

eUmzug Kanton Bern

Informationen zur Modulabschlussarbeit

Modul Geschäftsprozessintegration im BSc Wirtschaftsinformatik

Autor

Björn Scheppler
scep@zhaw.ch

Beteiligte/Lektorat

Peter Heinrich
heip@zhaw.ch

Simon Näpflin
naep@zhaw.ch

Herausgeber

Zürcher Hochschule für Angewandte Wissenschaften
School of Management and Law
Institut für Wirtschaftsinformatik (IWI)
Stadthausstrasse 14
CH-8041 Winterthur
www.zhaw.ch/iwi

Vorliegende Version

1.5.0 (28.11.2018)

Änderungshistorie

Version (Datum)	Wer	Was
1.0.0 (4.9.2018)	scep	Initiale Fassung
1.1.0 (16.9.2018)	scep	Verbesserungen nach Feedback naep, Ergänzung ab Aufgabe 3
1.2.0 (3.10.2018)	scep	1. Bisherige Aufgabe 4 gestrichen => Tabelle 1 angepasst 2. Zusammenfassende Zielsetzung pro Aufgabe in Tabelle 1 eingeführt 3. Ab Aufgabe 3 Kapitel ausgelagert in eigenständige PDFs => Inhaltsübersicht und -verzeichnis kürzer, Kapitel 1.1 minimal gekürzt, Kapitel 5.1 angepasst, Kapitel 9 ff. gestrichen
1.3.0 (16.10.2018)	scep	Kapitel 5: Die Aufgaben 4 und 11 sind nicht abzugeben sowie teilweise Inhalte von Aufgabe 5 in Aufgabe 6 verschoben => Tabelle 1 angepasst (Texte durchgestrichen, Gewichtungen neu verteilt, Abgabedaten verschoben)
1.3.1 (4.11.2018)	scep	Kapitel 4.2: Berechnung Modulendnote angepasst (Rundung)
1.4.0 (10.11.2018)	scep	Kapitel 5.1: Anpassung Aufgaben und Gewichtung
1.5.0 (28.11.2018)	scep	Kapitel 5.1: Anpassung Abgabedaten

Management Summary

Von Semesterwoche 2 bis Weihnachten erarbeitet ihr in Gruppen eine **realitätsnahe Prozessapplikation** inklusive Einbindung von Umsystemen (im Folgenden **Modulabschlussarbeit** genannt): Ihr bildet den **Umzugsmeldeprozess** ab, der damit beginnt, dass eine oder mehrere Personen von einer Wohnung in eine andere umziehen wollen. Dabei müssen sich diese Personen bei der Wegzugsgemeinde abmelden und bei der Zuzugsgemeinde anmelden. Ohne eure Prozessapplikation müssen die Personen hierzu physisch bei den Gemeinden am Schalter erscheinen, mit eurer **eUmzug**-Applikation geht dies **online** 24 Stunden jeden Tag im Jahr.

Die Arbeit erstreckt sich **über das ganze Semester**, wobei sich das **Vorgehen** mehrheitlich nach folgendem **Schema** wiederholt: In einer Woche x lernt ihr bei Peter die erforderliche **Theorie** und deren Anwendung an «Hello World»-Beispielen in der Grossklasse. In derselben Woche x **übt** ihr die **Anwendung** bei Björn in der Kleinklasse anhand fortgeschrittener Beispiele und teilweise sogar anhand des eUmzug-Beispiels. Dann habt ihr bis Woche x + 2 Zeit, um eine zu dieser Theorie passende **Aufgabe** zu **lösen** und abzugeben. Bis am Ende des Semesters habt ihr somit die gesamte eUmzug-Applikation inklusive Umsystemen implementiert.

Inhaltsübersicht

Management Summary	I
Inhaltsübersicht	II
Inhaltsverzeichnis	III
1 Einleitung	1
2 Zum Projekt 'eUmzug Kanton Bern'	3
3 Anforderungen	5
4 Bewertung	7
5 Aufgabenüberblick	9
6 Aufgabe 1: Projektorganisation	12
7 Aufgabe 2: Technisch startklar	14
8 Aufgabe 3: Vom Happy Path zur bitteren Wahrheit	16

Inhaltsverzeichnis

Management Summary	I
Inhaltsübersicht	II
Inhaltsverzeichnis	III
1 Einleitung	1
1.1 Zur Aufgabenstellung	1
1.2 Lernziele	1
1.3 Deine Rolle	2
1.3.1 Fiktiv	2
1.3.2 Real	2
2 Zum Projekt 'eUmzug Kanton Bern'	3
2.1 Hintergrund eUmzug ZH und eUmzug CH	3
2.2 Die umgesetzte Lösung	3
2.3 eUmzug BE	4
3 Anforderungen	5
3.1 Originaldokumente	5
3.2 Angepasste Anforderungsspezifikation	6
4 Bewertung	7
4.1 Grundsätze	7
4.2 Berechnung Modulendnote	7
4.3 Bewertungskriterien und Gewichtung	7
4.4 Feedback	8
5 Aufgabenüberblick	9
5.1 Basis-Anforderungen	9
5.2 Fortgeschrittene Anforderungen	11
6 Aufgabe 1: Projektorganisation	12
6.1 Zielsetzung	12
6.2 Vorgeschlagenes Vorgehen	12
6.3 Abgabe auf Moodle	13
6.4 Bewertungskriterien	13
6.5 Feedback	13
7 Aufgabe 2: Technisch startklar	14
7.1 Zielsetzung	14
7.2 Vorgeschlagenes Vorgehen	14
7.3 Abgabe	15
7.4 Bewertungskriterien	15
7.5 Feedback	15
8 Aufgabe 3: Vom Happy Path zur bitteren Wahrheit	16

1 Einleitung

1.1 Zur Aufgabenstellung

Kapitel 1 bis 5 empfehle ich einmal **gründlich** zu **lesen**. Die eigentlichen Aufgaben hingegen liest Du sinnvollerweise erst, wenn Du die entsprechende Aufgabenstellung in Angriff nimmst.

Falls Du unsicher bist, ob Du etwas richtig verstanden hast, **frag Björn** in der Kleinklasse oder im Support-Forum und triff nicht ohne Rücksprache Annahmen.

