

MODUL 326 LB-1

1 AUSGANGSLAGE

Modul 326 prüft die Fähigkeiten im Bereich der Analyse, des Entwurf, der Implementierung und des Testens einer Anwendung.

2 PROJEKTARBEIT

Die Leistungsbeurteilung erfolgt anhand einer Projektarbeit.

2.1 Zu überprüfende Handlungsziele

- 1 Aufgrund der Businessanalyse Anwendungsfälle formulieren und daraus die fachlichen Klassen ableiten.
- 2 Ein objektorientiertes Design erstellen und mittels Unified Modeling Language darstellen.
- 3 Objektorientiertes Design implementieren.
- 4 Applikations-Architektur nachvollziehbar dokumentieren.
- 5 Erfüllung der fachlichen Anforderung prüfen.

2.2 Bewertungskriterien (Projektarbeit)

5%: Administratives (Einhalten der Prüfungsvorgaben, ...).

10%: Erstellen von Anwendungsfällen aus dem gegebenen Geschäftsfall bzw. der Businessanalyse.

20%: Aus dem gegebenen Fall die korrekten fachlichen Klassen und deren Beziehungen ableiten.

20%: OOD mit technischen Klassen in einem formal und inhaltlich korrekten Klassendiagramm darstellen. Korrekte Anwendung der Architekturmuster.

20%: Saubere Implementation des Designs.

5%: Geforderte Funktion wird erreicht.

10%: Erstellen korrekter Testfälle und durchführen der Tests, dokumentiert mittels Testprotokoll.

10%: Dokumentation der Architektur durch weitere UML-Diagramme, Analyse des erstellten Designs bezüglich Qualität und/oder beschreiben der verwendeten Architekturmuster.

Die Kriterien werden im Anhang präzisiert und durch Gütestufen hinterlegt.

2.3 Dauer

4-5 Halbtage Unterricht (4 Lektionen), zusätzlicher Aufwand in Form von Hausaufgaben ist nötig

2.4 Arbeitsform

Gruppenarbeit: 3 Lernenden bilden ein Entwicklungsteam

2.5 Produkte des Projektes

1. Dokumentation
2. Programmcode
3. Testprotokolle

3 PRODUKTE

3.1 Projektdokumentation

Die Dokumentation umfasst folgende Kapitel

- Einleitung und Übersicht zum Projekt, Projektauftrag.
- Organisation der Arbeit (Team, Zuständigkeiten, Zeitplan usw.).
- Analyse des Projektauftrags mit entsprechenden UML-Diagrammen.
- Design der Applikation mit entsprechenden UML-Diagrammen.
- Vereinbarung zum Umfang der Implementation.
- Wichtige Hinweise zum Programmcode.
- Testfälle auf Grund des Projektauftrags sowie die Testprotokolle.
- Bestätigung der Eigenleistung aller Teammitglieder.

Es wird erwartet, dass die Dokumentation in gutem Deutsch (Formulierungen, Lesbarkeit, Rechtschreibung) abgehalten ist und einer üblichen Formatierung folgt.

3.2 Programmcode

3.2.1 Umfang der Arbeit

Der Programmcode muss je nach Projektumfang nur für einen Teil des Designs umgesetzt werden. Der Umfang ist mit der betreuenden Lehrperson abzuklären.

3.2.2 Paketierung

Die Programmteile sind zwingend zu paketieren. Dabei müssen die einzelnen Pakete eine logische Einheit von Funktionalitäten und/oder Daten bilden.

3.2.3 Dokumentation

Alle Klassen müssen vollständig dokumentiert sein, so dass sich die API zur Applikation erstellen lässt (javadoc)

3.2.4 Lesbarkeit

Der Code ist nach den Gepflogenheiten gut lesbaren Codes zu erstellen. Die Algorithmen sind einsichtig und so einfach wie möglich realisiert.

Es gilt die Prämisse „*make it as simple as possible but not simpler*“

3.3 Testprotokoll

Die Testfälle müssen vor der Implementation des Codes erstellt und der betreuenden Lehrperson vorgelegt werden.

Die Testprotokolle folgen der im Unterricht Modul 226 erlernten Struktur.

Die Tests sind zwingend mit JUnit durchzuführen.

3.4 Persönliche Reflexion

Jede / jeder Lernende führt ein Tagebuch.

