

WILHELM BÜCHNER HOCHSCHULE

DOCTORAL THESIS

Thesis Title

Author:

Marco ISRAEL

Supervisor:

Dr. James SMITH

*A thesis submitted in fulfillment of the requirements
for the degree of Master of Business Administration*

in the

Research Group Name

Department or School Name

17. Juni 2021

Eidesstattliche Erklärung

I, Marco ISRAEL, declare that this thesis titled, „Thesis Title“ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

„Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.“

Dave Barry

WILHELM BÜCHNER HOCHSCHULE

Zusammenfassung

Faculty Name

Department or School Name

Master of Business Administration

Thesis Title

von Marco ISRAEL

Die vorliegende Vertiefungsarbeit überführt Entwicklungsschritte unterschiedlicher Vorgehensmodelle der Softwareentwicklung in eine Sammlung einheitlicher ThinkLets.

Einleitend beschreibt diese Arbeit zunächst die Problemstellung und stellt anschließend das Ziel dieser Arbeit dar.

Der darauf folgende Hauptteil teilt sich in zwei Teile: Im ersten Teil werden die Vorgehensmodelle mit ihren Schritten überfliegend dargestellt und letzteres in Kategorien eingeteilt.

Im zweiten Hauptteil werden die Kategorien mit ihren Schritten in thinkLet Sprache einheitlich neu geschrieben und zusammengeführt.

Diese Arbeit schließt mit einer Schlussbetrachtung und Aussicht ab.

Danksagung

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Inhaltsverzeichnis

Eidesstattliche Erklärung	iii
Zusammenfassung	vii
Danksagung	ix
Glossar	3
1 Einleitung	5
1.1 Die Problemstellung	5
1.2 Ziel dieser Arbeit	5
A Frequently Asked Questions	7
A.1 How do I change the colors of links?	7

Abbildungsverzeichnis

Tabellenverzeichnis

Physikalische Konstanten

Speed of Light $c_0 = 2.997\,924\,58 \times 10^8 \text{ m s}^{-1}$ (exact)

Symbolverzeichnis

a	distance	m
P	power	W (J s ⁻¹)
ω	angular frequency	rad

For/Dedicated to/To my...

Glossar

agile Softwareentwicklung Die agile Softwareentwicklung wird auch als leichtgewichtiger *Softwareentwicklungsprozess* bezeichnet. Anders als in der **klassische Softwareentwicklung** dient lediglich eine Grobe Vorstellung des Auftraggebers als Ausgangspunkt zur Entwicklung. In wiederkehrenden Phasen (die im Kern sehr ähnlich den Phasen der klassischen Softwareentwicklung sein können) entstehen oder verändern sich nacheinander kleine Teile (Inkremente) des Gesamtprojektes, die kontinuierlich dem Auftraggeber zur Verfügung gestellt werden. Das Projekt entsteht und wächst dynamisch entsprechend den Vorstellungen des Kunden. Ziel solcher Vorgehensmodelle ist es, dem Kunden möglichst schnell, wiederkehrende, lauffähige (aber noch nicht vollständige) Softwareteile auszuliefern, die dann als Grundlage für weitere Entscheidungen und so für neue oder geänderte Anforderungen dienen. Beispiele solcher Vorgehensmodelle in der Software Entwicklung sind: *Extreme Programming*, *Scrum* oder *Test getriebene Entwicklung (TDD)*.. 5

klassische Softwareentwicklung Klassische (Vorgehensmodelle der) Softwareentwicklung liegt ein linearer Ansatz zugrunde, mit definiertem Anfangszustand und definiertem Endzustand sowie definierten Zwischenschritten. Dabei baut ein (Zwischen)schritt auf den Ergebnissen des vorherigen Schrittes auf. Ein Schritt muss erst **vollständig** abgeschlossen sein, bevor der nächste Schritt beginnt. Nur in einigen (neueren) Vorgehensmodellen ist Vorgesehen, einen Prozessschritt zurück zu gehen um die Ergebnisse zu überarbeiten. Ein beispielhafter *klassischer* Entwicklungsprozess wäre: Erst die Anforderungen definieren, dann UML Software Modelle erstellen, auf Grundlage dieser Modelle die Softwaremodelle schreiben und zu guter letzt die Software zu testen und ausliefern. Beispiele klassischer Vorgehensmodelle sind das *Wasserfallmodell*, das *Spiralmodell* oder das *V-Modell (XT)* . 3, 5

mathematics Mathematics is what mathematicians do. 6

ThinkLet ThinkLets sind wiederverwendbare Muster oder Bausteine um kreative Gruppenprozesse zu gestalten und zu steuern (moderieren). Diese Bausteine entsprechen Vorgehens- und Verhaltensweisen bzw. -Methoden für kleinste Teilschritte eines gesamten Gruppenprozesses und sind in einer Einheitlichen Sprache (Notation) beschrieben, der s.g. *ThinkLet-Sprache*. Diese Vorgehensbeschreibungen bzw. Verhaltensbeschreibungen legen dar, wie in einzelnen Teile eines gesamten Gruppenprozesses vorzugehen ist, um sichere (qualitativ hochwertige und nahezu reproduzierbare) Teilergebnisse in Gruppen zu generieren und um (Teil-)Ergebnisse einzelner Schritte zielgerichtet weiterzuverarbeiten. Mehrere, aufeinanderfolgende ThinkLets Kombination bilden einen wiederholbaren Gruppenprozess. . 4, 6

