

Agility at Scale

...aneb RUP v agilním světě

Jan Valdman, Ph.D.
jvaldman@dns.cz

Business Development Manager – IBM SW
DNS a.s.

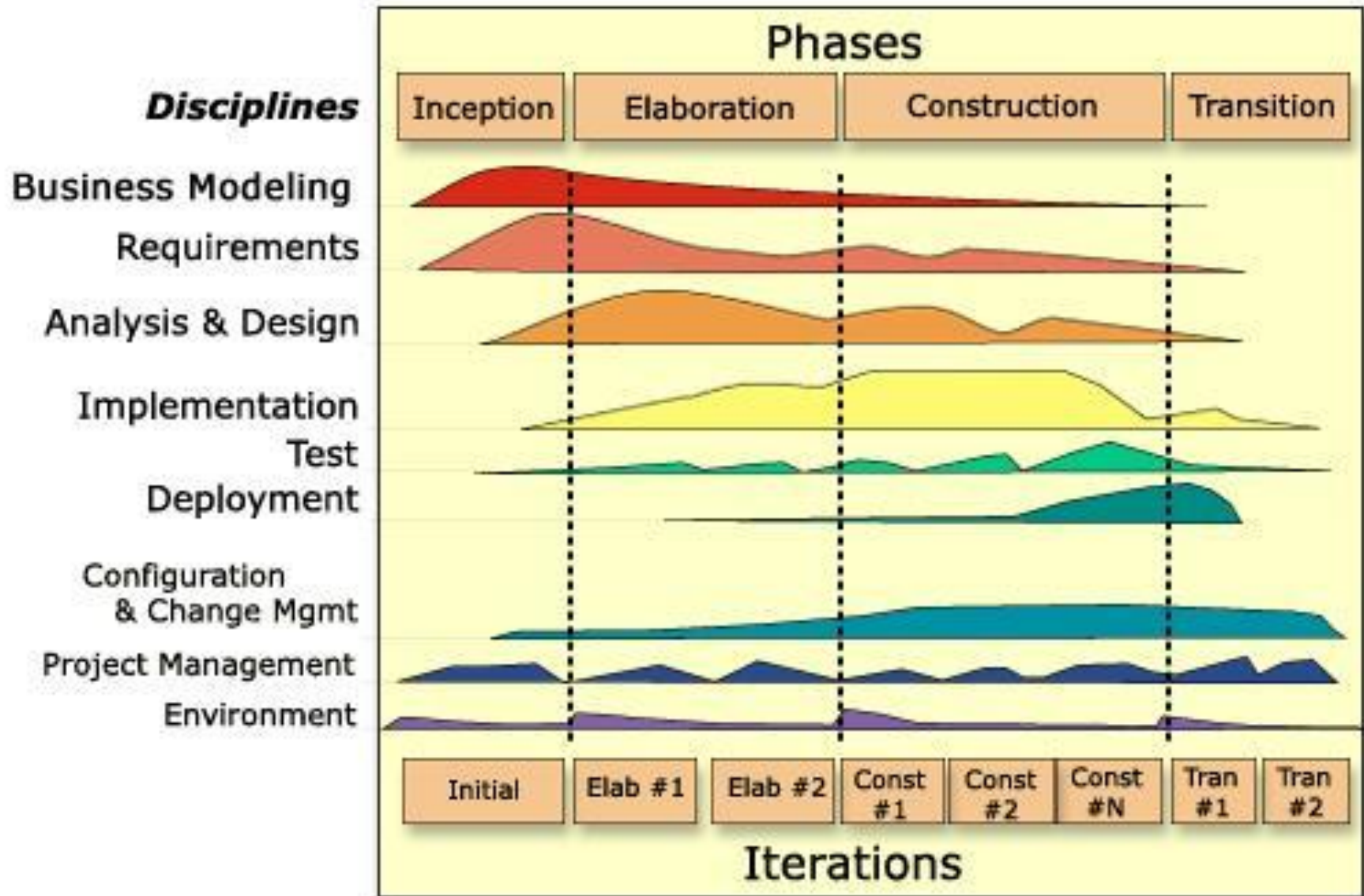
13.10.2010

- RUP
- „agile“
- Agility at scale
- Rational Team Concert, Rational Insight
- MCIF

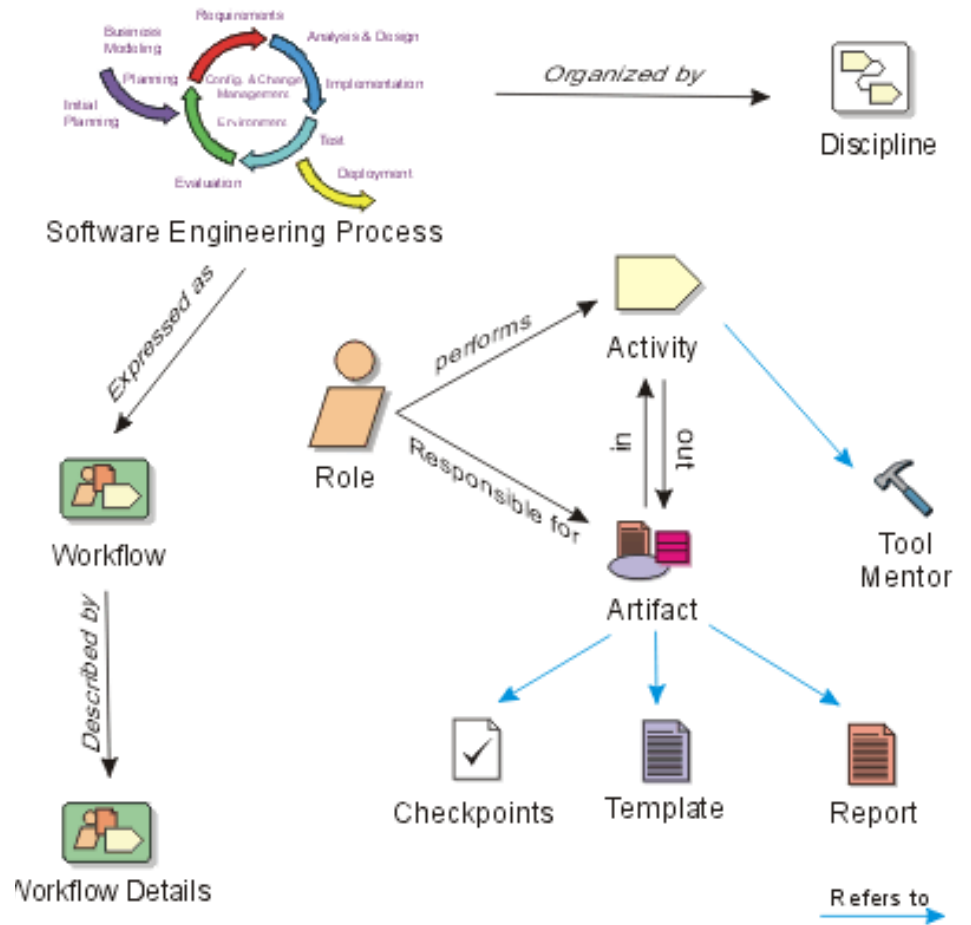
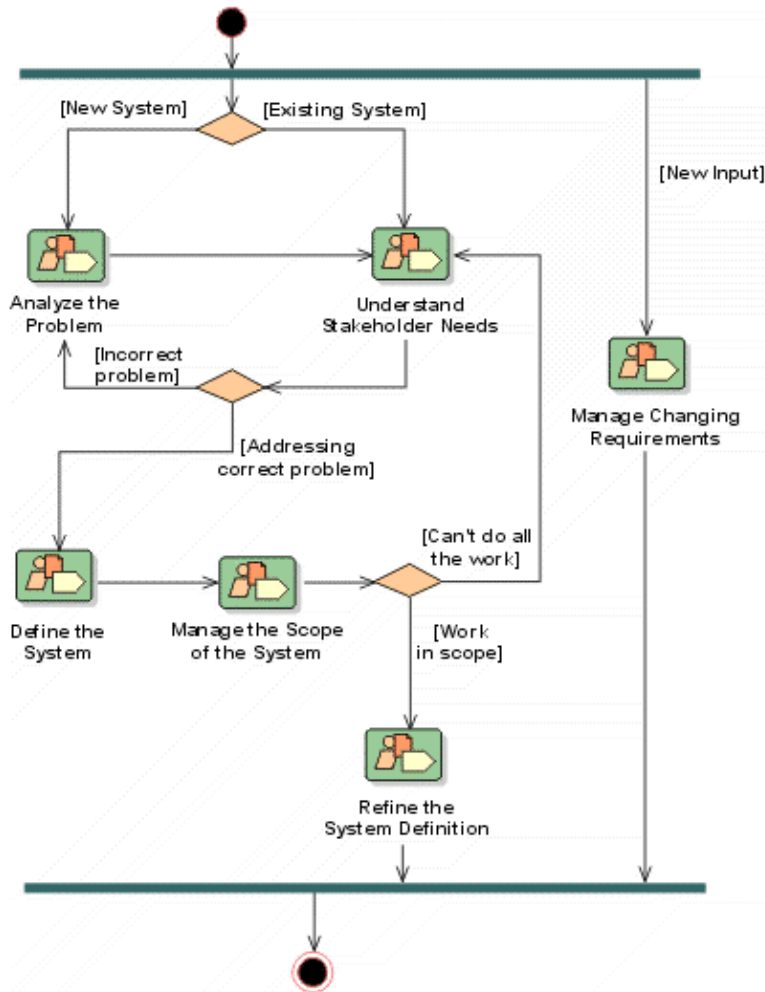
Rational Unified Process – RUP



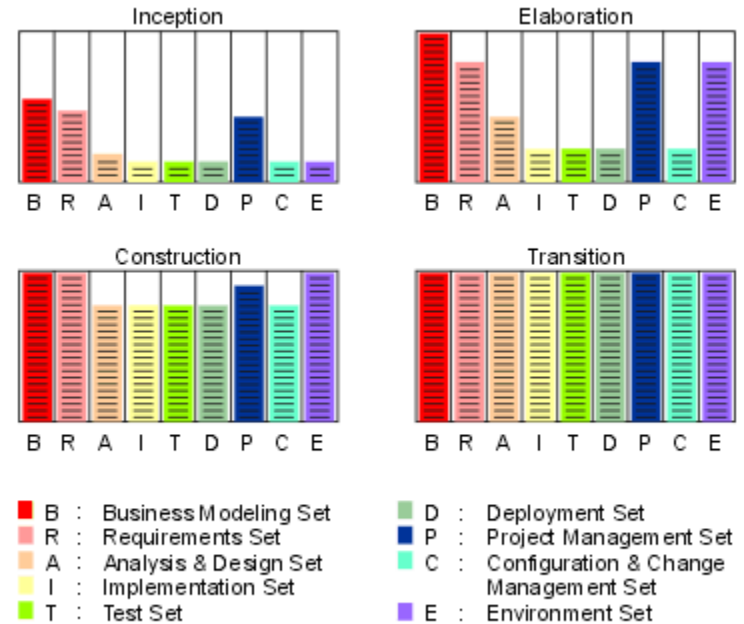
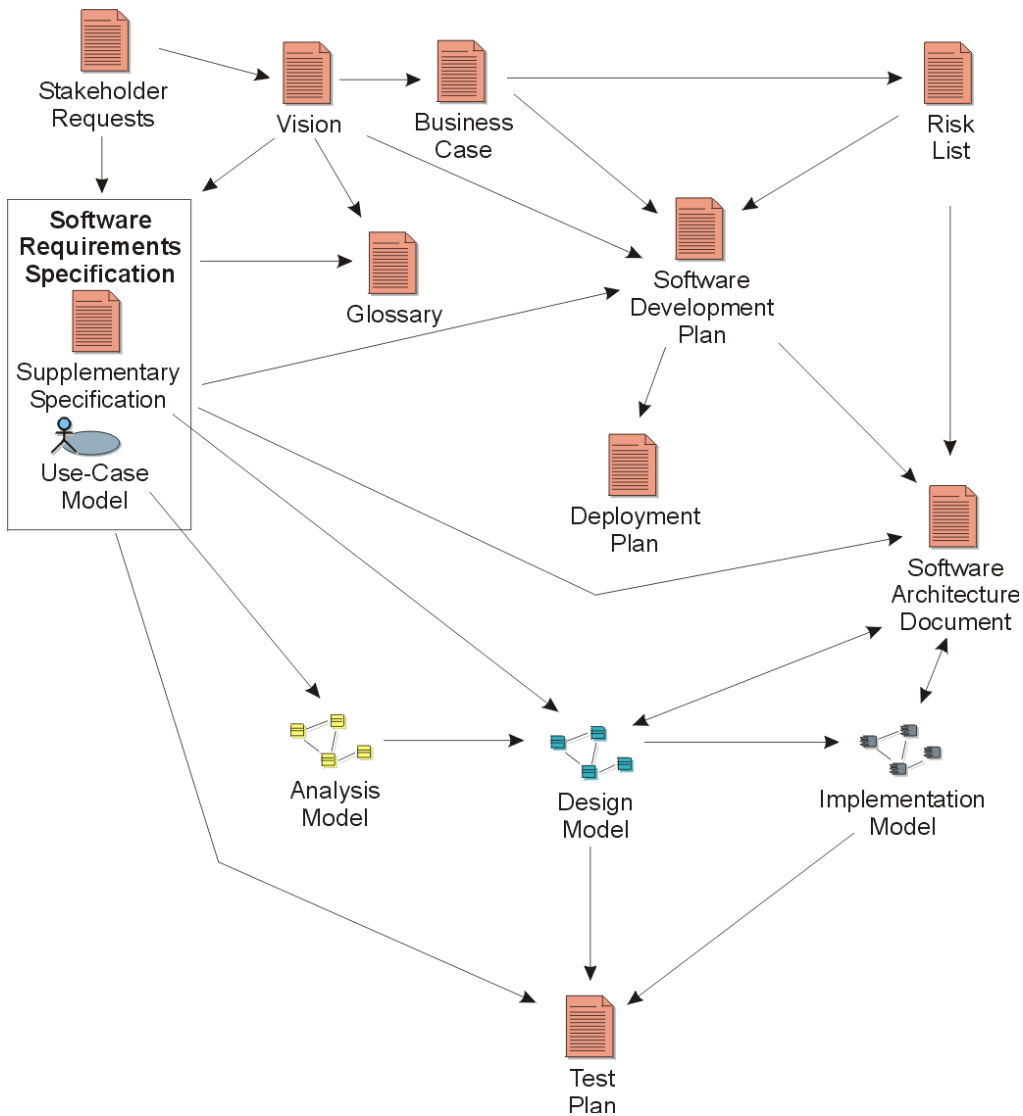
Rational Unified Process - RUP



RUP je formálně popsán...



