

Lernziele zur Vorlesung Software Engineering I

Version 1.5 vom 11.10.2004¹

Martin Glinz
Prof. Dr. rer. nat.
Universität Zürich

Vorbemerkung

Die Vorlesung Software Engineering I ist Bestandteil des Prüfungsgebiets Informatik Grundstufe in der Vorprüfung der Studiengänge in Ökonomie und Wirtschaftsinformatik.

Die Lernziele in diesem Katalog beschreiben den prüfungsrelevanten Stoff der Vorlesung. Die Lernziele sollen Ihnen bei der Prüfungsvorbereitung helfen, indem sie den Stoffumfang festlegen, die Art der zu erwartenden Aufgaben deutlich machen und Ihnen aufzeigen, in welcher Tiefe Sie den Stoff beherrschen müssen.

Lernziele in *kursiver* Schrift sind *nicht* Bestandteil der Prüfung Informatik Grundstufe. Sie beschreiben Stoff, den Studierende der Wirtschaftsinformatik zusätzlich beherrschen müssen, weil er Grundlage für das Software-Praktikum im vierten Semester und für das Hauptstudium in Wirtschaftsinformatik ist.

Die Prüfungsaufgaben werden teilweise als Multiple-Choice-Aufgaben (mit mehreren richtigen Antworten) und teilweise als Textaufgaben gestellt. Der Lernzielkatalog wird in der Regel jährlich aktualisiert. Für die Prüfung gilt die jeweils aktuelle Fassung.

Hinweise zur Terminologie in diesem Zielkatalog

- Etwas «sinngemäß richtig» wiedergeben heißt es nicht wörtlich, aber dem Sinn nach korrekt und vollständig aufschreiben. Das gilt insbesondere für die Wiedergabe von Definitionen.
- Eine Sache «kennen» heißt
 - in Multiple-Choice-Fragen richtige von falschen Aussagen zu dieser Sache unterscheiden,
 - das Wissen über diese Sache in einer gegebenen Fallstudie anwenden.
- Sachen «nennen» oder «aufzählen» heißt die Namen der Sachen angeben.
- «Skizzieren» heißt kennen und in Stichworten und/oder einer grafischen Skizze darstellen.
- Eine Sache «beschreiben» heißt kennen und ihre wesentlichen Elemente in einem zusammenhängenden Text gut verständlich darstellen.
- «Eine Aufgabe lösen» heißt die betreffende Aufgabe sowie andere Aufgaben der gleichen Art lösen können.

¹ Gegenüber der Vorgängerversion wurden die Detailziele zu Kapitel 5 um einen Punkt erweitert.

Lernziele Kapitel 1

Globalziele

- Über den Charakter von Software Bescheid wissen.
- Wissen, dass und warum Softwareentwicklung schwierig ist.
- Eine Grundvorstellung der Ideen, Ziele und Mittel des Software Engineerings haben.

Detailziele

- Die Definitionen 1.1 bis 1.4 sinngemäß richtig wiedergeben.
- Das Kostenverhältnis von Hardware, Software-Entwicklung und Software-Pflege (Bild 1.1) näherungsweise richtig wiedergeben.
- Die Feststellung 1.2 und ihre Konsequenzen kennen. Mindestens drei Konsequenzen aufzählen.
- Die Feststellung 1.7 und die Klein-Groß-Gegensätze aus Tabelle 1.1 kennen. Mindestens drei Gegensätze aufzählen.
- Die Regel 1.1 sinngemäß richtig wiedergeben.
- Den in Bild 1.6 beschriebenen Effekt skizzieren.
- *Die Hintergründe, welche Software Engineering notwendig machen (Kapitel 1.3), beschreiben.*
- Die Aufgaben 1.1 bis 1.5 lösen.

Lernziele Kapitel 2

Globalziele

- Die Rolle und Bedeutung von Zielsetzung und Zielverfolgung im Software Engineering kennen.
- Wissen, dass Ziele durch Messung quantitativ erfassbar werden.

Detailziele

- Das Klassifikationsschema für Ziele (Bild 2.2) sinngemäß richtig wiedergeben.
- Die informelle Definition eines Maßes (erster Abschnitt in Kapitel 2.4.1) sinngemäß richtig wiedergeben.
- Die fünf Skalentypen aufzählen. Die Skalentypen in Stichworten charakterisieren und ein Beispiel zu jedem Typ nennen.
- Zu einer gegebenen Skala den Skalentyp bestimmen.
- Wissen, dass es direkte und indirekte Maße gibt. Den Unterschied charakterisieren und je ein Beispiel nennen.
- Die Aufgaben 2.1 bis 2.5 lösen.

