Quantitative Software Management, Inc.
2000 Corporate Ridge, Suite 900
Mclean, VA 22102
703.790.0055 • 703.749.3795 (fax)
info@qsm.com • www.qsm.com

BEYOND THE HYPE: EVALUATING AND MEASURING AGILE DEVELOPMENT

Outline

- 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

Introduction

- "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

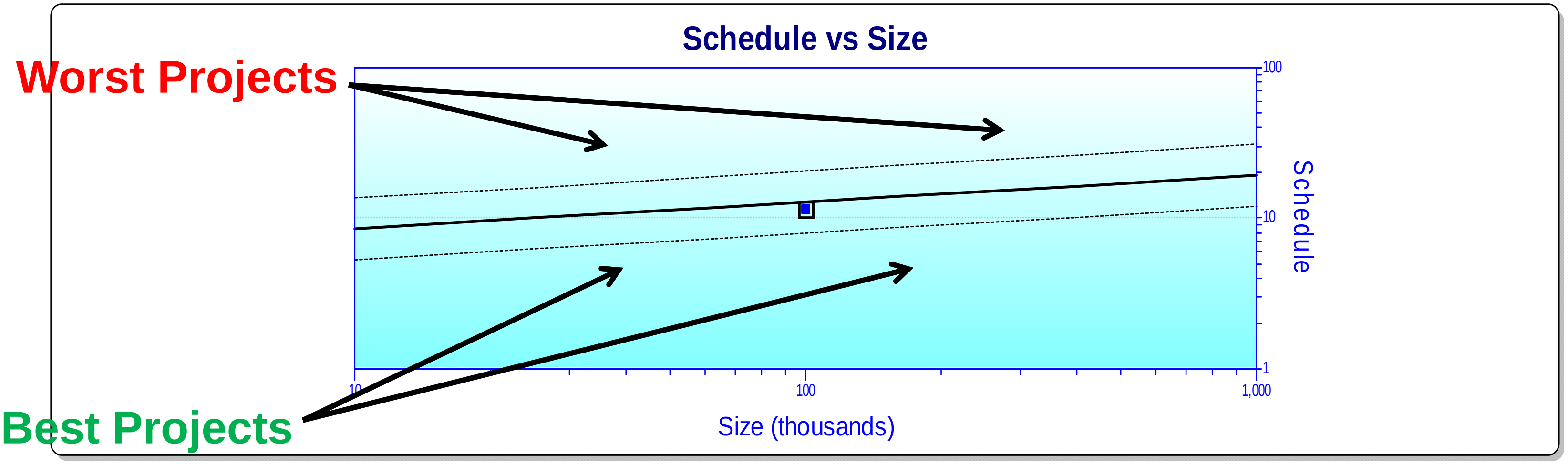
- Neither individually nor in concert with others have the “silver bullets” produced more than linear improvement in productivity, quality, or time to market
- Offer technical solutions to a non-technical problem
  - Paradigm has been to transform custom artisan work into assembly line production
  - 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 1σ (standard deviation) better than average for **both** time to market and cost/effort
  - **Worst projects** are 1σ 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

