



**Studs@Work**

*Consulting, Development & Education*

---

# Naukanu Sailing School

## Kursverwaltung

---

---

**Studs@Work AG**  
Max-von-Laue-Straße 9  
60439 Frankfurt am Main

Amtsgericht: Frankfurt am Main HRB 12345  
St-Nr.: 12/123/12345  
USt-IdNr: DE212345678  
Vorstand: Max Mustermann

Tel +49 (0)123 12 34 5-0  
Fax +49 (0)123 12 34 5-9  
<http://www.studsatwork.de>  
[info@studsatwork.de](mailto:info@studsatwork.de)



---

## Dokumenthistorie

---

Version	Datum	Autor(en)	Kommentar / Beschreibung
0.1	07.02.2014	Benjamin Böcherer, Stefan Müller, Dominik Schumacher, Tobias Meyer	Initiale Erstellung
0.2	13.03.2014	Benjamin Böcherer, Stefan Müller, Dominik Schumacher, Tobias Meyer	Korrektur
0.3	20.03.2014	Benjamin Böcherer, Stefan Müller, Dominik Schumacher, Tobias Meyer	Finalisierung



## Inhalt

1	Prolog .....	6
2	Ausgangssituation .....	6
3	Zielsetzung.....	7
4	Mitwirkung des Auftraggebers.....	7
5	Architektur.....	8
6	Hardware- und Systemvoraussetzungen .....	9
7	Entwicklung der Software .....	10
8	Verwendete Technologien .....	10
9	Funktionskatalog .....	11
9.1	Benutzerhandbuch .....	11
9.2	Kursverwaltung.....	11
9.2.1	Verwaltung des Kurstyps.....	11
9.2.2	Terminverwaltung .....	11
9.3	Mitarbeiterverwaltung.....	11
9.4	Materialverwaltung.....	12
9.5	Kundenverwaltung .....	12
9.6	Rechnungsverwaltung .....	13
10	Zusatzfunktionen.....	14
10.1	Bedienbarkeit .....	14
10.2	Dashboard (Übersichtsseite) .....	14
10.3	Integration von E-Mail.....	14
10.4	Benutzer- und Rollenverwaltung.....	14



10.5	Mehrsprachigkeit .....	14
11	Projektstrukturplan .....	15
12	Terminplan .....	16
13	Vorgehensmodell und Qualitätssicherung.....	17
14	Organisationswerkzeuge .....	21
15	Firmenprofil.....	23
16	Mitarbeiterprofile.....	25
17	Aufwandsschätzung.....	29
18	Signaturen .....	30
19	Anhang .....	31
19.1	Projektstrukturplan .....	31
19.2	Terminplan .....	32



Die nachfolgend formulierten Bedingungen gelten als Grundlage für die Entwicklung einer Software für die

## **„Kursverwaltung“**

die in Kooperation zwischen

### **Naukanu Sailing School**

Musterstraße 15

I-Gardasee

(Auftraggeber)

und

### **Studs@Work AG**

Max-von-Laue-Straße 9

D-60439 Frankfurt am Main

(Auftragnehmer)

entwickelt werden soll.



## 1 Prolog

Die Firma Naukanu Sailing School am Gardasee ist eine Segel- und Surfschule in Norditalien. Sie wurde 1928 von Felipe Santane gegründet. Im Sommer 2001 erwarb Stefan Marx die Segelschule und übernahm die Verantwortung für diese traditionsreiche Institution, in der schon Generationen von Seglern ausgebildet wurden. Die dort angebotenen Kurse können von Gruppen und Einzelpersonen gebucht werden. Die dafür eingesetzten Kursleiter sind freie Mitarbeiter, die saisonweise beschäftigt und kursweise bezahlt werden. Zum Nachweis entsprechender Kenntnisse der Teilnehmer gibt es eine Vielzahl unterschiedlicher Segel- und Surfscheine. Wer einen Segel- bzw. Surfschein machen möchte, hat bei der Naukanu Sailing School die Möglichkeiten diese unter kompetenter Anleitung zu absolvieren. Die Teilnehmer lernen das seemännische Handwerk vom Einsteiger bis zur Prüfungsreife für den Binnenführerschein.

## 2 Ausgangssituation

Die Naukanu Sailing School benötigt eine Anwendung für die komplette Kursverwaltung. Diese Anwendung soll folgende Aufgaben bewältigen:

- Verwaltung der freien Mitarbeiter (Vertragsmanagement der freien Mitarbeiter, Bezahlung der freien Mitarbeiter, Aufnahme und Verwaltung der persönlichen und beschäftigungsrelevanten Daten, Einteilung für Kurse)
- Verwaltung der Kurse (Termine, Zuordnung zu Kunden, Gebühren, Bereitstellung des Materials, Kursleiter)
- Materialverwaltung (Einsatzbereitschaft, Aussonderung, Reparaturverwaltung, Neubeschaffung, Daten zu Material wie Merkmale, Marke, Kaufpreis, Reparaturkosten)
- Kundenverwaltung (Daten, gebuchte Kurse, Rechnungserstellung, Zahlungsverfolgung, Mahnwesen)



## 3 Zielsetzung

Mit der Einführung der neuen Kursverwaltung werden nachfolgende Grundsätze und Ziele verfolgt:

- Neuer technischer Stand
- Benutzerfreundlichkeit
- Hoher Abdeckungsgrad der Anforderungen
- Flexible Erweiterbarkeit der Software

## 4 Mitwirkung des Auftraggebers

Der Auftraggeber stellt ab der Auftragserteilung und während der Vertragslaufzeit den Zugriff auf die notwendige Hardware, Software, Schnittstellen, Datenbanken, Räumlichkeiten und Testsysteme mit Testdaten zur Verfügung, sofern dies für die Durchführung der vereinbarten Tätigkeiten notwendig ist.

Der Auftraggeber ermöglicht den Zugriff auf die notwendigen Stamm- und Bewegungsdaten bzw. stellt vergleichbare Testdaten zur Verfügung, welche eine authentische und realitätsnahe Produktionsumgebung simulieren können.

Der Auftraggeber stellt die Daten und Datenstrukturen in konsistenter Form zur Verfügung.

Der Auftraggeber benennt einen fachkundigen Mitarbeiter als Ansprechpartner, der die entsprechenden Geschäftsinformationen in angemessener Zeit beschaffen kann. Zudem benennt der Auftraggeber für den Fall, in dem der Ansprechpartner dauerhaft nicht zur Verfügung steht (Krankheit, Urlaub, Ausscheiden), eine entsprechende Vertretung.

