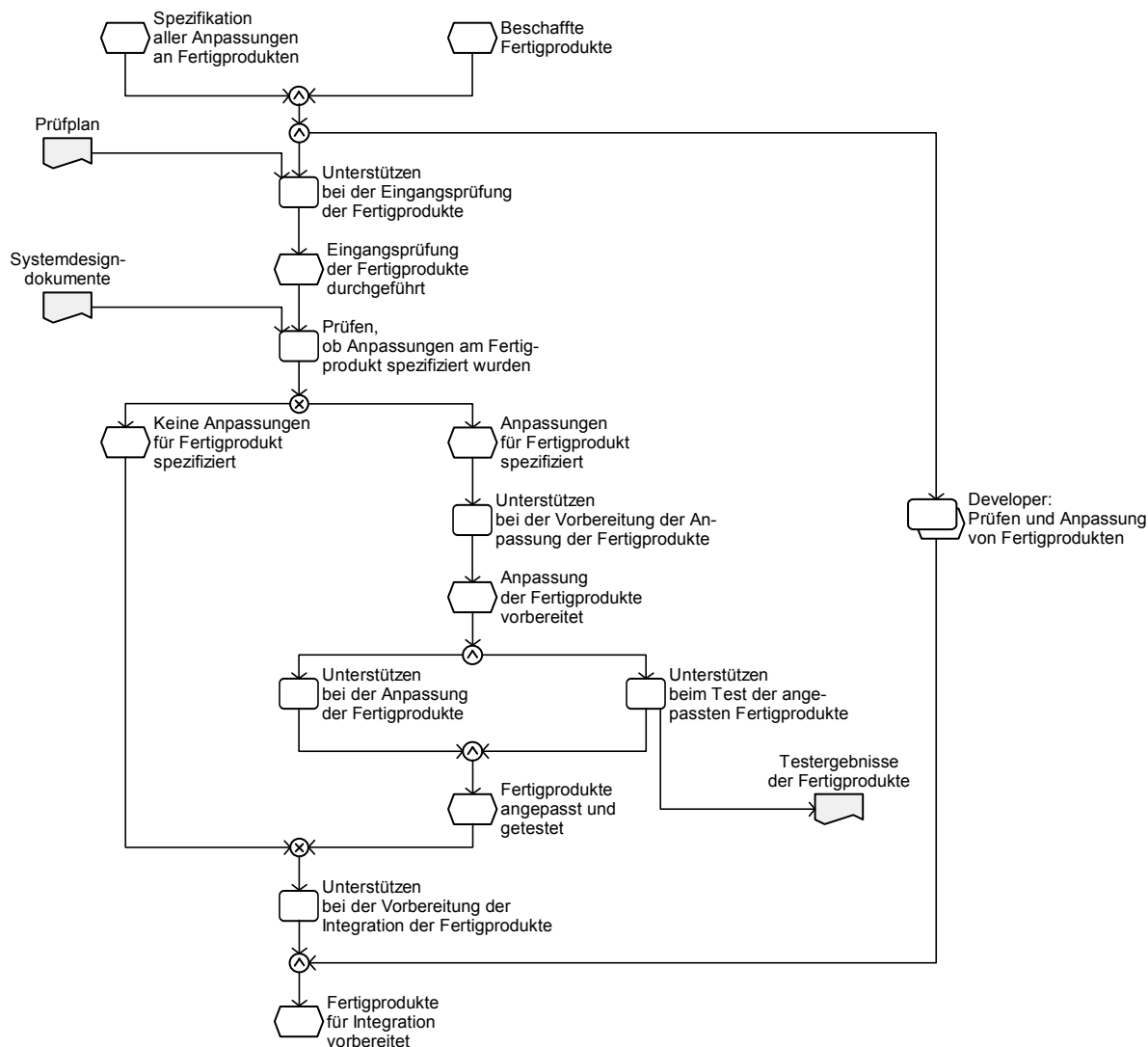


Abbildung 31: Begleiten der Prüfung und Anpassung von Fertigprodukten.

Nachdem die einzusetzenden Fertigprodukte beschafft sind, wird zunächst im Teilprozess „Begleiten der Prüfung und Anpassung von Fertigprodukten“ eine Eingangsprüfung der Fertigprodukte vorgenommen.

Wurden Anpassungen für das Fertigprodukt spezifiziert, beginnt danach die Vorbereitung der Anpassung der Fertigprodukte, wobei der IT Systems Developer unterstützend tätig wird. Im Anschluss daran unterstützt der IT Systems Developer bei den notwendigen Anpassungen und beim Test der Fertigprodukte.

Abschließend werden alle Fertigprodukte für die Integration vorbereitet.

3.1.4.22.1 Tätigkeiten: Begleiten der Prüfung und Anpassung von Fertigprodukten

- Unterstützen bei der Eingangsprüfung der Fertigprodukte
- Prüfen, ob Anpassungen am Fertigprodukt spezifiziert wurden
- Unterstützen bei der Vorbereitung der Anpassung der Fertigprodukte
- Unterstützen bei der Anpassung der Fertigprodukte

- Unterstützen beim Test der angepassten Fertigprodukte
- Unterstützen bei der Vorbereitung der Integration der Fertigprodukte

3.1.4.22.2 Kompetenzfelder: Begleiten der Prüfung und Anpassung von Fertigprodukten

Fähigkeiten/Fertigkeiten

- ausgehend von den Anforderungen an die Fertigprodukte und dem Prüfplan die Anforderungen an die Eingangstests der gelieferten Fertigprodukte vermitteln, Prüffälle und deren Ergebnisse kontrollieren können
- Systemdesign, spezifizierte Anforderungen an die SW-/HW-Komponente, die spezifizierten Anpassungen an den Fertigprodukten hinsichtlich der Integration im System sowie die Anforderungen an die Anpassungsumgebung verständlich und konsistent vermitteln können
- Kommunikationsfähigkeit
- Konflikte konstruktiv lösen können
- resultierende Änderungsanforderungen am Systemdesign einarbeiten und kommunizieren können
- Anpassungsprozess hinsichtlich der beschriebenen Risiken bei der Anpassung und Integration der Fertigprodukte beobachten und ggf. eingreifen können
- ausgehend vom Integrations- und Prüfplan die Anforderungen an die Unit-Tests der angepassten Fertigprodukte vermitteln, Prüffälle und deren Ergebnisse kontrollieren können
- notwendige Maßnahmen zur Vorbereitung der Integration der Fertigprodukte koordinieren und kontrollieren können

Werkzeuge/Methoden

- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (funktionaler und struktureller Test)
- Entwurf von Strukturen (z. B. Top-Down, Bottom-Up)
- Modularisierungsprinzipien (funktionale Abstraktion, Datenabstraktion, Information-Hiding, Schnittstellenminimalität)
- Modellierung von Funktionen (z. B. funktionale Dekomposition, Funktionsbaum)
- Klassen-/Objektmodellierung (z. B. UML-Klassen-Diagramme)
- Datenmodellierung (z. B. ER-Diagramme, Datenfluss-Diagramme)
- Prozessmodellierung (z. B. Ereignis-Prozess-Ketten, UML-Aktivitäts-Diagramme)
- Modellierung von Verhalten (z. B. UML-Zustandsübergangs-Diagramme)
- Interaktionsmodellierung (z. B. UML-Kollaborations-Diagramme, UML-Sequenz-Diagramme)
- Hardwarebeschreibungssprachen (z. B. VHDL)
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.22.3 Beispiel: Begleiten der Prüfung und Anpassung von Fertigprodukten

Nach Eingehen der bestellten Fertigprodukte wurden diese im Beispielprojekt hinsichtlich der spezifizierten Anforderungen untersucht. Bei der Eingangsprüfung, bei der Vorbereitung und

Durchführung der Anpassung, bei den Tests der angepassten Fertigprodukte sowie bei der Vorbereitung ihrer Integration wird der IT Systems Developer beratend tätig. Er verarbeitet Rückmeldungen, modifiziert u. U. das Systemdesign und die Anforderungen hinsichtlich der Anpassung der Fertigprodukte.

