

# Was fehlt Scrum?

The background of the slide is a large puzzle. Most of the puzzle pieces are orange with a red outline and a slight 3D effect. In the center, there is a large, irregularly shaped hole where a piece is missing. The hole is filled with a light blue color, matching the background of the slide.

31. März 2014  
Erich Oswald  
CTO Ergon Informatik AG

# Scrum ist eine Erfolgsstory

Aus der Praxis entstanden

Nachweislich erfolgreich

Gut geeignet für

- komplexe Probleme
- Produktentwicklung
- Herstellen einer Vertrauensbasis

Einfach zu verstehen

... aber schwierig zu meistern



Bild: [scrum.org](http://scrum.org)

# Scrum ist ein Opfer des eigenen Erfolges

## Kommerzielle Interessen



- Zertifizierung
- Coaches
- Autoren
- Scrum Alliance vs. [scrum.org](https://www.scrum.org)

## Ausweitung des Anwendungsbereichs



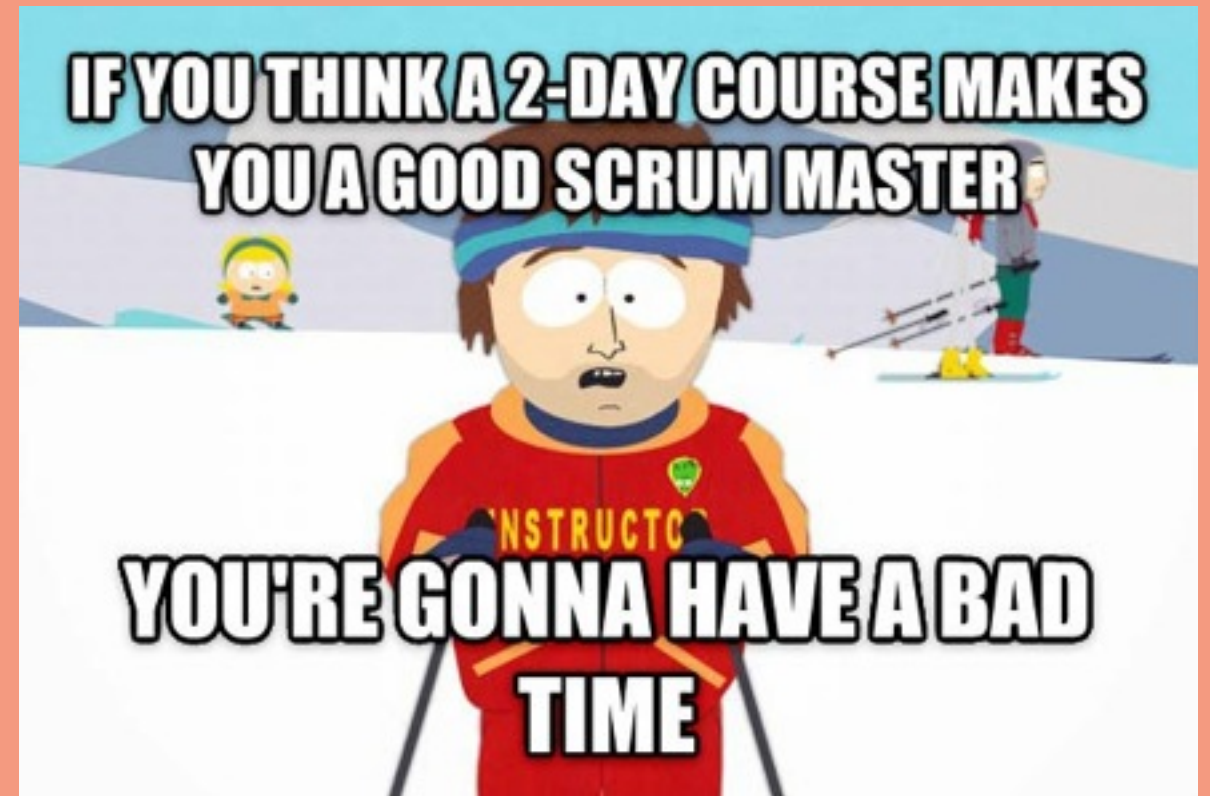
- Projekte
- Off-shoring, Near-shoring, ...
- Mehr als ein Team
- Unternehmungen