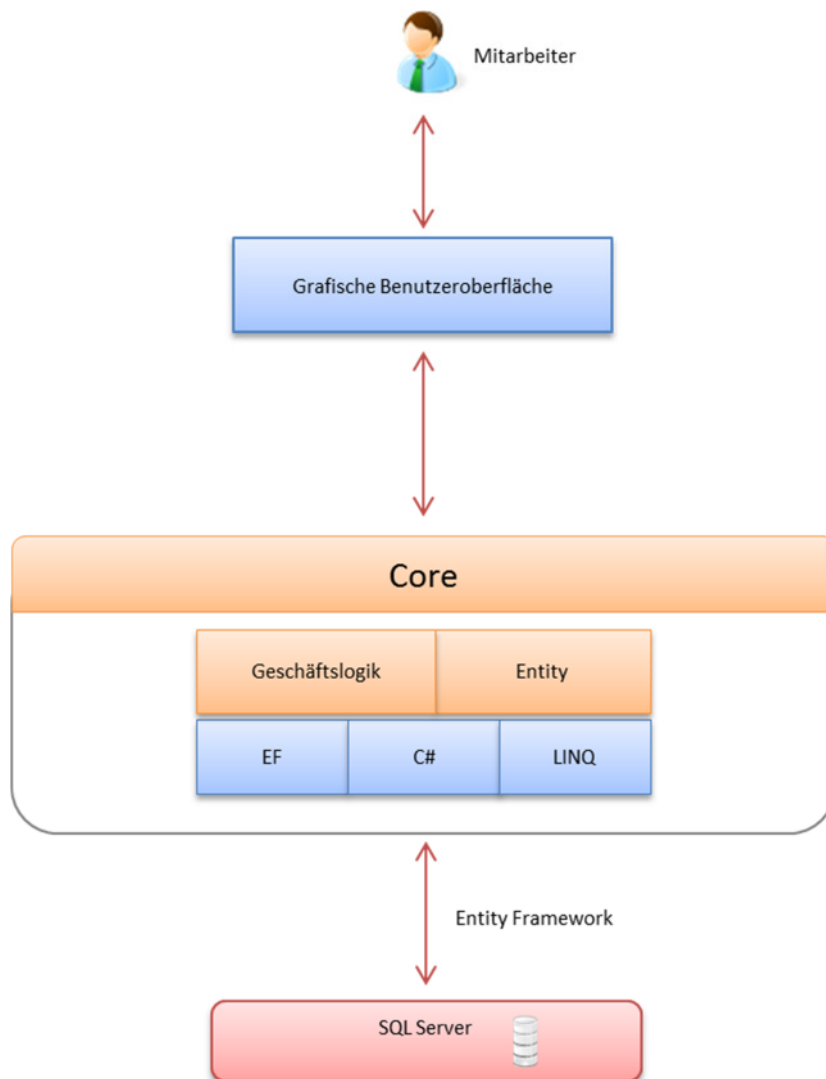
Der Auftraggeber stellt den Kontakt zu den Dienstleistern her, welche die Drittsysteme betreuen, sofern Drittsysteme bei der Realisierung des Vorhabens eingebunden werden müssen.



## 5 Architektur

Durch eine komponentenbasierte Softwareentwicklung, bei der die einzelnen fachlichen Anforderungen im Sinne der Separation of Concerns in getrennten Komponenten umgesetzt werden, die untereinander lediglich über Schnittstellen kommunizieren, wird ein modularer Aufbau der Software erreicht. Auf Basis dieses modularen Aufbaus können die einzelnen Komponenten unabhängig voneinander entwickelt und angepasst werden, was den Entwicklungsaufwand reduziert und die Wartbarkeit der Software erhöht.

Die folgende Abbildung zeigt die Grobarchitektur des zu entwickelnden Systems:







## **Grafische Oberfläche**

Die Präsentationsschicht wird durch eine WPF-Anwendung umgesetzt, die mit C# realisiert ist. In diesem Artefakt werden lediglich die grafischen Elemente (Seiten/Formulare) und die für diese Seiten notwendigen Komponenten abgebildet.

## **Geschäftslogik**

Das Geschäftslogikmodul kapselt alle UI-unabhängigen Aktionen, die zur Benutzung der Kernanwendung realisiert werden müssen. Damit ist gewährleistet, dass auch im Falle einer nachträglichen Änderung der Darstellungsschicht die Kernimplementierung nicht maßgeblich geändert werden muss.

## **Persistierung**

Das Persistenzmodul abstrahiert den lesenden und schreibenden Zugriff auf die zentrale Datenbank. Diese Abstraktion vereinfacht beispielsweise einen späteren Austausch des Persistenz-Mechanismus (Entity Framework).

## **Datenbank**

Die Studs@Work AG hat jahrelange Erfahrung mit verschiedenen Datenbankmanagementsystemen. Daher kann auf eine Open-Source Lösung wie MySQL oder kommerzielle Ansätze wie Microsoft SQL-Server aufgebaut werden. In diesem Projekt kommt ein Microsoft SQL-Server 2008 R2 zum Einsatz.

# **6 Hardware- und Systemvoraussetzungen**

Das System soll auf bereits bestehender Hardware betrieben werden. Für das Projekt kommt .Net 4.0 / 4.5 zum Einsatz. Dies setzt die Installation eines entsprechenden Frameworks voraus, das standardmäßig auf jedem aktuellen Windows-Betriebssystem zur Verfügung steht. Der Einsatz von anderen Betriebssystem (Unix, Mac OS) ist nicht vorgesehen.



## 7 Entwicklung der Software

Die Entwicklung der Applikation geschieht in den Räumlichkeiten der Studs@Work AG. Der Auftraggeber erhält lesenden Zugriff auf das GitHub Code-Repository des Projektes.

## 8 Verwendete Technologien

Zur Umsetzung des in diesem Dokument beschriebenen Software Entwicklungsvorhabens setzen wir folgende Technologien, Produkte bzw. Betriebssysteme ein

- Microsoft Windows Server 2008 R2
- Microsoft SQL Server 2008 R2 Express-Edition
- Microsoft Windows 7 Professional 64-Bit
- .NET 4.0/4.5
- C#
- XAML
- LINQ
- Modern UI
- Entity Framework 6
- MVVM



## 9 Funktionskatalog

Im Folgenden werden alle Features beschrieben, die von der Anwendung erfüllt werden müssen.

### 9.1 Benutzerhandbuch

Ein Benutzerhandbuch soll den Anwender bei der Bedienung der Anwendung unterstützen.

### 9.2 Kursverwaltung

Der wichtigste Teil der Anwendung, die Kursverwaltung, soll aus zwei Teilen bestehen:

#### 9.2.1 Verwaltung des Kurstyps

Hier wird der Name des Kurses, eine Kurzbeschreibung des Inhaltes, das benötigte Material, die Anzahl und Qualifikationen der notwendigen Kursleiter, die Dauer des Kurses in Stunden und Dauer in Tagen, die maximale Teilnehmerzahl pro Kurs und der Preis in Euro eingegeben.

#### 9.2.2 Terminverwaltung

Erstellung einer leistungsfähigen Terminverwaltung, in der die Termine eingegeben, geändert und gelöscht werden können. Eine grafische Darstellung wäre hilfreich.