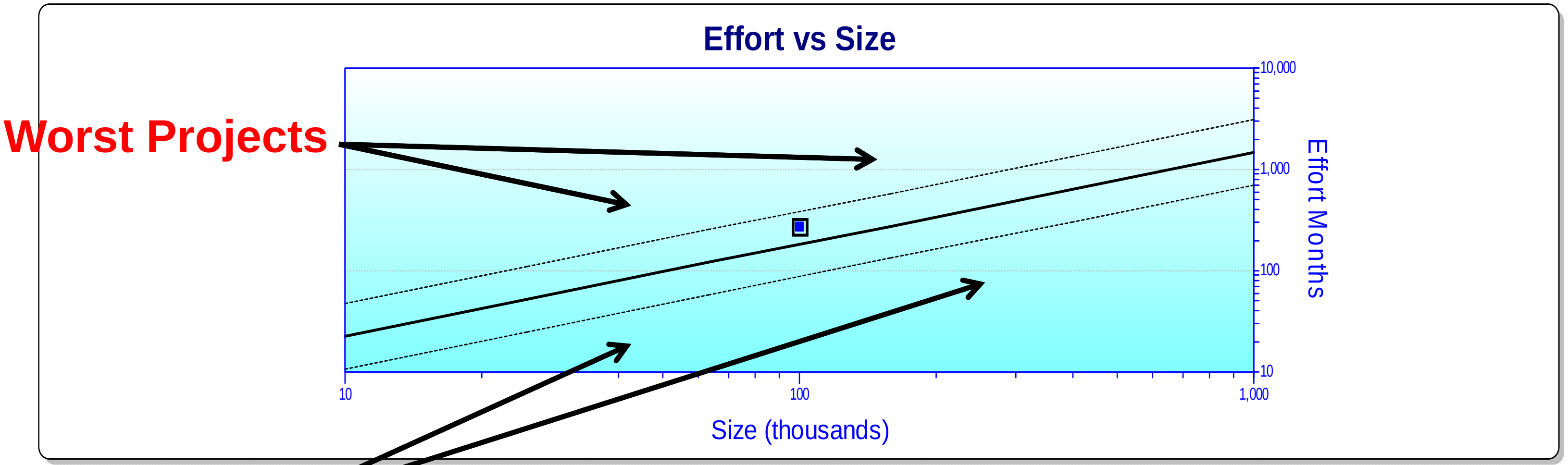
Best Project / Worst Projects

Worst Projects



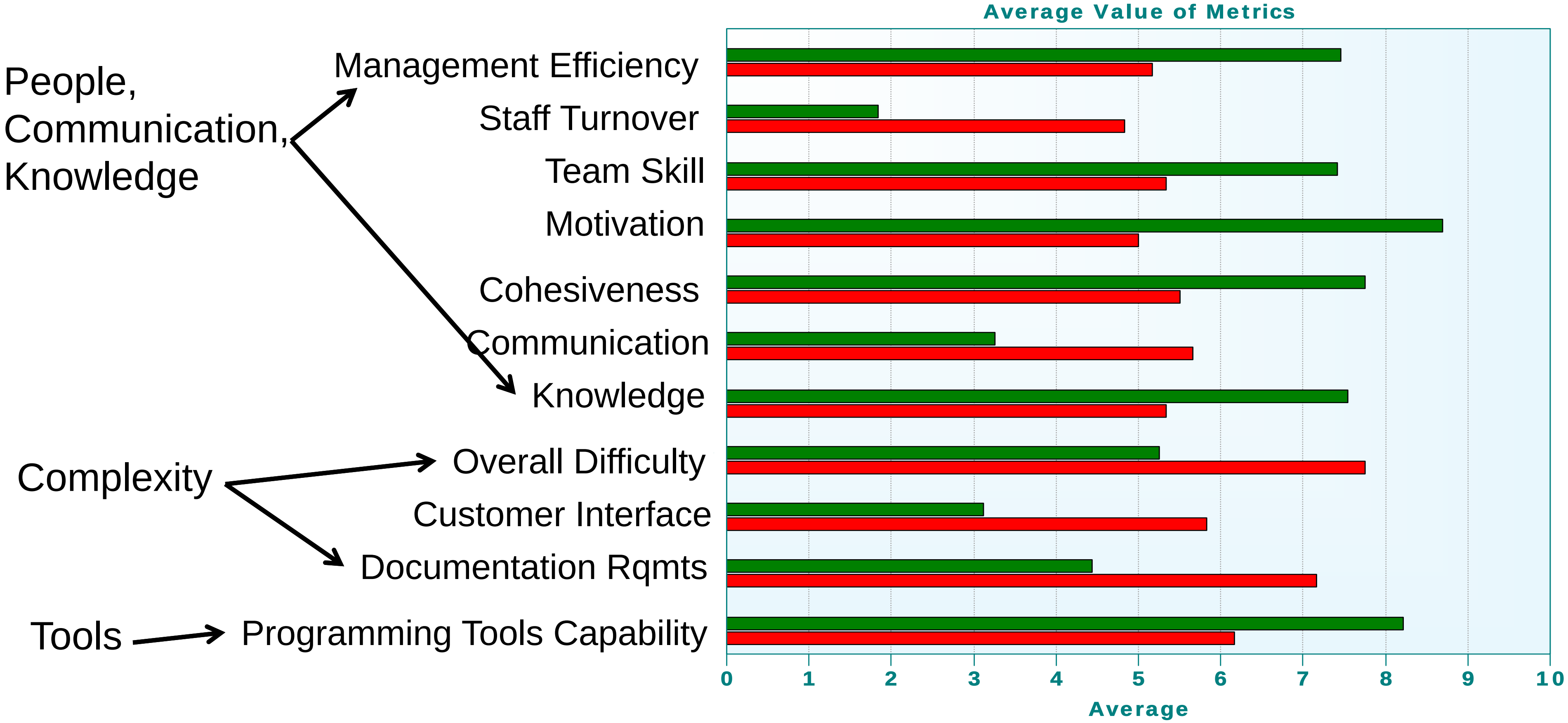
Best Projects

Worst Projects

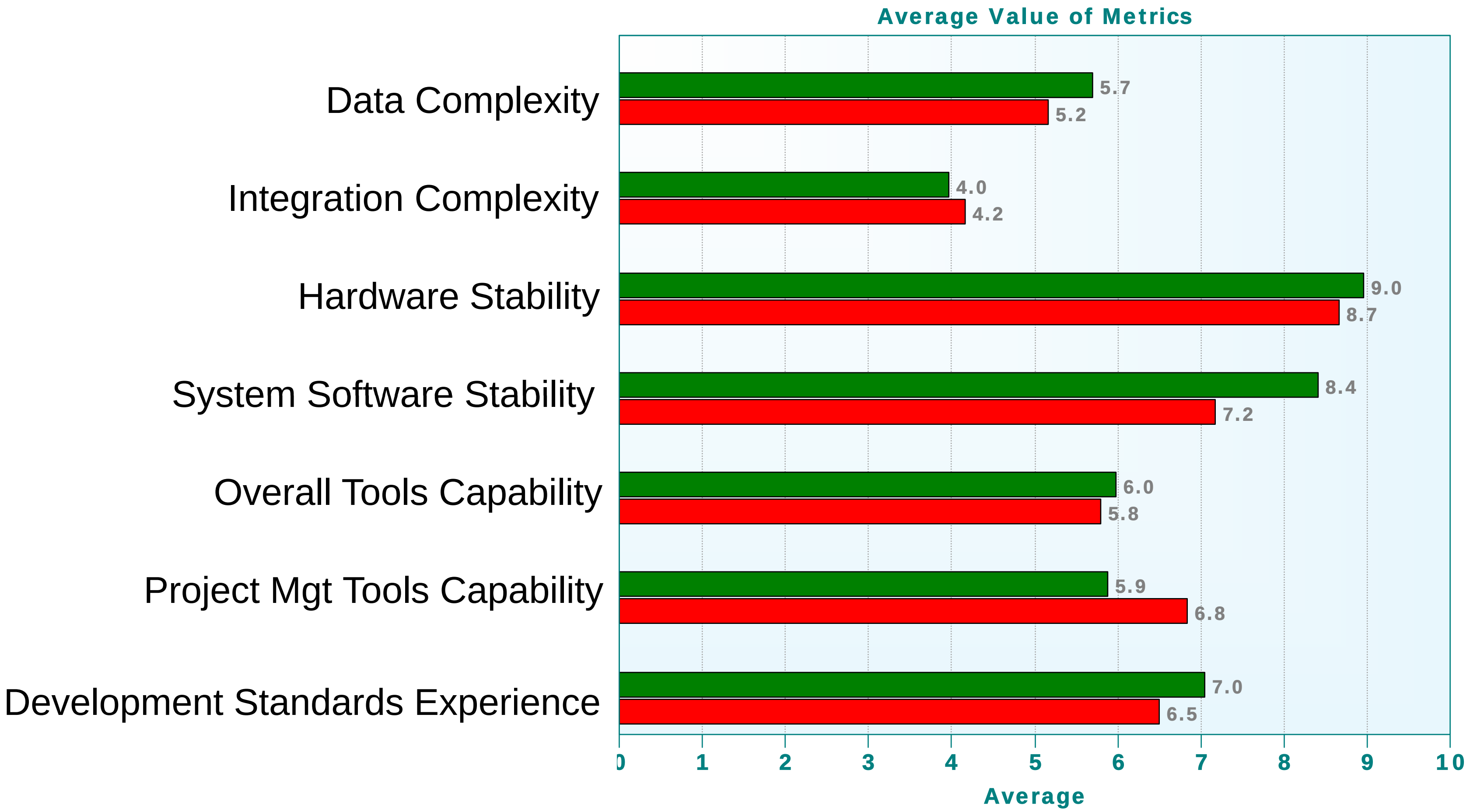


Best Projects

Differentiators



Things that Don’t Matter



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

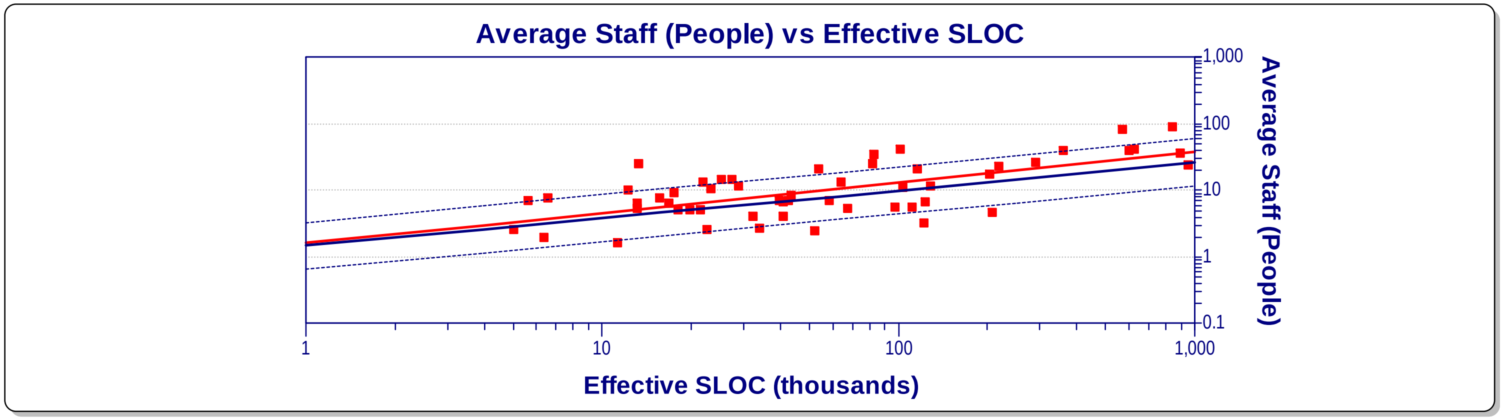
The Promise of Agile

- 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?

