QSM Research: Understanding the Physics of Software Projects

Topics to include:

How to Develop a Data-driven Estimation Model
Small Teams Deliver Lower Cost & Higher quality
History is the key to Estimation Success

Presentation Agenda

- About QSM
- How it all began
  - Review Larry Putnam, Sr.'s original research & data
- Does the initial research stand the test of time?
  - Follow up with new data
  - Regularly published research almanacs
- Does current agile data exhibit similar behavior?
- Summary

Quick Facts

- Industry Leadership
  - Established in 1978 by Larry Putnam, Sr., a pioneer, and renowned thought leader in the software estimation field
  - Headquartered in McLean, VA, with affiliate offices in Europe and across the U.S.
- Industry Leading Tools and World's Largest Database
  - 13,000+ completed projects providing deep insights into fundamental relationships at play in software and systems development
  - Industry- leading SLIM® modeling tools enabled by proven parametric algorithms
- 40 Years of Proven Field Success
  - Experience working with thousands of clients in every industry and sector
  - Measurable impact on hundreds of consulting engagements
  - Experts at helping customers build their own internal estimation capacity
  - $30M+ in product R&D investment guided by client needs and input over the years

QSM®

Success is Not an Option - Or is It?

- A Simple, But Clear Mission - To Help Clients Succeed:
  - By empowering them to negotiate well-informed decisions and execute successful software development and deployment
  - Through measuring and increasing efficiencies in their operations
  - Optimizing productivity and quality across their portfolio of projects
- We Do That By:
  - Offering state-of-the-art tools and training coupled with comprehensive consulting services and outstanding customer support
  - Turning data into defensible and actionable management information
  - Identifying project constraints; understanding organizational objectives; applying metrics, assumptions and proven algorithms to accurately identify realistic goals and alternatives
  - Providing transparency to all stakeholders to enable fact-based decisions
  - Staying on the leading edge through relevant and timely research in the estimation industry

Our Founder – Lawrence H. Putnam, Sr.

- Started out as a career army officer
  - Trained as a nuclear affects engineer
  - Took Fortran programming to do nuclear calculations
  - Selected to run the Army's IT budgeting operation at the Pentagon (because of Fortran programming expertise)
  - Lost $10 million at first congressional budget hearing
  - Decided better solutions were needed to manage software projects
- Retired from Army and started QSM in 1978
  - Created the Software Lifecycle Model (SLIM)
  - Expanded into a suite of tools over time
- Retired from QSM in 2007
  - Still comes into the office every day

