

Frequent Releases

Mehrere Inkremente innerhalb eines Sprints ausliefern ?

Entwicklertag Karlsruhe 2017

23.05.2017

Mustafa Yilmaz

mustafa.yilmaz@andrena.de

Christian Süssenbach

christian.suessenbach@andrena.de

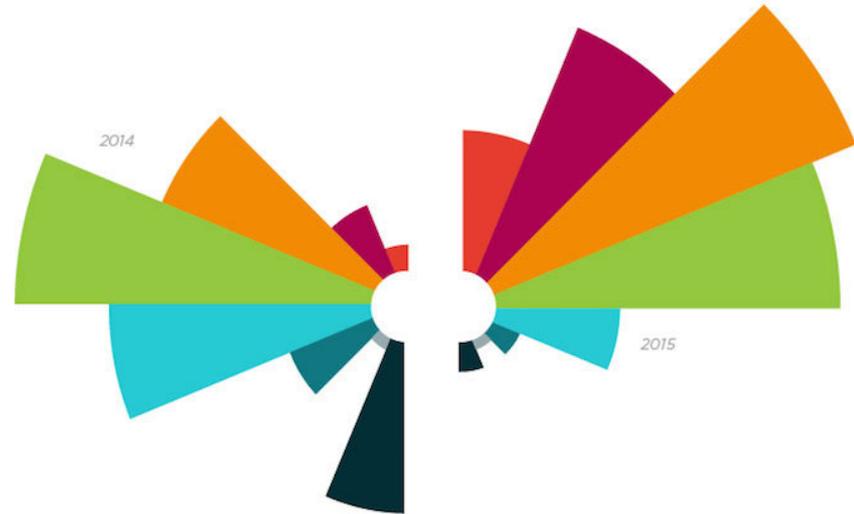


Fast 850 Befragungen

Mehr als 725 Firmen

Firmengrößen von
<100 bis >1000 Mitarbeitern

Erfahrungen von
Softwareentwicklern, Sales,
Marketing, IT Ops, Produktmanager



2014

4% Multiple times a day	7% Daily	17% Weekly	25% Monthly	19% Quarterly	8% Half year	3% Once	13% Never
-----------------------------------	--------------------	----------------------	-----------------------	-------------------------	------------------------	-------------------	---------------------

2015

11% Multiple times a day	19% Multiple times a week	27% Weekly	24% Monthly	10% Quarterly	4% Half year	1% Once	4% Never
------------------------------------	-------------------------------------	----------------------	-----------------------	-------------------------	------------------------	-------------------	--------------------

Frequency of deployments in 2014 and 2015.



Frequent Releases - Mehrere Inkremente innerhalb eines Sprints ausliefern

Im Scrumguide steht, dass das Inkrement eines Sprint sofort released werden kann. In der Realität stellt dies die meisten Unternehmen vor eine große Herausforderung. **Welche Prinzipien und Praktiken helfen, dieses Ziel zu erreichen?** Des Weiteren wird im Scrumguide explizit von mehreren Inkrementen pro Sprint gesprochen.

Was ist, wenn ein Team oder mehrere Teams sogar diese Fähigkeit erreichen, mehrere Inkremente pro Sprint zu releasen?

Steht dies im Widerspruch zu Scrum?

Welchen Sinn hat dann noch ein Sprint Review?

Verlieren die Teams dadurch den Fokus, da eine weitere Störquelle existiert?

Der Vortrag beleuchtet zum einen, welche Prinzipien und Praktiken notwendig sind, um mehrere Inkremente pro Sprint zu liefern, und untersucht, welche Auswirkung das auf die Events, Rollen und Artefakte von Scrum hat.



Inhalt

1. Begriffsklärung – gemeinsames Verständnis
2. Warum benötige ich “Frequent Releases“?
3. Scrum und Frequent Releases – passt das überhaupt zusammen?
4. Welche Werkzeuge benötige ich?
5. Fazit

Was ist Integration, Deployment und Delivery?