Im Tagebuch sind folgende Punkte pro Schultag aufgearbeitet

- Tagesziel und Zielerreichung (Soll-Ist-Vergleich)
- Kurzer Abriss zur Tätigkeit
- Erkenntnisse aus der Tätigkeit (neue Inhalte, Probleme usw.)
- Ein Fazit zum Tag mit zwei Aussagen
 - a) das hat mich gefreut
 - b) das hat mich geärgert

Das Tagesziel ist jeweils schriftlich festzulegen und zu Beginn des Unterrichts der Lehrperson abzugeben. (Mustervorgabe im Anhang)

3.5 Teamverhalten

Software-Entwicklung ist eine Teamarbeit. Die entsprechenden Kompetenzen sind massgebend, wie erfolgreich und zielgerichtet ein Produkt erstellt wird.

3.5.1 Kriterien für Teamverhalten

- Zuverlässigkeit
- Kommunikation
- Integration
- Fairness
- Arbeitsmenge und Qualität

Die Gütestufen finden sich im Anhang Kapitel 5.2.3.

Jedes Teammitglied füllt den Bewertungsbogen für die andern Teammitglieder aus und übergibt diesen am Ende des Projektes der Lehrperson.

4 LEISTUNGSBEURTEILUNG

Die Projektarbeit liefert 3 Teilnoten.

- | | |
|--|-----|
| a) Bewertung des Projektes mit den Produkten des Projektes
(Dokumentation, Testfälle, Programmcode) | 80% |
| b) Bewertung der persönliche Reflexion | 10% |
| c) Bewertung des Teamverhaltens | 10% |

Die Bewertung wird in einer Excel-Liste eingetragen und die Einzel- sowie Gesamtnote ermittelt.

5 ANHANG

5.1 Projektbeschrieb

Die Projektbeschreibung wird als Zusatzdokument abgegeben.

5.2 Bewertungskriterien

Die Bewertungskriterien werden entsprechend der Gütestufen mit 1 (Minimum) bis 3 (Maximum) gewichtet. **Fehlt ein Kriterium, wird dies mit 0 gewertet.**

Zu jedem Prüfungskriterium werden konkrete, messbare Kriterien gelistet.

5.2.1 Projektarbeit

Teammitglieder _____

5.2.1.1 Administratives (Einhalten der Prüfungsvorgaben, ...).

5% der Gesamtnote

Ein Terminplan , der die wichtigsten Phasen des Projektes aufzeigt, liegt vor .		
<i>Rudimentäre Planung ohne Ziele und ohne Meilensteine.</i>	<i>Rudimentäre Planung mit grob genannten Terminen oder Meilensteine. Ziele sind wenig oder nicht umschrieben</i>	<i>Vollständiger Plan mit Datum und Meilenstein. Ziele der Meilensteine sind SMART</i>
Die von der Lehrperson vorgegebenen Zeiten werden eingehalten .		
<i>Die Produkte werden verspätet eingereicht.</i>	<i>Es werden nur ein oder zwei Produkte fristgerecht eingereicht</i>	<i>Alle Produkte werden fristgerecht eingereicht.</i>

5.2.1.2 Erstellen von Anwendungsfällen aus dem gegebenen Geschäftsfall bzw. der Businessanalyse.

10% der Gesamtnote

Der Projektauftrag wird sinngemäss in ein UseCase-Diagramm übersetzt.		
<i>Das Diagramm genügt weder den formalen Anforderungen noch gibt es den Projektauftrag sinngemäss wieder.</i>	<i>Das Diagramm genügt entweder den formalen Anforderung ODER gibt den Projektauftrag sinngemäss wieder.</i>	<i>Das Diagramm genügt den formalen Anforderungen und gibt den Projektauftrag sinngemäss wieder.</i>
Jeder Anwendungsfall wird in einer eigenen UseCase-Beschreibung präzisiert.		
<i>Es fehlen wichtige Merkmale der Beschreibung wie auch ein sinnvoller Ablauf.</i>	<i>Es fehlen wichtige Merkmale der Beschreibung. Der beschriebene Ablauf ist sinnvoll.</i>	<i>Die Beschreibung ist vollständig und gibt einen sinnvollen Ablauf wieder.</i>