ThinkLet-Sprache Die ThinkLet-Sprache definiert, wie ein *ThinkLet* beschrieben werden muss. Hierzu gehört die Beschreibungsstruktur, aber auch die Inhalte, welche ein ThinkLet zu beschreiben hat. Solche Inhalte sind z.B. ein eindeutiger Name, der Sinn (die Aufgabe) dieses ThinkLets, Mögliche Werkzeuge oder Methoden, Eine Vorgehensanweisung (Skript) für Moderatoren oder welche Informationen (Daten) es als Einstieg braucht und welche Ergebnisse es daraus generiert. Die Thinklet-Sprache strukturiert somit in Einheitlicher Darstellung die Antworten auf alle möglichen Arten von W-Fragen (Was, Warum, Womit, Wodurch, Wo, Wohin, ...). 4, 6

Kapitel 1

Einleitung

1.1 Die Problemstellung

In der Entwicklung, hier gezielt in der Softwareentwicklung, haben sich über die Jahre verschiedene Vorgehensmodelle mit teils unterschiedlichen Entwicklungsschritten entwickelt; sei es **agile Softwareentwicklung** oder **klassische Softwareentwicklung** Vorgehen. Unternehmen nutzen diese Modelle, verzichten jedoch zum Teil (nahezu) vollständig auf einzelne Schritte und kürzen ein Modell aus unterschiedlichen, zum Teil gerechtfertigten Gründen, ab. Ken Kent Beck (* 1961, ein US-amerikanischer Softwareentwickler und Berater, sowie einer der Begründer von Extreme Programming (XP) und Autor mehrerer Bücher) weiß, dass Entwicklungsprozesse auf das Projekt und Unternehmen zugeschnitten werden müssen. Die überwiegende Anzahl Vorgehensmodelle gibt solche Flexibilität als Modell nicht her.

1.2 Ziel dieser Arbeit

Ziel dieser Arbeit ist es, die Entwicklungsschritte je Vorgehensmodelle darzustellen und zu kategorisieren. Entwicklungsschritte der Modelle werden sich dabei in Aufgabe oder dem Vorgehen überschneiden. Diese Überschneidungen bilden eine Kategorie oder Gruppe.

Dabei werden die einzelnen Vorgehensmodelle umschrieben und ihre Schritte im Original dargestellt. Der Leser erhält einen Überblick über bearbeitete Vorgehensmodelle in der Software Entwicklung. Für weitere Details einzelner Vorgehensmodelle wird dann auf die verwendete Literatur verwiesen. Ziel dieser Arbeit ist es

nicht, die einzelnen Vorgehensmodelle im Detail neu zu beschreiben.

Im zweiten Hauptteil entsteht aus jeder Kategorie ein **ThinkLet**. Dazu werden die Entwicklungsschritte je Vorgehensmodell und Gruppe in der **ThinkLet-Sprache** neu beschrieben. Es entstehen so einheitlich beschriebene Entwicklungsschritte, die für mehrere Vorgehensmodelle gelten.

Die so entstandene Sammlung von *ThinkLet-Bausteinen* kann verwendet werden, um Entwicklungsprozesse zu definieren, die auf der einen Seite zwar in ihren Schritten selbst standardisiert sind. Auf der anderen Seite kann der Entwicklungsprozess aber dynamisch einem Projekt, einem Unternehmen oder etwa einer Branche angepasst werden. Dabei gilt, dass nicht nur die Entwicklung eines Produktes *agil* sein muss, sondern auch der Entwicklungsprozesse selbst **mathematics** sein darf, solange wie er einheitlich definierten Mustern (thinkLets) folgt.

Oberbegriffe wie *Vorgehensmodelle* oder *Entwicklungsschritte* beziehen sich in dieser Arbeit auf die Software Entwicklung. Das in dieser Arbeit verwendete Vorgehen ließe sich sicher auf andere Entwicklungsbereiche übertragen.

Anhang A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

```
\hypersetup{urlcolor=red}, or  
\hypersetup{citecolor=green}, or  
\hypersetup{allcolor=blue}.
```

If you want to completely hide the links, you can use:

```
\hypersetup{allcolors=.}, or even better:  
\hypersetup{hidelinks}.
```

If you want to have obvious links in the PDF but not the printed text, use:

```
\hypersetup{colorlinks=false}.
```