

Scrum für kleine Teams

ProtoSoft AG



Carola Hojnacki

Technische Leitung, Produktmanagement

carola.hojnacki@protosoft.de

Procurat!

Schulverwaltungssoftware für
Schulen in freier Trägerschaft



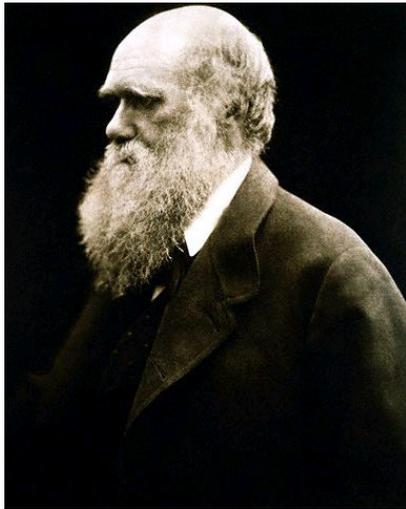
Zahlen und Fakten

- Seit 1998
- 70 Privatschulen, entspricht ca. 250 Anwendern
- 130 Tabellen, 650 Abfragen, 250 Formular, 200 Berichte
- 2 Entwickler

Warum bin ich der
Meinung, dass
SCRUM ein Thema
für die **AEK** ist?

Anforderungen
sind
CHAOTISCH





Anforderungen ÄNDERN sich

Charles Darwin:

*"It is not the strongest of the species that survive,
nor the most intelligent;
it is the one that is most adaptable to changes!"*

Verschwendung
ist
FATAL!



Was ist Scrum ?

- ❖ Einfaches **Framework** (Rahmen) für schrittweise Software-Entwicklung
 - ❖ **Wiederholungen** (Sprints) mit fester Länge (2 bis 4 Wochen)
 - ❖ Kleine, **selbstorganisierende** Teams (bis zu 7 Personen)
 - ❖ Auslieferbares Produkt nach **jedem** Sprint
-

Höhere Kundenorientierung durch

- Start mit der wertvollsten Funktion
- Frühes Kundenfeedback (z.B. durch Demos)
- Enge Zusammenarbeit zwischen PM* und Entwicklung

Flexiblere und schnellere Entwicklung durch

- Kurze Sprints (timeboxed)
- Regelmäßiges Priorisieren der Anforderungen
- Änderungskosten sind geringer weil WIP (Work in Progress) geringer

Bessere Qualität durch

- Kurze Feedback Zyklen
- Permanente Beurteilungs- und Anpassungszyklen
- Continuous Integration und hohe Testautomation

Scrum-Prinzipien

- ❖ Transparenz
 - ❖ Ergebnisorientierung
 - ❖ Priorisierung / das Wichtigste zuerst
 - ❖ Kontinuierliches Beobachten und Anpassen
 - ❖ Timeboxing (feste Zeitfenster)
 - ❖ Dinge abschließen
 - ❖ Teams scheitern nicht
-

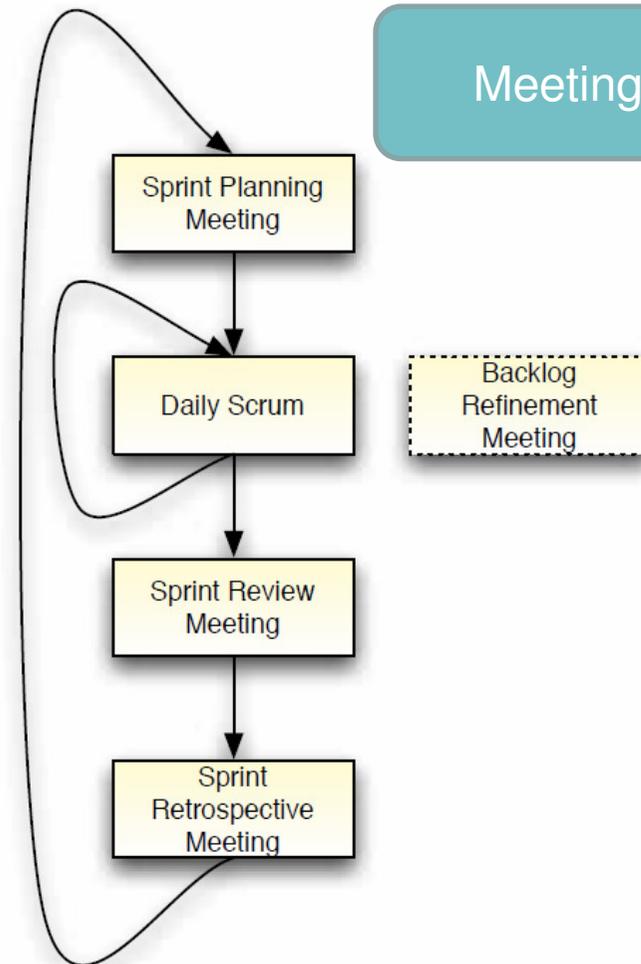
Rollen

- Product Owner
- Entwicklungsteam
- Scrum Master
- Kunde
- (Manager)