Integration

- Der Prozess des fortlaufenden Zusammenfügens von Komponenten zu einer Anwendung.

Delivery

- Die Entwicklungsversion einer Anwendung wird in einen auslieferfähigen Zustand überführt, ein „potentially shippable product increment“

Deployment

- Eine Version einer Anwendung wird auf einer Umgebung bereitstellen.



Was ist Continuous Integration, Deployment and Delivery?

Continuous Integration (CI)

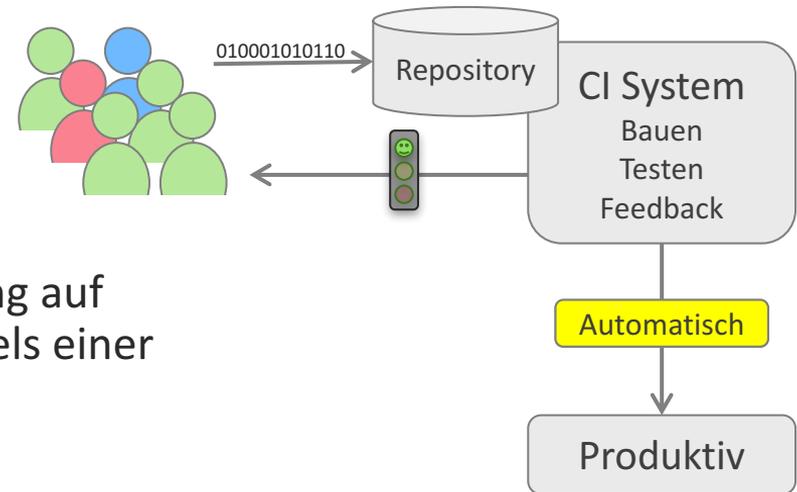
- Automatisches Bauen und Testen der Anwendung mittels eines CI Systems

Continuous Delivery

- Durch eine Entscheidung die Anwendung auf die Produktivumgebung ausliefern mittels einer Deployment Pipeline

Continuous Deployment

- Automatisch und regelmäßig die Anwendung auf die Produktivumgebung ausliefern



Was ist ein Release?

- Eine Business-Entscheidung, so dass der User ein neues (oder mehrere neue) Feature nutzen kann



„Release often, release early“
„and listen to your customers“

Eric S. Raymond

Was sind Frequent Releases?

Continuous oder frequent

Release sobald etwas fertig ist

oder

geplante, regelmäßige, kurzzyklische Releases

Eine (Business-)Entscheidungsfrage

- Wir entscheiden, wann wir unseren Kunden Mehrwert bereitstellen
- Wir entscheiden, wie viele Neuerungen wir unseren Kunden bereitstellen
- Wir entscheiden, wie schnell wir Feedback von unseren Kunden erwarten



Warum sind „Frequent Releases“ sinnvoll?

- Wettbewerbsvorsprung durch kürzere Time2Market
- Frühes Kundenfeedback
- Technische und fachliche Risikominimierung (Überraschungen vermeiden) durch kleine Inkremente neuer Funktionalität
- Vermeidung von „Waste“
- Kosten- und Aufwandsreduktion durch höheren Automatisierungsgrad
- Schnelleres und einfacheres Rollback bzw. Rollforward



Beispiele aus der Praxis

Google

- Mehrere Releases pro Woche

Facebook

- Zwei Releases pro Tag

Amazon

- Änderungen in Produktion alle 12 Sekunden (Stand 2011)

Netflix

- Release-Abstände von Wochen bis 7 Minuten

Spotify

- So viele Releases, dass Release Notes nur noch für große Änderungen gemacht werden



Mindestanforderungen für Frequent Releases

„Marktabsorption“ neuer Releases prüfen

- Möchte der Kunde häufig neue Funktionalität erhalten?
- Ist es möglich den Nutzen (Value) einzelner Funktionalitäten zu messen?