Lass Dich vom «scheinbaren» **Umfang der Aufgabenstellung** nicht abschrecken. Zum einen habt ihr abgesehen von dieser Aufgabenstellung keine weiteren Aufgaben während dem Semester zu lösen, zum andern werdet ihr Schritt-für-Schritt Woche für Woche geführt.

1.2 Lernziele

Mit der Durchführung dieses realitätsnahen Entwicklungs-Projekts verfolgen wir folgende **Lernziele**:

1. Du kennst die relevanten **Elemente bei Prozessautomatisierung- und Integrationsprojekten** von der Analyse über die Konzeption bis zur Implementation.
2. Du kannst Prozessapplikationen und Microservices auf dem Niveau eines **fortgeschrittenen Anfängers** entwickeln mithilfe des Spring (Boot) Frameworks und Camunda Plattform.
3. Über die Entwicklung und Nutzung dieser Applikationen hast Du ein **vertieftes Verständnis** von **BPMN**, von der Funktionsweise einer **Process Engine** sowie von **Integrationsmustern/-architekturen und -protokollen**.
4. Durch die Realitätsnähe kannst Du abschätzen, wie gross der Unterschied bezüglich **Aufwand und Kosten** zwischen einem Hello World-Beispiel, einem Prototypen und einer produktiven Applikation ist.
5. Bei der Analyse der Anforderungsspezifikation und Konzeption der konkreten Lösung hast Du die Kenntnisse aus dem Modul **Requirements Engineering** angewendet.
6. Bei der Entwicklung hast Du die Kenntnisse aus den Modulen **Software-Engineering, Information Management** und **Web Engineering** angewendet und vertieft.
7. Bei der Durchführung dieser Modulabschlussarbeit hast Du hast als Teil einer Gruppe in der gegebenen **Zeit** (bis Weihnachten) mit dem verfügbaren «**Geld**» (4 bis 6 Personen à ca. 70 h) die **inhaltlichen** Anforderungen (Bewertungskriterien) erreicht. Damit hast Du Deine Kenntnisse aus dem Modul (IT)-**Projektmanagement** angewendet.

Zusammengefasst: Du beschäftigst Dich mit dem Thema GeschäftsprozessINTEGRATION und INTEGRIERST dabei nicht bloss Umsysteme und Menschen in einem Geschäftsprozess, sondern auch zahlreiche Kenntnisse aus Vorgänger-Modulen in einem Software-Entwicklungsprojekt.

1.3 Deine Rolle

1.3.1 Fiktiv

Stell Dir vor, **Du bist** in einem IT-Unternehmen **angestellt** mit 50 bis 100 Mitarbeitenden (z.B. Emineo AG, <http://www.emineo.ch/team>). Dein Arbeitgeber hat in einer Ausschreibung den **Auftrag** gewonnen, den **eUmzug** für den Kanton Bern (Auftraggeber) zu implementieren.

Aufgrund von aktuellen Verfügbarkeiten und Qualifikationen wurde ein **Projektteam** zusammengestellt. Mit einigen in Deinem Projektteam hast Du schon in früheren Projekten zusammengearbeitet, andere sind Dir noch fremd. Als erstes legt ihr die **Rollen** im Projektteam fest:

1. Ihr bestimmt die **Projektleitung**, welche euch durch dieses Projekt als Team führen wird und die als Ansprechperson gegenüber dem Auftraggeber auftritt.
2. Um die Projektleitung administrativ zu entlasten, legt ihr zudem eine **PMO-Fachkraft** fest, welche in euren Projektmeetings und in Meetings mit dem Auftraggeber Aktennotizen erstellt.
3. Die übrigen Personen im Team sind «normale» **Projektmitarbeitende**, aber selbstverständlich arbeiten auch die ersten zwei Rollen im Projekt mit.

Als Mitglied in diesem Projektteam und als Mitarbeiter bist Du **motiviert**, einen super Beitrag zu leisten. Einerseits, damit Du innerhalb des Unternehmens wertgeschätzt und allenfalls sogar befördert wirst. Andererseits, damit der Auftraggeber nur Gutes über Deinen Arbeitgeber zu erzählen weiss und Du somit dazu beiträgst, dass Dein Arbeitgeber auch zukünftig Arbeit zu vergeben hat.

1.3.2 Real

Du bist **Student** im Studiengang Wirtschaftsinformatik im Modul Geschäftsprozessintegration. Um das Modul zu bestehen, haben die Dozierenden (Auftraggeber) entschieden, dass Du für das Bestehen des Moduls nebst einer Modulendprüfung diese Modulabschlussarbeit erstellen musst.

Für die **Zusammenstellung Deiner Gruppe** (das Projektteam) gibt es klare [Vorgaben](#).

Als Mitglied in Deiner Gruppe, bist Du vermutlich **mehr oder weniger motiviert**. Deine Motivation hängt von Faktoren ab wie intrinsisches Interesse an diesem Modul, Deinen bisherigen Noten in anderen Modulen, Deiner aktuellen zeitlichen Verfügbarkeit, usw. Hierzu folgende Hinweise:

1. Tausch Dich bereits bei der Gruppenbildung, spätestens aber danach mit den Gruppenmitgliedern darüber aus, **wer welche Note anstrebt**. Falls es hier Zielkonflikte gibt, einigt euch zu Beginn darauf, was dies für den Einsatz jedes Einzelnen von euch bedeutet.
2. Die Modulabschlussarbeit macht **50 % der Modulendnote** aus. Zudem gibt es an der Modulendprüfung Fragen zur Anwendung der Theorie am Beispiel der Modulabschlussarbeit.
3. Falls sich im Verlauf des Semesters Hinweise auf «übermässige» **«Trittbrettfahrer»** zeigen, werden wir Massnahmen ergreifen von einer Verwarnung bis hin zu einer Einzelbenotung.
4. Falls Du zu den besonders Motivierten zählst: Wir vergeben über alle Kleinklassen hinweg ein **Zertifikat** für die beste Modulabschlussarbeit an die entsprechenden Gruppenmitglieder sowie ein Zertifikat an die Person mit der höchsten, ungerundeten Modulendnote. Je nachdem ist dies wertvoll fürs Bewerbungsdossier.