Lernziele Kapitel 3

Globalziele

- Wissen, auf welchen Grundphänomenen Software-Prozesse aufbauen und diese Phänomene erklären können.
- Typische Prozessmodelle für Software-Entwicklung kennen.
- Arten und Einsatzgebiete von Prototypen kennen.
- Den Stellenwert von Wiederverwendung und Beschaffung im Entwicklungsprozess kennen.

- Wissen, was Software-Pflege ist und grundlegende Gesetzmäßigkeiten des Pflegeprozesses kennen.

Detailziele

- Die Definitionen 3.1 und 3.4 bis 3.10 sinngemäß richtig wiedergeben.
- *Die Definition 3.3 sinngemäß richtig wiedergeben.*
- Die drei Stadien des Software-Lebenslaufs aufzählen.
- Die Tätigkeiten in der Entwicklung einer Einzelkomponente kennen. Mindestens vier Tätigkeiten reihenfolgerichtig aufzählen.
- Die Rolle von Iterationen im Lebenslauf und in verschiedenen Prozessmodellen skizzieren.
- Ursachen und Folgen der Software-Evolution beschreiben; die Feststellungen 3.3 und 3.4 kennen; die Feststellung 3.4 begründen.
- Den Unterschied zwischen stabiler Software (S-Typ) und evolvierender Software (P- und E-Typ) kennen; die drei Typen skizzieren.
- Das Arbeiten mit Meilensteinen beschreiben. Begründen, warum Meilensteine messbar / überprüfbar sein müssen.
- Die Bedeutung des Wasserfallmodells aus heutiger Sicht nennen.
- Ein ergebnisorientiertes Phasenmodell mit mindestens 5 Phasen skizzieren. Seine Vor- und Nachteile kennen; je zwei Vor- und Nachteile aufzählen.
- Eine typische Situation skizzieren, in der ein ergebnisorientiertes Phasenmodell verwendet wird.
- Die Grundidee des Wachstumsmodells beschreiben. Seine Vor- und Nachteile kennen; je zwei Vor- und Nachteile aufzählen.
- Eine typische Situation skizzieren, in der ein Wachstumsmodell verwendet wird.
- *Das Prinzip der kontinuierlichen Integration sowie deren Vor- und Nachteile kennen.*
- *Die Grundidee des Spiralmodells von Boehm beschreiben und sein Einsatzgebiet nennen. Das Boehm'sche Spiralmodell von als "Spiralmodell" bezeichneten Wachstumsmodellen unterscheiden.*
- Die Prinzipien der agilen Software-Entwicklung aufzählen.
- *Die Voraussetzungen für das Funktionieren agiler Software-Entwicklung skizzieren.*
- Die drei Arten des Prototypings aufzählen und jeweils mit wenigen Sätzen charakterisieren.
- Typische Projektsituationen, in denen die verschiedenen Arten des Prototypings sinnvoll angewendet werden können, skizzieren.
- Für eine gegebene Projektsituation entscheiden und begründen, welche Art von Prototyping dort eingesetzt werden kann.
- Die Charakteristika von Wegwerfprototypen und Pilotsystemen kennen und je ein Charakteristikum aufzählen.
- Aufzählen, welche Prototyparten Wegwerfprototypen sind.
- Die wirtschaftliche Bedeutung von Mehrfachverwendung bei der Software-Entwicklung erläutern.
- Die Gründe kennen, warum Wiederverwendung sich nicht von selbst ergibt. Zwei Gründe aufzählen.
- Die Regeln 3.1, 3.2 und 3.6 und die Gründe dafür kennen. Je einen Grund aufzählen.
- *Die Regel 3.3 und die Gründe dafür kennen.*
- Die Aufgaben 3.1 bis 3.5 lösen.

Lernziele Kapitel 4

Globalziele

- Die Prinzipien der Projektplanung, -verfolgung und -lenkung kennen.
- Die Rolle und Bedeutung der Risikoführung bei der Führung von Software-Projekten kennen.