Artefakte

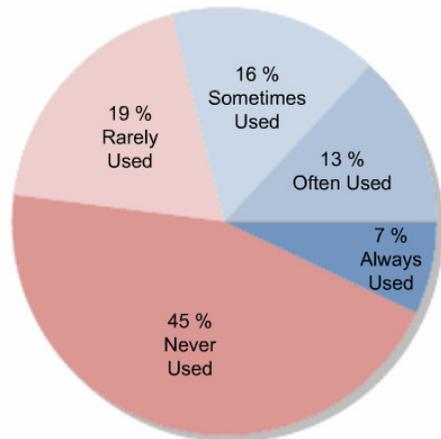
- Product Backlog
- Sprint Backlog
- Sprint Burndown Chart
- Product/ Release Burndown Chart
- (Impediment Backlog)

Meetings

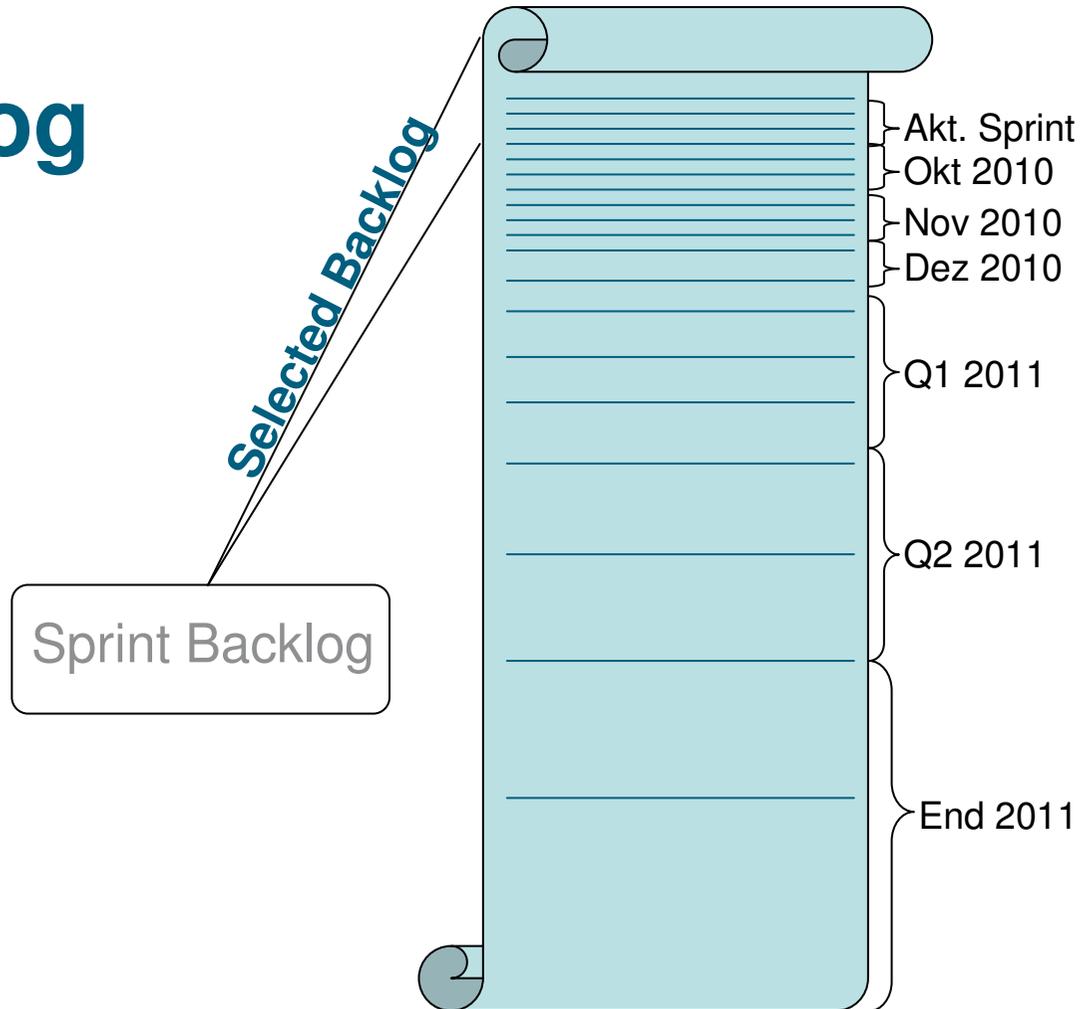


Product Backlog

- ❖ User Stories
- ❖ Priorisiert
- ❖ Ergebnisorientiert
- ❖ Transparent



!



