



Ingo Patett, Dorothea Schäfer

Herausgeber: Jürgen Gratzke

IT-Berufe

Fachstufe II
Fachinformatiker/-in
Anwendungsentwicklung,
Daten- und Prozessanalyse
Lernfelder 10–12

1. Auflage

Die in diesem Produkt gemachten Angaben zu Unternehmen (Namen, Internet- und E-Mail-Adressen, Handelsregistereintragungen, Bankverbindungen, Steuer-, Telefon- und Faxnummern und alle weiteren Angaben) sind i. d. R. fiktiv, d. h., sie stehen in keinem Zusammenhang mit einem real existierenden Unternehmen in der dargestellten oder einer ähnlichen Form. Dies gilt auch für alle Kunden, Lieferanten und sonstigen Geschäftspartner der Unternehmen wie z. B. Kreditinstitute, Versicherungsunternehmen und andere Dienstleistungsunternehmen. Ausschließlich zum Zwecke der Authentizität werden die Namen real existierender Unternehmen und z. B. im Fall von Kreditinstituten auch deren IBANs und BICs verwendet.

Zusatzmaterialien zu IT-Berufe Fachstufe II – Fachinformatiker/-in Anwendungsentwicklung, Daten- und Prozessanalyse – Lernfelder 10–12

Für Lehrerinnen und Lehrer:

Lösungen Download: 978-3-14-220076-7

Lösungen zum Arbeitsheft Download: 978-3-14-220084-2



BiBox Einzellizenz für Lehrer/-innen (Dauerlizenz): 978-3-14-220088-0

BiBox Kollegiumslizenz für Lehrer/-innen (Dauerlizenz): 978-3-14-220092-7

BiBox Kollegiumslizenz für Lehrer/-innen (1 Schuljahr): 978-3-14-107777-3

inkl. E-Book

Für Schülerinnen und Schüler:

Arbeitsheft: 978-3-14-220080-4



BiBox Einzellizenz für Schüler/-innen (1 Schuljahr): 978-3-14-220096-5

BiBox Klassensatz PrintPlus (1 Schuljahr): 978-3-427-82113-7

inkl. E-Book

© 2023 Bildungsverlag EINS GmbH, Ettore-Bugatti-Straße 6-14, 51149 Köln
www.westermann.de

Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Nutzung in anderen als den gesetzlich zugelassenen bzw. vertraglich zugestanden Fällen bedarf der vorherigen schriftlichen Einwilligung des Verlages. Nähere Informationen zur vertraglich gestatteten Anzahl von Kopien finden Sie auf www.schulbuchkopie.de.

Für Verweise (Links) auf Internet-Adressen gilt folgender Haftungshinweis: Trotz sorgfältiger inhaltlicher Kontrolle wird die Haftung für die Inhalte der externen Seiten ausgeschlossen. Für den Inhalt dieser externen Seiten sind ausschließlich deren Betreiber verantwortlich. Sollten Sie daher auf kostenpflichtige, illegale oder anstößige Inhalte treffen, so bedauern wir dies ausdrücklich und bitten Sie, uns umgehend per E-Mail davon in Kenntnis zu setzen, damit beim Nachdruck der Verweis gelöscht wird.

Druck und Bindung: Westermann Druck GmbH, Georg-Westermann-Allee 66, 38104 Braunschweig

ISBN 978-3-14-220072-9

Vorwort

Dieses Schülerbuch dient der Berufsausbildung in der Fachstufe der neu geordneten IT-Berufe. Die Ausbildungsordnung dieser Berufsgruppe ist seit August 2020 in Kraft. Herausgeber und Autoren der Buchreihe haben sich zum Ziel gesetzt, auf der Basis der neuen Ausbildungsordnung und des entsprechenden Rahmenlehrplans handlungs- und kompetenzorientierte Unterrichtsmedien und -hilfen zur Verfügung zu stellen.

Der Fachstufenband II bezieht sich im Schwerpunkt auf die Lernfelder 10a, 11a und 12a des Rahmenlehrplans der Berufsgruppe Fachinformatiker/-in Anwendungsentwicklung und baut auf den Lernfeldern 1–9 des Grundstufen- bzw. des Fachstufenbandes I auf. Da sich viele Inhalte von den Berufsgruppen Fachinformatiker/-in Anwendungsentwicklung und Fachinformatiker/-in für Daten- und Prozessanalyse überschneiden, kann das Buch aber auch bei der Ausbildung der Daten- und Prozessanalysten/-innen genutzt werden. Dazu wurde, zu den genannten Lernfeldern, noch das Lernfeld 10c „Werkzeuge des maschinellen Lernens“ als ein extra Kapitel in das Buch mit aufgenommen. Auch wenn das Lernfeld 10c nur in der Berufsgruppe Fachinformatiker/-in für Daten- und Prozessanalyse unterrichtet wird, bietet dieses Kapitel viele Inhalte, welche auch für die Fachinformatiker/-in Anwendungsentwicklung interessant sind und in den Unterricht mit aufgenommen werden können.

Wegen der vorgegebenen spirallcurricularen Vorgehensweise werden die erarbeiteten Kompetenzen der beiden vorangegangenen Jahrgänge vertieft und erweitert. Dazu werden in allen Lernfeldern neue Aufgabenbereiche erschlossen. Insbesondere gilt dieses für das Lernfeld 10c. Bezüge zum Grundstufen- und Fachstufenband I werden hergestellt und fördern damit eine durchgängige und effiziente Bereitstellung der vielfältig geforderten Kompetenzen. Auch für andere Schulformen mit einer IT-Orientierung, z. B. der BFS oder FOS Informatik sowie in der IT-Weiterbildung, kann diese Buchreihe eine handlungs- und praxisnahe Verwendung finden. Trotz der Stofffülle wurde der Stoffkatalog der Ausbildungsordnungen und der Prüfung so umfassend wie möglich behandelt. Möglichkeiten der weitergehenden Recherche werden jeweils angegeben.

Wir haben uns entschlossen, entweder genderneutral zu formulieren oder die männliche Form geschlechtsneutral und verallgemeinernd für Schülerinnen und Schüler zu verwenden. Wir bitten um Verständnis.

Alle Bände der Reihe stellen ihre Inhalte handlungsorientiert im Rahmen des Modellunternehmens JIKU IT-Solutions GmbH dar, eines innovativen Systemhauses der IT-Branche. Damit sollen zugleich ein stärkerer Praxisbezug und ein größerer Handlungsbezug erreicht werden.

Da die Ausbildungsordnung einerseits die kritische Auseinandersetzung mit Produkten der IT-Wirtschaft fordert, andererseits innovative Produkte und Verfahren i. d. R. an Unternehmen gebunden sind, lässt sich die Nennung von Produkten und Unternehmen sowie die Darstellung von Geschäftszeichen zur ersten Orientierung nicht vermeiden. Weder eine Herausstellung noch eine Werbung für diese Produkte ist damit beabsichtigt. Wenn möglich, wurden mehrere Produkte und ergänzend passende Informationsportale zum Vergleich und zur weiteren Recherche genannt.

Die digitale Transformation der Geschäftsprozesse bzw. die Digitalisierung und die damit einhergehenden Veränderungen in den Betrieben und im beruflichen Alltag sind das Hauptanliegen der IT-Berufe. Daher versuchen wir, neue und bedeutende Einsatzfelder und Entwicklungstrends zu erkennen und zu berücksichtigen.











Um viele Handlungsmöglichkeiten im Sinne der Ausbildungsordnung und des Rahmenlehrplans zu bieten, wird auch in den Schülerbüchern jedes Kapitel und möglichst jeder größere Lernabschnitt durch eine kurze Handlungssituation eingeleitet. Viele Kompetenzübungen sollen im Schülerbuch und im Arbeitsbuch die fachliche Auseinandersetzung und selbstständige Erarbeitung und Reflexion von Kompetenzen fördern. Eine Methodensammlung und Vorlagen im Downloadbereich der Buchreihe sollen die didaktische Jahresplanung und die kompetenzorientierte Lernarbeit unterstützen.

Im Schülerbuch werden wichtige Lerninhalte kompakt in Infoboxen bereitgestellt. Diese sollen eine zügige Orientierung und Bearbeitung der Kompetenzübungen, weitere Recherchen im Internet und das Nachschlagen für weitergehende Aufgaben erleichtern. Trotz teilweise gleicher Thematiken in der Ausbildung zum Fachinformatiker/-in Anwendungsentwicklung bzw. zum Fachinformatiker/-in für Daten- und Prozessanalyse können die Nutzer dieser Fachbuchreihe unterschiedliche Schwerpunkte setzen, je nach Beruf, Region und aktueller Situation, Kapitel unterschiedlich weit vertiefen oder auch zeitlich reduziert behandeln.

Insbesondere der Rahmenlehrplan fordert einen Unterricht nach Lernsituationen bzw. Lernarrangements. Um die vorgegebenen Handlungskompetenzen möglichst gut trotz unterschiedlicher Rahmenbedingungen entfalten zu können, wurde zum Schülerbuch passend ein Arbeitsbuch mit Lernsituationen entwickelt.

Ziel ist es hier, kompetenzorientiert und möglichst in vollständigen Handlungen zu arbeiten. Informieren und Planen der Lernsituationen soll über das Schülerbuch erleichtert werden. Daher werden im Arbeitsbuch Hinweise zu passenden Kapiteln im Schülerbuch gegeben. Zusätzliche Materialien im Arbeitsbuch und Dateien im Downloadbereich helfen, die Lernsituationen praxis- und handlungsbezogen zu gestalten. Es werden auch offene Handlungsaufgaben und bei gebundenen Aufgaben Varianten zugelassen, sodass die Lernenden in den Durchführungsphasen eigene Wege beschreiten können.

Zur leichteren Orientierung wurden folgende Symbole ergänzt:

	Definitionen und Erläuterungen
Kompetenzcheck 	Kompetenzübungen zur Kontrolle und Vertiefung
 A1	Verweis auf Arbeitsbuch
	Verweis auf freien Download
	Verweis auf freien Download – detaillierte Ansicht der Grafik
	Verweis auf Löser-Download
Lernsituation 1: 	Lernsituation
	Merksätze und wichtige Hinweise
	Situationsbeschreibung
	Wissenscontainer

Freie Downloads zum Schülerbuch sind als ergänzende Materialien unter <https://www.westermann.de/artikel/220072> zu finden.

