Development productivity in an agile world – past, present, and future

Maria Ericsson
Executive IT Specialist
IBM Rational Tiger Team

GOTO Prague 2011
Historical Perspective on Productivity

- SLOC was easy to measure … still is:
  - There is a large pool of SLOC based productivity statistics
  - These statistics are still the most complete set of data existing

- Function Points & Use Case Points were developed to solve shortcomings, however:
  - Different programmers code differently
  - Not all code adds value
  - Functions or Use Cases can have different “sizes”

- Experience indicates that utility of these measures is limited:
  - Intra-team, it can be used (*carefully!*) to calibrate productivity within the team
  - Across organizations & platforms, comparisons break down
Modern Perspective on Productivity

- Many development organizations have sophisticated dashboards
  - Richer measurement set
  - Much less human intervention

- However, productivity is not only about asset volume —
  Productivity is about *business value produced*
  - Challenge: determining a consistent meaning for business value that is useful to the entire organization
  - Challenge: making the measures meaningful, transparent, real-time

- Conclusion: a productivity strategy should include
  - Transparency at all levels in an organization
  - Consistent and useful leverage of a carefully crafted chain of related measures
  - Using modern development technologies that include measurement facilities to foster continuing relevance, accuracy, and real-time results
Measures to address business value

Transparency and critical success factors

CEO/CFO

CIO/CTO

Project Manager

Developer

Value chain

Core dev loop
Sprint cycles
Burn rate, etc

Project/release
Quality metrics
Code stability
Test coverage, etc

Operational effectiveness
Time to market
Predictability
Cost

Portfolio management
Portfolio complexity analysis
Total Cost of Ownership
Investment analytics
Investment prioritization

Business management
Earnings per share
Profit

Portfolio management

Operational effectiveness

Core dev loop

Business

Team & Organization

Team & Organization

Individual

Individual

IBM Software Group
Software Development That Supports IT Agility

Software Development Must Deliver:
- ✓ More business solutions
- ✓ More flexibility
- ✓ More quality
- ✓ More business value

-yet-
- ✓ Less risk
- ✓ Less costs
- ✓ Less or flat resources

Effective software development promotes a delicate balance of resources, reducing costs while delivering more, higher-quality business solutions.
In the end, it is about return on investments made

Investment Analytics: Estimating streams of value accrued over time

- Benefits streams
- Development costs streams
- Operations and maintenance streams

Each symbol represents a variable indicating an uncertain monetary amount. Each symbol's amount is appropriately discounted; all are summed (∑) to obtain NPV.

Investment Value expressed as NPV

Capability provided by Investment Analyzer, a component of Rational FocalPoint.
Organization Dynamics

- Concerns flow down the organization while measures (and data) flow up
- We need tools to plan, track, and deliver on our commitments at every level
- Productivity needs to be measured at every level, and it is measured differently

Line of Business Executive
- commits to Profit, Internal Rate of Return

Senior Manager
- commits to Delivery of business value through the optimal use of resources

Project Manager or Team Lead
- commits to Project deliverables, cost and schedule

Staff member
- commits to Work item, artifact completion

Concerns

Commitments
To be practical, a metrics framework is needed

**Measured Capability Improvement Framework**

- **Business Goals**
- ** Operational Goals**
- **Practices / Process Definition**
- **Process Enactment / Governance Enforcement / Process Awareness**

**Performance Measurement**

- **Value Metrics**
  - *e.g.*, ROI, ROA for SSD

- **Operational Effectiveness Metrics**
  - *e.g.*, Time to market, Quality, FP

- **Practice Adoption/Maturity**
  - Subjective
- **Practice Artifacts**
  - Objective

**IBM Software Group**

**Jazz Platform**
Accruing Technical Debt
Activities that a development team or a team member choose not to do well now, which will impede future development if left undone

- A financial liability
- Technical debt incurs interest payments:
  - Extra effort spent in future development tasks due to less effective design – translates to labor hours.
- We can choose to continue to pay that interest, or pay down the principal by investing labor in refactoring the design to something more effective.
- Some are deliberate decisions, e.g. to meet deadlines
  - If known, you can attempt an assessment of the interest cost vs. refactoring cost
- Others are inadvertent and not known

Martin Fowler: TechnicalDebt, TechnicalDebtQuadrant
Chris Sterling: Managing Software Debt: Building for Inevitable Change
Questions we hear ...

