

Praktiken als Voraussetzung zur Verkürzung von Releasezyklen qualitativ hochwertiger Software

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

Status: Abstract

Datum: 20. November 2011

Inhaltsverzeichnis

1	Einleitung (mit Hintergrund, Motivation)	3
2	Beschreibung der Praktiken	3
2.1	Übergang von jährlichen zu dreimonatigen Deployments	3
2.2	Übergang von dreimonatigen zu monatlichen Deployments	3
2.3	Übergang von monatlichen zu wöchentlichen Deployments	3
2.4	Übergang von wöchentlichen zu täglichen Deployments	3
2.5	Übergang von täglichen zu stündlichen Deployments	3
3	Success Stories	3
4	Schlussfolgerung / Ausblick	3
	Literatur	3

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 (mit Hintergrund, Motivation)

2 Beschreibung der Praktiken

2.1 Übergang von jährlichen zu dreimonatigen Deployments

2.2 Übergang von dreimonatigen zu monatlichen Deployments

2.3 Übergang von monatlichen zu wöchentlichen Deployments

2.4 Übergang von wöchentlichen zu täglichen Deployments

2.5 Übergang von täglichen zu stündlichen Deployments

3 Success Stories

4 Schlussfolgerung / Ausblick

Literatur

- [1] AGARWAL, P. Continuous SCRUM : Agile Management of SAAS Products. In *Proceedings of the 4th India Software Engineering Conference* (2011), pp. 51–60.
- [2] BOEG, J. *Priming Kanban*. Trifork, 2011.
- [3] CHOW, T., AND CAO, D. A survey study of critical success factors in agile software projects. *Journal of Systems and Software* 81, 6 (June 2008), 961–971.
- [4] FITZ, T. Continuous Deployment. <http://timothyfitz.wordpress.com/2009/02/08/continuous-deployment/>, 2009.
- [5] FITZ, T. Continuous Deployment at IMVU: Doing the impossible fifty times a day. <http://timothyfitz.wordpress.com/2009/02/10/continuous-deployment-at-imvu-doing-the-impossible-fifty-times-a-day/>, 2009.
- [6] HUMBLE, J. Continuous Delivery vs Continuous Deployment. <http://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/>, 2010.
- [7] HUMBLE, J., AND FARLEY, D. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison Wesley Signature Series. Addison-Wesley, 2010.
- [8] HUMBLE, J., AND MOLESKYJOANNE. Why Enterprises Must Adopt Devops to Enable Continuous Delivery. *Cutter IT Journal* 24, 8 (2011), 6–12.

- [9] HUMBLE, J., READ, C., AND NORTH, D. The Deployment Production Line. In *Proceedings of the conference on AGILE 2006* (2006), IEEE Computer Society, pp. 113–118.
- [10] LI, J., MOE, N. B., AND DYBÅ, T. Transition from a Plan-Driven Process to Scrum – A Longitudinal Case Study on Software Quality. *Science And Technology* (2010).
- [11] MCCAFFERY, F., THIEL, S., RICHARDSON, I., AND O’LEARY, P. An Agile process model for product derivation in software product line engineering. *Journal of Software Maintenance and Evolution: Research and Practice* (2010).
- [12] MISRA, S. C., KUMAR, V., AND KUMAR, U. Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software* 82, 11 (Nov. 2009), 1869–1890.
- [13] TAIPALE, M. Huitale - A Story of a Finnish Lean Startup. In *Lean Enterprise Software and Systems*. Springer Berlin Heidelberg, 2010, pp. 111–114.
- [14] WAN, J. Empirical Research on Critical Success Factors of Agile Software Process Improvement. *Journal of Software Engineering and Applications* 03, 12 (2010), 1131–1140.