3.1.4.23 Vorbereiten der Systemintegration

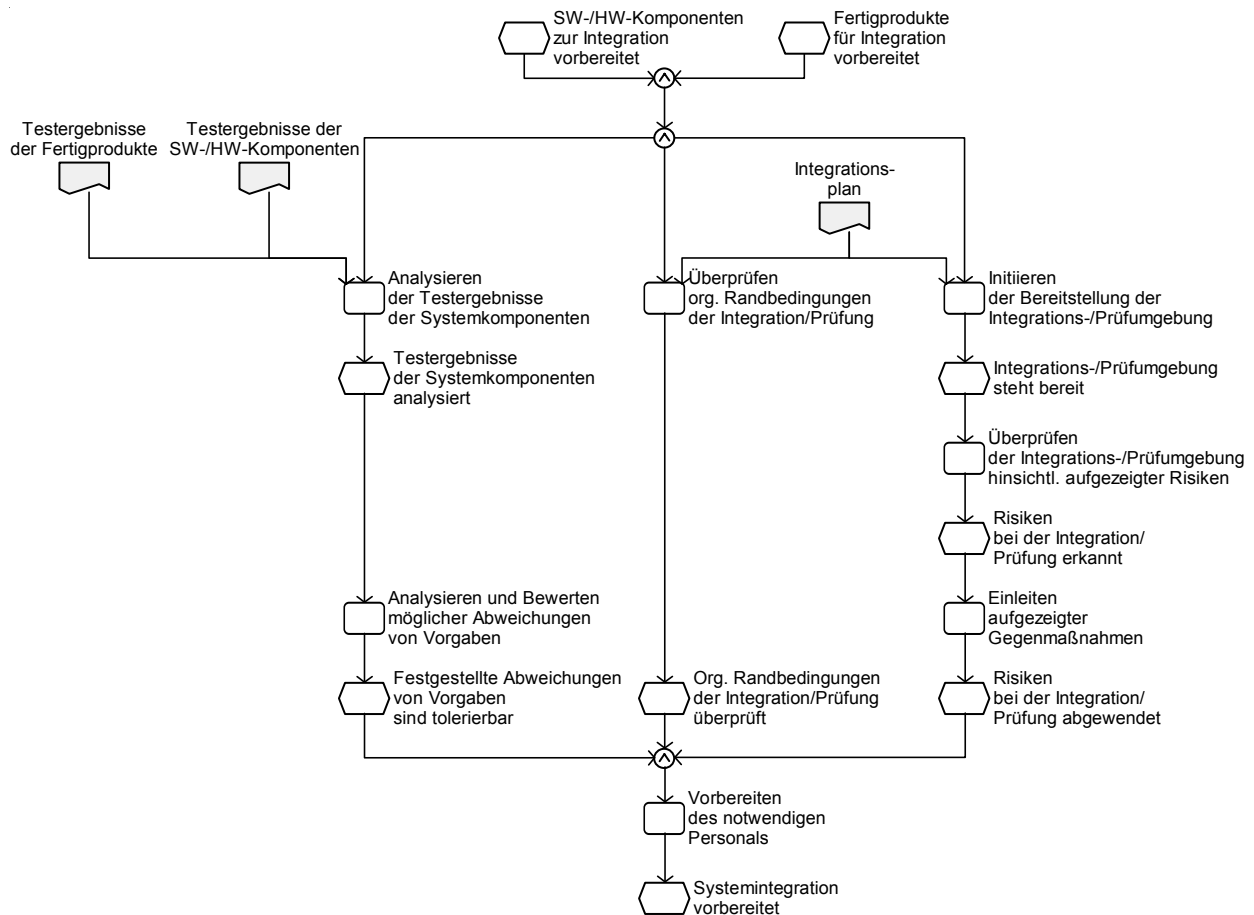


Abbildung 32: Vorbereiten der Systemintegration.

Nachdem SW-/HW-Komponenten und Fertigprodukte erstellt bzw. angepasst, getestet und zur Integration vorbereitet wurden, werden im Teilprozess „Vorbereiten der Systemintegration“ zunächst die Testergebnisse ausgewertet und überprüft, ob eventuelle Abweichungen von Vorgaben tolerierbar sind.

Organisatorische Rahmenbedingungen der Integration und Prüfung des Systems werden kontrolliert und die Bereitstellung der Integrations-/Prüfumgebung initiiert. Die Integrations-/Prüfumgebung wird hinsichtlich aufgezeigter Risiken überprüft und geeignete Gegenmaßnahmen werden eingeleitet.

Abschließend wird das notwendige Personal für die Integration und Prüfung des Systems vorbereitet.

3.1.4.23.1 Tätigkeiten: Vorbereiten der Systemintegration

- Analysieren der Testergebnisse der Systemkomponenten
- Analysieren und Bewerten möglicher Abweichungen von Vorgaben
- Überprüfen organisatorischer Randbedingungen der Integration/Prüfung
- Initiieren der Bereitstellung der Integrations-/Prüfumgebung
- Überprüfen der Integrations-/Prüfumgebung hinsichtlich aufgezeigter Risiken
- Einleiten aufgezeigter Gegenmaßnahmen
- Vorbereiten des notwendigen Personals

3.1.4.23.2 Kompetenzfelder: Vorbereiten der Systemintegration

Fähigkeiten/Fertigkeiten

- Testergebnisse der zu integrierenden Systemkomponenten hinsichtlich der Erfüllung der spezifizierten Anforderungen untersuchen können
- Abweichungen in den Testergebnissen ermitteln, untersuchen, deren mögliche Ursachen finden und dokumentieren können
- die Abweichungen hinsichtlich der spezifizierten Anforderungen bewerten und deren Tolerierbarkeit beurteilen können
- die Beurteilungen dokumentieren können
- notwendige Korrekturmaßnahmen einleiten können
- ausgehend vom Integrationsplan die definierten organisatorischen Randbedingungen zur Integration der Systemkomponenten hinsichtlich ihrer Erfüllung untersuchen können
- die Bereitstellung der Integrations- und Prüfumgebung auslösen und den Bereitstellungsprozess koordinieren und kontrollieren können
- nach Bereitstellung der Integrations- und Prüfumgebung diese hinsichtlich der Anforderungen und aufgezeigten Risiken untersuchen können
- die Durchführung notwendiger Gegenmaßnahmen auslösen, koordinieren und kontrollieren können
- Kommunikationsfähigkeit
- das bei der Integration und Prüfung beteiligte Personal einweisen und schulen bzw. dessen Schulung auslösen können

Wissen

- betriebliche Abläufe und Verantwortlichkeiten

Werkzeuge/Methoden

- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (funktionaler und struktureller Test)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Methoden des Risikomanagements (z. B. SWOT-Analyse, Impact/Probability Matrix)
- Visualisierungs- und Moderationstechniken

3.1.4.23.3 Beispiel: Vorbereiten der Systemintegration

Im Beispielprojekt wurden in diesem Teilprozess zunächst die Testergebnisse der erstellten Software- und angepassten Hardwarekomponenten analysiert und mögliche Abweichungen bewertet. So darf z. B. die Digitalkamera nach dem Weißabgleich keinerlei Toleranzen in der Farbwiedergabe aufweisen.

