



## Scrum in der SAP-Softwareentwicklung: Business as usual?

Thorsten Deuter, Head of QA & IT, zetVisions AG

Darmstadt, 28. September 2011



---

1 Was macht zetVisions?

---

2 Wie es vorher war

---

3 Wünsch Dir was

---

4 Ein Jahr später

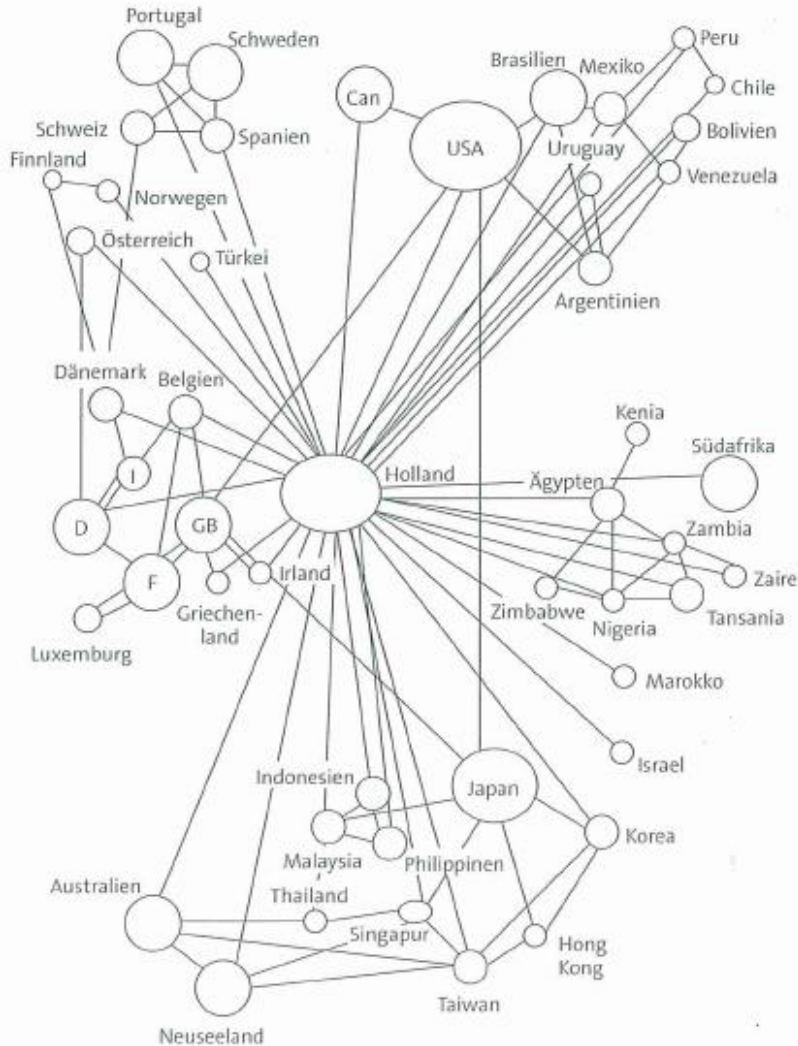
---

5 Quo vadis, Scrum?

---

1

Was macht zetVisions?



Bsp. Konzernstruktur Philips

- National unterschiedliche Rechtssysteme
- **Verschiedene Gesellschaftsformen**
- Unterschiedliche Beteiligungshöhen
- **Unterschiedliche strategische Bedeutung für den Konzern**
- Beteiligungslebenszyklus („junge“ vs. „alte“)
- Wachstumsstrategie (Organisch vs. M&A)
- **Stammdaten und Transaktionsdaten**
- U.v.m....

## Hersteller von Software für das Thema Beteiligungsmanagement

- Mittelständisches Softwarehaus
- Ca. 65 Mitarbeiter
- 400+ Projekte in allen Branchen und Unternehmensgrößen
- 200+ Kunden in ganz Europa (12 der DAX-30-Unternehmen)
- 100.000+ verwaltete Beteiligungen
- 3 Produkte für Beteiligungsmanagement
- Unterschiedliche Technologieplattformen

## Kunden (Auszug)



## Drei Produkte, drei Baustellen, drei Backlogs?

- Organisatorisch – Ehemals klassische Hierarchie:
  - Entwicklungsleiter plus ein Teamleiter pro Produkt
  
- Technisch – Drei Plattformen:
  - Powerbuilder / Oracle / Business Objects
  - SAP NetWeaver 2004s / WebDynpro / SAP BW
  - Java / MS-SQL / Actuate Reporting
  
- Thematisch ähnliche Product Backlogs, tatsächliche Synergie-Effekte gering

## Alle drei Produkte ‚scrumifiziert‘, aber:

- Besondere Herausforderung SAP Add-on „zetVisions CIM“, da:
  - Singuläre Architektur / Technologie
  - Neubildung der Teams, Homeoffices
  - Geringster Reifegrad der drei Produkte
  - Umfangreichstes Product Backlog
  
