IBM Technical **Summit**

> Stay ahead.

Name: Sreerupa Sen
Title: Fast Track your Agile Project Teams using Rational Team Concert

Outline

- What is Rational Team Concert?
- What is Agile?
- Rational Team Concert for Agile Teams

What if…as a developer today…

Everything I need to do my job is in one place at my fingertips

All my code, my tasks, my builds, my to do lists, my plans, my backlog

I know exactly what's going on

Work items and RSS event feeds track everything -- change sets, builds, and approvals

Everyone else knows what's going on (so they can stop bugging me!)

When can I test your fix?

What's your status? When will you be done? Did the plan change?

I can work smarter AND faster

I can fix goofs, juggle tasks, patch in seconds, easily add or remove features.

I can figure out and fix my mistakes AND the "other guy's" mistakes…

I can experiment with new ideas without impacting the rest of the team

IBM

What if…as a team today…we could…

Work more like a startup – all in one virtual room?

Our team could collaborate around all development artifacts and events in one place

Automatically keep everyone informed of project status?

Web based dashboards track quality, status, important project updates

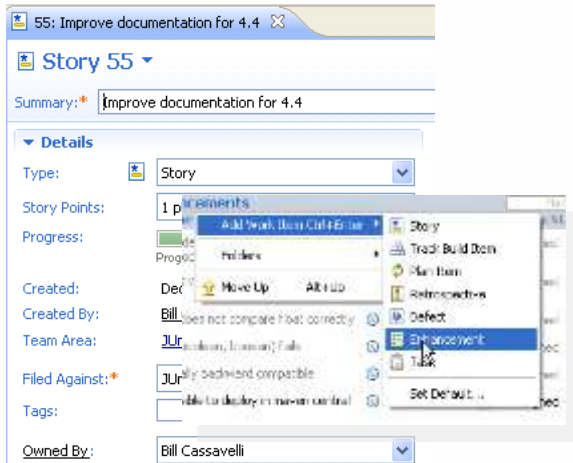**Move away from a culture of over committing and under delivering?**

Our team is empowered with data to understand the impact of adding more work.

Transform the way we work so we can adopt rapid release cycles or agile?

We need tools that help us transform - we need a breakthrough approach, not just another point tool..

Rational Team Concert – All our work …

- **Work Items**
- **Planning**
- **Source Control**

- **Builds – Continuous**
- **Dashboards & Reporting**
- **Process Enforcement and Automation**
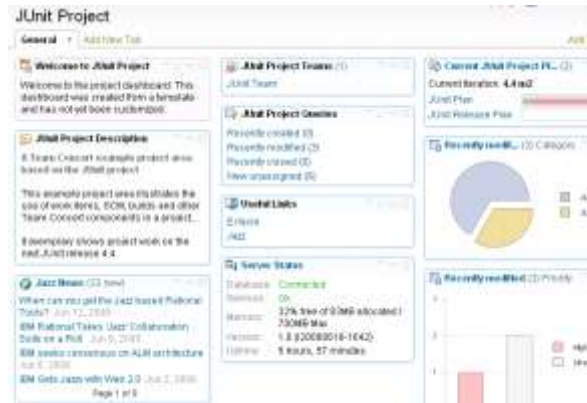
Linked together

"Everyone else has point product integrations. Rational Team Concert has a fusion of workflows that enable developers to work faster and smarter"

