BEYOND THE HYPE:
EVALUATING AND MEASURING AGILE DEVELOPMENT

© Quantitative Software Management, Inc.
Some silver bullets. Is Agile one?
What are the characteristics of successful projects?
The promise of Agile
Agile staffing
Agile effort
Agile schedule
Agile productivity
Agile quality
Measuring and tracking Agile projects
• “There is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, in reliability, in simplicity.”
  - Frederick Brooks in “No Silver Bullet – Essence and Accidents of Software Engineering”

• “There is no new thing under the sun”
  - Ecclesiastes 1.9

• “It has always been agile”
  - Philip G. Armour
Some Silver Bullets

- Structured programming
- Lifecycle methodologies
- CASE tools, Code generators
- 3gl, 4gl, ... languages
- Object oriented programming
- Graphical user interface (GUI)
- ERP packages
- CMMI
- Service oriented architecture (SOA)
- Cloud computing
- Outsourcing
Silver Bullet Problems

• Neither individually nor in concert with others have the “silver bullets” produced more than linear improvement in productivity, quality, or time to market

• Offer technical solutions to a non-technical problem
  ▪ Paradigm has been to transform custom artisan work into assembly line production
  ▪ Software is not a manufacturing process. Solutions designed to improve manufacturing are not applicable to software development

• Software: a knowledge acquisition process with a technical component
Characteristics of Successful Projects

• Case study Best projects vs. Worst projects
  - **Best projects** defined as those that are more than 1\(\sigma\) (standard deviation) better than average for both time to market and cost/effort
  - **Worst projects** are 1\(\sigma\) worse than average for both time to market and cost/effort
  - Projects evaluated on 58 criteria in Tools & Methods, Technical Complexity, Personnel, and Re-use
Best Project/ Worst Projects

Worst Projects

Best Projects

Schedule vs Size

Worst Projects

Best Projects

Effort vs Size

Worst Projects

Best Projects

(#7) 7/12/2011
Differentiators

- People,
- Communication,
- Knowledge

- Management Efficiency
- Staff Turnover
- Team Skill
- Motivation
- Cohesiveness
- Communication
- Knowledge
- Complexity
- Overall Difficulty
- Customer Interface
- Documentation Rqmts
- Tools
- Programming Tools Capability

Average Value of Metrics

Average 7/12/2011
Things that Don’t Matter

Average Value of Metrics

- Data Complexity: 5.7
- Integration Complexity: 4.0
- Hardware Stability: 9.0
- System Software Stability: 7.2
- Overall Tools Capability: 5.8
- Project Mgt Tools Capability: 5.9
- Development Standards Experience: 6.5

Average

(#9) 7/12/2011
The Promise of Agile

WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.

THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.

I'M GLAD IT HAS A NAME.

THAT WAS YOUR TRAINING.
The Promise of Agile: Agile Manifesto

- Individuals and Interactions over processes and tools
- Working Software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- Key traits
  - Frequent delivery
  - Business people and developers work together daily
  - Face to face conversations
The Promise of Agile

• It appears that Agile development embraces the People, Knowledge, and Communication traits that were found in highly successful projects.
• Agile is very focused on the social component of software development.
• So, how well do Agile projects compare to traditional development?
Demographics

- 54 recently completed Agile projects
- 12 different companies
- 87% business, 7% scientific applications, 6% system software
- Team size clustered in 5-10 and 20-50 ranges
- Median size 42.9k lines of code
- Median effort 47 staff months
- Median staff 7.5
- Median duration 6.1 months
- Principally new development and major enhancements
Agile Staffing

**Comparison of Projects being Assessed to QSM Business Average Staff vs Effective SLOC**

<table>
<thead>
<tr>
<th>C&amp;T Average Staff (People) Values</th>
<th>at Min</th>
<th>at 25% Quartile</th>
<th>at Median</th>
<th>at 75% Quartile</th>
<th>at Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark Reference Group:</td>
<td>5840</td>
<td>18838</td>
<td>42870</td>
<td>122444</td>
<td>952614</td>
</tr>
<tr>
<td>QSM Business</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comparison Data Set:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projects being Assessed</td>
<td>3.40</td>
<td>6.21</td>
<td>9.03</td>
<td>14.58</td>
<td>37.16</td>
</tr>
<tr>
<td>Difference From Benchmark</td>
<td>0.50</td>
<td>1.18</td>
<td>1.94</td>
<td>3.59</td>
<td>11.26</td>
</tr>
</tbody>
</table>

