

# Muster der Softwaretechnik-Lehre

Valentin Dallmeier · Florian Gross · Clemens Hammacher · Matthias Höschele ·  
Konrad Jamrozik · Kevin Streit · Andreas Zeller

Lehrstuhl für Softwaretechnik, Universität des Saarlandes  
{dallmeier, fgross, hammacher, hoeschele, jamrozik, Streit, zeller}@cs.uni-saarland.de

## 1 Wissen der Softwaretechnik- Lehre organisieren

Das Organisieren von Softwaretechnik-Projekten ist für neue Dozenten stets eine Herausforderung: Man muss Kunden finden, Projekte definieren, Tutoren schulen, Anforderungen definieren, Terminpläne machen, Regeln aufstellen. . . und zum Schluss das Projektergebnis bewerten. Eine ungenügende Planung und Umsetzung kann schnell das gesamte Projekt gefährden. Daher empfiehlt es sich, für die häufigsten Fragen und Probleme Lösungen *explizit* aufzuschreiben. Dies können zunächst immer wiederkehrende Dinge sein, wie die Frage, wann welche Vorlesungsinhalte benötigt werden, oder wie die Begutachtung von Praktikums-Dokumenten organisiert werden muss. Andere Fragen mögen zunächst banal erscheinen, machen aber schnell deutlich, warum es sich empfiehlt, Wissen explizit zu dokumentieren:

- Ich möchte eine Grillfeier für 160 Teilnehmer organisieren. Wieviele Getränke und Speisen muss ich pro Person ansetzen? Woher bekomme ich Grills, Würstchen, Besteck, Salate?
- Ich möchte eine „Messe“ organisieren, auf dem meine Praktikums-Teilnehmer ihre Projekte vorstellen können. Was brauche ich, und woher bekomme ich es? Wann muss ich wen verständigen?

Um das Wissen über Softwaretechnik-Lehre zu verwalten, haben wir ein *Wiki* eingerichtet, um das Wissen über Softwaretechnik-Lehre zu organisieren und zu dokumentieren – und zwar in einer Form, die sowohl *lokales* als auch *standortübergreifendes* Wissen dokumentieren kann. Die Artikel nutzen das Format der *Muster* – analog zu Entwurfsmustern als *Lösungsschablonen für immer wiederkehrende Probleme*:

**Name.** Jedes Muster hat einen *Namen*, damit es unter diesem Namen einfach referenziert und nachgeschlagen werden kann.

**Ziel.** Jedes Muster löst ein bestimmtes *Problem*, das hier allgemein charakterisiert werden kann.

**Motivation.** Das Problem sollte *relevant* sein – also häufig oder in verschiedenen Ausprägungen immer und immer wieder auftreten.

**Anwendung.** Dies ist eine Beschreibung, wie das Muster umgesetzt wird – sowohl in abstrakter

(standortunabhängiger) Form als auch illustriert an konkreten Beispielen. Die Anwendung sollte zeigen, wie das Muster das Problem löst.

### Zusammenspiel mit anderen Mustern.

Voraussetzungen in Form von anderen Mustern sowie Konflikte mit anderen Mustern sollten ebenfalls dokumentiert werden.

**Folgen.** Was sind die Folgen dieses Musters? Hier sind insbesondere *unerwartete* und *negative* Folgen gemeint, die dem Leser bewusst sein sollten.

**Tipps und Tricks.** In diesem Abschnitt können Hilfestellungen zur Umsetzung des Musters, sowie gemachte Erfahrungen dokumentiert werden.

**Bekannte Einsätze.** Dieser Abschnitt verweist auf die Veranstaltungen, in denen das Muster umgesetzt wurde. Die Wiki-Einträge zu den Veranstaltungen wiederum beschreiben die dort eingesetzten Muster, sowie ggf. Kontaktpersonen und Webseiten mit weiterführenden Informationen.

Abbildung 1 zeigt eins der derzeit gut 50 Muster.

## 2 Ein Wiki für Erfolgsrezepte

Auf [www.sewiki.de](http://www.sewiki.de) haben wir unser internes (lokales) Wissen in Form von Mustern zu organisieren. Wir möchten damit unser Wissen der Gemeinschaft bereitstellen, es aber auch jedem Interessierten ermöglichen, sie mit eigenen Erfahrungen weiter zu verfeinern, oder eigene Muster hinzuzufügen. Noch spiegeln diese Seiten unsere eigenen „Kochrezepte“ wieder, die zudem stark lokal geprägt sind. Unsere Vision ist aber, dass wir weitere solche Muster sammeln und weiter ausbauen können – und so das Wissen über erfolgreiche Lösungen der Softwaretechnik-Lehre lebendig halten können. In diesem Sinne rufen wir alle auf, die sich mit der Lehre der Softwaretechnik beschäftigen, ihre eigenen Erfolgsrezepte beizusteuern unter

<http://www.sewiki.de/>

**Danksagung.** Die Idee, ein Wiki zu nutzen, um Konzepte der Softwaretechnik-Lehre auszutauschen, entstand 2008 in Diskussionen mit Kurt Schneider (Hannover), dem wir zu tiefem Dank verpflichtet sind. Die Muster der Saarbrücker Veranstaltungen gehen auf jahrelange Feinarbeit mit zahlreichen Tutoren zurück, für deren Anregungen wir ebenfalls sehr dankbar sind.