User Story

Eine **User Story** („Benutzergeschichte“) ist eine in Alltagssprache formulierte Software-Anforderung. Sie ist bewusst kurz gehalten und umfasst in der Regel nicht mehr als zwei Sätze.

Als Wer will ich Was, so dass Warum

„Als Consultant will ich bei der AEK vortragen, damit ich berühmt werde.“

„Als Benutzer möchte ich, dass der Login Name mit dem ich mich eingeloggt habe im UI erscheint, damit ich mich als Besitzer mehrerer Accounts schnell orientieren kann.“

Taskeigenschaften

- konkrete Einzelaufgaben einer Story
- Grundlage für das Taskboard
- werden auf Post-it's geschrieben
- nicht länger als einen Arbeitstag
- werden in Stunden geschätzt
- Werden für jeden Kalendertag mit einem roten Punkt gekennzeichnet
- Werden nacheinander abgearbeitet (i.d.R. ein Task pro Entwickler)
- Können im nachhinein gesplittet werden (zeitlichen Aufwand je Task reduzieren.)

Never write a line of code before you have a failing test!

Definition of Done (DoD)

1. Er erfüllt alle (funktionalen und nicht funktionalen) Anforderungen.
2. Er enthält keine bekannten Fehler.
3. Er ist kommentiert und dokumentiert.
4. Er wurde zu zweit entwickelt (pair-programmed) oder reviewed.
5. Die Entwicklung erfolgte testgesteuert.
6. Er benötigt kein Refactoring und muss nicht neu angeordnet werden.
7. Die Entwicklung erfolgte nach bekannter Best-Practice.
8. Die Entwicklung erfolgte nach den definierten Codingstandards.
9. Bei der Kompilierung gibt es keine Fehler oder Warnungen.
10. Er wurde in das Deploy eingebunden und der Buildprozess liefert keine Fehler.
11. Er wurde in die Sourcecodeverwaltung eingecheckt.

Vor dem Sprint

Sprint Planning 1

- Analyse
„Was mache ich?“
- User Stories im PBL gemäß
Priorisierung durcharbeiten
- Aufhören, wenn Sprint voll
(Planning Poker)

Sprint Planning 2

- Design
„Wie mache ich es?“
- In SP1 ausgewählte Stories in
Tasks aufbrechen
- Schätzung aus SP1 bestätigen

Ergebnis

- Sprint Backlog mit User Stories und Tasks
- Architektur der Stories (Pageflows, Domain
Modelle, Sequenz Diagramme, ERM, ...)
- Commitment, Sprintankündigung



Velocity

Entwicklungsgeschwindigkeit =
Anzahl der StoryPoints, die das Team in
einem Sprint umsetzt.

Bestimmung:

- Tatsächliche Geschwindigkeit aus Vorgänger-Sprint (bis Sprint #3)
- Berechnete Geschwindigkeit aus dem Median der Vorgänger-Sprints

Median

- Mittlerer Wert aus einer ungeraden Zahlenreihe
- $(x+y)/2$, wobei x und y die zwei mittleren Werte einer geraden Zahlenreihe bezeichnen.

Planning Poker

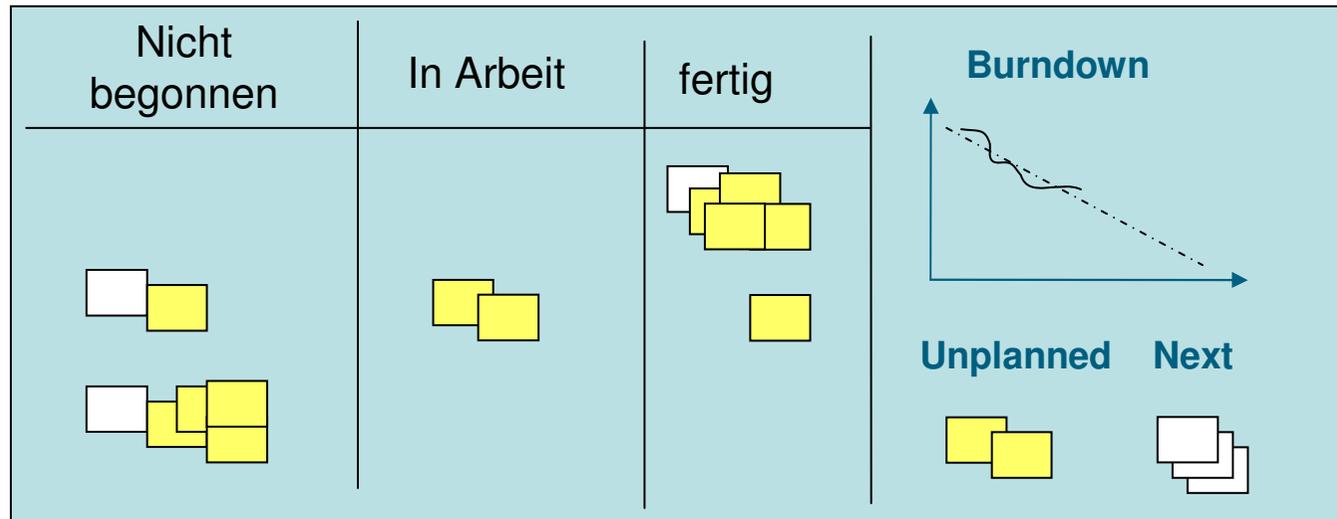
Beteiligte: ProductOwner, Teammitglieder,
End-User/Kunde, ScrumMaster