RUP zná cca 80 artefaktů



Information set evolution over the development phases.

Efektivně!

ΕΓΚΛΙΝΟΥΣΙ

Reuse



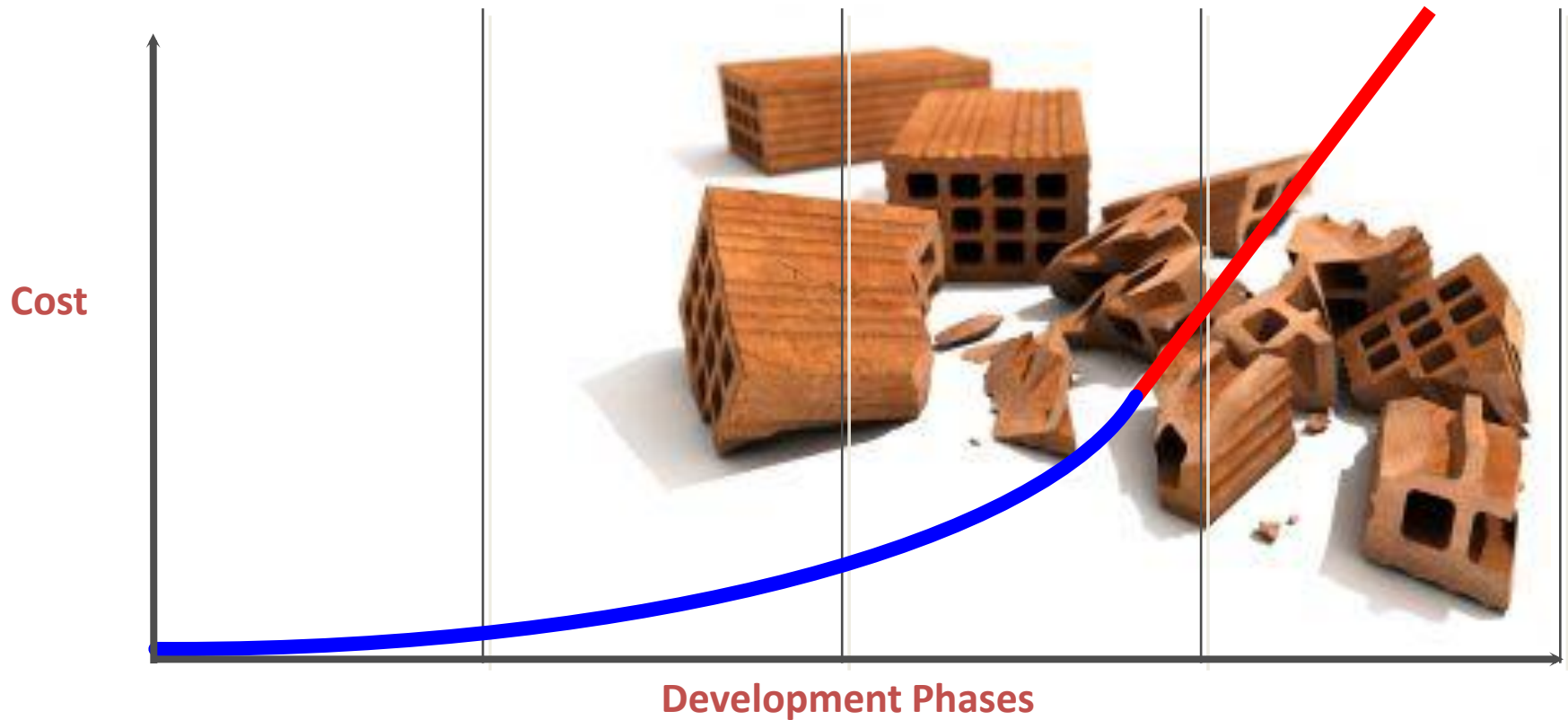
Reduce complexity



Model Visually



Continuously Verify Quality



BEST PRACTICE



Manage Change

Hlavní principy (Concept)



Adapt the Process

Balance Competing Stakeholder Priorities

Collaborate Across Teams

Demonstrate Value Iteratively

Elevate Level of Abstraction

Focus Continuously On Quality

RUP Tailoring

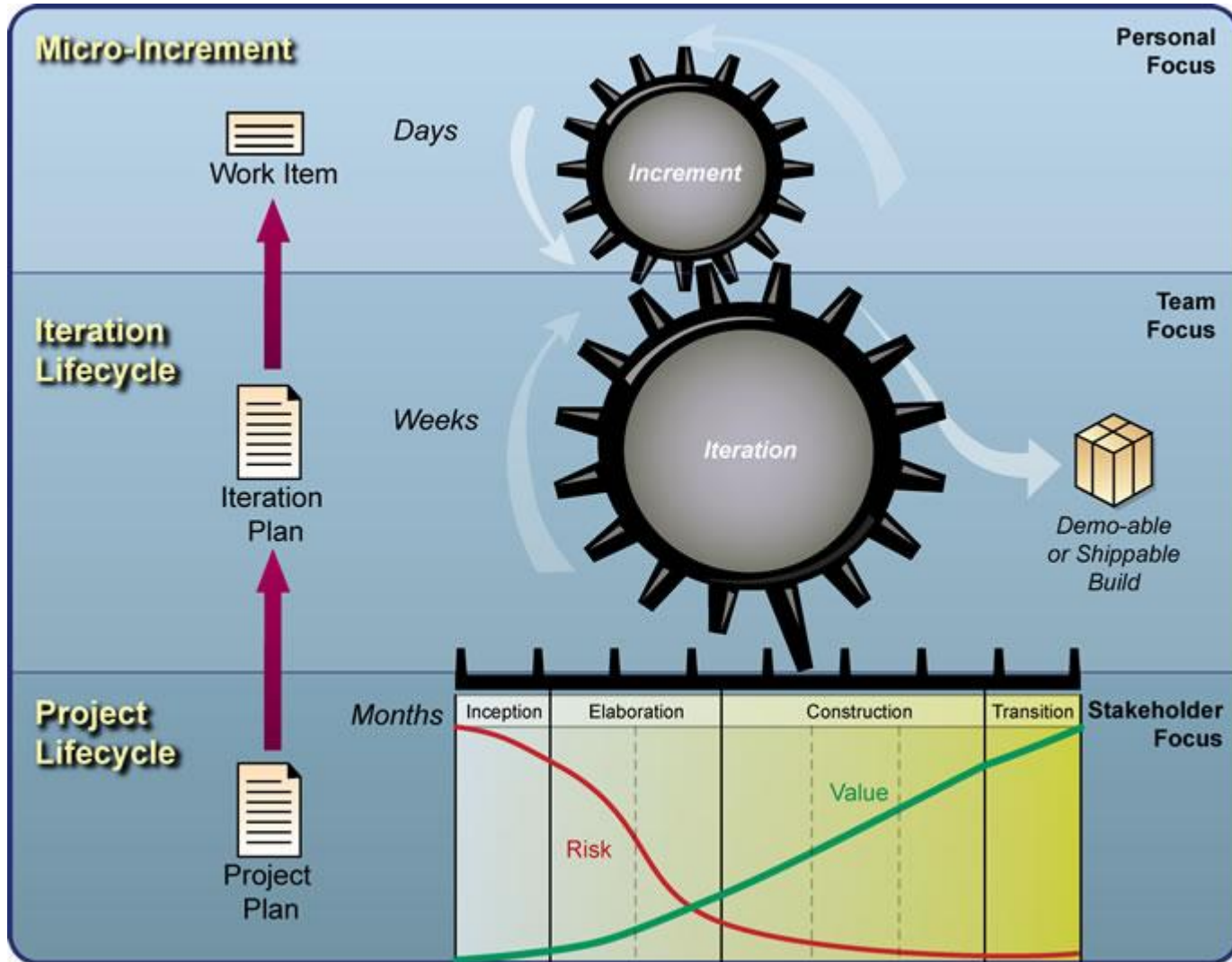
A man in a dark pinstriped suit is seen from behind, sitting on a concrete ledge. He is looking out over a large body of water. In the distance, a city skyline is visible on a hillside, and a large dark blue ship with "MSC" written on its side is docked. The sky is clear and blue.

Small – medium – large
Process library

ADOPT THE PROCESS !!!

Tak kde je problém???





„Agile“



Rigorózní metodiky:



plánovat – měřit – řídit

Agilní metodiky:

Vývoj software je
„empirický proces“

CHAOS

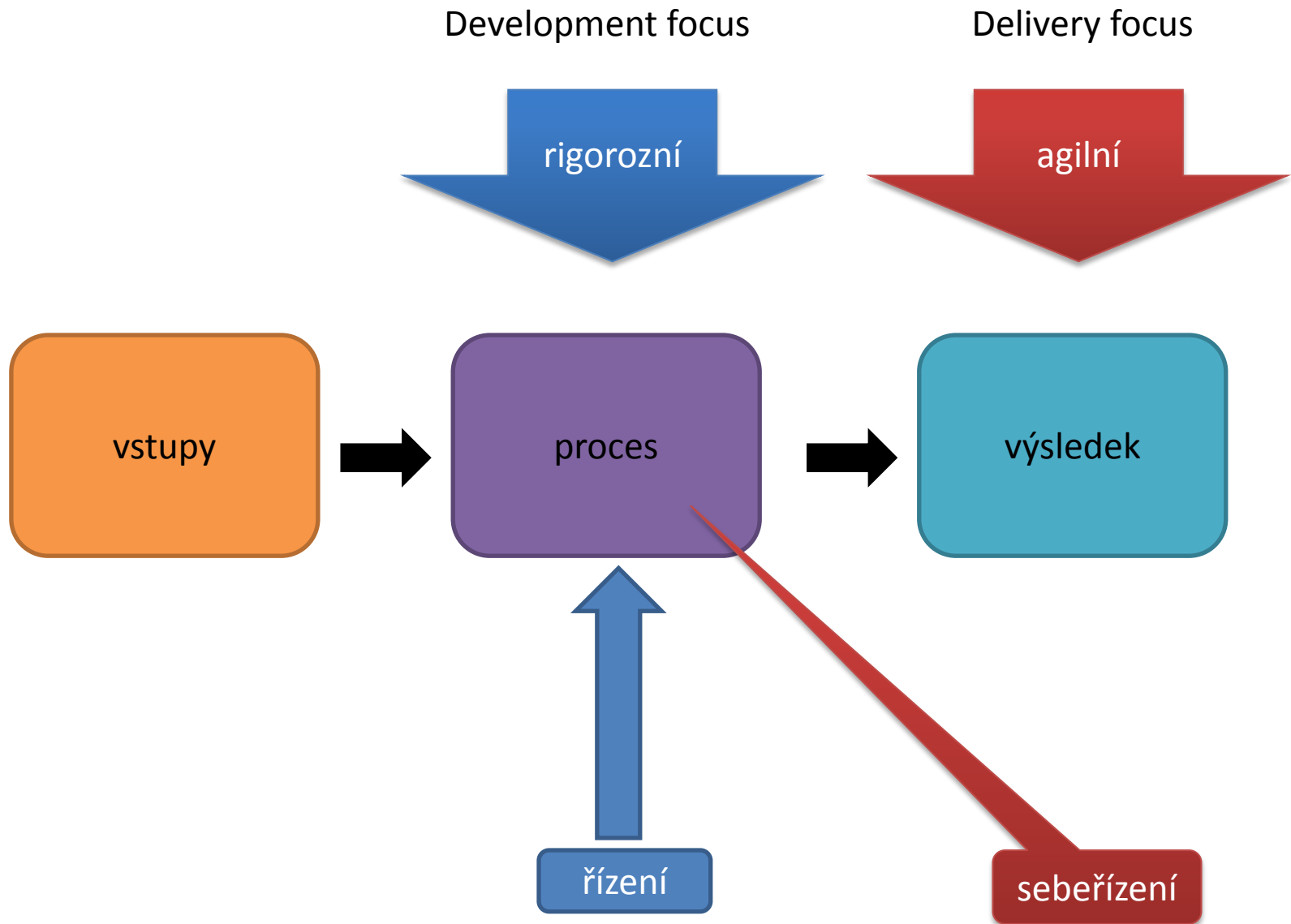


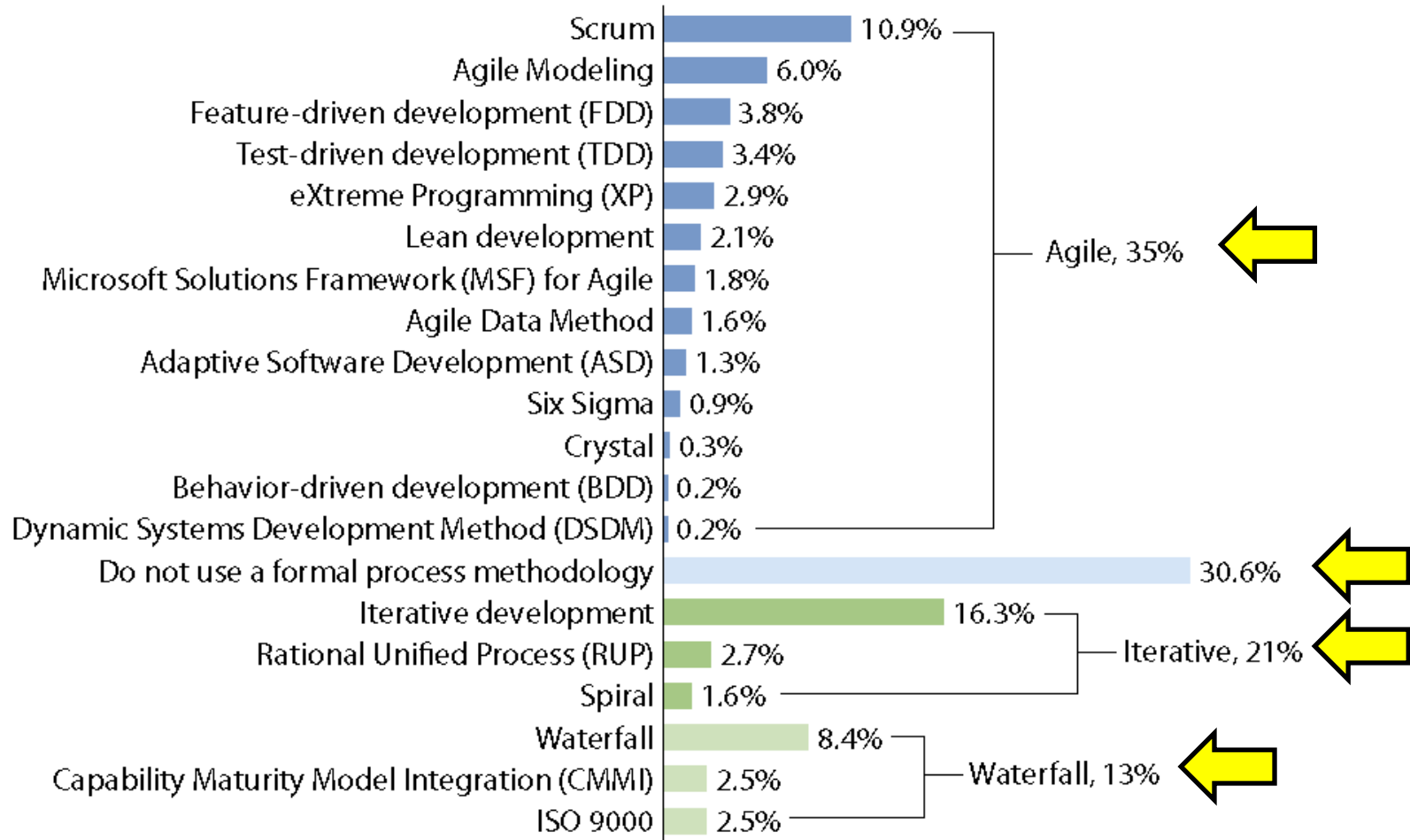


rigorózní

X

agilní

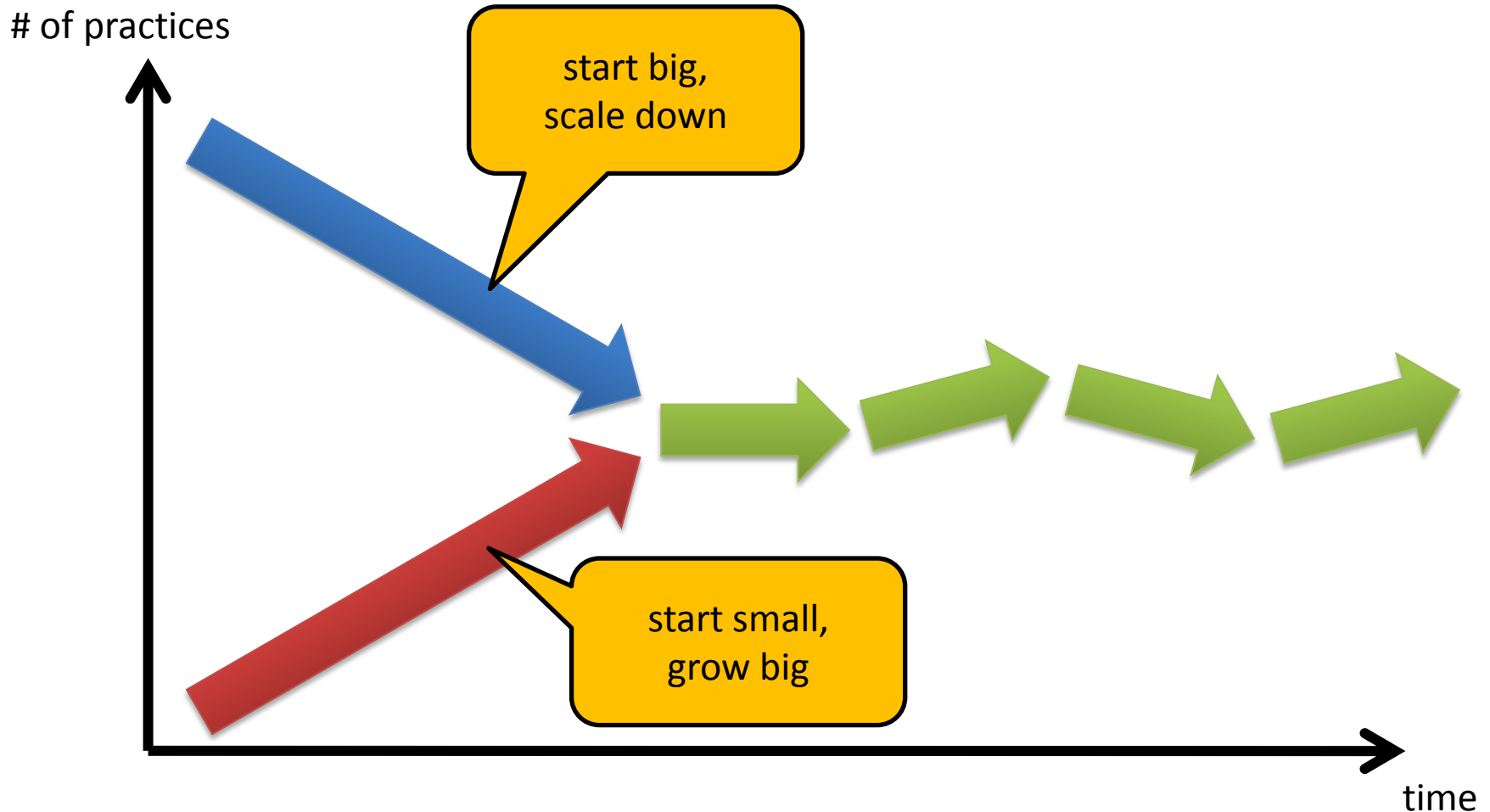




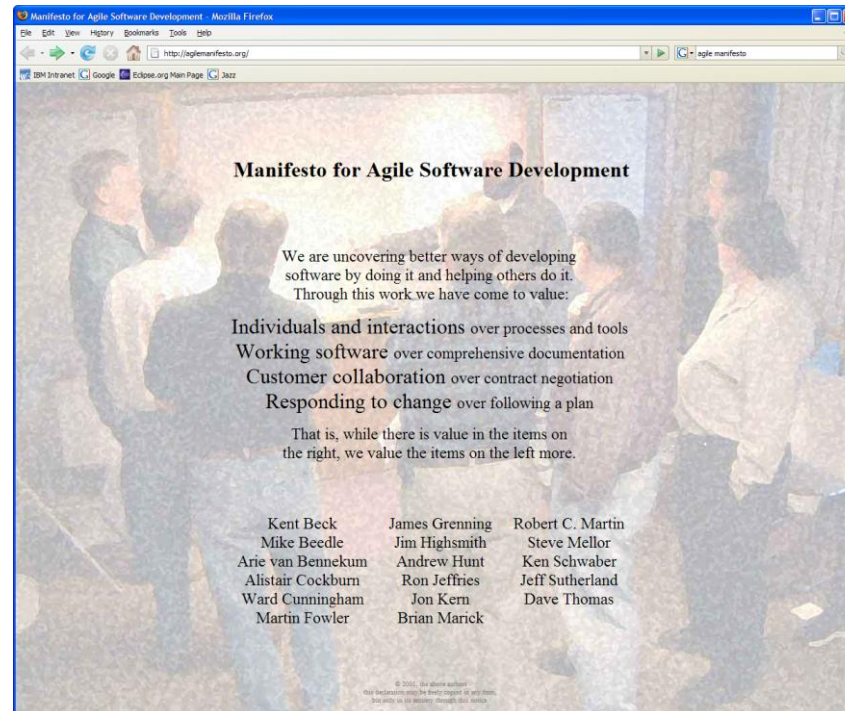
Base: 1,298 IT professionals

Source: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009

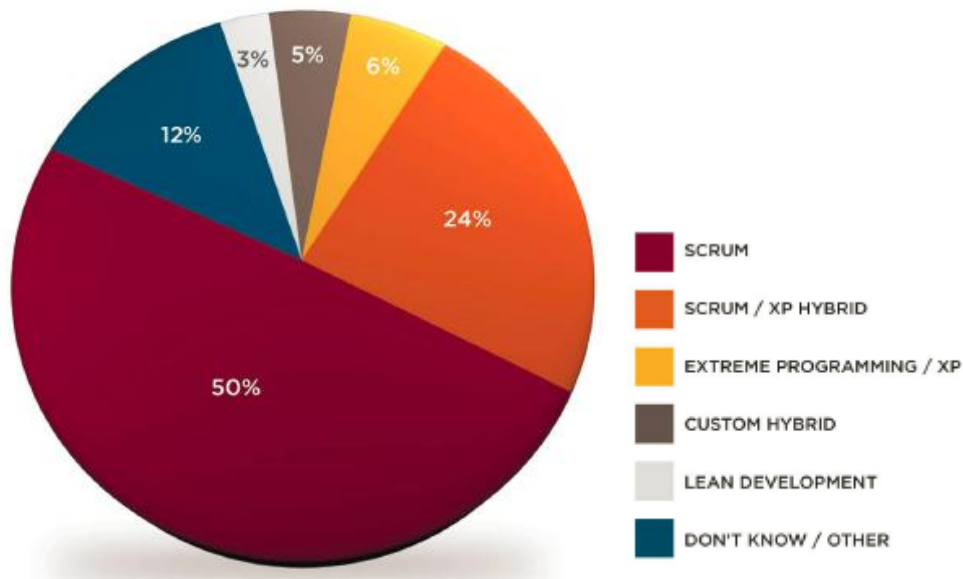
- Úspěšnost agilních a iterativních metodik je přibližně stejná, ale agilní metodiky dodávají dříve



- Agilní manifest (4 hodnoty, 12 principů)
- Vývojáři jsou v centru dění (tj. oni vlastní sw. proces)
- Omezení „paperwork“ a zlepšení komunikace výrazně pomáhá