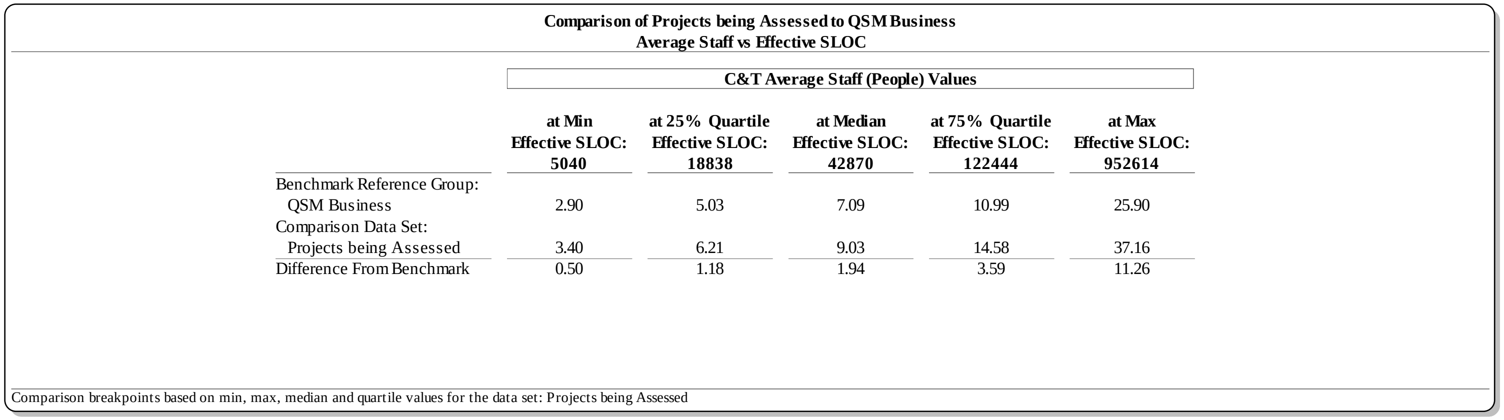
Demographics

- 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



Average Staff (People) vs Effective SLOC



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

	C&T Average Staff (People) Values				
	at Min Effective SLOC: 5040	at 25% Quartile Effective SLOC: 18838	at Median Effective SLOC: 42870	at 75% Quartile Effective SLOC: 122444	at Max Effective SLOC: 952614
Benchmark Reference Group:					
QSM Business	2.90	5.03	7.09	10.99	25.90
Comparison Data Set:					
Projects being Assessed	3.40	6.21	9.03	14.58	37.16
Difference From Benchmark	0.50	1.18	1.94	3.59	11.26

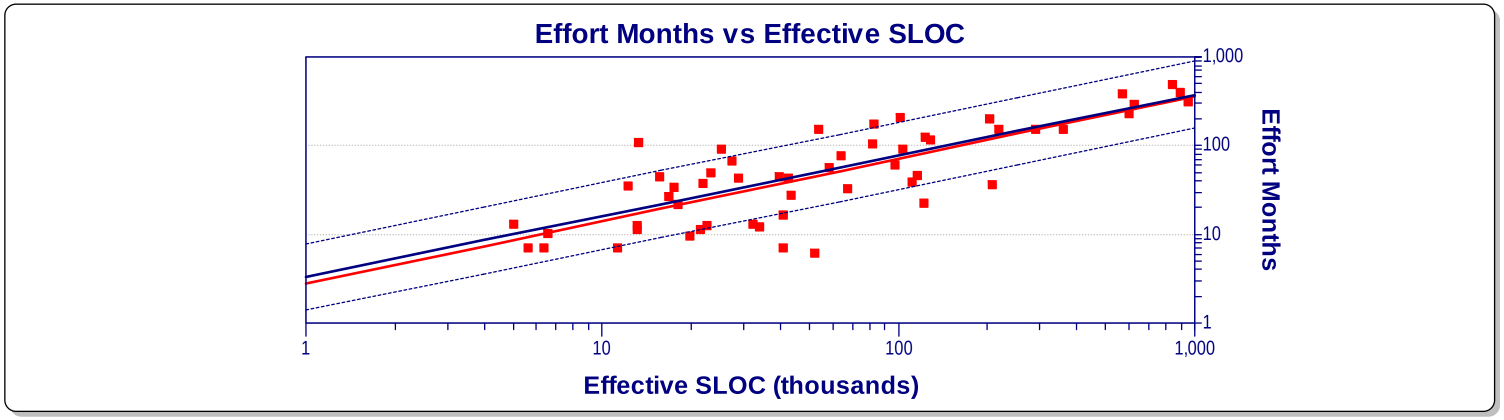
Comparison breakpoints based on min, max, median and quartile values for the data set: Projects being Assessed

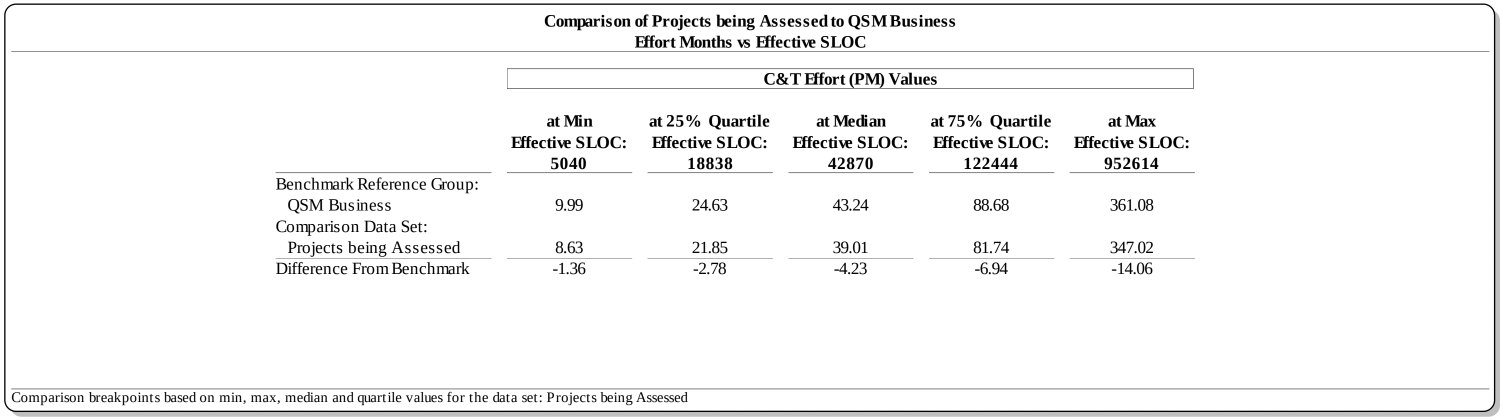
The blue trend lines in this and subsequent graphs are the QSM business average with plus & minus 1 standard deviation. The red line is the Agile dataset average

Agile Staffing Observations

- The agile projects use slightly more staff than non-agile business projects although the trend is very similar