Dauer: max. 1 Std. (timeboxed)

.....
Regeln:

1. Jeder Entwickler hat ein eigenes Set Karten.
2. Die Story wird kurz von dem beschrieben, der den Inhalt dazu kennt.
3. Die Entwickler einigen sich auf eine Referenz-Backlog-Story der Größe 2.
4. Jeder Entwickler entscheidet sich geheim für die Größe der Story bezogen auf das Referenz-Objekt und legt die Karte bereit.
5. Alle zeigen gleichzeitig ihre Karte.
6. Sind sich die Entwickler bei der Größe einig, wird die nächste Story geschätzt.
7. Bei abweichenden Schätzungen erklären die zwei Entwickler mit der höchsten bzw. niedrigsten Punktzahl ihre Schätzung (max. je 1 min)
8. Punkte 4 – 7 (max. 3x) wiederholen und sich auf eine gemeinsame Größe für die Story einigen.
9. Punkte 3 – 9 für alle Stories aus dem Selected-Backlog wiederholen.

Das Taskboard



... ist das Entwicklung Cockpit

Daily Scrum

- Was habe ich seit dem letzten Daily Scrum erreicht?
- Was plane ich bis zum nächsten Daily Scrum?
- Welche Hindernisse gibt es?

Wichtig:

- Vorbereitung
- Disziplin
- Kurzfassen
- Zuhören



Scrum im Rugby:
Ein angeordnetes Gedränge

Nach dem Sprint

Sprint Review

- Demonstration der Ergebnisse
- Feedback durch Kunden
- Input für nächsten Sprint