Iteration Plan

planned for

Release

found in

Work Item

reported against

Build

promoted

subscribes approves reviews

included

User

related

implements

included

Change Set

included

Snapshot

built from

Artifacts

Stream

Workspace

change flow
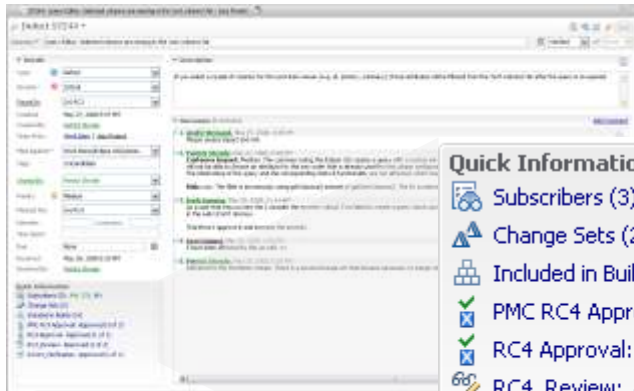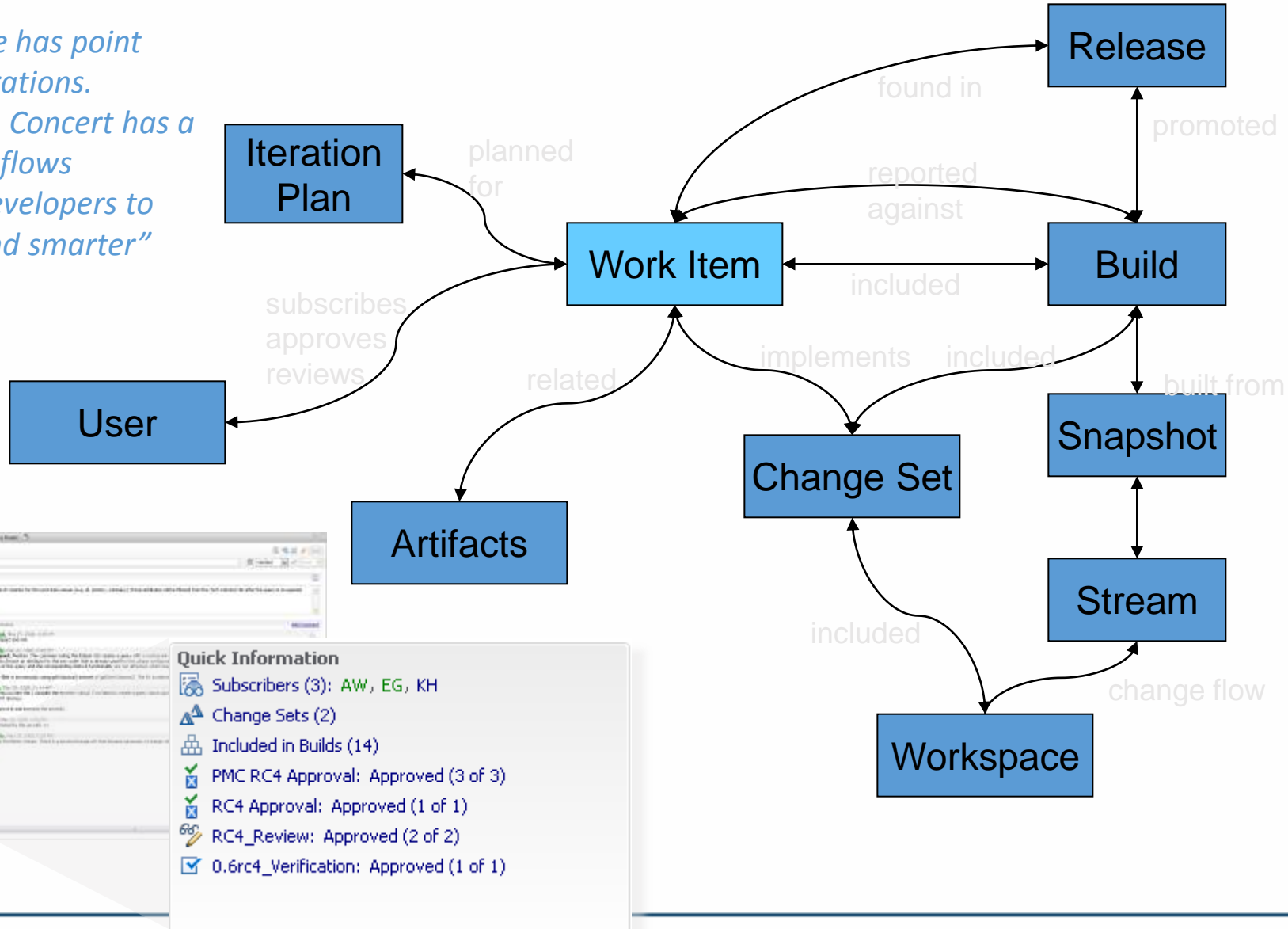
Quick Information
- Subscribers (3): AW, EG, KH
- Change Sets (2)
- Included in Builds (14)
- PMC RC4 Approval: Approved (3 of 3)
- RC4 Approval: Approved (1 of 1)
- RC4_Review: Approved (2 of 2)
- 0.6rc4_Verification: Approved (1 of 1)

IBM

In one place

Friction-free communication

Across any platform

Open, extensible, linked data platform that enables other tools to participate and provides future portability as your platform needs evolve

Eclipse Clients
- Jazz Client Extensions
- Eclipse Platform

Web Clients
- Web 2.0

Microsoft .NET Clients
- Visual Studio
- Windows Shell
- MSSCCI

Rational Desktop Clients
- Rational Software Architect
- Rational Systems Developer
- Rational Business Developer
- Rational Developer for z
- Rational Developer for i

Other Clients
- SCM CLI
- ISPF

Rational Team Concert

Best Practices

Collaboration

Presentation: *Mashups*

Discovery

Query

Storage

Administration: *Users, projects, process*

JAZZ SERVICES

Business Partner Extensions

Your Extensions

IBM Rational Extensions

Visible to everyone

RTC enables "viral visibility" – free read access to web dashboards for ALL users

Outline

- What is Rational Team Concert?
- What is Agile?
- Rational Team Concert for Agile Teams

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	

What is Agile?

- Agile is **not** about
  - No design
  - No documentation
  - Stand-up meetings
  - Rapid coding
  - Speed of delivery
 - Doing more with less...