In der Realität hast Du es bei diesem Automatisierungsprojekt natürlich mit einer Vielzahl von Stakeholdern zu tun, angefangen vom Projektteam, über Deine Vorgesetzten zur Ansprechperson des Auftraggebers bis zu den Endnutzern. Im Rahmen der verfügbaren Zeit im Semester müssen wir dieses Netz an Stakeholdern im Hinblick auf die Ansprechpersonen massiv vereinfachen. Und zwar gibt es für Dein Team nur eine **Ansprechperson**: Björn verkörpert alle **Stakeholder** ausserhalb Deines Projektteams. Insbesondere vertritt er die Sicht des Auftraggebers und kennt die Bedürfnisse der Endnutzer.

2 Zum Projekt 'eUmzug Kanton Bern'

2.1 Hintergrund eUmzug ZH und eUmzug CH

Vor mehr als fünf Jahren haben der Verband schweizerischer Einwohnerdienste (VSED) und das Informatik Steuerungsorgan des Bundes (ISB) damit begonnen, die **Basis** dafür zu legen, dass schweizweit **Umzüge** innerhalb von Gemeinden als auch zwischen Gemeinden **elektronisch gemeldet** werden können, ohne dass sich die umziehenden Personen physisch am Schalter der betroffenen Gemeinden melden müssen.

Verschiedene technische, rechtliche und organisatorische **Hürden** standen diesem Vorhaben im Weg und tun es teilweise heute noch. Entsprechend langsam ging es mit dem Projekt vorwärts. Dem Kanton Zürich zu langsam. Aus diesem Grund hat der **Kanton Zürich** die Stabstelle eGovernment damit beauftragt, ein solches Vorhaben innerhalb des Kantons Zürich umzusetzen, dabei aber mit den schweizweiten Entwicklungen konform zu bleiben.

Dem Kanton Zürich ist dies gelungen: Seit Mitte 2017 sind **alle Gemeinden des Kantons Zürich an der kantonalen Umzugsplattform angeschlossen** und erste andere Kantone übernahmen dieselbe Lösung, so dass zwischen diesen Kantonen und dem Kanton Zürich sogar ein kantonsübergreifender elektronischer Umzug möglich ist.

Die vom Kanton Zürich erfolgreich eingeführte Umzugsplattform entwickelte sich wie erwähnt parallel zu den Arbeiten von VSED und ISB. Weil der Kanton Zürich in diesen Arbeiten ebenfalls involviert war, stellte der Kanton Zürich sicher, dass seine Lösung nicht dereinst eine inkompatible Insellösung sein würde. Im Gegenteil wurde die Lösung **schweizweit** positiv aufgenommen und beeinflusste das **Referenzmodell** (mehr dazu später), welches die Vorgabe für weitere Umzugsplattformen darstellt.

Inzwischen hat sich schweizweit die Organisation etwas geändert: Das **eUmzugCH**-Projekt wird inzwischen von der Schweizerischen Informatik-Konferenz (SIK) als Teil des strategischen Projekts "E-Government Schweiz" geleitet. Die vom Kanton Zürich erstellte Lösung steht als **Verbundlösung** auch anderen Kantonen zur Verfügung, betrieben durch eine neu geschaffene Organisation "**eOperations Schweiz**". Die Bevölkerung kann unter eumzug.swiss einen Umzug melden. Wer mehr über eOperations Schweiz wissen möchte, findet [hier](#) die entsprechende Info-Seite.

2.2 Die umgesetzte Lösung

Wie man sich diesen elektronischen Umzug **schematisch vereinfacht** vorstellen kann, sieht man in [diesem Video](#), welches zwar vom Kanton Zürich erstellt wurde, aber bereits auf die schweizweite Lösung Bezug nimmt, die technisch zu 100% der Kanton Zürich-Lösung entspricht.

Und in [diesem Video](#) zeigt Björn, wie ein **Endbenutzer** die **tatsächliche Anwendung** erlebt.

2.3 eUmzug BE

So weit so gut, aber **wieso** erhält Dein Team nun einen **Auftrag zu diesem Entwicklungsprojekt**, wenn ja offenbar bereits eine vom Kanton Zürich entwickelte Lösung schweizweit zur Verfügung steht? Keine Gemeinde und kein Kanton ist verpflichtet, sich der Verbundlösung anzuschliessen. Sie kann eine andere Lösung einsetzen, wenn diese mit dem schweizerischen Referenzmodell kompatibel ist.

Dies kann einerseits eine **bestehende Lösung** sein: So gab es auch schon vor der Plattform des Kantons Zürich die Möglichkeit, einen Umzug elektronisch zu melden, sofern Wegzugs- und Zuzugs-Gemeinde dieselbe Einwohnerkontroll-Lösung verwenden und die entsprechende Zusatzoption gebucht haben - bei VRSG ist dies etwa die Option [Loganto eUmzug](#). Dass sich offenbar auch weiterhin Gemeinden für solche Lösungen anstelle der Verbundlösung entscheiden, scheint Realität zu sein, so etwa die [Stadt Bern](#).

Es kann aber auch eine neue **Eigenentwicklung** sein, und davon gehen wir nun für den Kanton Bern als Auftraggeber aus: Wir nehmen völlig fiktiv (!!!) an, dass der Kanton Bern Vorbehalte gegenüber Lösungen des Kantons Zürich habe und daher eine eigene Lösung in Auftrag gebe. Das heisst: Der Betreiber der neuen Lösung und damit Dein Auftraggeber ist der Kanton Bern selbst (KAIO), nicht Deine Firma, diese ist «nur» für die Entwicklung der Lösung zuständig. Dies ist natürlich frei erfunden, denn der Kanton Bern will noch 2018 auf die Lösung des Kantons Zürichs aufspringen.

3 Anforderungen

3.1 Originaldokumente

Wie im vorhergehenden Kapitel erläutert, hat das Thema eUmzug eine lange Geschichte. Entsprechend viele **Grundlagen** sind vorhanden. Über diesen [Link](#) könnt ihr eine **ZIP-Datei** herunterladen, welche zahlreiche (wohl kaum alle) Dokumente enthält, die für die Entwicklung einer kompatiblen eUmzug-Lösung relevant sind.