5.2.1.3 Aus dem gegebenen Fall die korrekten fachlichen Klassen und deren Beziehungen ableiten.

20% der Gesamtnote

Die fachlichen Klassen liegen vor und sind mit einer einfachen Beschreibung (Attribute, Methoden) ergänzt.		
<i>Es fehlen wichtige Klassen ODER Bei den dargestellten Klassen fehlen Attribute und Methoden.</i>	<i>Das Diagramm weist Mängel in der Beschreibung der dargestellten Klassen auf.</i>	<i>Das Diagramm ist bezüglich der dargestellten Klassen vollständig, d.h. inkl. wichtiger Attribute und Methoden.</i>
Die Beziehungen der Klassen sind vollständig dokumentiert .		
<i>Die Beziehungen der Klassen sind unklar dokumentiert.</i>	<i>Die Beziehungen der Klassen sind teilweise dokumentiert.</i>	<i>Die Beziehungen der Klassen sind vollständig dokumentiert (inkl. Kardinalitäten)</i>
Der Entwurf weist eine passende Paketstruktur auf .		
<i>Der Entwurf basiert auf einem einzigen Paket (nicht default Paket).</i>	<i>Der Entwurf weist eine (eher zufällige) Paketstruktur auf.</i>	<i>Der Entwurf weist eine begründete Paketstruktur auf.</i>

5.2.1.4 OOD mit technischen Klassen in einem formal und inhaltlich korrekten Klassendiagramm darstellen. Korrekte Anwendung der Architekturmuster.

20% der Gesamtnote

Der Entwurf ist mit den benötigten technischen Klassen und Schnittstellen ergänzt.		
<i>Die technischen Klassen sind erwähnt aber im Entwurf nicht dokumentiert.</i>	<i>Die technischen Klassen sind im Entwurf teilweise dokumentiert.</i>	<i>Die technischen Klassen sind vollständig im Entwurf dokumentiert, insbesondere mit deren Beziehungen.</i>
Die für die Aufgabe typischen Entwurfsmuster sind begründet und dokumentiert.		
<i>Entwurfsmuster werden im Code angewendet, aber nicht dokumentiert.</i>	<i>Die Entwurfsmuster sind im UML-Diagramm ersichtlich</i>	<i>In der Dokumentation werden die verwendeten Entwurfsmuster begründet. Im UML-Diagramm sind sie mit den Schnittstellen ersichtlich.</i>

5.2.1.5 Saubere Implementation des Designs.

20% der Gesamtnote

Die Klassen sind entsprechend den UML-Diagrammen implementiert und paketiert.		
<i>Im Code finden sich zusätzliche Elemente (Attribute, Methoden, Klassen usw.), die im Entwurf fehlen.</i>	<i>Die Klassen sind korrekt und vollständig implementiert. Es fehlt die Paketierung.</i>	<i>Alle Vorgaben des Entwurfs sind im Code umgesetzt.</i>

Die Dokumentation jeder Klasse ist vollständig (Autor, Version, Datum, Beschreibungen) und die API erstellt .		
<i>Die Beschreibung der Klasse und deren Methoden erklärt deren Anwendung nicht oder zu wenig klar.</i>	<i>Die Beschreibung ist nicht vollständig. Es fehlen Elemente wie Version usw.</i>	<i>Zu jeder Klasse liegt eine vollständige und aussagekräftige API vor.</i>
Der Code ist verständlich und wartbar.		
<i>Die verwendeten Programmstrukturen wirken zufällig, so dass Änderungen nur schwer zu implementieren sind.</i>	<i>Der Code ist mehrheitlich gut strukturiert. Für Änderungen bedarf es einer vertieften Analyse, um den Code umfassend zu verstehen.</i>	<i>Der Code ist gut strukturiert (funktionale Blöcke) und wo nötig mit erklärendem Kommentar unterlegt. Änderungen können einfach vorgenommen werden.</i>

5.2.1.6 Geforderte Funktion wird erreicht.

5% der Gesamtnote

Die Applikation erfüllt die geforderten Funktionalitäten .		
<i>Es werden nur wenige der funktionalen Anforderungen erfüllt.</i>	<i>Die funktionalen Anforderungen werden zu einem grossen Teil erfüllt.</i>	<i>Es werden alle funktionalen Anforderungen der Spezifikation erfüllt.</i>