What is Agile?

- For every stable development methodology, there need to be checks and balances

What you aim for	How you get to it
➢ Software that works ➢ Is delivered on time ➢ Responds to changes	✓ Iteration and Release Planning ✓ Just enough design ✓ Evolving processes ✓ Continuous integration ✓ Test driven development ✓ Stakeholder collaboration throughout the cycle ✓ End of iteration demos ✓ Retrospectives

Agile Practices

continuous testing

continuous integration

consume your own output

validate

enable

sign off

drive with open eyes

end game

reduce stress

community involvement

transparency

milestones first

live betas

show progress

attract to latest

always have a client

feedback

new & noteworthy

validate

update

learn

API first

adaptive planning

retrospectives

End of iteration demos/reviews

enable

Ranked Product Backlog

component centric feature teams

Burndown

Stories

validate

Daily Standup

Rules of the Road

Adoptions Expectations

Buddy Review

Feature teams

Start with the Planning…

It's all in the Planning!

- **Strategy** – vision for a business need or direction.

- **Portfolio** – overall product offering that consists of applications and tools and how they integrate.

- **Product** – product vision and the road-map for the product.

- **Release** – prioritized backlog of product features committed to be delivered in a given time period.

- **Iteration** – short, fixed-length subsets of releases, typically 1-4 week time frame.

- **Day** – actions taken in a day to achieve the iteration goals.

Strategy

Portfolio

Product

Release

Iteration

Day
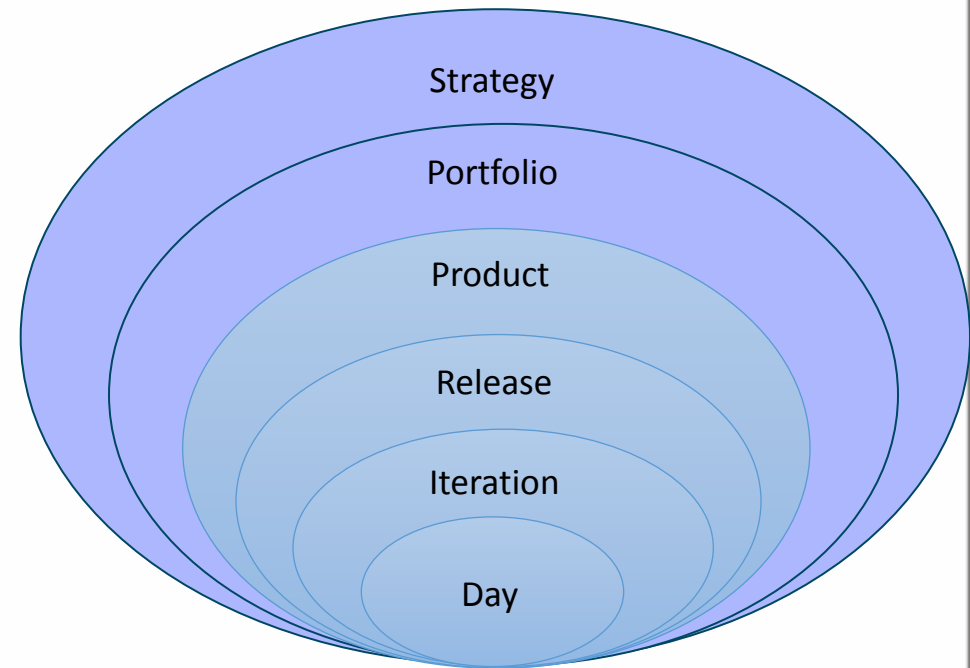
In a Development Project the four innermost levels of the planning onion are the most important.

Go Agile!

- Optimize your agile teams through
  - Real-time planning
  - Traceability
  - In-context collaboration
  - Development Intelligence
  - Continuous Improvement
- Get up and running quickly
- Extend as your needs evolve
- **Choose the right tools**

Outline

- What is Rational Team Concert?

- What is Agile?

- Rational Team Concert for Agile Teams

Smart Tools, Winning Teams

- Plan your way
  - Out-of-the-box templates for agile workflows
  - Plans linked to development execution – see immediate impacts of change
  - Not an isolated planning tool

- Track your tasks
  - Smartly linked work items that behave how you want them to

- Be Team aware
  - Day one: up, running, coding, and productive
  - Linked work, enables teams to find, fix and test faster
  - Artifact feeds, make work transparent across teams
  - Live project status: Not manually cobbling bad data from multiple tools

Smart Tools, Winning Teams

- Process simplified
    - An agile, adaptive process: trained by you
    - Learn, Adapt, Reflect, Change

- Manage your sources
    - Create, undo, share, update, review, suspend, resume, patch

- Keep builds healthy
    - Build continuously, at multiple levels, with the technology of your choice