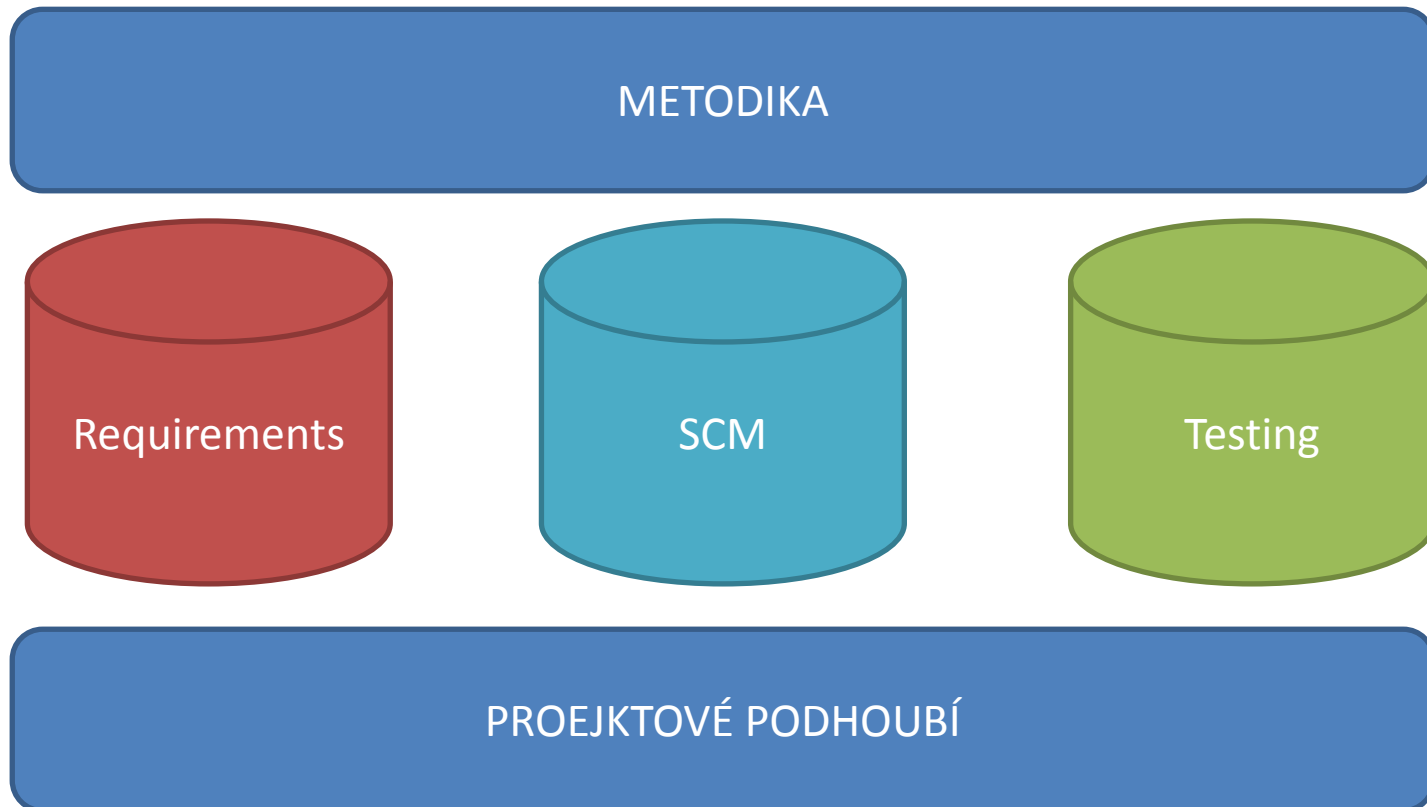


- Funguje, ale používá se prakticky jen SCRUM a XP
- Buzzword
- V reálu několik týmů/projektů – nikoliv plošná adopce
- Široká škála situací a projektů
- Objevují se nové problémy (viz dále)



64% SAID
FASTER
TIME TO COMPLETION

- Používá se celá řada praktik
- Nejčastěji „používáme modifikovaný SCRUM“



- Malý projekt = 1-2 mil Kč, 150 MD, 6 měsíců
- 6 měsíců = 130 pracovních dnů
 - Velikost týmů 5-6 lidí
 - kdo to řídí a zodpovídá za projekt - PM
 - kdo zjistí detaily zadání - analytik/konzultant
 - Kdo vymyslí jak architekt/designer
 - Kdo to rukama udělá kodér/programátor
 - Kdo zajistí kvalitu QA/QE/tester
 - Kdo to nainstaluje a nakonfiguruje IT - administrátor
 - Kdo tomu dá vzhled – grafik
- Nutná profesionalita -> specializace, efektivní využití zdrojů
- Přítomnost specialistů jen v konkrétní fázi projektu

- Nedisziplinované týmy
- Znalost 1-2 článků
- Novátorská, svobodomyšlná a nedokumentovaná softwarová kreativita

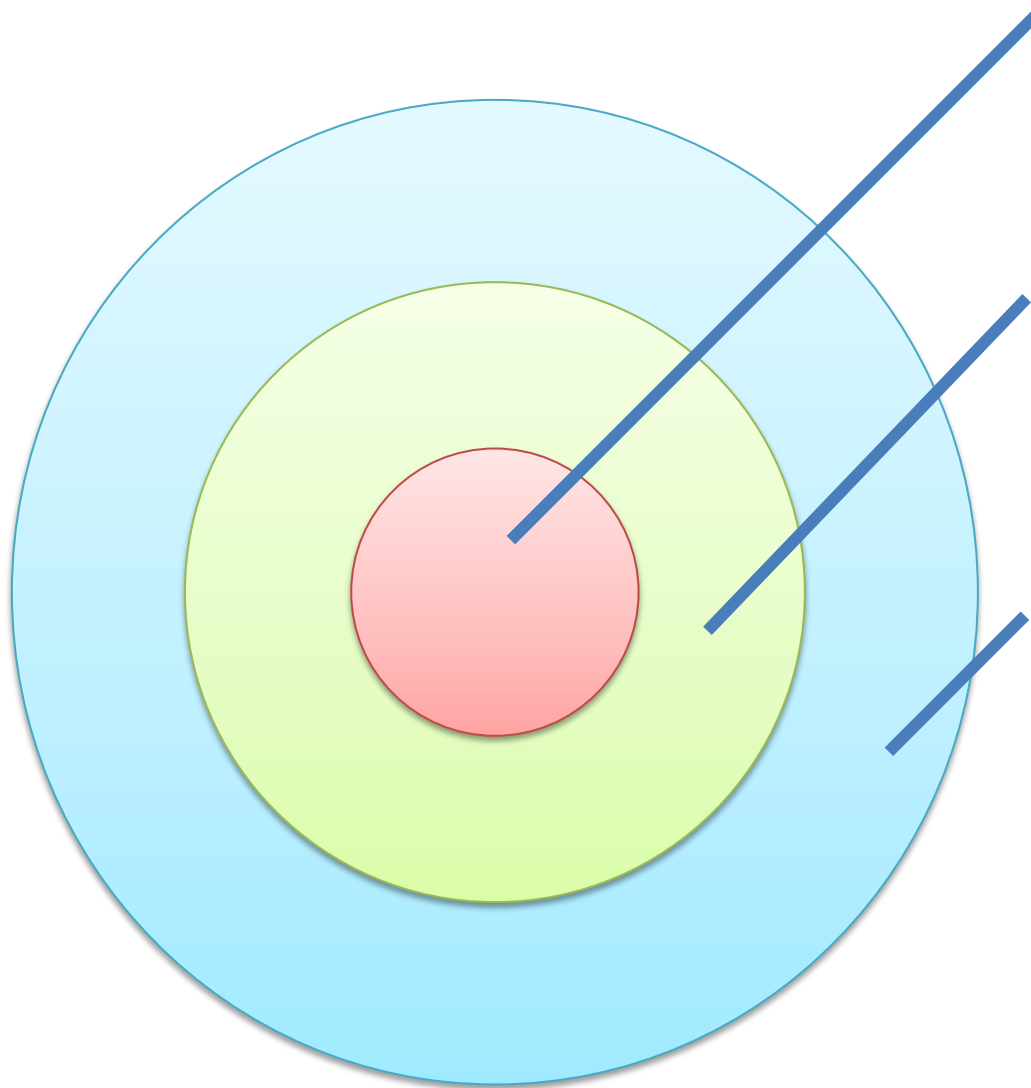
- ...existuje možnost ověřit, případně sada doporučení

Agility at Scale



- Sdílení statusu je časově náročné
 - Vyplňování wiki
 - Reportování subtýmů
- Mnoho praktik potřebuje automatizaci
 - Management post-it papírků
 - Nedostatek tabulí
- Retrospektivy vyžadují informace
 - Jak sebrat podklady
- Různé situace vyžadují různé praktiky
- Filosofické problémy byznys – IT

- Velikost týmu 10-20 osob
- Geografické rozložení týmu jedna místnost
- Regulace a compliance ISO, CMMI
- Složitost problému velký problém
- Organizační bariéry divize, partneři
- Technická složitost API, DB, codebase
- Firemní kultura vodpád, více agile
- Podniková architektura EA, strategie



Core Agile Development

- Value drive lifecycle
- Self-organizing teams
- Focus on construction

Disciplined Agile Delivery

- risk+value driven lifecycle
- Self-organizing w. governance framework
- Full delivery lifecycle

Agility at Scale

- Disciplined agile delivery when scaling factors apply
 - Large team
 - Geographic distribution
 - Regulatory compliance
 - Domain complexity
 - Organization distribution
 - Technical complexity
 - Organizational complexity
 - Enterprise discipline

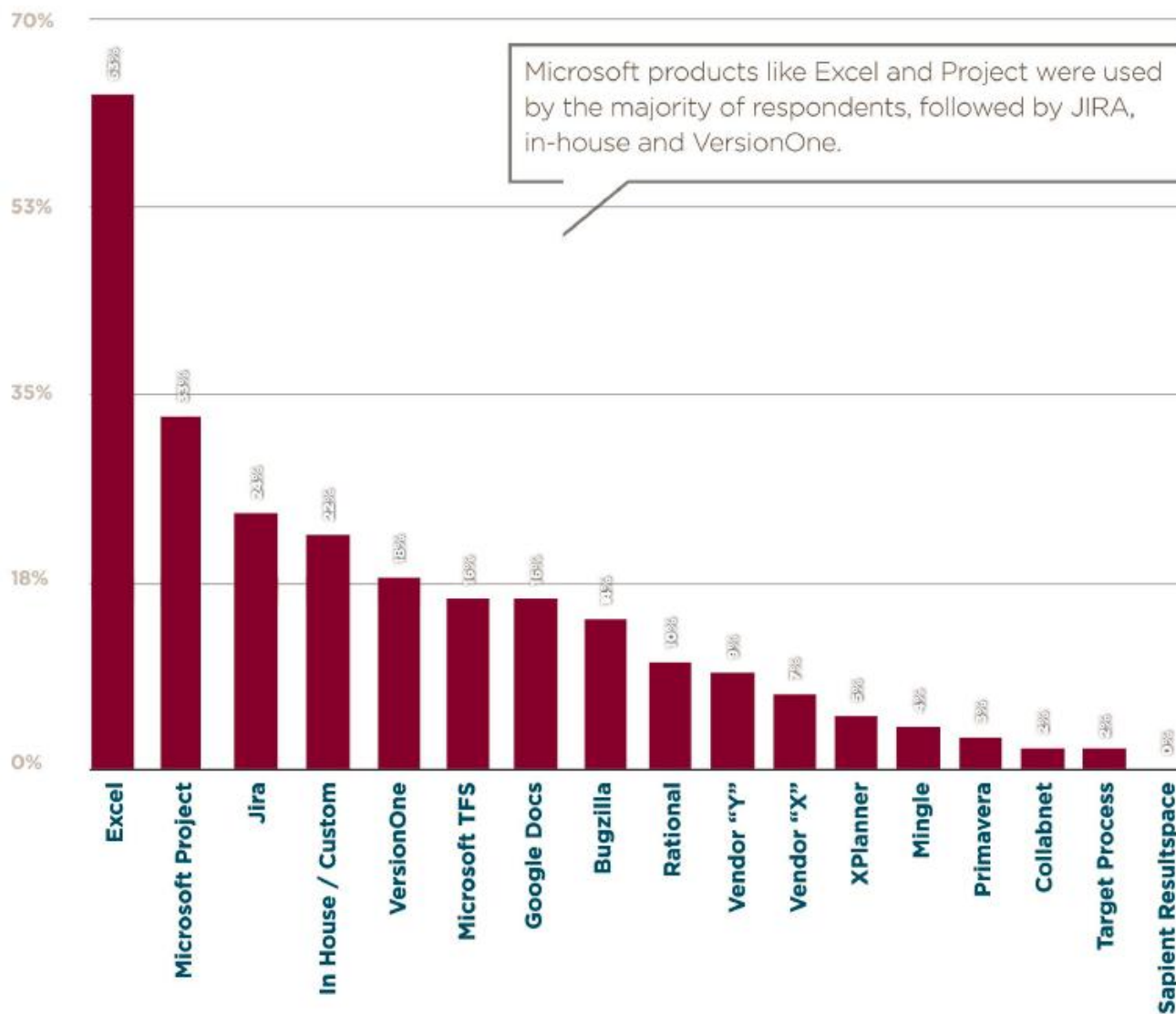
- Agilní jádro
 - Tradiční agilní praktiky, XP, SCRUM
 - Některé činnosti nebo etapy
- Disciplinovaný agilní delivery proces
 - Plný rozsah metodiky
 - Dodávají řešení, ne jen software
 - Řídí rizika i hodnoty
 - Governance framework
- Agility at scale
 - Scaling faktory...
 - Automatizace
 - instrumentace

Agile Core	Disciplined Agile Delivery	Agility at Scale
Agile Data (AD) Agile Modeling (AM) Extreme Programming (XP) Feature Driven Development (FDD) Scrum	Agile Unified Process (AUP) Agile With Discipline (AWD) Eclipse Way Rational Unified Proces (RUP) Open Unified Process (OpenUP)	Enterprise RUP (EUP) Rational Measured Capability Improvement (MCIF)