Danach wurden die organisatorischen Randbedingungen der Integration und Prüfung überprüft. Beispiel hierfür sind personelle und organisatorische Randbedingungen für die Integration des Server-Subsystems in das Rechenzentrum der Klinik. Dabei muss sichergestellt werden, dass die Integration und Prüfung keinerlei Auswirkungen auf den Betrieb der anderen Systeme hat.

Es wurde geprüft, ob die anderen Klinikanwendungen hinreichend gegen mögliche Fehlfunktionen des Server-Subsystems geschützt sind, um unnötige Risiken bei der Integration und Prüfung zu vermeiden. Die aufgezeigten Gegenmaßnahmen zur Reduzierung dieser Risiken wie z. B. zusätzliche Backups wurden durchgeführt.

3.1.4.24 Durchführen der Systemintegration und Systemtests

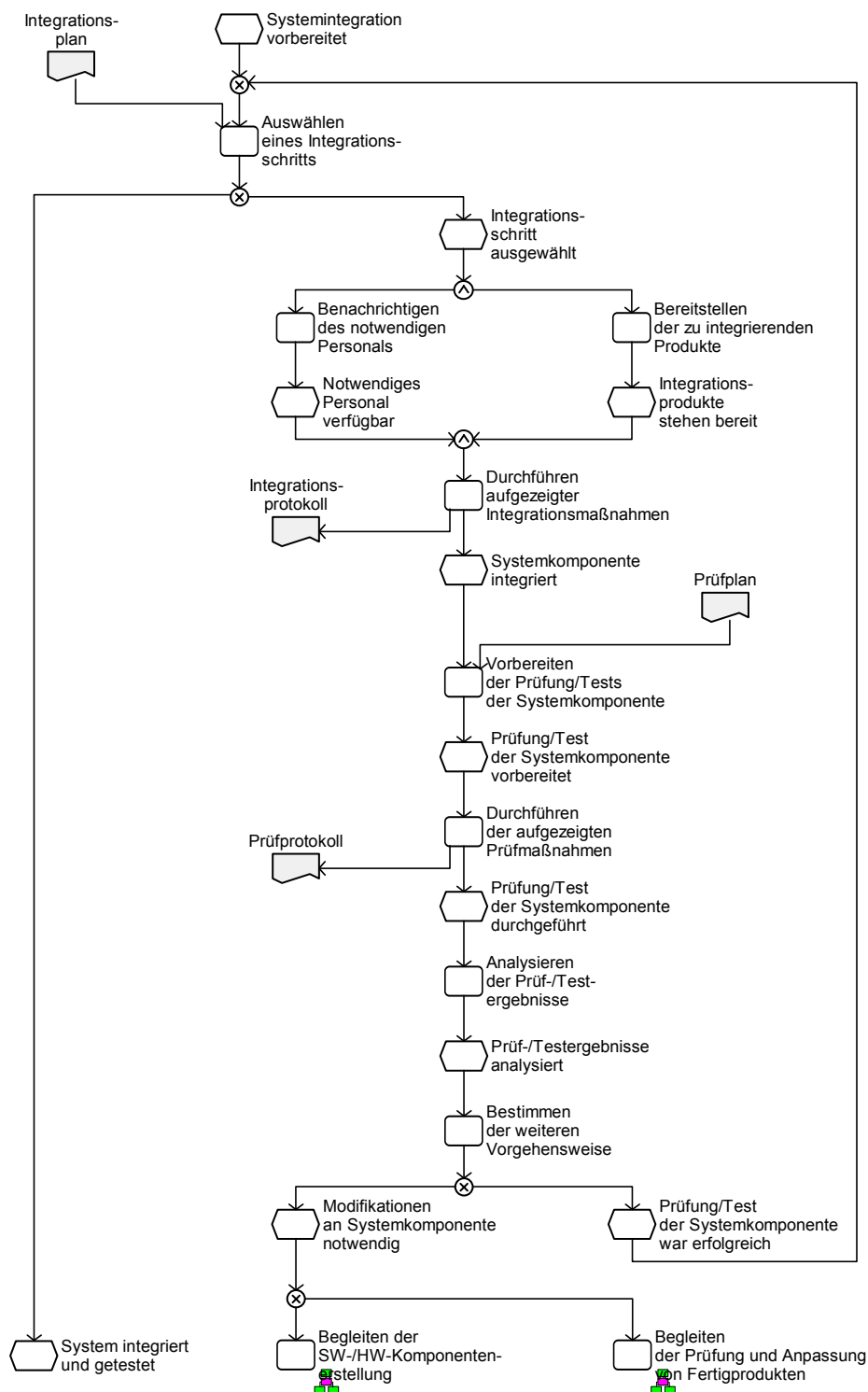


Abbildung 33: Durchführen der Systemintegration und Systemtests.

Nach der Vorbereitung der Integration und Prüfung des Systems sowie der SW-/HW-Komponenten und Fertigprodukte kann mit der Systemintegration begonnen werden. Dazu wird im Teilprozess „Durchführen der Systemintegration und Systemtests“ zunächst ein Integrationsschritt entsprechend dem Integrationsplan ausgewählt. Zu diesem Integrationsschritt werden alle Integrationsprodukte (SW-/HW-Komponenten, Fertigprodukte,

bereits integrierte Systemkomponenten) bereitgestellt. Eventuell notwendiges Personal wird benachrichtigt. Danach werden schrittweise die aufgezeigten Integrationsmaßnahmen durchgeführt und in einem Integrationsprotokoll dokumentiert. Wurde die Systemkomponente erfolgreich integriert, kann das System zur Prüfung bzw. zum Test vorbereitet werden. Nach Durchführung der aufgezeigten Prüfmaßnahmen werden deren Ergebnisse analysiert und über die weitere Vorgehensweise entschieden. Im Ergebnis kann eine Modifikation der Systemkomponente bzw. von deren Integrationsprodukten notwendig sein, oder die Integration kann mit dem nächsten Integrationsschritt fortgesetzt werden. Nach erfolgreicher Durchführung aller Integrationsschritte und der Prüfmaßnahmen ist das System vollständig integriert und getestet.

