Measuring Capability Improvement - A Rational Perspective

Dr Laurence James,
IBM-Rational
Email: ljames@uk.ibm.com
The Defining Challenges In Effective Software Delivery

“Only 37% are satisfied with the speed of software development”

“Only 42% of users are satisfied with quality”

“50% of outsourced projects are expected to under perform”

Isolated people, process, and projects

Geographic barriers
- Poor communication
- Language, culture, time
- Process gaps resulting in rework
- High degree of friction

Organizational barriers
- No meaningful collaboration
- Weak project governance
- Lack of domain expertise
- Poor LOB oversight
- Secure IP when outsourcing

Infrastructure barriers
- Incompatible tools and repositories
- Unreliable access artifacts
- Lengthy on-boarding
- Inflexible tooling integration
But First… Going To The Doctor…

- What you want to know
  - How long will I live?

- What the doctor does
  - A health assessment

- What the doctor will measure
  - Heart rate, cholesterol, blood pressure, body fat, …

- There is a correlation between the values of what a doctor measure and your life expectancy
  - But you can outlive somebody with lower cholesterol than you have…

Experiences shows that there is a similar correlation between software best practices and attainment of desired business results
Going To A Personal Trainer

- **What you want to do**
  - Run a marathon on a personal best time

- **What the trainer will do**
  - A fitness assessment to understand your strengths and weaknesses
  - Personalized plan is produced based on where you are and where you want to be
    - Run sprints / intervals, run medium distances, strength training, eat healthy, …
  - Plan will be adjusted based on progress in each area
    - Ongoing monitoring
    - Your plan will be personalized and continuously adapted, but based on known patterns of success…

*Business value is maximized through “personalized” software process improvement plans that are continually adapted based on measurable results*
What Is “Measured Capability Improvement Framework (MCIF)”?

- A systematic approach for improving your business
  - Couple business problems to incrementally adoptable practices
  - Accelerate incremental rollout of practices (w. associated tools) through reusable assets
  - Measure effectiveness of practice adoption and business value delivered

- Captures +10 years of Rational experiences in incremental adoption
  - Key aspects have been used in agile transformation’s of +80 IBM internal projects
  - Process independent – used in conjunction with RUP, XP, Scrum and other processes
Measured Capability Improvement Framework

- Identify desired business objectives
  - Reduce Time-to-Market, Improve Quality, Increase Innovation, …
- Identify and select target practices and tooling to drive desired business objectives
  - Leverage assessments and out-of-the-box business objectives to practices mappings
- Effectively deploy well-governed practices
  - Process guidance, training courses, enablement material, etc.
  - Understand what aspects of which tools to adopt to effectively adopt practices
- Measure results and take corrective actions
  - Understand whether target practices are successfully adopted
  - Understand whether desired business outcomes are achieved or not
Practices As A Foundation For MCIF

- Practices represent a useful unit of knowledge (you can e.g. teach a course on it)
  
- Practices can be independently / incrementally adopted
  
- Practices provide one-stop shop to courses, tool features, services, articles, process content, enactment, ...
  
- Practices map to practitioner pain points
  
- Practices map to organizational business objectives
  
- Practice adoption can be measured
Measured Capability Improvement
Map business value to software delivery practices

**Example: Financial Service Company**

**Customer Business Challenges**
- Create financial products more quickly
- Functionality of customer web falling behind competition
- Inconsistencies with integrated financial reporting
- Recent SOX audit failure

**Operational Objectives**
- Reduce time-to-market
- Improve productivity
- Increase innovation
- Improve consistency/predictability
- Improve oversight
- Enable flexible/global resourcing
- Satisfy compliance mandate

**Software Delivery Best Practices**
- Use-case driven development
- Continuous integration
- Shared vision
- Whole team
- Risk-value lifecycle
- 2-level project planning
- Test-driven development
- Asset-based development
- Asset governance
- Iterative development
- SOA modeling

**Business Metrics**

<table>
<thead>
<tr>
<th>Project</th>
<th>Time to Market (M)</th>
<th>Quality (Defect Density)</th>
<th>Innovation (Cust. Sat.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>22</td>
<td>2.3</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>14</td>
<td>1.4</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>18</td>
<td>1.6</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>0.3</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>0.4</td>
<td>8</td>
</tr>
</tbody>
</table>