- Build, deploy, test, fix
    - Continuous integration

- Dashboards for status
    - Easy-to-create, web-based dashboards

Planning (and Tracking) in RTC

- Plans are at the heart of an agile process
- The product lead works on the product backlog
  - Business stakeholders, market direction
- At every release, feature teams work on the prioritized items of the ranked product backlog.
  - Release planning
- At every milestone, the feature team lead creates a sprint plan.
- Every day, developers work off their personal work views
  - My Work, My Dashboard

IBM

Rational Team Concert (RTC) Plan Definition
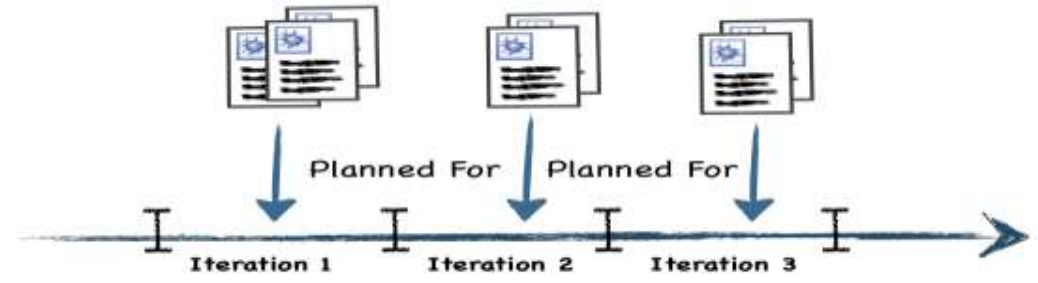
Work item summary

Defect, Enhancement, Task, Story, ..

Category

Planned For

Working on the … Product Backlog

- Prioritized feature list

- Plan **Views**
  - Ranked List
  - Work Breakdown

Planned Items ? Links Sn
View As: Ranked List ▼
Ranked List
Work Breakdown

Create Plan
- Sprint Backlog
- Release Backlog
- Product Backlog
- Cross Project Plan
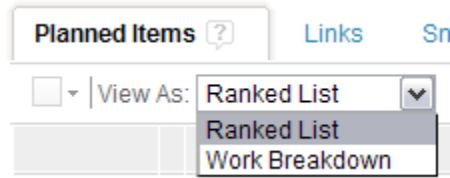
- **WorkItem Types**
  - Plan items: Work items that are relevant for planning. Ex: Epics, Stories, etc.

- Defect
- Task
- Story
- Epic
- Track Build Item
- Impediment
- Adoption Item
- Retrospective

Product
Release
Iteration
Day
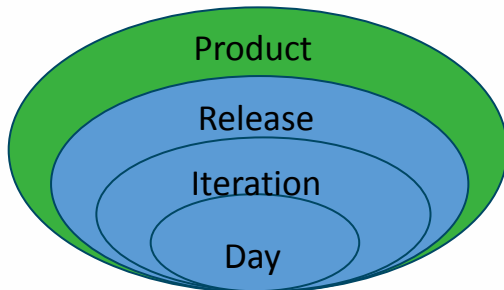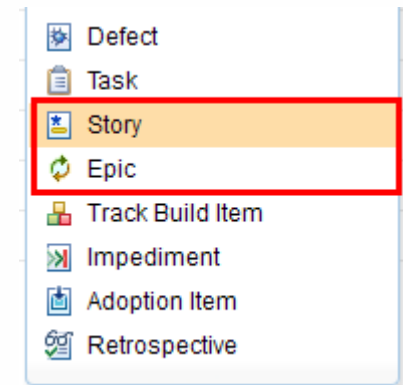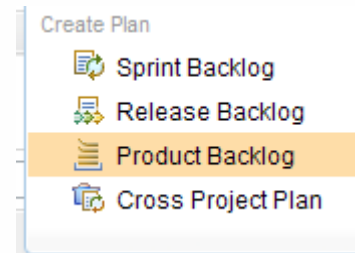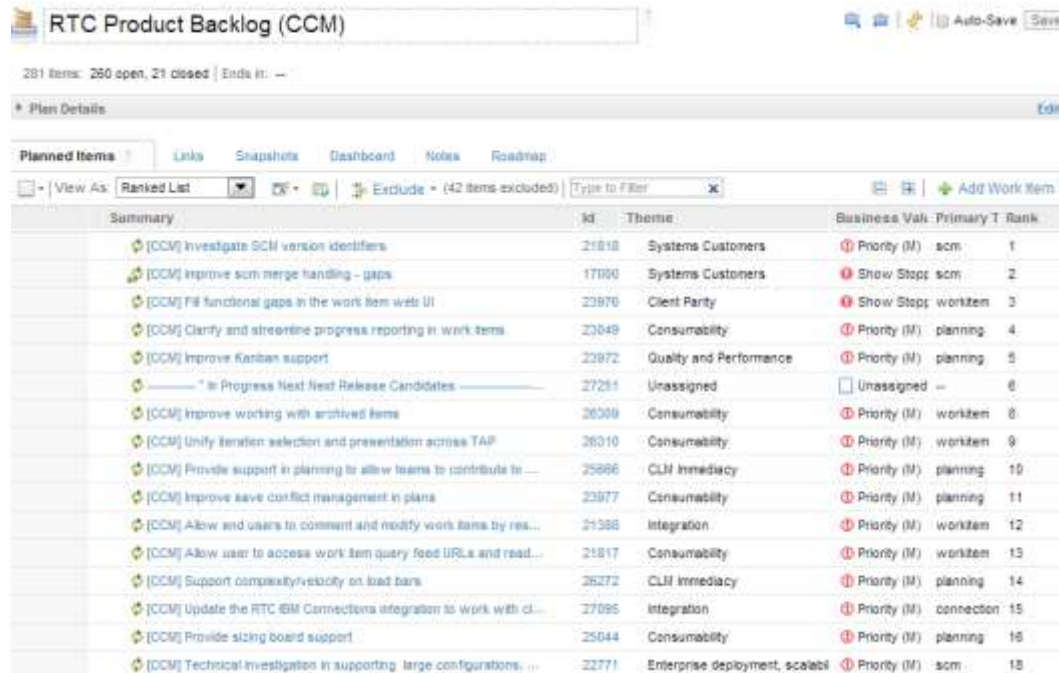
- **Reports**
  - How your product backlog is evolving?
  - How many items are there on the product backlog?
  - Can we find out how long it would take to complete the product backlog?