## Muster: Spiel als Aufgabenstellung

### Ziel

Ziel ist es zum einen, die Zeit zu minimieren, die benötigt wird um sich domänenspezifisches Wissen bezüglich der Aufgabe anzueignen. Zum anderen jedoch, und viel wichtiger, zeigen sich die Studierenden bei Verwendung eines Spiels als Implementierungsziel besonders motiviert. Der Sinn der Aufgabenstellung wird kaum kritisiert.

### Motivation

Insbesondere in einem  $\rightarrow$ *Vollzeit-Praktikum*, das innerhalb recht kurzer Zeit stattfindet, ist es wichtig, dass nicht zu viel Zeit aufgewendet werden muss, um die Aufgabenstellung und die sich daraus ergebenden Konsequenzen zu verstehen. [...] Mit der Verwendung von Spielen, insbesondere von Brettspielen, als Implementierungsziel wurde die Erfahrung gemacht, dass die Studierenden innerhalb von einem bis maximal zwei Tagen die Aufgabe klar verstanden haben.

Ein weiterer Vorteil von Spielen ist die gesteigerte Motivation und Lernbereitschaft der Studierenden. Die initiale Hürde, die genommen werden muss, um sich qualifiziert mit der Aufgabenstellung auseinanderzusetzen und darüber diskutieren zu können, ist relativ niedrig. Unterschiede in der Vorbildung der Studierenden kommen wenig zum tragen. Selten werden einzelne Gruppen-Mitglieder abgehängt und außen vorgelassen.

Zu guter Letzt sind massenweise geeignete und interessante Spiele, samt deren Anleitungen, verfügbar. Auf diesen Fundus zurückzugreifen spart das komplette Konzipieren einer Aufgabenstellung und erlaubt es dem Veranstalter, sich auf die Umsetzung zu konzentrieren.

### Anwendung

Die Umsetzung besteht im Wesentlichen aus folgenden Schritten:

1. Auswahl eines geeigneten Spiels.
  - (a) Die Regeln des Spiels müssen einfach zu verstehen, also nicht zu komplex sein.
  - (b) Das Spiel sollte Multiplayer-fähig sein. Somit können die Teilnehmer in einem finalen  $\rightarrow$ *Turnier* gegeneinander antreten.
  - (c) Kann das Spiel als  $\rightarrow$ *Client/Server Architektur* umgesetzt werden, kann der Veranstalter einen zentralen Spielserver stellen, der von allen Teilnehmern während der Veranstaltungsdauer genutzt werden kann, um gegeneinander anzutreten und zu testen.
  - (d) Es sollte relativ einfach möglich sein, für das gewählte Spiel eine einfache künstliche Intelligenz zu schreiben.
  - (e) Es sollte möglich sein ein vernünftiges (graphisches) User-Interface zu implementieren, das die Teilnehmer nutzen können, um ihre Implementierung zu testen und zu debuggen.
  - (f) Ist das Spiel geeignet um von Menschen gespielt zu werden, kann der Mensch gegen seine eigene KI antreten, was zusätzlich motiviert. [...]
  - (g) Die Spielregeln sollten genug Raum für eine kreative Umsetzung geben. Auch die Implementierung einer Strategie in der KI sollte genug Möglichkeiten der Differenzierung zwischen den Teilnehmern bieten.
  - (h) Die Implementierung des Spiels sollte einige algorithmische Möglichkeiten bieten.
  - (i) Soll die Aufgabenstellung modifiziert werden ( $\rightarrow$ *Modifikation der Aufgabenstellung*), sollten die Regeln des Spiels dies erlauben.
2. Es sollten keine Implementierungen des Spiels frei verfügbar sein, die die Studierenden einfach kopieren können.
3. Ist die Spielidee rechtlich geschützt, sollten die Rechteinhaber kontaktiert werden, um die Bedingungen der Benutzung zu klären. [...]
4. Erstellen der Aufgabenstellung Ist eine spätere  $\rightarrow$ *Modifikation der Aufgabenstellung* vorgesehen, so kann diese Erweiterte Aufgabenstellung bereits mit erdacht werden.
5. Implementieren der vom Veranstalter vorgegebenen Teile [...]
6. Testbarkeit prüfen [...]
7. Anfertigen einer Referenz-Implementierung der von den Studierenden zu implementierenden Teile
8. Infrastruktur aufsetzen

Für die einzelnen Schritte sollte ausreichend Zeit eingeplant werden. Zum Vergleich: Im  $\rightarrow$ *Software-Praktikum an der Universität des Saarlandes* beginnt der erste Schritt (Auswahl eines geeigneten Spiels) etwa fünf Monate vor Praktikumsbeginn.

### Zusammenspiel mit anderen Mustern

Ein Spiel als Aufgabenstellung eignet sich sehr gut für ein abschließendes  $\rightarrow$ *Turnier*.

### Folgen

...

Abbildung 1: Muster „Spiel als Aufgabenstellung“ (aus [www.sewiki.de](http://www.sewiki.de), gekürzt)