Die (Business-)Entscheidung muss auch getroffen werden

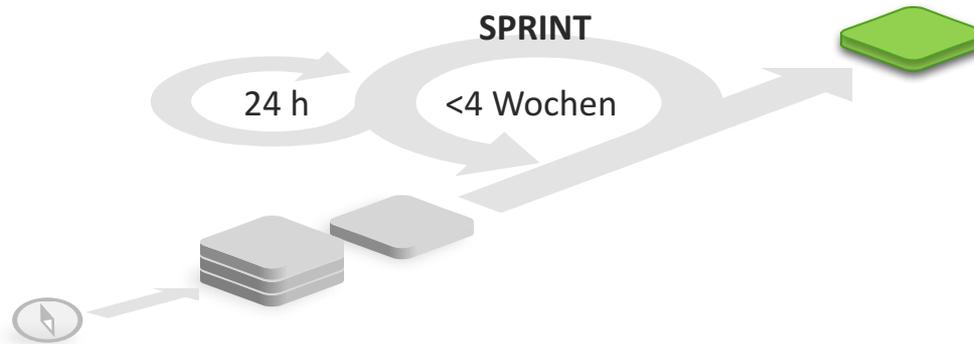
- Möchte das Unternehmen häufig neue Funktionalität liefern?

Hoher Grad an Automatisierung und Professionalisierung erforderlich

- Wer nicht in der Lage ist ein potentiell release-fähiges Produktinkrement am Ende eines Sprints zu liefern, sollte (noch) nicht über Frequent Releases nachdenken!



Scrum und Frequent Releases – passt das überhaupt zusammen?



"Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback." - Scrum Guide



Planning

„The Product Owner discusses **the objective that the Sprint should achieve** and the Product Backlog items that, if completed in the Sprint, would achieve the **Sprint Goal**.“

- Die Art und Weise wie Sprintziele formuliert werden und wie Stories geschnitten werden ist entscheidend
- Gute Beispiele
 - Administrationsbereich für Benutzer ist erstellt
 - Umsatzsteigerung im Online-Shop um mindestens 3 %



Daily Scrum

„... to **synchronize activities** and **create a plan for the next 24 hours**
...done by **inspecting the work** since the last Daily Scrum and **forecasting the work**
that could be done before the next one.“

- Wurde seit dem letzten Daily Funktionalität released?
- Steht ein Release von Funktionalität bis zum nächsten Daily an?
- Was sagen uns Taskboard, Metriken, Log-Files und Nutzerfeedback über die Erreichung des Sprint Goals?



Produktinkrement

„The purpose of each Sprint is to deliver increments of potentially releasable functionality ...“

„This Increment is useable, so a Product Owner may choose to **immediately** release it“

- Das Produktinkrement „wächst“ auch innerhalb des Sprints inkrementell um neue Funktionalität
- Der Product Owner kann sich jederzeit entscheiden das aktuelle Produktinkrement zu releasen



Product Owner

„Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows **what the Scrum Team will work on next**“

- Anwesenheit und Verfügbarkeit wird im Umfeld von Frequent Releases noch wichtiger
- Entscheidet darüber welche Features in ein auszulieferndes Produktinkrement übernommen werden
- Durch Releases im Sprint erhält der Product Owner bereits Feedback vor dem Review, das er in das nächste Review und Planning einbringen kann



Definition of Done (DoD)

„When a Product Backlog item or an Increment is described as ‚Done‘, **everyone must understand what ‚Done‘ means.**“

„Each Increment is additive to all prior Increments and **thoroughly tested, ensuring that all Increments work together.**“

- Ergänzungen für die DoD
 - Ready to Release (für Frequent Releases)
 - Released (für Continuous Deployment)



Review

„During the Sprint Review, the Scrum Team and stakeholders collaborate about **what was done** in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees **collaborate on the next things** that could be done to **optimize value**.“