Für Lehrkräfte ist zusätzlich ein Löser-Download unter <https://www.westermann.de/artikel/220076> erhältlich. Er enthält Lösungen, Lösungsvorschläge, komplette Lösungsdateien sowie Dateien für die Unterrichtsplanung. Bei offenen Aufgaben sollen insbesondere Selbst- und Reflexionskompetenzen gefördert werden. Hier können wegen des zu hohen Seitenaufwands keine Lösungen angeboten werden.

Das Arbeitsbuch ist unter <https://www.westermann.de/artikel/220080> erhältlich. Auch hierzu werden freie Downloads sowie ein Löser-Download für Lehrkräfte angeboten.

Gerne nehmen wir Anregungen oder Kritik unter service@westermann.de entgegen.

Inhaltsverzeichnis

	Vorwort	3
1	Benutzerschnittstellen gestalten und entwickeln (Lernfeld 10a)	7
	1.1 Anforderungen an eine Benutzerschnittstelle ermitteln	9
	1.1.1 Den Prozess der Anforderungsspezifikation präsentieren	10
	1.1.2 Arbeitsabläufe und Geschäftsprozesse im Rahmen einer Ist-Analyse beschreiben ...	13
	1.1.3 Anforderungskatalog erstellen	16
	1.2 Ein Softwareprodukt designen	20
	1.2.1 Benutzerschnittstellen in den Kontext der Softwarearchitektur einordnen	21
	1.2.2 Architekturmuster unterscheiden	24
	1.3 Benutzerschnittstellen funktionsgerecht und ergonomisch konzipieren	28
	1.3.1 Benutzerschnittstellen unterscheiden und Zukunftstrends präsentieren	28
	1.3.2 Kriterien zur Gestaltung und Bewertung von Benutzerschnittstellen präsentieren ...	33
	1.3.3 Den Designprozess und Designwerkzeuge präsentieren	37
	1.3.4 Grafische Benutzerelemente einer GUI unterscheiden und allgemeine Gestaltungskriterien präsentieren	40
	1.4 Benutzerschnittstellen in einer Programmiersprache implementieren	48
	1.4.1 Bibliotheken und Frameworks auswählen	49
	1.4.2 Benutzerschnittstellen in Python realisieren	51
	1.4.3 Benutzerschnittstellen in Java realisieren	65
	1.4.4 Webbasierte Benutzerschnittstellen realisieren	81
	1.5 Benutzerschnittstellen testen	96
	1.5.1 Grafische Benutzerschnittstelle testen	97
	1.5.2 Das Testen der User Experience beschreiben	102
	1.6 Benutzerschnittstellen dokumentieren	105
2	Funktionalität in Anwendungen realisieren (Lernfeld 11a)	109
	2.1 Modulare Softwarekomponenten planen	110
	2.1.1 Das Prinzip der Modularisierung präsentieren	111
	2.1.2 Softwarekomponenten mithilfe von UML beschreiben	113
	2.1.3 Entwurfsmuster unterscheiden und verwenden	126
	2.1.4 Clean-Code-Prinzipien präsentieren	131
	2.2 Datenstrukturen von Softwarekomponenten planen und implementieren	136
	2.2.1 Grundlegende Datenstrukturen beschreiben	136
	2.2.2 Datenstrukturen bedarfsgerecht auswählen	144
	2.2.3 Grundlegende Datenstrukturen von Python unterscheiden und einsetzen	146
	2.2.4 Grundlegende Datenstrukturen von Java unterscheiden und einsetzen	152
	2.3 Funktionalität von Softwarekomponenten planen und implementieren	158
	2.3.1 Algorithmen entwerfen und implementieren	158
	2.3.2 Rekursive Algorithmen umsetzen	165
	2.3.3 Sortierverfahren beschreiben, implementieren und anwenden	167
	2.3.4 Komprimierungsverfahren beschreiben, implementieren und anwenden	171
	2.3.5 Verschlüsselungsverfahren beschreiben, implementieren und anwenden	177
	2.3.6 Hashfunktionen beschreiben, implementieren und anwenden	182
	2.3.7 JSON-Dateien im- und exportieren und Webservices nutzen	187
	2.3.8 Funktionalität im Zusammenhang mit Datenbanken realisieren	192

2.4	Funktionalität von Softwarekomponenten testen	200
2.4.1	Testverfahren auswählen	200
2.4.2	Testfälle formulieren	203
2.4.3	Automatisierte Unittests entwerfen, implementieren und anwenden	207
2.5	Softwarekomponenten und deren Schnittstellen dokumentieren	211
3	Werkzeuge des maschinellen Lernens einsetzen (Lernfeld 10c)	215
3.1	Einsatzmöglichkeiten maschinellen Lernens analysieren	216
3.1.1	Einsatzbereiche maschinellen Lernens beschreiben	217
3.1.2	Die Wirtschaftlichkeit maschinellen Lernens beurteilen	219
3.1.3	Verfahren der künstlichen Intelligenz bewerten	221
3.2	Daten aus heterogenen Quellen analysieren und aufbereiten	223
3.2.1	Daten beziehen	224
3.2.2	Daten analysieren	227
3.2.3	Daten aufbereiten	232
3.3	Regelbasierte Verfahren zum maschinellen Lernen einsetzen	236
3.3.1	Regeln mit Entscheidungsbäumen beschreiben	238
3.3.2	Ein Expertensystem mit Prolog realisieren	239
3.3.3	Ein Vorhersagemodell präsentieren	241
3.4	Künstliche neuronale Netze zum maschinellen Lernen einsetzen	242
3.4.1	Aufbau und Funktionsweise künstlicher neuronaler Netze beschreiben	243
3.4.2	Werkzeuge für Deep-Learning-Verfahren auswählen	249
3.4.3	Ein Deep-Learning-Verfahren mit TensorFlow umsetzen	251
4	Kundenspezifische Anwendungsentwicklung durchführen (Lernfeld 12a)	253
4.1	Das Basiswissen über Projekte und Projektmanagement wiederholen und erweitern	254
4.1.1	Projektmanagementstandards unterscheiden	256
4.1.2	Die Projektorganisation innerhalb von Unternehmen präsentieren	257
4.2	Kundenaufträge im Rahmen von Projekten planen und bearbeiten	263
4.2.1	Zielsetzung des Kundenauftrags erfassen	264
4.2.2	Das Projekt planen und die Wirtschaftlichkeit beurteilen	267
4.2.3	Projekt durchführen	275
4.2.4	Die Softwarelösung testen und dokumentieren	278
4.2.5	Die Softwarelösung präsentieren	280
4.2.6	Das Projektergebnis bewerten und die Projektdurchführung reflektieren	283
	Sachwortverzeichnis	285
	Bildquellenverzeichnis	292

Um solche Benutzerschnittstellen umzusetzen, bedarf es mehrerer Schritte. Zunächst muss der Prozess, welcher digital abgebildet werden soll, genau analysiert werden. Aus dieser Analyse werden die Anforderungen an das zukünftige Softwareprodukt abgeleitet. Auf Grundlage dieser Anforderungen werden danach u. a. die Benutzerschnittstellen dann geplant und implementiert.



Bei der Gestaltung der Benutzerschnittstellen spielen viele Faktoren eine Rolle. Diese reichen von der Art der Benutzerschnittstelle, über das UX Design bis hin zu Fragen der barrierefreien Gestaltung. In vielen Fällen müssen auch mehrere, sich stark unterscheidende Benutzerschnittstellen für verschiedene Zielgruppen einer Software entwickelt werden. Für die Umsetzung der Benutzerschnittstellen stehen in der Regel verschiedene Frameworks zur Verfügung, deren Vor- und Nachteile sorgfältig gegeneinander abgewogen werden müssen.

Die Inhalte dieses Lernfelds werden an verschiedenen Beispielen verdeutlicht. Die Entwicklung, das Testen und die Dokumentation einer Benutzerschnittstelle für die Liegeplatzverwaltung eines „Yachthafen Resorts“ dient dabei als Grundlage:

Softwareprojekt „Yachthafen Resort“



Für ein „Yachthafen Resort“ soll eine umfangreiche Software entwickelt werden, welche u. a. eine Mitarbeiter-, Gebäude-, Event- und Buchungsverwaltung umfasst. Im Rahmen dieses Projektes soll zudem eine Liegeplatzverwaltung erstellt werden.

Die wesentlichen Anforderungen für das Gesamtprojekt wurden bereits erfasst und liegen vor. Die einzelnen Module müssen noch spezifiziert, geplant und entwickelt werden.



1.1 Anforderungen an eine Benutzerschnittstelle ermitteln

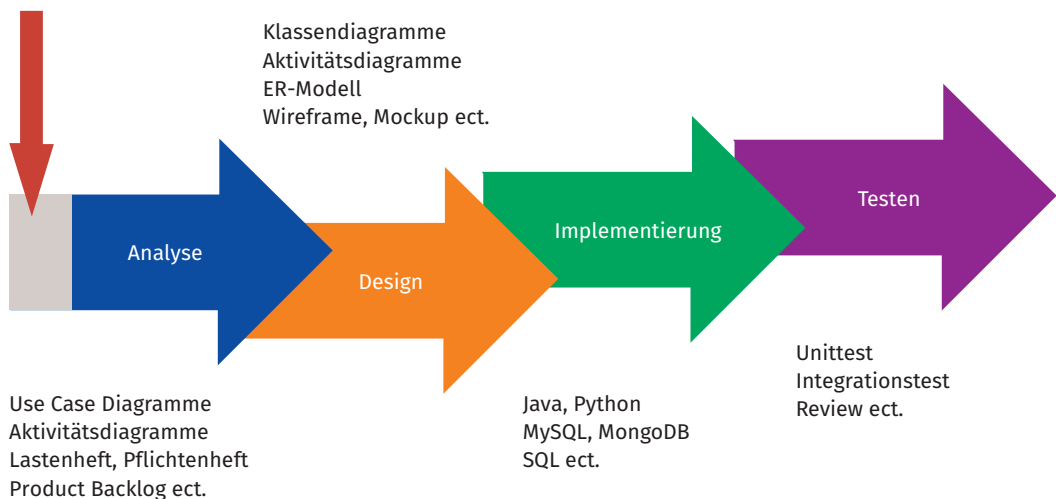
S Sie sollen Ihr Basiswissen zur Ermittlung der Anforderungen an ein Softwareprodukt festigen und erweitern.

Ein erster Schritt bei der Umsetzung von Softwareprojekten ist die Erfassung der Kundenbedürfnisse. Diese zu kennen und zu erfüllen ist die Grundlage für den Erfolg eines Unternehmens. Auf den ersten Blick scheint die Erfassung der Kundenwünsche relativ einfach zu sein: Man muss den Kunden doch einfach nur fragen, was er haben möchte. Leider erweist sich in der Praxis, dass die Bestimmung der Kundenerwartungen und der sich daraus ergebenden Anforderungen an die Software, weitaus schwieriger ist als zunächst angenommen.