Agile Effort





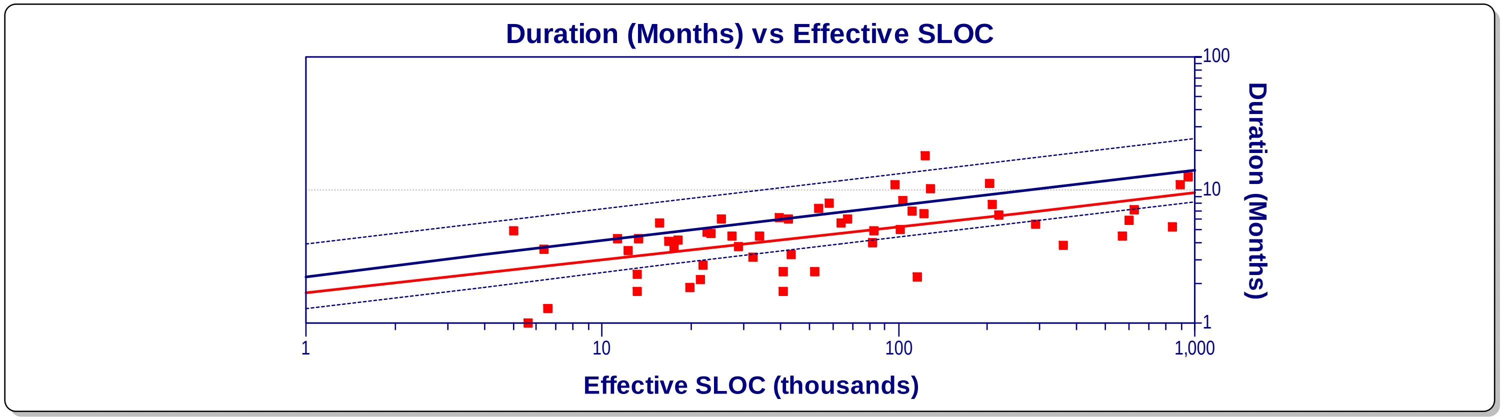
Comparison of Projects being Assessed to QSM Business
Effort Months vs Effective SLOC

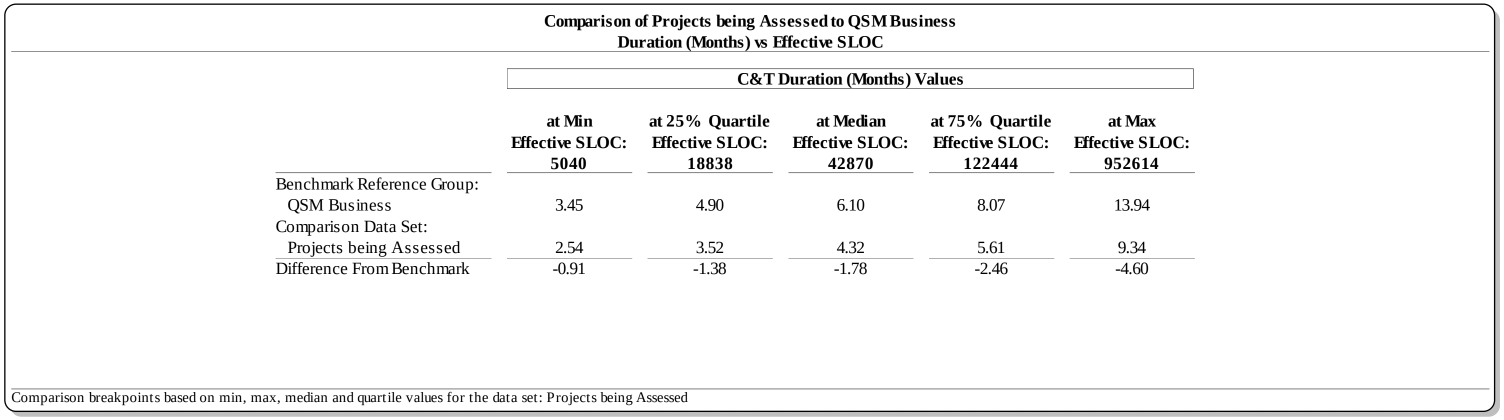
	C&T Effort (PM) Values				
	at Min Effective SLOC: 5040	at 25% Quartile Effective SLOC: 18838	at Median Effective SLOC: 42870	at 75% Quartile Effective SLOC: 122444	at Max Effective SLOC: 952614
Benchmark Reference Group:					
QSM Business	9.99	24.63	43.24	88.68	361.08
Comparison Data Set:					
Projects being Assessed	8.63	21.85	39.01	81.74	347.02
Difference From Benchmark	-1.36	-2.78	-4.23	-6.94	-14.06

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

Agile and non-Agile projects use nearly the same amount of project effort for projects with similar amounts of delivered functionality

Agile Schedule Length




Comparison of Projects being Assessed to QSM Business
Duration (Months) vs Effective SLOC

	C&T Duration (Months) Values				
	at Min Effective SLOC: 5040	**at 25% Quartile Effective SLOC: 18838**	**at Median Effective SLOC: 42870**	**at 75% Quartile Effective SLOC: 122444**	**at Max Effective SLOC: 952614**
Benchmark Reference Group:					
QSM Business	3.45	4.90	6.10	8.07	13.94
Comparison Data Set:					
Projects being Assessed	2.54	3.52	4.32	5.61	9.34
Difference From Benchmark	-0.91	-1.38	-1.78	-2.46	-4.60

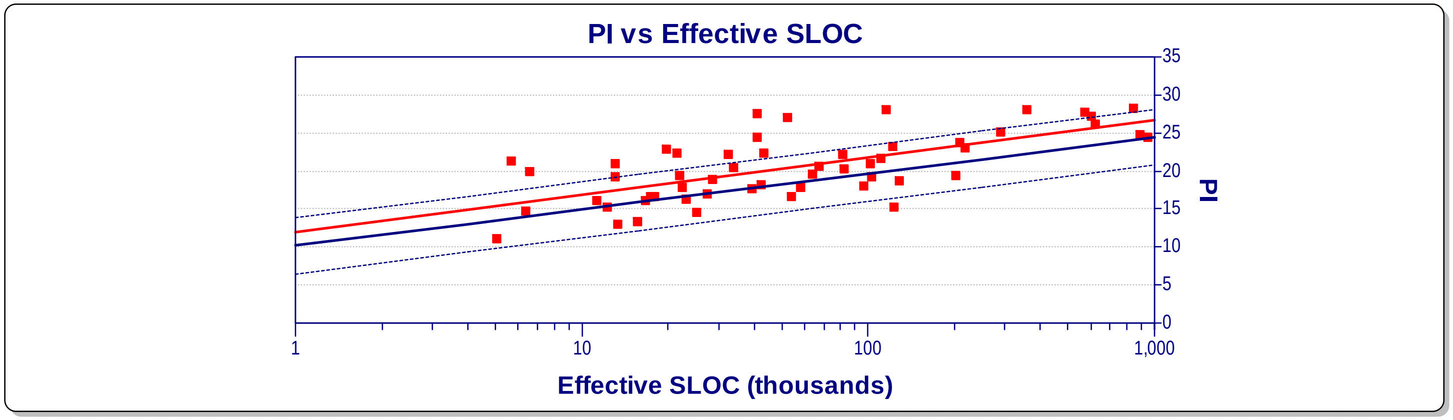
Comparison breakpoints based on min, max, median and quartile values for the data set: Projects being Assessed