# SAFE? DAD?

«Software development needs to be seen as a service and as an ecology not as a manufacturing process»  
— Dave Snowden (#Cynefin)

«I don't believe „Agile“ needs the help of any framework to be scaled. I'd say it is quite the opposite: „scaling“ needs help from „Agile“.»  
— Woody Zuill (#NoEstimates)

# Typische Problemzonen mit



# #1 – Prozessfanatiker, Eiferer, Dogmatiker, Besserwisser

## Symptome

- «Eine Story muss immer Wert für den Anwender haben»
- «Eine Story muss immer die Form „Als X will ich Y damit Z“ haben
- «Am Daily Scrum darfst du nur sagen, was du gemacht hast, was du tun wirst und was dich blockiert»
- «Der PO darf beim Sprint-Planning Teil 2 nicht dabei sein»
- «Im Backlog dürfen keine ungeschätzten Epics sein»
- «Wir dürfen für Story-Points nur die Fibonacci-Reihe verwenden»
- «Die Retrospektive muss gleich im Anschluss an den Sprint-Review folgen»
- «Wir müssen alle Schätzungen mit Planning Poker machen»

# Scrum ist kein Ziel

«Scrum is a cooking pot. You don't eat the pot. You put what you have in the pot. The pot helps you make a meal.»  
— Ron Jeffries

«Scrum is a reference implementation. A basis to optimize from, not a nirvana to optimize to.»  
— Peter Stevens

### The bottom line

If you achieve these you can ignore the rest of the checklist. Your process is fine.

- Delivering working, tested software every 4 weeks or less
- Delivering what the business needs most
- Process is continuously improving

- Clearly defined product owner (PO)
  - PO is empowered to prioritize
  - PO has knowledge to prioritize
  - PO has direct contact with team
  - PO has direct contact with stakeholders
  - PO speaks with one voice (in case PO is a team)

- Team has a sprint backlog
  - Highly visible
  - Updated daily
  - Owned exclusively by the team

- Daily Scrum happens
  - Whole team participates
  - Problems & impediments are surfaced

- Demo happens after every sprint
  - Shows working, tested software
  - Feedback received from stakeholders & PO

- Have Definition of Done (DoD)
  - DoD achievable within each iteration
  - Team respects DoD

### Core Scrum

These are central to Scrum. Without these you probably shouldn't call it Scrum.

- Retrospective happens after every sprint
  - Results in concrete improvement proposals
  - Some proposals actually get implemented
  - Whole team + PO participates
- PO has a product backlog (PBL)

## The bottom line

If you achieve these you can ignore the rest of the checklist. Your process is fine.

- Delivering working, tested software every 4 weeks or less

- Delivering what the business needs most

- Process is continuously improving

- Always end on time
- Team not disrupted or controlled by outsiders
- Team usually delivers what they committed to
- Team members sit together
- Max 9 people per team

### Recommended but not always necessary

Most of these will usually be needed, but not always all of them. Experiment!

- Team has all skills needed to bring backlog items to Done
- Team members not locked into specific roles

- PBL items are broken into tasks within a sprint
  - Sprint tasks are estimated
  - Estimates for ongoing tasks are updated daily

- Velocity is measured
  - All items in sprint plan have an estimate
  - PO uses velocity for release planning
  - Velocity only includes items that are Done

- Team has a sprint burndown chart
  - Highly visible
  - Updated daily

- Daily Scrum is every day, same time & place
  - PO participates at least a few times per week
  - Max 15 minutes
  - Each team member knows what the others are doing

### Positive indicators

Leading indicators of a good Scrum implementation.

- Having fun! High energy level.
- Overtime work is rare and happens voluntarily
- Discussing, criticizing, and experimenting with the process

These are pretty fundamental to any Scrum scaling effort.