Die Erfahrung zeigt, dass es den meisten Kunden eher schwerfällt, die Anforderungen an die zukünftige Software genau zu formulieren. Sie wissen meist nur ungefähr, was die Software leisten soll. Das Spektrum ihrer Erwartungen und Anforderungen kann von leicht zu erfassenden Punkten bis hin zu Kriterien verlaufen, welche der Kunde zwar hat und voraussetzt, aber nicht explizit erwähnt. So kann ein Projekt durch unzureichende, falsche oder erst gar nicht definierte Anforderungen scheitern. Deswegen wird in der Analysephase des Softwareentwicklungsprozesses eine genaue Anforderungsspezifikation vorgenommen (siehe auch Jahrgangsband 1, Lernfeld 5).

Anforderungsspezifikationen



Je nach Größe des Projektes kann die Bestimmung der Anforderungen an ein Softwareprodukt ein einmaliger oder ein sich wiederholender Vorgang sein. Bei größeren Neuentwicklungen werden zunächst die Anforderungen an das Gesamtprojekt ermittelt und danach Schritt für Schritt für die einzelnen Bereiche verfeinert, z.B. Benutzerschnittstellen oder bestimmte Module. In anderen Fällen müssen bestehende Anwendungen bloß um neue Module erweitert werden, z.B. um eine neue Benutzerschnittstelle. Auch in diesem Fall wird zunächst eine Anforderungsspezifikation vorgenommen, auch wenn diese bei Weitem nicht so umfangreich ausfallen wird wie bei einer Neuentwicklung.

Bei einer Anforderungsspezifikation werden zunächst im Rahmen einer Ist-Analyse die bestehenden Prozesse und deren Informationsfluss analysiert. Auf Grundlage dieser Analyse werden danach die Anforderungen an die zu entwickelnde Software bestimmt und festgehalten, z.B. in Lastenheften oder Product Backlogs. Diese bilden dann wiederum die Grundlage für das Design der Software, z.B. die Softwarearchitektur oder die Benutzeroberfläche.

1.1.1 Den Prozess der Anforderungsspezifikation präsentieren

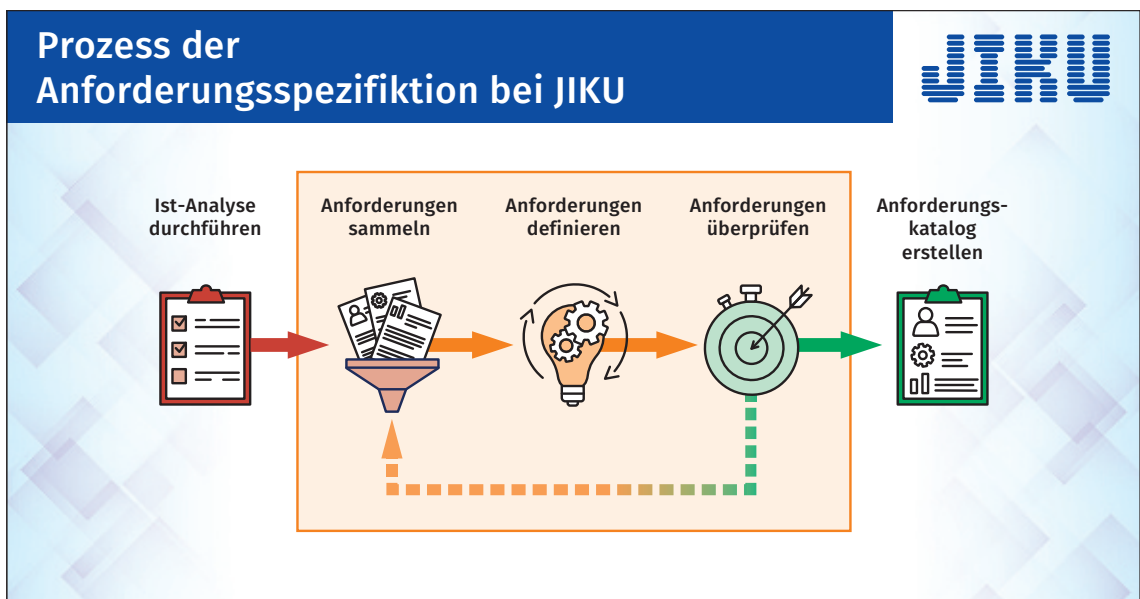
S Sie sollen sich die einzelnen Prozessschritte der Anforderungsspezifikation erarbeiten und Ihr Ergebnis präsentieren.

Am Anfang eines jeden Softwareentwicklungsprozesses steht eine Anforderungsspezifikation. Dabei spielt es zunächst keine Rolle, nach welchem Vorgehensmodell die Software entwickelt werden soll. Allerdings hängt es später stark vom Vorgehensmodell ab, wann und wie häufig eine Anforderungsspezifikation vorgenommen wird. Beim klassischen Wasserfallmodell (siehe auch Jahrgangsband 1, Lernfeld 5) wird beispielsweise eine Anforderungsanalyse nur einmal und am Anfang des Projektes durchgeführt. Bei agilen Vorgehensmodellen hingegen, z.B. Scrum (siehe auch Jahrgangsband 2, Lernfeld 8), wird diese jeweils vor dem nächsten Entwicklungszyklus wiederholt und die vorliegenden Anforderungen werden ergänzt, angepasst und verfeinert.



Anforderungsspezifikation

Der Prozess der Anforderungsspezifikation umfasst das Ermitteln, Analysieren, Spezifizieren und Validieren (Überprüfen) aller Anforderungen an das zu entwickelnde Softwaresystem.



Die Anforderungsspezifikation selbst ist ebenfalls ein Prozess, welcher in verschiedene Schritte unterteilt werden kann, deren Ergebnis am Ende ein Anforderungskatalog ist. Die Form und Bezeichnung dieses Katalogs hängt wieder stark von dem verwendeten Vorgehensmodell ab. Dieser kann z. B. ein Lastenheft beim Wasserfallmodell oder ein Product Backlog bei Scrum sein, wobei ein Product Backlog nie vollständig ist und während der Entwicklung der Software regelmäßig durch neue Anforderungsspezifikationen angepasst wird.

Prozessschritte der Anforderungsspezifikation

1. Ist-Analyse

Die Ist-Analyse ist ein erster wichtiger Schritt im Prozess der Anforderungsspezifikation. Zunächst geht es um die Schaffung eines Gesamtüberblicks über alle für das Projekt wichtigen Abläufe und Prozesse, welche im Anschluss genau definiert werden. Ziel der Ist-Analyse ist es, vorhandene Defizite und Optimierungspotenziale bei den bestehenden Prozessen aufzudecken. Durch eine genaue Ist-Analyse soll sichergestellt werden, dass nur wirklich benötigte Funktionalitäten des zukünftigen Softwaresystems im Anforderungskatalog erfasst werden.

Folgende Punkte können in einer Ist-Analyse untersucht werden:

- Aufgabe des Prozesses: Art und Umfang aller anfallenden Aufgaben und deren Besonderheiten
- Struktur des Prozesses: die Organisation des Prozessablaufes
- Kommunikation innerhalb des Prozesses: Ablauf und Vorrichtungen, welche zur Kommunikation genutzt werden
- Prozessdaten: Art und Umfang der im Prozess zu verarbeitenden Daten
- Prozessdokumentation: Art und Umfang der Dokumentation des Prozesses
- Schwachstellen des Prozesses: bekannte Mängel, Behinderungen oder Redundanzen des Prozesses

2. Anforderungen sammeln

In diesem Schritt werden die Anforderungen an das zu entwickelnde Softwaresystem zusammengetragen. Die Grundlage dafür bildet die Ist-Analyse. Dabei sollten auch weitere Anforderungen berücksichtigt werden, welche z. B. vom Management oder zusätzlichen Nutzern kommen. Das Ziel dieses Schrittes sollte eine vollständige Bedarfsübersicht sein, deren Inhalte beispielsweise in folgende Kategorien unterteilt werden können:

- funktionale Anforderungen
- technische Anforderungen
- organisatorische Anforderungen
- allgemeine Kriterien

3. Anforderungen definieren

Aus den gesammelten Anforderungen werden nun konkrete Anforderungen an das Softwareprodukt definiert und ausformuliert. Dabei sind u. a. folgende Überlegungen einzubeziehen:

- Können die Anforderungen überhaupt erfüllt werden?
- Welche technischen Einschränkungen gibt es?
- Welche Kosten fallen an und welche Ressourcen sind nötig?
- Welche Risiken existieren?

Die konkreten Anforderungen werden nun Softwarekriterien zugeordnet und in funktionale und nicht funktionale Anforderungen klassifiziert.

4. Anforderungen überprüfen

In vielen Fällen können nicht alle gestellten Anforderungen berücksichtigt werden. Das mag zum einen daran liegen, dass sie technisch nicht machbar oder in dem gegebenen Zeitraum nicht umsetzbar sind. Auf der anderen Seite sind nicht immer alle Anforderungen unbedingt für die Kernfunktion der Software notwendig. Daher sollten die Anforderungen dahingehend überprüft werden, ob sie im Anforderungskatalog verbleiben und dort dann priorisiert werden. Mögliche Priorisierung wäre:

- nicht erforderlich
- nice-to-have
- unbedingt erforderlich

5. Anforderungskatalog fertigstellen und genehmigen

In diesem Schritt wird der Anforderungskatalog fertiggestellt. Dieser kann sich, je nach Vorgehensmodell im Aufbau und in der Bezeichnung unterscheiden (z. B. Lastenheft, Product Backlog). In einigen Fällen ist der Inhalt dann feststehend, in anderen wird dieser in sich wiederholenden Durchläufen angepasst bzw. erweitert. Zu einem Anforderungskatalog sollten in jedem Fall alle Beteiligten ihre Zustimmung geben, bevor auf dessen Grundlagen das Projekt weiter vorge setzt wird.

Für die Erfassung der notwendigen Informationen bei der Anforderungsanalyse gibt es verschiedene Methoden. Einige davon werden nachfolgend genauer betrachtet.