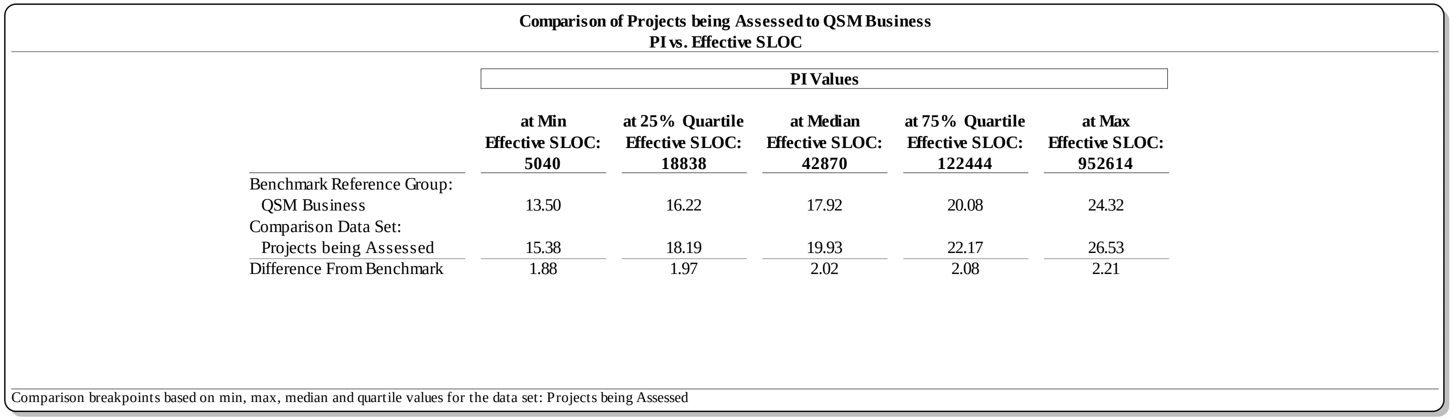
Agile projects complete much more rapidly

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)





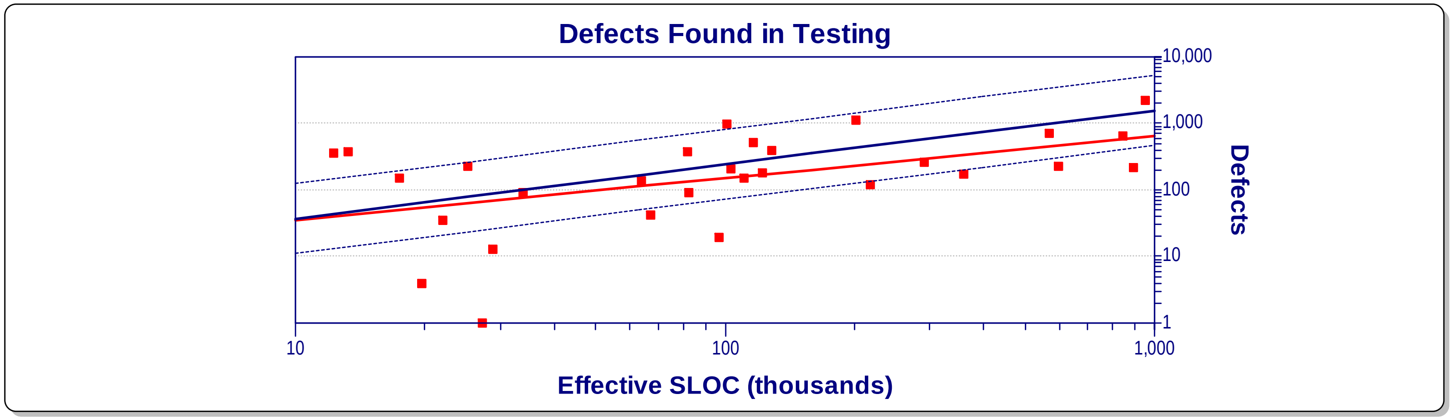
Comparison of Projects being Assessed to QSM Business
PI vs. Effective SLOC

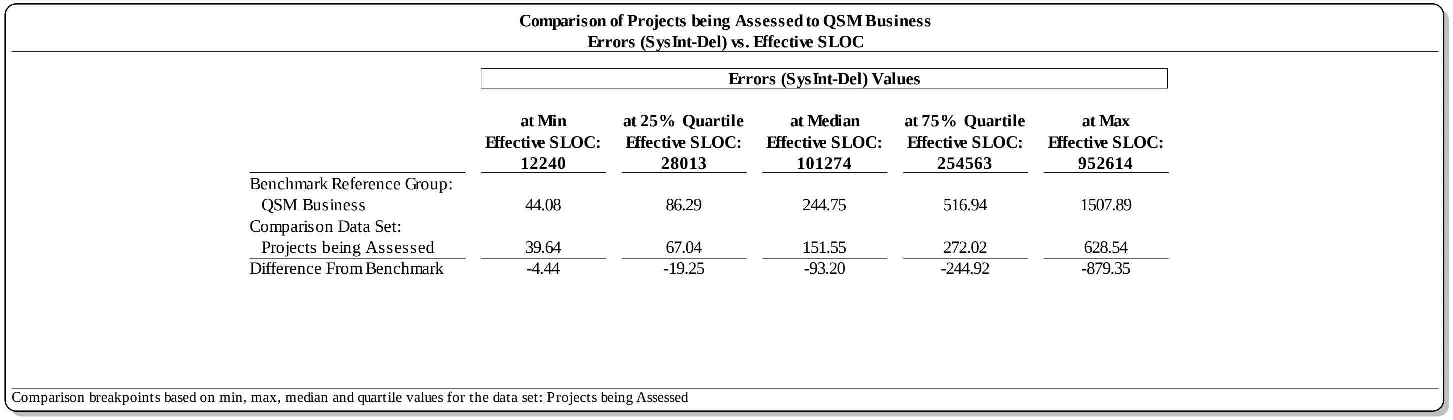
	PI Values				
	at Min Effective SLOC: 5040	**at 25% Quartile Effective SLOC: 18838**	**at Median Effective SLOC: 42870**	**at 75% Quartile Effective SLOC: 122444**	**at Max Effective SLOC: 952614**
Benchmark Reference Group: QSM Business	13.50	16.22	17.92	20.08	24.32
Comparison Data Set: Projects being Assessed	15.38	18.19	19.93	22.17	26.53
Difference From Benchmark	1.88	1.97	2.02	2.08	2.21

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

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

Agile Quality




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

	Errors (SysInt-Del) Values				
	at Min Effective SLOC: 12240	**at 25% Quartile Effective SLOC: 28013**	**at Median Effective SLOC: 101274**	**at 75% Quartile Effective SLOC: 254563**	**at Max Effective SLOC: 952614**
Benchmark Reference Group:					
QSM Business	44.08	86.29	244.75	516.94	1507.89
Comparison Data Set:					
Projects being Assessed	39.64	67.04	151.55	272.02	628.54
Difference From Benchmark	-4.44	-19.25	-93.20	-244.92	-879.35

