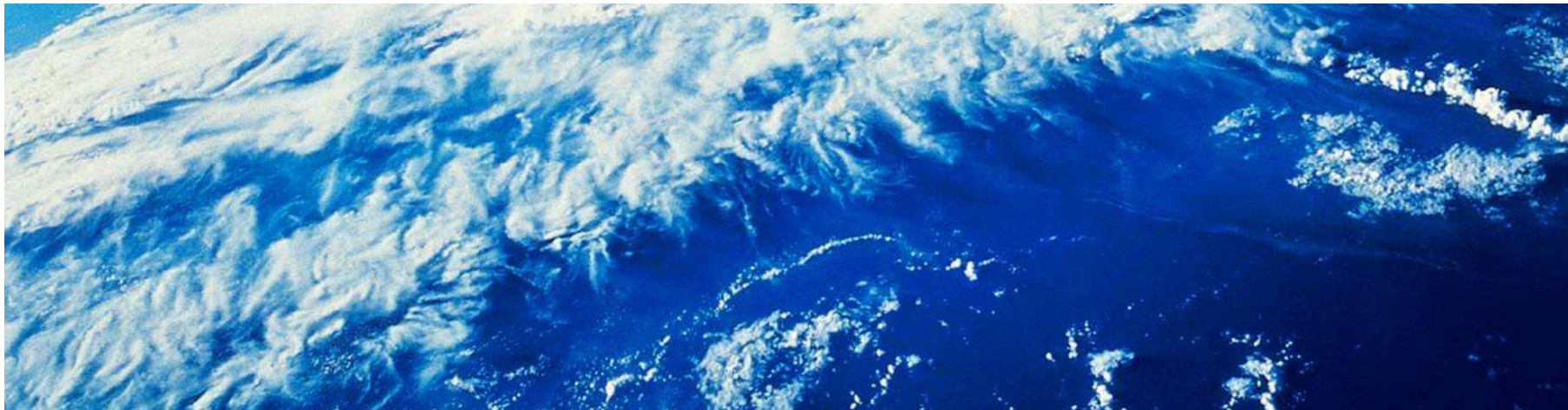


**Telefonkonferenz am 07.05.2010**



# **Rational Team Concert mit der JAZZ Plattform für agile Softwareentwicklung**



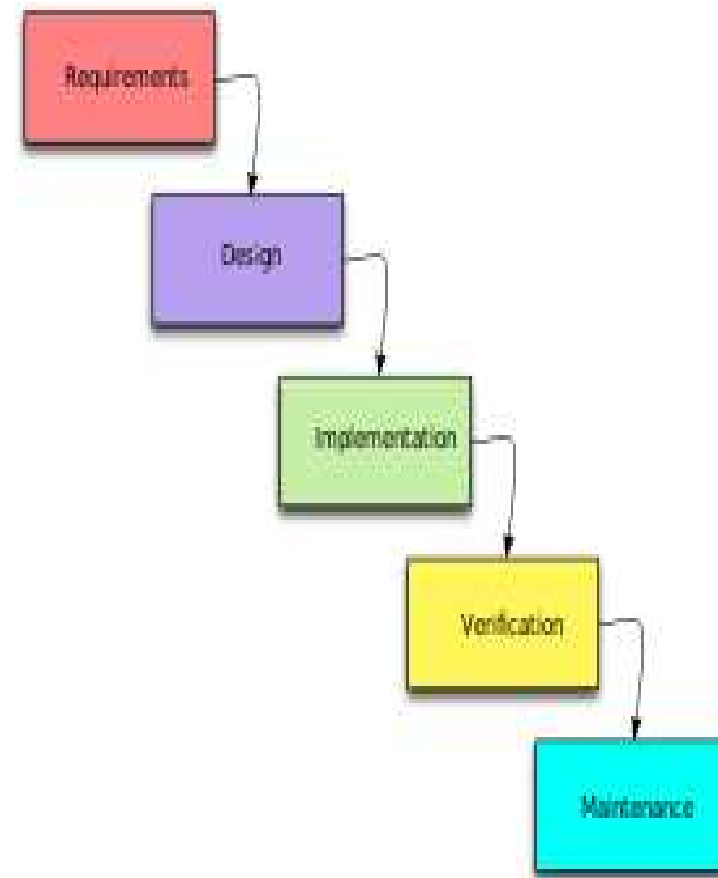
## Agenda

- **Einführung in SW Entwicklungsprozesse**
- **Agile SW Entwicklung**
- **SCRUM**
- **Agile Entwicklung mit Rational Team Concert**



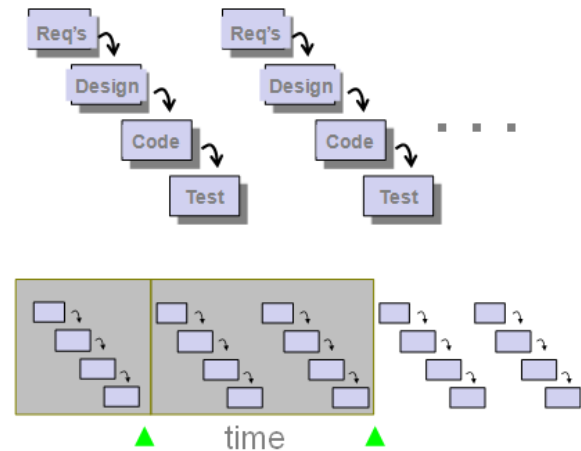
# Das klassische Wasserfallmodell

- Pros:
  - einfach, wohldefinierte Meilensteine, diszipliniert, geordnet
  - sehr verbreitet
- Cons:
  - Das Risiko wird erst am Ende aufgelöst
  - Was, wenn die Architektur nicht funktioniert?
  - Was, wenn der System Test kritische Fehler oder fehlende Anforderungen aufzeigt?



# Der iterative Prozess

- Ebenfalls wohl definiert
- Risiko löst sich sehr früh auf
- Die SW wird in zeitlich begrenzten Iterationen entwickelt
- Beispiele:
  - ▶ Rational Unified Process (RUP), Spiral, MSF, OpenUP



## Agenda

- Einführung in SW Entwicklungsprozesse
- Agile SW Entwicklung
- SCRUM
- Agile Entwicklung mit Rational Team Concert





## Manifesto for Agile Software Development

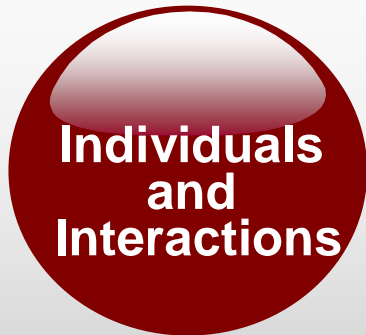
We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

## Reaktion auf Prozessbürokratie



**over  
processes and tools**



**over  
comprehensive documentation**



**over  
contract negotiation**



**over  
following a plan**

**Manifesto for Agile Software Development**

## Agile Praktiken

- Iteratives und inkrementelles Entwickeln
- Zweistufiges Planen
- Häufige Releases
- Selbst organisierende Teams
- Andauernde Kundenbeteiligung
- Zeitnahe Anforderungsdefinition
- Ständiges Testen
- Laufende Integration
- Refactoring
- Unit testing & Test Driven Development (TDD)
- Retrospektiven





## Agenda

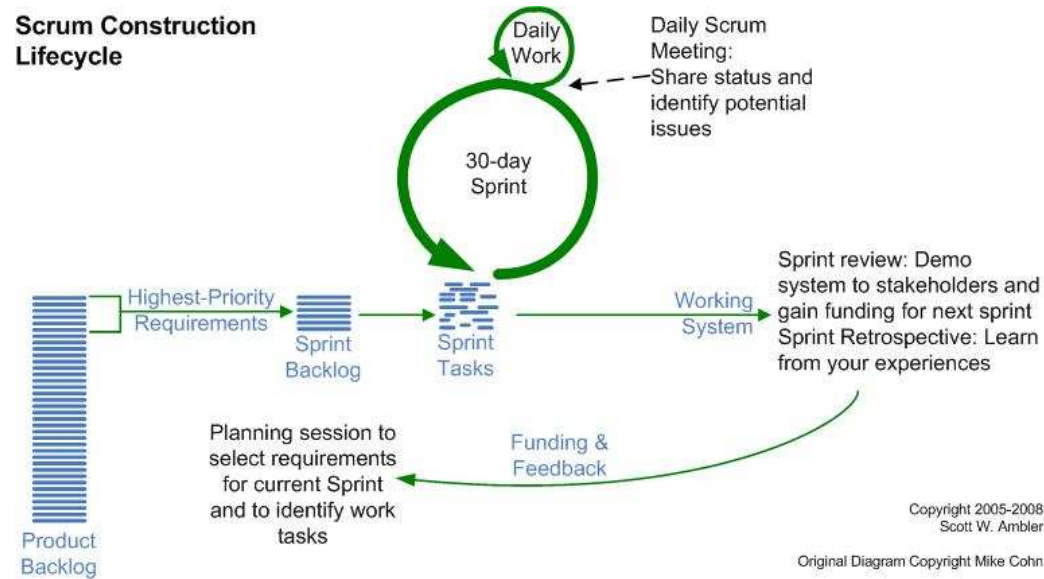
- Einführung in SW Entwicklungsprozesse
- Agile SW Entwicklung
- **SCRUM**
- Agile Entwicklung mit Rational Team Concert