Sprint Retrospektive

- Daten sammeln
- Einsichten generieren
- Entscheiden
- Ziele formulieren

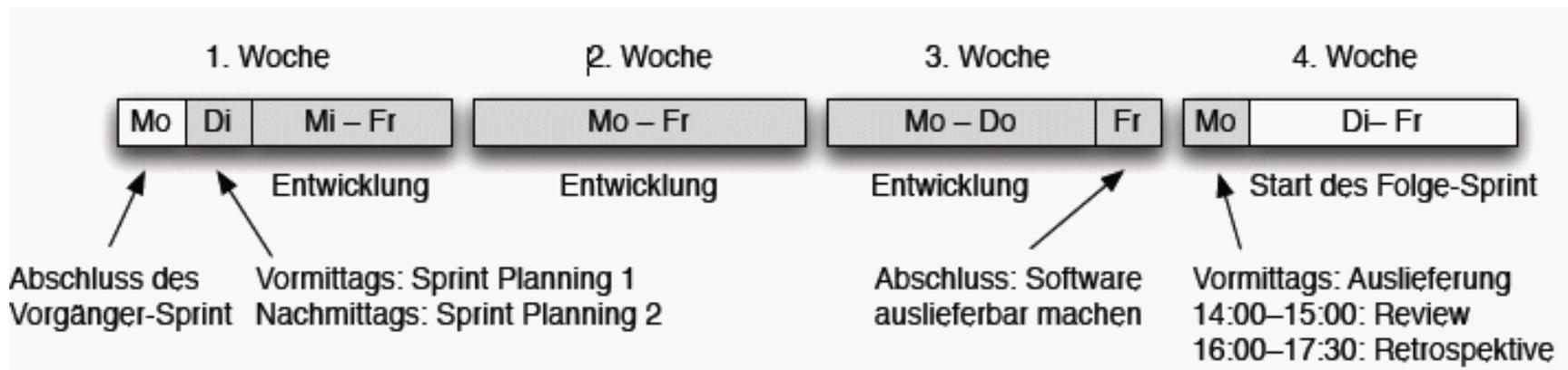


Ergebnis

- Ziele definiert
- Aktionen festgelegt
- Verantwortliche benannt

Der Sprint

Beispiel: 3 Wochen – Sprint

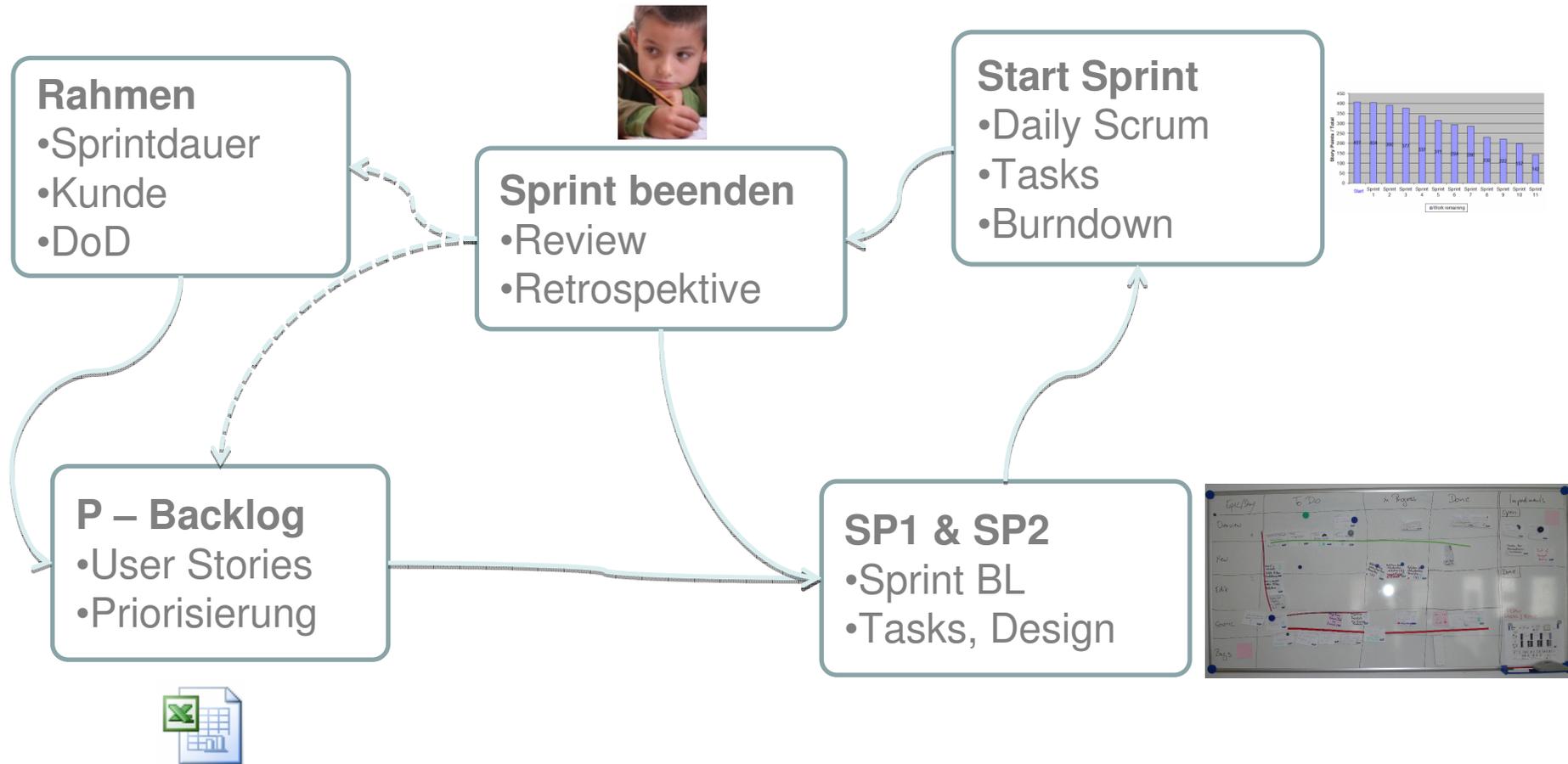


Nach Ralf Wirdemann: Scrum mit UserStories

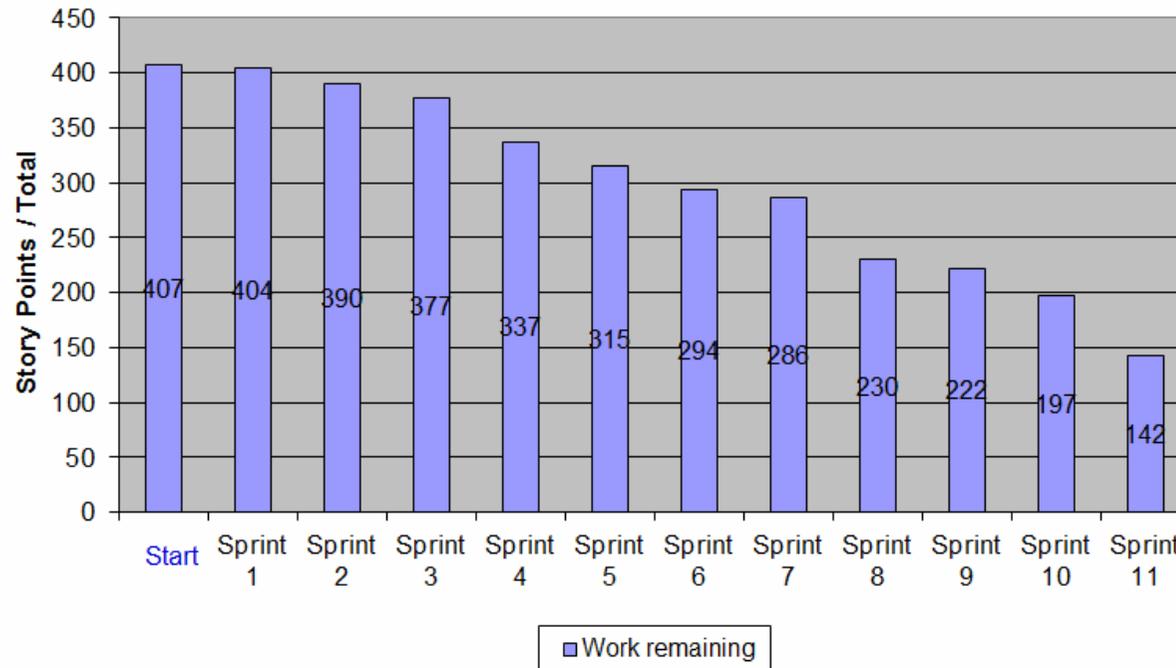
Solo Scrum

- einfache Vorgehensweise, die auch Einzelkämpfer zwingt, zur richtigen Zeit das richtige zu tun
 - Systematik zur zielgerichteten, priorisierten Arbeitsweise (keine goldenen Henkel!)
 - Zeit für Reflektion, Priorisierung, Fokussierung
 - Unterschiedliche Rollen bewusst spielen