**Ongoing Adoption Assessment**

- Iterative
- High Quality
- Stakeholder Feedback
- Reflections
- Test Early
- Time-Boxed Iteration
- Consumable
- Daily Scrum

**Adopt Practice**

- Target: Phase 1
- Already implemented
- Outside scope
Best Practices Provide Customizable Usage Models For The Rational Software Delivery Platform

Adopt Best Practice

- Use-case driven development
- Continuous integration
- Shared vision
- Whole team
- Staged integration
- Multi-team management
- Risk-value lifecycle
- Asset-based development
- Asset governance
- Iterative development
- SOA modeling

Rational

ClearQuest

Rational

Forge

Rational

RequisitePro

Rational

ClearCase

Rational

Quality

Manager

Rational

Method
Composer

Rational

Asset
Manager

Other offerings

Rational

Requirements
Composer

Rational

Team Concert

Rational

Team Concert

Example

- Risk-value lifecycle
- 2-level project planning
- Iterative development
- Continuous integration
- Use-case driven development

- Work Items
- Build

- SCM

- Whole team

- Team collaboration
- Customer involvement
- Self organization
- Transparency

Best practice asset repository
Business-Focused Change And Value Realization

- **Business Value**
  - Business drivers
  - Roadmap
  - Practices
  - Tools
  - Services

- **Traceability**
  - Establish ROI
  - Health Assessment
  - Value delivered

- **Products and Services**
  - ROI
  - KPIs

- **Understanding Needs**

- **Realization**

- **Delivery**
  - Business Metrics
  - Self Check
  - Practices

- **Are we on track?**

- **Establishment of Value**
  - Business drivers
  - Metrics
Rational’s MCIF Assets

- Establish roadmap
  - Value Traceability Trees: Linking Business Objectives – Practices – Products & Services
  - Health Assessment for Software Delivery: Service offering - Understand where you are at today, and establish a roadmap forward

- Execute roadmap
  - IBM Rational Self Check for Software Teams: Service offering - Facilitated self assessment
  - Rational Method Composer 7.5: Guidance on practices and effective usage of products
  - Objective measurements for practices and business objectives (RTC, Vega, …)
Value Traceability Tree: Development Productivity

**Deliver Faster**
- **Measurements:**
  - SLOC
  - Function points
  - Use case points
  - Story points
  - ...
- **Practices:**
  - Whole team
  - Test management
  - Quality management
  - Continuous integration
- **Key Products:**
  - Rational Team Concert
  - Rational Quality Manager
  - Rational BuildForge
  - ClearCase / ClearQuest
  - ...
- **Key Services:**
  - ...

**Deliver Smarter**
- **Measurements:**
  - Survey of value delivered
  - Survey of feature usage
  - Runtime analysis of code usage
  - ...
- **Practices:**
  - Iterative development
  - Requirements management
  - Use-Case Driven
  - Whole team
- **Key Products:**
  - Rational RequisitePro
  - Rational Team Concert
  - ...
- **Key Services:**
  - ...

**Optimize Resources**
- **Measurements:**
  - Cost / person month
  - SLOC
  - Function points
  - Use case points
  - Story points
  - ...
- **Practices:**
  - Outsourcing
  - Improve resource skill level
  - Tool automation
- **Key Products:**
  - ClearCase
  - ClearQuest
  - WBTs and RMC
  - ...
- **Key Services:**
  - ...

**Align Operations & Development**
- **Measurements:**
  - Effort to deploy application
  - Effort to fix critical defects
  - ...
- **Practices:**
  - Shared test environments
  - Shared operational mod.
  - Shared diagnostics info
  - ...
- **Key Products:**
  - Rational Perf. Tester
  - Rational TestLab Manager
  - ...
- **Key Services:**
  - ...
Service Offering: Health Assessment for Software Delivery

- Identify **drivers** for delivery improvements
- **Current state**: Organization & Collaboration, Oversight, Software Delivery Practices, Skills, Development Environment, and Metrics
- **Future state**: An initial roadmap outlining how you can evolve your delivery approach to address your key business drivers
- List of **inhibitors** that may prevent successful implementation of roadmap
- Identify **metrics** to track progress on the implementation of your roadmap
Shows Health Assessment being answered in a current / future mode
Service Offering: IBM Rational Self Check for Software Teams

- Systemic approach for teams to assess their adoption of desired practices
- Enables teams to learn, improve their effectiveness, and share their learnings

I’m a developer.
- I want to learn and remember practices.
- I want new ideas to stick.
- I don’t want to be beaten by a stick.

