Quality Cost Tradeoff Model Calibrated with TSP Data

William Nichols

Software Engineering Institute

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-1277

Quality Cost Tradeoff Model Calibrated with TSP Data

The Problem

Managing Software Project Cost, Schedule, Scope, and Quality Outcomes

TSP Defect Model

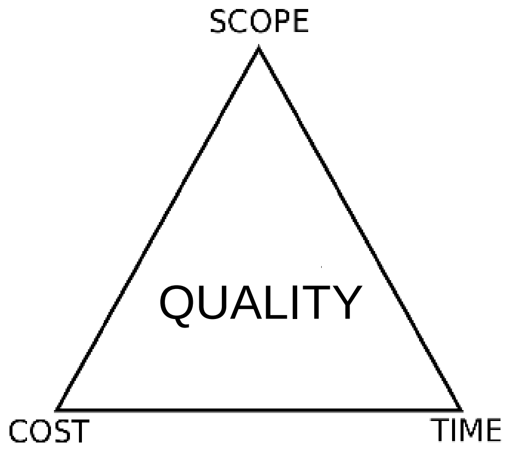
Is similar to COQUALMO in some key ways, but also has differences

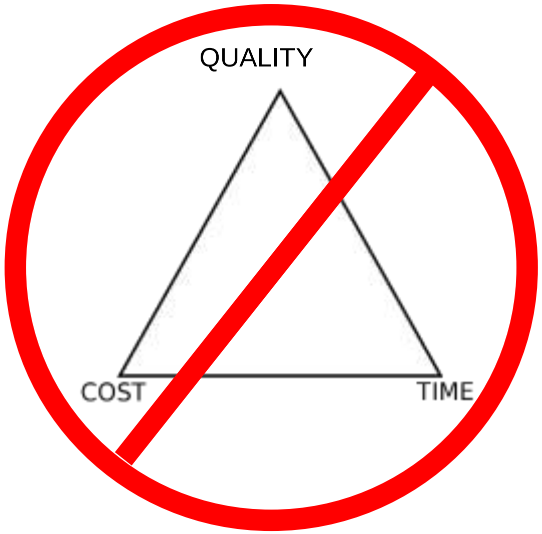
1) Defect fix effort is explicitly quantified,
2) We calibrate with local TSP measurement (using TSP parameters)

As with COQUALMO, the assumption is that there is an underlying causal mechanism in software development. This leads us to the SCOPE project.

The Iron Triangle, (Quality and Beyond)
Fix two factors to determine the third

The Role of quality is misunderstood





The Iron Triangle represents the relationship among three interrelated variables, cost, schedule, and scope.

Quality is not included because it is not a simple trade-off. But,

- How does quality affect cost, schedule, and scope? (COQUALMO?)
- What factors determine quality? (SCOPE)

Defect Reduction Top 10 List

Boehm and Basili, IEEE Computer 2001

Expert option suggests mechanisms, some of which are *causal.*

Finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase. $( ) *(find defects sooner may save money)*

About 80 percent of avoidable rework comes from 20 percent of the defects. -> $ *(the vital few, but which ones are they?)*

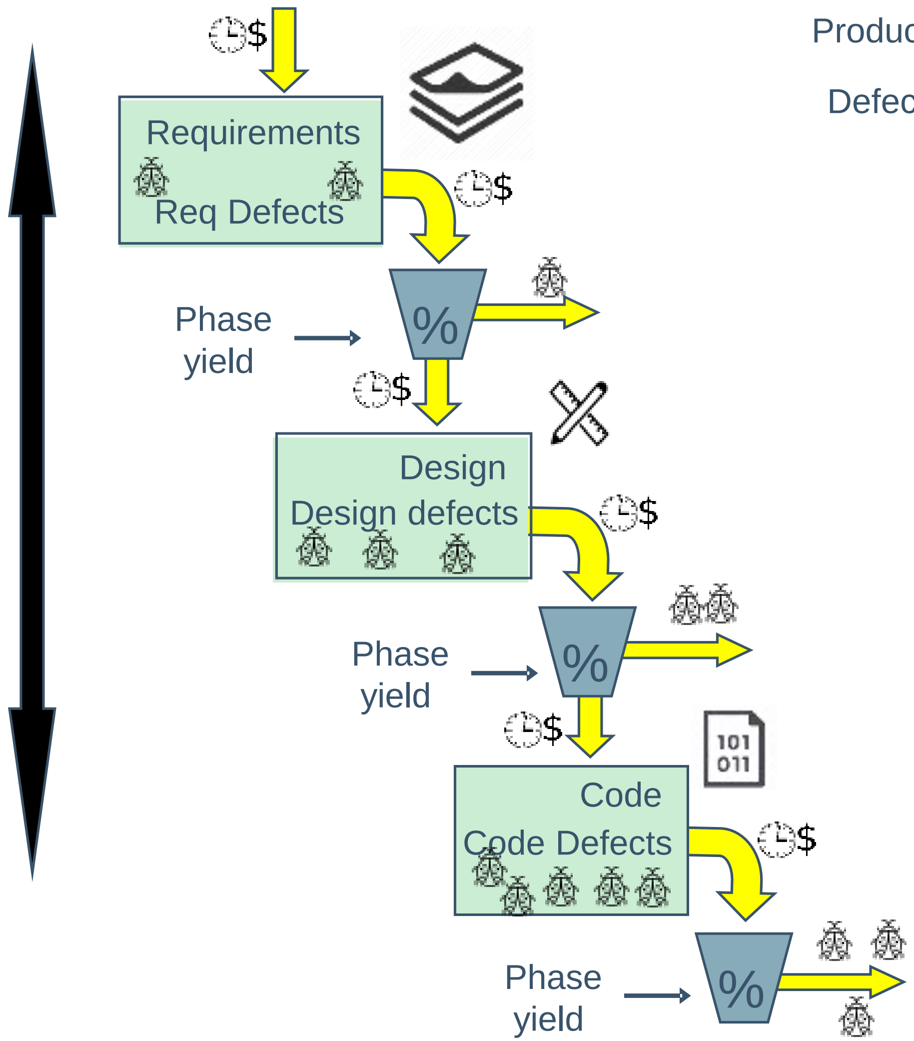
Peer reviews catch 60 percent of the defects. *(do we do enough?)*

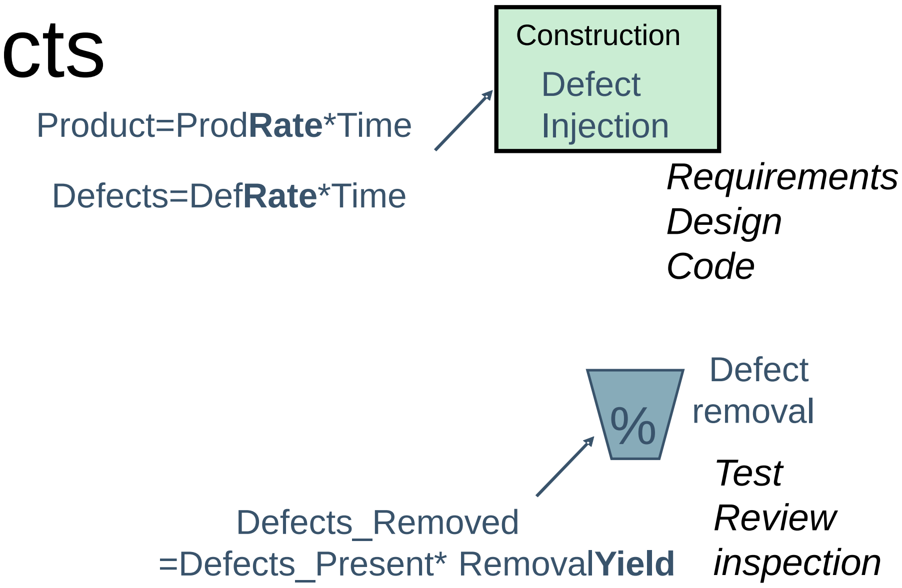
Disciplined personal practices can reduce defect introduction rates by up to 75 percent. *(prevention pays, but what does it cost?)*

< <$

*All other things being equal, it costs 50 percent more per source instruction to develop high-dependability software products than to develop low-dependability software products. **However, the investment is more than worth it if the project involves significant operations and maintenance costs.*** *(where is the payoff tipping point?)*

A Mechanistic Model relates Effort, Product, and Defects





Similar to Jones "Tank and Filter",

Simplifies assumptions found in Boehm/Chulani COQUALMO

Integrates defect and cost projections

Nichols, W. R. Plan for Success , Model the Cost of Quality. (2012). *Software Quality Professional*, *14*(2), 4–11.

