

## **Praktiken als Voraussetzung zur Verkürzung von Releasezyklen qualitativ hochwertiger Software**

<b>Name</b>	<b>Matr-Nr. / Kennz.</b>	<b>E-Mail Adresse</b>
Bernhard Fleck	0325551 / 937	bernhard.fleck@gmail.com
Claus Polanka	0225648 / 534	e0225648@student.tuwien.ac.at

Status: Abstract

Datum: 14. Dezember 2011

**Inhaltsverzeichnis**

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Related Work</b>	<b>3</b>
<b>3</b>	<b>Research Questions (Best-Practices)</b>	<b>3</b>
3.1	Jahr auf Quartal . . . . .	3
3.2	Quartal auf Monat . . . . .	3
3.3	Monat auf Woche . . . . .	3
3.4	Woche auf Tag . . . . .	3
<b>4</b>	<b>Success Stories (Case Studies falls es welche gibt)</b>	<b>3</b>
<b>5</b>	<b>Schlussfolgerungen</b>	<b>3</b>
	<b>Literatur</b>	<b>4</b>

**Abstract**

Diese Arbeit beschäftigt sich mit der Ausarbeitung von qualitätssichernden Maßnahmen, die es Software-Unternehmen ermöglichen sollen, Deploymentzyklen drastisch zu verkürzen um rasch auf Veränderungen am Markt reagieren zu können. IT-Unternehmen die nur einmal jährlich eine neue Version ihrer Produkte zur Verfügung stellen, könnten künftig große Probleme haben konkurrenzfähig zu bleiben. Diese Arbeit befasst sich daher mit der Frage, welche Entwicklungspraktiken zu einem bestimmten Vorgehensmodell hinzugefügt bzw. von diesem entfernt werden müssen um die Deploymentgeschwindigkeit von jährlichen auf dreimonatige, auf monatliche, auf wöchentliche, auf tägliche und zu guter Letzt auf stündliche Zyklen zu verkürzen. Da sich die Disziplin des Software-Engineerings nicht verallgemeinern lässt, können bestimmte Techniken für bestimmte Deploymentzyklen positive, für andere jedoch negative Auswirkungen haben. Wenn man sich vorstellen würde, dass man zwei Entwicklungsteams, die mit unterschiedlich langen Deploymentzyklen arbeiten, nach ihren Praktiken befragen würde, was würde man als Antwort bekommen? Eines ist klar, Software-Entwickler müssen prinzipiell dieselben Probleme lösen und zwar von der Idee bis zum tatsächlichen Bereitstellen des Produkts für den Endanwender. Nur die Art und Weise wie die Software umgesetzt wird unterscheidet sich dramatisch je nach Länge des verwendeten Deploymentzyklus. In dieser Arbeit wollen wir detailliert darauf eingehen, welche Techniken notwendig sind, um einerseits hohe Qualität der Software zu garantieren und um andererseits die Entwicklungszyklen drastisch zu verkürzen.

## **1 Einleitung**

Allgemeine Einleitung der Thematik. Vielleicht eine Abgrenzung zwischen Continuous Delivery und Continuous Deployment vornehmen.

## **2 Related Work**

Konzepte von Entwicklung und Deployment beschreiben. Die Frage herausarbeiten warum kürzere Releasezyklen notwendig/besser sind. Dabei nicht auf die Qualitätsaspekte vergessen.

## **3 Research Questions (Best-Practices)**

Beschreibt und beantwortet die in Related Work erarbeitete Fragestellung indem Best Practices vorgestellt/herausgearbeitet werden. Weiters wird unser Vorgehen beschrieben. Und zwar wie iterativ eine immer kleinere Kadenz erreicht werden kann. In den Unterkapiteln werden die einzelnen Iterationsschritte vorgestellt.