 - Rituale
 - Höhere Effizienz
-

Und jetzt? ... einfach ausprobieren!



Burndown Diagramm



- Produkt
- Release
- Sprint

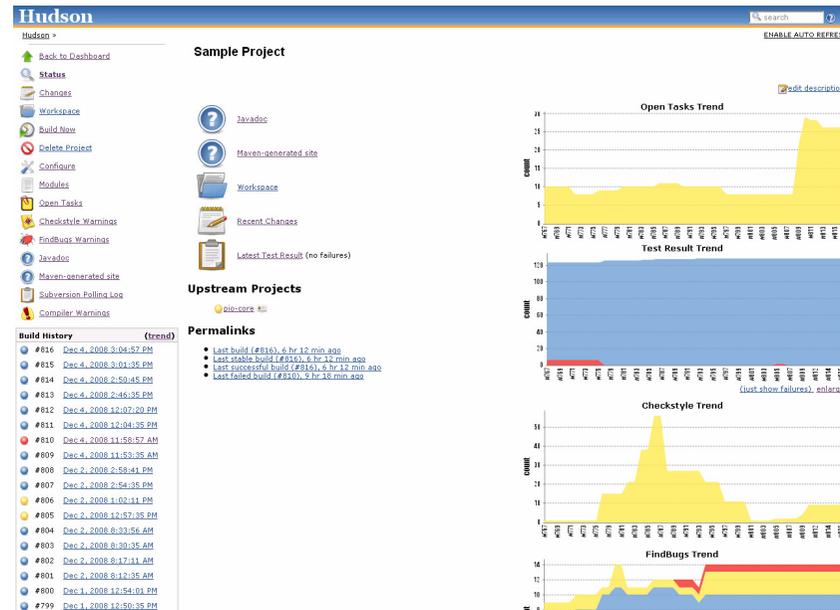
Großer Mangel bei Access

Bisher keine
Testautomatisierung
möglich

Die Lösung

der Vortrag von Paul Rohorzka

Pragmatisches Testen von Access-
Anwendungen



Tools für Scrum

- ❖ Whiteboard und Post-It's
 - ❖ Excel
 - ❖ Beispieldatei wird zum Download bereitgestellt
 - ❖ Greenhopper
 - ❖ Jira – Plugin
 - ❖ Zur Zeit 10\$ bzw. 20\$ mit Jira (bis 10 User) pro Jahr
 - ❖ <http://www.atlassian.com/software/greenhopper/>
 - ❖ Agilefant
 - ❖ Open Source
 - ❖ <http://www.agilefant.org/>
-