Quality Cost Tradeoff Model Calibrated with TSP Data

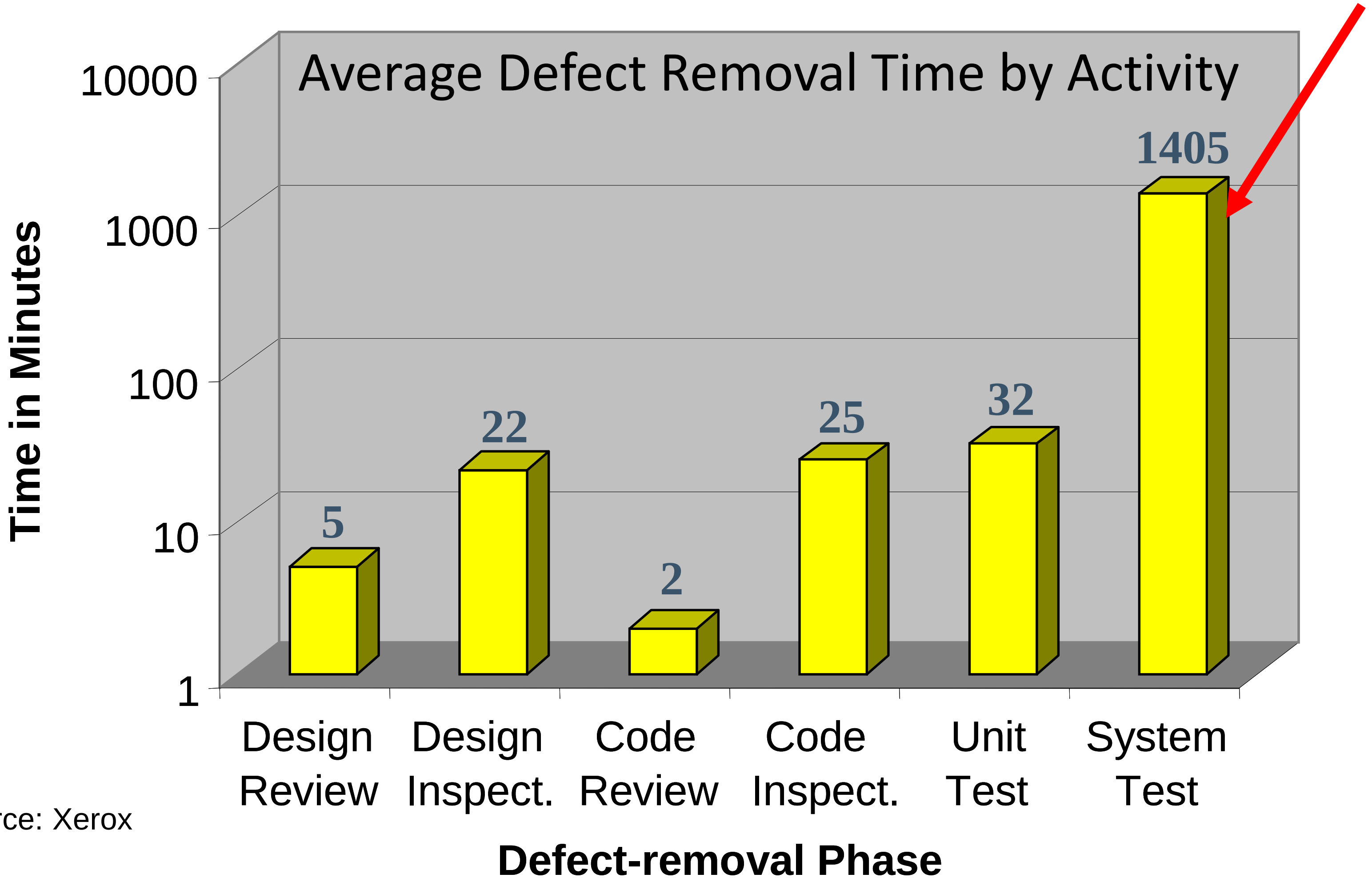
Calibrate the Model Locally

With TSP Data

Close the "Cost Loop" with Rework Effort

Model how defects get into the system,
And how they are removed

This is what we want to avoid.
What are the control points?