## Scrum Roles

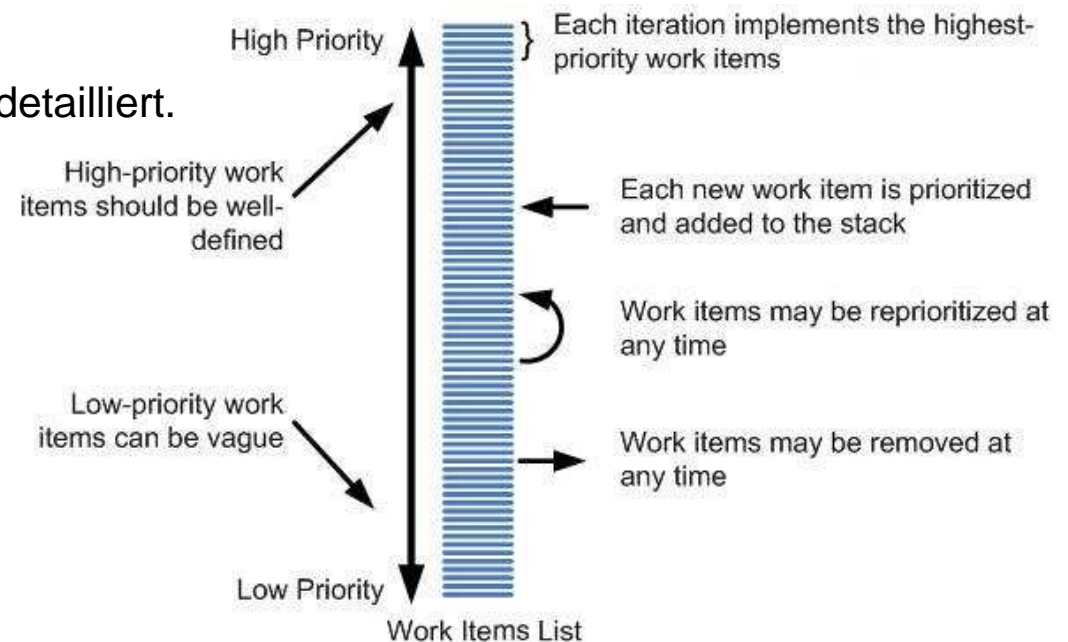
- Scrum Master
  - Member of the team
  - Leader/coach
  - Facilitates scrum meetings such as sprint planning, sprint demo, and daily Scrum meetings
- Product Owner
  - Represents the stakeholders
  - Owns the product backlog, including the requirements
  - Provides information and makes decisions in a timely manner
  - “The one neck to wring”
- Developer/Team Member
  - Anyone else on the team

# The Scrum Life Cycle



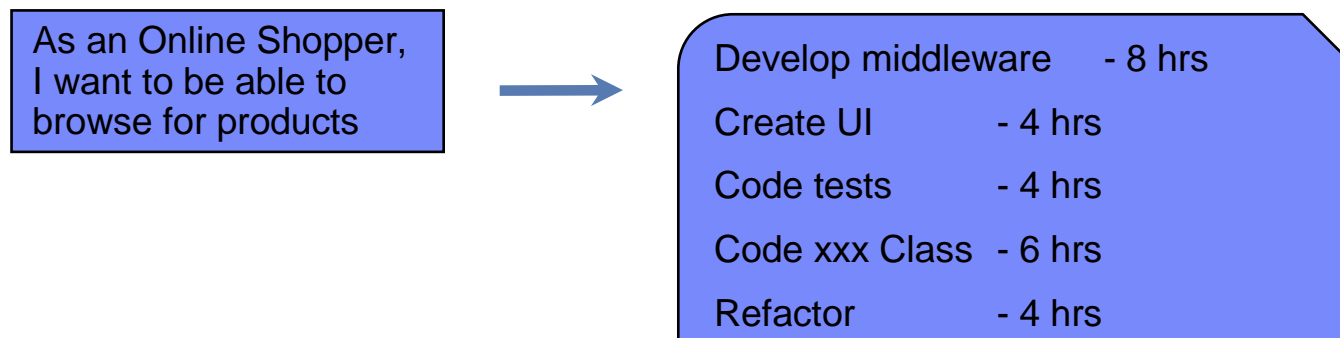
## Scrum - Der Product Backlog

- Enthält alle nicht verplanten Storys des Projektes
- Geschätzt vom Team in relativen Größen (Story Points)
- Die Punkte werden im Backlog vergeben je nach Größe und Schwierigkeit in Relation zu den anderen Backlog Items
- Vorteil: man benötigt nicht alle Requirements zu Anfang
- Erst im Sprint Planning Meeting findet eine zeitliche Festlegung statt
- Die Requirements werden just-in-time detailliert.



## Scrum - Sprint Backlog

- Detaillierter Iterations Plan, erzeugt vom Team, welches die Items aus dem Produkt Backlog auswählt.
- Alle Product Backlog Items werden in Aufgaben (Task) heruntergebrochen, die nötig sind, um das Item zu erfüllen.
- Gemeinsame Schätzung des Aufwandes in Stunden (max. 16Std.)
- Die einzelnen Teammitglieder wählen selber die Tasks aus, die sie erledigen möchten
- Die verbleibende Zeit für eine Task wird vom Teammitglied täglich aktualisiert





## Daily Scrum Meetings

- Ziel: Informationsaustausch
- 15min
- Beantwortung von 3 Fragen:
  - ▶ Was habe ich getan?
  - ▶ Was plane ich zu tun?
  - ▶ Welche Hindernisse gibt es?
- Keine Problemanalysen!

## Agenda

- Einführung in SW Entwicklungsprozesse
- Agile SW Entwicklung
- SCRUM
- **Agile Entwicklung mit Rational Team Concert**



