

## **Ist das Update schon drauf?**

Dominik Unzicker

(hat Scrum Master bei Jeff Sutherland, Scrum Inc., gemacht)

Poster: The SCRUM Framework (UHP Software)

"We do SCRUM!"

California Agile vs Aggressive Scrum

California Agile: am Ende des Sprints gibt es kein lauffähiges Produkt

Aggressive Scrum: am Ende des Sprints gibt es ein lauffähiges Produkt -> stark fokussiertes Team

Kann man den Sprint in 2-3 Stunden nicht planen, kann man den auch nicht entwickeln (= zu groß, zu viel unklar)

**Grooming:** Userstories durchsprechen, stelle sicher dass die Ready sind, u.a. auch Akzeptanzkriterien  
Userstory hilft sich in den User hineinzusetzen, Akzeptanzkriterien hilft die Umsetzung abzugrenzen  
keine "warten auf API"

### **Definition of Ready:**

Userstories mit Akzeptanzkrit.,

APIs sind fertig

alle Steps müssen drin sein, u.a. inkl. Testing

<http://www.projectcartoon.com/cartoon/2>

### **Planing:**

nur mit Ready Userstory

planningpoker.com

Prio nach Businessvalue (definiert durch PO)

Komplexität schätzen mit Storypoints (Fibonacci-Zahlen) oder auch Tshirt-Größen

mit zunehmender Komplexität wird die Schätzung immer schwieriger

Komplexität größer 20 -> zu komplex

Ziel: kleine Aufgaben im Sprint, die man gut abarbeiten kann um eine gute Pace zu bekommen

### **Defintion Of Done:**

Unit-Tests, Code Reviews, Tests durch QA, Deploy auf Staging

### **Rollen definieren**

WHAT -> PO

HOW -> Team

IMPROVE -> Scrum Master

### **Pizza Team**

maximal so groß, dass das Team von 2 großen Pizzen satt werden -> 2-6 Leute

Kommunikationswege ->  $n(n-1)/2$  --> 2 -> 2, 6 -> 15

neue Devs senken kurzfristig das Lieferniveau

### **Deliver Early, Deliver Often**

im ersten Sprint wenig reinpacken, schauen wieviel wird

Yesterdays weather, Team entscheidet wieviel gemacht wird -> Ownership

alle Prozesse die nicht mit der eigentlichen Entwicklung zu tun haben müssen automatisiert werden

### **Accelerate**

Ergebnis der Retrospektive per Mail verteilen -> jeder ist informiert

was ist das wichtigste um die bestmögliche Beschleunigung im Team zu erreichen -> Kaizen Item as User Story

### **People first**

happy people shjg good results, means happy customers

keine Overtime

Story zwischen rein schieben -> gibt Pattern für sowas, Interrupt-Pattern

## **Keep going**

### **--- Fragerunde.. ----**

PO kann sich weigern Akzeptanzkriterien zu schreiben -> immer wieder in der Retro thematisieren  
ein Scrum Dev Team kann nur ein Thema bearbeiten  
ein Tester kann in mehrere Projekte bzw. Scrum Teams sein  
Klippen im Burnout-Chart sind ggf. Symptom für Testing-Stau  
gewungenermaßen große Teams kann man in mehrere kleine Streams aufteilen, die jeweils ihren Fachbereich bearbeiten und eigene Scrum-Meetings haben