Measuring Software Development Productivity

www.construx.com

@stevemconstrux
ACM Highlights

- **Learning Center** tools for professional development: [http://learning.acm.org](http://learning.acm.org)
  - 4,500+ trusted technical books and videos by O’Reilly, Morgan Kaufmann, etc.
  - 1,000+ courses, virtual labs, test preps, live mentoring for software professionals covering programming, data management, cybersecurity, networking, project management, more
  - Training toward top vendor certifications (CEH, Cisco, CISSP, CompTIA, ITIL, PMI, etc.)
  - Learning Webinars from thought leaders and top practitioner
  - Podcast interviews with innovators, entrepreneurs, and award winners

- Popular publications:
  - Flagship *Communications of the ACM (CACM)* magazine: [http://cacm.acm.org/](http://cacm.acm.org/)
  - *ACM Queue* magazine for practitioners: [http://queue.acm.org/](http://queue.acm.org/)

- **ACM Digital Library**, the world’s most comprehensive database of computing literature: [http://dl.acm.org](http://dl.acm.org).

- International conferences that draw leading experts on a broad spectrum of computing topics: [http://www.acm.org/conferences](http://www.acm.org/conferences).

- Prestigious awards, including the ACM A.M. Turing and Infosys: [http://awards.acm.org](http://awards.acm.org)

- And much more... [http://www.acm.org](http://www.acm.org).
Measuring Software Development Productivity

www.construx.com

@stevemconstrux
Copyright Notice

These materials are © 2013-2016 Construx Software Builders, Inc.

All Rights Reserved. No part of the contents of this presentation may be reproduced or transmitted in any form or by any means without the written permission of Construx Software Builders, Inc.
I Invite You to Join Me on a Journey
A Journey to Measure Productivity?
A Journey to Measure Productivity!
Why Measure Productivity?
Levels of Productivity Measurement
Why Measure Productivity? (Organizational Level)

- Assess competitiveness with other organizations
- Track and evaluate progress over time
- Support performance evaluation of software executives
- Support bonus allocation among software executives
- Decide allocation of resources to onshore / offshore / outsourced

These are not my reasons; these are reasons given by clients who state they want to measure productivity.
Why Measure Productivity?
(Team/Workgroup Level)
Why Measure Productivity? (Individual Level)

- Support allocation of resources (people) across teams
- Contribute to individual performance review process
- Support allocation of bonuses among individual contributors

These are not my reasons; these are reasons given by clients who state they want to measure productivity.
Potential Issues
Two Potential Issues
What is Productivity?
Productivity = Output / Input

This is simple, but many issues related to productivity can be resolved by referring back to this definition.
What is “Output”

This is the Key Question!

Is "lines of code" an "output" in economic terms?

Are "function points" an "output" in economic terms?

Is work on a project that gets cancelled "output"?

Is work on a project that is delivered successfully but ultimately fails in the marketplace "output"?
Candidate Outputs
Most of these measures of output are:

- Impossible to measure at the individual level
- Extremely difficult to measure at the team level
- Problematic to measure even at the company level
So What Outputs do we Measure?
Candidate Inputs

* Difficult to measure above the individual contributor level
Even the Easy Inputs are More Difficult to Measure Than You Think
Observations about the Simple Definition of Productivity
Observations about the Simple Definition of Productivity

Note we have not found any great solutions to the problem of even defining what input or output we want to measure, much less actually measuring it.
Measurement
Underlying 10x Differences in Productivity
Origin of 10x

We have to talk about “10x” because everything else in measuring software productivity depends on understanding that, first
Purpose of their study was to obtain data on differences in online vs. offline performance. The original goal of their research was thwarted by the fact that individual productivity differences drowned out differences attributable to online vs. offline performance.
Sackman, Erickson, Grant, 1968

All programmers had at least 7 years experience

- Range of initial coding times: 20:1
- Range of "debugging" times: 25:1
- Range of program sizes produced: 5:1
- Range of program execution speeds: 10:1

These differences have continued to plague software engineering research since 1968.
Differences in Productivity

A

B
Differences in Methods

Team A Used

Pair Programming

Team B Used

Formal Inspections

Which method is better?
Differences in Capability

- Team A Had Star Performers
- Team B Had Average Performers

Now which method is better?
Differences in Capability

Team A's Normal Range

Team B's Normal Range

Now which method is better?
Effect of Variations in Human Productivity on Measuring Productivity