Keine Angst: Ihr müsst nur einen kleinen Teil all dieser Dokumente mit Tausenden von Seiten lesen und momentan noch gar keine. Dennoch stellen wir euch alle Dokumente zur Verfügung, denn falls ihr noch nie in Berührung gekommen seid mit Anforderungsdokumenten in grösseren Software-Projekten, speziell im Umfeld der öffentlichen Hand, dürfte es ganz eindrücklich sein, darin zu schmökern. Für Interessierte daher eine Erläuterung über den **Aufbau der ZIP-Datei**:

1. **01_Einführende Dokumente:** Ein Faktenblatt und ein Leitfaden für Gemeinden
2. **02_Vorbereitende Berichte:** Als Vorbereitung auf das eigentliche eUmzug-Projekt wurden zahlreiche Teilprojekte in Auftrag gegeben, welche jeweils einen Teilaspekt abklärten (z.B. die unterschiedlichen rechtlichen Ausgangslagen in den Kantonen).
3. **03_eCH-Standards:** eCH ist ein Verein, bei welchem unter anderem Bund, Kantone und Gemeinden Mitglied sind. Er hat zum Ziel, schweizweit akzeptierte Standards im eGovernment-Bereich zu entwickeln. Wie bei anderen Normierungs-Gremien wie z.B. der ISO sind die Standards nummeriert. Folgende Unterordner sind enthalten:
 - a. **03-01_eUmzug-spezifische Standards:** Einerseits eCH-0221, welcher definiert, wie eine eUmzug-Lösung aufgebaut sein muss. Andererseits eCH-0194, welcher detaillierter auf die Meldungen zwischen der eUmzugsplattform und Umsystemen eingeht. Diese zwei Dokumente stellen für Lösungsanbieter die zentrale Anforderungsspezifikation dar.
 - b. **03-02_eUmzug-unspezifische Standards:** Ausgewählte eCH-Standards, welche für die eUmzugsplattform besonders von Interesse sind. In der Regel handelt es sich um Elemente, auf die in eCH-0194 referenziert wird.
 - c. **03-03_Schema-Definitionen:** Für die genannten Standards gibt es in der Regel auch XML Schema Definitions (XSD), welche formalisiert und maschinenlesbar abbilden, was in den Dokumenten im vorhergehenden Ordner für Menschen beschrieben ist.
 - d. **03-04_Beiispiel-Requests:** Für einige der in eCH-0194 beschriebenen XSD hat Björn einen passenden Request (XML) erstellt.
4. **04_Gebäude- und Wohnungsregister:** Beschreibung des Aufbaus des eidgenössischen Gebäude- und Wohnungsregisters, welches von der Umzugsplattform aufgerufen wird.
5. **05_Sedex-Plattform:** Fast alle Nachrichten-Übermittlungen zwischen der Umzugsplattform und den Umsystemen findet über eine vom Bundesamt für Statistik bereitgestellte Plattform statt, welche einen besonders sicheren Austausch von Informationen im Kontext der öffentlichen Hand gewährleistet. Der Einfachheit halber verzichten wir bei unserer Lösung darauf, aber wer mehr dazu wissen möchte, findet hier mehr als genug Material.

Die aufgeführten **Original-Dokumente** findet man übrigens über folgende Kanäle:

1. [eCH-Standards nach Nummer geordnet](#)
2. [eUmzug ZH](#)
3. [eUmzug CH](#)
4. [SEDEX](#)
5. [VSED](#)

3.2 Angepasste Anforderungsspezifikation

Natürlich lässt sich ein **Projekt dieser Grösse** nicht innerhalb eines Semesters entwickeln und schon gar nicht mit Studierenden, welche nur teilweise über Erfahrung in der Softwareentwicklung verfügen.

Wir begegnen dieser Herausforderung wie folgt:

1. Etliche **Teile der Lösung** haben wir als Dozierende **bereits** für euch **umgesetzt**, so dass ihr nicht bei null starten müsst.
2. Einige **Teile der Lösung** erarbeiten wir **gemeinsam in der Kleinklasse**.
3. Für die **verbleibenden** von euch selbständig zu erstellenden **Teile** gilt:
 - a. Teilweise haben wir – etwas überspitzt formuliert – schon mal die Umrissse wie in einem **Malbuch** gezeichnet und ihr müsst sie «nur» noch «ausmalen». Was heisst das konkret:
 - i. Wir lassen euch nur vereinzelt die genannten Original-Dokumente lesen, sondern haben stattdessen die **Anforderungen zusammengefasst**.
 - ii. Nicht bloss die Anforderungen, sondern auch das **Lösungsdesign** ist teilweise schon exakt **vorgegeben**.
 - b. Wir haben die Original-Anforderungen manchmal stark **vereinfacht**.
 - c. Wir unterscheiden bei den selbständig zu erstellenden Teilen **zwei Typen**:
 - i. **Basis-Anforderungen**, also solche, deren Umsetzung von allen Gruppen erwartet wird, die eine Modulabschlussarbeit-Note **bis 5.00** anstreben.
 - ii. **Fortgeschrittene Anforderungen**, aus denen Gruppen Einzelne davon zur Umsetzung auswählen können, falls sie eine Note grösser als 5.00 anstreben.

Die **eigentlichen Anforderungen** werden nicht hier an zentraler Stelle aufgeführt, sondern dort, wo sie für euch am wertvollsten sind, nämlich direkt bei den einzelnen Aufgaben und in den BPMN-Modellen.

4 Bewertung

4.1 Grundsätze

Haltet euch an folgende Grundsätze, um **mühsame Diskussionen** mit uns zu **vermeiden**:

1. Lest diese **Aufgabenstellung** gründlich und haltet euch daran.
2. Fragt nach, wenn ihr etwas **nicht versteht**.
3. Fragt auch nach, wenn ihr etwas aus guten Gründen **anders als in der Aufgabenstellung** machen wollt, bevor ihr es einfach macht.
4. Macht euch ausführlich mit den **Bewertungskriterien** vertraut und fragt nach, wenn darin etwas unklar ist. Die Bewertungskriterien findet ihr zum einen im übernächsten Unterkapitel als auch detailliert in den mit «Bewertungskriterien» gekennzeichneten Unterkapiteln ab Kapitel 6.