### **3.1 Jahr auf Quartal**

### **3.2 Quartal auf Monat**

### **3.3 Monat auf Woche**

### **3.4 Woche auf Tag**

## **4 Success Stories (Case Studies falls es welche gibt)**

Um dem Ganzen einen fundierten Unterbau zu geben ein paar Success Stories vorstellen, welche bereits CD erfolgreich einsetzen. Gut wäre es hier natürlich wenn es nicht nur Beispiele aus der Industrie sind, sondern wissenschaftliche Veröffentlichungen.

## **5 Schlussfolgerungen**

Kurze Zusammenfassung der „Ergebnisse“, bzw. Auflisten der Vorteile wenn man sich an die genannten Best Practices hält. Dabei kann man sich auch stark an dem Abstract orientieren.

## Literatur

- [1] Puneet Agarwal. „Continuous SCRUM : Agile Management of SAAS Products“. In: *Proceedings of the 4th India Software Engineering Conference*. 2011, S. 51–60. ISBN: 9781450305594.
- [2] Jasper Boeg. *Priming Kanban*. Trifork, 2011, S. 81. ISBN: 0984521402. URL: <http://www.infoq.com/resource/minibooks/priming-kanban-jesper-boeg/en/pdf/Primingkanban-JesperBoeg.pdf>.
- [3] T Chow und D Cao. „A survey study of critical success factors in agile software projects“. In: *Journal of Systems and Software* 81.6 (Juni 2008), S. 961–971. ISSN: 01641212. DOI: [10.1016/j.jss.2007.08.020](https://doi.org/10.1016/j.jss.2007.08.020).
- [4] Timothy Fitz. *Continuous Deployment*. 2009. URL: <http://timothyfitz.wordpress.com/2009/02/08/continuous-deployment/>.
- [5] Timothy Fitz. *Continuous Deployment at IMVU: Doing the impossible fifty times a day*. 2009. URL: <http://timothyfitz.wordpress.com/2009/02/10/continuous-deployment-at-imvu-doing-the-impossible-fifty-times-a-day/>.
- [6] Jez Humble. *Continuous Delivery vs Continuous Deployment*. 2010. URL: <http://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/>.
- [7] Jez Humble, Chris Read und Dan North. „The Deployment Production Line“. In: *Proceedings of the conference on AGILE 2006*. IEEE Computer Society, 2006, S. 113–118. DOI: [10.1109/AGILE.2006.53](https://doi.org/10.1109/AGILE.2006.53).
- [8] Jez Humble und MoleskyJoanne. „Why Enterprises Must Adopt Devops to Enable Continuous Delivery“. In: *Cutter IT Journal* 24.8 (2011), S. 6–12.
- [9] J Humble und D Farley. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison Wesley Signature Series. Addison-Wesley, 2010. ISBN: 9780321601919.
- [10] Jingyue Li, Nils B Moe und Tore Dybå. „Transition from a Plan-Driven Process to Scrum – A Longitudinal Case Study on Software Quality“. In: *Science And Technology* (2010).
- [11] Fergal Mccaffery u. a. „An Agile process model for product derivation in software product line engineering“. In: *Journal of Software Maintenance and Evolution: Research and Practice* (2010). DOI: [10.1002/smr](https://doi.org/10.1002/smr).
- [12] Subhas Chandra Misra, Vinod Kumar und Uma Kumar. „Identifying some important success factors in adopting agile software development practices“. In: *Journal of Systems and Software* 82.11 (Nov. 2009), S. 1869–1890. ISSN: 01641212. DOI: [10.1016/j.jss.2009.05.052](https://doi.org/10.1016/j.jss.2009.05.052).
- [13] Marko Taipale. „Huitale - A Story of a Finnish Lean Startup“. In: *Lean Enterprise Software and Systems*. Springer Berlin Heidelberg, 2010, S. 111–114.

- [14] Jiangping Wan. „Empirical Research on Critical Success Factors of Agile Software Process Improvement“. In: *Journal of Software Engineering and Applications* 03.12 (2010), S. 1131–1140. ISSN: 1945-3116. DOI: [10.4236/jsea.2010.312132](https://doi.org/10.4236/jsea.2010.312132).