Typical Variation in Method Productivity (~20%)

Typical Variation in Individual Productivity (20:1) and Team Productivity (3–10:1)
Selected Other Research

<table>
<thead>
<tr>
<th>Study</th>
<th>Observed Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sackman, Erickson, &amp; Grant 1968</td>
<td>5:1 to 28:1</td>
</tr>
<tr>
<td>Daly, Brooks, et al, 1996</td>
<td>3.2:1 to 7.3:1</td>
</tr>
<tr>
<td>Cartwright &amp; Sheppard, 1998</td>
<td>1.9:1 to 2.2:1</td>
</tr>
<tr>
<td>Curtis, 1981</td>
<td>7.8:1 to 22.3:1</td>
</tr>
<tr>
<td>DeMarco &amp; Lister 1985</td>
<td>5.6:1</td>
</tr>
<tr>
<td>Humphrey 1995</td>
<td>20.4:1</td>
</tr>
<tr>
<td>Boehm 1981</td>
<td>4.2:1</td>
</tr>
<tr>
<td>Boehm 2000</td>
<td>5.3:1</td>
</tr>
<tr>
<td>Card 1987</td>
<td>8.8:1 to 21.6:1</td>
</tr>
</tbody>
</table>
Why Does This Matter?

Why do you want to measure productivity?

If you're measuring to assess the impact of a process, practice, or environmental factor:

- The measurement will be subject to the confounding factor of 10x Variation
- The measurement is not likely to be valid
Why Does This Matter?

Why do you want to measure productivity? Even if you're measuring to assess individual or team productivity, the measurement will be confounded by process, practice, and environmental differences between projects. The measurement is not likely to be accurate.
Measurement in Research vs. Commercial Settings

- Purposes of measurement are different
- Possible ways to measure are different
- The problem just described is a problem even in academic research
- The problem becomes much more significant in commercial settings, because of more confounding factors
- This presentation focuses on measuring productivity in a commercial setting
Evaluating Measures of Individual Productivity
Evaluating Productivity Measures
Common Individual Productivity Measures (for Developers)
Criteria for a Good Individual Productivity Measurement

- Measurement truly reflects "productivity"
- Directly or indirectly accounts for all work output
- Useful for measuring work of non-programmers (e.g., testers), directly or indirectly
- Resists "gaming" by Individual Contributors
- Strongly correlated with business value created
- Objective, independently verifiable
- Measures "output" the same, regardless of programming language used
- Supports cross-project comparisons
- Accounts for best people getting most difficult assignments
- Data can be collected easily and cheaply
Measurement Considerations

Measurement truly reflects "productivity"
Measurement Considerations

Directly or indirectly accounts for most or all work output.
Measurement Considerations

Useful for measuring work of non-programmers (e.g., testers, documenters, scrum masters, business analysts, etc.), directly or indirectly.
Measurement Considerations

I hope this drives the right behavior. I’m going to code myself a new minivan this afternoon.
Measurement Considerations

- Resists "gaming" by Individual Contributors
- This is a big, big, big deal with simplistic measurement approaches
- "What gets measured gets done" (What doesn't get measured doesn't get done)
- Count on work sliding from measured activities into unmeasured activities
- Avoid single-dimension productivity measures
Measurement Considerations

From Facebook

Pie I have eaten
Pie I have not yet eaten
Measurement Considerations
Measurement Considerations

Construx®

Loc/FP

<table>
<thead>
<tr>
<th>Language</th>
<th>LOC/FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>128</td>
</tr>
<tr>
<td>C++</td>
<td>53</td>
</tr>
<tr>
<td>Cobol</td>
<td>91</td>
</tr>
<tr>
<td>Java</td>
<td>53</td>
</tr>
<tr>
<td>VB</td>
<td>29</td>
</tr>
</tbody>
</table>

...
Measurement Considerations

[Images of Gantt charts]
Measurement Considerations

2.0 = (v)sin(35) + \frac{(v^2)}{2} + \frac{5}{2}(-1.8)(15)^2

\sqrt{2} = \sin(15\degree)

Construx®
Measurement Considerations

Data can be collected easily and cheaply.
Evaluating the Measures: Scale

- Excellent / 5
- Good / 4
- Neither Good nor Bad / 3
- Bad / 2
- Terrible / 1
Evaluating the Measures

