

SOFTWARE WAS HAT ENGINEERING MIT WIRTSCHAFT ZU TUN?

DIE ABSTRAKTION REALER BEGEBENHEITEN

```
return new SavedState[size];  
  
public FragmentTabHost(Context context) {  
    // Note that we call through to the version that takes a AttributeSet  
    // because the simple Context construct can result in a RuntimeException  
    super(context, null);  
    mFragmentManagerTabHost(context, null);  
}  
  
public FragmentTabHost(Context context, AttributeSet attrs) {  
    super(context, attrs);  
}
```

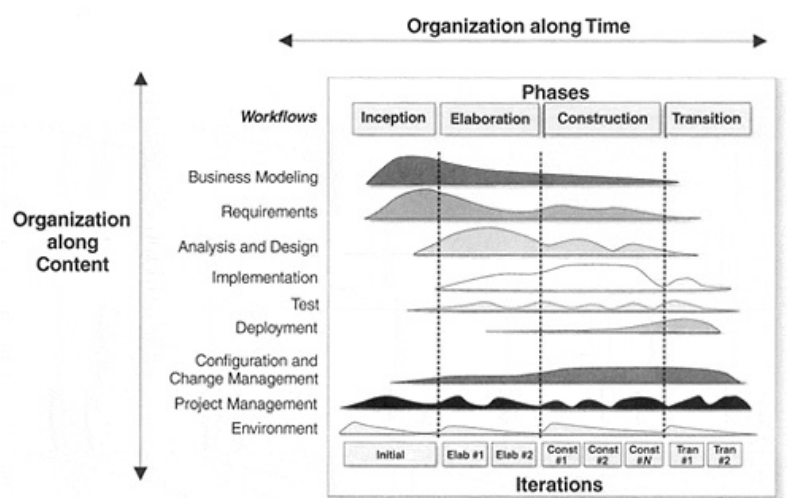
Schattenspiele werden zu Abstraktion realer Begebenheiten genutzt. Der Zuschauer ist dabei gezwungen sich mit der Performance der Künstler auseinander zu setzen, um die Sinnhaftigkeit des Stückes zu verstehen. Im Kopf des Zuschauers bildet sich ein Bild, welches tiefgründiger als die vordergründige Performance ist. Die abstrakte Performance wird somit zu einem fühlbaren Bild des Zuschauers.

Bei einem Beratungsgespräch im Software-Engineering verhält es sich nicht anders. Der Auftraggeber vermittelt seine Vorstellung einer Applikation den Auftragnehmern. Die Auftragnehmer, also die Softwareentwickler, versuchen daraufhin die reale Vorstellung in ein abstraktes Modell für die Applikation zu überführen. Bei dieser Übertragung muss beachtet werden, dass das Ergebnis den Kundenanforderungen entsprechen und zugleich vom Kunden unbeachtete Eigenschaften beinhalten soll.¹

Diese zusätzlichen Eigenschaften, sollen auf die Physiologie, Psychologie und die Handlungsprozesse der Nutzer eingehen und somit unmerklich den Komfort des Nutzers erhöhen.^{2,3} Die zu verarbeitenden Daten und Kommandozeilen, sollen dem Gedächtnis und der Sinnesreize des Menschen angepasst werden. In der Psychologie sind die Gestaltgesetze, d.h. die räumliche Anordnung der Objekte oder der Tiefen- und Bewegungseindruck wichtig. Bei den Handlungsprozessen soll die Applikation auf die

unterschiedlichen Handlungsebenen des Menschen und das Verhindern von Fehlern eingehen. Der Komfort bedingt auch eine hohe Effizienz der Applikation. D.h. der Nutzer ist nicht an einer großen Vielfalt von Funktionen interessiert, wenn er zugleich lange benötigt, um seine benötigte Funktion auszuführen. Die Folge wäre eine geringere Zufriedenheit des Kunden.⁴

Da durch diese vielen Bedingungen eine hohe Komplexität entsteht, dient die abstrakte Modellierung der Strukturbildung.⁵ Entsprechend dem Wasserfallmodell besitzt jedes Entwicklungsprojekt eine Struktur, jedoch nicht zwangsweise repetitive Prozesse. Damit kann die abstrakte Modellierung nicht unbedingt für eine spätere Prozessoptimierung genutzt werden.⁶ Eine Prozessoptimierung wird auch darin erschwert, dass der Kernprozess nicht, wie in Abbildung 3⁷ dargestellt, in der Vorbereitung, Elaboration und Überführung liegt, sondern in der eigentlichen Konstruktion der Applikation liegt.