- Durch ein Release während des Sprints ist es möglich noch früher Feedback durch den Kunden zu erhalten und so den Impact des Releases früher zu beurteilen
- Der Product Owner kann Kunden-Feedback, Nutzungsstatistiken, Umsatz uvm. des vergangenen Sprints dazu nutzen, um Anpassungen am Product Backlog vorzunehmen und sich dafür Feedback von den Stakeholdern einholen
- Verlagerung des **Review-Fokus** auf die **Zukunft** des Produkts



Retrospektive

„Inspect how the last Sprint went with regards to **people, relationships, process, and tools**“

„Create a **plan for implementing improvements** to the way the Scrum Team does its work.“

- Das Entwicklungsteam kann auf die bereits durchgeführten Releases blicken, um Verbesserungspotential ausfindig zu machen und in den Folgesprints umsetzen
- Potentielle Themen: Deployment Pipeline, Metriken, Kundenfeedback, Releasehäufigkeit, ...



Scrum und Frequent Releases – passt das überhaupt zusammen?

"Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback."

- Wir sehen keinen Widerspruch, da Scrum als Rahmenwerk mehrere Releases während des Sprints erlaubt
- Man kann Scrum verwenden um Frequent Releases zu etablieren, aber man muss das Rahmenwerk passend einsetzen
 - Dafür ist Professionalität notwendig



Welche Werkzeuge benötige ich?



Quality & Delivery

Als Entwicklungsteam möchte ich:

- **Eine passende Infrastruktur mit**
 - Versionskontrolle & Virtualisierung
 - Build-Pipelines mit Automatisierung der Tests & des Deployments
- **Eine passende Softwarearchitektur mit**
 - Micro-Services
 - Einfachen Berechtigungskonzepten
 - Feature Toggles/Flags
 - Multi-Versioning



ACHTUNG: Häufige Releases sind keine Lösung für schlechte Qualität



Value

Als Product Owner möchte ich:

- **Eine passende Produkt- und Releasestrategie etablieren**
 - Canary-Releases
 - Automatisierte Rollout- und Rollback-Strategien
 - Releases im Einklang mit der Marketingstrategie
- **Feedbackmechanismen etablieren und nutzen**
 - Stakeholder Feedback während des Sprints, um Erwartungen schnell in die Entwicklung meines Produktes einfließen zu lassen
 - Value-Metriken
 - A/B-Testing
 - Möglichst automatische Log-File Analysen
- **Ein passendes Application Lifecycle Management etablieren**
 - Support, Betrieb, Schulungen, ...



Value

Als Kunde / Anwender möchte ich:

- Immer auf dem aktuellsten Stand sein
- Nicht durch häufige Änderungen an der Software verwirrt werden
- Schnell neue Features testen
- Unterbrechung des Arbeitsflusses soll nicht beeinträchtigt werden
- Selbst entscheiden, wann ich meine Software aktualisiere
- Keinen Aufwand für die Aktualisierung meiner Software haben
- Aktuelle Änderungen kostenlos erhalten
- Eine CD in einer schönen Produktbox im Handel kaufen
- ...



It depends 😊
Ask the customer



Unser Fazit

- **Mindestvoraussetzungen für Frequent Releases**
 - Der **Kunden erwartet**, dass häufig neue Funktionalität geliefert wird
 - Der **Product Owner verfolgt** eine passende Produktstrategie und hat ein passendes Geschäftsmodell
 - Das **Entwicklungsteam sorgt** für einen hohen Grad an Automatisierung und liefert professionell mindestens 1x pro Sprint ein Inkrement in hoher Qualität
 - Die **Organisation ermöglicht** es dem Scrum Team selbstorganisiert zu arbeiten
- **Dabei helfen uns ebenfalls die Scrum-Werte**
 - Offenheit, Mut, Commitment, Respekt und Fokus

Wenn das alles vorhanden ist, dann

Go for it!



Zeit für Fragen

Worüber denken sie gerade nach?

Quellenangaben

- [1] Von Ziko-C - Eigenes Werk, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=3756419>
- [2] Von ThE cRaCkEr - Eigenes Werk, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=20107675>
- [3] New Relic Survey Report 2015 „Data Culture in Software Development“