- → Andere Ausprägung von Scrum
  
- Deshalb hier im Fokus

2

Wie es vorher war

## Releaseplanung

- Releaseumfang durch Produktmanagement und Entwicklungsleiter **festgelegt**  
(grobe Kapazitätsplanung und Aufwandschätzung)
- Planung von vorneherein **sehr knapp kalkuliert**:
  - Releasetermin durch geplante Kundenprojekte vorgegeben
  - Zugesagte Features

## Releaseentwicklung

- **Zentrale** Verteilung von Aufgaben
- **Wenig** Abstimmung zwischen Entwicklern
- Anforderungen wurden dann im Laufe des Entwicklungsprozesses mit dem zuständigen Produktmanager diskutiert
- ‚Private‘ **Excelliste** pro Thema / Entwickler, in der Verbesserungen und Fehler geführt wurden – aber **ständiges Hinzufügen** neuer Punkte
- Stets **Rückläufer** ‚alter‘ Themen
  - Immer **größere** Anzahl ‚unfertiger‘ Themen
- Ausgeprägte **Spezialisierung** der Entwickler → sehr effiziente Entwicklung möglich, aber Engpässe bei Wartung, Krankheit, Urlaub

## Fertigstellung des Release und Auslieferung

- Gegen Ende des Release standen mehrere Testwochen an (durch QA, Support und Consulting)
- Dabei wurde eine große Menge an Tickets zu allen Themen aufgemacht
- In der heißen Phase der Fertigstellung wurde regelmäßig entweder der **Releaseumfang reduziert** und/oder der **Entwicklungsstop verschoben**

## Probleme aus diesem Vorgehen:

- Großer **Arbeitsdruck** aufgrund ‚unfertiger‘ Features
- **Missverständnisse** durch mangelnde Abstimmung
- Abhängigkeiten häufig zu spät erkannt
- Spezialisierung konnte Stillstand erzwingen
- Die Anforderungen änderten sich zu oft
- Klare **Aussage über den Fertigstellungsgrad nicht möglich**, da vieles nur zu 90 % erledigt waren (Letzte 10 % benötigen...)
- Erst sehr spät klar, welche Features tatsächlich fertiggestellt werden können
- **→ Die Qualität des Produktes hat gelitten**

3

Wünsch Dir was

- Im Herbst 2009: Wunsch nach neuer Entwicklungsmethode
- Evaluierung und Entscheidung für Scrum
- Wesentliche Ziele der Einführung waren:
  - Engpässe und Probleme im Prozess früh erkennen
  - Änderungen aufgrund von Kundenwünschen zu ermöglichen
  - Bessere Planbarkeit in der laufenden Entwicklung
  - Verbesserung Informationsfluss
  - Entwickler stärker in den Prozess – auch in Richtung Kunde – einzubinden
  - Know-how streuen
  - Höhere Transparenz, auch für das Management
  - Ständige Verbesserung des Prozesses
  - Letztendlich den Produktwert durch eine höhere Qualität zu steigern

- Wichtig erschien:
  - Ziele klar **formuliert** und **benannt zu haben**
  - Unterstützung des **gesamten** Unternehmens einzuholen und einzufordern
  - Scrum soll sich konkret an der Leistung für das Unternehmen messen lassen
  - Einführung mit **externer** Unterstützung
  - Training nicht nur für Entwickler und Produktmanager, sondern auch für das Managementteam

- Besonderheit: SAP-basiertes System, Entwicklung in ABAP
- Begrenzte Anzahl von Tools zur Unterstützung von Scrum zur Verfügung, (vs. Eclipse), eingesetzte Tools sind:
  - Der **CodeInspector** prüft nach vorgegebenen Metriken die erstellten Entwicklungsobjekte und weist auf problematischen Code hin
  - Mit dem **Unit Test Framework** der SAP können automatisierte Testfälle entwickelt werden, die nächtlich geprüft werden
- Der Build erfolgt nach dem Aktivieren der jeweiligen Entwicklungsobjekte automatisch, kein Build-Tool notwendig
- Eingebaute Quellcodeverwaltung

4

Ein Jahr später

## Engpässe und Probleme im Entwicklungsprozesse früh erkennen

- Entscheidung für kurze Sprints (14-tägige)
  
- Nach wenigen Sprints:
  - Geübter Umgang mit Task Board und Sprintbacklog
  - Störungen im Sprint Burndown bemerkbar
  - Frühzeitiges Gegensteuerung möglich

## Änderungen aufgrund von Kundenwünschen zu ermöglichen

- Jeden Change Request hat der PO in eine User Story umgemünzt
- Story nahm den gewohnten Scrum-Verlauf unabhängig davon, ob aktuelles Release, oder Custom Development
- Wir haben erkannt:
  - Nicht nur **strukturell besserer Prozess** im Custom Development,
  - sondern auch **Schutz** der Entwickler **vor Störungen** der Arbeitsabläufe