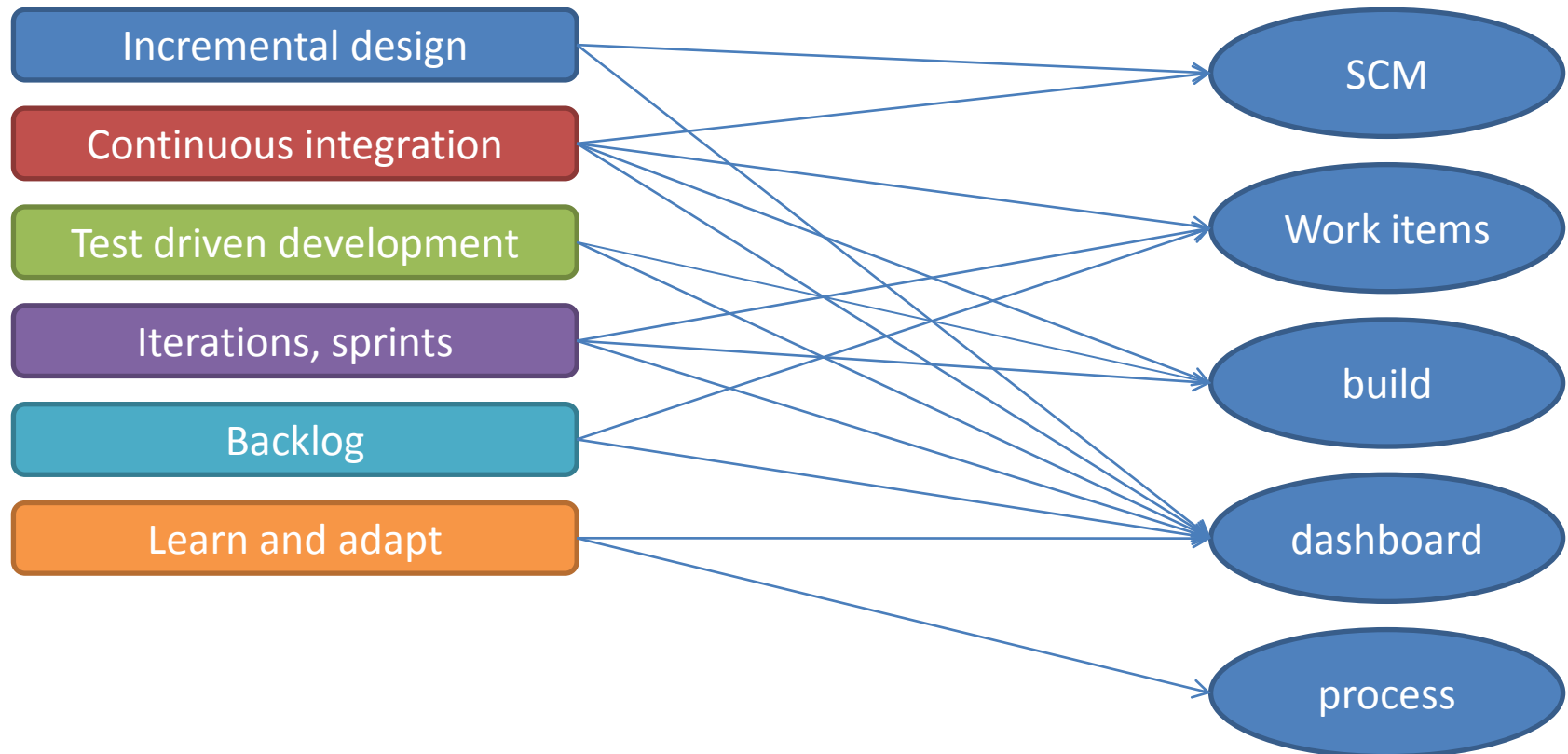
Nástroje pro agilní development



Jaké nástroje se používají?



- Každý větší i agilní projekt si potřebuje uchovávat různé artefakty ...
- Agilní vývoj ve velkém potřebuje dokonce lepší nástroje...



- Traceability a vztahy mezi artefakty
- Automatizace high-level procesů
- Přehled (visibility) o postupu development procesu

- Team awareness
- Build awareness
- Project awareness

Nejlepší nástroj (compiler, linker, tester...)



Nejlepší IDE (integrace pro jednotlivce)



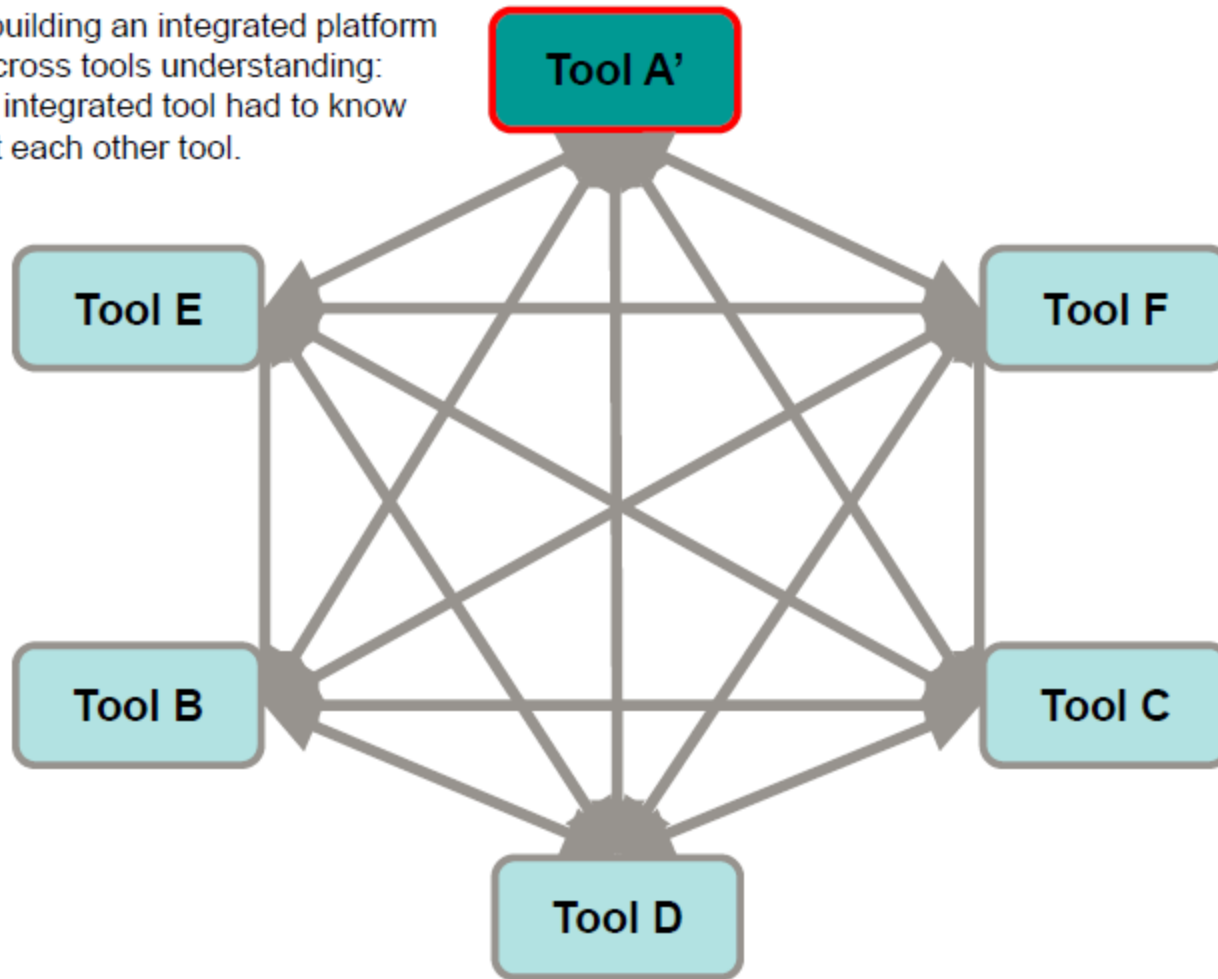
Nejlepší prostředí pro tým

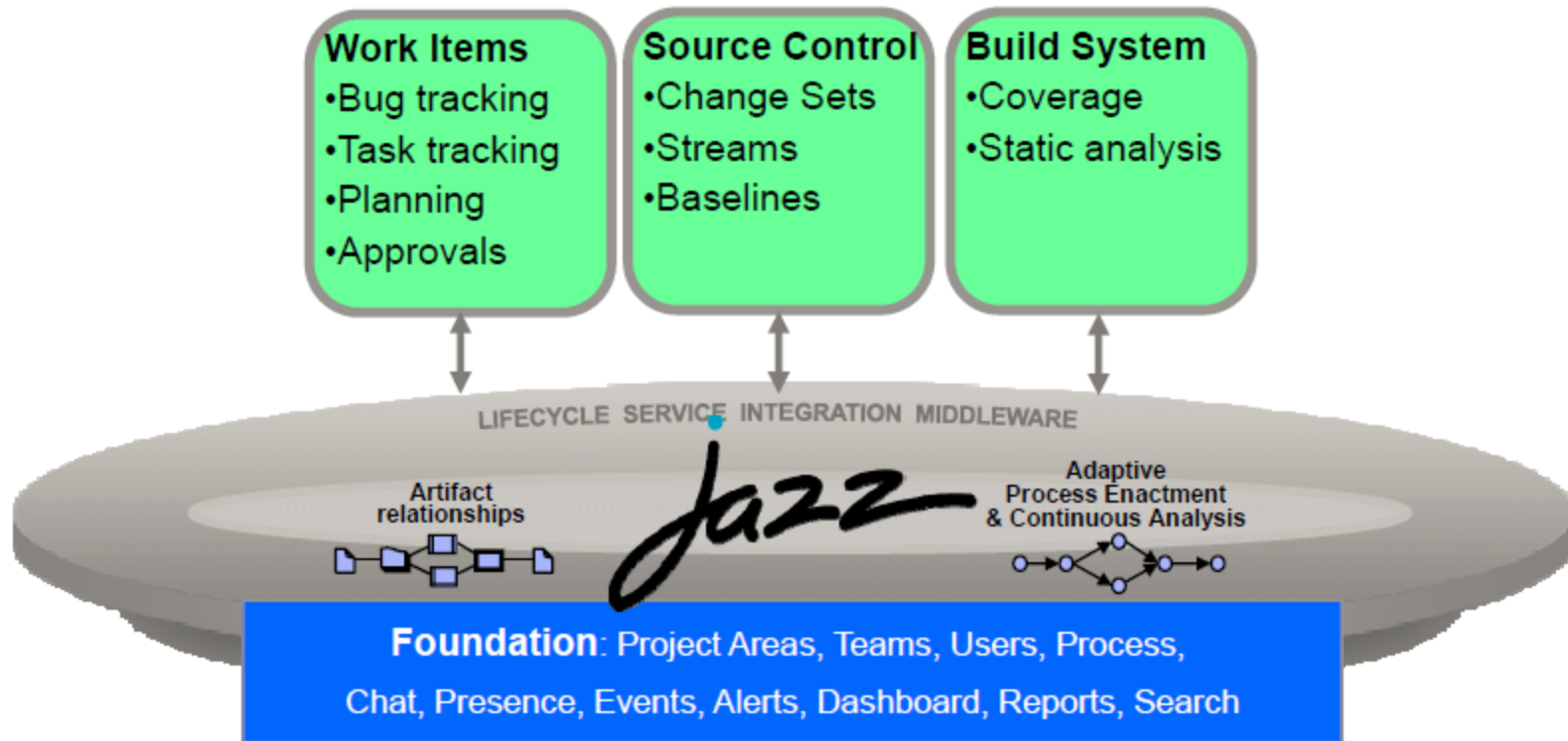
Desktop integration -> Lifecycle integration

Function first -> Team first

Manual -> Process aware

Until now, building an integrated platform required a cross tools understanding:
Each integrated tool had to know about each other tool.