Methoden zur Erhebung des Istzustandes und der Anforderungen	
Interview	<p>Beim Interview werden die Beteiligten in Form eines mündlichen Fragebogens zu einem oder mehreren Themenkomplex befragt.</p> <p>Vorteile:</p> <ul style="list-style-type: none"> • persönlicher Kontakt fördert die Bereitschaft zur Auskunft • gestellte Fragen werden mit großer Wahrscheinlichkeit auch beantwortet • Unklarheiten können direkt geklärt werden <p>Nachteile:</p> <ul style="list-style-type: none"> • hoher zeitlicher Aufwand • subjektive Prägung der Antworten • erschwerte Erfassung von Faktendaten
Fragebogen	<p>Hierbei füllen die Beteiligten einen Fragebogen aus. Dieser kann mit Multiple-Choice-Fragen und offene Fragen enthalten.</p> <p>Vorteile:</p> <ul style="list-style-type: none"> • gleichzeitige Informationsgewinnung innerhalb eines großen Personenkreises • weniger zeitaufwendig als ein Interview oder eine Beobachtung • keine zusätzliche Dokumentation der Antworten erforderlich • Möglichkeit der automatischen Auswertung der Antworten <p>Nachteile:</p> <ul style="list-style-type: none"> • höherer Vorbereitungsaufwand als beim Interview • der Rücklauf der Fragebögen ist manchmal unzureichend • Unklarheiten lassen sich schwer klären
Beobachtung	<p>Hierbei werden vor Ort direkt Prozesse begleitet und analysiert.</p> <p>Vorteile:</p> <ul style="list-style-type: none"> • Prozesse sind objektiv zu beobachten • sehr effektive Erfassung möglich • besonders für die Erfassung zeitlicher und quantitativer Informationen über den Prozess geeignet <p>Nachteile:</p> <ul style="list-style-type: none"> • zeitlicher Aufwand sehr hoch • es können teilweise nur kleine Ausschnitte eines Prozesses beobachtet werden • Beobachtungen werden i. d. R. aus unangenehm empfunden

Bei allen diesen Methoden müssen verschiedene, auf den Prozess abgestimmte Fragen beantwortet werden. Diese werden im Vorfeld gründlich zusammengestellt. Nachfolgend eine Übersicht an Fragen, welche im Rahmen eines Interviews, eines Fragebogens oder einer Beobachtung eine Rolle spielen können.

Beispielfragestellungen

- Um was für einen Prozess handelt es sich?
- Ist es ein Hauptprozess oder ein Unterprozess?
- Welche Tätigkeiten werden im Prozess in welcher Reihenfolge ausgeführt?
- Welche Prozessschnittstellen gibt es?

- Welche Qualitätsanforderungen sind für einen optimalen Prozessablauf erforderlich?
- Welche Daten werden wann, von wo, wie oft für den Prozess benötigt?
- Wer fällt, welche Entscheidungen in dem Prozess?
- Welche Aspekte der Informationssicherheit sind in welchem Umfang für den Prozess relevant?

Zusätzlich gibt es noch andere Varianten zur Erfassung der notwendigen Informationen, z. B. Brainstorming, das Sichten von Dokumenten oder Simulationen, welche teilweise im Rahmen von Workshops durchgeführt werden. In der Regel sollten verschiedene Methoden zum Einsatz kommen. Einen idealen Weg gibt es dabei aber nicht. Vielmehr sollte die Form der Analyse dem Kunden angepasst werden, um optimalen Ergebnisse zu erzielen.

Kompetenzcheck ✓

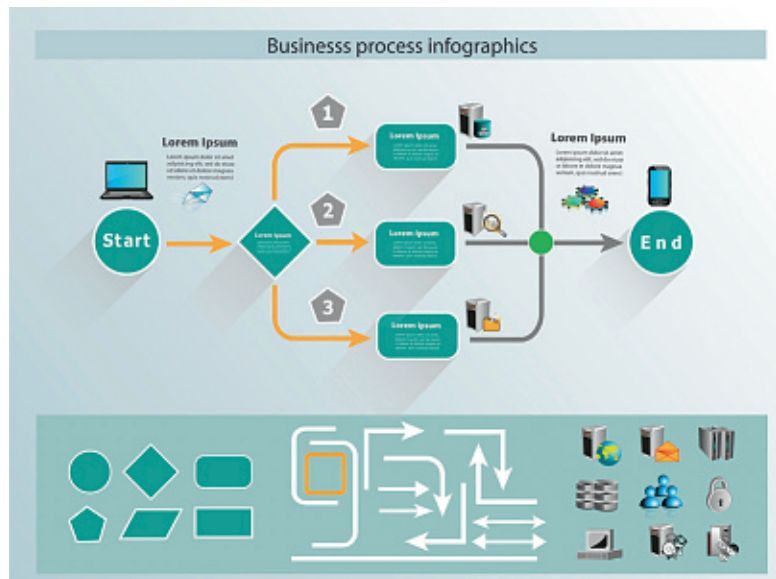
- 1 Geben Sie an, welche Aussagen richtig sind.
 - a) Eine Anforderungsspezifikation wird grundsätzlich immer nur einmal durchgeführt, um spätere Verwirrungen zu vermeiden.
 - b) Die Ist-Analyse ist ein wichtiger Bestandteil der Anforderungsspezifikation.
 - c) Bei der Ist-Analyse wird u. a. die Verarbeitung der am Prozess beteiligten Daten erfasst.
 - d) Die erfassten Anforderungen an die Software werden in funktionale und nicht funktionale Kriterien unterteilt.
 - e) Das Ergebnis der Anforderungsanalyse ist in jedem Fall ein Lastenheft.
 - f) Interviews und Fragebögen sind Möglichkeiten, um die notwendigen Informationen bei der Anforderungsspezifikation zu erfassen.
- 2 Bearbeiten Sie im Arbeitsbuch von der Lernsituation 1 die Aufgaben 1 und 2.


 A1,2

1.1.2 Arbeitsabläufe und Geschäftsprozesse im Rahmen einer Ist-Analyse beschreiben

S Sie sollen Arbeitsabläufe und Geschäftsprozesse analysieren und modellhaft darstellen.

Der Zweck der Entwicklung von Software besteht u. a. darin, Arbeitsabläufe oder Prozesse in Unternehmen zu automatisieren. Dazu müssen diese gewisse Voraussetzungen erfüllen: Zum einen müssen die Abläufe klar beschreibbar sein und zum anderen müssen sie sich in der gleichen Form wiederholen. Der letzte Punkt ist Voraussetzung dafür, dass sich die Automatisierung überhaupt lohnt. Laufen Prozesse nur einmal oder selten ab, kann im normalen wirtschaftlichen Umfeld die Steuerung durch den Menschen kostengünstiger sein. Um festzustellen, welche Prozesse mithilfe



von Softwareanwendungen ganz oder teilautomatisiert werden können, müssen diese und ihre Schnittstellen im Rahmen von Softwareprojekten zunächst analysiert werden. Daraus werden dann die Anforderungen an die zu entwickelnde Software abgeleitet. Dieses erfolgt im Rahmen einer Ist-Analyse (siehe auch S. 11). Dabei werden die einzelnen Aspekte eines Prozesses analysiert und erfasst.



Grundlegende Begriffe

Prozessanalyse

Eine Prozessanalyse ist eine systematische Untersuchung von Prozessen, um eventuelle Schwachstellen zu erkennen und Verbesserungspotenziale aufzudecken.

Prozess

Unter einem Prozess versteht man die Gesamtheit aller aufeinander einwirkenden Vorgänge innerhalb eines Systems.

Geschäftsprozess

Ein Geschäftsprozess ist eine sich wiederholende Abfolge von Tätigkeiten die durch einen Input gestartet werden und welche zu einem wirtschaftlich verwertbaren Ergebnis führen.

Prozessschnittstelle

Eine Prozessschnittstelle ist ein Punkt in einem Prozess, an dem der Prozess mit anderen Prozessen in Verbindung tritt. Typische Prozessschnittstellen sind:

- Eingang von Informationen oder Materialien am Anfang oder im Verlaufe des Prozesses
- Austausch von Informationen oder Materialien während des Prozesses
- Ausgang des Prozessergebnisses in Form von Informationen oder Material am Ende des Prozesses

Informationsfluss (Datenfluss)

Der Informationsfluss beschreibt den Weg, welchen mündliche, schriftliche, visuelle oder auditive Daten nehmen, um von einer Quelle zu den Empfängern zu gelangen. In der Informationstechnik spricht man in diesem Zusammenhang auch von Datenfluss.

Eine Prozessanalyse teilt sich in folgende Schritte auf.



Zunächst muss festgestellt werden, um welche Art von Prozess es sich handelt. Je nach Unternehmen oder Einrichtung kann es sich dabei um sehr unterschiedliche Prozesse, z. B. technische, politische, soziologische oder betriebswirtschaftliche (Geschäftsprozesse) Prozesse handeln. Exemplarisch werden hier Geschäftsprozesse näher vorgestellt. Geschäftsprozesse können wiederum in weitere Prozesse unterteilt werden.

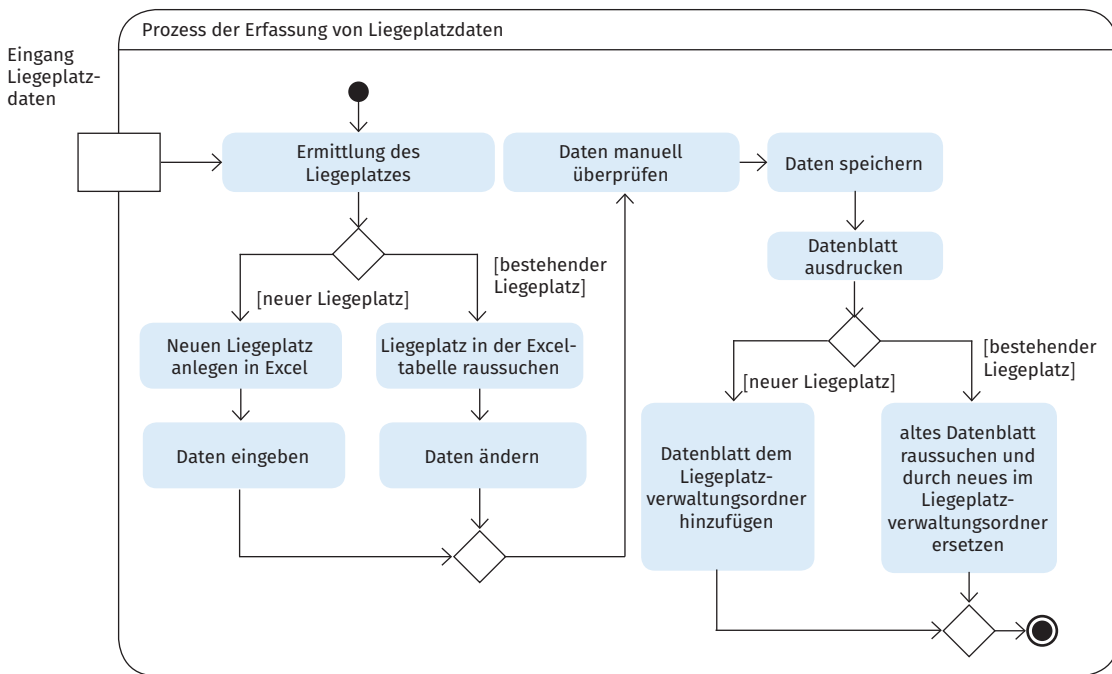


Auch diese Prozesse bestehen wieder aus verschiedenen Teilprozessen. Da alle diese Teilprozesse wiederum von verschiedenen Abteilungen oder Funktionseinheiten ausgeführt werden können, ist es mitunter sehr schwer, einen Überblick über den Gesamtprozess zu bekommen.