## Bessere Planbarkeit in der laufenden Entwicklung

- Bessere Planbarkeit nur möglich, wenn das Product Backlog gefüllt / gepflegt ist
- Pflege des Product Backlogs (inkl. Schätzungen) wurde **Wettlauf gegen die Zeit**
- Erst **spät** Aussagen in Form eines Release Burndowns möglich
- Trotzdem: Zum Release-Ende noch nie so gut „gelandet“
- Hier spielte Scrum eine große **Stärke** aus:
  - **Effektives** Steuerungsinstrument für den PO
  - Burn-Out Phase vor Entwicklungsschluss **blieb aus**
- Großer Optimismus für das nächste Release, das vom gut gepflegten Product Backlog profitieren wird
- Schätzungen waren realistisch, Entwicklungsgeschwindigkeit stabil
- In **Zukunft** frühe Prognosen über den Verlauf der Entwicklung möglich

## Verbesserung Informationsfluss

- Die nicht unübliche **Skepsis** gegenüber den Artefakten von Scrum ist nach einigen Sprints gewichen
- Die ungewohnte Form des Stand-Ups im Daily Scrum ist zur **Selbstverständlichkeit** geworden
- Scheuklappen und Tunnelblick sind Vergangenheit
- Fallende Rate von **Missverständnissen**
- Nebeneffekt (soziale Komponente): **Zusammenwachsen der Teams** und allgemeine Verbesserung des Umgangs miteinander

## Know-How streuen

- Im Umfeld der SAP-Entwicklung **intensive Spezialisierung** (gesteigert durch spezifische fachliche Kompetenz für einzelne Komponenten), Scrum half zu fokussieren:
  - **undurchsichtige**, oft beklagte **Einzelbaustellen** aufzuräumen,
  - durch die Mitwirkung jeden Entwicklers am Gesamtprozess die **Identifikation mit dem Produkt** zu erhöhen
- Know-How Streuung begünstigt **Entlastung durch Verteilung** bei Komponentenspezifischen Peaks
- Die Teams äußern sich darüber hinaus positiv zum entstanden **Team Spirit**

## Höhere Transparenz, auch für das Management

- Kein **Micro-Management**, dafür:
- strukturierter Prozess, **Stories** einzubringen, deren Priorisierung, Abarbeitung und Fortschritt völlig **transparent** sind
- An die Stelle von „Rückwärts-Auswertungen“ tritt der Blick nach vorn:
- Durch Sprintbacklog und Release Burndown: **Status Quo und Prognose über Zielerreichung**
- Management gewann **Sicherheit** in der Kundenkommunikation
- Reviews und Planungen sollten intensiver von ‚Chickens‘ wahrgenommen werden

## Ständige Verbesserung des Prozesses

- **Working Agreements** laufend angepasst (Retrospektiven)
- Weiterhin Potenzial, den Prozess zukünftig zu **verbessern**, auch QA-seitig Raum für Optimierungen
- Große **Themen in naher Zukunft**: TDD, Ausbau der Unittests sowie Pair Programming sein

→ **Scrum hat für uns den Weg in Richtung agiler Techniken geebnet**

## Letztendlich den Produktwert durch eine höhere Qualität zu steigern

- Die **Qualität** der Neuentwicklungen ist erheblich gestiegen
- Immer mehr Qualitätssicherungsmechanismen beschlossen → nach Review wirklich „**potentiell auslieferungsfähig**“
- Im ersten Schritt **Codereview** und Erstellung von **Unittests** vereinbart
- Dann: für jede Userstory eine CCRU-Task (Codereview, CodeInspector, Refactoring und Unittest)

## Weitere qualitätssichernde Maßnahmen sind:

- Durchführung eines Pre-Reviews
- Cross-Tests
- Vor Scrum kein **formalisierter Prozess** zur Abnahme von Features
- Die Formalisierung (klare Definition) war wesentlicher **Fortschritt**
- Das Release enthält nur, was der PO gesehen und für gut befunden hat
- **Wertsteigerung** des Produktes, Entlastung des Support

5

Quo vadis, Scrum?

- Einführung von Scrum bei zetVisions erfolgreich
- Erwartungen und Ziele nicht zu hoch gesteckt
- Befürchtung, Scrum ließe sich in einem ‚konservativen‘ SAP-Umfeld nicht umsetzen, haben wir selbst widerlegt
- Weiteres Verbesserungspotenzial sicherlich vorhanden
  
- **Scrum zeigt an manchen Stellen grundlegende strukturelle Probleme auf, bietet aber keine Lösungen an. Diese zu erarbeiten bleibt in der Verantwortung aller Beteiligten.**

Siehe auch: [http://www.sigs-datacom.de/fileadmin/user\\_upload/zeitschriften/os/2011/05/brandeis\\_deuter\\_OS\\_05\\_11.pdf](http://www.sigs-datacom.de/fileadmin/user_upload/zeitschriften/os/2011/05/brandeis_deuter_OS_05_11.pdf)



**Thorsten Deuter**  
Head of QA & IT  
Scrum Master

**zetVisions AG**  
Speyerer Straße 4  
69115 Heidelberg  
Germany

Tel. +49 (0)6221-33938-643  
Fax +49 (0)6221-33938-922  
thorsten.deuter@zetvisions.com  
www.zetvisions.de

Follow us



SAP, SAP NetWeaver and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serve informational purposes only. National product specifications may vary.

Compliance.  
Excellence.  
Value.