- Open Services for Lifecycle Collaboration

Source Code Management

The screenshot displays the Team Concert interface with several key components highlighted by callouts:

- Flow browser to show how the changes flow from the developer to the team.** (Top right callout)
- Create project level baselines i.e. snapshots** (Bottom left callout)
- Version tree browser** (Bottom center callout)
- Component level baselines and history** (Bottom right callout)

Merges	Comment	Creator
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	20573: Design replacement for vtime ε...	Dmitry Karasik
	27033: IChangeSet.getComment() ret...	John Camelon
	26894: Remove IScmService deprecate...	Carol Yutkowitz
	26747: reduce deprecated usage in sc...	John Camelon
	26669: Data model constraints differ f...	John Camelon

Component	Time	Version
Blues	1/23/08 7:26 PM	7: v20080123-Traceability
	1/23/08 7:21 PM	6: workshop.squawk.core.continuous.build_20080123-1621
	1/23/08 6:47 PM	5: v20080123-M2StudentSquawkers
		4: v20080123-M1-Complete
		3: v20070123a-M1-Complete
		2: v20070123-M1-Complete
		1: Initial Baseline

Support for discussions around

Attach files and Screen shots

SS Feed support

Support Approv

view

ClearQuest Connector

team men

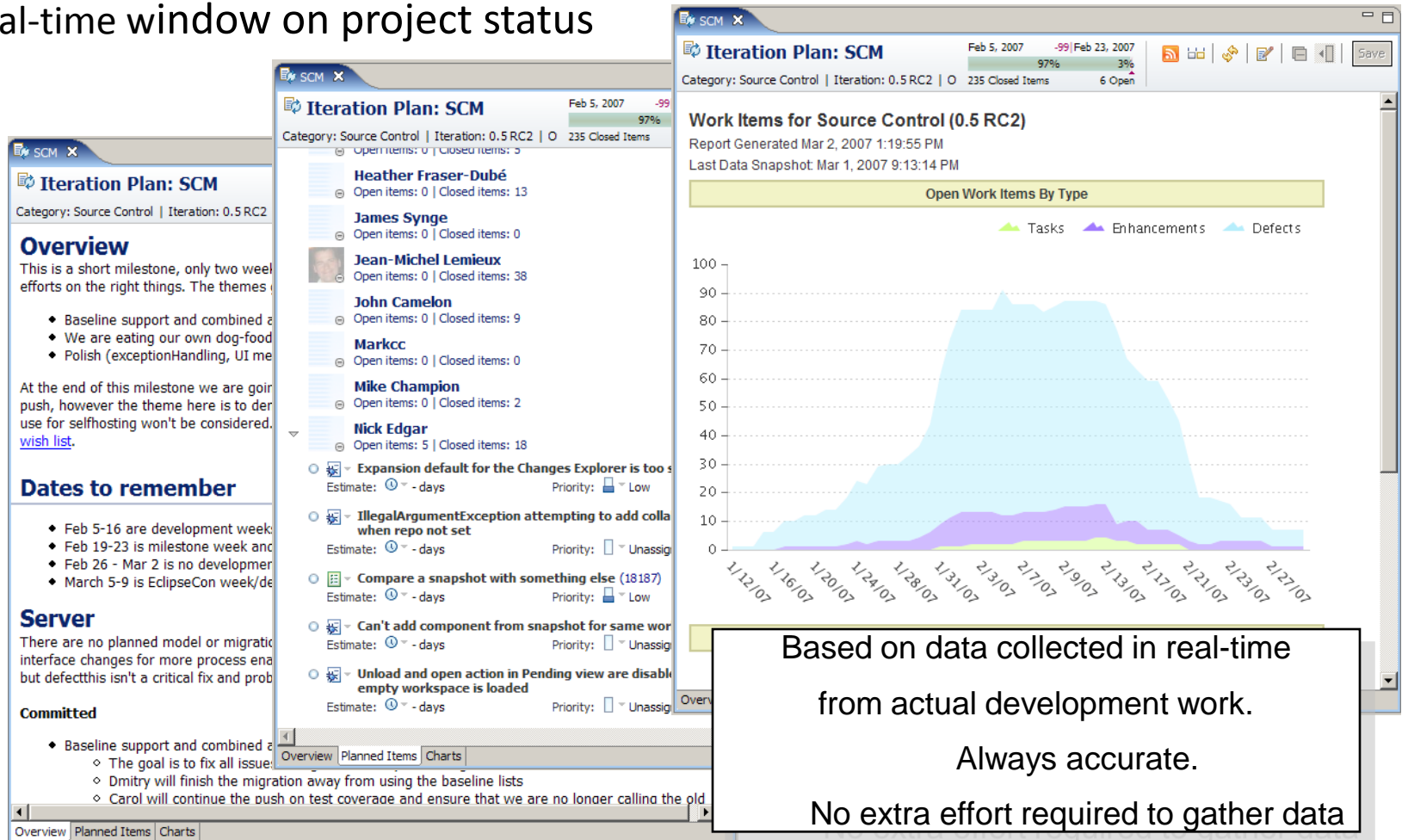
The screenshot displays a software interface with a 'Work Item Query' window. A dialog box titled 'External Repository Connection Properties' is open, allowing users to edit the properties of an external repository connection. The dialog includes the following fields and options:

- Name:** ClearQuest SAMPL
- Connection info:** http://localhost/InteropGateway/InteropServlet
- User ID:** [Empty field]
- Password:** [Empty field]
- Disable outgoing synchronization

The background interface shows a task list with columns for 'Id', 'Name', and 'Status'. A table at the bottom lists work items with IDs 26 and 21, and statuses 'Resolved' and 'Resolved'. The status bar at the bottom indicates '<No Current Work>'.

Iteration Planning and Execution

- Integrates planning with execution
- Easily see what needs to be done, what is being worked
- Real-time window on project status



Create build definitions for team and private builds

Supports Ant and Command line build tools

Create local or remote build servers

Identify work items and change sets that went into the build

Historical view of the Build Queue with Status

The screenshot shows a web-based build management interface. On the left is a tree view of 'Team Artifacts' with 'Builds' expanded. The main area displays details for a build: 'Build workshop.squawk.core.continuous.build B20080303-1042-workshop.squawk.core.continuous.build'. It shows a 'Completed' status with a duration of 27 seconds, start time of March 3, 2008 1:41:40 PM, and completion time of March 3, 2008 1:42:07 PM. A 'Status Trend' bar shows a sequence of green and red blocks. Below this is a 'Contribution Summary' with links for 'Changes', 'Logs', 'Repository Workspace', 'Snapshot', and 'Work items'. To the right is a 'General Information' section with fields for 'Requested by', 'Build Definition', 'Build Engine', 'Build History', and 'Tags'. At the bottom, a 'Summary' tab shows a table of build history.

	Build	Label	Progress	Estimated Completion	Start Time	Duration	Tags
✓	workshop.squawk.core.c...	B20080310-2231...	Completed		March 10, 2008 10:30...	34 seconds	
✓	workshop.squawk.core.c...	B20080305-2325...	Completed		March 5, 2008 11:25...	19 seconds	
✓	workshop.squawk.core.c...	B20080305-2324...	Completed		March 5, 2008 11:24...	33 seconds	
✓	workshop.squawk.core.c...	B20080303-1057...	Completed		March 3, 2008 1:57:0...	24 seconds	
✓	workshop.squawk.core.c...	B20080303-1048...	Completed		March 3, 2008 1:48:1...	28 seconds	
✗	workshop.squawk.core.c...	B20080303-1042...	Completed		March 3, 2008 1:41:4...	27 seconds	
✗	workshop.squawk.core.c...	20080303-1017...	Completed		March 3, 2008 1:17:5...	26 seconds	
✓	workshop.squawk.core.c...	B20080201-1707...	Completed		February 1, 2008 8:06...	30 seconds	v20...
✓	workshop.squawk.core.c...	B20080128-1359...	Completed		January 28, 2008 4:59...	39 seconds	wor...
✓	workshop.squawk.core.c...	B20080123-1623...	Completed		January 23, 2008 7:23...	10 seconds	
✓	workshop.squawk.core.c...	B20080123-1621...	Completed		January 23, 2008 7:21...	18 seconds	

Singular Structure to store all project related artifacts

The screenshot displays the 'Project Area' configuration page for 'Squawk'. On the left, a tree view shows the repository structure under 'Squawk [localhost]', including 'Builds', 'Build Engines', 'Plans', 'Reports', 'Streams', 'Work Items', 'Favorites', 'Feeds', 'My Repository Workspaces', 'My Team Areas', and 'Work Item History'. The main content area is divided into several sections:

- Details:** Includes a 'Summary' field with the text 'Instructional project for user training' and a 'Description' field with the text 'This workshop was created and maintained by the Jazz Jumpstart Team:'. Below the description, a list of names is shown: Jim D'Anjou, Philippe Krief, Paul Vanderlei, and Steve Wasleski.
- Members:** A section titled 'Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for project and team operations. The roles assignments below are also valid in all the project's team areas. Unless configured otherwise, all users in the repository play the 'default' role.' It features a list of roles: 'build' (with a silhouette icon) and 'Jemy Jazz' (with a photo icon). Buttons for 'Add...', 'Create...', 'Remove', and 'Process Roles...' are present.
- Administrators:** A section titled 'Project area administrators have administrative privileges outside of the permissions defined by the process. This applies also to all team areas. When you don't have the permissions you need, please contact one of the administrators below.' It shows a list of administrators with a photo icon and buttons for 'Add...', 'Create...', and 'Remove'.
- Process Description:** A section titled 'Eclipse Way Process (In Progress) The process originally developed by the Eclipse Development Team. The 'Eclipse Way' is an agile, iteration-based process with a focus on consistent, on-time delivery of quality software.'
- Process Iterations:** A section showing a tree view of iterations: 'Main Development Line' (containing 'Release 1 [1/15/08 - 12/31/08]', '1.0 M1 [1/15/08 - 1/25/08]', and '1.0 M2 [1/25/08 - 12/31/08]'), 'Development', and 'End Game'. To the right of this tree are buttons for 'Edit Properties...', 'Duplicate...', 'Create Line...', 'Create Iteration...', and 'Delete...'.
- Attachments:** A section at the bottom with a 'Save' button and a 'Show in Team Organization' link.

Define iterations and state the current state of the project

Team Areas

You can create teams and sub teams (i.e. teams of contractors)

Core Library [Show in Team Organization](#) [Open parent](#) [Open project area](#) [Save](#)

Details

Summary

Description

Members

Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for team operations. The roles assignments below are also valid in all child team areas. Unless configured otherwise, all users in the repository play the 'default' role.

Name	Process Roles
<input type="checkbox"/> April Blues	contributor
<input checked="" type="checkbox"/> Jerry Jazz	contributor, teamlead
<input type="checkbox"/> Zara Intern	contributor

[Add...](#)
[Create...](#)
[Remove](#)
[Process Roles...](#)

Development Line

The development line determines the permissions and the behavior in the project area's process specification that apply to this team area.

Main Development Line [inherited] [Browse...](#)

Process: customized

Description: inherited

Artifacts

- Builds
- Feeds
- Plans
- Work Items
- Reports
- Streams

Administrators

Attachments

Manage team memberships

Customize process for the team or inherit from the project

Control operation behavior based on Roles

Define roles that span across all aspects of Rational Team Concert

Modify permissions

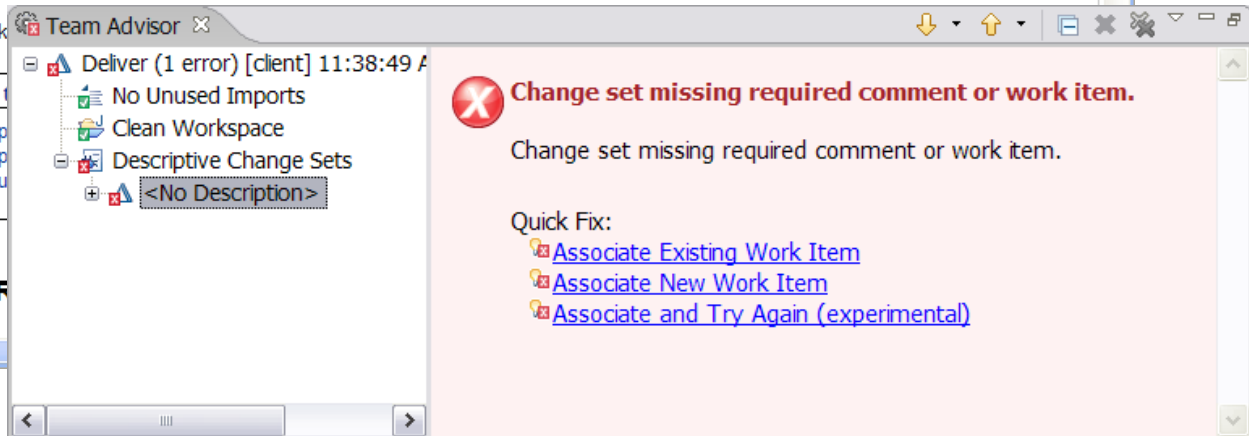
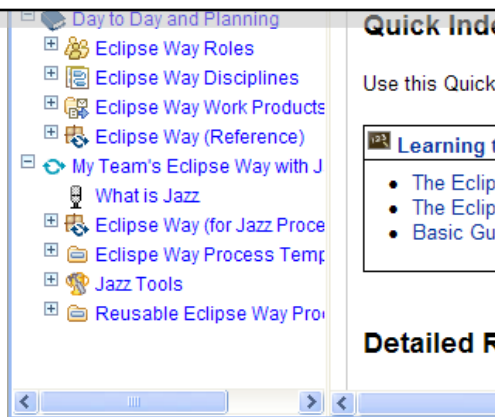
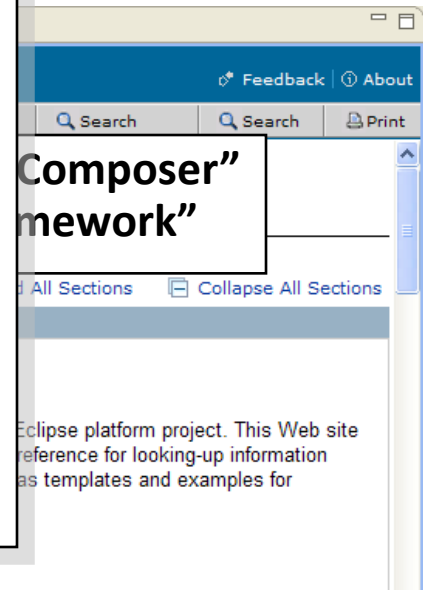
Provide flexible process at each line of development

The screenshot displays the Rational Team Concert Process Editor interface. On the left, a tree view shows the project structure, including 'Roles', 'Project Configuration', 'Project Area Initialization', 'Permissions', 'Behavior', 'Configuration Data', 'Team Configuration', and 'Team Permissions'. The main area is divided into 'Configuration Options' and 'Permissions'. The 'Permissions' section is active, showing a table of actions and their permissions for different roles.