- **Top down** Planning

IBM

Managing the Product Backlog

- High-level plan that contains stories not planned for any release or iteration yet
- Targeted for the iteration marked as **Backlog Iteration**
- **Prioritized** feature list based on risk, business value, etc.

- Plan items – include only non execution items such as Epics and Stories
  - **Customize** with themes, business value, etc
  - Define the **Done** Criteria

- **Sizing** in Story points - rough estimates based on complexity and development effort

- **Ranking** - ensure that the most valuable functionality is delivered first

Managing the Product Backlog

- Progress on Product Backlog
  - **Story Points Remaining**
  - **Open Vs Closed Work items**
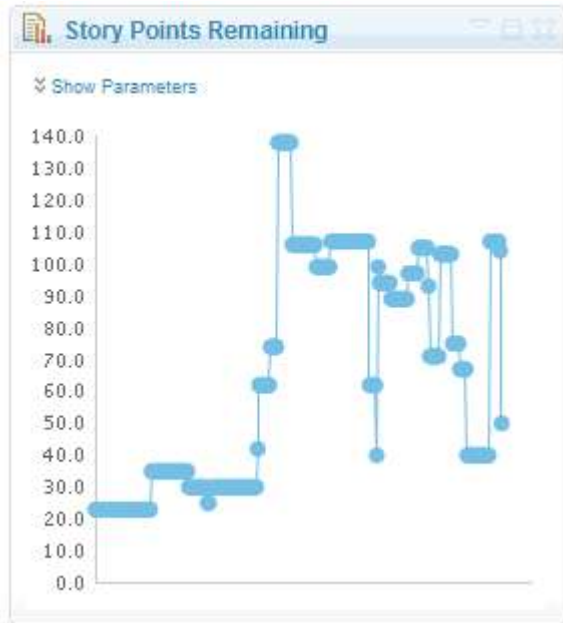
Working on the … Release Backlog

- Planned/Committed features for the release of the product

- Plan **Views**
    - Iterations
    - Ranked List
    - Roadmap
    - Teams
    - Traceability (optional)
    - Work Breakdown
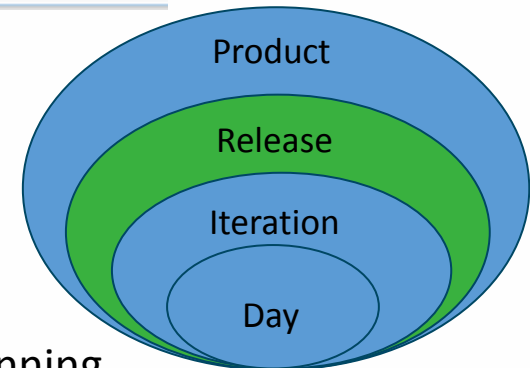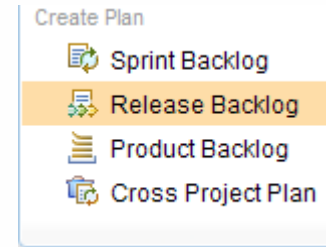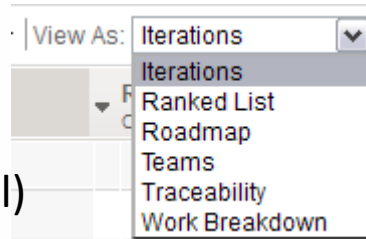
- **Work Item Types**
    - Plan items: Work items that are relevant for the planning.
      Examples: Stories
- **Reports**
    - Team Velocity
    - Release Burndown
- **Top down** Planning