## Agenda

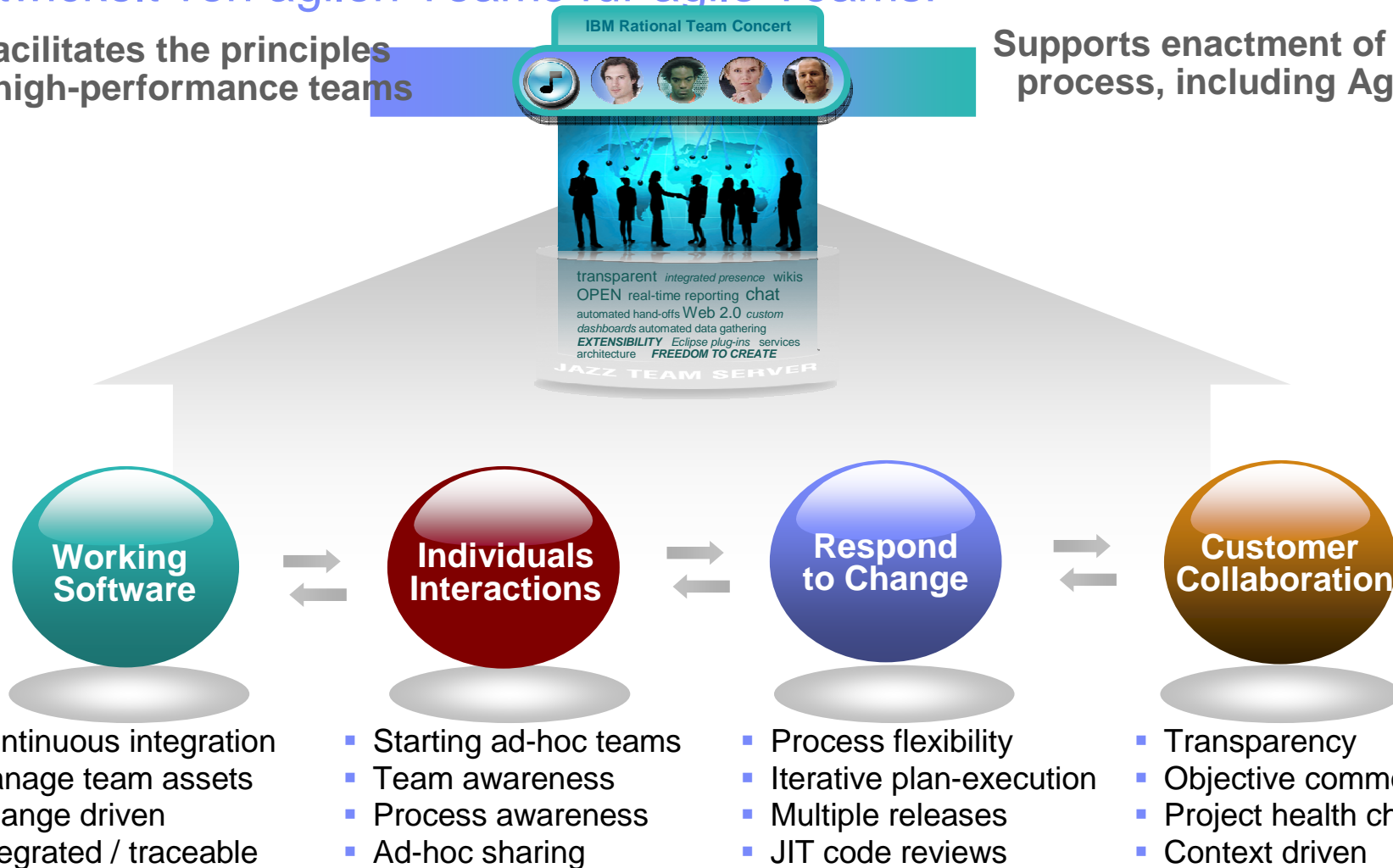
- Einführung in SW Entwicklungsprozesse
- Agile SW Entwicklung
- SCRUM
- Agile Entwicklung mit Rational Team Concert
  - Überblick Rational Team Concert
  - Agiles Planen mit RTC
  - Change Management
  - Integriertes Build Management



# IBM Rational Team Concert: Entwickelt von agilen Teams für agile Teams!

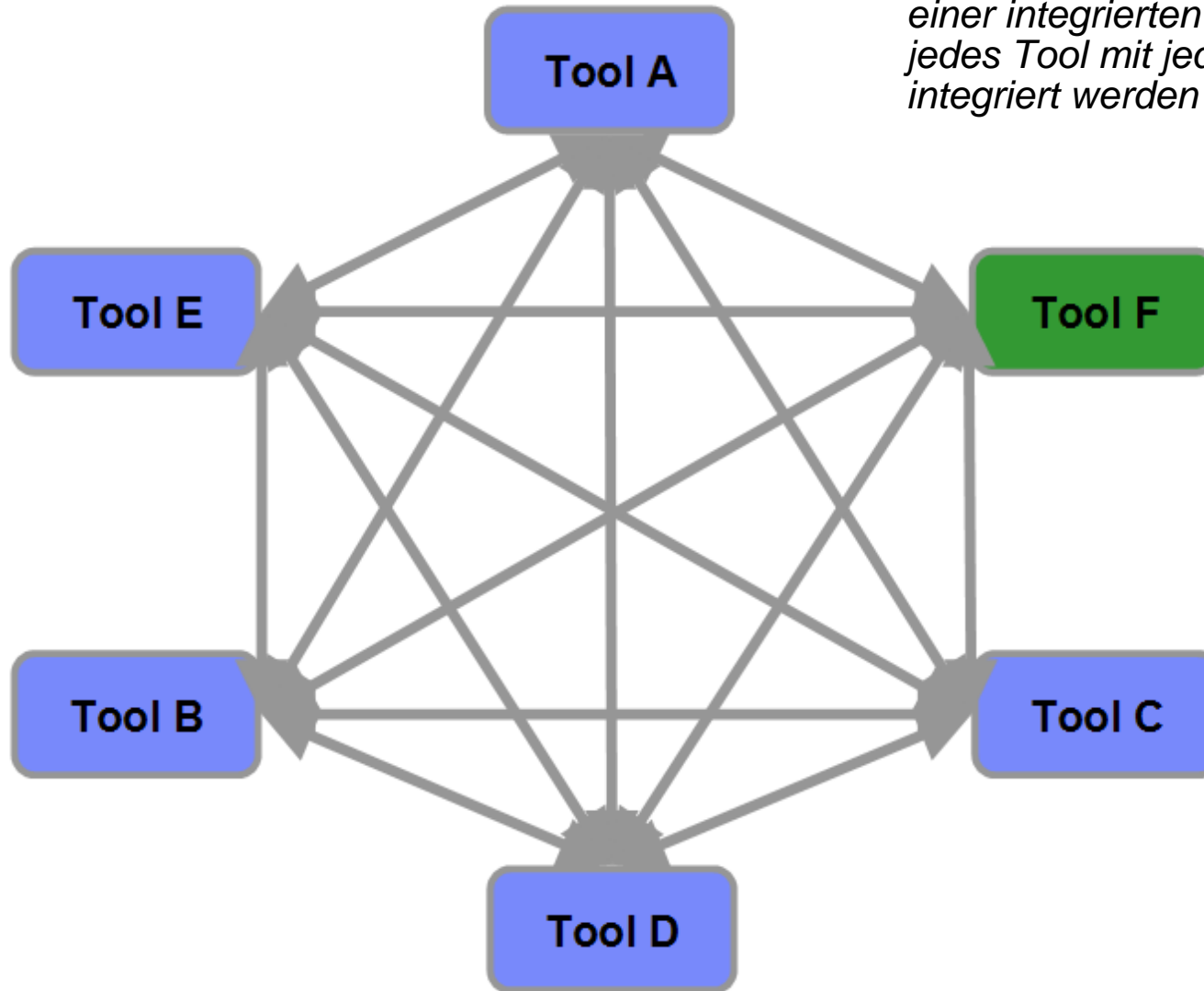
Facilitates the principles of high-performance teams

Supports enactment of any process, including Agile



# Toolintegration bedeutet heute quadratische Komplexität!

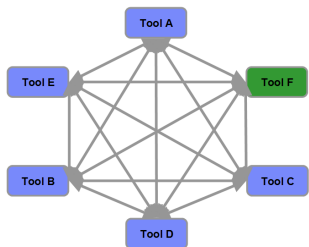
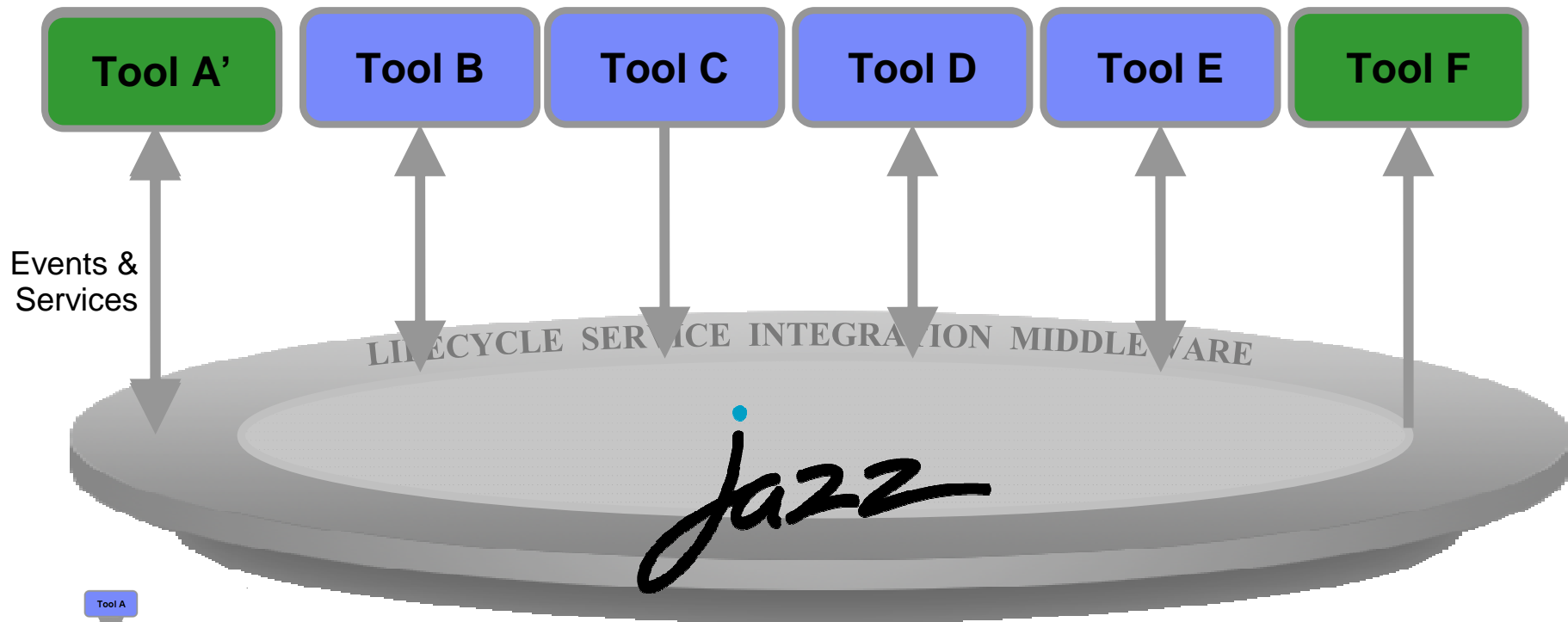
*Bis heute bedeutet das Aufbauen einer integrierten Plattform, dass jedes Tool mit jedem anderen Tool integriert werden muß!*