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

Agile projects produced fewer defects

In Summary

Typical Sized Agile and Business IT Projects

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

- 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

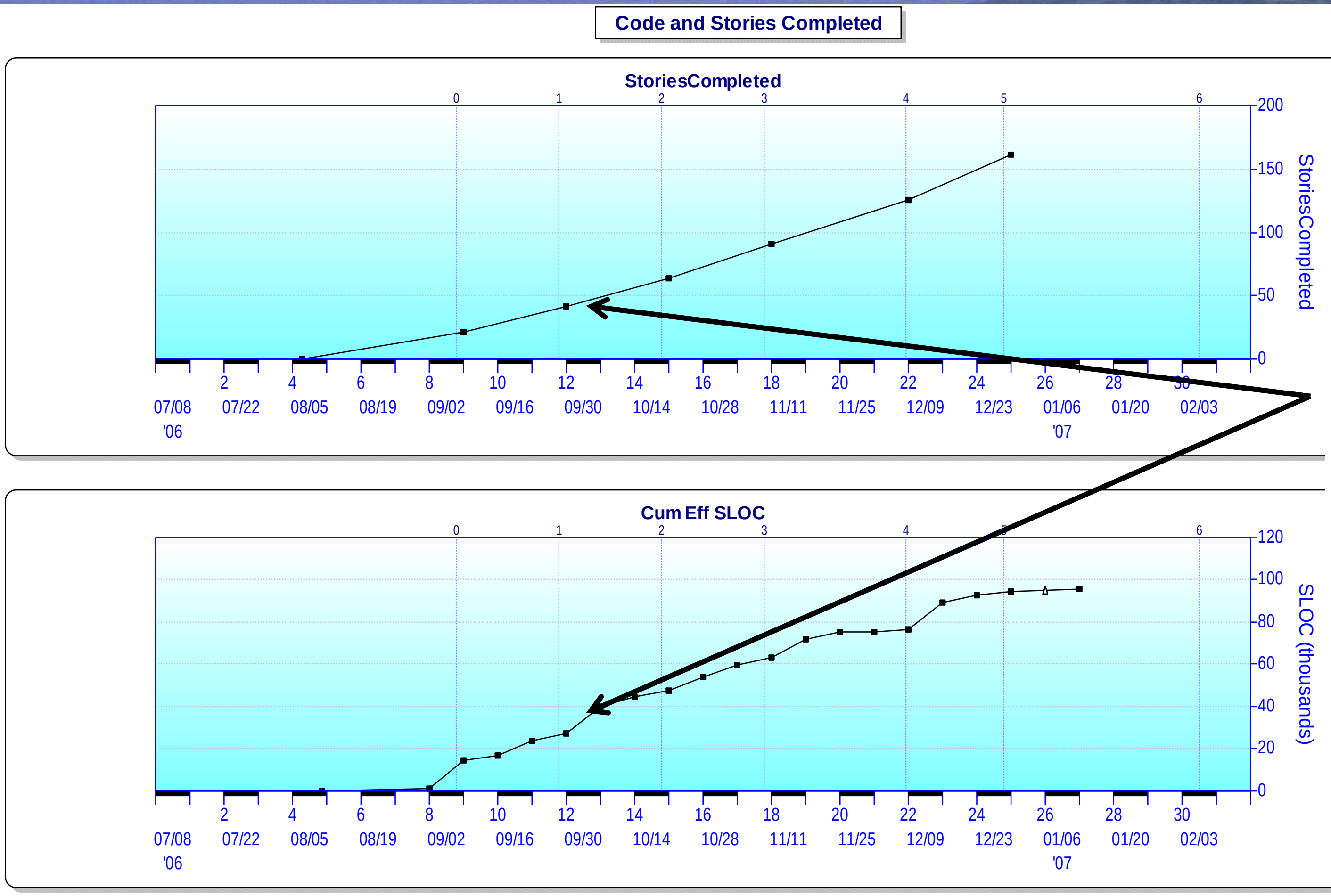
Estimating Size of Agile Projects

- 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

Stories and Code



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

Data from actual project monitored by QSM consultants

Code & Story Data by Iteration

Iteration	Code Per release	Stories Completed	Story Gearing
Iteration 0	14295	21	680.7
Iteration 1	13165	22	598.4
Iteration 2	20130	22	915.0
Iteration 3	15794	27	585.0
Iteration 4	13348	36	370.8
Iteration 5	17940	35	512.6
Average	15779	27	610.4

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

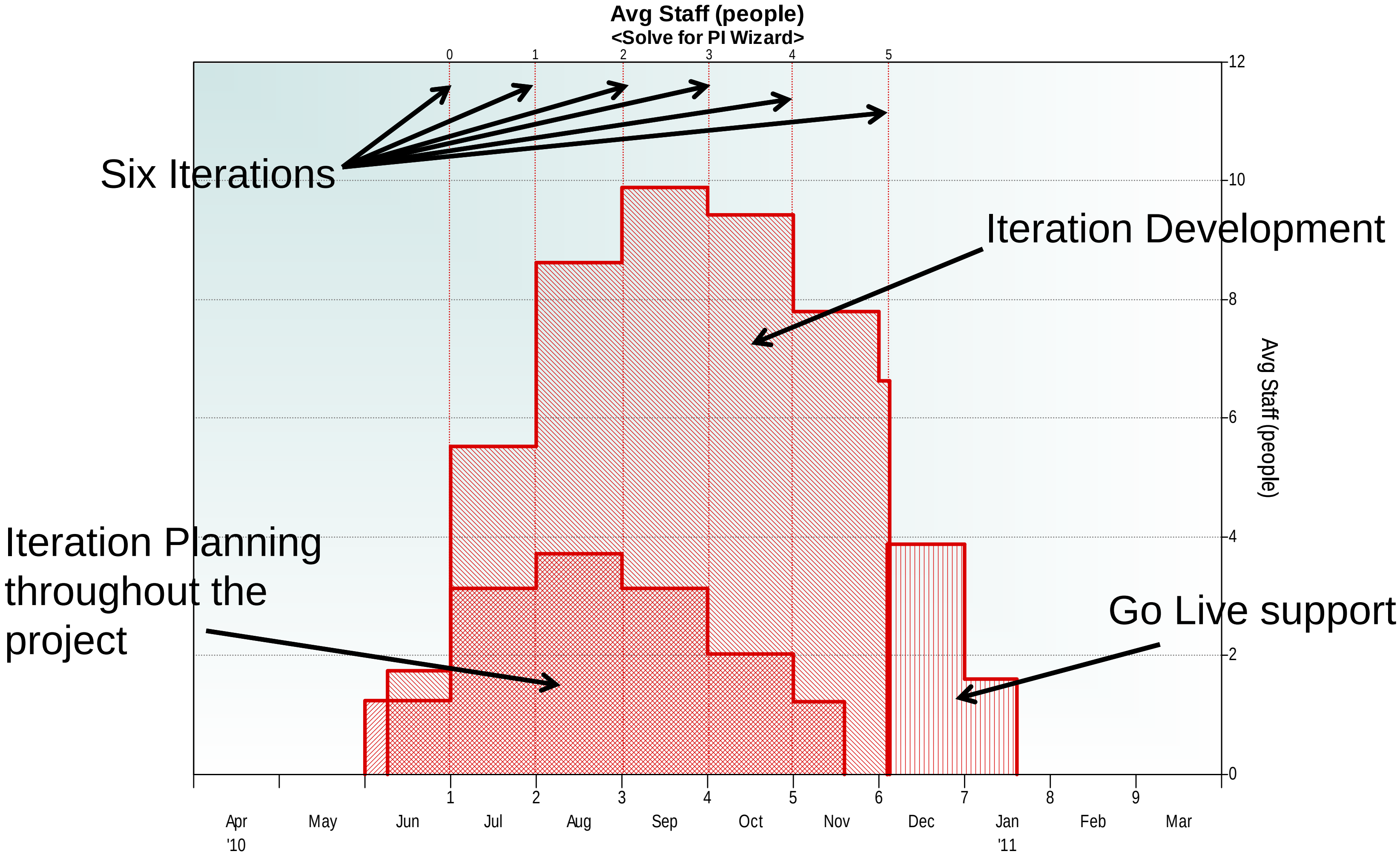
Low complexity	370
Average complexity	610
High complexity	915