Sobald die Art des Prozesses ermittelt wurde, wird mit der Aufnahme des Prozesses begonnen. Dazu wird der Prozess erst einmal in seine einzelnen Teilschritte zerlegt und grafisch dargestellt. Ein wichtiger Punkt dabei ist auch die Aufnahme und Darstellung der Prozessschnittstellen und des Informationsflusses.

Die Darstellung der Prozesse erfolgt i. d. R. mithilfe eines Business Process Model (BPM), von einem E3-Value-Model oder von ereignisgesteuerten Prozessketten (EPK). Eine weitere Möglichkeit besteht in der Darstellung in Form einer Wertschöpfungskette. Aber auch ein UML-Aktivitätsdiagramm (siehe auch Kapitel 2.1.2) oder schlicht ein PAP (siehe auch Jahrgangsband 1, Lernfeld 5) eignen sich dazu. Durch eine anschauliche Darstellung werden die Prozesse für alle am Softwareprojekt Beteiligten nachvollziehbar.

Nachfolgend soll im Rahmen des Projektes zur Erstellung einer Software für das „Yachthafen Resort“ der Prozess der Erfassung der Liegeplatzdaten untersucht und optimiert werden. Dieser Prozess ist ein Teilprozess des Kernprozesses „Liegeplatzvermietung“. Dazu wurde mit den am Prozess beteiligten Personen ein Interview geführt und die daraus gewonnenen Informationen über den bisherigen Prozessablauf mithilfe eines UML-Aktivitätsdiagramms dargestellt.



Nachdem der Prozess erfasst wurde, wird dieser bewertet und analysiert. Hier geht es vor allem darum, Schwachstellen zu erkennen und Verbesserungspotenzial aufzudecken. Des Weiteren wird dabei bestimmt, welche Teile des Prozesses automatisiert werden können. In diesem Fall werden, wie im UML-Diagramm ersichtlich ist, die Daten mithilfe von Excel erfasst und verwaltet. Gleichzeitig werden die Daten auch noch in gedruckter Form in einem entsprechenden Ordner abgelegt. Dieser Prozess ist relativ langsam und fehleranfällig. Durch die redundante Verwaltung der Daten kann es beispielsweise zu Dateninkonsistenzen kommen. Im Rahmen des Softwareprojektes für das „Yachthafen Resort“ soll u. a. dieser Prozess optimiert und teilweise automatisiert werden, z. B. die Überprüfung der eingegebenen Daten. Dieses und alle anderen Ergebnisse der Analyse des jeweiligen Prozesses münden in der Formulierung der Anforderungen an das zukünftige Softwareprodukt.

Kompetenzcheck ✓

- 1 Geben Sie an, welche Aussagen richtig sind.
 - a) Die Analyse von Prozessen und deren Schnittstellen ist eine wesentliche Voraussetzung für die Bestimmung der Anforderungen an das zukünftige Softwareprodukt.
 - b) Geschäftsprozesse können in Management-, Kern- und Unterstützungsprozesse eingeteilt werden.
 - c) Die Montage von Autoteilen gehört bei einem Autohersteller zu den Unterstützungsprozessen.
 - d) Der Prozess der Qualitätssicherung gehört bei einem Autohersteller zu den Managementprozessen.
 - e) Der Vertrieb der Autos gehört zu den Kernprozessen eines Autoherstellers.
 - f) Eine Hardware- oder Softwareschnittstelle kann eine Prozessschnittstelle sein.
 - g) Prozesse können mithilfe von BPMN (Business Process Model and Notation) dargestellt werden.
- 2 Entwickeln Sie in Partnerarbeit Ideen, wie der Prozess der Erfassung der Liegeplatzdaten im Rahmen des Gesamtprojektes optimiert und automatisiert werden kann. Diskutieren Sie Ihre Ideen im Klassenverband.
- 3 Führen Sie für einen Prozess eine Ist-Analyse durch und stellen Sie diesen grafisch dar. Bearbeiten Sie dazu im Arbeitsbuch die Aufgaben 3, 4 und 5 der Lernsituation 1.

A3,4,5

1.1.3 Anforderungskatalog erstellen

S Sie sollen auf der Grundlage der Ist-Analyse die Anforderungen an eine Benutzerschnittstelle formulieren.

Aus den Ergebnissen der Ist-Analyse werden konkrete Anforderungen an das Softwareprodukt definiert und ausformuliert. Diese werden im Anschluss daran noch einmal dahingehend überprüft, ob sie überhaupt notwendig sind und sie technisch und zeitlich zu realisieren sind.

Die verbliebenen Anforderungen werden sodann Softwarekriterien zugeordnet und in funktionale und nicht funktionale Anforderungen klassifiziert (siehe auch Jahrgangsband 1, Lernfeld 5).



Funktionale Anforderungen beziehen sich darauf, was das System leisten soll. Diese sind von Software zu Software sehr unterschiedlich und spezifisch. Bei den funktionalen Anforderungen werden die benötigten Eingabedaten, die erforderlichen Verarbeitungsschritte und die erwarteten Ausgabedaten genau beschrieben. Nicht funktionale Anforderungen beziehen sich darauf, wie das System seine Leistungen erbringen soll. Sie werden in der Anforderungsspezifikation genau wie die funktionalen Anforderungen beschrieben. Die nicht funktionalen Anforderungen werden zudem in Qualitätsanforderungen und Randbedingungen unterteilt. Die Qualitätsanforderungen richten sich dabei meistens an Softwarequalitätskriterien wie z. B. Effizienz oder Wartbarkeit aus, während die Randbedingungen technologische, organisatorische und normative Anforderungen beschreiben.

Nach der Einteilung der Anforderungen in verschiedene Kategorien sollten diese zum Schluss noch nach Wichtigkeit priorisiert werden, um dem Kunden zügig erste Ergebnisse vorstellen zu können. Wie die einzelnen Anforderungen beschrieben werden und wie der Anforderungskatalog später aussieht, hängt viel von dem verwendeten Vorgehensmodell ab. So werden beispielsweise bei Scrum (siehe auch Jahrgangsband 2, Lernfeld 8) zur Beschreibung der Anforderungen Epics und User Stories verwendet. Diese werden immer aus der Perspektive des Benutzers geschrieben, um ein besseres Verständnis für

die Belange des Benutzers zu entwickeln. Es wird zunächst mit der Beschreibung der Epics begonnen. Epics geben grob die wichtigsten Anforderungen an die Software wieder. Diese werden dann später in einzelne User Stories aufgeteilt, welche mindestens aus ein oder zwei Sätzen bestehen. Im Anforderungskatalog, z.B. dem Product Backlog bei Scrum, werden alle Anforderungen an die zukünftige Software zusammengefasst. Da es in diesem Lernfeld schwerpunktmäßig um die Entwicklung von Benutzerschnittstellen für Software geht, werden nachfolgend nur die Anforderungen an diese näher betrachtet.

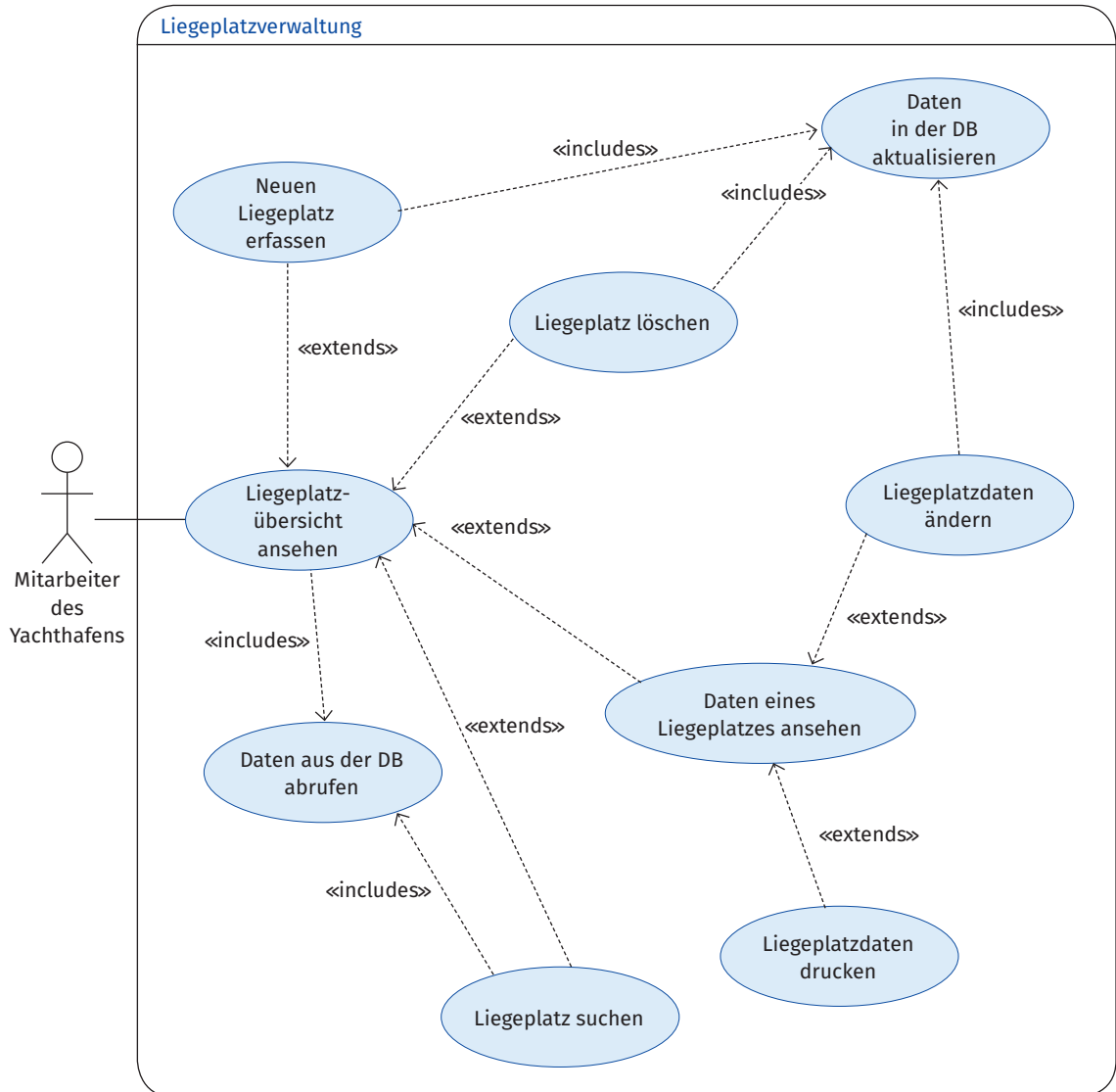
(1) Funktionale und nicht funktionale Anforderungen an Benutzerschnittstellen