## Tools collaboration based on middleware services

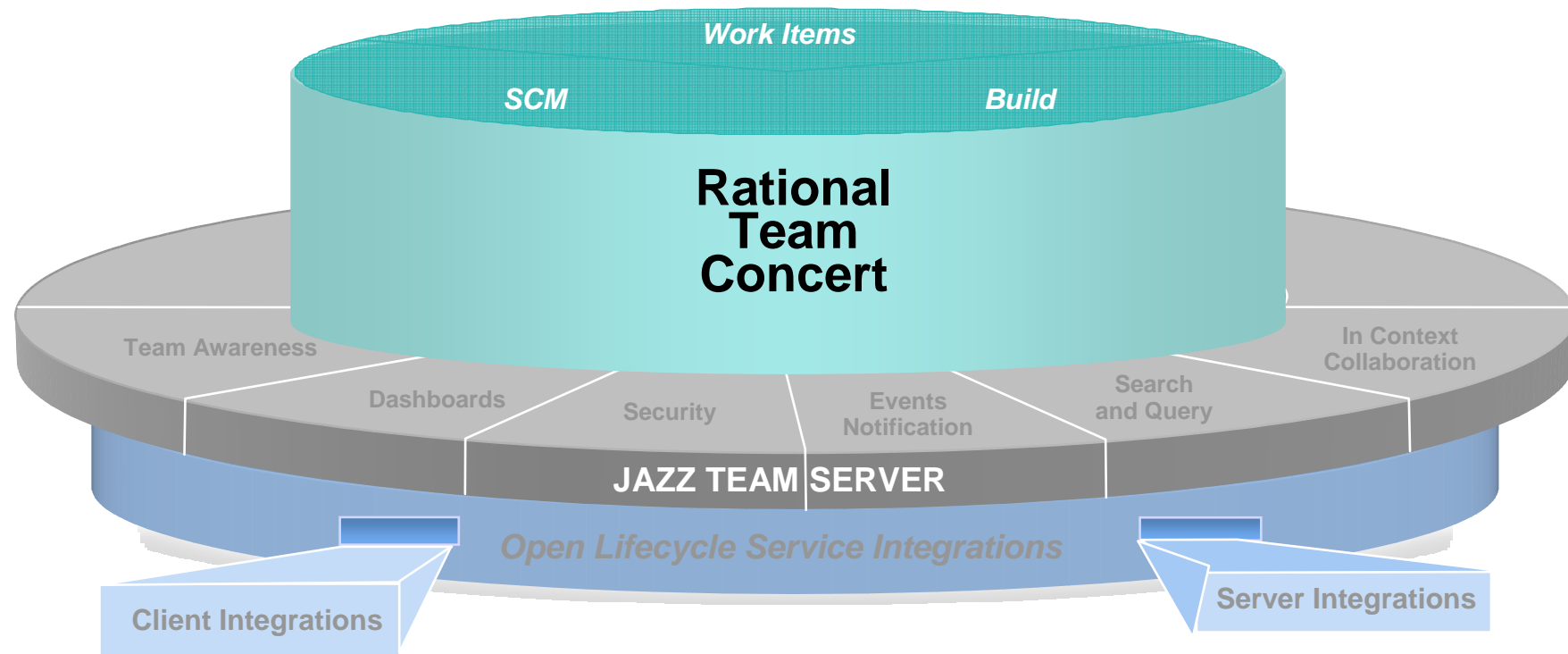
- With the Jazz platform, the tools communicate only with the platform:
  - By listening to normalized/standardized events from the platform
  - By sending normalized/standardized events to the platform



### Team Collaboration Services

## Rational Team Concert (RTC)

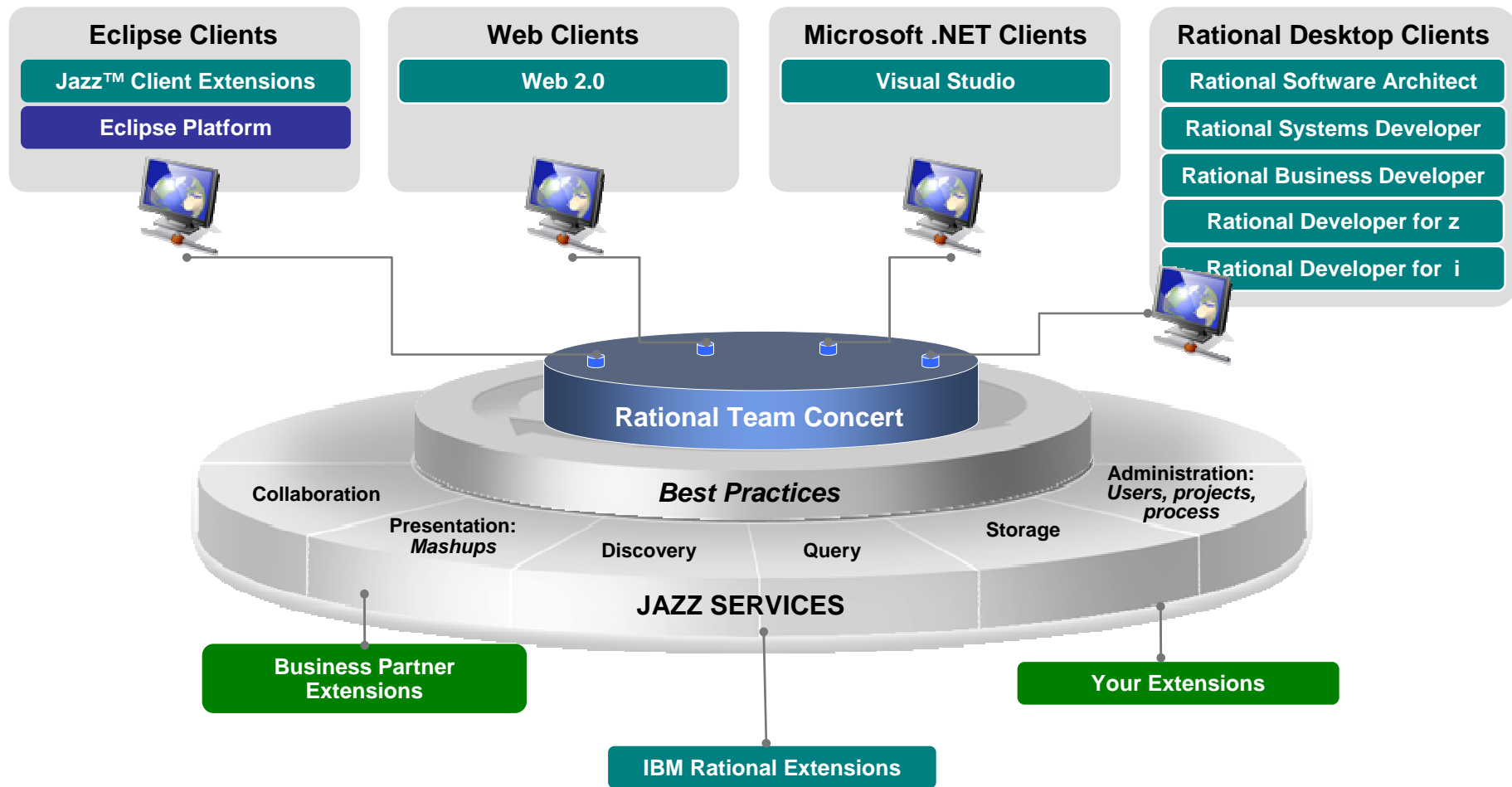
Das erste Produkt auf der Jazz Plattform