Bei Eingabe oder Änderung eines Kurses sollte der (oder die) Kunde(n), Anzahl der Teilnehmer, Kursleiter und explizit das einzusetzende Material vermerkt werden können. Zu beachten ist hier:

- Ein Kurs wird einem Termin zugeordnet, wobei Terminkollisionen hinsichtlich des Personal- und des Materialeinsatzes zu berücksichtigen sind.
- Ein Kurs kann nur dann durchgeführt werden, wenn
  - o Ein Kursleiter (mit nötigen Lizenzen) vorhanden ist
  - o Material in ausreichender Anzahl vorhanden ist – und zwar inklusive aller Unterelemente, die für den sicheren Betrieb des Materials nötig sind.

### 9.3 Mitarbeiterverwaltung

Die Mitarbeiterverwaltung umfasst die Aufnahme der üblichen persönlichen Daten, zusätzlich noch die relevanten Qualifikationen (Segel- und Surfscheine, evtl. Lehrberechtigung mit ausstellender Stelle, alternativ



Referenzen), Daten eventueller früherer Einsätze bei der Naukanu Sailing School, ein „Zufriedenheitsrating“ nach Schulnoten sowie das Honorar auf Stundenbasis.

Zusätzlich sollten noch die Verfügbarkeitszeiten aufgenommen werden können.

Eine Honorarverwaltung (zu zahlende Honorare, gezahlte Honorare, etc.) soll ebenfalls implementiert werden.

## **9.4 Materialverwaltung**

Der Kursleiter sollen nach Abschluss jedes Kurses einen Materialstatus (z.B. nach Schulnoten von 1 bis 5; 1 uneingeschränkt einsatzbereit, 2=einsatzbereit, 3=eingeschränkt einsatzbereit; 4=nicht einsatzbereit, muss repariert werden, 5=Schrott) und explizit Schäden melden können.

Zudem sollen die Materialwarte eventuelle Reparaturmaßnahmen eingeben können. Die Schulungsleiter können das Material (nach Reparatur) dann wieder höherstufen oder zur Aussonderung empfehlen.

Die Materialwarte können nach erfolgter Reparatur diese im System melden. Hierbei ist ein Ticketsystem hilfreich.

Es ist vorzusehen, dass die Materialien in Funktionsgruppen zusammengefasst werden können.

## **9.5 Kundenverwaltung**

Die Kursverwaltung soll zwei Arten von Kunden vorsehen:

- Gruppen (z.B. über Reiseveranstalter gebucht)
- Einzelkunden (entweder über Vorbuchung der als Laufkundschaft)

Es müssen alle üblichen Daten eines Kunden erfasst werden. Neben den Stammdaten wie die Anschrift und Zahlungsart auch die gebuchten Kurse und Termine.

Eine automatische Erstellung der Buchungsbestätigung/Rechnung als PDF-Dokument soll implementiert werden.

Es ist außerdem eine Kundenzufriedenheitsabfrage mit Auswertung nach Kurs und Kursleiter umzusetzen.



Bei Gruppenbuchungen muss außerdem ein Hauptansprechpartner mit kompletter Anschrift hinterlegt werden. Des Weiteren wäre es sinnvoll eine Kundengruppierung in Einmal-/Regelmäßiger Kunde vornehmen zu können.

## 9.6 Rechnungsverwaltung

Der Nutzer muss aus einer Kursteilnahme Rechnungen erzeugen können. Der Nutzer hat dabei die Möglichkeit, einzelne Rechnungspositionen zu erstellen, zu bearbeiten und zu löschen.

Das System berechnet automatisch alle Rechnungsbeträge und Gebühren. Zudem ist ein Mahnwesen mit Zahlungsverfolgung umzusetzen

Rechnungen müssen jederzeit vom Nutzer storniert werden können. Das System generiert zu dieser Rechnung dann automatisch eine passende Storno-Rechnung.

Das System vergibt automatisch eine passende Rechnungsnummer für alle Rechnungen. Diese besteht aus den letzten zwei Ziffern des aktuellen Jahres (13,14, usw.) gefolgt von einer achtstelligen fortlaufenden Nummer. Die gesetzlich geforderte Einmaligkeit der Rechnungsnummern wird vom System sichergestellt.

Das System stellt sicher, dass erstellte Rechnungen, nachdem sie gedruckt wurden, nicht mehr verändert werden können und von Änderungen referenzierter Entitäten nicht länger betroffen sind.

Standardmäßig werden Rechnungen tabellarisch anhand des Leistungsdatums sortiert dargestellt.

Das System muss gewährleisten, eine Rechnung drucken oder in einer PDF-Datei speichern zu können. Dabei muss der Kunde die Möglichkeit haben, einen beliebigen Freitext auf der Rechnung hinzuzufügen. Außerdem müssen die Rechnungen der Briefnorm DIN 5008 entsprechen.



## 10 Zusatzfunktionen

Die nachfolgenden Funktionen sind optional und werden in dem Projekt nicht umgesetzt. Sie können aber zu einem späteren Zeitpunkt einfach in die Software integriert werden.

### **10.1 Bedienbarkeit**

Der Nutzer soll die Anwendung durch verschiedene Tastenkombinationen und den Einsatz der F-Tasten steuern können

### **10.2 Dashboard (Übersichtsseite)**

Nach der Anmeldung muss das System dem Nutzer eine Übersichtsseite anzeigen, in der Hinweise zu Verträgen angezeigt werden, welche bald auslaufen, sowie zu Kursen, für die noch keine Rechnungen erstellt wurden.

### **10.3 Integration von E-Mail**

Die Anwendung soll eine Anbindung an Microsoft Outlook erlauben, um z. B. per E-Mail Rechnungen oder hinterlegte PDF-Dokumente an Kunden senden zu können.

### **10.4 Benutzer- und Rollenverwaltung**

Der Nutzer kann sich mit einem Benutzer und Passwort an das System anmelden. Danach wird ihm entsprechender seiner Benutzerberechtigung eine angepasste Menüstruktur angezeigt.

### **10.5 Mehrsprachigkeit**

Die Anwendung kann durch den Einsatz von Sprachdateien in kürzester Zeit zu einer mehrsprachigen Anwendung erweitert werden.



## 11 Projektstrukturplan

Für die Umsetzung der erstellten Anwendung und deren Funktionen wurde ein Projektstrukturplan erstellt. Sie finden den Projektstrukturplan im Anhang dieses Dokumentes.

In der Projektorganisation werden die projektübergreifenden Themen behandelt. Ein sauber und klar strukturierter Prozess ist Grundlage für die nachfolgenden Entwicklungen.

Eine gut strukturierte sowie einheitlich gehaltene Oberfläche ist für die Benutzerfreundlichkeit ein sehr wichtiger Aspekt. Aus diesem Grund haben wir für dieses Thema ein separates Arbeitspaket gebildet. Ziel ist es, ein bedienerfreundliches und intuitiv bedienbares User-Interface zu erstellen.

In den restlichen Paketen werden die Anforderungen und Ergebnisse aus dieser Anforderungsanalyse umgesetzt.