I’m a coach.
- I want efficient continuous improvement.
- I want to hear from quiet people on the team.
- I want to learn from teams like mine.

I’m an executive.
- I want to understand if my teams are improving.
- I want to involve team members in the change effort.
### Components: IBM Rational Self Check for Software Teams

#### The Assessment Tool

<table>
<thead>
<tr>
<th>Practice</th>
<th>Comments</th>
<th>Average</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Cases</td>
<td>Use Stories instead</td>
<td>7.8</td>
<td>0.50</td>
</tr>
<tr>
<td>Unit Test</td>
<td>Not enough, not in build</td>
<td>6.8</td>
<td>1.89</td>
</tr>
<tr>
<td>Stakeholder Feedback</td>
<td>Want more</td>
<td>8.5</td>
<td>0.58</td>
</tr>
<tr>
<td>Time-Boxed Iterations</td>
<td>Not well specified</td>
<td>8.3</td>
<td>0.50</td>
</tr>
<tr>
<td>Daily Scrum</td>
<td>Not Fridays</td>
<td>9.3</td>
<td>0.50</td>
</tr>
<tr>
<td>Test Early</td>
<td>Not enough TDD</td>
<td>4.5</td>
<td>1.91</td>
</tr>
<tr>
<td>Reflections</td>
<td>Need clear actions</td>
<td>6.4</td>
<td>0.50</td>
</tr>
<tr>
<td>Static Analysis</td>
<td>Not automated yet.</td>
<td>3.3</td>
<td>0.96</td>
</tr>
</tbody>
</table>

#### Big Picture of Practice Adoption

- **Evol. Architecture**
- **Vision**
- **Use Cases**
- **Automated Unit Tests**
- **Scrum meeting**
- **Iterative**
- **Non Solo**
- **Reflections**

#### Experience Report

![Experience Report Diagram]

#### Deep Dive on Practice

![Deep Dive on Practice Diagram]
Example: We Are Iterative Aren’t We?

The Big Picture

Deep Dive Insight

235 people. Multi-shore. Java
8 week “iterations”. 24 Month project. New to agile.
Understand how Rational tools supports the practice, and how to measure your adoption. Have you done a great job yet implementing the practice?

Each practice is clearly called out, with information:
- Why should you use this practice?
- How do you adopt this practice?
- How do you learn about this practice? (Courses, books, etc)

Find enablement material and sign up for courses, etc.
Guideline: Iteration Code Health

Overview

Iteration Code Health shows the information about health of a build in an iteration. It measures the intervals in unit of time that you have a failed build and the intervals in unit of time that you have a clean build. Iterative development recommended incrementally building working software. As a result, the build should not stay failed for long especially at the end of the iteration. Another important information to measure in this metric is number of total tests that are used for testing the build.

Measurement Method

Number of Tests = Total number of test in a test suit at a given time
Clean Build Range = Internal time that the build stay clean
Failed Build Range = Internal time that the build failed

Measurement Analysis

A good way to monitor iteration code health over time is to plot the chart. The following picture is example of build health chart.
MCIF Summary

- Allows clients to focus on addressing business objectives and challenges
- Helps teams understand what practices drive what business benefits
- Process improvement teams can deploy solutions more rapidly through improved deployment guidance and sharing early successes through standardized experience reports
- Allows teams to continuously assess adoption progress, and take corrective action
- Facilitates measurement of realised business benefits
- MCIF is modular - only use elements that will benefit the team
QUESTIONS
THANK YOU

Learn more at:

- IBM Rational software
- IBM Rational Software Delivery Platform
- Process and portfolio management
- Change and release management
- Quality management
- Architecture management
- Rational trial downloads
- Leading Innovation Web site
- developerWorks Rational
- IBM Rational TV
- IBM Rational Business Partners