**Einfache Inbetriebnahme und kurze Einarbeitungszeit**

# Rational Team Concert: An open, extensible architecture

*Supporting a broad range of desktop clients, IDE's and languages*



## Agenda

- **Einführung in SW Entwicklungsprozesse**
- **Agile SW Entwicklung**
- **SCRUM**
- **Agile Entwicklung mit Rational Team Concert**
  - **Überblick Rational Team Concert**
  - **Agiles Planen mit RTC**
  - **Change Management**
  - **Integriertes Build Management**



# Project Plans

- A plan revolves around the following elements
  - ▶ Teams
  - ▶ Time
  - ▶ Work
  
- Planning levels
  - ▶ Release
  - ▶ Iteration/Sprint
  - ▶ Your day-to-day work

**Squawk Teams (7)**

- ▼ Squawk Maintenance Team
  - Core Library M1 Maintenance
- ▼ Squawk Team
  - Core Library
  - Documentation
  - Release Engineering
  - User Interface

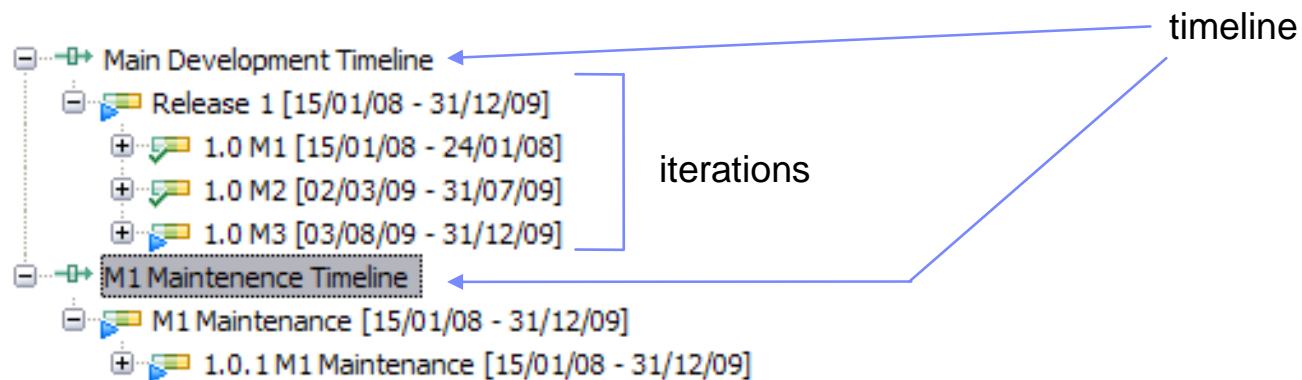
**1.0 M3** 3 Aug 2009 - 31 Dec 2009

	493	Create futuristic squawkers
	470	Support theme based squawk parties
	471	Support other types of technologies like audio and video squawkers
	484	Predefine template squawker classes
	469	Allow new squawkers to be defined from a user interface instead of code

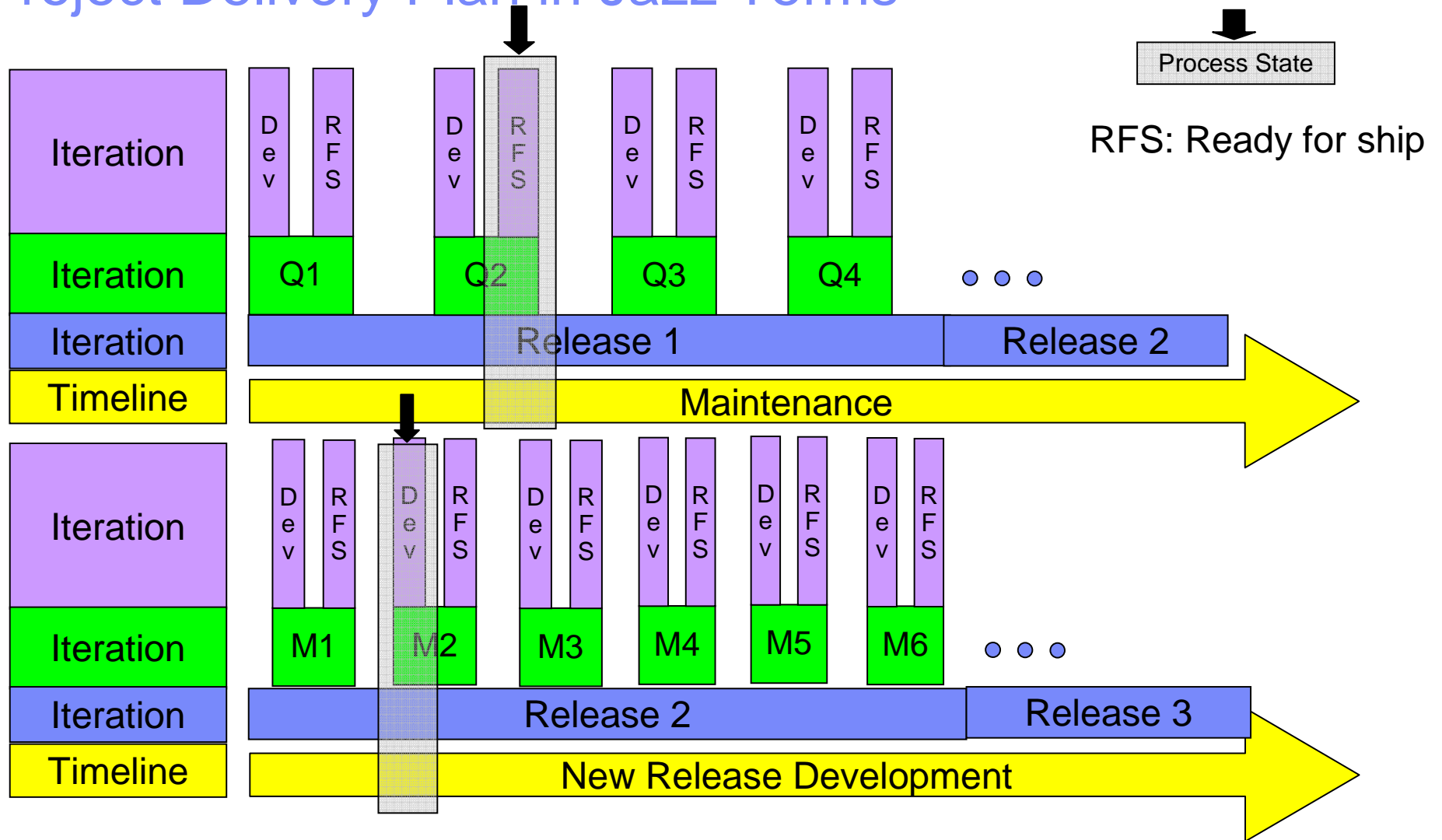


# Time

- A project has one or more schedules or **timelines**.
- Each timeline is broken down into a series of iterations
- Each iteration can be broken into smaller iterations as required
- A milestone marks the end of an **iteration**
- At the end of any iteration, you may produce a **release**

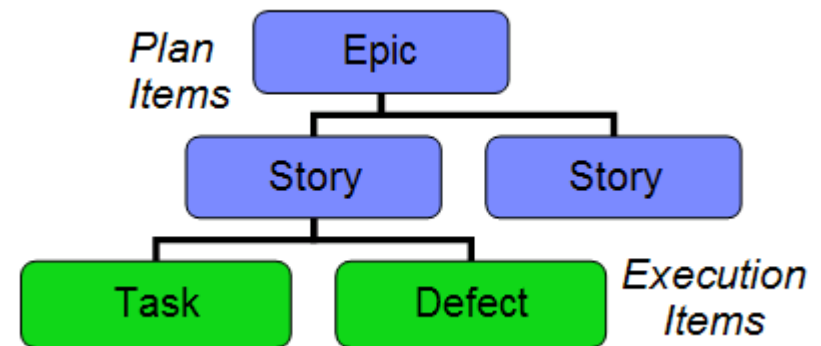


# Project Delivery Plan in Jazz Terms



## Work

- All work in your project is tracked as one or more Work Items
- Different kinds of work items are available including
  - ▶ Plan work item types, for example:
    - Plan Item
    - Epic
    - Story
  - ▶ Execution work item types, for example:
    - Task
    - Defect
  - ▶ Plan work item types are used to capture high-level plan elements
  - ▶ Execution work item types are used to capture the lower-level details and the work should be completed in a single iteration
- Each kind of work item has its own lifecycle