This process can be used to determine appropriate gearing factors in different environments

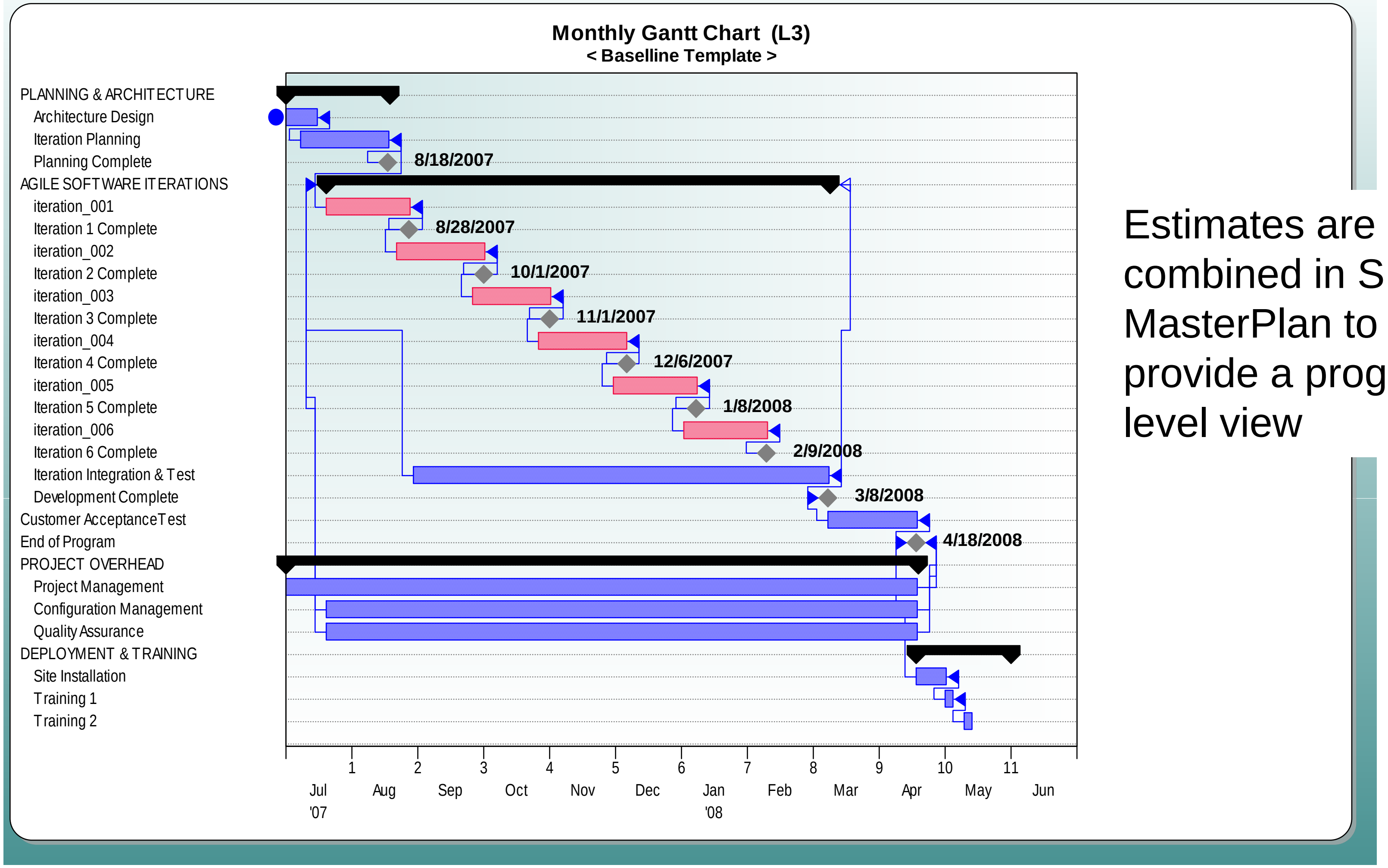
Estimating Agile Projects

- 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



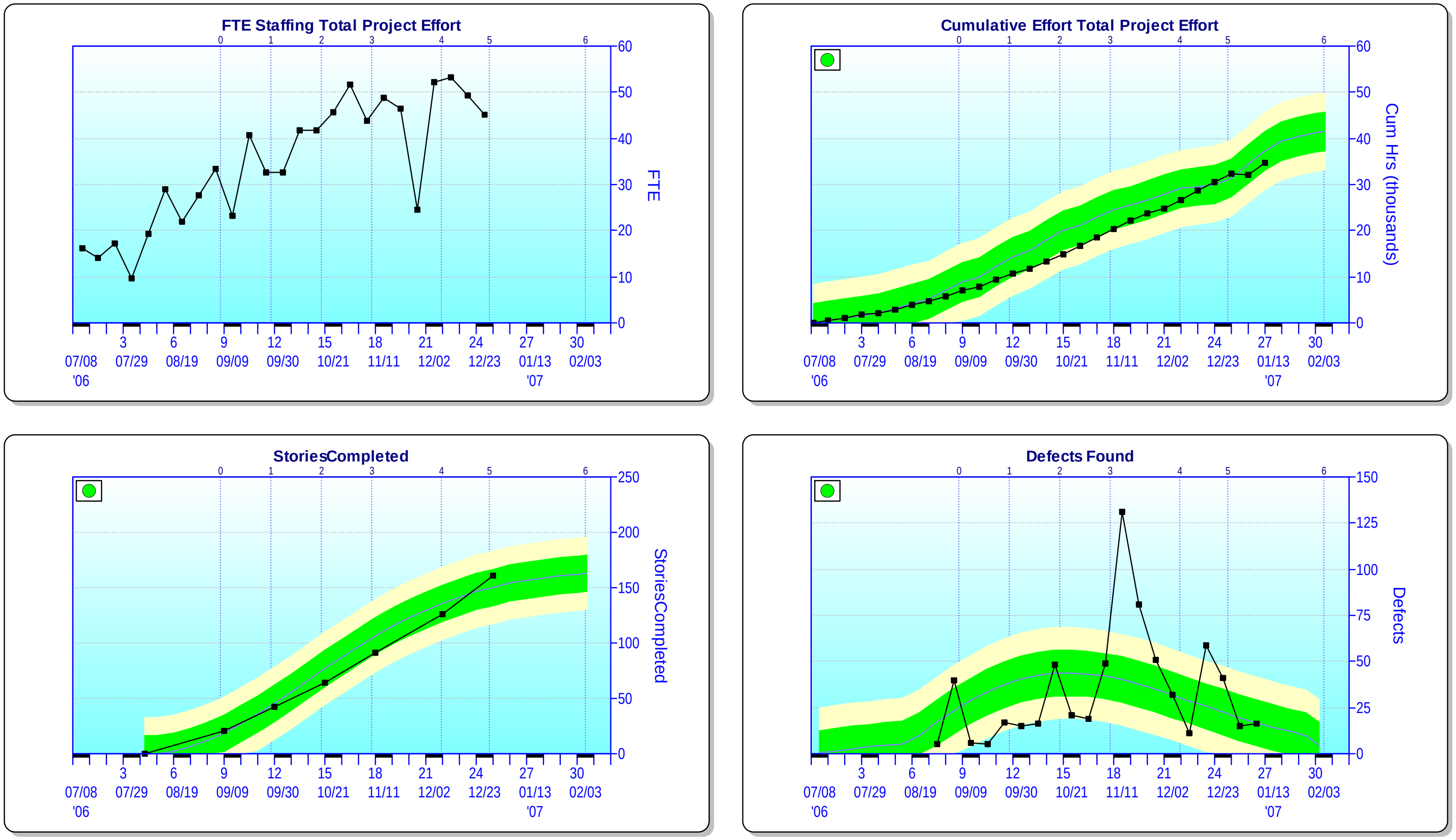
Estimating
One Estimate per Iteration



Estimates are combined in SLIM MasterPlan to provide a program level view