3.1.4.24.1 Tätigkeiten: Durchführen der Systemintegration und Systemtests

- Auswählen eines Integrationsschritts
- Benachrichtigen des notwendigen Personals
- Bereitstellen der zu integrierenden Produkte
- Durchführen aufgezeigter Integrationsmaßnahmen
- Vorbereiten der Prüfung/Tests der Systemkomponente
- Durchführen aufgezeigter Prüfmaßnahmen
- Analysieren der Prüf-/Testergebnisse
- Bestimmen der weiteren Vorgehensweise

3.1.4.24.2 Kompetenzfelder: Durchführen der Systemintegration und Systemtests

Fähigkeiten/Fertigkeiten

- einen Integrationsschritt des Integrationsplans bzgl. des notwendigen Personals und der zu integrierenden Produkte vorbereiten können
- die in der Integrationsanleitung aufgeführten Integrationsmaßnahmen vollständig abarbeiten und die Ergebnisse kontrollieren können
- die Durchführung der Integrationsmaßnahme und deren Ergebnisse einschließlich möglicher Abweichungen protokollieren können
- entsprechend dem Prüfplan die aufgezeigten Voraussetzungen für die Prüfung der integrierten Systemkomponente schaffen bzw. kontrollieren können
- die in der Prüfanleitung aufgeführten Prüfmaßnahmen vollständig abarbeiten und die Prüfergebnisse im Prüfprotokoll dokumentieren können
- die Prüf- bzw. Testergebnisse hinsichtlich der Erfüllung der Anforderungen untersuchen, Abweichungen und deren Ursachen ermitteln und dokumentieren können
- Abweichungen in den Prüf- bzw. Testergebnissen beurteilen und daraus die weitere Vorgehensweise ableiten können

Werkzeuge/Methoden

- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (funktionaler und struktureller Test)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)

3.1.4.24.3 Beispiel: Durchführen der Systemintegration und Systemtests

Zur Durchführung der Systemintegration und Systemtests wurden im Beispielprojekt die aufgezeigten Integrationsschritte aus dem Integrationsplan abgearbeitet. Bei jedem Integrationsschritt wurden zunächst das notwendige Personal informiert und die zu integrierenden Produkte bereitgestellt. Bei der Integration des Server-Subsystems sind dies u. a. die notwendige Hardware, der angepasste Apache-Web-Server mit PHP-Modul und OpenSSL-Erweiterung, die PHP-Skripte, die PostgreSQL-Datenbank und die notwendigen SQL-Skripte des Datenbankschemas.

Der Apache-Web-Server und die PostgreSQL-Datenbank wurden installiert, konfiguriert und getestet. Dabei ist z. B. beim Apache der verschlüsselte Zugriff über HTTPS und Port 443 zu testen. Wichtig ist auch, dass der unverschlüsselte Zugriff über HTTP und Port 80 verhindert wird. Danach wurden das Datenbankschema und Testdaten in die PostgreSQL-Datenbank eingespielt. Nach dem Installieren der PHP-Skripte konnte dann der Zugriff auf die Datenbank über den Apache und PHP getestet werden. Dabei wurde insbesondere geprüft, ob ein unverschlüsselter oder nicht autorisierter Zugriff auf die Daten der Datenbank möglich ist.

Die Testergebnisse wurden ausgewertet und notwendige Anpassungen abgeleitet. Ist z. B. ein nicht autorisierter Zugriff auf die Daten möglich, müssen zusätzliche Maßnahmen wie den Einsatz einer Firewall ergriffen werden.

3.1.4.25 Vorbereiten der Installation in der Kundenumgebung

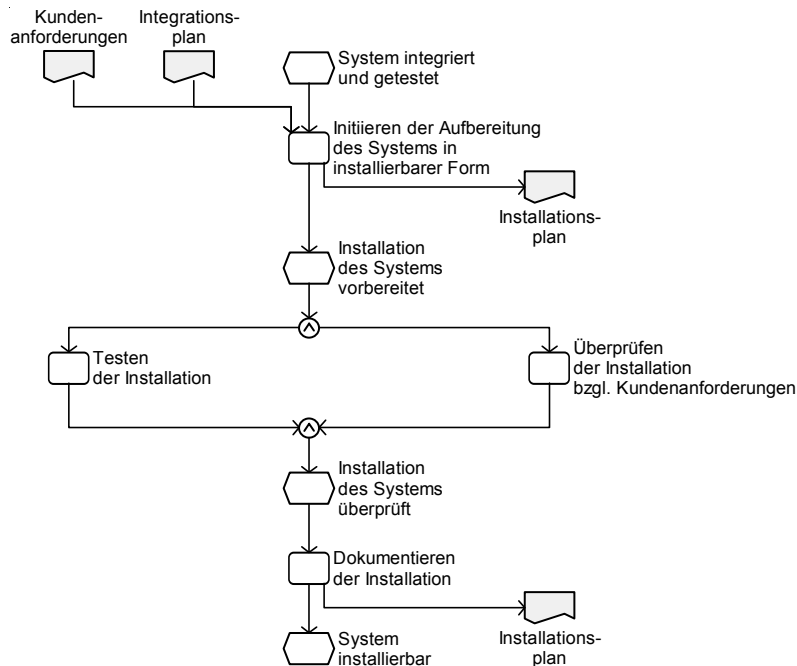


Abbildung 34: Vorbereiten der Installation in der Kundenumgebung.

Wurde das System erfolgreich integriert und getestet, wird es im Teilprozess „Vorbereiten der Installation in der Kundenumgebung“ so aufbereitet, dass der Kunde das System in seiner Umgebung installieren kann.

Die Aufbereitung des Systems in installierbarer Form wird zunächst vom IT Systems Developer initiiert. Danach testet er die Installation und überprüft sie hinsichtlich der Kundenanforderungen. Im Anschluss wird die Installation des Systems dokumentiert.

3.1.4.25.1 Tätigkeiten: Vorbereiten der Installation in der Kundenumgebung

- Initiieren der Aufbereitung des Systems in installierbarer Form
- Testen der Installation
- Überprüfen der Installation bzgl. Kundenanforderungen
- Dokumentieren der Installation

3.1.4.25.2 Kompetenzfelder: Vorbereiten der Installation in der Kundenumgebung

Fähigkeiten/Fertigkeiten

- die Aufbereitung des Systems in installierbarer Form auslösen und die Kundenanforderungen hinsichtlich der Installation vermitteln können
- den Aufbereitungsprozess koordinieren und kontrollieren können
- notwendige Installationsschritte in einem Installationsplan dokumentieren können
- Installation gemäß dem Installationsplan beispielhaft durchführen bzw. simulieren können
- Installationsschritte und deren Ergebnisse dokumentieren können

- Installationsprozess und Installation hinsichtlich der Erfüllung der Kundenanforderungen untersuchen, mögliche Abweichungen ermitteln, dokumentieren, bewerten und die Tolerierbarkeit beurteilen können
- eventuell notwendige Korrekturmaßnahmen einleiten können
- Änderungen in den Installationsplan einarbeiten können

Werkzeuge/Methoden

- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (funktionaler und struktureller Test)
- Methoden zur Darstellung und Beschreibung von Varianten (Checklisten)
- Methoden zur Beurteilung und Auswahl von Varianten (gewichtete Entscheidungsmatrix, Nutzwertanalyse)
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.25.3 Beispiel: Vorbereiten der Installation in der Kundenumgebung

Im Beispielprojekt „Die Teltra Televisite“ wurden z. B. das Betriebssystem und die Patientenapplication als Festplatten-Image aufbereitet. Dadurch kann die Einrichtung des Patientenrechners automatisiert und beschleunigt werden. Dieses Vorgehen wurde getestet und dokumentiert.