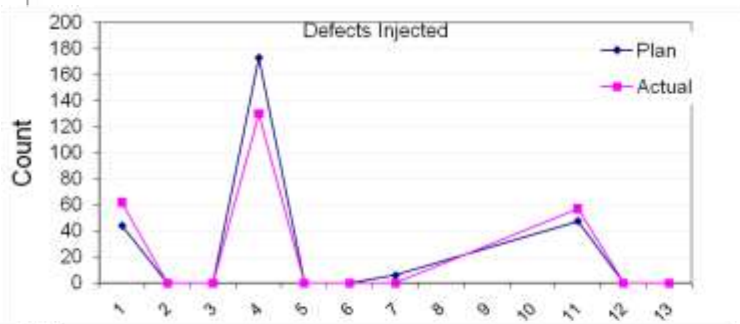


Source: Xerox

Collect Project Level Data for Effort and Defects

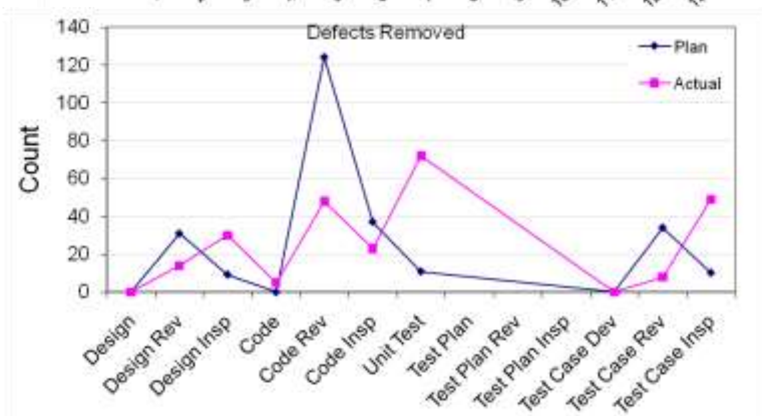
Plan/Actual

Phase Effort



Defects Injected

is predictable

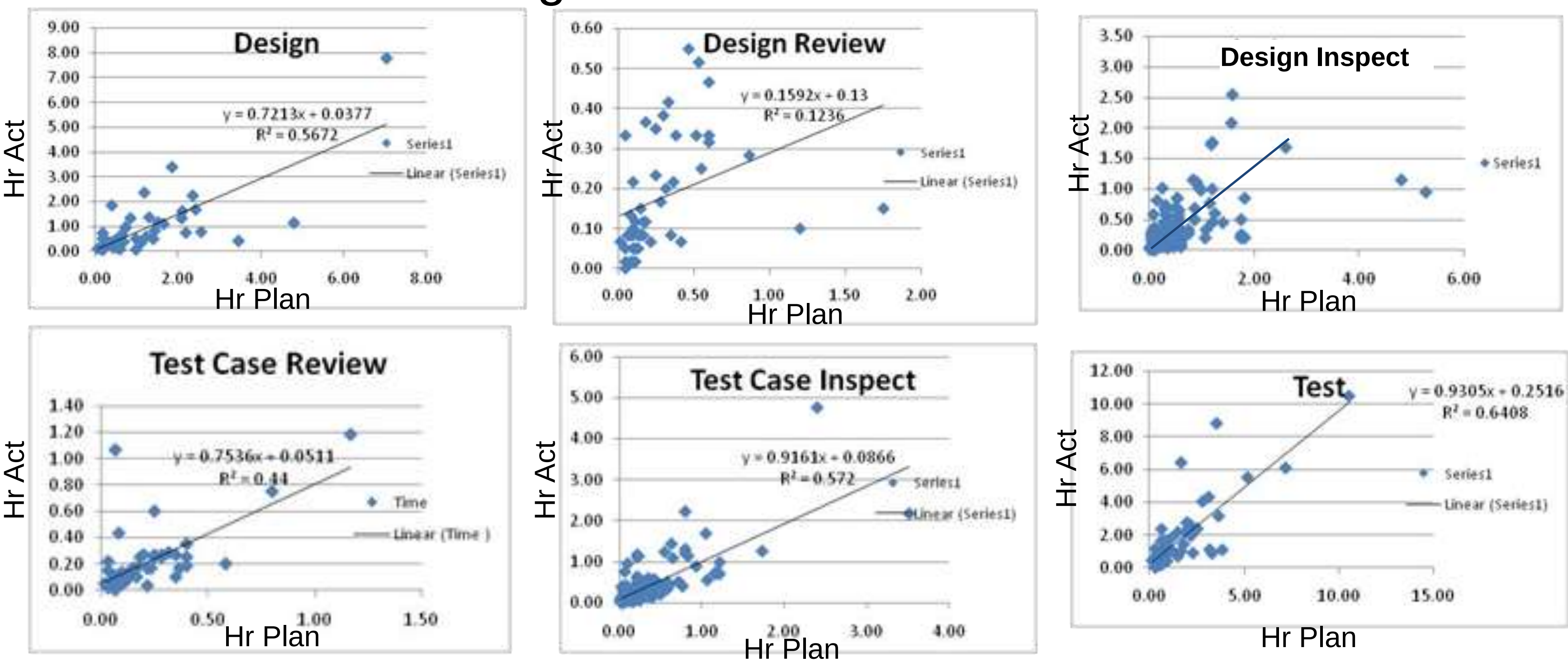


Defects Removed

is optional and variable

Calibrate with local data

Below are scatterplots of predicted and actual hours of effort found in individual components for a single project.
Similar data is available for defects injected and removed.
The data is variable, but predictable.
Repeatable is one indication of an underlying causal structure,
We will take advantage of this in SCOPE.



TSP Measures-> Model Parameters

Where does the data come from?

>900 projects planning cycles of data for benchmarking

(Bring your own data!) Projects bootstrap by collecting local data to update the models.

Defect injection rates and removal yields	Code Rate [LOC/Hr]
Defect density (defects and size)	Phase Removal Yield [% removed]
Review rates	Zero Defect Test time [Hr]
Phase Injection Rate [defects/Hr]	Phase “Find and Fix” time [Hr/defect]
Phase Effort Distribution [%] total time	Review/Inspection Rate
Size [LOC]	

Quality Cost Tradeoff Model Calibrated with TSP Data

Model Demonstration

Using the Model

Using the Model

Will you achieve your goals?

- How much do you want to deliver?
- What is your desired schedule?
- How many defects do you expect the user to find?

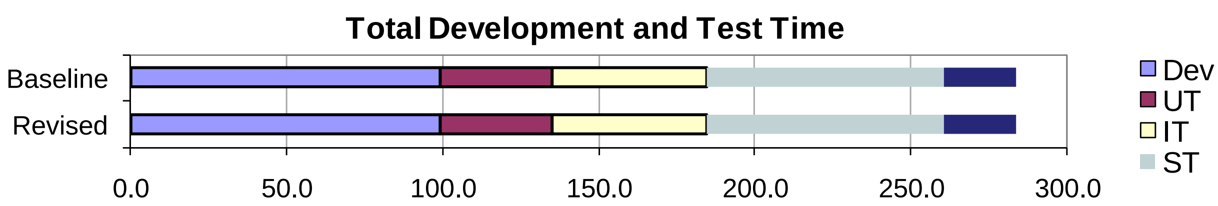
What do you know? Do you have relevant historical data?

What decisions can you make?

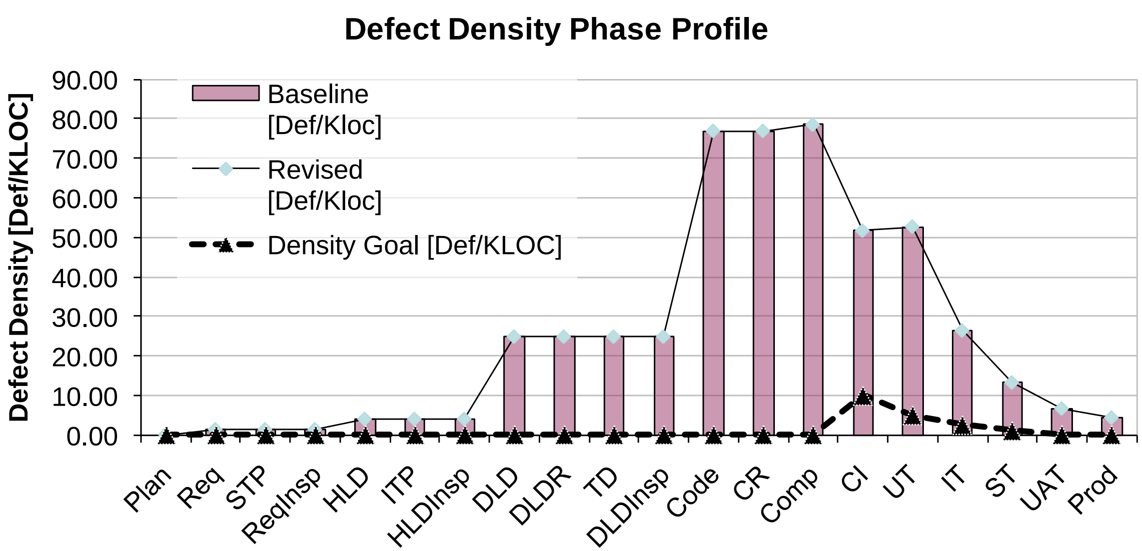
- Do you have enough staff?
- Do the staff have the right skills?
- How will you allocate the effort?