Comparison breakpoints based on min, max, median and quartile values for the data set: Projects being Assessed

The blue trend lines in this and subsequent graphs are the QSM business average with plus & minus 1 standard deviation. The red line is the Agile dataset average.
Agile Staffing Observations

• The agile projects use slightly more staff than non-agile business projects although the trend is very similar
Agile and non-Agile projects use nearly the same amount of project effort for projects with similar amounts of delivered functionality.
Agile Schedule Length

**Agile projects complete much more rapidly**
Agile Schedule Observations

- Agile projects complete much more quickly than non-agile projects while expending about the same amount of effort (Cost)
- Since schedule is frequently an important project driver, this is a significant advantage
Agile Productivity Index (PI)

Comparison of Projects being Assessed to QSM Business
PI vs. Effective SLOC

Comparison breakpoints based on min, max, median and quartile values for the data set: Projects being Assessed.

<table>
<thead>
<tr>
<th>PI Values</th>
<th>at Min</th>
<th>at 25% Quartile</th>
<th>at Median</th>
<th>at 75% Quartile</th>
<th>at Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSM Business</td>
<td>5040</td>
<td>18838</td>
<td>42870</td>
<td>122444</td>
<td>952614</td>
</tr>
<tr>
<td>Projects being Assessed</td>
<td>13.50</td>
<td>16.22</td>
<td>17.92</td>
<td>19.93</td>
<td>20.08</td>
</tr>
<tr>
<td>Difference From Benchmark</td>
<td>1.88</td>
<td>1.97</td>
<td>2.02</td>
<td>2.13</td>
<td>2.21</td>
</tr>
</tbody>
</table>

Productivity indices for Agile projects were significantly higher than the business average.

(#19) 7/12/2011
Agile Quality

Defects Found in Testing

- Defects vs. Effective SLOC (thousands)

Comparison of Projects being Assessed to QSM Business
Errors (SysInt-Del) vs. Effective SLOC

<table>
<thead>
<tr>
<th>Benchmark Reference Group:</th>
<th>QSM Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparison Data Set:</td>
<td>Projects being Assessed</td>
</tr>
<tr>
<td>Difference From Benchmark</td>
<td>-4.44</td>
</tr>
<tr>
<td>at Min Effective SLOC:</td>
<td>12240</td>
</tr>
<tr>
<td>at 25% Quartile Effective SLOC:</td>
<td>28013</td>
</tr>
<tr>
<td>at Median Effective SLOC:</td>
<td>101274</td>
</tr>
<tr>
<td>at 75% Quartile Effective SLOC:</td>
<td>254563</td>
</tr>
<tr>
<td>at Max Effective SLOC:</td>
<td>952614</td>
</tr>
</tbody>
</table>

Errors (SysInt-Del) Values

- Agile projects produced fewer defects

(#20) 7/12/2011
## In Summary

### Typical Sized Agile and Business IT Projects

<table>
<thead>
<tr>
<th></th>
<th>Agile</th>
<th>Business IT</th>
<th>Difference</th>
<th>%Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size in SLOC</td>
<td>42,900</td>
<td>42,900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Staff</td>
<td>9</td>
<td>7.1</td>
<td>1.9</td>
<td>26.8%</td>
</tr>
<tr>
<td>Devel. Duration (Mths)</td>
<td>4.3</td>
<td>6.1</td>
<td>-1.8</td>
<td>-29.5%</td>
</tr>
<tr>
<td>Effort Months</td>
<td>39</td>
<td>43</td>
<td>-4.0</td>
<td>-9.3%</td>
</tr>
<tr>
<td>Defects (testing)</td>
<td>152</td>
<td>245</td>
<td>-93.0</td>
<td>-38.0%</td>
</tr>
<tr>
<td>Productivity Index</td>
<td>19.93</td>
<td>17.92</td>
<td>2.0</td>
<td>11.2%</td>
</tr>
</tbody>
</table>