Detailziele

- Typische Organisationsstrukturen für kleine, mittlere und große Projekte skizzieren.
- Die sechs W-Fragen, die ein Projektplan beantworten muss, aufzählen.
- *Die charakteristischen Unterschiede bei der Planung und Abwicklung verschiedener Projektarten (Muster 4.1 bis 4.4) kennen (keine Details, die kann man bei Bedarf nachschlagen).*
- Die Schritte bei linearer und bei inkrementeller Planung (Muster 4.6 und 4.7) grafisch skizzieren.
- Die Vorgehensweise bei Terminverfolgung, Sachzielverfolgung, Kostenverfolgung und Risikoverfolgung beschreiben.
- Beschreiben, wann Lenkungsmaßnahmen erforderlich sind und wann der Projektplan angepasst werden muss. Mögliche Lenkungsmaßnahmen bei Terminüberschreitung kennen, drei Maßnahmen nennen.
- Die Nichtlinearität der Wirkung von Lenkungsmaßnahmen auf ein Software-Projekt und das Problem unbeabsichtigter Nebenwirkungen kennen.
- *Die notwendigen Arbeiten beim Abschluss eines Projekts kennen.*
- Vier typische Risiken bei Software-Projekten (Tabelle 4.1) nennen.
- Das Vorgehen bei der Risikoführung, insbesondere die Beurteilung von Risiken mit Hilfe von Risikofaktoren skizzieren.
- In einer Grafik der Art von Bild 4.7 die Gefährlichkeit gegebener Risiken beurteilen.
- Die Aufgaben 4.1 bis 4.3 lösen.

Lernziele Kapitel 5

Globalziele

- Die Problematik der Aufwandschätzung bei Software verstehen.
- Wichtige empirische und algorithmische Schätzverfahren kennen.

Detailziele

- Zwei Gründe für die oft krassen Fehleinschätzungen von Aufwendungen für Software nennen.
- Die empirischen Schätzverfahren «Expertenschätzung» und «Delphi-Methode» skizzieren und je einen Vorteil und einen Nachteil nennen.
- Die Techniken und Probleme der Expertenschätzung kennen.
- Die Charakterisierung algorithmischer Verfahren (Feststellung 5.3) sinngemäß richtig wiedergeben.
- Die Prinzipien von COCOMO beschreiben: Schätzung der Produktgröße als Basis, $MM = a KDSI^b$, $TDEV = c MM^d$, Unterscheidung von drei verschiedenen Software-Arten, Korrektur des Nominalwerts mit dem Produkt verschiedener problem- und projektspezifischer Kostenfaktoren. Die genauen Werte für die Faktoren und Exponenten a, b, c, d müssen nicht auswendig gelernt werden; die ungefähre Größenordnung muss aber bekannt sein.
- Die Prinzipien des Function Point-Verfahrens beschreiben: Maß zur Bewertung der Funktionalität, Aufwandschätzung über Erfahrungswerte (Aufwand pro Function Point), Zählung von Dateneingaben, -ausgaben, Anfragen, Schnittstellen und internen Datenbeständen, Ge-

wichtung nach «einfach», «mittel» und «komplex», Function Point Rohwert als gewichtete Summe, Gewichtung des Rohwerts mit dem Wertkorrekturfaktor, drei Einflussfaktoren als Beispiele nennen.

Die genauen Werte der Tabellen 5.1, 5.2 und 5.3 müssen nicht auswendig gelernt werden.

- Die Verfahren COCOMO und Function Point miteinander vergleichen.
- Die Faustregeln 5.2 und 5.3 für Pflegekosten sinngemäß richtig wiedergeben.
- *Den in Bild 5.8 skizzierten Einfluss der Schätzung auf den tatsächlichen Aufwand beschreiben und begründen.*
- Die Aufgaben 5.1 und 5.2 lösen (bei 5.2 unter der Annahme, dass die Berechnungstabellen und Formeln vorgegeben werden).

Lernziele Kapitel 6

Globalziele

- Wissen, was Konzipieren von Lösungen ist und wozu man das braucht.
- Wichtige Entwurfsprinzipien und die Hauptaufgaben des Architekturentwurfs kennen.
- Einen Überblick über Vorgehensweisen, typische Stile und Arten der Wiederverwendung haben
- *Die Entwurfsprinzipien, den Entwurfsprozess und wichtige Architekturstile kennen.*
- *Die Möglichkeiten der Wiederverwendung von Architekturen und Entwurfswissen kennen.*