The Iterative Model graph shows how the process is structured along two dimensions

Der maßgebliche Punkt liegt somit bei dem Entwickler, seiner Erfahrung und der Anzahl an Codezeilen.⁸ Somit konzentriert sich die derzeitige Forschung auf die Sicherstellung einer effektiven Entwicklung, Umsetzung der Anforderungen und späteren Pflege der Applikation.⁹

Die Effektivität der Umsetzung wird im Software Engineering durch die Normierung unterstützt. Hierzu zählen ISO, Rational Unified Process und das Spiralmodell.¹⁰ Obwohl die Prozessmodelle Prototypenentwicklung und Risikomanagement empfehlen, ist zu bedenken, dass sie trotzdem als unzeitgemäß und zu abstrakt kritisiert werden.¹¹ Eine weitere Möglichkeit ist das Top-down- oder Bottom-Up-Verfahren, bei denen dem

Entwickler unterschiedliche Herangehensweisen eröffnet wird.¹² Die Wiedererkennung, d.h., sich wiederholende Prozesse, soll zu Skaleneffekten führen. Die Übernahme von kontrollierten oder automatischen Prozessen wird hierfür empfohlen.¹³

Es ist zu erwarten, dass sich die Komplexität der Programme weiter erhöht.¹⁴ Auch wenn auch MeteorJS als vermutlich zukünftige Entwicklungsplattform für Web-Applikationen die Entwickler gut unterstützt. So werden sich zukünftige Interaktionstechniken, vermutlich als noch größere Herausforderung für den Entwickler herausstellen.¹⁵

■
**Markus Rettenmeier, Student der Wirtschaftsinformatik
an der Friedrich-Schiller-Universität Jena**

-
- 1 Dunn (1993): Software-Qualität, S. 17ff.
 - 2 Vgl. Heinecke (2004): Mensch, Computer, Interaktion
 - 3 Vgl. Zühlke (2012): Nutzergerechte Entwicklung von Mensch-Maschine-Systemen, Kapitel 2
 - 4 Heinecke (2004): Mensch, Computer, Interaktion, S. 32
 - 5 Vgl. Suhr und Suhr (1993): Software Engineering, Oldenbourg, S. 66 ff.
 - 6 Vgl. Becker, Kugeler, Rosemann (2012): Prozessmanagement, S. 20, S. 229ff.
 - 7 Vgl. Rational (2011): Rational Unified Process, S. 3
 - 8 Vgl. Fisher (1990): Produktivität durch Information Engineering, S. 5
 - 9 Vgl. Boehm (1976): Guidelines for Verifying and Validating Software Requirements and Design Specification,
 - 10 Vgl. Pomberger, Pree (2004): Software Engineering, Kapitel 2
 - 11 Vgl. Heinecke (2004): Mensch, Computer, Interaktion, S. 42
 - 12 Vgl. Kurbel (1990): Programmentwicklung, S. 53 f.
 - 13 Vgl. Heinecke (2004): Mensch, Computer, Interaktion, S. 82 ff.
 - 14 Vgl. Knöll, Busse (1991): Aufwandsschätzung von Softwareprojekten in der Praxis, 27 f.
 - 15 Vgl. Zühlke (2012): Nutzergerechte Entwicklung von Mensch-Maschine-Systemen, Kapitel 7

Literaturverzeichnis

- Becker, K. R. (2012). Prozessmanagement. Berlin: Springer Gabler.
- Boehm. (1976). Guidelines for Verifying and Validating Software Requirements and design specifiaction . IEEE Software, S. 20.
- Dunn. (1993). Software Qualität. London: Hanser.
- Fisher. (1990). Produktivität durch Information Engineering. Braunschweig: Vieweg & Sohn.
- Heinecke, A. (2004). Mensch-Computer-Interaktion. Springer.
- Knöll, B. (1991). Aufwandsschätzung von Softwareprojekten in der Praxis. Mannheim: BI Wissenschaftsverlag.
- Kurbel. (1990). Programmentwicklung. Wiesbaden: Gabler.
- O'Dell. (25. Juli 2012). [www.techcrunch.com. Von http://techcrunch.com/2012/07/25/andreessen-horowitz-keeps-eating-the-software-world-with-11-2-million-investment-in-javascript-framework-company-meteor/](http://techcrunch.com/2012/07/25/andreessen-horowitz-keeps-eating-the-software-world-with-11-2-million-investment-in-javascript-framework-company-meteor/) abgerufen
- Pomberger, P. (2004). Software Engineering. München, Wien: Hanser.
- Rational. (Januar 2011). [www.ibm.com. Von https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf](https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf) abgerufen
- Suhr, S. (1993). Software Engineering- Technik und Methodik. München, Wien: Oldenbourg.
- Zühlke, D. (2012). Nutzergerechte Entwicklung von Mensch-Maschine-Systemen. Springer.

Titelbild

<https://www.flickr.com/photos/110751683@N02/13334080323/>
Yuri Samoilov - zuletzt geprüft am 30.11.14 um 17:17Uhr

UNSERE VISION:

WISSENSCHAFTSWIRTSCHAFT



WWW.JENVISION.DE
WWW.WISSENSCHAFTSWIRTSCHAFT.DE
JenVision e.V. - studentische Unternehmensberatung | Carl-Zeiss-Straße 3 | 07743 Jena