The following scoring is subjective. But it is explicit. It is structured.
## Comparison of Individual (Developer) Productivity Measures

<table>
<thead>
<tr>
<th></th>
<th>LOC/SM</th>
<th>FP/SM</th>
<th>Story points / staff month</th>
<th>360 degree Peer Reviews</th>
<th>Manager Evaluation</th>
<th>Task Completion Predict'y</th>
<th>Test Cases Passed</th>
<th>Defect Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement truly reflects “productivity”</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Directly or indirectly accounts for all or most work output</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Useful for measuring work of non-programmers (e.g., testers)</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Resists “gaming” by Individual Contributors</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Strongly correlated with business value created</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Objective, independently verifiable</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Measures &quot;output&quot; the same, regardless of programming language used</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Supports person-to-person comparisons within a team</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Accounts for best people getting most difficult assignments</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Data can be collected easily and cheaply</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>2.5</td>
<td>3.8</td>
<td>3.8</td>
<td>3.9</td>
<td>3.9</td>
<td>3.1</td>
<td>4.1</td>
<td>3.7</td>
</tr>
<tr>
<td><strong>Rank</strong></td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>
### Comparison of Individual (Developer) Productivity Measures

<table>
<thead>
<tr>
<th>Measurement</th>
<th>LOC/SM</th>
<th>FP/SM</th>
<th>Story points / staff month</th>
<th>360 degree Peer Reviews</th>
<th>Manager Evaluation</th>
<th>Task Completion Predict’y</th>
<th>Test Cases Passed</th>
<th>Defect Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement truly reflects “productivity”</td>
<td>○</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Directly or indirectly accounts for all or most work output</td>
<td>▲</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Useful for measuring work of non-programmers (e.g., testers)</td>
<td>▼</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
</tr>
<tr>
<td>Resists “gaming” by Individual Contributors</td>
<td>▼</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Strongly correlated with business value created</td>
<td>▼</td>
<td>○</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
</tr>
<tr>
<td>Objective, independently verifiable</td>
<td>▲</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Measures &quot;output&quot; the same, regardless of programming language used</td>
<td>▼</td>
<td>○</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
</tr>
<tr>
<td>Supports person-to-person comparisons within a team</td>
<td>○</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
</tr>
<tr>
<td>Accounts for best people getting most difficult assignments</td>
<td>▼</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Data can be collected easily and cheaply</td>
<td>○</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>○</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
</tr>
<tr>
<td><strong>Rank</strong></td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>
Observations about Individual Productivity Measures

Loc/SM is easily the worst measure (score of 2.5)
Observations about Individual Productivity Measures
Observations about Individual Productivity Measures

The top 6 measures are closely ranked (3.7 - 4.1)
Observations about Individual Productivity Measures

“Test cases passed” resists gaming if you have independent testing; without independent testing its overall score drops to 3.9 (i.e., moves into 6-way tie for 1st place (3.7-3.9))
Observations about Individual Productivity Measures

Manager evaluation" does better than you might expect as a "measure," it has the best "effort" rating, and is normally the most readily available...
Steve’s Conclusion

The business problems that individual productivity “measurement” needs to address can be addressed more effectively by a non-measurement technique.
Evaluating Measures of Team Productivity
Possible Team Productivity Measures

* Different from individual measures
Team-Level Measurement Considerations

- Measurement truly reflects “productivity”
- Directly or indirectly accounts for all work output
- Useful for measuring work of the whole team, directly or indirectly
- Resists gaming by the team
- Strongly correlated with business value created
- Objective, independently verifiable
- Measures “output” the same, regardless of programming language used
- Accurately reflects output of teams working on diverse kinds of projects
- Accounts for best teams getting most difficult assignments
- Data can be collected easily and cheaply

* Different from individual measures
## Comparison of **Team-Level** Productivity Measures

<table>
<thead>
<tr>
<th>Feature</th>
<th>LOC/SM</th>
<th>FP/SM</th>
<th>Story points / staff month</th>
<th>Manager Stack-Rank Evals</th>
<th>Project Completion Predict’y</th>
<th>Test Cases Passed</th>
<th>Defect Rates</th>
<th>Score Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement truly reflects “productivity”</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Directly or indirectly accounts for all work output</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Useful for measuring work of whole team</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Resists gaming by the team</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Strongly correlated with business value created</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Objective, independently verifiable</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Measures &quot;output&quot; the same, regardless of programming language</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Accurately reflects output of teams working on diverse projects</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Accounts for best teams getting most difficult assignments</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Data can be collected easily and cheaply</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**Average**