Detailziele

- Die Gründe, warum die Erstellung eines Lösungskonzepts notwendig ist, beschreiben.
- Die Definitionen 6.1 bis 6.3 sinngemäß richtig wiedergeben.
- Das Entwurfsprinzip «Modularität» (ohne die verschiedenen Arten von Kohäsion und Kopplung) skizzieren. Die Definitionen 6.5 und 6.6 sinngemäß richtig wiedergeben.
- Das Geheimnisprinzip skizzieren. Die Definition 6.7 sinngemäß richtig wiedergeben.
- Die Grundlagen des Entwurfsprinzips «Nebenläufigkeit» (erster und zweiter Abschnitt von 6.3.5) kennen.
- Die Entwurfsprinzipien «Nutzung von Vorhandenem» und «Qualität» kennen.
- *Alle zehn Entwurfsprinzipien beschreiben.*
- *Die Einbettung des Architekturentwurfs in den Software-Entwicklungsprozess (6.5.1) beschreiben.*
- Die drei Hauptaufgaben des Architekturentwurfs (Tabelle 6.1) nennen. Jede der drei Aufgaben skizzieren.
- *Das Vorgehen beim Konzipieren (6.5.3) und die Variantenbehandlung (6.5.4) beschreiben.*
- Die fünf Schritte bei der Beschaffung von Lösungen aufzählen. *Jeden Schritt skizzieren.*
- Die Grundideen der objektorientierten Architektur skizzieren. Die Kooperationsmechanismen und die Vor- und Nachteile der objektorientierten Architektur kennen.
- Die Grundideen der komponentenorientierten Architektur skizzieren. Client/Server- und Middleware-Architektur kennen.
- *Wissen, was ein Architekturstil ist. Die Grundideen und Eigenschaften der funktionsorientierten, der datenorientierten, der objektorientierten, der prozessorientierten und der komponentenorientierten Architektur mit ihren jeweiligen Eigenschaften beschreiben.*
- Die Grundideen von Entwurfsmustern und Rahmen skizzieren.
- *Die Definitionen 6.10 bis 6.12 sinngemäß richtig wiedergeben.*
- *Die in 6.7.1 und 6.7.3 beschriebenen Muster skizzieren.*
- Das Prinzip der Architekturmetapher kennen. Drei Metaphern aufzählen und skizzieren.

- *Alle dargestellten Architekturmetaphern skizzieren. Die Definition 6.13 sinngemäß richtig wiedergeben.*
- Die Aufgaben 6.1 und 6.2 lösen.

Lernziele Kapitel 7

Globalziele

- Wissen, was Spezifikation von Anforderungen ist und wozu man das braucht.
- Den Spezifikationsprozess kennen.
- Einen Überblick über die grundsätzlichen Möglichkeiten der Darstellung von Anforderungen und über typische Spezifikationsmethoden haben.
- Den Stellenwert des Spezifizierens von Anforderungen im Software-Entwicklungsprozess beurteilen.

Detailziele

- Die Definitionen 7.1 bis 7.3 sinngemäß richtig wiedergeben.
 - Begründen, warum das Spezifizieren von Anforderungen wirtschaftlich ist.
 - Die charakteristischen Eigenschaften einer guten Spezifikation kennen. Mindestens vier Eigenschaften aufzählen.
 - Die charakteristischen Eigenschaften eines guten Spezifikationsprozesses kennen. Mindestens zwei Eigenschaften aufzählen.
 - Den Spezifikationsprozess skizzieren (Die Details von Bild 7.3 müssen nicht auswendig gelernt werden).
 - Die vier inhaltlichen Aspekte einer Anforderungsspezifikation kennen und aufzählen.
 - Die vier möglichen Techniken zur Gewinnung von Anforderungen (7.4.2) kennen und aufzählen.
 - Drei Techniken zur Informationsgewinnung kennen und aufzählen.
 - Die Bedeutung der Einbettung eines Systems in seinen Kontext und der Abgrenzung zwischen System und Kontext für die Gewinnung von Anforderungen kennen.
- Die Gründe pro und contra IST-Analyse kennen und je einen Grund aufzählen.
- Die Darstellungsaspekte konstruktiv-deskriptiv, Formalitätsgrad und Gliederungs-/Abstraktionsmittel kennen.
 - Die Vor- und Nachteile informaler, teilformaler und formaler Darstellung kennen und je einen aufzählen.
 - *Die drei Abstraktionsarten skizzieren.*
 - Die Grundideen von datenflussorientierter Spezifikation, Verhaltensspezifikation mit Automaten, objektorientierter Spezifikation und der Spezifikation mit Anwendungsfällen skizzieren.
 - *Alle in Abschnitt 7.5.4 genannten Spezifikationsmethoden beschreiben.*
 - Das Prüfen von Anforderungen skizzieren. Die vier Prüfverfahren nennen.
 - Die Aufgaben 7.1 bis 7.3 lösen.

Lernziele Kapitel 8

Globalziele

- Einen Überblick über Vorgehensweisen und Probleme der Realisierung haben.
- *Kriterien und Merkmale guter und schlechter Programme kennen.*

Detailziele

- Die Möglichkeiten der Realisierung aufzählen.
- *Die Kriterien des systematischen Programmierens kennen. Mindestens drei Punkte aufzählen.*
- *Bei einem gegebenen Programm beurteilen, ob es den Kriterien des systematischen Programmierens genügt.*
- Das Vorgehen zur Prüfung und Integration von Programmcode skizzieren.