Die einzelnen Arbeitspakete werden den Projektmitarbeitern zugeordnet. Die Zuordnung der Verantwortlichen je Arbeitspaket wird wie folgt vorgenommen:

Arbeitspaket	Hauptverantwortlicher
Projektorganisation	Benjamin Böcherer
Rechnungsverwaltung	Benjamin Böcherer
Kundenverwaltung	Tobias Meyer
Mitarbeiterverwaltung	Dominik Schumacher
Kursverwaltung	Benjamin Böcherer / Stefan Müller
Terminverwaltung	Dominik Schumacher / Tobias Meyer
Materialverwaltung	Stefan Müller
Grafische Oberfläche	Benjamin Böcherer / Dominik Schumacher / Tobias Meyer / Stefan Müller



In manchen Bereichen ist es sinnvoll mehrere Verantwortliche für das Paket zu haben, da viele Aktivitäten parallel ablaufen werden. Nur so sind die Anforderungen in solch einem engen Zeitplan umsetzbar. Die Hauptverantwortung bedeutet nicht, dass das gesamte Arbeitspaket von dem Projektmitarbeiter alleine abgearbeitet wird.

## 12 Terminplan

Die Übersicht über die Umsetzung der Pakete finden Sie im Anhang dieses Dokumentes.

Der Terminplan gibt auf oberster Projektstrukturplan-Ebene eine grobe Übersicht über den möglichen Projektverlauf. Viele Aktivitäten stehen in engem Zusammenhang zueinander, andere können parallel abgearbeitet werden.

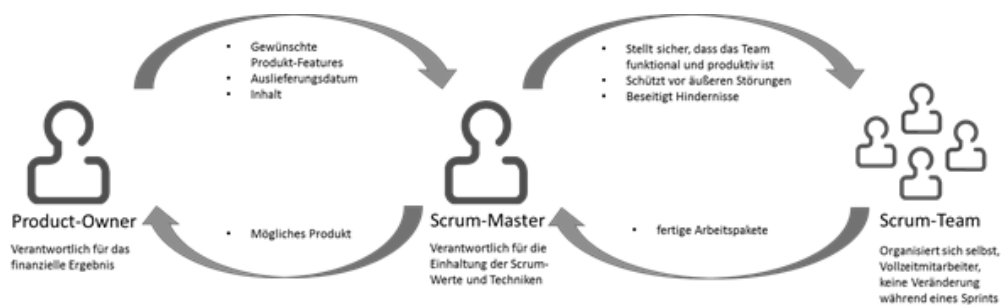




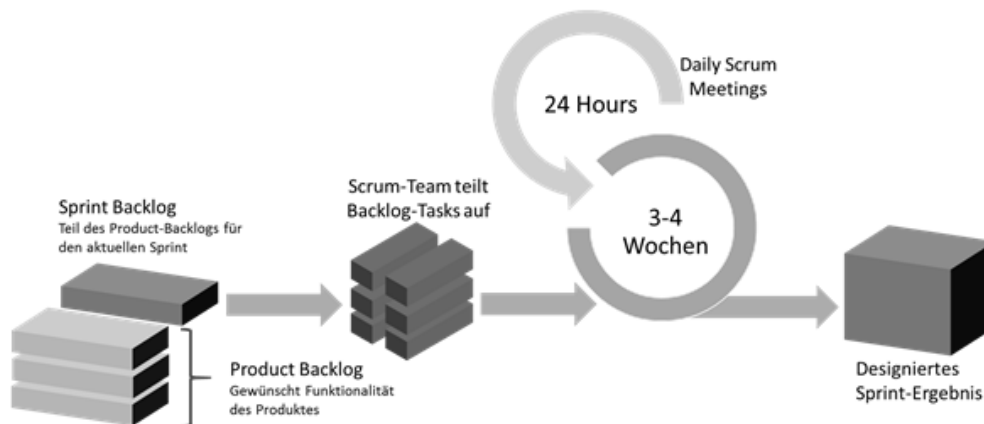
## 13 Vorgehensmodell und Qualitätssicherung

Grundlegend kommt als Vorgehensmodell SCRUM zum Einsatz, was aufgrund des iterativ-inkrementellen Ansatzes eine hohe Flexibilität bietet und wenig organisatorischen Zusatzaufwand mit sich bringt. Die Iterationen werden als Sprints bezeichnet. Jeder Sprint resultiert in einem designierten Ergebnis (meist eine neue Version der Software bzw. ein entsprechender Zwischenstand/Meilenstein), was die kontinuierliche und transparente Dokumentation des Projektfortschritts ermöglicht.

Das Vorgehensmodell wird mit einem klassischen SCRUM-Aufbau umgesetzt. Die Rolle des Product Owner wird durch einen Mitarbeiter des Auftraggebers übernommen, die des Scrum Masters durch einen Mitarbeiter der Studs@Work AG, ebenso das selbst organisierende Scrum Team.



Die einzelnen Sprints haben jeweils eine Dauer von drei bis vier Wochen. Zu Beginn eines Sprints findet gemeinsam mit dem Auftraggeber ein Sprint Planning Meeting statt, das auf Wunsch beim Auftraggeber vor Ort durchgeführt wird und in dem die Aufgaben und Ziele für den nächsten Sprint festgelegt werden. Auf diese Weise kann die Software iterativ weiterentwickelt und der Auftraggeber aktiv in den Prozess einbezogen werden. Nach Abschluss eines Sprints führt das Entwickler-Team eine Retrospektive durch, in der die Erfahrungen und Ergebnisse des Sprints reflektiert werden.



Während eines Sprints führen die Entwickler der Studs@Work AG tägliche, kurze (max. 15-minütige) Status-Meetings (sog. Daily Scrums) durch, bei dem die Ergebnisse seit dem letzten Treffen und die Planung bis zum nächsten Treffen besprochen werden.

Trotz dieses agilen Vorgehensmodells legt die Studs@Work AG Wert auf eine ausführliche, stets aktuelle und gepflegte Dokumentation von Prozessen und Quellcode.

Die technischen und projektorganisatorischen Mitarbeiter der Studs@Work AG verwenden das zuvor vorgestellte SCRUM Vorgehensmodell in Verbindung mit dem „Continuous Integration“ (CI) Ansatz, sofern vom Auftraggeber kein anderes Vorgehensmodell vorgegeben wird.

Dabei erstellt der Entwickler zunächst in seiner lokalen Arbeitsumgebung den Quellcode, der innerhalb der Entwicklungsumgebung (IDE) kompiliert und lokal installiert wird. Zuvor und sukzessive schreibt er für die fachlichen und nicht-fachlichen Anforderungen einen oder mehrere Unit Tests und führt diese aus (z. B. TestNG, JUnit, MSUnit etc.). Dies kann innerhalb der IDE oder als separater Build (Maven, Ant, MSBuild) geschehen. Das Implementieren von Komponententests dient der Qualitätssicherung und ist Bestandteil jeder Softwareentwicklung der Studs@Work AG.