Die funktionalen Anforderungen an eine Benutzerschnittstelle ergeben sich zunächst aus den funktionalen Anforderungen an die Anwendung. Durch die Benutzerschnittstelle muss der Nutzer in die Lage versetzt werden, mit der Anwendung zu interagieren, um z.B. dem System entsprechende Eingabedaten zur Verfügung zu stellen, welche dann weiterverarbeitet werden. Weiterhin zählt zu den funktionalen Anforderungen, welche Endgeräte zum Einsatz kommen sollen. Oftmals sind mobile Versionen der Benutzerschnittstelle notwendig, welche auch eine andere Art der Interaktion erfordern. Bei mobilen Versionen kann sich die Benutzeroberfläche im Gegensatz zu Desktopanwendungen erheblich unterscheiden. Neben der „fachlichen“ Funktionalität gibt es aber auch Anforderungen an eine Funktionalität, welche rein auf die Interaktion des Benutzers mit der Oberfläche ausgerichtet ist, z.B. Scroll- oder Zoomfunktionen. Im Bereich der nicht funktionalen Anforderungen spielt bei den Qualitätsanforderungen das Qualitätskriterium „Benutzbarkeit“ die größte Rolle. In diesem Bereich sind aber auch Zuverlässigkeit und Effizienz sehr wichtig.

Funktionale und nicht funktionale Anforderungen an eine Benutzerschnittstelle		
Funktionale Anforderungen	fachlich; bezogen auf den Arbeitsablauf oder Geschäftsprozess	<ul style="list-style-type: none"> • Eingabe und LösCHFunktionalität für notwendige Daten des Prozesses • Anzeigen von Grafiken und Bildern mit Inhalten des jeweiligen Prozesses
	bezogen auf die Interaktion mit der Benutzerschnittstelle	<ul style="list-style-type: none"> • Bedienbarkeit, z. B. mit Maus, Tastatur, Sprache • Scroll- und Zoomfunktionen • Anpassbarkeit des Aussehens der Oberfläche
	bezogen auf geplante Endgeräte und Softwareumgebungen	<ul style="list-style-type: none"> • Tablets, Smartphone, Desktopanwendung, stationäre Touch Panel für Automaten • Betriebssysteme, Browser
Nicht funktionale Anforderungen	Qualitätsanforderungen	<p>Benutzbarkeit:</p> <ul style="list-style-type: none"> • Anforderungen an die Selbstbeschreibbarkeit • Anforderungen an die Erwartungskonformität <p>Zuverlässigkeit:</p> <ul style="list-style-type: none"> • Anforderungen an die Fehlertoleranz der Benutzerschnittstelle <p>Effizienz:</p> <ul style="list-style-type: none"> • Anforderungen an die Darstellung, z. B. von 3-D-Animationen • Anforderungen an den Ressourcenverbrauch, z. B. bei speziellen Endgeräten
	technische, organisatorische und normative Rahmenbedingungen	<ul style="list-style-type: none"> • vorgegebene Betriebssysteme und Browser • Rechtvergabe für bestimmte Funktionalität • Grad der Barrierefreiheit oder Mehrsprachigkeit

(2) Anforderungskatalog für das Beispielszenario

An dieser Stelle sollen nun exemplarisch die Anforderungen an die Benutzerschnittstelle für die Liegeplatzverwaltung bestimmt werden. Als Grundlage dafür dient das abgebildete Anwendungsfalldiagramm (siehe auch Kapitel 2.1.2 oder Jahrgangsband 2, Lernfeld 8), welches die Grundfunktionalität der Liegeplatzverwaltung beschreibt.



Da ein Anwendungsfalldiagramm auch immer die Interaktion eines oder mehrerer Benutzer mit dem System abbildet, können hieraus unmittelbar Anforderungen an die Benutzerschnittstelle abgeleitet werden. In diesem Fall kann daraus abgeleitet werden, dass die Benutzerschnittstelle u. a. Möglichkeiten zum Hinzufügen und Löschen von Liegeplätzen bereitstellen muss.

In der folgenden Vorlage hat eine Auszubildende der JIKU IT-Solutions die Anforderungen für die Benutzerschnittstelle für die Liegeplatzverwaltung erfasst.

Anforderungen an die Benutzerschnittstelle

Liegeplatzverwaltung



Kriterien	Anforderungen
Funktionale Anforderungen (prozessbezogen)	<p>Die Benutzeroberfläche soll Nutzern folgende Funktionalität bieten:</p> <ul style="list-style-type: none"> • alle Liegeplätze in einer zentralen Übersicht ansehen, • bestimmte Liegeplätze suchen, • Liegeplätze nach bestimmten Kriterien filtern, • neue Liegeplätze erfassen, • bestehende Liegeplätze löschen • Daten bestehender Liegeplätze zu ändern.
Funktionale Anforderungen (interaktionsbezogen)	<p>Es soll eine grafische Benutzeroberfläche (GUI) entwickelt werden, welche mit der Maus und der Tastatur bedient werden kann.</p>
Funktionale Anforderungen (technisch)	<p>Es soll zunächst nur eine Desktop-Oberfläche für stationäre Computersysteme entwickelt werden (Betriebssystem: Windows).</p>
Nicht funktionale Anforderungen (Qualitätsanforderungen)	<p>Die Oberfläche soll auf der Grundlage der Interaktionsprinzipien ISO-Norm 9241-110 gestaltet werden. Außerdem ist auf ein gutes UX Design zu achten.</p>
Nicht funktionale Anforderungen (Rahmenbedingungen)	<p>Es sind die Vorgaben zur Corporate Identity des „Yachthafen Resorts“ zu beachten und umzusetzen, z. B. die Verwendung des Logos des Unternehmens.</p>

Kompetenzcheck ✓

- Geben Sie an, welche Aussagen richtig sind.

 - Angaben zur Wartbarkeit einer Software gehören zu den funktionalen Anforderungen.
 - Anforderungen an die Fehlertoleranz einer Software zählen zu den nicht funktionalen Anforderungen.
 - Rahmenbedingungen, z. B. den Einsatz der Software im deutschsprachigen und nicht deutschsprachigen Raum kann eine mehrsprachige Gestaltung der Benutzerschnittstelle notwendig machen.
 - Wenn die Benutzerschnittstelle 3-D-Animationen aufweisen soll, können Fragen der Effizienz eine große Rolle spielen.
 - Betriebssysteme und Browser spielen bei den Anforderungen an eine Benutzerschnittstelle überhaupt keine Rolle.
 - Die Sprachausgabe von Inhalten gehört zu interaktionsbezogenen Funktionalität von Benutzerschnittstellen.
 - Vorgaben für Farben oder Schriftarten müssen nicht erfasst werden, da diese für die Gestaltung der Benutzerschnittstelle nicht von Bedeutung sind.
- Finden Sie in Gruppenarbeit auf Grundlage der hier vorgestellten Beispiele weitere funktionale und nicht funktionale Anforderungen für Benutzerschnittstellen. Diskutieren Sie Ihre Ergebnisse im Klassenverband.
- Bearbeiten Sie im Arbeitsbuch die Aufgabe 6 der Lernsituation 1 und überprüfen Sie im Rahmen der Prüfungsvorbereitung Ihr Wissen bezüglich der Anforderungs- und Prozessanalyse mithilfe der Aufgabe 7.

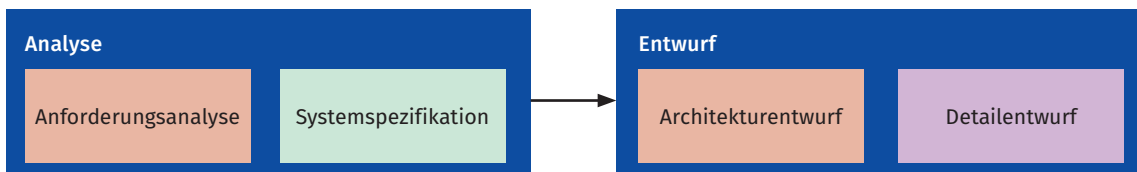


A6,7

1.2 Ein Softwareprodukt designen

S Sie sollen Ihr Basiswissen über das Planen und Entwerfen von Software vertiefen und erweitern.

Das Designen (Entwerfen) einer Software ist der Teil des Softwareentwicklungsprozesses (siehe Jahrgangsband 1, Lernfeld 5), bei dem die Architektur, die Komponenten, die Schnittstellen und andere Eigenschaften wie die Datenstrukturen und Algorithmen einer Software definiert und geplant werden. Der Entwurf der Software ist eine sehr wichtige Phase, da die gesamte Anwendung auf den Entscheidungen, welche hier getroffen werden, aufbaut.



Mit dem Entwurf wird begonnen, nachdem die grundlegenden Anforderungen an die Software feststehen. Diese müssen dabei noch nicht vollständig sein, wie es vor allem bei agilen Vorgehensmodellen der Fall ist. Abhängig vom gewählten Vorgehensmodell (siehe Jahrgangsband 1, Lernfeld 5 und Jahrgangsband 2, Lernfeld 8) ist der Softwareentwurf eine in sich abgeschlossene Phase, z. B. beim Wasserfallmodell, oder wird wiederholt durchlaufen werden, z. B. bei Scrum. Dabei gibt es mehrere Entwurfsebenen, welche fließend ineinander übergehen und deren Ergebnisse während der Entwicklungszeit angepasst, verfeinert und optimiert werden.