3.1.4.26 Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien

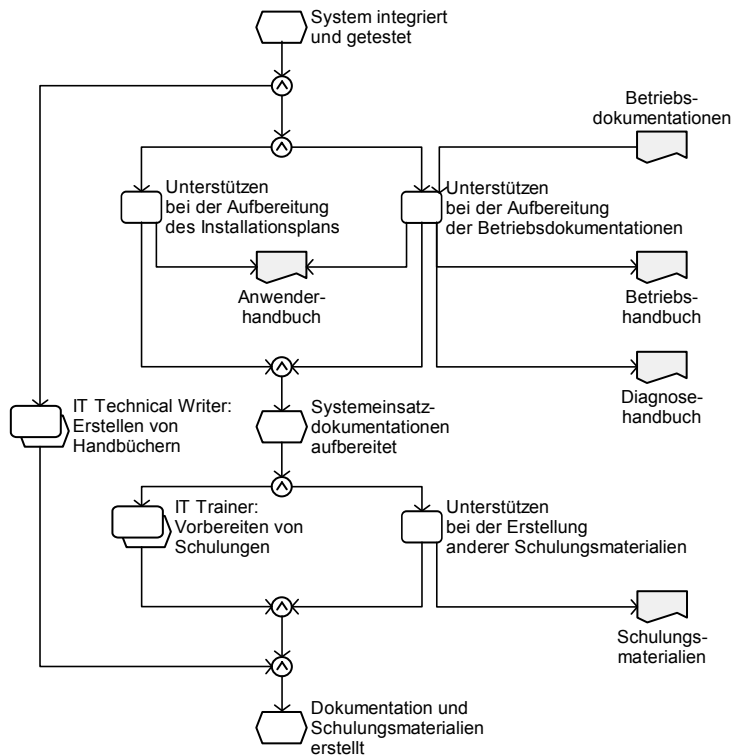


Abbildung 35: Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien.

Im Teilprozess „Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien“ unterstützt der IT Systems Developer den IT Technical Writer und den IT Trainer bei der Erstellung von Anwender-, Betriebs- und Diagnosehandbuch bzw. von Schulungsmaterialien. Hierbei werden insbesondere Betriebsdokumentationen und Installationsplan verarbeitet.

3.1.4.26.1 Tätigkeiten: Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien

- Unterstützen bei der Aufbereitung des Installationsplans
- Unterstützen bei der Aufbereitung der Betriebsdokumentationen
- Unterstützen bei der Erstellung anderer Schulungsmaterialien

3.1.4.26.2 Kompetenzfelder: Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien

Fähigkeiten/Fertigkeiten

- Systemdokumentationen für die Erstellung von Kundendokumentation und Schulungsmaterialien vollständig zusammenstellen können
- die Kundenanforderungen bezüglich der Dokumentation und Schulung verständlich und konsistent vermitteln können
- Dokumentationen hinsichtlich ihrer Vollständigkeit und Konsistenz untersuchen können
- Änderungen an System und Dokumentationen koordinieren und kontrollieren können
- Kommunikationsfähigkeit

Werkzeuge/Methoden

- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.26.3 Beispiel: Mitarbeiten beim Erstellen der Dokumentation und Schulungsmaterialien

In diesem Teilprozess wird der IT Systems Developer bei der Ausarbeitung der Dokumentation und Schulungsmaterialien unterstützend tätig. Im Beispielprojekt ist es z. B. notwendig, Dokumentationen und Schulungsmaterialien für die Patienten und Ärzte zu erarbeiten. Die Patienten werden in der Regel vom Personal der Firma Teltra in der Bedienung der Digitalkamera, des Tablet PCs und der Patientenanzwendung geschult. Hierfür wurden entsprechende Schulungsmaterialien ausgearbeitet. Patienten- und Arztanwendung sind mit einer Hilfe versehen, welche die Nutzer bei der Bedienung der Anwendung unterstützen soll.

3.1.4.27 Begleiten der Installation und des Abnahmetests

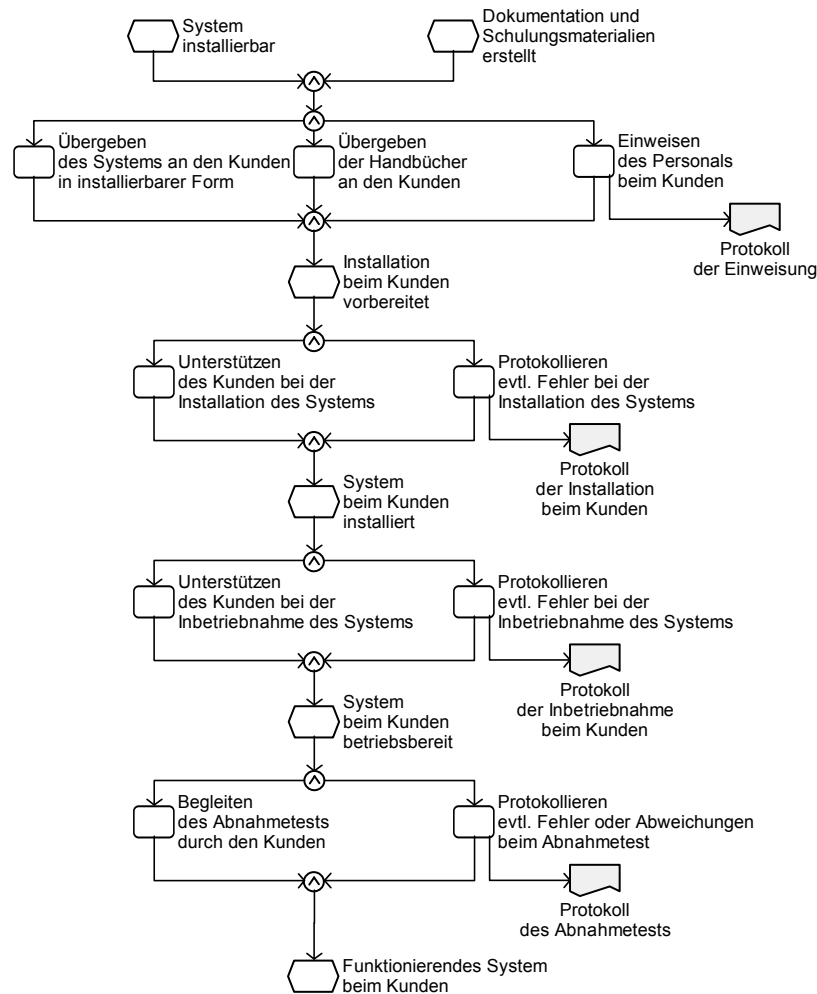


Abbildung 36: Begleiten der Installation und des Abnahmetests.