# Overall Project Plan

**Backlog** 📄 💰

Team Area: Squawk Team | Iteration: Release 1 (15/01/2008 - 31/12/2009) | 12 Closed | 7 Open

Overview | **Planned Items** | Charts

Progress: 80/135 Story Points Estimated: 100%

View As: Teams ✎ 📄 ✕ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸

Release 1		Progress: 80/135 Story Points		Estimated: 100%	
Closed Items: 12   Open Items: 7					
▶	🔗 Create a broader collection of squawkers	0/5		High	466
	🔗 Predefine template squawker classes	5 pts	0/0	Medium	484
	🔗 Support other types of technologies like audio and video squawkers	20 pts	0/0	Unassigned	471
	🔗 Support theme based squawk parties	8 pts	0/0	Unassigned	470
	🔗 Allow new squawkers to be defined from a user interface instead of code	12 pts	0/0	Unassigned	469
1.0 M1		Progress: 33/33 Story Points		Estimated: -	
Closed Items: 6   Open Items: 0					
	🔗 Create application build environment	5 pts	0/0	High	464
	🔗 Create core application based on initial prototype	5 pts	0/0	High	464
	🔗 Define application documentation	3 pts	0/0	Medium	466
	🔗 Define a handful of example squawkers	3 pts	0/0	Medium	463
	🔗 Create a basic user interface	12 pts	0/0	Unassigned	462
	🔗 Test Core Library team process change for iteration M2 endgame	5 pts	0/0	Unassigned	27
1.0 M2		Progress: 47/47 Story Points		Estimated: -	
Closed Items: 6   Open Items: 0					
	🔗 Streamline personal build	5 pts	0/0	High	483
▶	🔗 Create a .net implementation of the Squawker application	20 pts	22/22	Medium	492
1.0 M3		Progress: 0/5 Story Points		Estimated: 100%	
Closed Items: 0   Open Items: 1					
	🔗 Set up a project status web page	5 pts	0/0	Unassigned	468

# Team Project Plan

## Team's Project Plan

Team Area: Core Library | Iteration: Release 1 (15/01/2008 - 31/12/2009) | 2 Closed | 0 Open

Overview

Planned Items

Charts

Progress: 2/2 Story Points

Estimated: --

View As: Iterations

<b>▼</b>	<b>Release 1</b> Closed Items: 2   Open Items: 0			
		<div style="border: 1px solid #ccc; width: 100%; height: 15px; background-color: #669933;"></div>	Progress: 2/2 Story Points	Estimated: --
	<b>1.0 M1</b> Closed Items: 0   Open Items: 0	<div style="border: 1px solid #ccc; width: 100%; height: 15px; background-color: #fff; text-align: center;">No Work</div>	Progress: 0/0 Story Points	Estimated: --
<b>▼</b>	<b>1.0 M2</b> Closed Items: 2   Open Items: 0	<div style="border: 1px solid #ccc; width: 100%; height: 15px; background-color: #669933;"></div>	Progress: 2/2 Story Points	Estimated: --
←	<b>▼</b> Create a .net implementation of the Squawker application	20 pts	2/2	Medium
	Define a Jazz build definition that runs MS Build	1 pt	0/0	Unassigned
	C# implementation	1 pt	0/0	Unassigned
	<b>1.0 M3</b> Closed Items: 0   Open Items: 0	<div style="border: 1px solid #ccc; width: 100%; height: 15px; background-color: #fff; text-align: center;">No Work</div>	Progress: 0/0 Story Points	Estimated: --



# Team Iteration Plan

**Core for M1** 📄 🏷️

Team Area: Core Library | Iteration: 1.0 M1 (15/01/2008 - 24/01/2008) | 5 Closed | 0 Open

Overview | **Planned Items** | Charts | Release Notes | Retrospective

Progress: 14/14 | 0 h Estimated: --

View As: Work Breakdown 📝 📁 ✖️ ⏪ ⏩ ➕ ⏪ ⏩ ✎

<b>April Blues</b> Closed Items: 2   Open Items: 0 Load: 0/0 <span style="float: right;">No Work Time Left</span> Estimated: --	
Define a handful of example squawkers Create Car squawker Create Lion squawker	Medium 6/6 h 2 hours Medium 2 hours Medium
<b>Jerry Jazz</b> Closed Items: 3   Open Items: 0 Load: 0/0 <span style="float: right;">No Work Time Left</span> Estimated: --	
Create core application based on initial prototype Core Code Define a handful of example squawkers Cat Implementation Dog Implementation	High 8/8 h 1 day Medium Medium 6/6 h 1 hour Medium 1 hour Medium

# Developer's Taskboard

**Markus Kent**  
 Closed Items: 4 | Open Items: 7

Progress: 26/36.25 h      Estimated: 71%

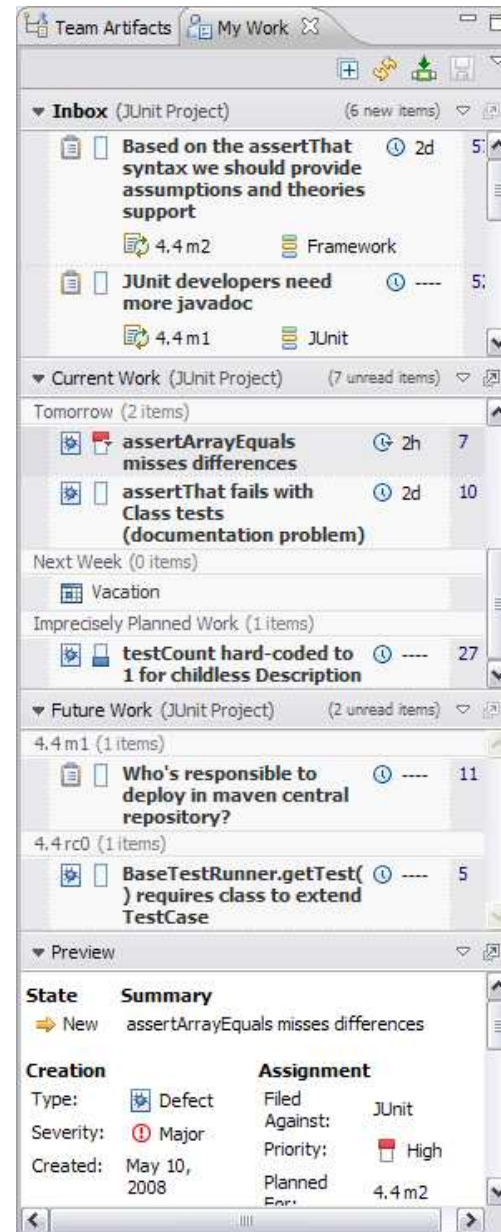
	To Do	In Progress	Done
<ul style="list-style-type: none"> <li>Improve documentation for 4.4 (55)</li> <li>Provide improved Assertion syntax (60)</li> </ul>	<ul style="list-style-type: none"> <li>javadoc updates for @Ignore in 4.3 (30)</li> <li>Based on the assertThat syntax we should provide assumptions and theories support (59)</li> </ul>		<ul style="list-style-type: none"> <li>[Docs] Cookbook TestRunner section incorrect (23)</li> </ul>
	<ul style="list-style-type: none"> <li>assertArrayEquals misses differences (7)</li> <li>testCount hard-coded to 1 for childless Description (27)</li> <li>Tests on protected methods fail (14)</li> <li>assertThat fails with Class tests (documentation problem) (10)</li> </ul>	<ul style="list-style-type: none"> <li>assertArrayEquals misses differences</li> </ul>	<ul style="list-style-type: none"> <li>shows green bar while assert false (44)</li> <li>Should not call derived's afters if super's before failed (47)</li> <li>@After method not called after my test timeout in 4.3.1 (46)</li> </ul>

See the work currently in progress

Drag and drop work items to change their state.