5.2.1.7 Erstellen korrekter Testfälle und durchführen der Tests, dokumentiert mittels Testprotokoll.

10% der Gesamtnote

Die gewählten Testfälle geben die Funktionalität der UseCases wieder.		
<i>Es werden nur einige UseCases abgedeckt.</i>	<i>Es wird eine Mehrheit der UseCases abgedeckt.</i>	<i>Es werden alle UseCases abgedeckt.</i>
Die Testfälle sind vor der Implementierung erstellt worden.		
		<i>Es liegt ein unterschriebenes Dokument vor, das dokumentiert, dass die Testfälle vor der Implementierung erstellt wurden.</i>
Die Testfälle sind durch JUnit abgedeckt.		
<i>Einige wenige Testfälle können durch JUnit nachgewiesen werden.</i>	<i>Eine Mehrheit der Testfälle können durch JUnit nachgewiesen werden.</i>	<i>Alle Testfälle können mit JUnit-Test nachgewiesen werden.</i>
Das Testprotokoll ist vollständig .		
<i>Das Protokoll ist bezüglich Soll-Ist-Vergleich unvollständig.</i>	<i>Das Protokoll enthält korrekte Soll-Ist-Vergleiche. Es fehlen Hinweise betreffend nötiger Massnahmen.</i>	<i>Das Testprotokoll enthält korrekte Soll-Ist-Vergleiche und wo nötig zu ergreifende Massnahmen.</i>

5.2.1.8 Dokumentation der Architektur durch weitere UML-Diagramme, Analyse des erstellten Designs bezüglich Qualität und/oder beschreiben der verwendeten Architekturmuster.

10% der Gesamtnote

Die Analyse beinhaltet Aktivitätsdiagramme .		
<i>Es werden nur vereinzelte Aktivitäten durch Diagramme dokumentiert.</i>	<i>Die Aktivitäten werden mehrheitlich durch Diagramme dokumentiert</i>	<i>Die Aktivitäten werden vollständig durch Diagramme dokumentiert.</i>
Das Design beinhaltet Zustandsdiagramme und/oder Sequenzdiagramme .		
<i>Es werden nur vereinzelte Abläufe und/oder Zustände dokumentiert.</i>	<i>Es werden einige wichtige Abläufe und Zustände dokumentiert.</i>	<i>Kritische Abläufe und Zustände sind vollständig dokumentiert.</i>
Das gewählte Design unterstützt die einfache Realisierung der Anwendung.		
<i>Design ist nicht der Aufgabe angemessen.</i>	<i>Design weist Schwächen auf.</i>	<i>Design ist klar und auf Problemstellung bezogen.</i>

5.2.2 Persönliche Reflexion

Für _____ Klasse _____

5.2.2.1 Abgabe

5% der Gesamtnote

Die von der Lehrperson vorgegebenen Zeiten werden eingehalten .		
Die Reflexion wird ver- spätet eingereicht		Die Reflexion wird frist- gerecht eingereicht.

5.2.2.2 Tagesziel und Zielerreichung (Soll-Ist-Vergleich)

15% der Gesamtnote

Die Tagesziele sind zu Beginn des Unterrichts bekanntgegeben .		
Es liegen für einzelne Tage Tagesziele vor.		Es liegen für alle Arbeits- tage die Tagesziele vor.
Die Zielerreichung (Soll-Ist) wird aussagekräftig reflektiert .		
Der Soll-Ist Vergleich zeigt nur im Ansatz auf, warum die Ziele er- reicht/nicht erreicht wur- den.		Der Soll-Ist Vergleich zeigt genau auf, warum die Ziele erreicht/nicht erreicht wurden.

5.2.2.3 Kurzer Abriss zur Tätigkeit

30% der Gesamtnote

Für jeden Arbeitstag sind die Tätigkeiten erläutert.		
Es ist nur ein Tag doku- mentiert.	Es sind 2 – 3 Tage doku- mentiert	Es ist pro Arbeitstag eine Reflexion erfolgt.
Die Beschreibungen sind vollständig und verständlich.		
Die Beschreibung ist we- nig bis nicht verständlich UND die Fachsprache genügt nicht den Ansprüchen	Die Beschreibung ist we- nig bis nicht verständlich ODER die Fachsprache genügt nicht den Ansprüchen.	Die Beschreibung ist nachvollziehbar und in korrekter Fachsprache gehalten.

5.2.2.4 Erkenntnisse aus der Tätigkeit (neue Inhalte, Probleme usw.)

40% der Gesamtnote

Zu den wichtigsten Tätigkeiten sind die Erkenntnisse fest gehalten .		
Die Erkenntnisse sind für den Leser nicht nachvoll- ziehbar.		Die Erkenntnisse sind stringent und nachvoll- ziehbar.

5.2.2.5 Ein Fazit zum Tag mit zwei Aussagen

10% der Gesamtnote

Das Tagesfazit ist aussagekräftig und kurz gefasst.		
Es liegt nur ein Fazit vor UND das Statement ist nicht auf den Punkt gebracht.	Es liegt nur ein Fazit vor ODER das Statement ist nicht auf den Punkt gebracht.	Es liegen beide Fazit als kurzes, aussagekräftiges Statement vor.