Entwurfsebenen beim Softwaredesign

Architekturentwurf

Das architektonische Design stellt die höchste abstrakte Version des Systems dar. Es identifiziert die Software als ein System, bei dem viele Komponenten miteinander interagieren. Dabei geht es noch nicht um einen detaillierten Entwurf, sondern es werden die Zusammenhänge zwischen den Anforderungen und dem zu konstruierenden System zunächst beschrieben und begründet. In einem ersten Schritt wird der grundlegende Aufbau des Softwaresystems festgelegt. Dieser wird dann in einem zweiten Schritt in einzelne Subsysteme und Module weiter unterteilt und deren Beziehungen untereinander beschrieben. Das Ergebnis dieses Prozesses ist die sogenannte Softwarearchitektur. Die Softwarearchitektur wird neben den funktionalen Anforderungen wesentlich durch die Anforderungen an Softwarequalitätskriterien wie beispielsweise Wartbarkeit, Sicherheit oder Performance bestimmt.

Detailentwurf

Der Detailentwurf beschreibt den genauen Aufbau der Module und deren Implementierungen. Hier werden die logische Struktur jedes Moduls, seine Komponenten und seine Schnittstellen definiert. Außerdem werden die Datenstruktur und die Algorithmen der verschiedenen Module entworfen. Das Ergebnis der detaillierten Entwurfsphase wird in der Regel als Modulentwurf bezeichnet.

Nachfolgend wird die Architektur einer Software näher betrachtet. Der Detailentwurf wird genauer in Kapitel 2.1.2 erläutert.

1.2.1 Benutzerschnittstellen in den Kontext der Softwarearchitektur einordnen

S Sie sollen sich die Grundlagen über die Architektur von Software erarbeiten.

Der erste Schritt bei einem Softwareentwurf ist der Architekturentwurf. Beim Architekturentwurf geht es noch nicht um einen detaillierten Entwurf, sondern es werden die Zusammenhänge zwischen den Anforderungen und dem zu konstruierenden System beschrieben und begründet. Je nach Umfang der zu entwickelnden Software kann dabei die Architektur sehr einfach oder auch sehr komplex ausfallen.



Softwarearchitektur

Die Softwarearchitektur stellt den Bauplan eines Softwaresystems dar. Dabei werden die einzelnen Komponenten und deren Verbindungen untereinander beschrieben.

Bei kleinen Softwareprojekten wird die Architektur vom Entwickler selbst bestimmt oder er verwendet vorgefertigte Architekturmuster (siehe auch Kapitel 1.2.2), bei größeren Softwareprojekten wird diese meistens von erfahrenen Softwarearchitekten entworfen und der Entwickler muss diese im Anschluss umsetzen. Hierbei sind grundlegende Kenntnisse über verschiedenen Softwarearchitekturen von Vorteil. Dieses trifft vor allem auch dann zu, wenn eine bestehende Software erweitert werden soll. In diesem Fall ist ein grundlegendes Verständnis für den architektonischen Aufbau der Software sehr hilfreich, um sich schnell in die bestehende Implementierung hineinzudenken. Eine Softwarearchitektur sollte sorgfältig geplant werden, denn einmal in diesem Bereich getroffene Entscheidungen sind später nur mit hohem Aufwand zu verändern. Eine gute Softwarearchitektur erhöht generell die Qualität der Software in Bereichen wie Wartbarkeit, Effizienz oder Zuverlässigkeit, was am Ende z. B. zu einer hohen User Experience (siehe S. 35) sowie Kundenzufriedenheit und damit zu einer hohen Kundenbindung führt.

Für eine gute Softwarearchitektur sind verschiedene Merkmale entscheidend. Exemplarisch sind einige davon in der nachfolgenden Tabelle aufgeführt.

Wesentliche Merkmale einer guten Softwarearchitektur

Modularität

Die Aufteilung eines großen Softwareprojekts in kleinere Teile wird als Modularität bezeichnet. Die Modularität ist eine Grundvoraussetzung für ein skalierbares und wartbares Softwaredesign. Dadurch, dass das Projekt in verschiedene Komponenten unterteilt ist, ist es möglich, diese parallel zu entwickeln. Durch die Modularität wird auch das Testen vereinfacht.

Entkopplung

Die Kopplung bezieht sich auf das Ausmaß der wechselseitigen Abhängigkeit (Interdependenz) zwischen den einzelnen Softwaremodulen und wie eng diese miteinander verbunden sind. Eine niedrige Kopplung ist ein Merkmal eines guten Designs. Bei geringer Kopplung können Änderungen in jedem Modul einzeln vorgenommen werden, ohne dass die anderen Module davon beeinflusst werden.

Abstraktion

Der Prozess der Identifizierung des wesentlichen Verhaltens der Module durch das Trennen von der Implementierung und dem Entfernen irrelevanter Details wird als Abstraktion bezeichnet. Eine hohe Abstraktion trägt entscheidend zu einer geringen Kopplung bei.

Einfachheit

Das Ziel eines guten Softwaredesigns ist Einfachheit. Jede Aufgabe hat ein eigenes Modul, das eigenständig genutzt und modifiziert werden kann. Dieses hilft den Quellcode wartbar zu halten und minimiert die Anzahl der potentiellen Fehler.

Vollständigkeit

Ein gutes Softwaredesign stellt die Hinlänglichkeit und Vollständigkeit der Software in Bezug auf die festgelegten Anforderungen sicher.

Parallelität

Unter Parallelität wird die Möglichkeit der Software verstanden, mehrere Aufgaben gleichzeitig auszuführen. Im Softwaredesign wird Parallelität implementiert, indem die Software in mehrere unabhängige Ausführungseinheiten aufgeteilt und diese dann parallel ausgeführt werden. Damit wird die Effizienz der Software gesteigert.

Ein wesentlicher Bestandteil der Softwarearchitektur sind die Module. Diese werden im Zuge der Modularisierung eines Softwaresystems (siehe auch Kapitel 2.1) entwickelt. Die Beschreibung der Architektur des Softwaresystems, dessen Module und deren Beziehungen untereinander kann u. a. mithilfe von UML-Komponentendiagrammen und UML-Verteilungsdiagrammen erfolgen (siehe auch Jahrgangsbuch 2, Lernfeld 8).

**Modul**

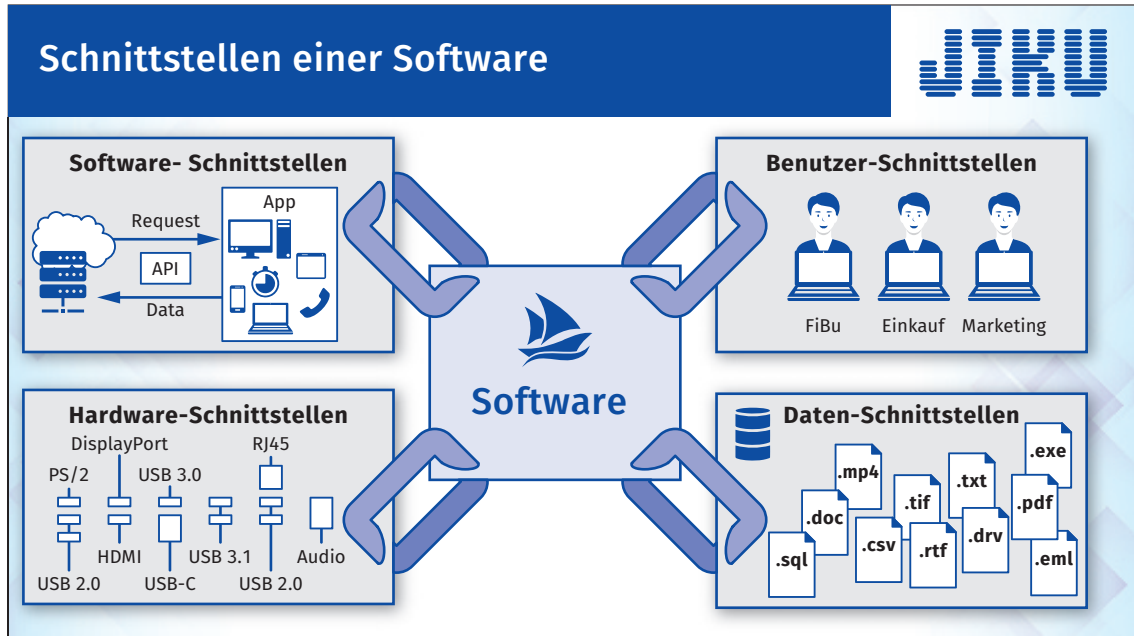
Ein Modul ist ein Baustein eines Softwaresystems, welcher eine funktional geschlossene Einheit darstellt und einen bestimmten Dienst bereitstellt, welcher über eindeutig spezifizierte Schnittstellen genutzt werden kann.

Ein weiterer Bestandteil der Softwarearchitektur ist neben der Beschreibung der Softwaremodule die Beschreibung der Schnittstellen der Software. Dabei werden interne und externe Schnittstellen unterschieden. Die internen Schnittstellen der einzelnen Module dienen zur Interaktion mit anderen Modulen der Software.

**Schnittstelle**

Eine Schnittstelle ist eine Verbindung zwischen zwei Systemen, die eine Kommunikation bzw. eine Übertragung ermöglicht.

Die externen Schnittstellen ermöglichen die Kommunikation mit der Software. Diesen kommt eine besonders große Bedeutung zu. Über sie können z. B. heterogene Systeme miteinander verbunden und so Insellösungen vermieden werden. Zudem ermöglichen sie eine unkomplizierte Interaktion der Benutzer mit der Software. Es werden Daten-, Hardware-, Software- und Benutzerschnittstellen unterschieden.



Unterscheidung von Schnittstellen einer Software

Benutzerschnittstelle

Eine Benutzerschnittstelle stellt eine Schnittstelle zwischen der Software und dem Benutzer der Software dar. Sie nimmt die Eingaben des Benutzers entgegen, leitet diese zur Verarbeitung weiter und stellt dem Benutzer danach die Ergebnisse wieder zur Verfügung. Bei den Benutzerschnittstellen werden noch verschiedene Formen unterschieden, z. B. CLI, GUI (siehe Kapitel 1.3.1).

Softwareschnittstelle

Softwareschnittstellen bilden den Übergang zwischen unterschiedlichen Programmen und ermöglichen dadurch den Datenaustausch zwischen diesen. Zu den Softwareschnittstellen werden vor allem API (Application Programming Interface) und ABI (Application Binary Interface) gezählt. Durch eine API kann eine Software in ein anderes System eingebunden werden. Damit ein Programm seine Funktionalität anderen Programmen zur Verfügung stellen kann, muss eine entsprechende Programmierschnittstelle implementiert werden.