4.2 Berechnung Modulendnote

Die **Modulendnote** ergibt sich zu je 50 % aus den **gerundeten** Noten für die Modulabschlussarbeit und die Modulendprüfung. Als **Formel** ausgedrückt:

$$\left(\frac{\text{Note Abschlussarbeit gerundet}}{2} + \frac{\text{Note Modulendprüfung gerundet}}{2} \right) \text{gerundet auf 0.5}$$

Beispiel: Die Modulabschlussarbeit sei mit 5 benotet, die Modulendprüfung mit 4.5. Daraus ergibt sich nach obiger Formel eine gerundete Modulendnote von 5.

Jedes Gruppenmitglied erhält dieselbe Note für die Abschlussarbeit, ausser es sei eine entsprechende Massnahme bezüglich «**Trittbrettfahrer**» vereinbart worden (siehe Kapitel 1.3.2).

4.3 Bewertungskriterien und Gewichtung

Jede der in den folgenden Kapiteln vorgestellten Aufgabe hat eigene formale und inhaltliche Bewertungskriterien, die in den jeweiligen Kapiteln detailliert erläutert werden. In der Tabelle 1 auf der übernächsten Seite ist unter anderem aufgeführt, welche Aufgabe wie stark zur Note bis 5.00 (**Basis-Anforderungen**) der Abschlussarbeit beiträgt. Für die **fortgeschrittenen Anforderungen** (FA) gelten folgende Hinweise:

1. Gruppen, die «nur» Basis-Anforderungen erfüllen, dort allerdings bei allen Bewertungskriterien die maximale Punktzahl erreichen, erhalten die ungerundete **Note 5.00**.
2. Gruppen, die FA erfüllen, können **maximal einen zusätzlichen Notenpunkt** erreichen, selbst wenn sie «alle» FA erfüllen würden. Beispiel: Eine Gruppe habe bei den Basis-Anforderungen die Note 4.70 erreicht. Bei den FA hat sie das Maximum von + 1 Notenpunkt erreicht, also insgesamt die Note 5.70.
3. Wie hängen die Erfüllung von FA und der **Notenanstieg** zusammen?
 - a. FA, welche in den jeweiligen Aufgaben als «**Einfach**» gekennzeichnet sind, tragen weniger stark zum Notenanstieg bei als mit «**Komplex**» gekennzeichnete.
 - b. FA, welche **formal, fachlich und technisch korrekt implementiert** werden, führen zu einem höheren Notenanstieg als z.B. erste nicht funktionierende Versuche.

4.4 Feedback

Es ist uns wie euch wichtig, dass ihr während dem Semester **keinen Blindflug** macht und dann am Ende des Semesters die böse Überraschung folgt.

Aus diesem Grund und um den Aufwand gleichmässig zu verteilen, ist die **Modulabschlussarbeit in einzelne Aufgaben unterteilt** (siehe Tabelle 1 auf der nächsten Seite). Für jede dieser Aufgaben gibt es definierte **Lieferergebnisse**, die zu einem bestimmten Zeitpunkt abzugeben sind.

Innerhalb von ein bis zwei Wochen danach werden wir euch mindestens die **erreichte Punktzahl** pro Bewertungskriterium **mitteilen**. Manchmal wird es zudem schriftliche und/oder mündliche **Erläuterungen** geben.

Davon ausgenommen sind die **fortgeschrittenen Anforderungen**. Schriftliches und/oder mündliches Feedback geben wir auf Wunsch bereits während dem Semester, aber den erreichten Notenanstieg erfahrt ihr erst nach Abschluss des Semesters. Mehr dazu steht in Kapitel 5.2.

5 Aufgabenüberblick

5.1 Basis-Anforderungen

Tabelle 1 auf der folgenden Seite führt alle abzugebenden Aufgaben auf. Nebst der Angabe von Aufgaben-Nummer (*Nr.*) und *Bezeichnung* ist aufgeführt, in welcher Semesterwoche und an welchem Tag (*Abgabe in SW*) die Aufgabe abzugeben ist. Die Uhrzeit ist jeweils 20 Uhr. Der Spalte *Gewicht* kann entnommen werden, wie stark jede Aufgabe zur Note beiträgt (Details siehe 4.3).

Die Spalte **Typ** enthält Buchstaben, die den Charakter einer Aufgabe widerspiegeln:

I (Initialisierung): Zu Beginn (1-2) sind die Aufgaben intellektuell nicht besonders anspruchsvoll, aber sie sind erforderlich für die Durchführung der nachfolgenden Aufgaben.

A (Analyse): Dieser Buchstabe steht bei keiner Aufgabe, wäre aber mit Ausnahme der I-Aufgaben immer zutreffend, da ihr jeweils zunächst die Anforderungen analysieren müsst.

E (Entwurf): In den Aufgaben 3, 4 und 8 seid ihr vor allem konzeptionell unterwegs und erstellt teilweise ein Lösungsdesign.

R (Realisierung): Dieser Typ ist am häufigsten vertreten, damit ihr wirklich konkret etwas umsetzt, was ausgeführt werden kann.

Die **eigentliche Aufgabenstellung** ist in den folgenden Kapiteln zu finden für Aufgabe 1 und 2, respektive in eigenständigen Dokumenten ab Aufgabe 3.