Actions	contributor	teamlead	default
Process			
Generate Team Invitation (server)	✗	✗	✓
Save Team Area (server)	✗	✓	✗
<Uncategorized>			
Abandon Build (server)	✓	✓	✗
Cancel Pending Build Request (server)	✓	✗	✗
Control The Build Lifecycle (server)	✓	✓	✗
Create Private Component (server)	✗	✗	✓
Create Repository Workspace (server)	✗	✗	✓
Delete Build Definition (server)	✗	✓	✗
Delete Build Engine (server)	✗	✓	✗
Delete Build Result (server)	✗	✓	✗
Delete Iteration Plan (server)	✗	✓	✗
Delete Query (server)	✓	✗	✗
Deliver (server)	✓	✗	✗
Deploy Report (server)	✓	✗	✗
Deploy Report Template (server)	✓	✗	✗
Display Report (server)	✗	✗	✗
Manage Report Folder (server)	✓	✗	✗
Request Build (server)	✓	✓	✗
Save Build Definition (server)	✗	✓	✗
Save Build Engine (server)	✗	✓	✗
Save Build Result (server)	✓	✗	✗
Save Component (server)	✗	✓	✗
Save Iteration Plan (server)	✓	✗	✗
Save Query (server)	✓	✗	✗

■ Team Advisor

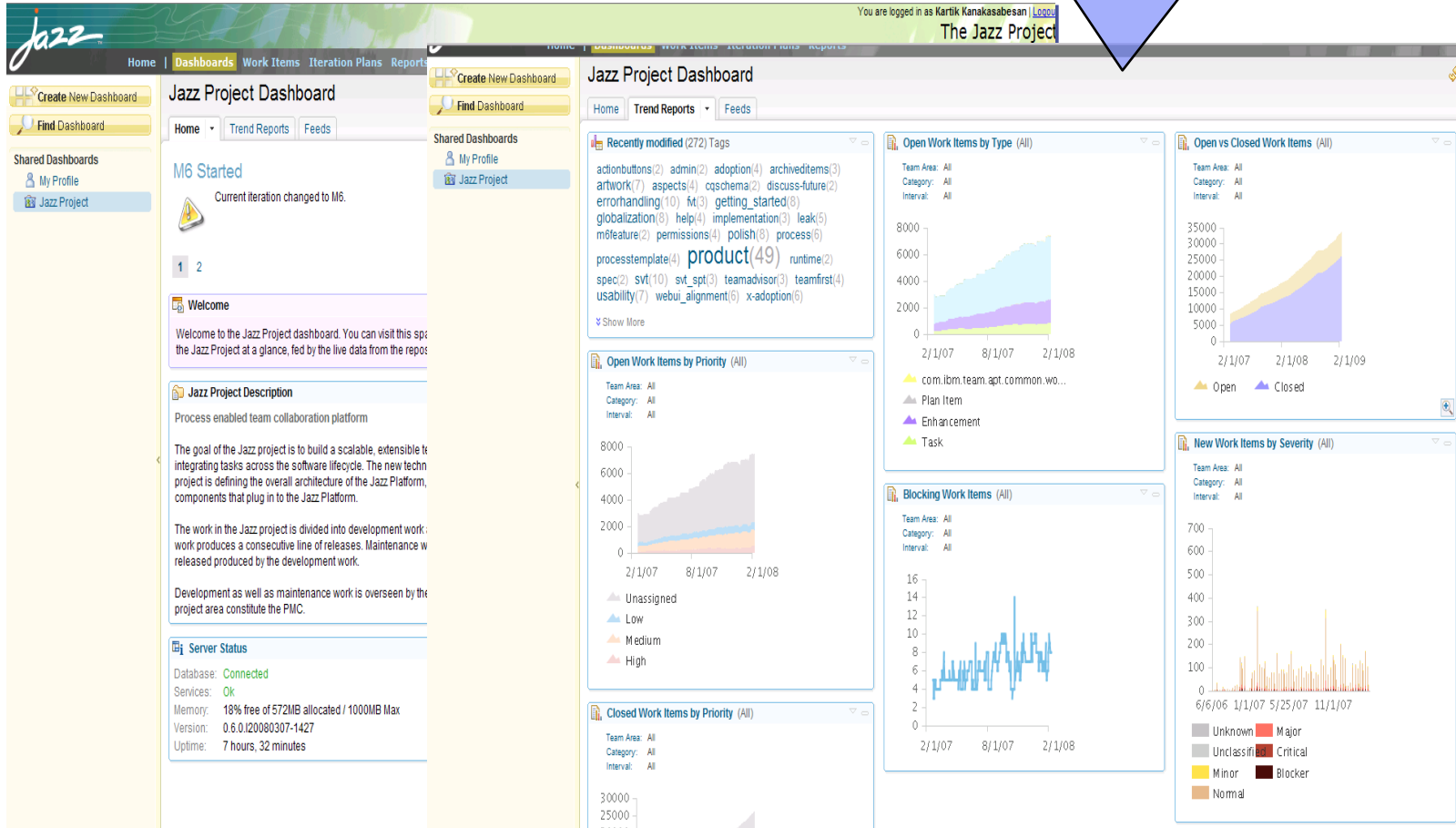
- Rules can be run when delivering changes to enforce team or organizational standards
- Helps ensure higher quality results through enforcement of agreed-upon standards
- Rules are configurable
- “Quick Fixes” can be specified to simplify corrective action
- Process rules can be defined, refined “on the fly”, enabling continual improvements



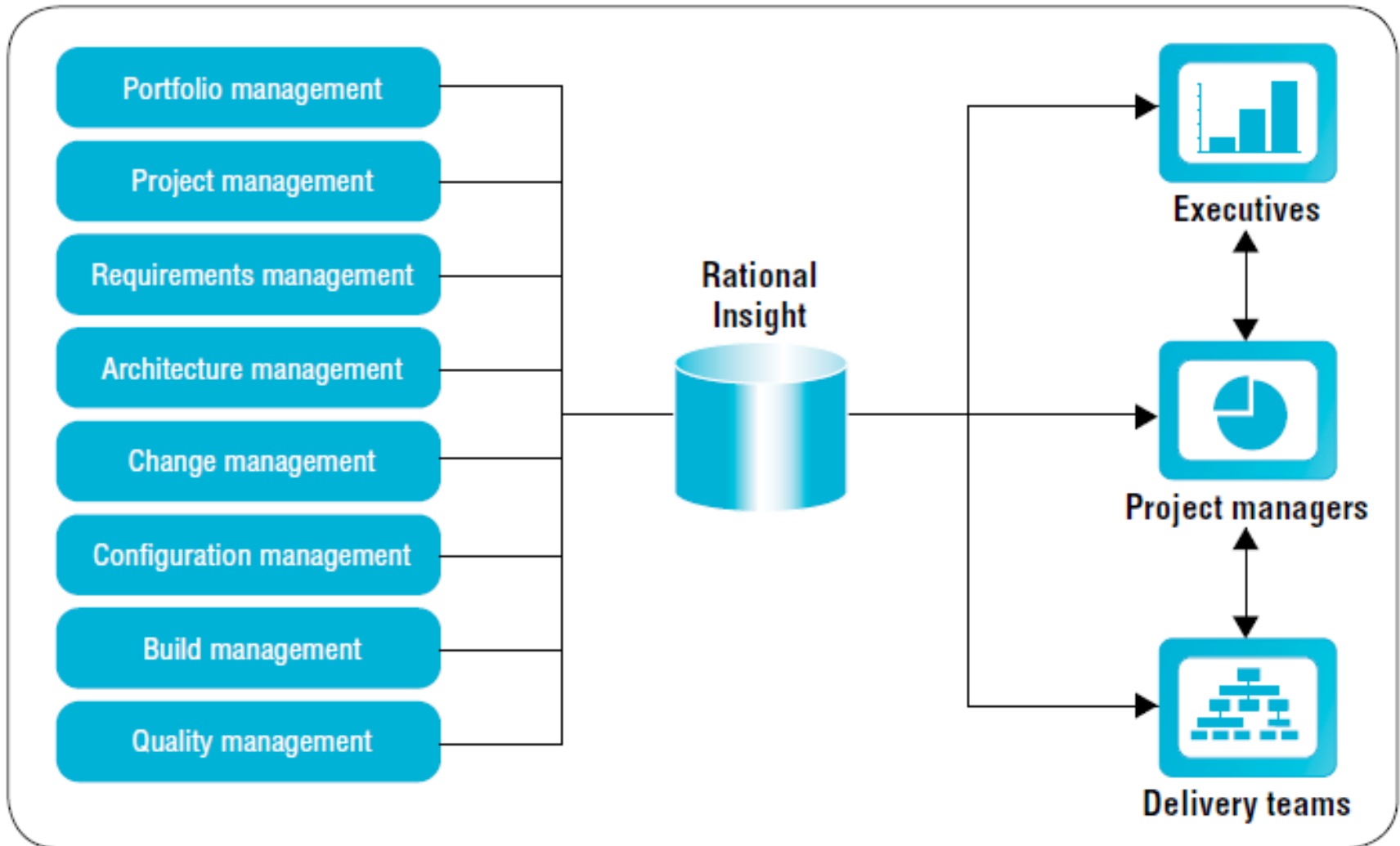
Project Dashboard and Reports

Get project Status at a Glance

Report on current status of the project in real time



- Project burn-down chart
- Completed lines of code
- Open defects (by age, by priority,...)
- Build health
- Requirements with no test case
- Test Execution Status
- Open/closed work items (by ...)

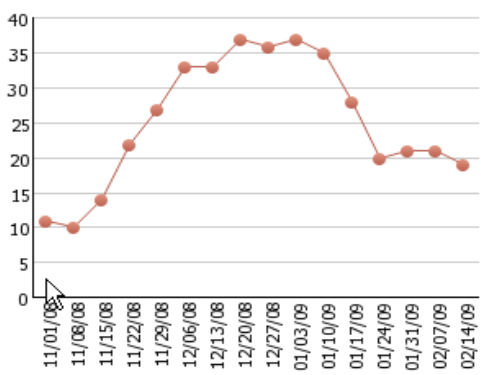


Rational Insight

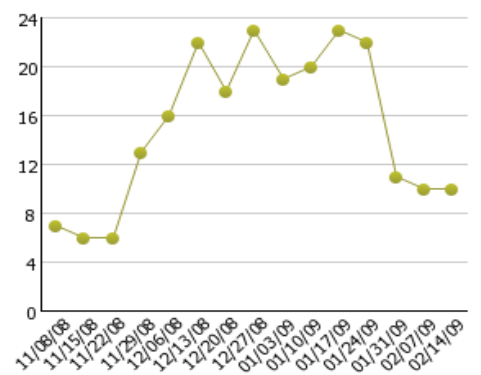
Filters
Program:
Project:
Information
This dashboard is at project level.