Werden alle Unit Tests erfolgreich in der lokalen Umgebung ausgeführt, so „checkt“ der besagte Entwickler seine Änderungen in einem zentralen Quellcodeserver ein (Subversion, Team Foundation Server, GitHub).



Zu vorgegebener Zeit greift nun ein sog. CI-Server (z. B. Jenkins, Team Foundation Server) dieses Repository ab und führt ein „Check-Out“ durch, so dass der gesamte Quellcode aller Entwickler nun innerhalb des Servers vorliegt. Danach werden die entsprechenden Quellcodedateien kompiliert und zusammen mit eventuell vorhandenen Ressourcendateien zu Artefakten gebunden (jar, war, ear, rar, exe, dll etc.). Zudem werden nun die bestehenden Tests mittels der CI-Engine ausgeführt. Dies ermöglicht den Einsatz von sogenannten Code Coverage-Werkzeugen. Diese Werkzeuge ermitteln, wie hoch die Testabdeckung des Projekts ist. Weiterhin ist es möglich, durch den Einsatz von statischer Quellcodeanalyse, Inkonsistenzen bei der Benennung und/oder Formatierung des Codes zu identifizieren (i. d. R. Checkstyle, PMD, FindBugs).

Durch geeignete Schwellenwerte kann hier der CI-Server entscheiden, ob es sinnvoll und lohnenswert erscheint, ein neues Release zu erstellen. Typische Kennzahlen hierfür sind:

- Sind alle fachlichen Tests positiv verlaufen?
- Haben alle Negativ-Tests das gewünschte Ergebnis erzielt?
- Konnten die Performance-Tests innerhalb des designierten Zeitrahmens ausgeführt werden?
- Wurden mindestens 75 % der funktionalen Anforderungen durch Tests abgedeckt?
- Wurden alle erforderlichen Formatierungsregeln eingehalten?
- Gibt es keine offensichtlichen Fehler im Quellcode?

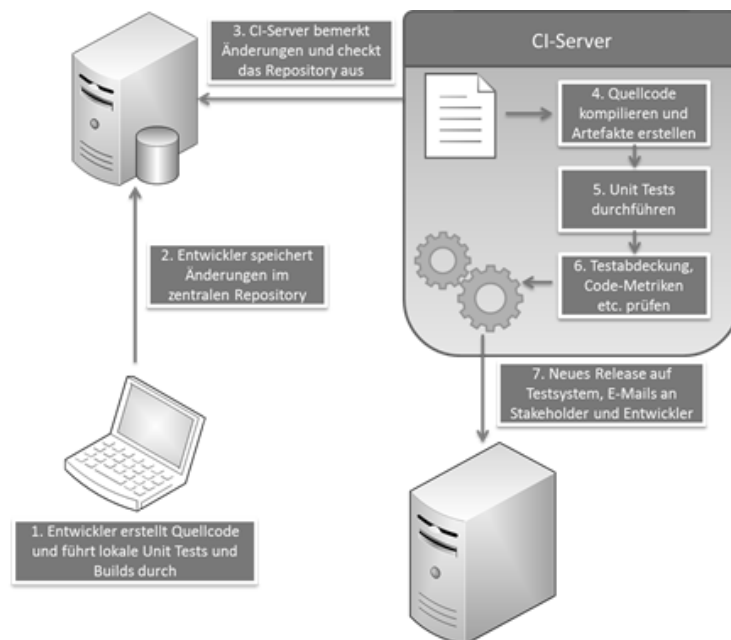
Können alle oben angegebenen Fragen positiv beantwortet werden, so wird ein Release erstellt und automatisch in der Testumgebung installiert. Somit steht nun ein neues Release (Zwischenergebnis) für „Beta-Tester“ bzw. Kunden zur Verfügung. Entwickler und Kunden haben dadurch eine sehr konkrete Vorstellung vom aktuellen Entwicklungsstand und den letzten umgesetzten Änderungen (direkte Aktions-Reaktions-Analyse). Basierend auf diesem Release kann nun ein gezielter Dialog geführt werden, wenn es darum geht, Anforderungen anzupassen oder zu erweitern.

Zusammengefasst ergeben sich aus dem Konzept „Continuous Integration“ folgende Vorteile:

- Umsetzung der Release-Often-Paradigmen der agilen Softwareentwicklung
- Zentrale Quellcodeversionierung, Möglichkeit des „Zurückspringens“ auf ältere Versionen



- Zeitnahes Testen unter „Realbedingungen“. Wenn signifikante fachliche Fehler oder Performance Probleme auftreten, kann die Ursache schnell gefunden werden, da zwischen zwei Releases wenige Änderungen stattfinden.
- Fach- und Performancetests als integraler Bestandteil der ganzheitlichen Softwareentwicklung. Keine funktionale Anforderung wird eingecheckt ohne zugehörigen Test.
- Identifikation der Testabdeckung. Wenn beispielsweise 95% des Gesamt-Quellcodes getestet ist und keine Fehler auftraten, dann kann sehr sicher davon ausgegangen werden, dass die Software macht, was sie soll.
- Identifikation von Hot Spots und Bottlenecks bei jedem Release. Welche Methoden werden besonders oft aufgerufen, welche nehmen absolut und relativ am meisten Zeit ein? Wo ist somit Optimierungspotential?
- Prüfung, ob vorgegebene Quellcodemetriken und Dokumentationsregeln eingehalten wurden.
- Identifikation von offensichtlichen Programmierfehlern.
- Automatische Benachrichtigungsfunktion beim Erstellen des Release-Artefakts (E-Mail an Fachbereich, falls neues Release vorhanden, E-Mail an Entwicklerkreis im Falle eines Fehlers).