Control Panel	Rate [LOC/hr]	Yield (per insp)	# Insp
Design Review	200	50.0%	0
Design Inspection	200	50.0%	0
Code Review	200	50.0%	0
Code Inspection	200	50.0%	0

Baseline Adjustable Parameters



Baseline Outcomes

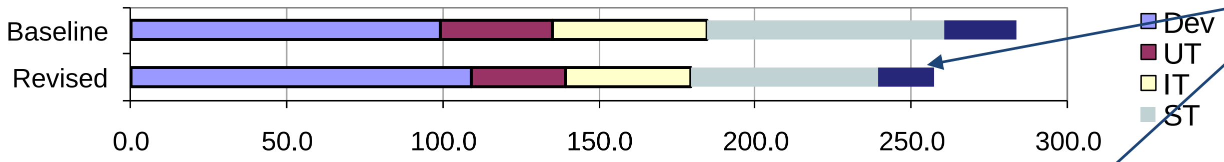


Compare our performance to a baseline

Control Panel	Rate [LOC/hr]	Yield (per insp)		# Insp
Design Review	200	50.0%		1
Design Inspection	200	50.0%		0
Code Review	200	50.0%		0
Code Inspection	200	50.0%		0

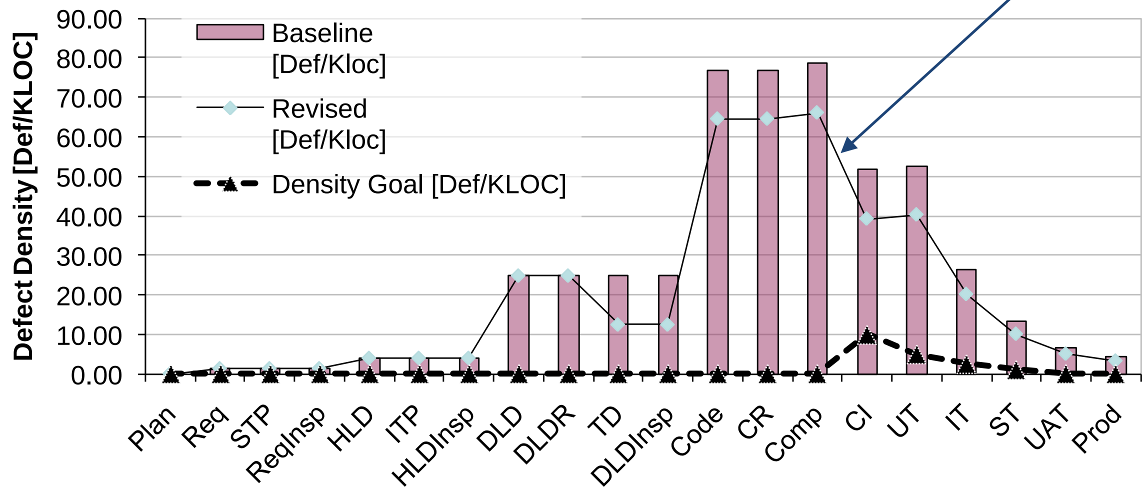
Trial Parameter

Total Development and Test Time



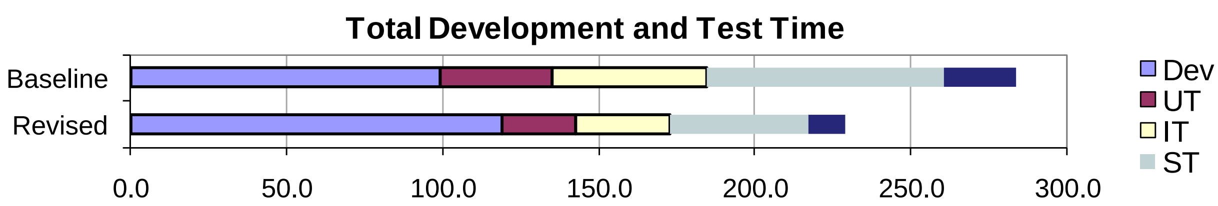
Trial Outcomes

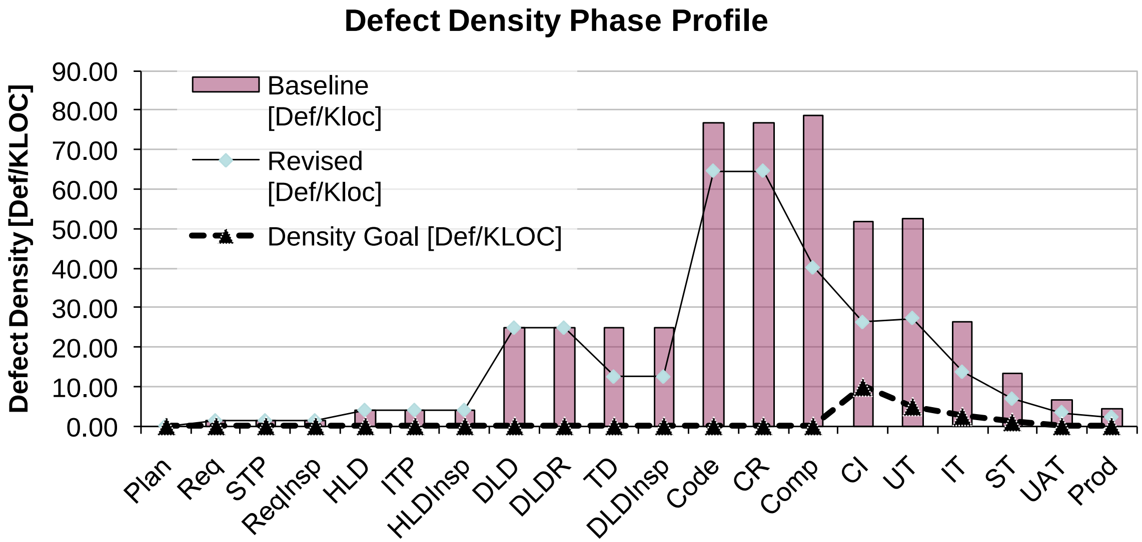
Defect Density Phase Profile



Perform a personal design review

Control Panel	Rate [LOC/hr]	Yield (per insp)	# Insp
Design Review	200	50.0%	1
Design Inspection	200	50.0%	0
Code Review	200	50.0%	1
Code Inspection	200	50.0%	0



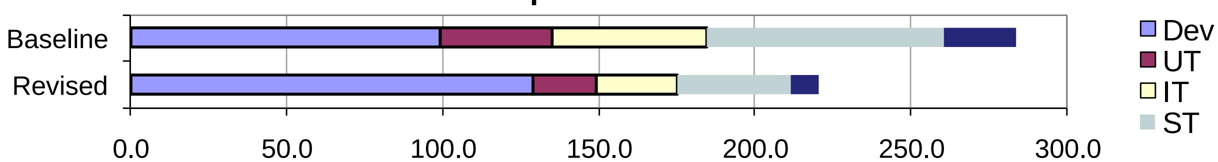


Include a peer design review

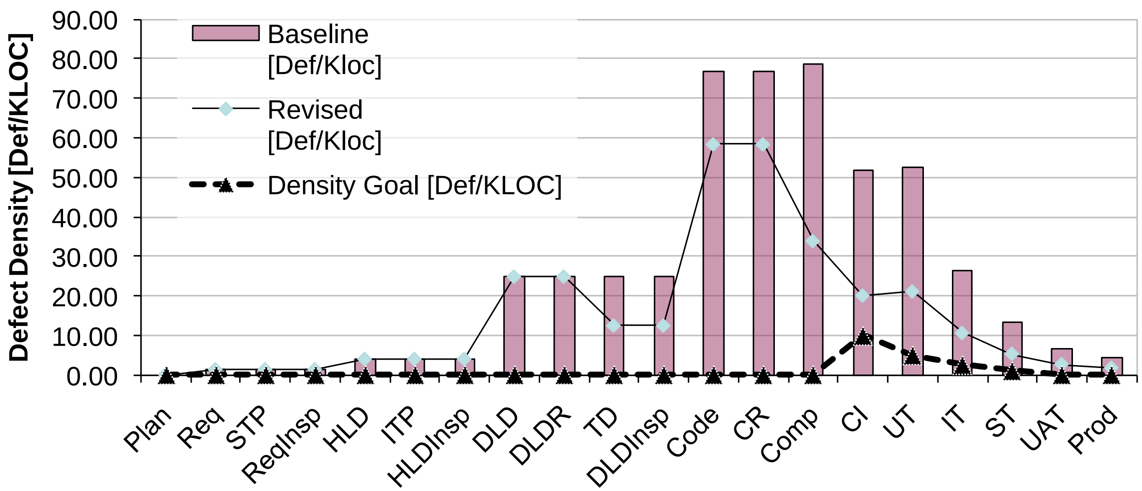
Control Panel

	Rate [LOC/hr]	Yield (per insp)		# Insp
Design Review	200	50.0%		1
Design Inspection	200	50.0%		1
Code Review	200	50.0%		1
Code Inspection	200	50.0%		0