5.2.3 Bewertung Teamverhalten

Bewertung für

Name

Vorname

Klasse

Bewertung von

Name

Vorname

Kriterium	Gütestufe 1	Gütestufe 2	Gütestufe 3
Zuverlässigkeit	<i>Aufträge werden nicht auf Zeit ausgeführt.</i>	<i>Aufträge werden nur nach expliziter Aufforderung auf Zeit ausgeführt.</i>	<i>Aufträge werden selbständig auf den angegebenen Zeitpunkt hin ausgeführt.</i>
Arbeitsmenge und Qualität	<i>Die geleistete Arbeit genügt in Menge UND Qualität nicht den Anforderungen des Teams.</i>	<i>Die geleistete Arbeit genügt in Menge ODER Qualität nicht den Anforderungen des Teams.</i>	<i>Die geleistete Arbeit erfüllt sowohl in Menge als auch Qualität die Ansprüche des Teams.</i>
Kommunikation	<i>Die Kommunikation fehlt bzw. erfolgt nur auf explizites Nachfragen und ist selten vollständig.</i>	<i>Es wird kommuniziert. Die Information ist aber unvollständig und es Bedarf der Nachfrage.</i>	<i>Die Kommunikation erfolgt proaktiv, d.h. dass wichtige Information jederzeit im Team geteilt wird.</i>
Integration	<i>Wirkt im Team als Fremdkörper</i>	<i>Wirkt im Team mit, kann sich aber nicht voll integrieren</i>	<i>Ist im Team voll und ganz integriert</i>
Fairness	<i>Das Verhalten im Team ist störend, da Arbeiten nicht übernommen werden und unsachgemässe Kritik geäussert wird.</i>	<i>Das Verhalten ist nicht störend, der Einsatz kann aber besser sein.</i>	<i>Das Verhalten ist gewinnbringend für das Team.</i>

Die Bewertung erfolgt durch markieren der zugeteilten Gütestufe. Es sind alle 5 Kriterien zu bewerten.

Datum:

Unterschrift:

5.3 Dokument Tagesziel

Das Tagesziel ist durch jede Lernende / jeden Lernenden persönlich festzuhalten und der Lehrperson zu Beginn des Unterrichts abzugeben. Die Form der Abgabe wird durch die Lehrperson festgelegt.

Das Dokument weist den gezeigten Aufbau auf.

Name und Vorname
Datum
Zielformulierung (SMART)
- Spezifisch
- Messbar
- Attraktiv
- Realistisch
- Terminiert
Unterschrift

Das Tagesziel ist Teil der persönlichen Reflexion (Soll-Ist-Vergleich).

5.4 Gliederung Tagebuch

Das Tagebuch umfasst die folgenden Abschnitte:

5.4.1 Soll-Ist-Vergleich

Hier muss ein Bezug zu den Dokumenten gemäss 5.3 erfolgen. Wichtig ist, dass die erreichten Arbeiten mit der Planung zu vergleichen und Abweichungen zu begründen sind.

5.4.2 Abriss der Tätigkeit

In einem kurzen Abschnitt (10 bis 20 Sätze) wird die Tätigkeit des Tages beschrieben. (vergleichbar einem Deutschauufsatz)

5.4.3 Erkenntnisse

In diesem Abschnitt erfolgt die Reflexion des Tages. Aus Soll-Ist-Vergleich und der Tätigkeitsbeschreibung erfolgt ein Resümee. Es zeigt auf, welche Erkenntnis der aktuelle Tag gebracht hat. Die Erkenntnis kann sich auf die persönliche Situation, das Team, die eigenen Fortschritte usw. beziehen und ist in ich-Form festzuhalten.

5.4.4 Fazit

Ihre Gedanken zum aktuellen Tag zu

- a) das hat mich gefreut
- b) das hat mich geärgert

UND BITTE BEACHTEN! Der Autor sollte im Dokument erwähnt sein.

5.5 Gestaltung Bericht

Der Bericht weist ein Deckblatt und ein Inhaltsverzeichnis auf. Auf dem Deckblatt sind die Namen der Teammitglieder festzuhalten.

Auf der letzten Seite findet sich eine Deklaration zur Bestätigung der Eigenleistung. Diese muss von jedem Teammitglied unterschrieben werden.