

# Softwaretechnik

## Einsendeaufgabe 1

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>2</b>
<b>1    Aufgabe 1 – Definition von Softwaretechnik (einige Sätze) .....</b>	<b>I</b>
<b>2    Aufgabe 2 – Warum ist Softwarequalität so wichtig (einige Sätze)? .....</b>	<b>I</b>
<b>3    Aufgabe 3 – Was ist für Sie Softwarequalität (einige Sätze)? .....</b>	<b>I</b>
<b>4    Die drei größten Probleme, warum Softwareprojekte schief gehen können (<math>\geq 9</math> Sätze). ....</b>	<b>II</b>
<b>5    Phasen des Softwarelebenszyklus.....</b>	<b>II</b>
<b>6    Persönlicher Bezug zur Softwaretechnik (2-3 Links).....</b>	<b>III</b>

## 1 Aufgabe 1 – Definition von Softwaretechnik (einige Sätze)

Eine allgemeine Definition, wie ich sie einem Freund geben würde, wäre vermutlich dies:

*„In das Feld der Softwaretechnik fällt alles rund um Software, welches nicht das Schreiben des Quellcodes selbst ist.“*

Etwas formeller ausgedrückt: *„Der Begriff der Softwaretechnik beschreibt alle Tätigkeiten, Methoden und Aktivitäten zur Konzeption, Erstellung/Veränderung und Bereitstellung von Anwendungssystemen.“*

Das bedeutet, dass alle Tätigkeiten außer der Programmierung selbst in das Themenfeld fallen, wie etwa Anforderungsanalyse, Softwaredesign und -architektur oder Entwicklungsmethoden.

## 2 Aufgabe 2 – Warum ist Softwarequalität so wichtig (einige Sätze)?

Allgemein formuliert ist Qualität die Summe aller Eigenschaften eines Objektes. Für Software ist dies allerdings nicht ganz einfach zu fassen, da

- Software komplex ist
- verschiedene Stakeholder in Softwareprojekten unterschiedliche Ansprüche an Softwarequalität stellen

Softwarequalität stellt sicher, dass alle Ansprüche aller Beteiligten bei der Planung und Umsetzung von vornherein berücksichtigt werden und somit das Endergebnis für alle zufriedenstellend ist.

## 3 Aufgabe 3 – Was ist für Sie Softwarequalität (einige Sätze)?

In erster Linie umfasst der Begriff Softwarequalität sämtliche Einflussfaktoren auf die Gebrauchstauglichkeit einer Anwendung. Hierzu gehören:

- Funktionalität: eine Anwendung muss den Zweck erfüllen, zu dessen Erfüllung sie erstellt wurde.
- Gebrauchstauglichkeit (Softwareergonomie): die Anwendung sollte so gestaltet sein, dass Anwender sich leicht zurechtfinden und ohne Schwierigkeiten ihre Aufgaben erledigen können.

- Performance: die Geschwindigkeit der Software sollte (entsprechend der an sie gestellten Aufgaben) angemessen sein. Bspw. dürfen einfache Suchanfragen nicht übermäßig lange dauern.
- Zuverlässigkeit: eine Anwendung muss zuverlässig verfügbar sein (dies gilt insbesondere für Server- und Webanwendungen).
- Wartbarkeit: eine Anwendung muss sich leicht pflegen, ändern und erweitern lassen, etwa damit sie später geänderten Bedürfnissen angepasst werden kann.

#### 4 Die drei größten Probleme, warum Softwareprojekte schief gehen können ( $\geq 9$ Sätze).

Aus meiner Sicht sind die drei größten Probleme:

1. Unterschätzen der Komplexität  
Software ist inhärent komplex. Selbst für den Menschen einfache oder geradezu selbstverständliche Problemstellungen müssen in der Softwareentwicklung anders angegangen werden (bspw. Abbruchbedingungen, Edge-Cases etc.). In der Praxis führt dies oftmals dazu, dass am Anfang noch nicht alle (Rand-) Bedingungen klar definiert sind. In der Folge tritt dann häufig Hofstadter's law ein, sodass sich das Softwareprojekt verspätet und/oder die Qualität leidet.
2. Zu wenig kritisches Hinterfragen  
Anforderungen an Softwaresysteme sind oftmals unvollständig und nicht immer richtig. Nur durch kritisches Hinterfragen lässt sich prüfen, ob eine Anforderung überhaupt sinnvoll ist. Andernfalls besteht die Gefahr, dass das Endprodukt trotz Erfüllung der Anforderungen nicht gebrauchstauglich ist.
3. Unzureichendes Qualitätsmanagement  
Die Qualität von Softwaresystemen muss fortwährend getestet werden, damit Fehler frühzeitig gefunden und korrigiert werden können. Je später Fehler gefunden werden, desto höher sind die Kosten für die Korrektur.

#### 5 Phasen des Softwarelebenszyklus

1. Analysephase
2. Definitionsphase
3. Entwurfsphase

4. Implementierungsphase
5. Abnahme- und Einführungsphase
6. Wartungs- und Pflegephase

## 6 Persönlicher Bezug zur Softwaretechnik (2-3 Links)

Ich erhoffe mir vom Modul, dass ich meine Erfahrungen mit theoretischem Wissen und Methoden rund um Entwicklung und Qualität, Change- und Anforderungsmanagement unterfüttern kann. Hierzu gehören:

1. Verknüpfung von Projektmanagement mit Softwareentwicklung (Stichwort [Peopleware](#))
2. Design Patterns wie OOD ([1](#), [2](#))
3. [Software Architektur](#)