Total Development and Test Time

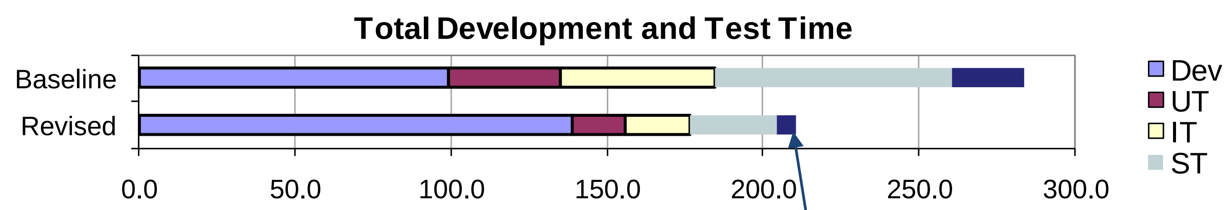


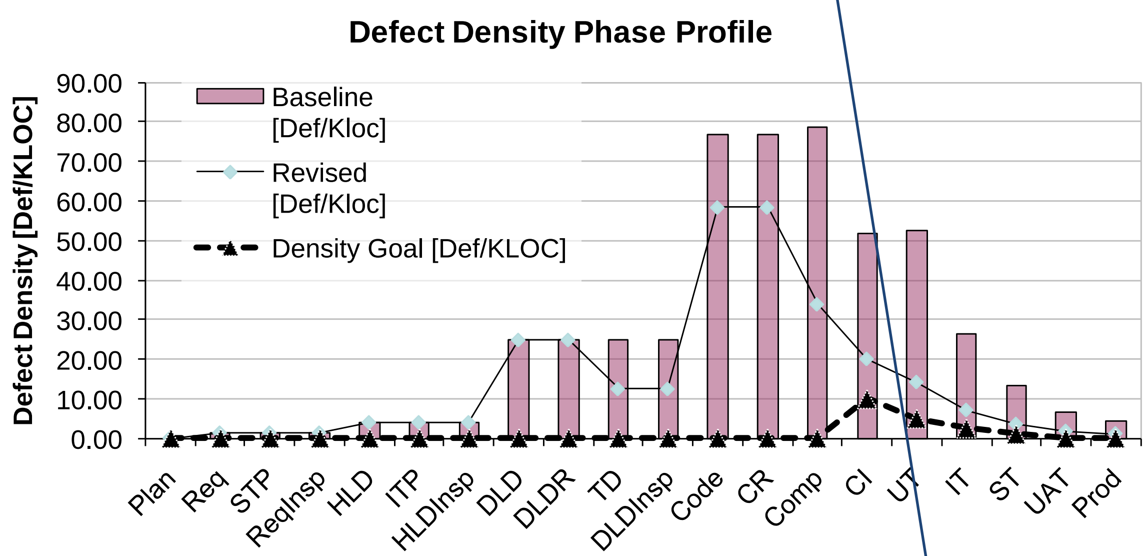
Defect Density Phase Profile



Have a peer inspect the code

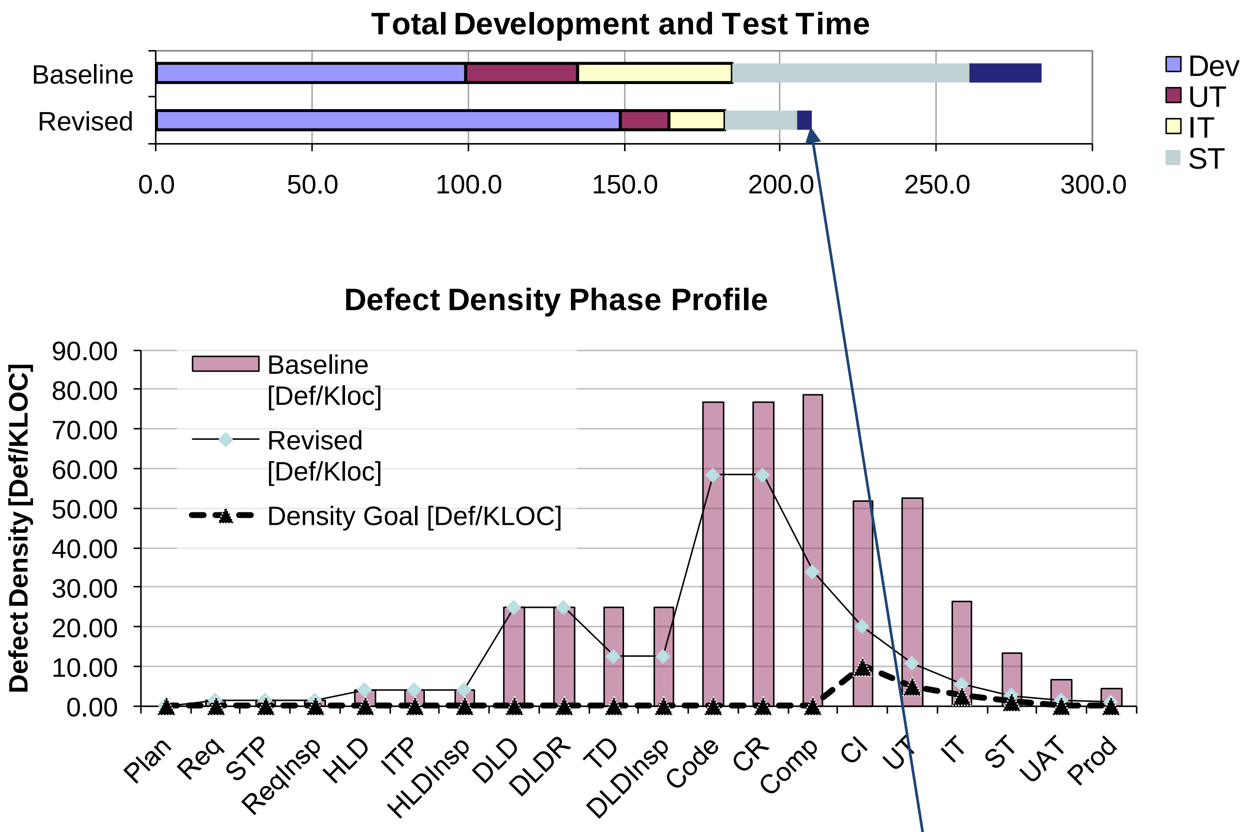
Control Panel	Rate [LOC/hr]	Yield (per insp)	# Insp
Design Review	200	50.0%	1
Design Inspection	200	50.0%	1
Code Review	200	50.0%	1
Code Inspection	200	50.0%	1





At some point we cross the “quality is free” Boundry!