Im Teilprozess „Begleiten der Installation und des Abnahmetests“ werden zunächst das System in installierbarer Form sowie die Handbücher an den Kunden bzw. dessen Personal übergeben. Daraufhin wird das Personal des Kunden eingewiesen. Nach der Vorbereitung der Installation des Systems begleitet der IT Systems Developer die Installation und protokolliert die durchgeführten Schritte sowie die dabei aufgetretenen Fehler. Ist die Installation des Systems beim Kunden abgeschlossen, kann das System in Betrieb genommen werden. Auch hierbei unterstützt der IT Systems Developer den Kunden und protokolliert die durchgeführten Schritte sowie die aufgetretenen Fehler. Anschließend kann der Kunde seinen Abnahmetest durchführen, wobei er wiederum vom IT Systems Developer begleitet wird, der eventuelle Fehler und Abweichungen protokolliert.

3.1.4.27.1 Tätigkeiten: Begleiten der Installation und des Abnahmetests

- Übergeben des Systems an den Kunden in installierbarer Form
- Übergeben der Handbücher an den Kunden
- Einweisen des Personals beim Kunden
- Unterstützen des Kunden bei der Installation des Systems
- Protokollieren eventueller Fehler bei der Installation des Systems

- Unterstützen des Kunden bei der Inbetriebnahme des Systems
- Protokollieren eventueller Fehler bei der Inbetriebnahme des Systems
- Begleiten des Abnahmetests durch den Kunden
- Protokollieren eventueller Fehler oder Abweichungen beim Abnahmetest

3.1.4.27.2 Kompetenzfelder: Begleiten der Installation und des Abnahmetests

Fähigkeiten/Fertigkeiten

- alle notwendigen Systemkomponenten und Handbücher in geeigneter Form für die Übergabe zusammenstellen können
- die Übergabe des Systems in geeigneter Form durchführen und dokumentieren können
- dem Personal des Kunden die notwendigen Kenntnisse und Fertigkeiten für die Installation und Inbetriebnahme des Systems vermitteln können
- den Prozess der Installation und Inbetriebnahme in nachvollziehbare Schritte zerlegen und vermitteln können
- den Prozess der Installation und Inbetriebnahme des Systems beim Kunden beobachten, kontrollieren und dokumentieren können
- fehlerhaft verlaufene Schritte bei der Installation und Inbetriebnahme sowie aufgetretene Fehler dokumentieren können
- den Abnahmetest des Kunden beobachten und aufgetretene Fehler oder Abweichungen von den Kundenanforderungen dokumentieren können

Werkzeuge/Methoden

- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)
- statische Testmethoden (z. B. Inspektion, Review)
- dynamische Testmethoden (funktionaler und struktureller Test)

3.1.4.27.3 Beispiel: Begleiten der Installation und des Abnahmetests

Im Beispielprojekt „Die Teltra Televisite“ erfolgt die Installation des Gesamtsystems nicht in einem Schritt zu einem bestimmten Zeitpunkt. Das Kernstück, das Server-Subsystem wurde nicht bei einem externen Kunden installiert, sondern wird unter Verantwortung der Firma Teltra betrieben und wurde in das Rechenzentrum der Kliniken integriert. Das dortige Personal wurde entsprechend eingewiesen.

Die Patienten-Subsysteme sind mobil und werden in einem betriebsbereiten Zustand entsprechend dem Bedarf beim Patienten aufgestellt. Unter Umständen ist es notwendig Anpassungen vorzunehmen, wenn am Wohnort des Patienten kein Mobilfunknetz verfügbar ist. In diesem Fall kann auf das Festnetz der Telekom ausgewichen werden.

Beim Arzt muss lediglich der kalibrierte Monitor aufgestellt und angeschlossen werden. PC mit Betriebssystem und Web-Browser sind vorhanden. Gegebenenfalls müssen die Einstellungen des Web-Browsers angepasst werden.

Der Abnahmetest des Gesamtsystems wurde vom Management der Firma Teltra durchgeführt.

3.1.4.28 Übergeben an den IT Systems Administrator

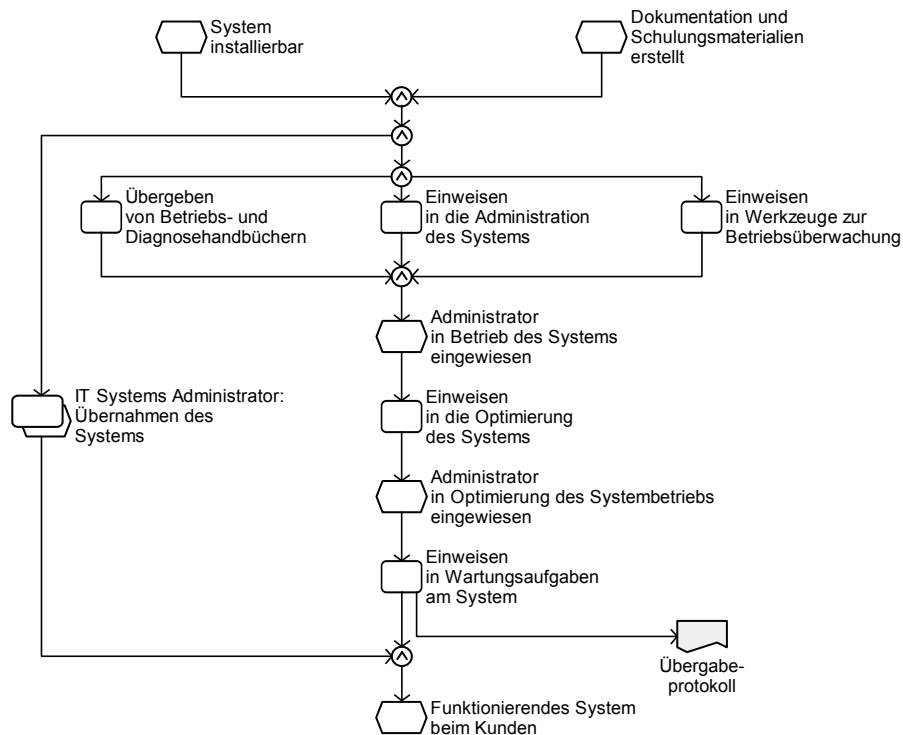


Abbildung 37: Übergeben an den IT Systems Administrator.

Neben der offiziellen Übergabe des Systems an den Kunden, verbunden mit dem Abnahmetest, wird im Teilprozess „Übergeben an den IT Systems Administrator“ das System unter einem technischen Gesichtspunkt an den IT Systems Administrator übergeben.

Dazu übergibt der IT Systems Developer dem IT Systems Administrator alle Betriebs- und Diagnosehandbücher des Systems und weist ihn in die Administration des Systems sowie in die Werkzeuge zur Betriebsüberwachung ein.

Danach erfolgt eine Einweisung in die Optimierung und Wartung des Systems. Die Einweisung des IT Systems Administrator wird in einem Übergabeprotokoll dokumentiert.