Excel

Product Backlog

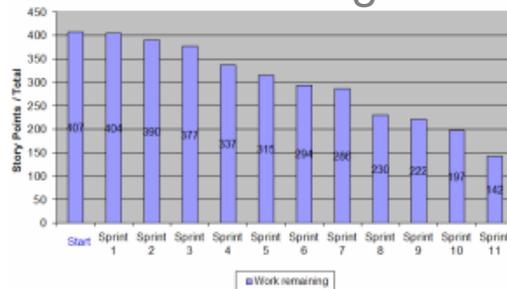
ID	Sprint	Priority	Name	Name (englisch)	Story	Story (englisch)	Acceptance Criteria
myPE_0023	1	4	Erstellungsgeschicht verwalten	create history	Als Medialer-Developer möchte ich die neuen Features entwickeln in einer komfortablen Entwicklungsumgebung, damit ich produktiv an der Medialer-App arbeiten kann.	Als neues Feature Developer kann ich es komfortabel entwickeln und es in der Medialer-App implementieren.	1) Ich als Entwickler kann die Story in der Medialer-App erstellen. 2) Die Medialer-App zeigt die Story in der Medialer-App. 3) Die Medialer-App zeigt die Story in der Medialer-App. 4) Die Medialer-App zeigt die Story in der Medialer-App.
myPE_0017	1	2	MediaStore anzeigen	display MediaStore	Als Publisher möchte ich eine MediaStore haben, um meine Medien zu verwalten.	Als Publisher kann ich eine MediaStore in der Medialer-App erstellen und sie anzeigen.	1) Die MediaStore ist in der Medialer-App. 2) Die MediaStore ist in der Medialer-App. 3) Die MediaStore ist in der Medialer-App. 4) Die MediaStore ist in der Medialer-App.
myPE_0010	1	3	MediaStore Code kopieren	copy MediaStore Code	Als Publisher möchte ich den Code der MediaStore kopieren können, um ihn in meiner App zu verwenden.	Als Publisher kann ich den Code der MediaStore in der Medialer-App kopieren.	1) Der Code wird in der Medialer-App kopiert. 2) Der Code wird in der Medialer-App kopiert. 3) Der Code wird in der Medialer-App kopiert.
myPE_0019	1	4	App mit statistischem Banner	app with statistical banner	Als Publisher möchte ich eine App haben, die dem Benutzer statistische Informationen über die App zeigt.	Als Publisher kann ich eine App mit statistischen Informationen erstellen.	1) Der Benutzer kann die App öffnen. 2) Der Benutzer kann die App öffnen. 3) Der Benutzer kann die App öffnen.
myPE_0022	2	5	Dufock App Settings anzeigen	display default app settings	Als Publisher möchte ich mir die Default Settings der App anzeigen lassen können.	Als Publisher kann ich die Default Settings der App anzeigen lassen.	1) App kann 0..n Settings haben (einen Key). 2) App zeigt Default Value für Settings.



Sprint Backlog

Sprint Nr.	Dauer	Committed	Mobile Management	Settings Management	Mobile Website Integration
Sprint #01	1.12.2010 - 15.12.2010	100%			
Sprint #02	16.12.2010 - 20.12.2010	100%	myPE_0017	myPE_0010	
Sprint #03	21.12.2010 - 25.12.2010	100%	myPE_0019	myPE_0022	
Sprint #04	26.12.2010 - 30.12.2010	100%	myPE_0023	myPE_0010	myPE_0019
Sprint #05	31.12.2010 - 4.1.2011	100%	myPE_0017	myPE_0010	myPE_0019

Burndown Diagramm



Agilefant (OpenSource)

Product Backlog

Project: Version 1.0
Backend server > Version 1.0

Info Actions

Name	Version 1.0
Reference ID	backlog3
Start Date	2010-03-28 07:00
End Date	2011-03-16 15:00
Planned Size	400h
Baseline load	—
Assignees	OC, AP, JA, DR
Description	(empty)

Story tree Leaf stories Iterations

Story tree Filters Expand all Create story

- Version 1.0 > Sprint 2 10 Create the navigation system in supermarket
- Version 1.0 > Sprint 1 10 Feature X
- Version 1.0 > Sprint 1 8 Feature Y
- Version 1.0 > Sprint 1 20 Feature Z