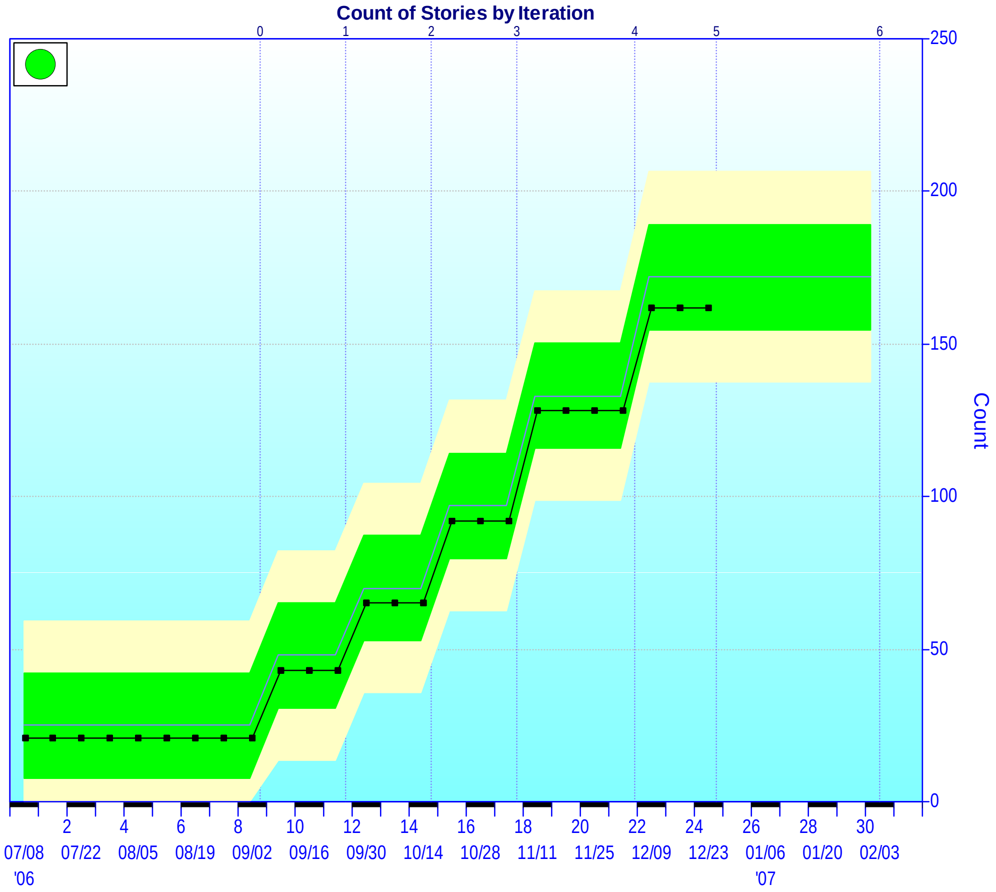
Tracking Agile Projects

Core Metrics View



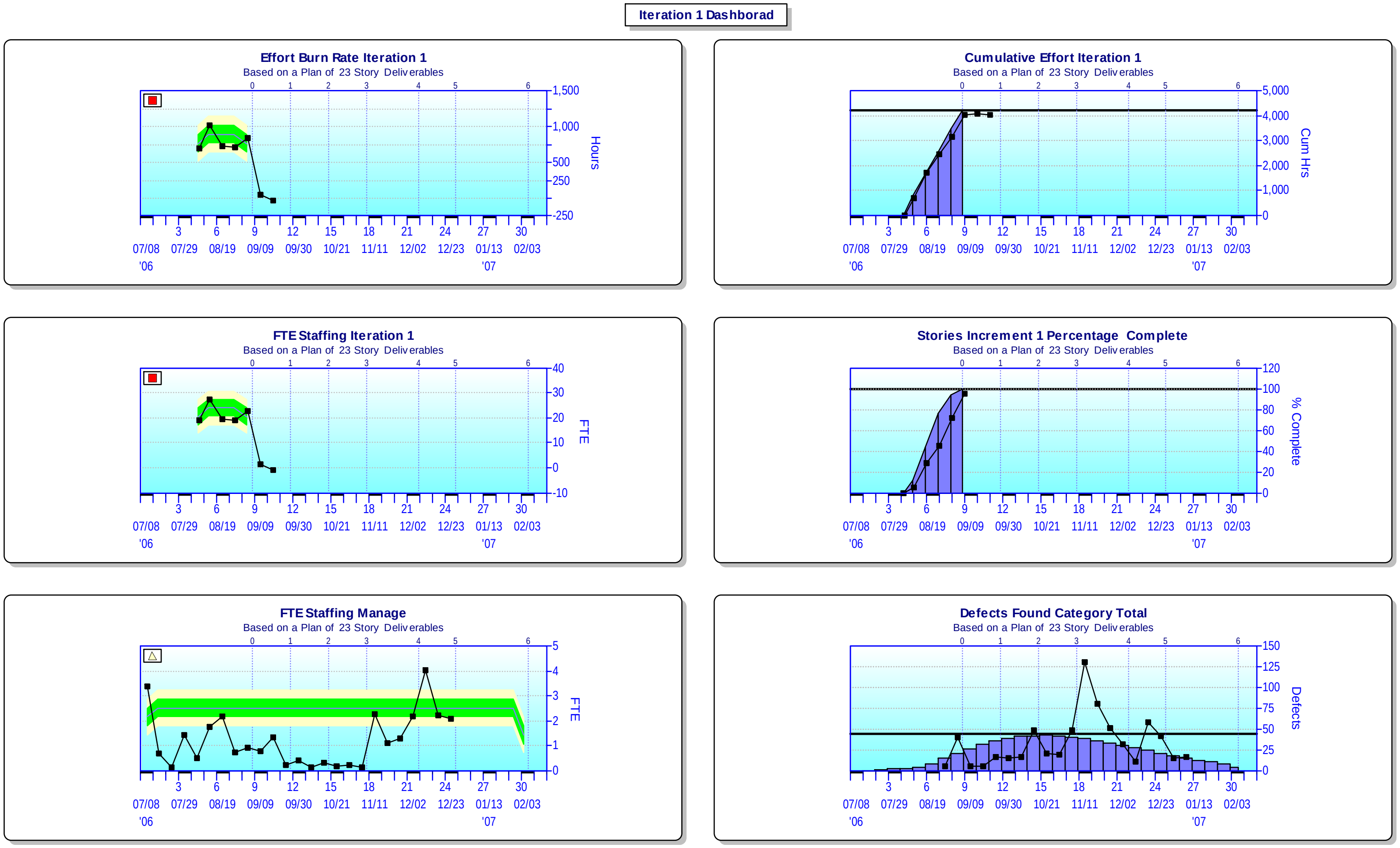
Overall view of effort, defects, and stories completed

Tracking Agile Projects
Stories



This slide shows the stories completed by iteration

Tracking Agile Projects



Here are tracking metrics for a single iteration within the project

QUESTIONS?