Smarter Planet Online Auction

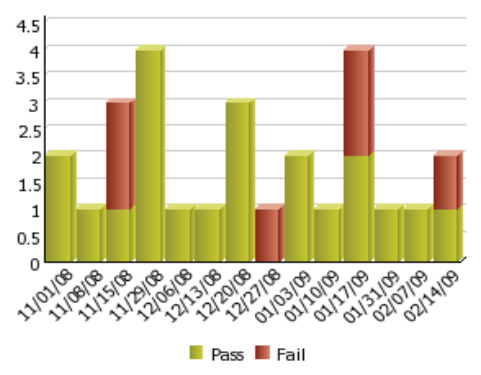
Outstanding Work



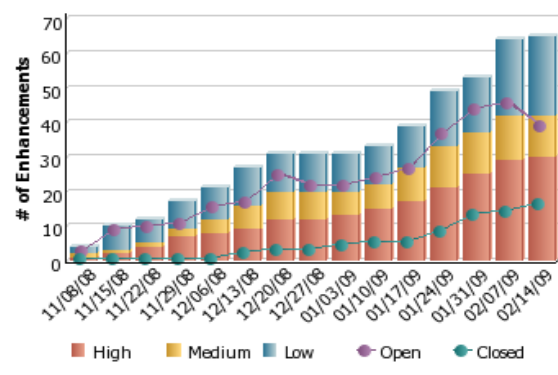
Work Completed



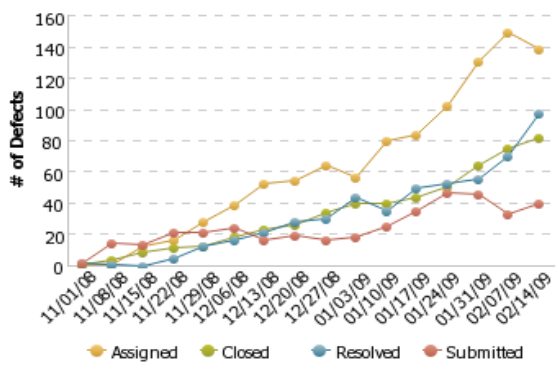
Build Health



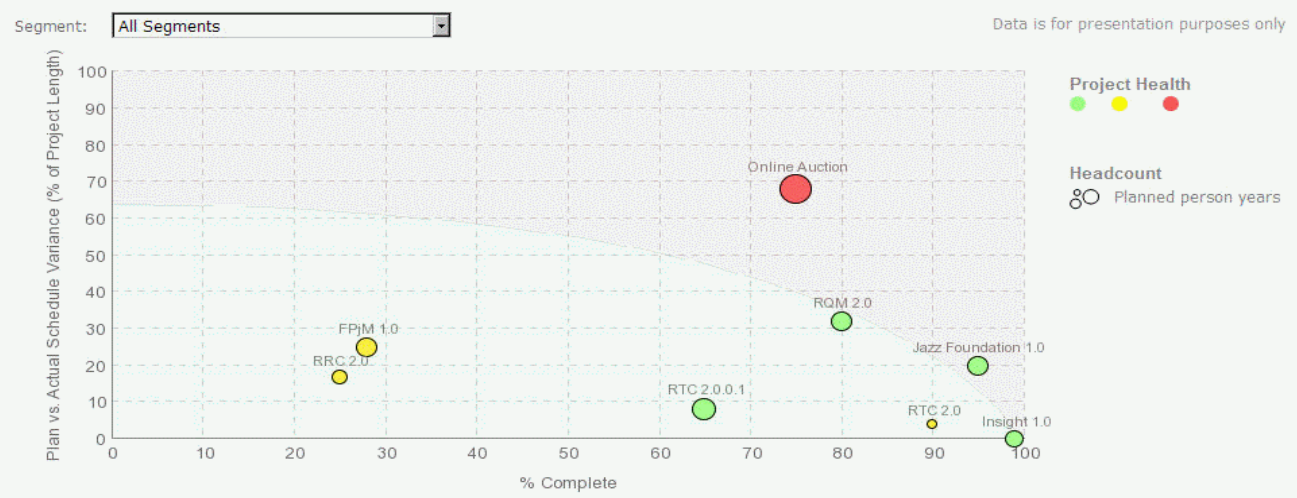
Enhancement Request Backlog



Actual Defect Trends



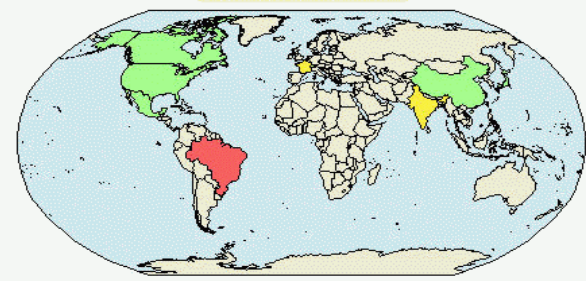
- Executive Dashboard
- Headcount
- Improve Project Health
- Achieve Profit Margin Objective
- Improve Market Share



Hide Table

Segment	Product	Revenue Year-To-Date (M)	Revenue % of Plan	Pipeline (M)	Release	eGA Plan	eGA Outlook	eGA Plan-Outlook	Headcount (PV)	Project Health	Overall Profit Margin	Overall Market Share
Application Lifecycle Management (ALM)	Jazz Foundation	75.7 ▼	96% ▼	87.9 ▼	Jazz Foundation 1.0	May 28, 2009	Jun 12, 2009	15 days	52 ◆	71% ◆	◆	▲
	RTC	115.2 ▲	99% ◆	109.6 ◆	Online Auction	Mar 31, 2009	Jul 7, 2009	98 days	118 ▼	50% ▼	▼	▼
					RTC 2.0	Jun 15, 2009	Jun 19, 2009	4 days	12 ◆	90% ◆	◆	▲
Governance Solution	Focal Point for Project Management	29.1 ◆	93% ◆	84.2 ▲	FPJM 1.0	Oct 15, 2009	Nov 17, 2009	33 days	60 ◆	74% ▲	◆	◆
	Rational Insight	108.7 ▼	98% ▼	162.7 ▼	Insight 1.0	May 26, 2009	May 26, 2009	0 days	40 ◆	83% ▼	▼	◆
Quality Management (QM) Solution	RQM	35.2 ▲	112% ▲	79.6 ▲	RQM 2.0	Jun 15, 2009	Jul 31, 2009	46 days	56 ◆	89% ▲	◆	▲
	RRC	28.1 ◆	83% ▲	45.1 ▼	RRC 2.0	Nov 4, 2009	Nov 24, 2009	20 days	32 ▲	85% ◆	◆	▼

Project Health by Country

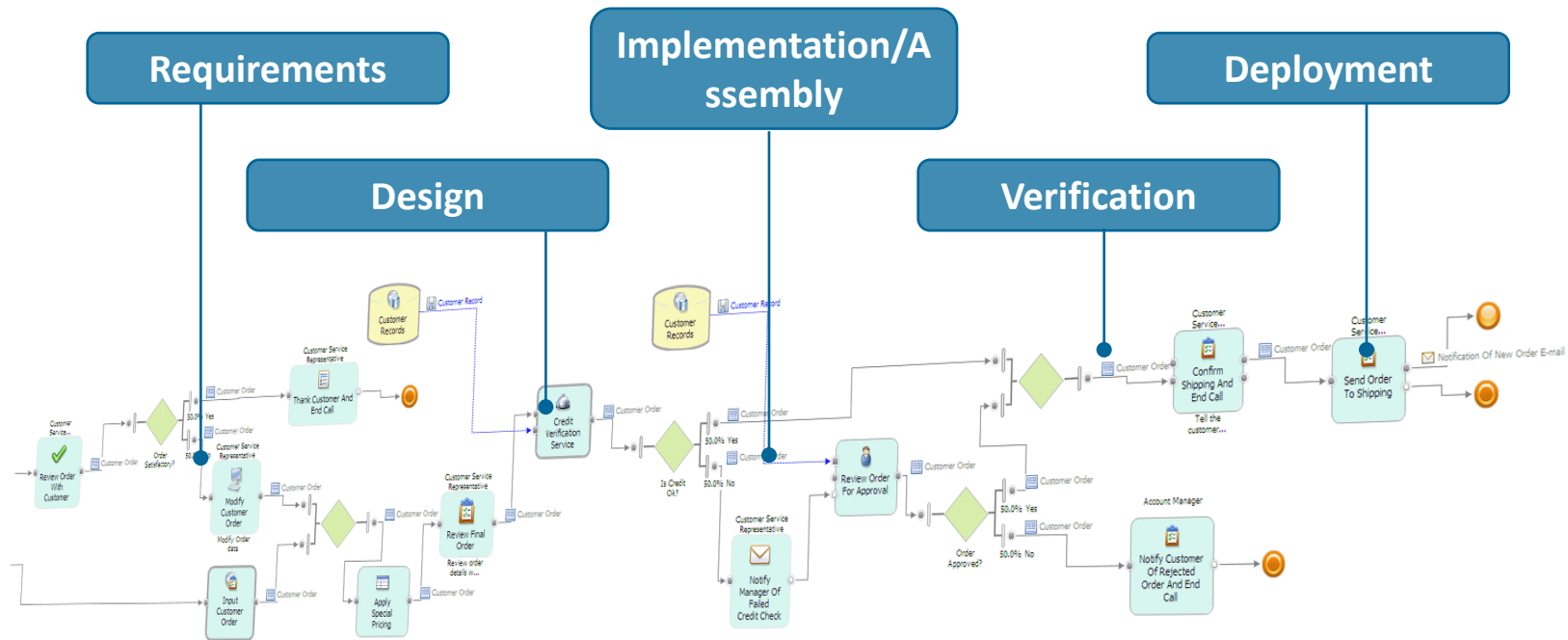


Rational MCIF



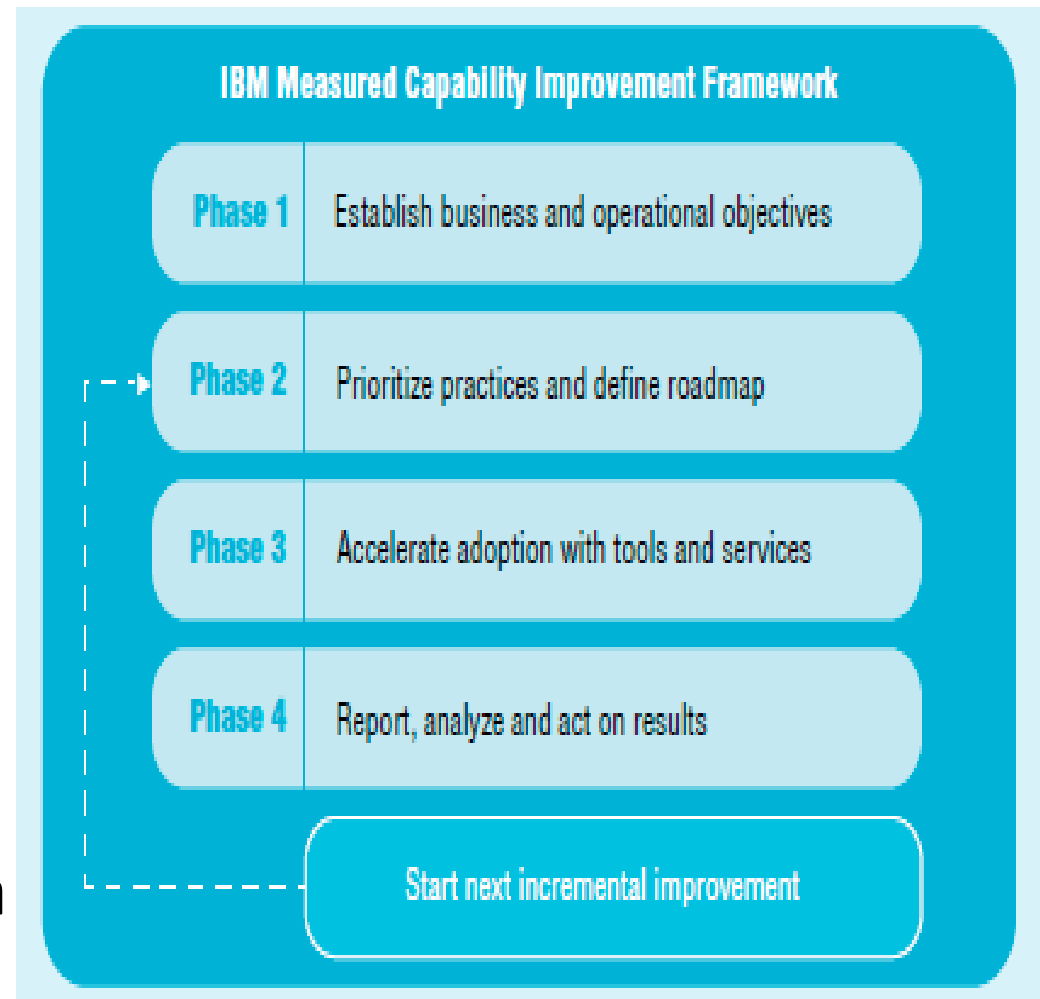
Kde začít? Co má být prioritou?

- Software delivery is a business process that *must* be continuously improved



Source: Gartner, "Making the Difference: The 2008 CIO Agenda," Jan. 2008

- Strukturovaný přístup ke zlepšování softwarových týmů skrze soustavné a měřitelné zlepšování procesu
- Optimalizované pro agilní týmy a metodiky
- Cyklus měření – vyhodnocení – náprava



Solution Area	Symptoms, Observations and Comments	Calculated Average
Improve Build Process	<p>We have a primarily manual and error prone build process</p> <ul style="list-style-type: none"> ▪ We have a primarily manual and error prone build process ▪ Builds are infrequent (less than daily). ▪ Tests (a.k.a. smoke tests) are not automatically triggered once builds are done. 	2,7
Improve lifecycle collaboration	<p>Our team is functionally stove-piped, or has ineffective communication across functions</p> <ul style="list-style-type: none"> ▪ Our team is functionally stove-piped, or has ineffective communication across functions. ▪ Our team spends a lot of time in status meetings and to figure what needs to be done. ▪ Our teams are often having handover problems and miscommunication between different functional groups or teams on different sites. ▪ We have no effective means of tracing requirements, to design, code, to test, at least not without extensive overhead. 	3,0
Improve Developer Efficiency	<p>We lack an integrated development environment supporting design, patterns, code generation, local compiles, high-level languages, and automated unit testing.</p> <ul style="list-style-type: none"> ▪ We lack an integrated development environment ▪ We leverage lower level languages (vs. Java, VB, 4GLs, ..) 	3,0
Improve Test Effectiveness, Improve Security Testing, Improve Performance Testing	<p>Function and system-level testing is primarily manual, error prone, or does not cover all necessary types of testing.</p> <ul style="list-style-type: none"> ▪ Function and system-level testing is primarily manual, error prone, or does not cover all necessary types of testing. ▪ Most tests are manual. This make continuous regression testing difficult / expensive. ▪ We do not do security testing throughout the lifecycle. 	3,2

- Agile se prosazuje a funguje, ať už to znamená cokoliv :-)
- Je potřeba brát v úvahu celý delivery process
- Větší projekty nebo plošná adopce vyžaduje automatizaci a nové nástroje
- Vyzkoušejte Rational Team Concert, pro tým do 10 lidí je ZDARMA (www.jazz.net)