Evidence of the emerging need for a productivity strategy

- How can we support team collaboration better?
  - Taskboards (developer cockpit) track progress

- How can we make status of projects more transparent?
  - Dashboards can provide insight

- How can we achieve improved operational efficiency?
  - Executive level dashboards to support prioritization
  - Portfolio management
  - Portfolio analytics (emerging)

*The observation is that a foundational challenge is to provide accurate and relevant information to the right person in real time – in other words: transparency*
The Jazz Architecture: An **open architecture for lifecycle tool integration**

*Focusing on integrating the information not the development tools*

- **Built for the 21st century:** designed using Web architectural principles, implemented with Web technologies
- **Realistic:** recognizes that customers will not replace their current investments wholesale
- **Pragmatic:** allows tools and services to be upgraded independently, without sacrificing rich integration
- **Open:** supports the requirement to have a variety of tools from different sources – 3rd party as well as open source/cheapware
Envisioning a platform that can transform software delivery

*Transparency: Dashboards, taskboards, reporting, seamless integration*

- A scalable, extensible team collaboration platform
- A community at Jazz.net where you can see Jazz-based products being built
- Our vision of the future of systems and software delivery, supporting globally distributed teams
- An integration architecture enabling mashups and non-Jazz based products to participate
- An evolution of our portfolio which will evolve to leverage Jazz technology over time

**Jazz** is a project and platform for **transforming how people work together** to deliver greater value and performance from their software investments.
Examples

Danske Bank
IBM Development
Danske Bank: Measuring a Large-scale Agile Process Improvement Effort

- 2000+ developers
- 6 business units
- Development teams are often geographically distributed
Danske Bank Results

"We intend to enhance the efficiency of our IT development process by 10% and reduce the time to market from approximately 14 months to an average of nine months. The first business deliveries will even be provided in the course of just four months."

PETER RASMUSSEN, SENIOR VICE PRESIDENT IT DEVELOPMENT PROCESSES & TOOLS, DANSE BANK.

"These years, the market is changing at lightning speed, and it is crucial for the organization that the expectations of the business units and, ultimately, the expectations of the market can be matched. Our vision is to deliver new and exciting services more efficiently and faster than ever. We intend to enhance the efficiency of our IT development process by 10% and reduce the time to market from approximately 14 months to an average of nine months. The first business deliveries will even be provided in the course of just four months."

Rational Team Concert are expected to grow in 2011 by another 1,000, and the aim is to give all of the 2,000 IT developers in locations in Denmark and India access to Rational Team Concert.

"Above all, Rational Team Concert is a teaming tool. A lot of people can interact to achieve the best possible result. It is easier for the project manager to manage tasks and create transparency among the team members across the geographies. In addition, data can be collected and reportings can be made.
Measures help answer key questions

**Business-Related Measures**
- Are we meeting business objectives?
  - Project deliver faster than today
  - Systems created or updated in the projects have the agreed quality
  - The development organisation is a learning organisation
  - Employee satisfaction

**IT-Related Measures**
- Are we seeing the benefit where we expected?
  - Appropriate level of management and analysis activities
  - Efficient requirements definition and signoff
  - Fewer breakages when solution elements are integrated
  - Less “solution hardening” needed

**Agile-Related Measures**
- Are we agile?
  - Agile practice adoption
  - Agile work product adoption
  - Agile task adoption
  - Agile process adoption

Are we meeting business objectives?
Are we seeing the benefit where we expected?
Are we agile?
The business case for a more efficient development organisation

**The target:** 10% increase in efficiency over a 3 year period
- Reduced time-to-market (faster)
- Reduced cost (cheaper)
- The value of the solution (the best solution)
- The quality of the solution (the right quality)

**The measurements:**
- Idea Qualification phase (IQ) duration
- Pre-analysis phase duration
- Average Time to Market for first business release
- Average CMMI Level-3 Score
- Average (Effort/Function Point) Productivity
The results – so far

Average IQ Length

Average Pre analysis length

Average time to market

Average CMMI Level-3 Score
Conclusions

Change of focus to results, not documents
• Moved from planning activities to planning deliveries
• Still overly precise plans and requirements

Faster time to market
• IQ phase forces early collaboration on design
• Big uncertainties are addressed earlier