- You have a Chief Product Owner (if many POs)
- Dependent teams do Scrum of Scrums
- Dependent teams integrate within each sprint



## #2 – User Stories

Als Besucher möchte ich eine personalisierte Seite sehen

Als Benutzer möchte ich mich einloggen können

Als Entwickler möchte ich einen zentralen Logging-Service

### Stories

- Ersetzen weder Features noch Use Cases noch technische Spezifikationen
- Brauchen keinem Schema zu folgen, aber sollten INVEST-Kriterien erfüllen
- Wer bin ich? Wo bin ich? Was sehe ich? Was tue ich? Was passiert?
- «Story cards are a reminder for a conversation»

# #3 - Backlog-Struktur

Wasserfall

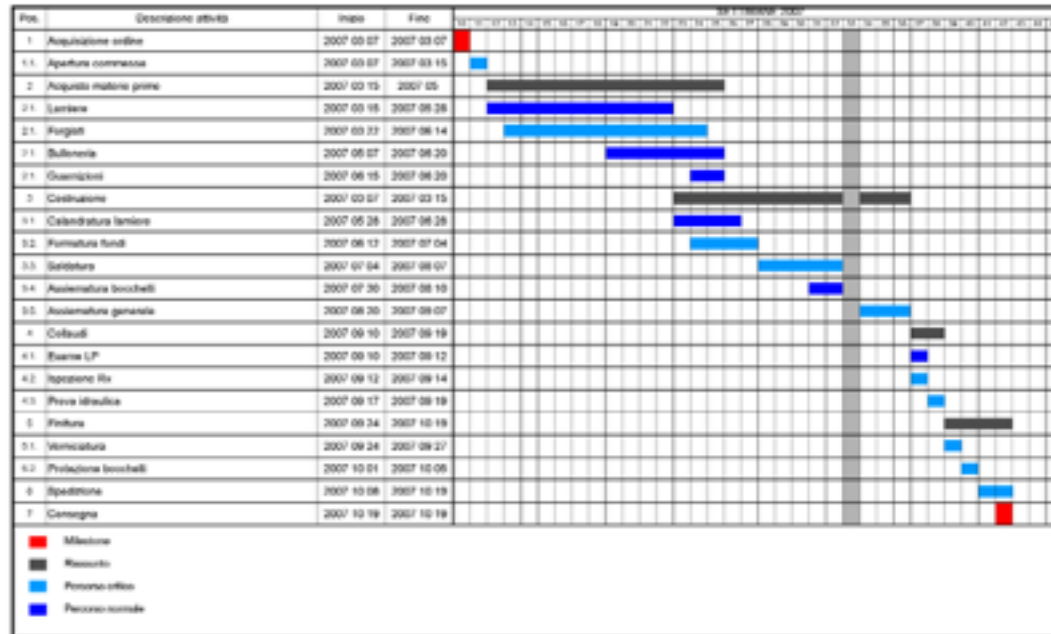
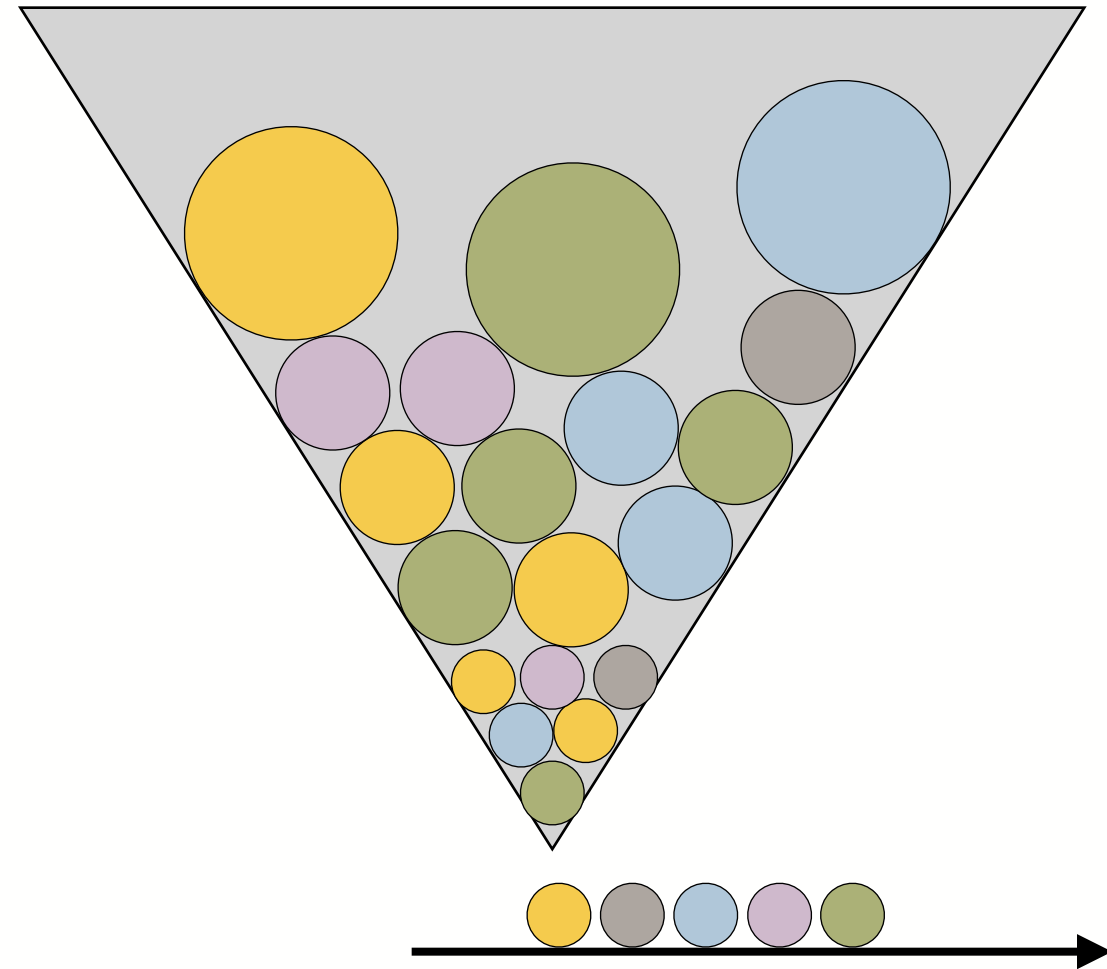