Lernziele Kapitel 9

Globalziele

- Die Grundsätze des Qualitätsmanagements und eines Qualitätsmanagementsystems kennen.
- Einen Überblick über Prinzipien und Verfahren der Qualitätsprüfung haben.
- Die Technik des Reviews kennen.

Detailziele

- Die Definitionen 9.1 bis 9.6 sinngemäß richtig wiedergeben.
- Die typische Aufbauorganisation in einem Qualitätsmanagementsystem (Bild 9.3) skizzieren.
- Die sechs Grundsätze des Qualitätsmanagements kennen. Mindestens drei Grundsätze aufzählen.
- Die Aufgaben und Mittel von Qualitätsplanung, Qualitätslenkung und Qualitätsprüfung skizzieren. Ihre Zusammenhänge (Bild 9.4) skizzieren.
- In einer Skizze der Art von Bild 9.5 angeben, wo es um Validierung und wo um Verifikation geht.
- Die vier wichtigsten Prüfverfahren aufzählen und skizzieren.
- Die Verfahren Inspektion und Walkthrough skizzieren.
- Die Schritte im Ablauf eines Reviews nennen; jeden Schritt skizzieren.
- Die vier Rollen der an einem Review beteiligten Personen nennen; jede Rolle skizzieren.
- Die Review-Regeln (9.4.4) kennen. In einer gegebenen Beschreibung eines Reviewablaufs erkennen, wo welche Regel angewendet wird bzw. gegen welche Regel verstoßen wird.
- Entstehung und Behandlung des Review-Berichts skizzieren.
- Die zwei Testziele sinngemäß richtig wiedergeben.
- Begründen, warum die Korrektheit eines Programms nicht mit Testen bewiesen werden kann.
- Die drei Phasen eines Tests nennen; jede Phase skizzieren. Die Rolle der Fehlerbehebung in diesem Zusammenhang skizzieren.
- *Die zwei Klassen von Testverfahren beschreiben.*
- Die Aufgaben 9.1, 9.2 und 9.4 lösen.
- *Die Aufgabe 9.3 lösen.*

Lernziele Kapitel 10

Globalziele

- Die Bedeutung der Dokumentation für Software beurteilen.
- Einen Überblick über die wichtigsten Dokumente haben.

Detailziele

- Die zwei Dokumentationsarten nennen.
- Die drei Hauptaufgaben der Dokumentation skizzieren.
- Die Produktdokumente kennen. Mindestens vier Dokumente nennen und skizzieren.
- Die Projektdokumente kennen. Mindestens ein Dokument nennen und skizzieren.
- Die Aufgaben 10.1 und 10.2 lösen.

Lernziele Kapitel 11**Globalziele**

- Wissen, welche Bedeutung die Konfigurationsverwaltung in der Entwicklung und Pflege von Software hat.
- Die zentralen Elemente der Konfigurationsverwaltung kennen.

Detailziele

- Die Definitionen 11.1 bis 11.3 sinngemäß richtig wiedergeben.
- Den Ablauf einer Änderung (Bild 11.5) skizzieren.
- Die Umgebungen für die Herstellung, Verwaltung und Nutzung von Software-Einheiten und die Zusammenhänge zwischen diesen Umgebungen (Bild 11.4) skizzieren.
- Den typischen Bearbeitungsablauf für Problemmeldungen skizzieren.
- In einem gegebenen Fallbeispiel Probleme erkennen und nennen, die durch Konfigurationsverwaltung gelöst oder gemildert werden. Für das gegebene Fallbeispiel skizzieren, welche Mittel der Konfigurationsverwaltung mit welcher Wirkung eingesetzt werden können.

Lernziele Kapitel 12**Globalziele**

- Die Rolle von Werkzeugen, Mehrfachverwendung und beteiligten Menschen für die Software-Produktivität kennen.

Detailziele

- Den Kontext von Werkzeugen in einer Strategie für computergestütztes Software Engineering (Bild 12.1) skizzieren.
- Den Investitionscharakter von Werkzeugen kennen.
- *Die in 12.1.3 genannte Klassifizierung von Werkzeugen kennen und die einzelnen Klassen skizzieren.*
- Die drei Regeln für die Disponibilität von Personal kennen und aufzählen.
- *Alle in 12.3.1 genannten Regeln skizzieren.*
- Die Aufgaben 12.1 und 12.2 lösen.