<table>
<thead>
<tr>
<th>Measure</th>
<th>Average</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC/SM</td>
<td>2.4</td>
<td>8</td>
</tr>
<tr>
<td>FP/SM</td>
<td>3.8</td>
<td>3</td>
</tr>
<tr>
<td>Story points / staff month</td>
<td>3.1</td>
<td>7</td>
</tr>
<tr>
<td>Manager Stack-Rank Evals</td>
<td>3.8</td>
<td>3</td>
</tr>
<tr>
<td>Project Completion Predict’y</td>
<td>3.8</td>
<td>3</td>
</tr>
<tr>
<td>Test Cases Passed</td>
<td>3.7</td>
<td>6</td>
</tr>
<tr>
<td>Defect Rates</td>
<td>3.9</td>
<td>2</td>
</tr>
<tr>
<td>Score Card</td>
<td>4.5</td>
<td>1</td>
</tr>
</tbody>
</table>
## Comparison of **Team-Level** Productivity Measures

<table>
<thead>
<tr>
<th>Feature</th>
<th>LOC/SM</th>
<th>FP/SM</th>
<th>Story points / staff month</th>
<th>Manager Stack-Rank Evals</th>
<th>Project Completion Predict’y</th>
<th>Test Cases Passed</th>
<th>Defect Rates</th>
<th>Score Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement truly reflects “productivity”</td>
<td>○</td>
<td>▲</td>
<td>▼</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>▲</td>
</tr>
<tr>
<td>Directly or indirectly accounts for all work output</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Useful for measuring work of whole team</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Resists gaming by the team</td>
<td>▼</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Strongly correlated with business value created</td>
<td>▼</td>
<td>○</td>
<td>○</td>
<td>▲</td>
<td>○</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Objective, independently verifiable</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>○</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Measures &quot;output&quot; the same, regardless of programming language</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Accurately reflects output of teams working on diverse projects</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Accounts for best teams getting most difficult assignments</td>
<td>○</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Data can be collected easily and cheaply</td>
<td>○</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>○</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td><strong>Rank</strong></td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Observations about Measures of Team Productivity

Again, LOC/SM is easily the worst measure (score of 2.4)

Score card is clear #1 (score of 4.5)

The next 5 measures are closely ranked (3.7 - 3.9)

"Test cases passed" is more susceptible to gaming at the team level than at the individual level
Scorecard Approach: What **Output** do we want from the Team?

<table>
<thead>
<tr>
<th>Category</th>
<th>Possible Score</th>
<th>Actual Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-Time Delivery</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Initial Defect Rate</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>+90 Day Defect Rate</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>% of Planned Functionality Delivered</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Early notification of problems</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Executive’s Satisfaction with Project Execution and Delivery</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Sales’ Satisfaction with Project Execution and Delivery</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>60</strong></td>
<td><strong>53</strong></td>
</tr>
</tbody>
</table>

(88%)
Comments about the Scorecard

[Content]

- Only significant weakness of the scorecard is its independent verifiability. It is not "measurement" per se, but...".

- It is structured such that it avoids over-optimization for one measure and can provide for cross-project comparisons.

- It can be made public and can be reviewed.

- The scorecard can and should be tailored to support organizational goals.
Conclusions
Conclusions

True productivity measures in software are significantly limited:

- Agreeing on a definition of productivity is significantly problematic.
- Meaningful outputs are difficult or impossible to measure.
- Real business outputs are rarely measured, instead proxies for the real output are measured and are subject to significant measurement error.
- Inputs are subject to significant measurement error.
- The whole measurement exercise is subject to massive measurement error because of the "10x variation" phenomenon.
The End of Our Journey

The questions that businesses want to address through measuring productivity can be addressed effectively through non-measurement or quasi-measurement approaches.

These alternative approaches stack up very favorably vs. measurement, especially when you account fully for the limitations involved in true measurement.
Construx Software is committed to helping individuals and organizations improve their software development practices. For information about our training and consulting services, contact stevemcc@construx.com.
Questions about this webcast? learning@acm.org

ACM Learning Webinars (on-demand archive): http://learning.acm.org/webinar

ACM Learning Center: http://learning.acm.org

ACM SIGSOFT: http://sigsoft.org/

ACM Queue: http://queue.acm.org/