Hardwareschnittstelle

Die Hardwarechnittstelle ist eine Schnittstelle zwischen physikalischen Systemen in der Elektrotechnik und Elektronik. Damit eine Software mit den erforderlichen Hardwareanschlüssen, z. B. USB-Anschlüssen, interagieren kann, muss eine entsprechende Funktionalität im Programm implementiert werden (siehe auch Jahrgangsband 1, Lernfeld 2 und Lernfeld 3).

Datenschnittstelle

Durch Datenschnittstellen wird festgelegt, welche Daten bzw. Datenformate die Software einlesen oder ausgeben kann. Datenschnittstellen dienen vor allem dazu, größere Datenmengen automatisiert dem Programm zur Verfügung zu stellen oder dieses aus dem Programm auszulesen.

Während früher dem Entwurf und dem Design der Benutzerschnittstelle nicht so viel Aufmerksamkeit geschenkt wurde, gewinnt dieser Bereich immer mehr an Bedeutung, sodass er heutzutage in vielen Fällen schon einen eigenen umfassenden Entwicklungsschritt im Softwareentwicklungszyklus darstellt.

Kompetenzcheck ✓

- 1 Geben Sie an, welche Aussagen richtig sind.
 - a) Die Softwarearchitektur stellt einen Bauplan für ein Softwaresystem dar.
 - b) Beim Architekturentwurf wird das Softwaresystem in verschiedene Subsysteme und Module unterteilt und deren Beziehungen untereinander beschrieben.
 - c) Für eine gute Softwarearchitektur wird eine niedrige Kopplung zwischen den Modulen angestrebt.
 - d) Eine Schnittstelle stellt eine Verbindung zwischen zwei Systemen dar.
 - e) Eine Software besteht nur externe Schnittstellen, da intern alle Module direkt aufeinander zugreifen.
 - f) Eine Benutzerschnittstelle ermöglicht die Interaktion des Benutzers mit der Software.
 - g) Durch die Datenschnittstellen einer Software wird definiert, welche Daten importiert bzw. exportiert werden können und in welchen Formaten.

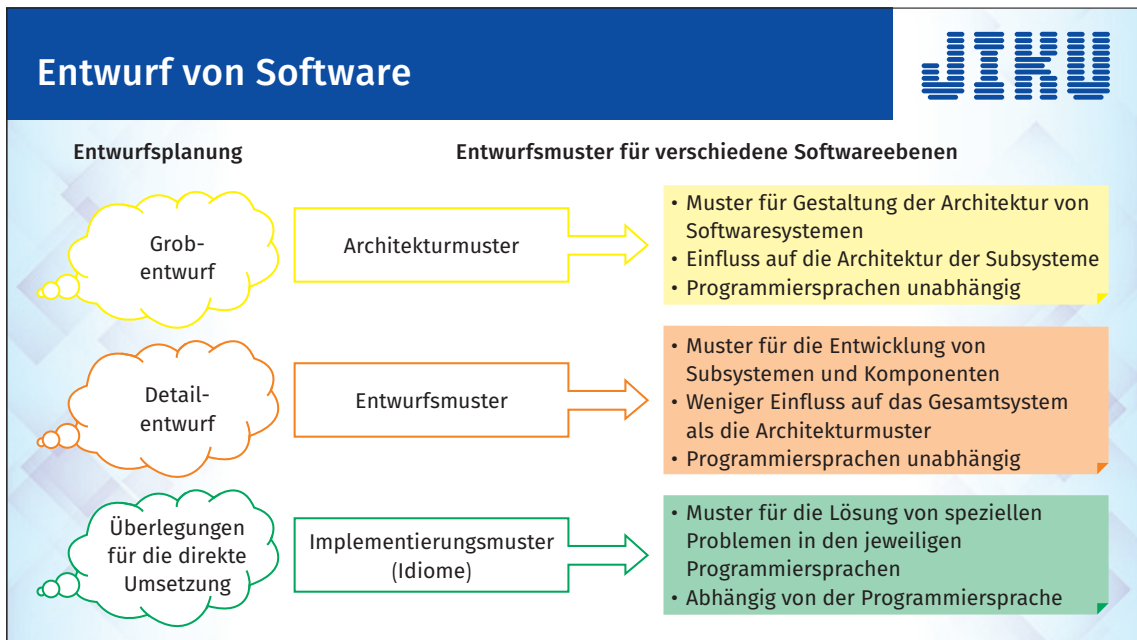
A1,2

- 2 Bearbeiten Sie im Arbeitsbuch von der Lernsituation 2 die Aufgaben 1 und 2.

1.2.2 Architekturmuster unterscheiden

- S** Sie sollen sich die Grundlagen über Entwurfsmuster erarbeiten und Architekturmuster unterscheiden und beschreiben können.

Um wiederkehrende Entwurfsprobleme nicht immer wieder neu lösen zu müssen, wurden sogenannte Entwurfsmuster entwickelt, welche in den verschiedenen Planungsstufen zum Einsatz kommen.



Zunächst kommen die Architekturmuster zum Einsatz. Im Gegensatz zu den Entwurfs- oder Implementierungsmustern wird durch diese die grundlegende Organisation und Interaktion zwischen den Komponenten einer Anwendung beschrieben. Die verschiedenen Architekturmuster lassen sich in vier große Bereiche einordnen:

Bildquellenverzeichnis

Anaconda Inc., Berlin: 233.1.
Apple Inc. Deutschland, Haar: 40.2.
Canonical Ltd. Ubuntu, London: 41.4.
Eclipse Foundation, Ottawa, Ontario: 66.1.
Gesellschaft für Informatik e.V., Bonn: 221.1.
GitHub, Inc., San Francisco: jupyter.org 228.1; keras.io 250.2; matplotlib.org 228.3; onnx.ai 250.4; pytorch.org 250.3; tensorflow.org 250.1.
Gratzke, Jürgen, Adendorf/bearbeitet von Hild, Claudia, Angelburg: 19.1, 266.2, 274.1.
Hild, Claudia, Angelburg: 8.7, 10.2, 14.1, 23.1, 24.1, 38.2, 48.2, 48.3, 62.2, 78.2, 81.2, 88.1, 98.1, 99.1, 101.2, 103.1, 105.2, 112.1, 114.1, 132.1, 144.1, 145.1, 145.3, 161.1, 217.1, 224.1, 234.1, 241.1, 242.2, 251.1, 254.8, 263.2, 266.1, 276.1.
Ingo Patett, Rostock: 29.2, 39.1, 41.5, 52.1, 53.1, 53.2, 54.1, 54.2, 54.3, 55.1, 58.1, 62.1, 62.4, 67.1, 67.2, 67.3, 68.1, 68.2, 68.3, 69.1, 74.1, 78.1, 78.4, 83.1, 84.1, 84.2, 85.1, 85.2, 85.3, 86.1, 86.2, 90.1, 90.2, 90.3, 90.4, 90.5, 90.6, 90.7, 90.8, 90.9, 90.10, 92.1, 95.1, 101.1, 170.1, 175.1, 176.1, 181.1, 181.2, 186.1, 186.2, 188.1, 188.2, 189.1, 189.2, 192.1, 195.1, 210.1, 213.1, 213.2.
Ingo Patett, Rostock/bearbeitet von Hild, Claudia, Angelburg: 10.3, 38.1, 255.1, 258.1, 258.2, 258.3, 258.4, 258.5, 258.6, 258.7, 258.8, 258.9, 259.1.
iStockphoto.com, Calgary: Dutton, Laurence 216.4.
Linux Mint: 41.3.
Microsoft Deutschland GmbH, München: 40.3, 41.1, 41.2.
NumFOCUS, Austin: 228.2, 233.2, 233.3.
Shutterstock.com, New York: bleakstar 269.1; cybrain 111.1; dizain 113.1; FGC 271.1; Lightspring 131.1.
stock.adobe.com, Dublin: 217.2; AK-DigiArt 171.1; artinspiring 203.1; artursfoto 37.1; bakhtiarzein 10.1; BalanceFormCreative 110.2; beermedia 81.3; Black Jack 105.1; Blue Planet Studio 264.1; bluedesign 275.1; canjoena 254.4; Cifotart 45.1; contrastwerkstatt 254.5; CrazyCloud 28.1; DDRockstar 283.1; Deeplab 31.1; dennizn 45.2; Destina 106.1; Diamond Hills 110.3; DragonImages 8.2; EdNurg 31.2; Elnur 37.2; eyetronic 85.4; Flametric 165.1; foto_don 29.1; fotomek 158.1; Gorodenkoff 8.4, 8.6, 216.3; Henlisatho 58.2, 62.3, 74.2, 78.3, 101.3; Herrbach, Nicolas 9.1; ileezhun 40.1, 47.1; juhumbert 114.2; knssr 242.1; Kujansuu, Tuomas 51.1, 65.1, 81.1, 146.1, 152.1; Kuklyshyn, Mykola 254.3; Le Moal, Olivier 182.1, 263.1; lienchen020_2 44.2; maglara 236.1; Mainka, Markus 145.2; mangpor2004 30.3; Margol, Grzegorz 110.4; metamorworks 30.2; Mike Mareen 241.2; MK-Photo 256.1; Monopoly919 216.5; Moreno Soppelsa 242.3; Muhammaddsy 102.1; nasir1164 192.2; NDABCREATIVITY 263.3; New Africa 28.2; nullplus 110.6; olly 267.1; orbcats 254.7; Orion Media Group 8.8, 48.4; pavlobaliukh 88.2; picoStudio 46.1, 145.4; Pimukova, Elena 207.1; Priewe, Jürgen 211.1; profit_image 97.1; r0b_ 104.1; Racle Fotodesign 109.1, 215.1, 253.1; relief 254.6; S.Schwedert Fotografie 177.1; Scanrail 110.5; sdecoret 216.2; SergeyBitos 8.3, 30.1; Silvia 158.2; Smile Studio AP 216.6; Song_about_summer 234.2; SpicyTruffel 267.2; stonepic 8.1; Tartila 48.1; tashatuvango 16.1; Trueffelpix 35.1, 254.2; ty 254.1; valerybrozhinsky 216.1; vallep 13.1, 21.1, 110.7; varflolomey 8.5; vchalup 278.1; vegefox.com 224.2; Visions-AD 144.2; Visual Generation Titel; vladimirfloyd 44.1; wachiwit 26.1; WoGi 161.2; WrightStudio 278.2; www.freund-foto.de 44.3; yurich84 110.1; z_amir 20.1, 136.1.
United Nations, New York, NY: <https://www.un.org/sustainabledevelopment/> and that you please include the following statement: "The content of this publication has not been approved by the United Nations and does not reflect the views of the United Nations or its officials or Member States" 222.1.