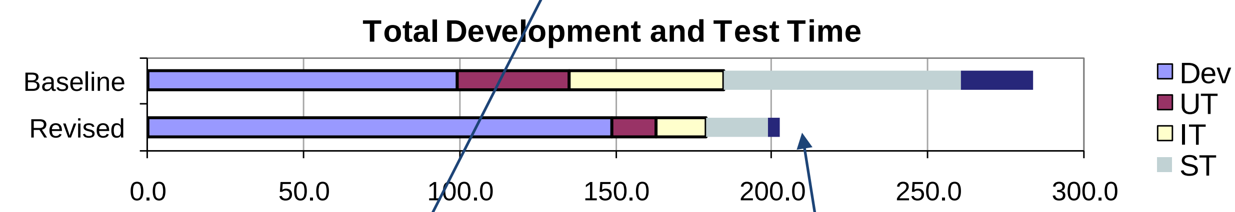
Control Panel	Rate [LOC/hr]	Yield (per insp)		# Insp
Design Review	200	50.0%		1
Design Inspection	200	50.0%		1
Code Review	200	50.0%		1
Code Inspection	200	50.0%		2

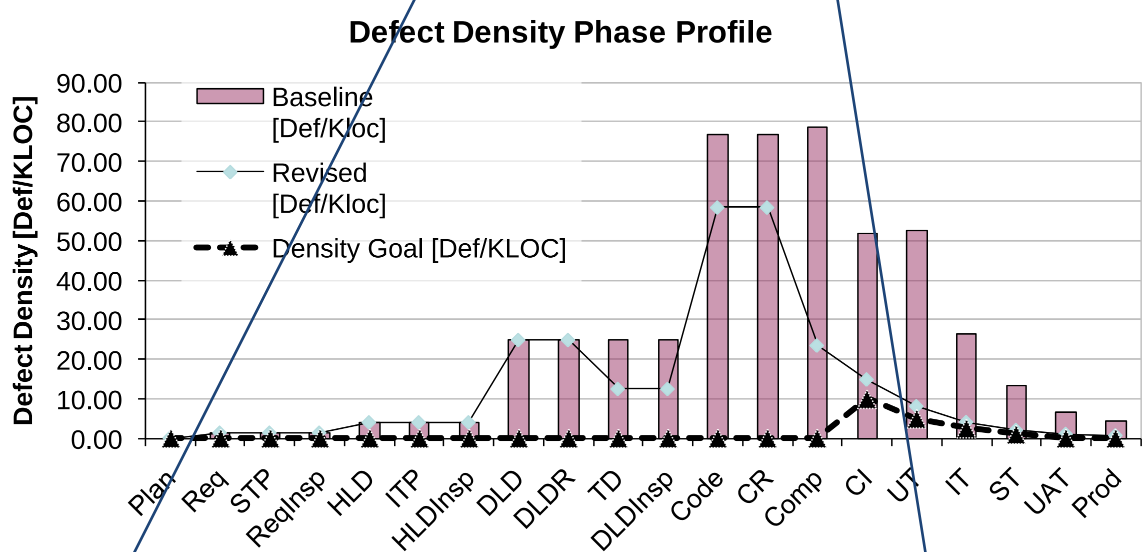


Crossing the “quality is free” point!

Control Panel

	Rate [LOC/hr]	Yield (per insp)	# Insp
Design Review	200	50.0%	1
Design Inspection	200	50.0%	1
Code Review	200	70.0%	1
Code Inspection	200	70.0%	2





Local parameters move the “quality is free” Boundry

Quality Cost Tradeoff Model Calibrated with TSP Data

Conclusion

Implications and Future Work

Implications

Defect reduction improves project performance. (quality is free?)

By modeling the *causal* structural mechanisms

- Test the sensitivity and limits of *controllable factors*
- Examine "what-if" scenarios before work

What other factors for control should be included in cost models?

- Inspection effectiveness varies widely but is a teachable Rombach, D., Münch, J., Ocampo, A., Humphrey, W. S., & Burton, D. Teaching disciplined software development. (2008). *Journal of Systems and Software*, *81*(5), 747–763. https://doi.org/10.1016/j.jss.2007.06.004
- Training to affect developer performance ranges
- Design techniques, effort, documentation, quality
- Programming language or technology stack
- Effort allocation (*balance effectiveness with resource expenditure*)

Only matter if the relationship is causal.

SCOPE studies the causal nature and structural relationship of factors

Call for Data

SCOPE studies Causal Analysis with observational data from real projects.

If you have data you can share, or

If you want to learn more about Causal Analysis and structural modeling

Contact

Bill Nichols wrn@sei.cmu.edu

Bob Stoddard rws@sei.cmu.edu

Mike Konrad mdk@sei.cmu.edu

Quality Cost Tradeoff Model Calibrated with TSP Data

Section Title

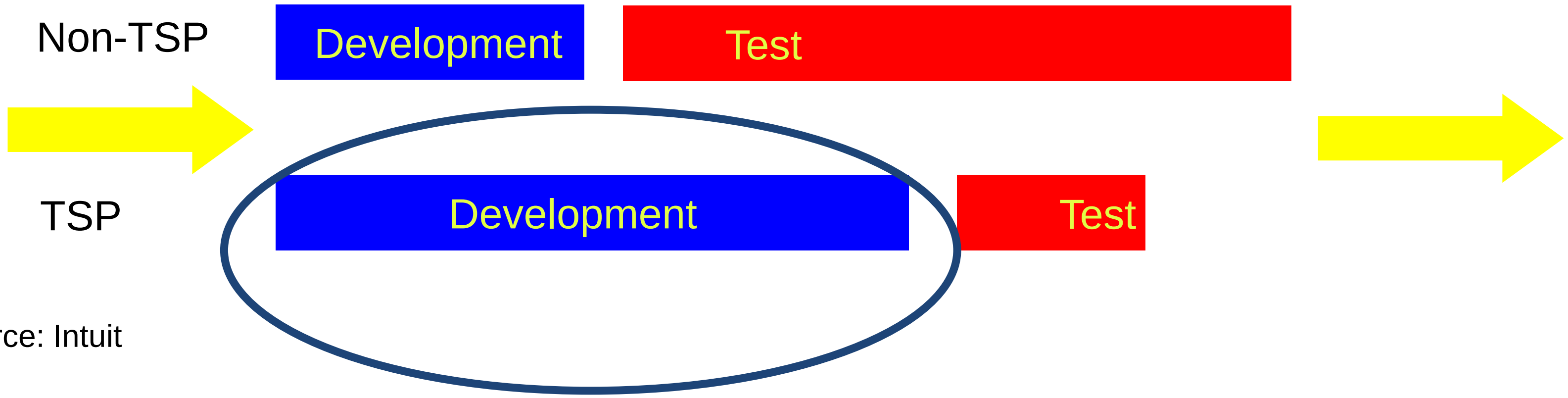
Backup

Intuit Productivity Improvement

Faster, better, cheaper (reduced rework)