Nr.	Typ	Bezeichnung und zusammenfassendes Ziel	Abgabe in SW	Gewicht*
1	I	Projektorganisation Rollen im Team sowie angestrebte Note sind geklärt	3 (03.10.)	2 / 2 %
2	I	Technisch startklar Für das kollaborative Entwickeln erforderliche Tools (Github, Netbeans und soapUI) und Kenntnisse (Versionsverwaltung) sind vorhanden	4 (10.10.)	5 / 5 %
3	ER	Vom Happy Path zur bitteren Wahrheit Mögliche Ausnahmefälle im Meldeprozess sind bekannt und deren Behandlung konzipiert sowie teilweise implementiert	6 (24.10.)	13 / 9 %
4	ER	Integrationstests- Das Funktionieren des Meldeprozess kann über sinnvolle Testfälle in soapUI automatisiert getestet werden	7 (31.10.)	0 %
5	R	Persistenz-Layer eUmzug-Plattform Für die eUmzug-Plattform sind alle Persistierungs-Komponenten vorhanden (Datenbank, Testdaten, Entities, Repositories und JavaDelegates)	8 (07.11.)	16 / 17 %
6	R	Persistenz-Layer VeKa-Dienst und GWR Für den Versichertenkarten-Dienst und das Gebäude- und Wohnungsregister sind alle Persistierungs-Komponenten vorhanden (Datenbank, Testdaten, usw.)	10 (21.11.)	16 / 17 %
7	R	User Tasks und Frontend Alle Benutzeraufgaben des Meldeprozesses sind inklusive Formulare implementiert	11 (28.11.)	20 / 20 %
8	E	Lösungsdesign Benachrichtigung und EKS-Kommunikation- Für die Implementation der Blanks-Aufgaben „xyz mitteilen“ und „Mit EK-Systemen kommunizieren“ liegt ein begründetes Lösungsdesign vor	12 (5.12.)	0 %
9	R	SOAP & REST für GWR und VeKa Die relevanten Funktionen des Gebäude- und Wohnungsregisters und des Versichertenkarten-Dienstes sind über Web Services mittels SOAP respektive REST bereitgestellt und von der eUmzugsplattform konsumiert	- (24.12.)	14 / 15 %
10	R	Einbinden Zahlungsanbieter Meldepflichtige können allfällige Gebühren im Meldeprozess über den Zahlungsanbieter Stripe begleichen	- (24.12.)	14 / 15 %
11	R	Implementation Benachrichtigung- Das Lösungsdesign aus Aufgabe 8 für „xyz benachrichtigen“ ist implementiert	14 (19.12.)	0 %
12	R	Implementation EKS und EKS-Kommunikation- Das Lösungsdesign aus Aufgabe 8 für „Mit EK-Systemen kommunizieren“ ist implementiert	-(24.12.)	0 %

Tabelle 1: Aufgabenüberblick.

* Die erste Ziffer gilt für Gruppen mit > 70 Punkten in Aufgabe 3, die zweite Ziffer für <= 70 Punkte.

5.2 Fortgeschrittene Anforderungen

Gruppen, welche keine fortgeschrittenen Anforderungen (FA) implementieren wollen, können dieses **Kapitel überspringen**.

Ab Aufgabe 4 gibt es bei jedem Kapitel ein **Unterkapitel «Fortgeschrittene Anforderungen» (FA)**. Dort ist jede Anforderung beschrieben. Falls dies nicht ausreicht, bei Björn nachfragen.

Weiter ist dort angegeben, ob die Implementation einer Anforderung «einfach» oder «komplex» eingeschätzt wird. Steht zusätzlich das Wort «vermutlich», also z.B. «vermutlich komplex», bedeutet dies, dass wir die Anforderung in unserer Musterlösung nicht implementiert haben und daher die **Komplexität** nur beschränkt einschätzen können. Für die Bewertung relevant ist, wie die Komplexität tatsächlich war und nicht, wie wir sie ursprünglich einschätzten.

Nebst unseren vorgeschlagenen FA könnt ihr **eigene FA vorschlagen**, die ihr implementieren wollt. Besprecht dies vorgängig (!) mit Björn und lasst euch schriftlich (Support-Forum Gruppen) bestätigen, dass euer Vorschlag akzeptiert wurde. Denn es kann sein, dass Björn einzelne FA ablehnt, z.B. wenn sie zu trivial oder fachlich nicht sinnvoll sind.

Ihr habt bei jeder Aufgabe **zwei Möglichkeiten**, **wann** ihr uns die Implementation der FA abgebt, eine Kombination innerhalb einer Aufgabe ist nicht möglich:

1. **Am Ende der Unterrichtszeit**, also am 24.12. mit der Abgabe der letzten Aufgabe könnt ihr uns gleichzeitig die Implementation für FA aus früheren Aufgaben abgeben. Der Vorteil hiervon ist, dass ihr noch bis am 24.12. über mehrere Wochen hinweg an der Umsetzung von FA arbeiten könnt und damit die Zeit individueller einteilen könnt.
2. **Gleichzeitig mit der Abgabe der Basis-Anforderungen-Implementation**. Der Vorteil hierbei ist, dass ihr bereits früher Feedback bekommt und nicht erst kurz vor Beginn der Prüfungssession.

Im Unterschied zu den Basis-Anforderungen ist die **Korrektur** eurer Abgaben **aufwändiger** und schwieriger. Daher gelten folgende **Regeln**:

1. Ihr müsst eindeutig **deklarieren**, ob und falls ja, welche FA ihr umgesetzt habt. Im Minimum muss dies in der README.md des Umzugsplattform-Projekts im Kapitel «Umgesetzte fortgeschrittene Anforderungen» vermerkt sein.
2. Dort muss auch angegeben sein, wo wir **weitere Informationen** finden, also z.B. ein zusätzliches Dokument/ein Screencast, die betroffenen BPMN-Modelle und darin die betroffenen Aktivitäten (allenfalls eingefärbt) und/oder die betroffenen weiteren Dateien (Java-Klassen, HTML-Forms, usw.).
3. Diese Informationen müssen so aufgebaut und formuliert sein, dass für uns unmissverständlich klar ist, **was ihr wie umgesetzt habt**. Das heisst, wir machen kein Reverse Engineering, sondern erwarten, dass ihr **umfassend, verständlich** und vor allem **nachvollziehbar** eure Implementation **dokumentiert**.
4. Für Aufgaben, welche ihr **am Ende der Unterrichtszeit** (24.12.) abgebt, geben wir euch Feedback (Erläuterungen und Notenanstieg) bis spätestens am 6.1.2019.
5. Für Aufgaben, welche ihr **gleichzeitig mit der Abgabe der Basis-Anforderungen-Implementation** abgebt, geben wir euch Feedback (nur Erläuterungen) spätestens drei Wochen danach. Der damit erreichte Notenanstieg wird spätestens am 6.1.2019 mitgeteilt.