## Daily Planning

- My Work View
- Developer plans assigned work
- Adjusts estimates



## Agenda

- **Einführung in SW Entwicklungsprozesse**
- **Agile SW Entwicklung**
- **SCRUM**
- **Agile Entwicklung mit Rational Team Concert**
  - **Überblick Rational Team Concert**
  - **Agiles Planen mit RTC**
  - **Change Management**
  - **Integriertes Build Management**

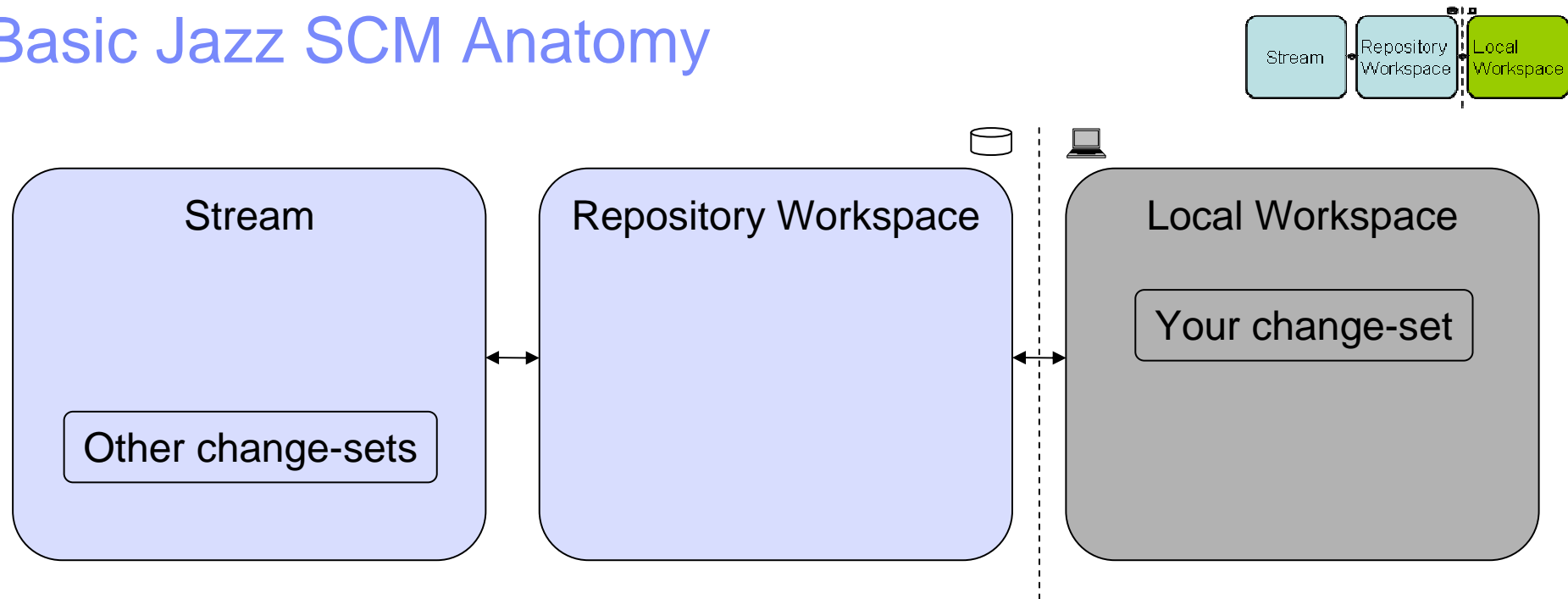




## Jazz Source Control

- Management von Quellcode und anderen Artefakten
  - Pflegt die vorherigen Konfigurationen des Quellcodes
  - Verhindert unautorisierten Zugriff auf die Artefakte
  - Meldet den Benutzern wenn ein Artefakt sich geändert hat
  
- Zusätzlich
  - Kennt die Struktur der einzelnen Komponenten
  - Pflegt eine integrierte **Historie** der Änderungen
  - Hilft bei Unterbrechungen und Konflikten
  - **Integriert** sich tief mit den anderen Jazz Teilen

# Basic Jazz SCM Anatomy



- Streams are for sharing
- Repository workspaces are your personal space
- The local workspace is a folder on your local files system where you develop and test
- Change-sets flow back and forth



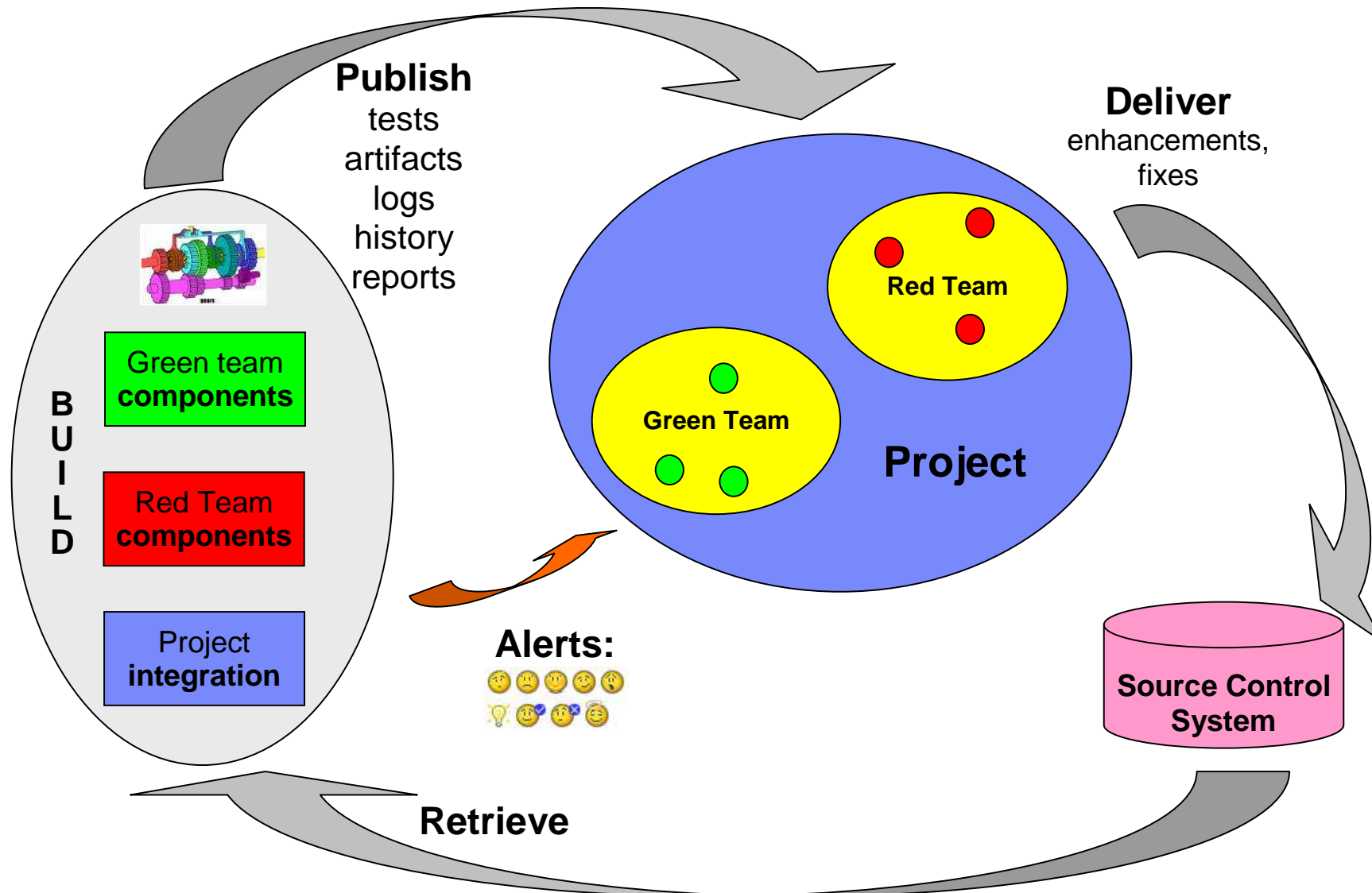
## Agenda

- **Einführung in SW Entwicklungsprozesse**
- **Agile SW Entwicklung**
- **SCRUM**
- **Agile Entwicklung mit Rational Team Concert**
  - **Überblick Rational Team Concert**
  - **Agiles Planen mit RTC**
  - **Change Management**
  - **Integriertes Build Management**





# Build im Kontext von agilem entwickeln





# Der Build ist sehr visibel für den Benutzer

The screenshot displays the Rational Team Concert interface for a build. The main window shows the build details for 'workshop.squawk.core.continuous.build'. A callout 'My builds' points to the build engine in the left sidebar. A callout 'Published build' points to the build status 'Completed'. A callout 'Recent builds' points to the 'Build' section in the right sidebar. A callout 'Alerts' points to an alert box at the bottom right. A callout 'History' points to the build history table at the bottom.