3.1.4.28.1 Tätigkeiten: Übergeben an den IT Systems Administrator

- Übergeben der Betriebs- und Diagnosehandbücher
- Einweisen in die Administration des Systems
- Einweisen in Werkzeuge zur Betriebsüberwachung
- Einweisen in die Optimierung des Systems
- Einweisen in Wartungsaufgaben am System

3.1.4.28.2 Kompetenzfelder: Übergeben an den IT Systems Administrator

Fähigkeiten/Fertigkeiten

- alle notwendigen Betriebs- und Diagnosehandbücher in geeigneter Form für die Übergabe an den IT Systems Administrator zusammenstellen und an ihn übergeben können
- Übergabe dokumentieren können

- dem IT Systems Administrator alle notwendigen Kenntnisse und Fertigkeiten für die Administration, Optimierung und Wartung des Systems vermitteln können
- die Prozesse zur Administration, Optimierung und Wartung in nachvollziehbare Schritte zerlegen und vermitteln können
- dem IT Systems Administrator alle notwendigen Kenntnisse und Fertigkeiten zur Benutzung der Werkzeuge der Betriebsüberwachung vermitteln können
- die Prozesse zur Betriebsüberwachung des Systems in nachvollziehbare Schritte zerlegen und vermitteln können
- Kommunikationsfähigkeit

Werkzeuge/Methoden

- Visualisierungstechniken
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.28.3 Beispiel: Übergeben an den IT Systems Administrator

Da im Beispielprojekt das System unter der Verantwortung der Firma Teltra von einem Teil der Entwickler betrieben wird, war die Übergabe an den IT Systems Administrator weniger formal als im Teilprozess beschrieben. Es war lediglich eine Abstimmung in der Zusammenarbeit mit den Systemadministratoren des Rechenzentrums der Kliniken notwendig. Diese führen z. B. auf Anweisung des Teltra-Personals einen Restart des Systems durch. Wird die Verantwortung für den Betrieb und die Administration des Systems an eine Person außerhalb des Entwicklerteams übergeben, werden die Aktivitäten des Teilprozesses notwendig.

3.1.4.29 Unterstützen von Schulungen

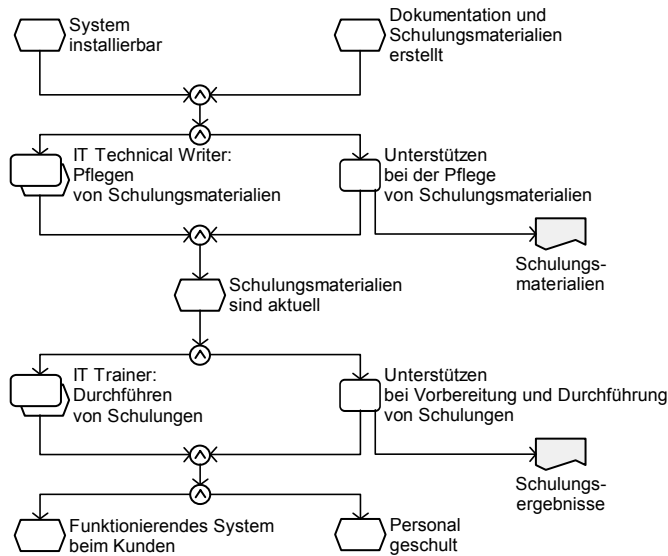


Abbildung 38: Unterstützen von Schulungen.

Im Teilprozess „Unterstützen von Schulungen“ steht der IT Systems Developer dem IT Technical Writer und dem IT Trainer als Experte bei der Pflege von Schulungsmaterialien bzw. bei der Vorbereitung und Durchführung von Schulungen unterstützend zur Seite.

3.1.4.29.1 Tätigkeiten: Unterstützen von Schulungen

- Unterstützen bei der Pflege der Schulungsmaterialien
- Unterstützen bei der Vorbereitung und Durchführung von Schulungen

3.1.4.29.2 Kompetenzfelder: Unterstützen von Schulungen

Fähigkeiten/Fertigkeiten

- aus den Kundenanforderungen und konkreten Merkmalsausprägungen des Systems Anforderungen an Schulungsinhalte, -formen und -materialien ableiten und dokumentieren können
- Schulungsinhalte, -formen und -materialien und die Ergebnisse der Schulungen hinsichtlich der Erfüllung der spezifizierten Anforderungen untersuchen können
- Abweichungen hinsichtlich der Tolerierbarkeit untersuchen und bewerten sowie geeignete Gegenmaßnahmen erarbeiten und dokumentieren können
- die Konsistenz zwischen Merkmalen des Systems und Schulungsinhalten, -formen und -materialien überwachen sowie den Bedarf an Änderungen priorisieren und dokumentieren können
- Änderungen an Schulungsinhalten und -materialien koordinieren, kontrollieren und dokumentieren können
- systemspezifisches Fachwissen vollständig, konsistent und verständlich vermitteln und präsentieren können

Wissen

- gesetzliche Vorschriften

Werkzeuge/Methoden

- Visualisierungstechniken
- Textverarbeitung
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)

3.1.4.29.3 Beispiel: Unterstützen von Schulungen

Im Beispielprojekt muss die Kundenanforderung nach Nutzung des Patientensystems durch ältere Menschen auch Niederschlag in der Art und Weise der Schulungen finden. Hier müssen entsprechende Anforderungen formuliert und u. U. andere Spezialisten wie z. B. Pädagogen und Psychologen hinzugezogen werden. Auch müssen die Konsistenz von Schulungsmaterialien und Systemfunktionalitäten überwacht und mögliche Änderungen an den Schulungsmaterialien koordiniert werden.