Improved collaboration among stakeholders
• Honest measures to identify the right improvements.
• Moving past cherry-picking
The IBM Story: Five Years Ago - Our Pain Points…

✓ joining a team
✓ get my environment configured to be productive
✓ what is happening in my team
✓ collecting progress status
✓ following the team’s process
✓ ad hoc collaboration/sharing of changes
✓ starting an ad hoc team

✓ is the fix in the build?
✓ run a personal build
✓ tracking a broken build
✓ why is this change in the build?
✓ reconstructing a context for a bug/build failure

✓ interrupting development due to a high priority bug fix
✓ working on multiple releases concurrently
✓ tracking the code review of a fix
✓ referencing team artifacts in discussions
✓ how healthy is a component?
✓ collecting project data/metrics?
✓ keeping plans up to date

Team awareness
Build awareness
Project awareness

Boring and painful
Over 150 Rational development projects (~2800 users) using Jazz technology (Rational Team Concert)

- Plus an additional 700+ projects around IBM – hosting 8500+ users!

- Boarding time for new projects - less than one day

- Applicable to agile/iterative and waterfall projects
# Team Awareness: Taskboards track Work in Progress

**Accelerate stand-up meetings, increased transparency**

<table>
<thead>
<tr>
<th>Task</th>
<th>Priority</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improve documentation for 4.4</td>
<td>55</td>
<td>javacoc updates for @chore in 4.3</td>
<td>In Progress</td>
</tr>
<tr>
<td>Provide improved Assertion syntax</td>
<td>60</td>
<td>Based on the assertThat syntax we should provide assumptions and theories support</td>
<td>In Progress</td>
</tr>
</tbody>
</table>

**Drag and drop work items to change their state.**

- assertArrayEquals misses differences
- testCount hard-coded to 1 for childless Description
- Tests on protected methods fail
- assertThat fails with Class tests (documentation problem)

**See the work currently in progress**

- [Docst Cookbook TestRunner section incorrect](#)
- [shows green bar while assert false](#)
- [Should not call derived's after if super's before failed](#)
- [After method not called after my test timeout in 4.3.1](#)
Build Awareness

- **Team of Team**
  - Scheduled weekly integration build
    - Stabilized until green
  - Continuous integration stream
    - Share changes, rarely green

- **Each Team**
  - Continuous local builds
    - Always green

- **A Developer**
  - Personal builds
Team Awareness
Project Awareness
Project Awareness

Endgame Focused and Disciplined

release

warm-up

M1a

M1

... endgame

decompression

retrospective

initial release plan

plan
develop
stabilize
plan
develop
stabilize
plan
develop
stabilize

6 weeks
6 weeks
6 weeks

fix - spit & polish
test
fix
test
# Tech Debt Measurements

*Quality Focused*

<table>
<thead>
<tr>
<th>Metric</th>
<th>Goal</th>
<th>2006 Measurement</th>
<th>2010 Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance / Innovation</td>
<td>50/50</td>
<td>42% / 58%</td>
<td>31% / 69%</td>
</tr>
<tr>
<td>Time to Market (Major)</td>
<td>12 Months</td>
<td>18 + Months</td>
<td>12.5 Months</td>
</tr>
<tr>
<td>Customer Calls</td>
<td>-5% YoY</td>
<td>~ 135,000</td>
<td>~100,000 (-19% since 2009)</td>
</tr>
<tr>
<td>Customer Defect Arrivals</td>
<td>-5% YoY</td>
<td>~ 6,900</td>
<td>~2200</td>
</tr>
<tr>
<td>On Time Delivery</td>
<td>65%</td>
<td>47%</td>
<td>92%</td>
</tr>
<tr>
<td>Defect Backlog</td>
<td>3 Months</td>
<td>9+ Months</td>
<td>3 months</td>
</tr>
<tr>
<td>Customer Sat Index</td>
<td>88%</td>
<td>83%</td>
<td>88%</td>
</tr>
</tbody>
</table>

*Note: Goals are either internal IBM statistics or industry benchmarks.*
Improving Bottom-Line Growth

SW Revenue per DE HC $M

Revenue per HC $M

Effort/Release as %

2004 2005 2006 2007 2008 2009

Rev per DE HC

E/R

---

IBM Software Group

29
Conclusions

- In the end, development productivity is about return on development assets
- Technology is emerging that help us provide real-time feedback at all levels in an organization
- Transparency is key to leaps in productivity improvement