**Build Details:**

- Status: **Completed**
- Duration: 23 seconds
- Time: April 7, 2008 6:53:51 PM
- Completed: April 7, 2008 6:54:14 PM
- Status Trend: [Progress bar]

**Contribution Summary:**

- Downloads: [1 download](#)
- External Links: [1 link](#)
- Logs: [1 log](#)
- Repository Workspace: [workshop.squawk.core.build.workspace](#)
- Snapshot: [workshop.squawk.core.continuous.build\\_B20080407-1553](#)
- Compile: [0 errors, 0 warnings](#)
- JUnit: [4 tests, 0 failures, 0 errors](#)
- Work items: [1 included in build](#)
- Changes: [Show changes](#)

**General Information:**

- Requested by: Zach Builder
- Build Definition: [workshop.squawk.core.continuous.build](#)
- Build Engine: [workshop.build](#)
- Build History: [16 builds](#)
- Tags: [ ]
- Deletion allowed

**Build History Table:**

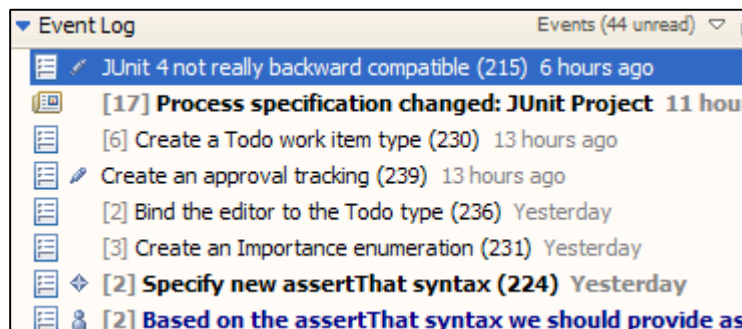
Build	Label	Progress	Estimated	Time	Duration
workshop.squawk.core.continuous.b...	B20080407-155...	Completed		April 7, 2008 6:53:51...	23 second
workshop.squawk.core.continuous.b...	B20080314-150...	Completed		March 14, 2008 3:07:...	28 second
workshop.squawk.core.continuous.b...	B20080313-133...	Completed		March 13, 2008 4:37:...	12 second
workshop.squawk.core.continuous.b...	B20080313-104...	Completed		March 13, 2008 1:45:...	38 second
workshop.squawk.core.continuous.b...	B20080307-145...	Completed		March 7, 2008 5:59:1...	11 second

**Alerts:**

- Build finished successfully
- workshop.squawk.core.co ntinuous.build
- B20080116-1350-worksh op.squawk.core.continuo us.build

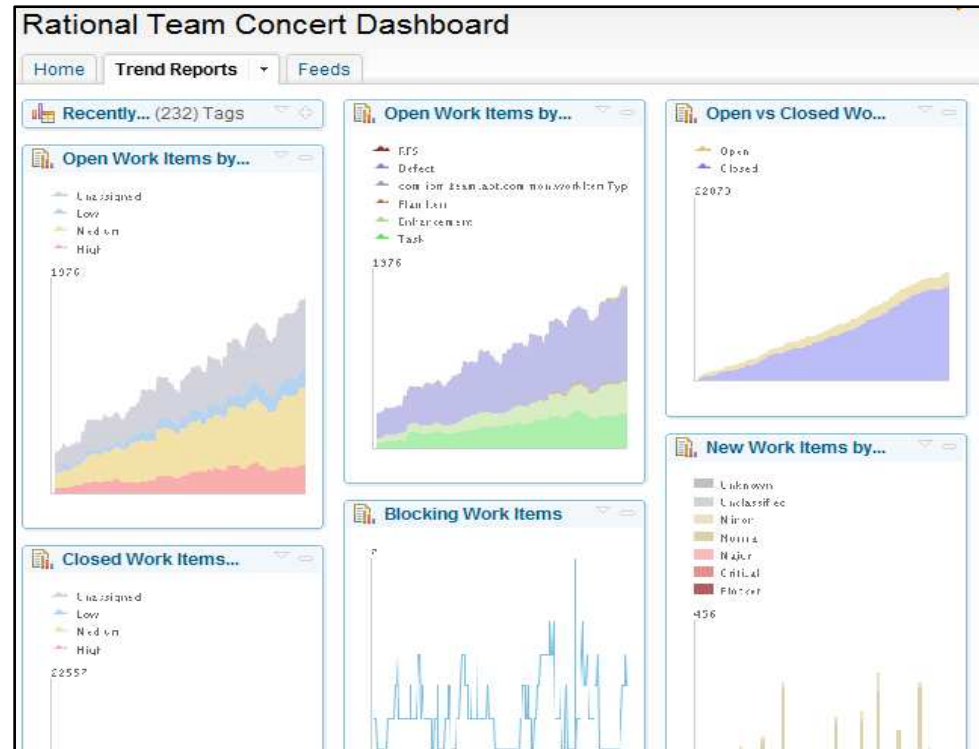
## Event Feeds and Notification

- RTC provides **immediate** feedback for all project events
  - ▶ Shows what other team members are doing
  - ▶ Approvals, builds, news, reports, CM changes, process changes, work changes, etc.
- Location-independent notification
  - ▶ Flexible notification – pop-up (right figure), email, RSS



# Dashboards

- Live, aggregated project information
  - ▶ Shows work, builds, schedules, load, news, events, etc.
  - ▶ Dashboards for projects, teams, individuals
  - ▶ Variety of presentation options – bar, line, pie, cloud
- **Certainty** – are we on track?
- **Immediacy** – live project data



# Wie IBM Rational Team Concert Express nutzen

## Beaverton



- Build
- Process

## Toronto



- Source Control
- Reporting
- Community Site

## Zürich



- UI Foundation
- Work Items
- Agile Planning
- Code Coverage

## Saint-Nazaire



- Static Analysis

## Lexington



- Interop
- Testing

## Raleigh











- Repository
- Web UI



- 2-Prozessor Xeon Server für Websphere Application Server
- 2-Prozessor Xeon Server für DB2
- 70 Entwickler, 20 Tester, 8 Dokumentierer
- 18,000 Change-Sets
- 35,000 Work Items
- 250 Repository Workspaces
- 66K Dateien
- 10GB Dateninhalte unkomprimiert
- Integration Stream hat 30K Dateien
- 40 Builds pro Tag
- Wöchentlichen Integration Build



## Editionen von Rational Team Concert

Features	 			
	 <b>Express-C</b>	 <b>Express</b>	 <b>Standard</b>	 <b>Enterprise</b>
Select your OS: <input type="text" value="Windows x86"/> Looking for the <a href="#">Mac Client</a> incubator?	Free for 10 users! Zip file download with Client for Eclipse IDE and server. Client for Microsoft Visual Studio IDE downloaded separately: <a href="#">Other Download Options...</a>	Great for small teams! Installation Manager Web Install for clients, server, and optional components.  <a href="#">Other Download Options...</a>	The standard for corporate teams! Installation Manager Web Install for clients, server, and optional components.  <a href="#">Other Download Options...</a>	Scalability for the enterprise! Installation Manager Web Install for client, server, and optional components.  <a href="#">Other Download Options...</a>
Server license	Permanent	Trial license	Trial license	Trial license
Developer Client Access Licenses	10 included FREE	3 included (50 in trial)	3 included (250 in trial)	3 included (No limit trial)
Contributor Client Access Licenses	Available for Purchase	Available for Purchase (Unlimited in Trial)	Available for Purchase (Unlimited in Trial)	Available for Purchase (Unlimited in Trial)
Change Management Client Access Licenses	Available for Purchase	Available for Purchase	Available for Purchase	Available for Purchase
IBM Support	Purchased licenses only	Full IBM Support	Full IBM Support	Full IBM Support
Maximum Developers / Contributors	10 / unrestricted	50 / unrestricted	250 / unrestricted	unrestricted / unrestricted
Databases supported	Derby (included), DB2, Oracle, SQL Server	Derby (included), DB2 (Workgroup Edition included), Oracle, SQL Server	Derby (included), DB2 (Workgroup Edition included), Oracle, SQL Server	Derby (included), DB2 (Workgroup Edition included), Oracle, SQL Server
Application servers supported	Tomcat (included)	Tomcat (included), WebSphere	Tomcat (included), WebSphere*	Tomcat (included), WebSphere*



## Zusammenfassung

***„Das Ganze ist mehr als die „ Summe seiner Teile“***



Rational Team Concert und die Jazz Plattform

## Noch Fragen?

## Kontaktieren Sie Werner Schoepe

**Technical Presales**

Tel: 0211 476 2163

Email: [werner.schoepe@de.ibm.com](mailto:werner.schoepe@de.ibm.com)