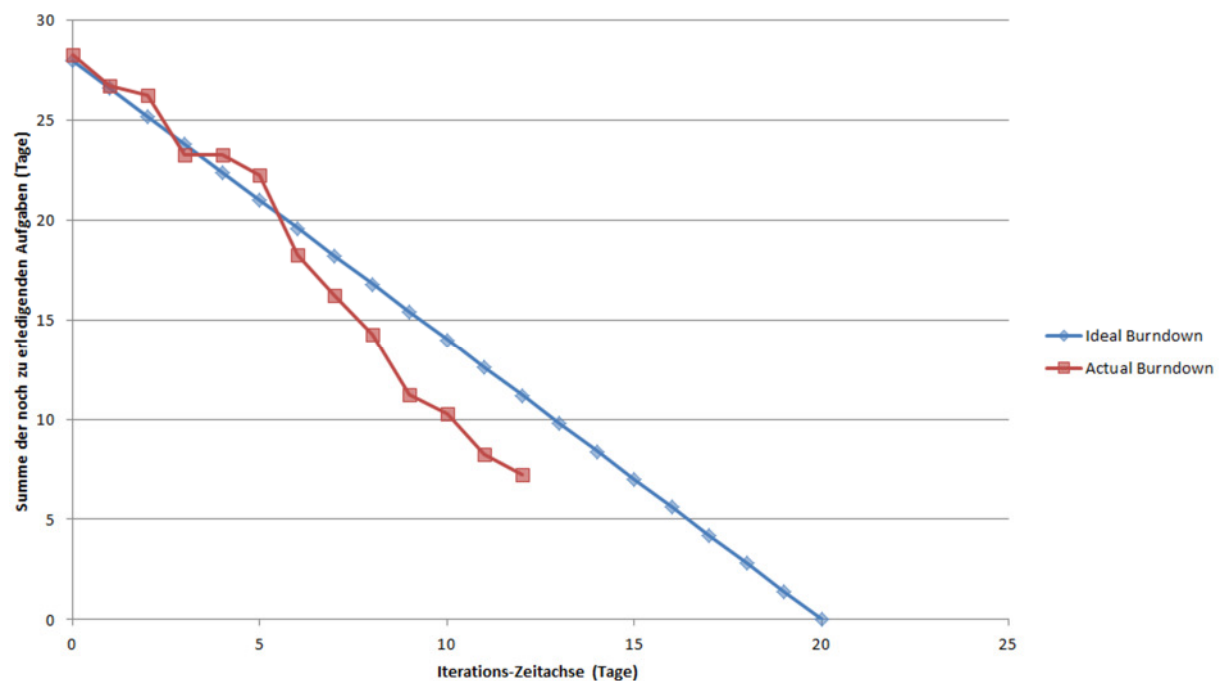




## 14 Organisationswerkzeuge

Zur Erfassung der fachlichen Vorgaben (User Stories), Aufgaben, Releases und identifizierten Bugs wird ein so genanntes Bug-Tracking-System eingesetzt (Atlassian JIRA in Verbindung mit Atlassian Greenhopper für agile Vorgehensmodelle). Somit können zu jeder Zeit Informationen über den aktuellen Projektstand ermittelt werden, die in aller Regel über folgende Kennzahlen definiert werden:

- Welche Aufgaben / User Stories / Bugs sind noch offen, in Bearbeitung, fertig und geprüft in dem aktuellen Release?
- Welche Aufwände wurden bereits geleistet und welche Aufwände stehen für das aktuelle Release noch aus?
- Können die noch ausstehenden Aufwände in der noch zur Verfügung stehenden Zeit geleistet werden?
- Gibt es eine Diskrepanz zwischen der ursprünglich geschätzten und tatsächlich benötigten Zeit?



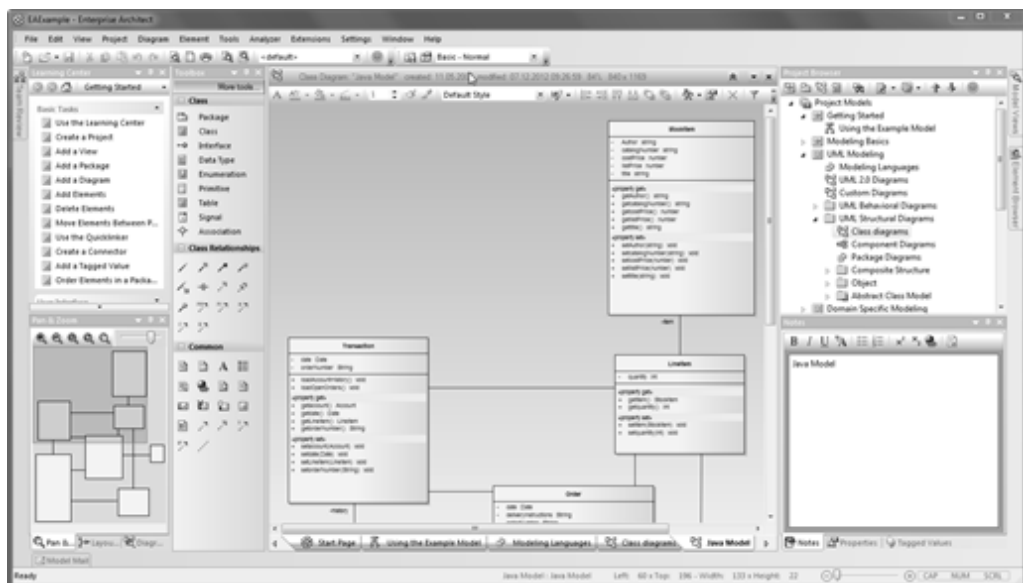
Durch die Visualisierung und Identifikation dieser Kennzahlen ist es kurzfristig möglich, einen objektiven Entwicklungsstand des Softwarevorhabens zu bekommen. Dadurch wird der Projektorganisation die



Möglichkeit gegeben, aktiv Ressourcen im Projekt zielorientiert zu steuern. Ist z. B. anhand der noch ausstehenden Tätigkeiten klar, dass in der zur Verfügung stehenden Zeit die Aufgaben nicht abgeschlossen werden können (ausgehend von einem 8 Std. Werktag), kann die Projektleitung nun aktiv Gegenmaßnahmen einleiten (Features aus dem Release herausnehmen, weitere Ressourcen kurzfristig binden, Fertigstellungstermin korrigieren etc.). Somit können zu jeder Zeit authentische Aussagen zur aktuellen Projektlage und den erwarteten Ergebnissen getätigt werden.

Die Dokumentation während der Entwicklung erfolgt in einem Wiki-System (Atlassian Confluence), wodurch die Entwickler kollaborativ an der Dokumentation arbeiten und diese kontinuierlich erweitern können.

Für die Erfassung und Planung von Prozessen kommen die Standard-Modelle der UML zum Einsatz. Als Werkzeug wird hierbei in der Regel Sparx Enterprise Architect verwendet.





## 15 Firmenprofil

<b>Firmenname</b>	Studs@Work AG
<b>Gesellschaftsform</b>	Aktiengesellschaft
<b>Gegründet</b>	1999
<b>Gründer und Geschäftsführer</b>	Max Mustermann
<b>Mitarbeiter</b>	Derzeit 49
<b>Adresse</b>	Max-von-Laue-Straße 9 60439 Frankfurt am Main

Die Studs@Work AG, mit Sitz in Frankfurt, wurde 1999 gegründet und beschäftigt 49 Vollzeitbeschäftigte (davon 46 Entwickler) und 3 Teilzeitbeschäftigte. Der Fokus der Studs@Work AG liegt auf der Individualsoftwareentwicklung im Enterprise-Bereich, dem Applikationsbetrieb (Betrieb und Support von Enterprise-Applikationen) und der IT-Beratung. Die Enterprise-Applikationen werden in Java Enterprise und Microsoft .NET entwickelt. Zum Kundenkreis gehören u.a. Automobilhersteller, Energielieferanten, Landmaschinenhersteller, Banken, Heiztechnikhersteller sowie diverse IT-Dienstleister und –Systemhäuser. Unser junges Team entwickelt Lösungen basierend auf fundierten betriebswirtschaftlichen Branchen-Know-How in Verbindung mit erstklassiger technischer Expertise.