Bild: Ruben Castelnuovo

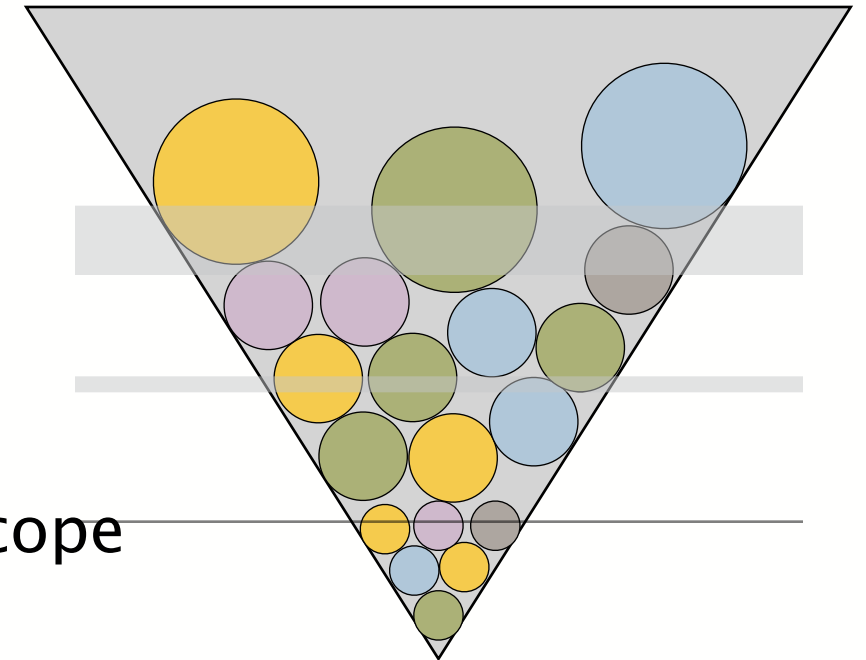
Scrum



## #4 – Schätzungen und Planung

- Fixed Time
- Fixed Budget
- Fixed Scope

- ▶ Schätzungen ermöglichen Entscheidungen
- ▶ Komplexe Probleme haben keinen fixen Scope
- ▶ Herunterbrechen von Epics ist nicht linear
- ▶ Velocity und Storypoints sind keine Hilfe



Planungsunschärfe im Backlog

## #5 – Meetings

### Sprint Planning

- Backlog nicht bereit
- Fokus auf Commitment statt auf Sprintziel

### Retrospektive

- Zuviel Analyse
- Zu wenige konkrete Experimente

### Daily Scrum

- «Gestern war ich an X dran, keine Hindernisse»
- Fokus auf Personen und Aktivitäten statt auf Ergebnissen
- Fehlende Ergebnisse sind automatisch Hindernisse