- Agile projects outperform conventional development in Productivity, Quality, and Time to Market
- Staffing levels are higher; but overall effort is slightly lower while achieving significant schedule compression
Measuring and Tracking

Agile Projects
Issues

• Agile definitely suffers from the “We’re not like other software development so we can’t be measured or tracked like them” syndrome

• Large projects may require formality in documentation and procedures that nullify Agile advantages

• Current business practices often conflict with Agile methods
  ▪ Outsourcing
  ▪ Splitting teams into onshore/offshore groups
  ▪ Multi-site development

• Agile methods are the key to the results we have seen
Overview

- Estimating size of Agile projects
  - Stories, story points, & lines of code
- Estimating Agile projects
  - One project with multiple iterations (sprints)
  - One project per iteration
- Tracking Agile projects
Estimating Size of Agile Projects

• **Story Points**
  - A relative size measure
  - No standard criteria for definition

• **Lines of Code**
  - Lines of code or their equivalent (implementation units) are the basis for sizing in all major parametric estimation tools
  - Not intuitive and difficult to accurately estimate beforehand
Estimating Size of Agile Projects

• Stories
  ▪ Discreet groups of functionality
  ▪ Sprints typically bundle a number of stories
  ▪ Stories that are not completed within the time frame of the sprint are moved to another sprint
  ▪ Sprints are time boxed: additional sprints may be added to a project; but a sprint will not be lengthened to complete work

• QSM has captured lines of code for completed sprints and has developed gearing factors for stories
  ▪ These allow Agile projects to be estimated by SLIM
Both stories and code created were measured throughout the project. We were able to determine gearing factors and refine our code estimates by the end of the second iteration.
## Code & Story Data by Iteration

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Code Per release</th>
<th>Stories Completed</th>
<th>Story Gearing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration 0</td>
<td>14295</td>
<td>21</td>
<td>680.7</td>
</tr>
<tr>
<td>Iteration 1</td>
<td>13165</td>
<td>22</td>
<td>598.4</td>
</tr>
<tr>
<td>Iteration 2</td>
<td>20130</td>
<td>22</td>
<td>915.0</td>
</tr>
<tr>
<td>Iteration 3</td>
<td>15794</td>
<td>27</td>
<td>585.0</td>
</tr>
<tr>
<td>Iteration 4</td>
<td>13348</td>
<td>36</td>
<td>370.8</td>
</tr>
<tr>
<td>Iteration 5</td>
<td>17940</td>
<td>35</td>
<td>512.6</td>
</tr>
<tr>
<td>Average</td>
<td>15779</td>
<td>27</td>
<td>610.4</td>
</tr>
</tbody>
</table>

The data in the table above was used to determine starting point gearing factors for stories:

- Low complexity: 370
- Average complexity: 610
- High complexity: 915

This process can be used to determine appropriate gearing factors in different environments.
Estimating Agile Projects

• Entire Agile development effort may be modeled as one estimate with milestones for the iterations (see slide 30)
  ▪ Track progress and adjust schedule based on performance (Are stories being deferred to future sprints?)

• Each Iteration (Sprint) can be an estimate
  ▪ These are combined to provide a program level view (see slide 31)
Estimating One Project Multiple Iterations

Six Iterations

Iteration Development

Iteration Planning throughout the project

Go Live support

Avg Staff (people)
<Solve for PI Wizard>

0 1 2 3 4 5

Apr '10 May Jun Jul Aug Sep Oct Nov Dec Jan '11 Feb Mar

(#30) 7/24/2009
Estimating
One Estimate per Iteration

Estimates are combined in SLIM MasterPlan to provide a program level view.
Tracking Agile Projects

Core Metrics View

- **FTE Staffing Total Project Effort**
- **Cumulative Effort Total Project Effort**
- **StoriesCompleted**
- **Defects Found**

Overall view of effort, defects, and stories completed
This slide shows the stories completed by iteration
Here are tracking metrics for a single iteration within the project
QUESTIONS?