Die Studs@Work GmbH hat es sich zur Aufgabe gemacht, als verlässlicher und kompetenter Partner seinen Kunden zur Verfügung zu stehen, denn nur so kann eine fundierte und langandauernde Partnerschaft aufgebaut werden. Dabei verwenden wir modernste Technologien und sind einer der führenden Entwickler von individual Software für Dienstleister, mittelständische Unternehmen und Konzerne in allen Branchen.

Im Jahre 2010 wurde ein weiterer wichtiger Meilenstein in der Firmengeschichte erfolgreich abgeschlossen. Nach 24 Monaten Bauzeit entstand ein moderner Bürokomplex mit Arbeitsplätzen für 100 Mitarbeiterinnen und Mitarbeiter und einem großzügigen Konferenzbereich mit hervorragender Infrastruktur für Schulungen, Kundenbesprechungen und Tagungen sowie einer Kantine.



Der Einsatz moderner Arbeitsgeräte ist für die Studs@Work AG selbstverständlich. So setzen unsere Entwickler modernste Notebooks mit Microsoft Windows 7 - 64bit ein, die mit einem Intel Core i5 und mindestens 8GB Arbeitsspeicher ausgestattet sind. Die Notebooks werden spätestens alle 2 Jahre erneuert. Zudem nutzt die Studs@Work AG VMWare-Server (vSphere ESXi) für die Bereitstellung der Entwicklungsumgebungen. Im Java-Umfeld wird Eclipse und im .NET-Umfeld Visual Studio 2012 als Entwicklungsumgebung eingesetzt. Als CI-Server wird Jenkins verwendet, als (Sub)Version Control wird TortoiseSVN genutzt und als Buildtools werden Maven sowie Ant eingesetzt. Die Datenbanksysteme von Oracle, Microsoft SQL, MySQL und PostgreSQL sind bei verschiedenen Kundenprojekten im Einsatz. Für den Einsatz von Qualitätssicherungs-Maßnahmen wird EMMA eingesetzt. Die Projektorganisation wird agil in der Regel mit dem Vorgehensmodell SCRUM oder Adaptionen daraus umgesetzt.





## 16 Mitarbeiterprofile

Projektleiter für dieses Projekt ist Tobias Meyer. Die weiteren Projektmitarbeiter sowie jeweils ein Kurzlebenslauf sind nachfolgend dargestellt.

<b>Name</b>	Dominik Schumacher
<b>Geburtsjahr</b>	1990
<b>Ausbildung</b>	Fachinformatiker Studium der Wirtschaftsinformatik
<b>Tätigkeit</b>	Consultant / Developer / Systemadministrator
<b>Schwerpunkte</b>	Realisierung von C# - Anwendungen sowie Applikationen auf Basis von PHP / Windows Powershell Realisierung von Windows Anwendungen mit dem .NET Framework 3.5 / 4.0
<b>Projekterfahrung</b>	<b>07 / 2012 – 01 / 2013</b> Implementierung eines User-SelfService – Portals für eine Zwei – Faktor SMS-Authentifizierung an Citrix Netscaler sowie einer umfangreiche Monitoring / Reporting – Schnittstelle zur Auswertung der gesammelten Daten <i>Architekt / Entwickler</i> MS-SQL-Server 2008 R2, Active – Directory – Anbindung per LDAP / GlobalCatalog, ASP.NET , Javascript, RADIUS-Authentifizierung, MS IIS 7.0, Prototyping <b>02 / 2013 - aktuell</b> Implementierung und Weiterentwicklung eines Billing-Systems für mandantenfähige ASP-Umgebungen und Datenübergabe an das zentrale ERP-System Sammeln verschiedenster Daten wie Benutzeranzahl, verbrauchte Speicherkontingente, gebuchte Applikationen / Services, Lizenzen, Rechenzeit und automatisierte Rechnungserstellung. <i>Entwickler</i> Umsetzung mit MS-SQL-Server 2008 R2, Microsoft Active Directory, Windows-Powershell, C#, WPF, Entity Framework Modell, Team Foundation Server



<b>Name</b>	Benjamin Böcherer
<b>Geburtsjahr</b>	1982
<b>Ausbildung</b>	Fachinformatiker Studium der Wirtschaftsinformatik
<b>Tätigkeit</b>	Consultant / Developer / Dozent
<b>Schwerpunkte</b>	Realisierung von Java Enterprise Edition Anwendungen mit JSF und EJB Realisierung von Windows Anwendungen mit dem .NET Framework 3.5 / 4.0 Dozent für Java und .NET-Technologien
<b>Projekterfahrung</b>	<b>08 / 2013 - heute</b>  Konzeptionierung und Entwicklung einer Software zur automatischen Bestellabwicklung, die die Verarbeitung von Daten aus unterschiedlichen Quellen (Datenbank, Textdatei) ermöglicht. Diese Bestelldaten werden in einem frei wählbaren Zeitintervall an den Mainframe gesendet. Das Tool läuft als Server-Komponente. Ein entsprechender Client zeigt den Status der Bestellungen an.  <i>Architekt und Entwickler</i>  C#, WPF, SQL Server 2008, Team Foundation Server  <b>06 / 2012 - aktuell</b>  Implementierung und Weiterentwicklung eines Portal Systems zur ganzheitlichen Verwaltung von Wertschöpfungs- und Verwaltungsprozessen, inkl. Buchhaltung, Forderungsmanagement, Mitarbeiterverwaltung, Auftragsverwaltung, Angebotsvergabe, Fuhrparkmanagement, Reise- und Hotelbuchung sowie verteilter Zeiterfassung.  Umsetzung im Rahmen eines mehrschichtigen, heterogenen Systems mit Java EE Backend Technologien sowie AJAX-basiertem Web 2.0 MS .NET Frontend sowie der ganzheitlichen Integration von 3rd Party Komponenten mittels Web Services.  <i>Entwickler</i>  ASP.NET 4.0, VB.NET, Telerik AJAX Controls, WCF, EJB 3.1, JAX-B, JAX-WS, JBoss 6, , WS-Security, WS-SecureConversation, IIS 7, SQLServer 2008, Scrum



<b>Name</b>	Tobias Meyer
<b>Geburtsjahr</b>	1986
<b>Ausbildung</b>	Wirtschaftsinformatiker (Handwerkskammer Unterfranken) Studium der Wirtschaftsinformatik
<b>Tätigkeit</b>	Consultant / Developer / Administrator / Projektleiter
<b>Schwerpunkte</b>	Realisierung von SAP Business Intelligence Projekten Realisierung von Windows Anwendungen mit dem .NET Framework 3.5 / 4.0 Dozent für SAP
<b>Projekterfahrung</b>	<b>08 / 2013 - heute</b>  Konzeptionierung und Entwicklung einer Software zur automatischen Bestellabwicklung, die die Verarbeitung von Daten aus unterschiedlichen Quellen (Datenbank, Textdatei) ermöglicht. Diese Bestelldaten werden in einem frei wählbaren Zeitintervall an den Mainframe gesendet. Das Tool läuft als Server-Komponente. Ein entsprechender Client zeigt den Status der Bestellungen an.  Architekt und Entwickler  C#, WPF, SQL Server 2008, Team Foundation Server  <b>05/2012 - 10/2013</b>  Entwicklung, Konzeption und Erweiterung eines automatischen Reportings für mehrere Niederlassungen, sowie die Erweiterung der bestehenden Business Warehouse Systeme  <i>Projektleiter, Dozent</i>  <b>05/2009 - 04/2012</b>  Realisierung der Implementierung von SAGE Office Line und Alphaplan sowie deren Weiterentwicklung.  <i>Projektleiter, Dozent</i>