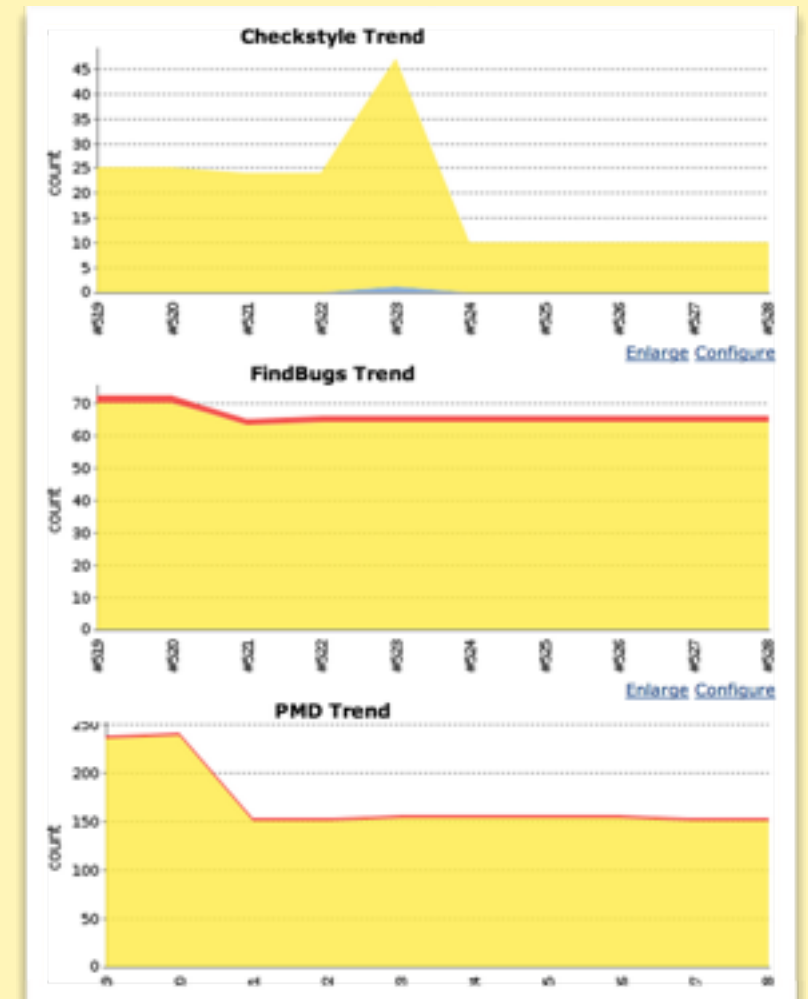
# #6 – Definition of Done

Eine Story/Issue gilt als *done*, wenn

- eine vollständige Implementierung vorliegt
- ein Review durchgeführt wurde
- Tests existieren
- für Features, die Anwenderlogik oder GUI betreffen, eine manuelle Abnahme/Test durchgeführt wurde
- die Benutzerdokumentation nachgeführt wurde (nur für Belimo, nicht für Endkunde) - DEUTSCH, inkl. N+N
- die technische Dokumentation nachgeführt wurde (signifikante architektonische Anpassungen / Erweiterungen)
- Issue resolved und dokumentiert

Oft leidenschaftlich diskutiert  
... auch gelebt?

Lieber realistisch als Idealvorstellung



# Sinnvolle Ergänzungen zu Scrum

«If your code is crap, stickies on  
the wall won't help»  
— Henrik Kniberg

# Sinnvolle Ergänzungen/Alternativen zu Scrum

## Extreme Programming (XP)

- Technische Praktiken
- Clean Code, Software Craftsmanship