Managing the Release Backlog …

- Overview of all sprints in the release

Working on the … Sprint Backlog

Create Plan
- Sprint Backlog
- Release Backlog
- Product Backlog
- Cross Project Plan

- Tasks planned for the iteration

- Plan **Views**
  - Planned Time
  - Ranked List
  - Roadmap
  - Taskboard / Kanban
  - Work Breakdown

Planned Items ? | Links | S...
☐ ▾ | View As: Work Breakdown ▾
Planned Time
Ranked List
Roadmap
Taskboard
Work Breakdown

Product
Release
Iteration
Day
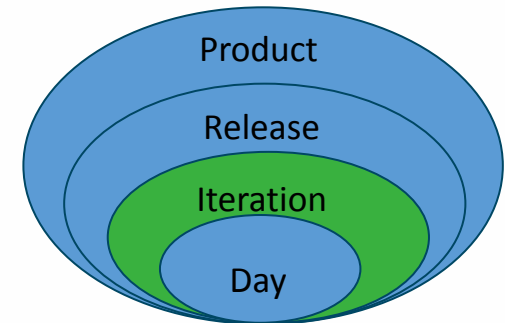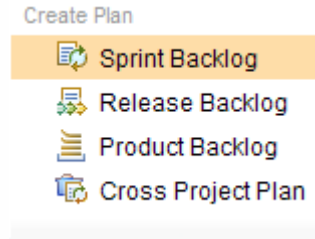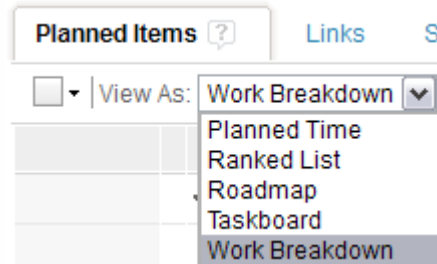
- **WorkItem Types**
  - Execution items: Work items which have work assigned.
  Examples: Task and Defect.

  Defect
  Task
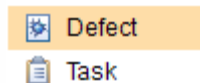
- **Reports**
  - Sprint Burndown

  Estimated hours required to complete remaining work items.

IBM

Planning a Sprint

- Focus on **Execution**: large stories broken down into smaller sub-stories and execution items
- **Fine grained:** Estimation of effort: typically estimated in **Ideal Hours/Days**
- **Custom view:** Can the execution task relate to the **business value** expressed on the product backlog?
- **Estimates** are important

A Day in the Life of an RTC Developer

- **My Work View**: Developers plan their assigned work and adjust estimates
    - Managing new work
    - Managing current work
    - Managing future work
    - Work Load
    - Find, filter, and colorize planned work items
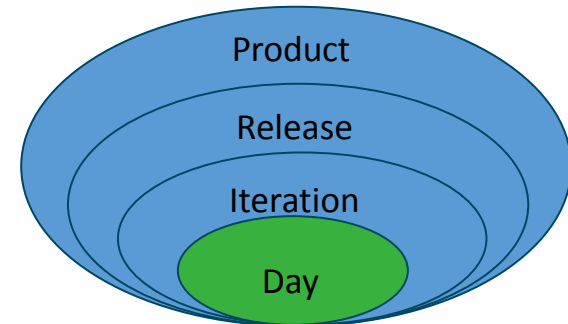
- Daily **Stand ups**: Developer **Task board** view to see what people are working on.

- **Planned Time**: Developers sort the work items in the order in which they intends to work on them

- **Bottom Up** Planning

My Work View / My Dashboard View

Daily Planning – stand-ups

- **Taskboard view:**
  - Shows plan items in the left column and execution items in the state group columns.
  - Visualize the progress of the stories and their children.
  - Ability to adjust the state quickly with drag and drop from this view.
  - Color tags can be used to show the state of the plan items.

Defining an Adaptive, Agile Process

Process Rules!

- Teams work on projects
  - each project follows a **process**
  - different processes for different projects

- Process manifested through:
  - artifact types and their states
  - preconditions and follow-up actions on operations manipulating artifacts
  - artifact change events
  - roles and permissions

- Described with **Process templates**

- Process can evolve as project progresses

- **Automatic Process enforcement!**

Problem
A work item must be associated with the change set or a comment must be set.

Reason
All change sets should be associated with a work item which is planned for the current iteration.