6 Aufgabe 1: Projektorganisation

6.1 Zielsetzung

1. **Ihr wisst** in der Gruppe, **wer welche Aufgaben übernimmt**. Insbesondere die folgenden zwei Rollen sind zentral, daneben könnt ihr aber natürlich auch weitere Rollen definieren wie z.B. Lead-Entwickler, Code-Dokumentierer, Tester, usw.:
 - a. **Projektleitung**: Diese Person hält die Fäden zusammen und ist die zentrale Ansprechperson sowohl gegenüber uns als auch innerhalb des Teams. Sie führt in der Regel die Team-Besprechungen. Sie überwacht den Projektfortschritt, usw.
 - b. **PMO-Fachkraft**: Diese Person stellt insbesondere sicher, dass während den Gruppenbesprechungen eure Entscheidungen (mit und ohne uns) festgehalten werden, dass also ausreichend Protokoll geführt wird.
2. **Ihr seid euch einig**, wie ehrgeizig eure Gruppe ist und wie ihr damit umgeht, wenn nicht alle denselben **Ehrgeiz** haben.
3. **Wir wissen**, wer die Projektleitung übernimmt und damit unsere Ansprechperson ist.

6.2 Vorgeschlagenes Vorgehen

1. Sobald die Gruppen in der Kleinklasse gebildet wurden,
 - a. organisiert ihr eine **Besprechung**, an welcher **alle** (!) Gruppenmitglieder physisch (!) **anwesend** sind.
 - b. trägt sich jedes Gruppenmitglied [hier](#) bei der korrekten Gruppe ein. Hierzu folgende Hinweise:
 - i. Wir benötigen diesen Moodle-Eintrag z.B. für die Zuordnung zum Support-Forum Gruppen.
 - ii. Erst wenn sich mindestens die PMO-Fachkraft hier eingetragen hat, darf sie den Schritt 3 unten durchführen.
 - iii. Es ist technisch nicht machbar, dass eine Person von euch alle Personen der Gruppe eintragen kann, was auch als Vorteil interpretiert werden kann: Personen, welche bereits diese einfache Aufgabe nicht erledigen, fallen früh im Semester auf und können daher bereits entsprechend früh verwarnet werden.
2. An dieser Besprechung **diskutiert** ihr die in der Zielsetzung formulierten Themen und haltet sie schriftlich für euch fest. Das Ergebnis ist nur für euch intern und soll nicht an uns mitgeteilt werden, insbesondere nicht der Punkt 2 betreffend Ehrgeiz.
3. Nun **erfasst** die als PMO-Fachkraft bestimmte Person **die für uns relevanten Angaben** in der Moodle-Aufgabe gemäss dem folgenden Kapitel.

6.3 Abgabe auf Moodle

1. Abgabe 1:

- a. Jede Person trägt sich [hier](#) in der richtigen Gruppe ein.
- b. Dies erfolgt spätestens bis zum im Betreff der verlinkten Aufgabe aufgeführten Zeit.

2. Abgabe 2:

- a. Die Abgabe erfolgt [hier](#) durch EINE Person im Team (vermutlich die PMO-Fachkraft).
- b. Dies erfolgt spätestens bis zum im Betreff der verlinkten Aufgabe aufgeführten Zeit.
- c. Die Abgabe besteht darin, einen Text im folgenden Format einzufügen und auf «Abgabe einreichen» zu klicken.
 - i. GRUPPENNUMMER;PROJEKTLEITUNG_VORNAME;PROJEKTLEITUNG_NACHNAME;PROJEKTLEITUNG_EMAIL;PROJEKTLEITUNG_MOBILFUNKNUMMER;PMO_FACHKRAFT_VORNAME;PMO_FACHKRAFT_NACHNAME
 - ii. Konkretes Beispiel: VZb 08;Peter;Heinrich;heip@zhaw.ch;076 000 0000;Björn;Scheppeler
 - iii. Wichtig: Wir importieren eure Angaben in eine Tabelle mit den Angaben aller Projektteams (Gruppen). Um uns dies frustfrei zu ermöglichen, muss das genannte Format strikt eingehalten werden. Ansonsten gibt es Abzüge (siehe nächstes Kapitel).

6.4 Bewertungskriterien

Es gibt lediglich ein einziges formales Bewertungskriterium mit folgender Punkte-Vergabe:

- 0 Punkte: Keine Abgabe innerhalb der Abgabefrist für Abgabe 2
- 1 Punkt: Angaben unvollständig (z.B. nur Projektleitung, aber keine PMO-Fachkraft) und Abgabeformat nicht eingehalten (z.B. , statt ; oder Vorname Nachname nicht wie in Moodle erfasst) und/oder mehr als eine Person haben sich noch nicht eingetragen in Abgabe 1*
- 2 Punkte: Angaben vollständig, aber Abgabeformat nicht eingehalten und/oder eine Person hat sich noch nicht eingetragen in Abgabe 1*
- 3 Punkte: Angaben vollständig und Abgabeformat eingehalten sowie alle Personen haben sich eingetragen bei Abgabe 1

* Wenn diejenigen Gruppenmitglieder, welche sich eingetragen haben, nachweisen können, dass sie mindestens 48 h vor Abgabefrist die restlichen Gruppenmitglieder schriftlich zur Abgabe aufgefordert haben (z.B. per Mail oder WhatsApp), dann wird kein Abzug für die Gruppe erteilt.

6.5 Feedback

Spätestens bis 6.10.2018 19:00 sieht diejenige Person, welche Abgabe 2 eingereicht hat, die erreichte Punktzahl und allfällige Feedback-Kommentare, wenn sie erneut auf die Moodle-Aufgabe klickt.

7 Aufgabe 2: Technisch startklar

7.1 Zielsetzung

Alle Gruppenmitglieder haben technisch und bezüglich Kenntnisse zu Versionsverwaltung die Voraussetzungen erfüllt, dass sie an der Modulabschlussarbeit mitentwickeln können. Konkret für jedes Gruppenmitglied:

1. JDK, Netbeans und soapUI sind installiert.
2. Ein Github-Konto ist angelegt und man ist Mitglied in der richtigen Github Classroom-Gruppe.
3. Man kann das Gruppen-Repository auschecken (clone), in dieses Änderungen hochladen (commit & push) und aus diesem die neusten Änderungen herunterladen (pull). Auch ist man in der Lage, allfällige Konflikte zu beheben.