- Wartbarkeit und Emergent Design

## Kanban

- Kontinuierliche Prozesse
- Projektarbeit

## Agile Anforderungsanalyse

- Product Canvas
- Impact Mapping
- Story Maps
- Lean Startup

## Agile Dokumentation

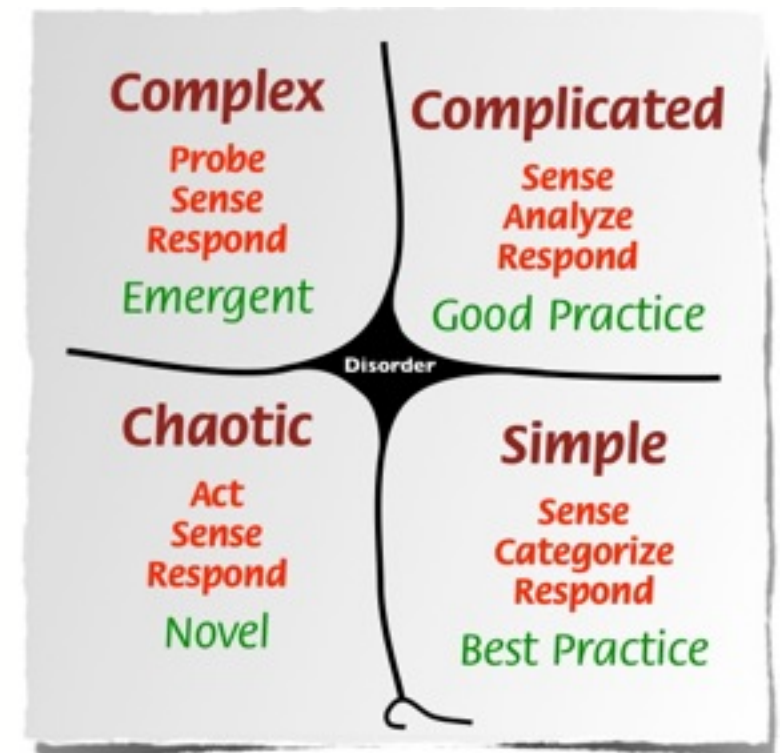
- Produktvision
- Living Documentation/  
Specification by Example

# Scrum braucht kein Phasenmodell

## Risiken mit Phasenmodellen

- Rein inkrementelle statt iterative Umsetzung
- Rückfall in Wasserfall-Muster und - Rollen
- Zuviel Prozess, zu umfangreiche Dokumente, zu viele Handovers
- «Do it right the first time» bei komplexen Problemen unrealistisch

Cynefin Framework





# Fazit

Scrum fehlt nichts. Scrum ist ein guter Hammer, aber die Welt besteht nicht nur aus Nägeln.

Kern der agilen Entwicklung:  
Anforderungen stückeln und priorisieren, regelmässig lauffähigen Code liefern, Prozess kontinuierlich evaluieren und anpassen.