3.1.4.30 Sammeln von Verbesserungsvorschlägen

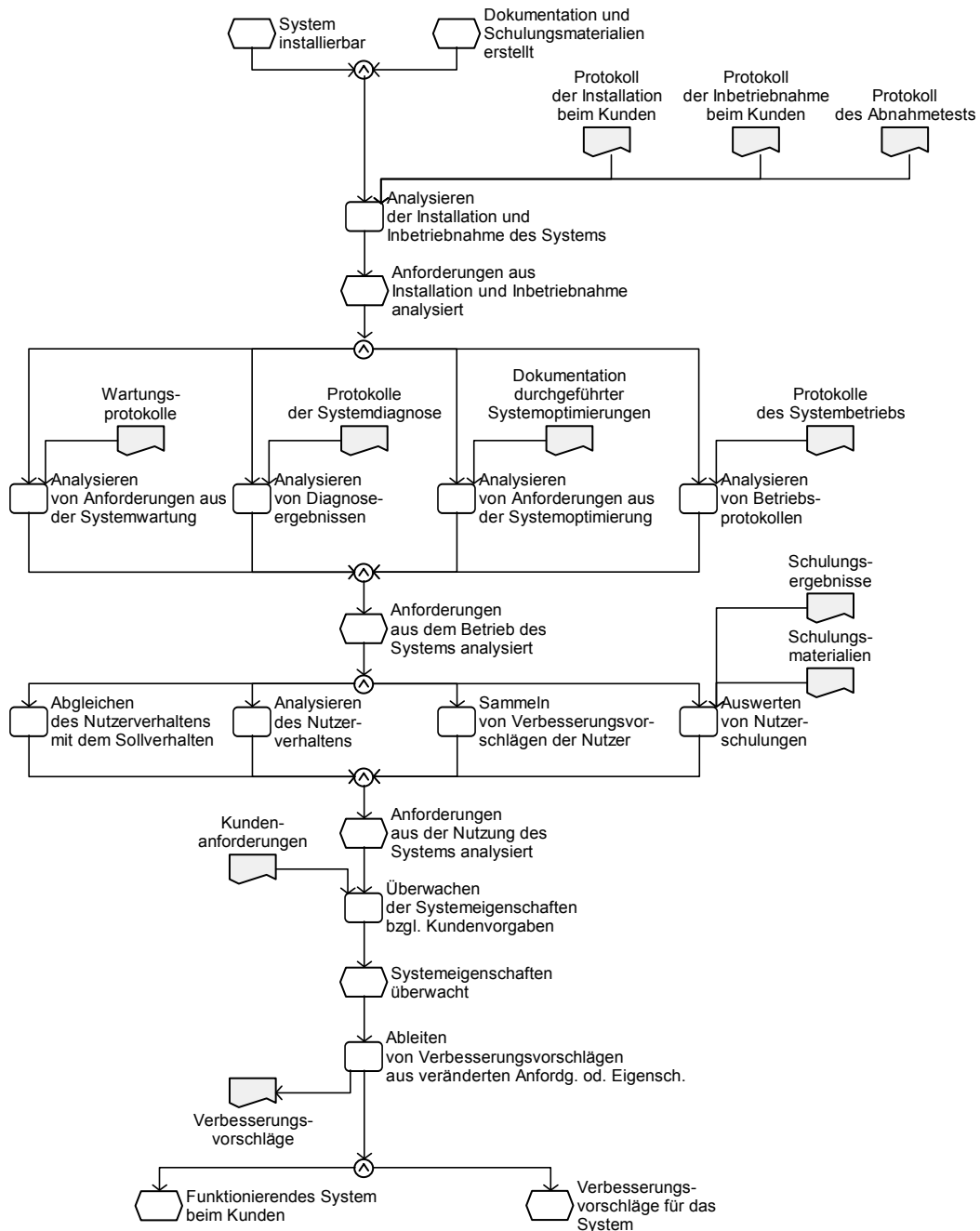


Abbildung 39: Sammeln von Verbesserungsvorschlägen.

Der Teilprozess „Sammeln von Verbesserungsvorschlägen“ soll aufzeigen, welche Tätigkeiten durchgeführt werden können, um nach der Fertigstellung eines Systems Verbesserungsvorschläge für dessen Überarbeitung zu sammeln.

Für eine Überarbeitung des Systems können zunächst Anforderungen aus der Analyse der Protokolle der Installation des Systems beim Kunden, der Inbetriebnahme und des Abnahmetests abgeleitet werden.

Hinsichtlich des Betriebs des Systems können sich Anforderungen aus der Analyse der Wartungsprotokolle, der Protokolle der Systemdiagnose, der Dokumentation der Systemoptimierung und der Betriebsprotokolle ergeben.

Bezüglich der Nutzung des Systems können Anforderungen zu dessen Überarbeitung aus der Analyse des Nutzerverhaltens und dem Abgleich mit dem Sollverhalten entstehen. Verbesserungsvorschläge der Nutzer sind zu sammeln und Nutzerschulungen auszuwerten. Schließlich ist auch zu überwachen, ob z. B. Systembetrieb oder nicht vorgegebenes Nutzerverhalten zu einer Veränderung der Systemeigenschaften führen.

Im Ergebnis der Analyse- und Überwachungstätigkeiten sind veränderte Anforderungen an das System zu formulieren und Vorschläge zu dessen Verbesserung abzuleiten.

3.1.4.30.1 Tätigkeiten: Sammeln von Verbesserungsvorschlägen

- Analysieren der Installation und Inbetriebnahme des Systems
- Analysieren der Anforderungen aus der Systemwartung
- Analysieren von Diagnoseergebnissen
- Analysieren von Anforderungen aus der Systemoptimierung
- Analysieren von Betriebsprotokollen
- Abgleichen des Nutzerverhaltens mit dem Sollverhalten
- Analysieren des Nutzerverhaltens
- Sammeln von Verbesserungsvorschlägen der Nutzer
- Auswerten der Nutzerschulungen
- Überwachen der Systemeigenschaften bzgl. der Kundenanforderungen
- Ableiten von Verbesserungsvorschlägen aus veränderten Anforderungen oder Eigenschaften

3.1.4.30.2 Kompetenzfelder: Sammeln von Verbesserungsvorschlägen

Fähigkeiten/Fertigkeiten

- Protokolle der Installation, der Inbetriebnahme und des Abnahmetests hinsichtlich Abweichungen von den Kundenanforderungen untersuchen können
- Betriebsprotokolle und Diagnoseergebnisse hinsichtlich Abweichungen von nichtfunktionalen Kundenanforderungen untersuchen können
- Nutzerverhalten und Systemeigenschaften hinsichtlich Abweichungen von funktionalen und nichtfunktionalen Kundenanforderungen untersuchen können
- Abweichungen dokumentieren können
- Abweichungen bzgl. ihrer Tolerierbarkeit bewerten und daraus Änderungsanforderungen ableiten können
- Anforderungen aus der Systemwartung und -optimierung, Verbesserungsvorschläge der Nutzer und Ergebnisse der Nutzerschulungen hinsichtlich notwendiger Anpassungen am System untersuchen und Änderungsanforderungen ableiten können
- alle ermittelten Änderungsanforderungen priorisieren und dokumentieren können

Werkzeuge/Methoden

- Methoden des Change Managements
- Methoden des Configuration Managements
- Methoden des Release Managements
- Methoden des Dokumentenmanagements (z. B. Versionierung, Bereitstellung, Archivierung)
- Trouble-Ticket-Systeme
- Textverarbeitung

3.1.4.30.3 Beispiel: Sammeln von Verbesserungsvorschlägen

Im Beispielprojekt „Die Teltra Televisite“ werden aus verschiedenen Quellen Rückmeldungen zu möglichen Verbesserungen oder Anpassungen des Systems verarbeitet. Diese Quellen können die Patienten und Ärzte sein, aber auch das Personal, das z. B. die Patientensysteme aufstellt sowie die Patienten in der Nutzung schult. Weiterhin können die Zugriffsprotokolle des Web-Servers und Betriebsprotokolle der PostgreSQL-Datenbank hinsichtlich des Nutzerverhaltens, der Verfügbarkeit und Leistung des Systems analysiert und daraus Hinweise für Verbesserungen abgeleitet werden.

Regelmäßige Wartungen und Überprüfungen z. B. der Digitalkameras und Tablet PCs ermöglichen eine ständige Kontrolle ihrer Eigenschaften und einen Abgleich mit den Kundenanforderungen.

Änderungsanforderungen werden nach ihrer Art und ihrer Resonanz priorisiert und die resultierenden Anpassungen am System entsprechend der Priorisierung vorgenommen.