From data on over 40 TSP teams, Intuit has found that

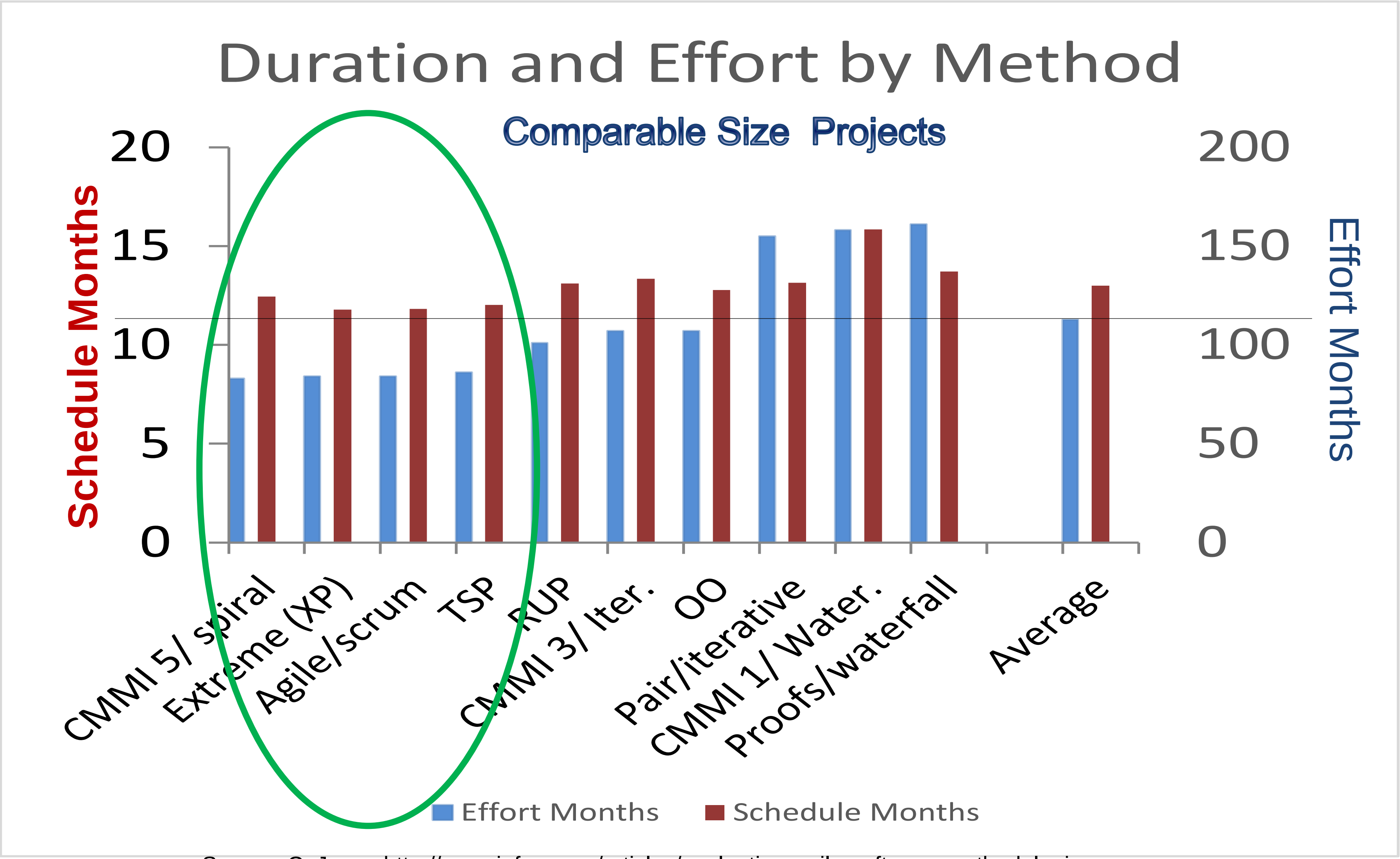
- post code-complete effort is 8% instead of 33% of the project
- Standard test times were cut from 4 months to 1 month or less.



Source: Intuit

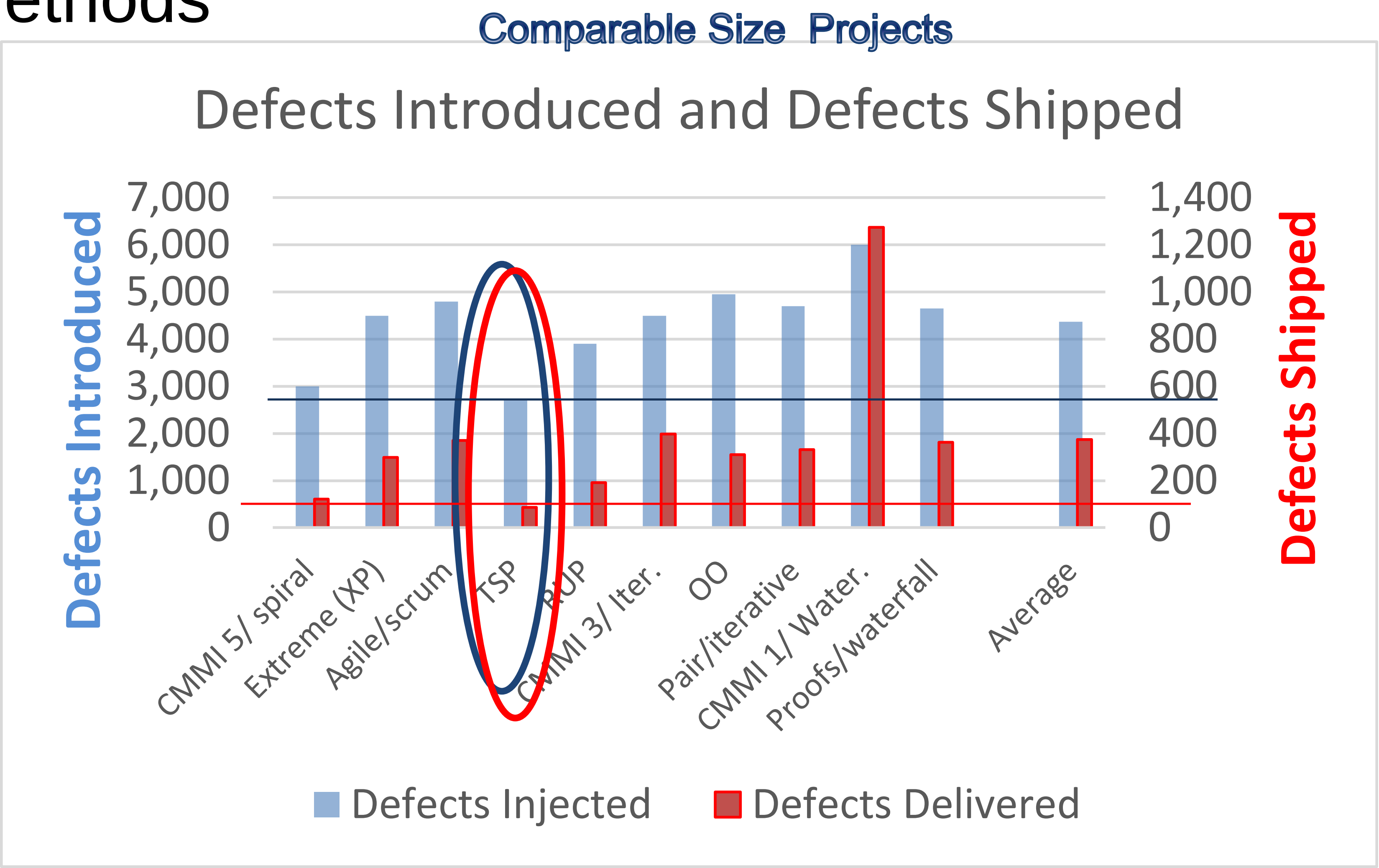
Organizations using TSP report productivity gains of 30% or more resulting in lower costs or more functionality in delivered software.