7.2 Vorgeschlagenes Vorgehen

1. JDK und Netbeans sollten alle bereits installiert haben. Ein Github-Konto sollte man ebenfalls schon angelegt haben. Falls nicht, [dieser Anleitung](#) folgen.
2. Nun die [Anleitung zum Thema Versionsverwaltung](#) schauen. Allenfalls die folgenden Schritte parallel dazu machen oder zunächst alles schauen und dann erst umsetzen.
3. Nun am besten alle gemeinsam physisch oder virtuell (Skype & Co.) im gleichen Raum sein, respektive den Bildschirm teilen, damit alle gleich sehen, was welcher Befehl zur Folge hat. Hierbei folgende Schritte durchführen:
 - a. Eine Person erstellt die Github Classroom-Gruppe. Nennt die Gruppe exakt wie folgt: Klasse+Nummer, also z.B. **VZa01** und nicht z.B. vza01, vza-01, VZa 01, usw.
 - b. Alle übrigen Personen treten dieser Github Classroom-Gruppe bei.
 - c. Jeder cloned initial die Github Classroom-Repo aus.
 - d. Alle erstellen eine neue leere Datei (New File > Other > Empty File) STUDENTENKUERZEL.txt (Kürzel ersetzen) in src\main\ressources (Unter «Other Sources»).
 - e. Jeder committed das Projekt mit dieser neuen Datei und dem Kommentar «STUDENTENKUERZEL.txt hinzugefügt» (Kürzel ersetzen).
 - f. Jeder pushed den Commit zur Remote Repository. Bei der ersten Person geht dies problemlos, die anderen müssen zunächst einen Pull (+ Merge) durchführen.
 - g. Wenn dies alle gemacht haben, machen alle (bis auf die letzte Person) einen Pull und sollten damit von allen Gruppenmitgliedern ein STUDENTENKUERZEL.txt haben.
 - h. Eine Person soll nun alle diese txt-Dateien wieder löschen und diese Änderung commiten und pushen mit dem Kommentar «Alle Text-Dateien pro Student gelöscht».
 - i. Alle übrigen Personen können nun wieder einen Pull machen, so dass auch bei ihnen die Dateien gelöscht sind.
 - j. Als nächstes fügt jede Person in der README.md (im Wurzelverzeichnis in der Files-Ansicht sichtbar) ihren Namen in der dritten Zeile nach «Björn Scheppler» hinzu, also z.B. «Björn Scheppler, Hans Meier» und ändert das Datum in der fünften Zeile auf das aktuelle Datum.
 - k. Eine Person committed und pushed diese Änderung mit dem Kommentar «README erweitert mit STUDENTENVORNAMEN» (Vorname ersetzen).

- I. Anschliessend versuchen dies auch die anderen, müssen aber einerseits zunächst einen Pull machen, dann einen Merge, dann manuell den Konflikt lösen und diese Lösung commiten und pushen. Hinweis: Macht dies am besten nicht gleich alle parallel, sondern einer nach dem anderen, um es nicht noch komplizierter zu machen.
4. Falls ihr es für erforderlich hält, spielt noch weitere Dinge durch, um etwas zu üben.
5. Eine Person (in der Regel die PMO-Fachkraft) gibt die Aufgabe formell ab, wie im nächsten Kapitel beschrieben.

7.3 Abgabe

Die Abgabe besteht aus verschiedenen Dingen:

1. Das vorgeschlagene Vorgehen durchführen und dadurch mindestens folgende Commits haben (allfällige weitere Commits werden von uns ignoriert):
 - a. Pro Gruppenmitglied ein «STUDENTENKUERZEL.txt hinzugefügt»-Commit, in welcher in src\main\ressources eine entsprechende Datei hinzugefügt ist.
 - b. Ein «Alle Text-Dateien pro Student gelöscht»-Commit, bei welchem die Dateien wieder gelöscht sind.
 - c. Pro Gruppenmitglied ein «README erweitert mit STUDENTENVORNAMEN»-Commit sowie zugehörige Merge-Commits, in welchem in der Datei README.md der eigene Name hinzugefügt ist zu den bestehenden Namen und das aktuelle Datum gesetzt ist.
2. Auf Moodle termingerecht (siehe Terminangabe in [Aufgabe 2](#)) nur den Link zu eurem Repository (z.B. «<https://github.com/zhaw-gpi/modulabschlussarbeit-vza01>») reinschreiben, damit wir formell wissen, dass ihr aus eurer Sicht die Aufgabe durchgeführt habt und wir euch eine Bewertung zukommen lassen können.

7.4 Bewertungskriterien

Es gibt lediglich ein einziges formales Bewertungskriterium mit folgender Punkte-Vergabe:

- 0 Punkte: Keine Abgabe innerhalb der Abgabefrist
- 1 Punkt: Auf Moodle abgegeben, aber auf Github keine Commits oder nur von weniger als der Hälfte der Gruppenmitglieder*
- 2 Punkte: Auf Moodle abgegeben und auf Github sind Commits von mindestens der Hälfte der Personen*, aber entweder sind nicht alle erforderlichen Commits vorhanden, diese teilweise nicht korrekt und/oder auf Moodle ist der Link zum Repository nicht korrekt und/oder der Name der Github Classroom-Gruppe ist nicht korrekt.
- 3 Punkte: Alles korrekt.

* Wenn diejenigen Gruppenmitglieder, welche sich beteiligt haben, nachweisen können, dass sie mindestens 48 h vor Abgabefrist die restlichen Gruppenmitglieder schriftlich zum Mitmachen aufgefordert haben (z.B. per Mail oder WhatsApp), dann wird kein Abzug für die Gruppe erteilt.

7.5 Feedback

Spätestens bis 13.10.2018 19:00 sieht diejenige Person, welche die Aufgabe eingereicht hat, die erreichte Punktzahl und allfällige Feedback-Kommentare, wenn sie erneut auf die Moodle-Aufgabe klickt.

8 Aufgabe 3: Vom Happy Path zur bitteren Wahrheit

Ab Aufgabe 3 sind die Aufgabenstellungen pro Aufgabe als eigenes PDF verfügbar. Diese befinden sich in Moodle jeweils direkt oberhalb der gleich benannten Moodle-Aufgabe.