This makes it easier for the team to track its progress through the iteration and... **Deliver** (failed)
why your changes were... ⊗ Missing work item or comment

Solutions
💡 Associate Existing Wo...
💡 Associate New Work Item
💡 Associate and Try Again (experimental)
↷ Overrule 'Descriptive Change Sets' Precondition

NO
LOITERING
ALCOHOL
BIKE RIDING
PEDDLING
PANHANDLING
BACK–IN–PARKING
NIGHT PARKING
DOGS WITHOUT LEASH
LOUD STEREOS
CAR REPAIRS

IBM

Process Templates

- **Different agile templates available:**
  - Agile
  - Scrum
  - OpenUP
  - Eclipse Way

- **Agile process specific work item types**

- **Process can be tweaked at any time**

IBM Technical Summit
Stay ahead

IBM
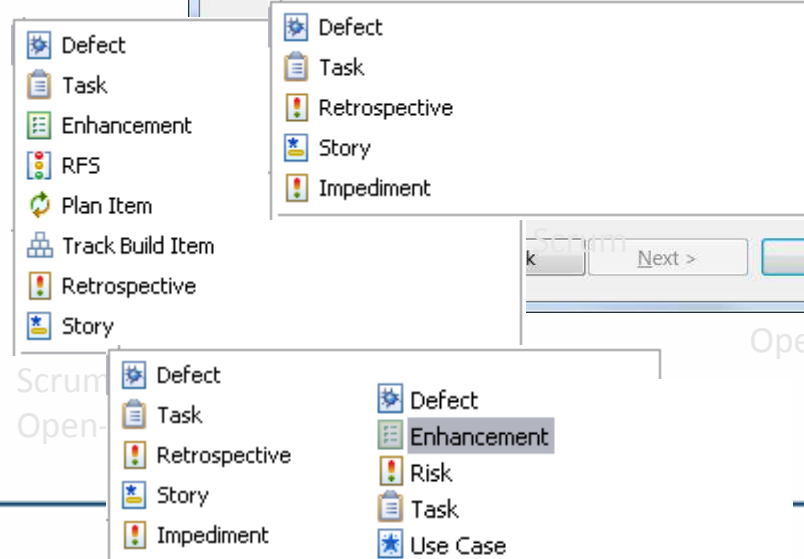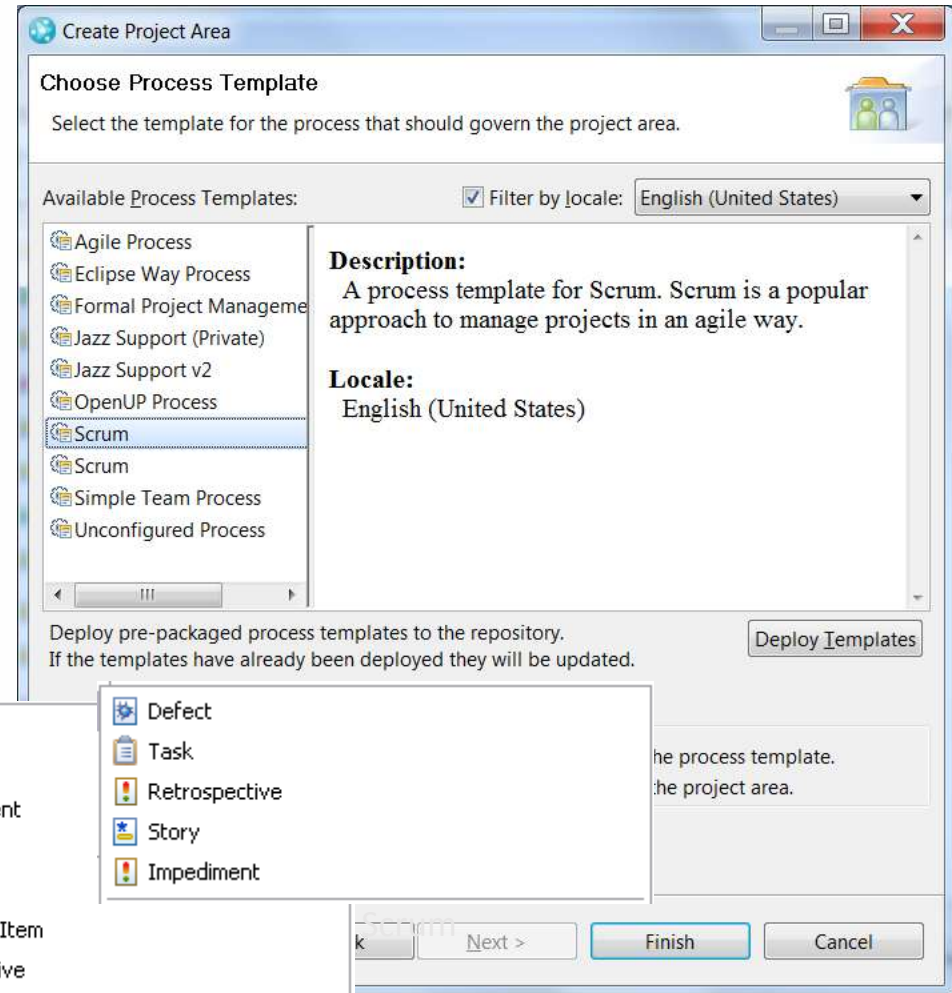
Parallel Development with SCM