, Duration/Effort to "project completion"



Source: C. Jones http://www.infoq.com/articles/evaluating-agile-software-methodologies

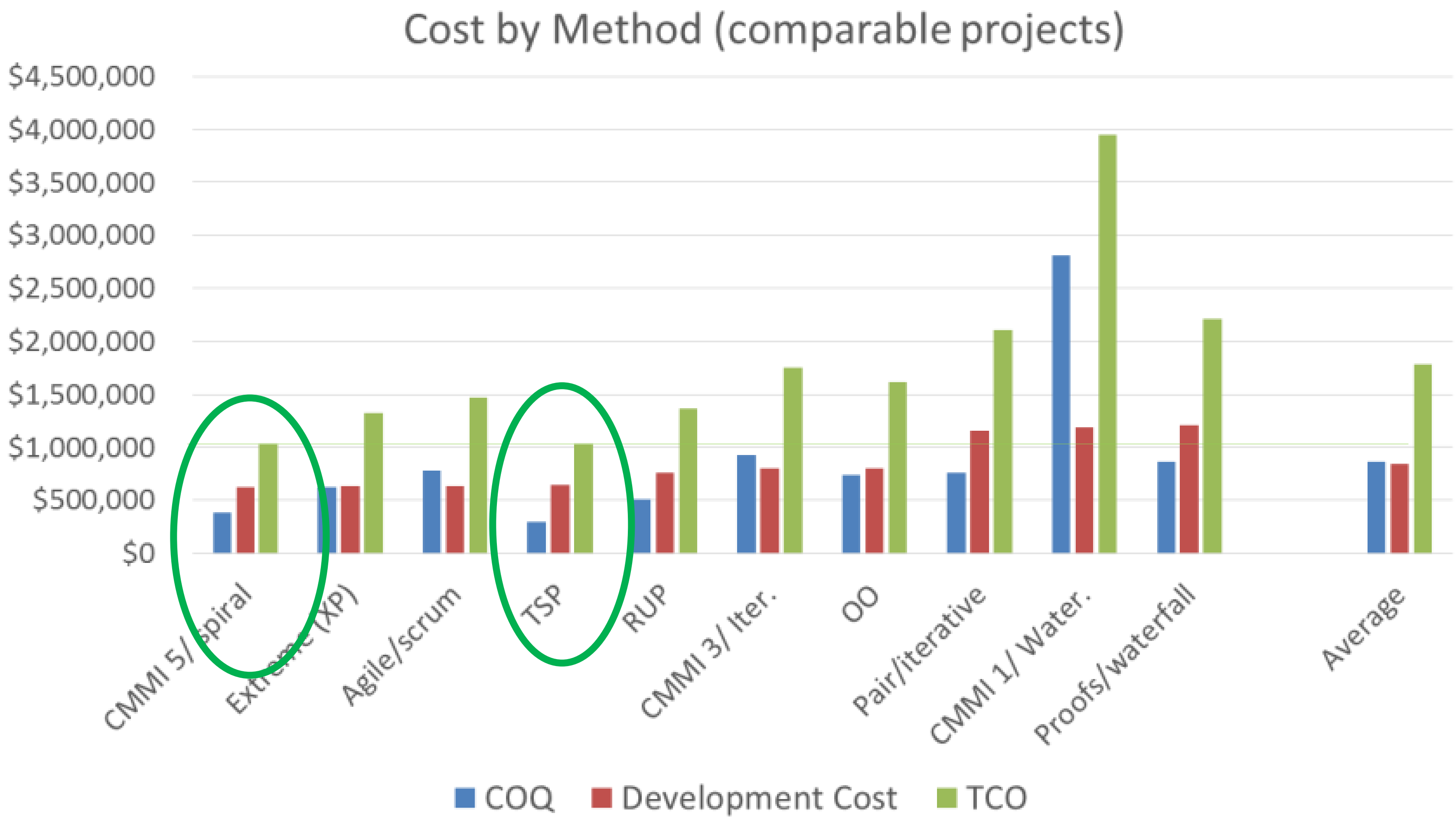
Defect Levels for various development methods



Source: C. Jones http://www.infoq.com/articles/evaluating-agile-software-methodologies

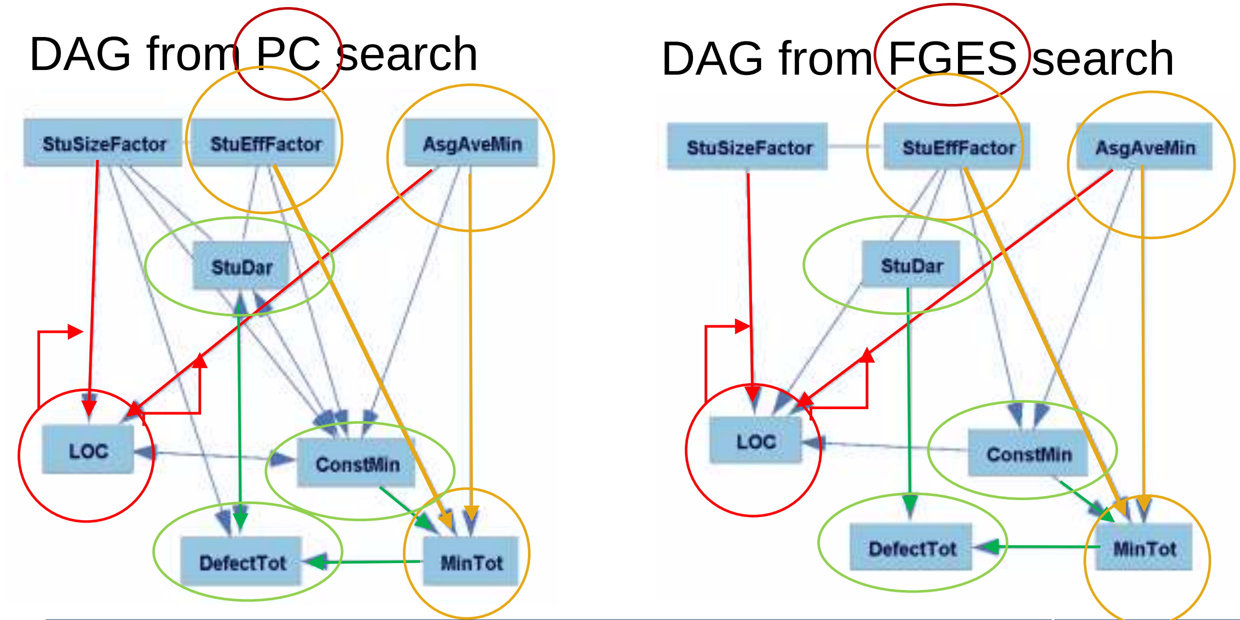
Total Cost of Ownership

Comparable Size Projects



Source: C. Jones http://www.infoq.com/articles/evaluating-agile-software-methodologies

Application of Causal Analysis of PSP



Common Direct Causal Edges

1. AsgAveMin → ConstMin
2. StuEffFactor → ConstMin
3. AsgAveMin → MinTot
4. StuEffFactor → MinTot
5. ConstMin → MinTot
6. MinTot → DefectTot
7. AsgAveMin → LOC
8. StuSizeFactor → LOC

Expected Relation (log transformed for linear effects)	Edges found	PC	FGEs
$\ln(LOC_{ij}) = \ln(ReqSize_i) + \ln(SSF_j)$	StuSizeFactor → LOC	Y	Y
	AsgAveMin → LOC	Y	Y
$\ln(MinTot_{ij}) = (ReqSize_i) + \ln(SEF_j)$	AsgAveMin → MinTot	Y	Y
	StuEffFactor → MinTot	Y	Y
$\ln(DefTot_{ij}) = \ln(ConstMin_{ij}) + \ln(StuDAR_j)$	StuDAR -> DefectTot Const -> DefTot	Bi I	Y I