<b>Name</b>	Stefan Müller
<b>Geburtsjahr</b>	1978
<b>Ausbildung</b>	Staatlich geprüfter Informatiker Fachrichtung Wirtschaft Studium der Wirtschaftsinformatik
<b>Tätigkeit</b>	Consultant / Developer / Dozent
<b>Schwerpunkte</b>	Realisierung von Java EE Anwendungen mit JSF und EJB Technologien Realisierung von Windows Anwendungen mit dem .NET Framework 3.5 / 4.0 Softwaretesting
<b>Projekterfahrung</b>	<b>11 / 2012 – aktuell</b>  Spezifikation und Durchführung von Testcases für ein webbasiertes User Interface zur zusammenhängenden Firmen- und Benutzerverwaltung mit integriertem Dienstmanagement, sowie einer darauf aufsetzenden Sim-Applet oder OTP-Authentifizierung über das Handy. Durchführung von Integration Tests, End2End Tests sowie UserAcceptanceTest in Zusammenarbeit mit dem Kunden unter Verwendung von HP Quality Center. Überarbeitung vorhandener Systemspezifikationen und deren Anpassung.  <i>Tester, stellv. Testleitung HP Quality Center</i>  <b>04 / 2012 – 11 / 2012</b>  Implementierung des Backends sowie der benötigten REST- und SOAP-Schnittstellen eines webbasierten User Interfaces zur zusammenhängenden Firmen- und Benutzerverwaltung mit integriertem Dienstmanagement, sowie einer darauf aufsetzenden Sim-Applet oder OTP-Authentifizierung über das Handy.  <i>Entwickler</i>  JBoss Application Server 7.1.1, Enterprise Java Beans (EJB) 3.1, Java Persistence API (JPA) 2, SOAP, REST, Eclipse, Maven 3.04, Oracle RAC, Solaris Cluster



## 17 Aufwandsschätzung

Aufgabenbeschreibung	Aufwand in Manntagen auf Basis von vier Mitarbeitern
Projektorganisation	10
Erstellung der Rechnungsverwaltung	40
Erstellung der Kundenverwaltung	20
Erstellung der Mitarbeiterverwaltung	20
Erstellung der Kursverwaltung	30
Erstellung der Terminverwaltung	30
Erstellung der Materialverwaltung	30
Grafische Oberfläche	40
Dokumentation/Benutzerhandbuch	20
Summe	240



## 18 Signaturen

Datum

Unterschrift

### Naukanu Sailing School

Herr Prof. Dr. Dr. Neunteufel  
Geschäftsführer/Projektleiter

---

---

### Studs@Work AG

Herr Stefan Müller  
Entwicklung

---

---

Herr Benjamin Böcherer  
Entwicklung

---

---

Herr Dominik Schumacher  
Entwicklung

---

---

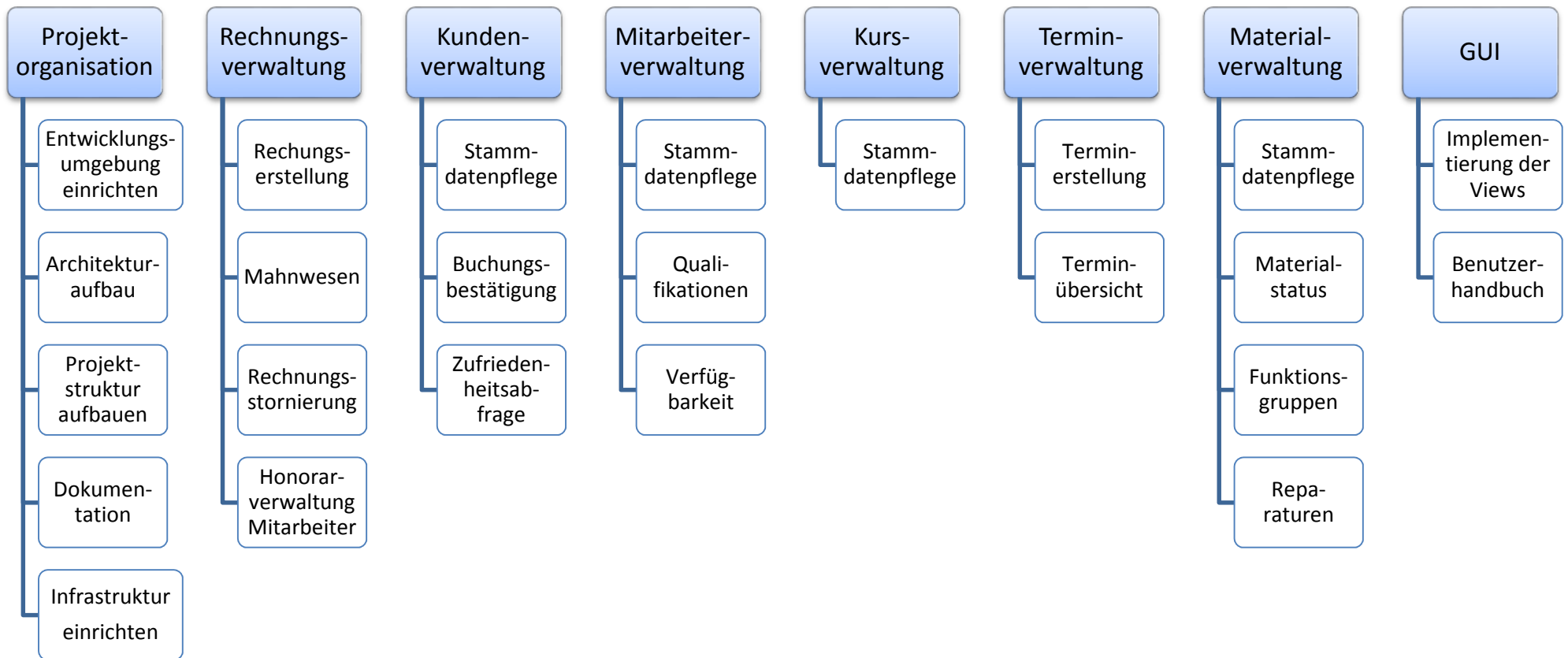
Herr Tobias Meyer  
Entwicklung/Projektleiter

---

---

## 1819 Anhang

### 18.119.1 Projektstrukturplan





## 18.219.2 Terminplan