- RTC SCM supports several levels of parallel development
  - Team Stream on Server
  - Personal Repository Workspace on Server
  - Sandbox, locally on Client system

- Collaboration based on Change sets:
  - Between Team Stream and Personal Workspace
  - Between sibling Personal Workspaces

- Traceability
  - Change sets get attached to work items
  - Get included in builds
  - Reviews
  - Locate, Search
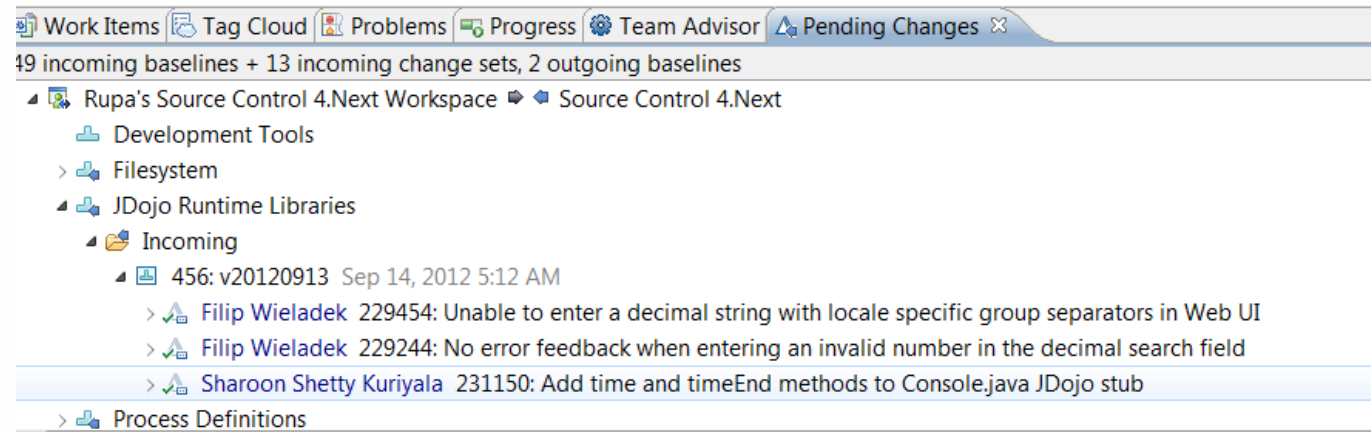
Isolation Levels – Isolate Work not People

- Repository workspaces – Provides individual isolation
    - You don't have to make your changes visible to the team to:
        - Backup or use the repository features.
        - Collaborate on a change with someone else

- Streams – Provides team isolation.

- Suspend and Resume – Provides task level isolation for personal work.

- Team areas – Provides process isolation.

Continuous Integration with builds

- Team of Team
  - Scheduled integration builds
    - Cross-component
  - Continuous integration builds

- Each Team
  - Continuous builds
  - Always green

- A Developer
  - Personal builds

- Unit tests

Transparency with Feeds, Dashboards, Reports

- Team event notifications
  - Build events
  - Source Control changes
  - Work item changes
  - Team Process changes

- Dashboards
  - Live project status
  - For all stakeholders

- Reports
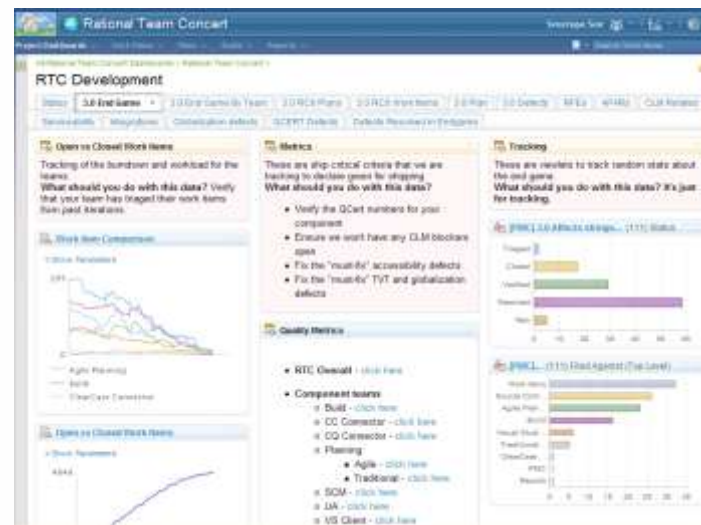
Retrospectives

- Teams reflect on what worked and what didn't

- How to tune the process

- PMC retrospective summarizes team retrospectives
  - Defines actions for the PMC

Best Practices for Agile Delivery

- Work off of a common backlog

- Have 3-6 week iterations

- Maintain one base for all your source code

- Update your work items regularly

- Automate the build

- Self-testing builds

- Commit often

- Keep your builds fast

- Upload your executables as build artifacts

- Make it transparent

www.ibm.com/software/rational

www.ibm.com/software/rational

IBM Technical **Summit**

> Stay ahead.

IBM