Sprint Backlog

Iteration: Sprint 1
Backend server > Version 1.0 > Sprint 1

Info Workload History Actions

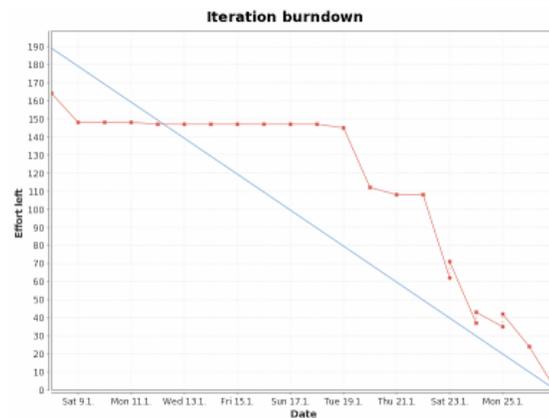
Name	Sprint 1
Reference ID	backlog4
Start Date	2010-06-26 15:00
End Date	2010-07-26 15:00
Planned Size	—
Baseline load	—
Assignees	(none)
Description	(empty)

Effort left 40h
Original estimate 11h
Story points 38
Spent effort 3h
Velocity / day
Stories done 0% (0 / 3)
Tasks done 67% (2 / 3)

Stories Show tasks Create story

#	Labels	Name	Points	State	Responsible	Σ(EL)	Σ(OE)	ES	Edit
1		Feature Z	20	Not Started	AP, demo, DR	40h	11h	3h	Edit
2		Feature Y	8	Not Started	(none)	—	—	—	Edit
3		Feature X	10	Not Started	(none)	—	—	—	Edit

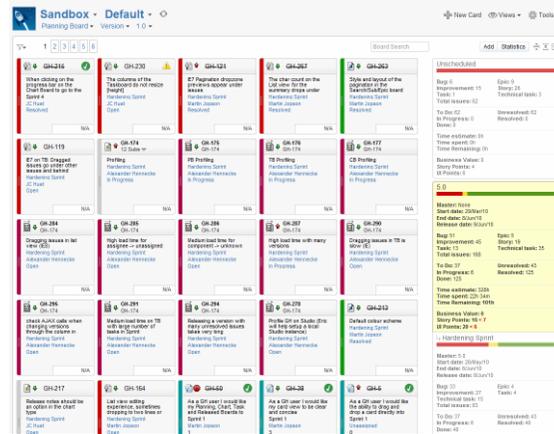
Tasks without story Create task



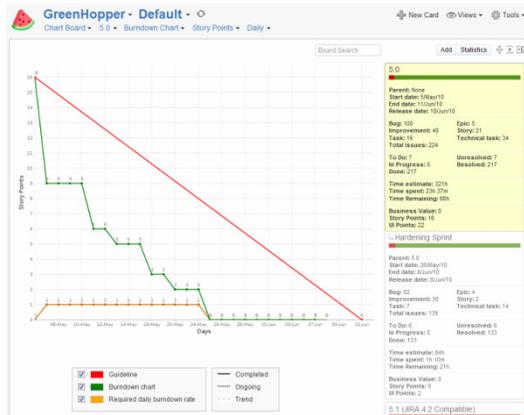
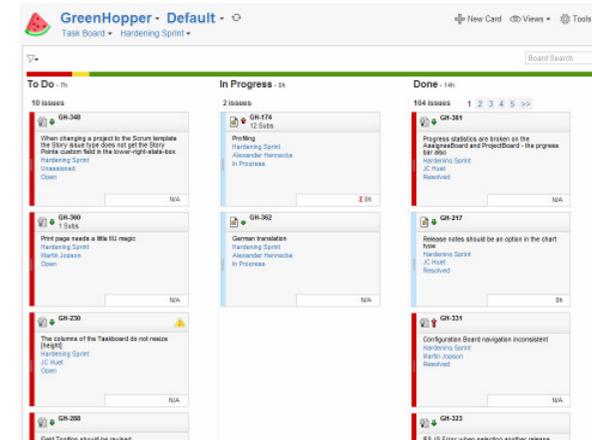
Burndown Diagramm

Greenhopper (based on Jira)

Product Backlog



Sprint Backlog



Burndown Diagramm

Literatur und Links

- Boris Gloger: Scrum



- Ralf Wirdemann: Scrum mit User Stories



- Henrik Kniberg: 10 Ways to screw up with Scrum and XP

<http://www.parleys.com/#id=36&sl=2&st=5>

Have Fun with Scrum

Unter www.dilbert.com gibt es zu den Stichwörtern Scrum und Agile viele wunderbare Comic-Strips!

Begriffsdefinitionen:

- Backlog: Arbeitsrückstand (To-Do-Liste)
 - To burn down: abbrennen
 - Task: Aufgabe
 - Taskboard: Aufgabenwand
 - DoD: Definition of Done = Def. für FERTIG
 - Comitment: Selbstverpflichtung
 - Velocity: Geschwindigkeit
 - Review: Besprechung / Präsentation
 - Retrospektive: Rückschau
 - Sprint: Entwicklungszyklus mit fester Dauer
-