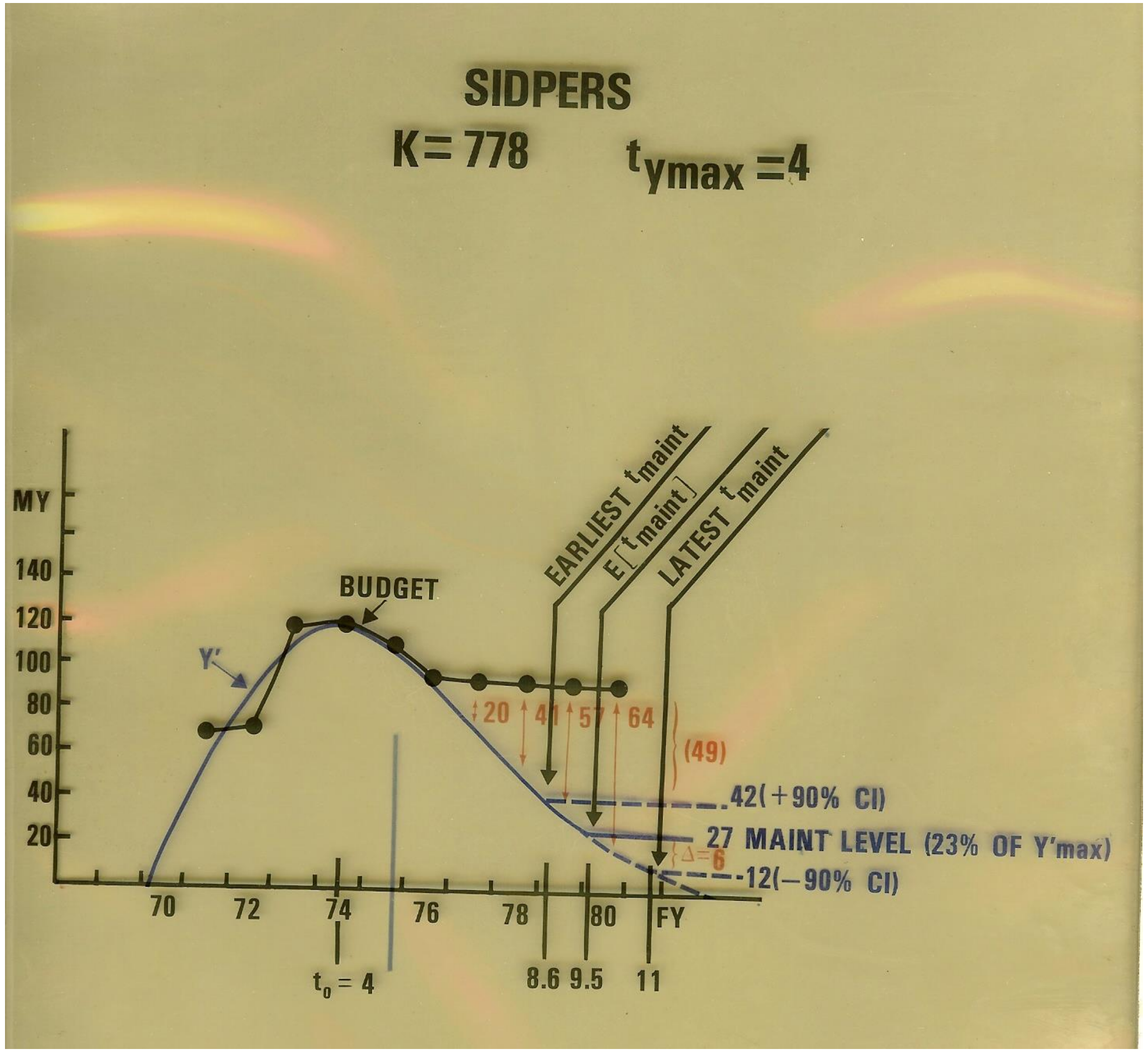
QSM®

Software Equation Background

- Time Frame 1975
- SIDPERS $10 Million Budget Loss
- Discovery of Norden-Rayleigh Equation
- Research to Discover the Rayleigh Parameters (Time and Effort)

Who had some data and what did it look like?

SIDPERS Budget Data



This is the project that lost Larry Sr. $10 Million out of the US Army's data processing budget.

Original Data Set Larry Had to Work With

USACSC SYSTEM CHARACTERISTICS

System	Life Cycle K (MY)	Size Development Time t_d (YRS)	Number of Files x_1	Number of Rpts. x_2	Number of Appl. Progs. x_3
MPMIS	73.6	2.28	94	45	52
MRM	84	1.48	36	44	31
ACS	33	1.67	11	74	39
SPBS	70	2.00	8	34	23
COMIS	27.5	1.44	14	41	35
AUDIT	10	2.00	11	5	5
CABS	7.74	1.95	22	14	12
MARDIS	91	2.50	6	20	27
MPAS	101	2.10	25	95	109
CARMOCS	153	2.64	13	109	229
SIDPERS	700	3.65	172	179	256
VTAADS	404	3.50	155	101	144
BASOPS-SUP.	591	2.73	81	192	223
SAILS AB/C	1028	4.27	540	215	365
SAILS AB/X	1193	3.48	670	200	398
STARCIPS	344	3.48	151	59	75
STANFINS	741	3.30	270	228	241
SAAS	118	2.12	131	152	120
COSCOM	214	4.25	33	101	130

The Original Army Data Set (19 Good projects)

1. Homogenous Data Set

   Same organization

   Same application type

   Same tooling

   Complete data (size-time-effort)

Man was I LUCKY !

Relationships Between Size, Time & Effort

There are relationships other than the difficulty that have significantly high correlation with measures of the product. These relationships are:

$$K/t_d^4 = f_1\ (x_1, x_2, x_3)$$

$$K/t_d^3 = f_2\ (x_1, x_2, x_3)$$

$$K/t_d^2 = f_3\ (x_1, x_2, x_3)$$

$$K/t_d = f_4\ (x_1, x_2, x_3)$$

$$K = f_5\ (x_1, x_2, x_3)$$

$$Kt_d = f_6\ (x_1, x_2, x_3)$$

$$K\ t_d^4 = f_7\ (x_1, x_2, x_3)$$

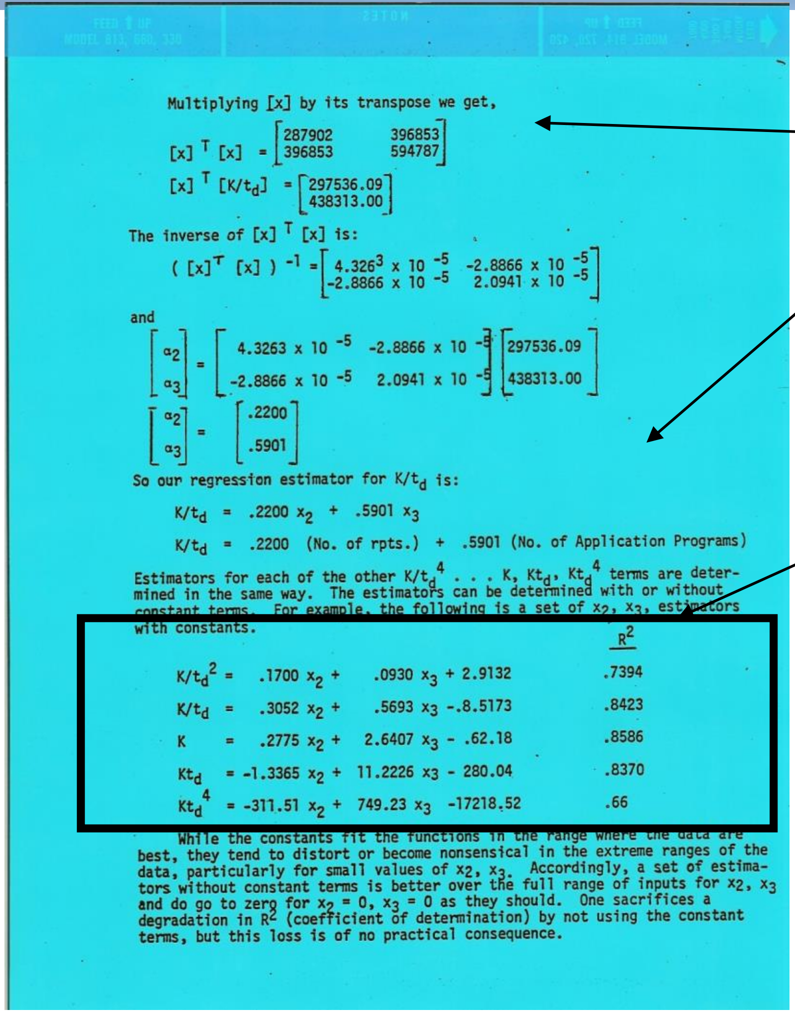
where K and t_d are the parameters of the Rayleigh equation.

(X1,X2,X3 are Size Metrics)

These are possible equations that would have to intersect (any 2) to provide the parameters for time and effort (K and td)

QSM®

Evaluating Early Software Equations



Multiplying [x] by its transpose we get,

$$[x]^T [x] = \begin{bmatrix} 287902 & 396853 \\ 396853 & 594787 \end{bmatrix}$$

$$[x]^T [K/t_d] = \begin{bmatrix} 297536.09 \\ 438313.00 \end{bmatrix}$$

The inverse of $[x]^T [x]$ is:

$$([x]^T [x])^{-1} = \begin{bmatrix} 4.326^3 \times 10^{-5} & -2.8866 \times 10^{-5} \\ -2.8866 \times 10^{-5} & 2.0941 \times 10^{-5} \end{bmatrix}$$

and

$$\begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 4.3263 \times 10^{-5} & -2.8866 \times 10^{-5} \\ -2.8866 \times 10^{-5} & 2.0941 \times 10^{-5} \end{bmatrix} \begin{bmatrix} 297536.09 \\ 438313.00 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} .2200 \\ .5901 \end{bmatrix}$$

So our regression estimator for K/t_d is:

$$K/t_d = .2200\, x_2 + .5901\, x_3$$

$$K/t_d = .2200\ (\text{No. of rpts.}) + .5901\ (\text{No. of Application Programs})$$

Estimators for each of the other K/t_d^4 . . . K, Kt_d, Kt_d^4 terms are determined in the same way. The estimators can be determined with or without constant terms. For example, the following is a set of x_2, x_3, estimators with constants.

	R^2
$K/t_d^2 = .1700\, x_2 + .0930\, x_3 + 2.9132$	.7394
$K/t_d = .3052\, x_2 + .5693\, x_3 -.8.5173$	.8423
$K = .2775\, x_2 + 2.6407\, x_3 - .62.18$	.8586
$Kt_d = -1.3365\, x_2 + 11.2226\, x_3 - 280.04$	.8370
$Kt_d^4 = -311.51\, x_2 + 749.23\, x_3 -17218.52$	.66

While the constants fit the functions in the range where the data are best, they tend to distort or become nonsensical in the extreme ranges of the data, particularly for small values of x_2, x_3. Accordingly, a set of estimators without constant terms is better over the full range of inputs for x_2, x_3 and do go to zero for $x_2 = 0$, $x_3 = 0$ as they should. One sacrifices a degradation in R^2 (coefficient of determination) by not using the constant terms, but this loss is of no practical consequence.

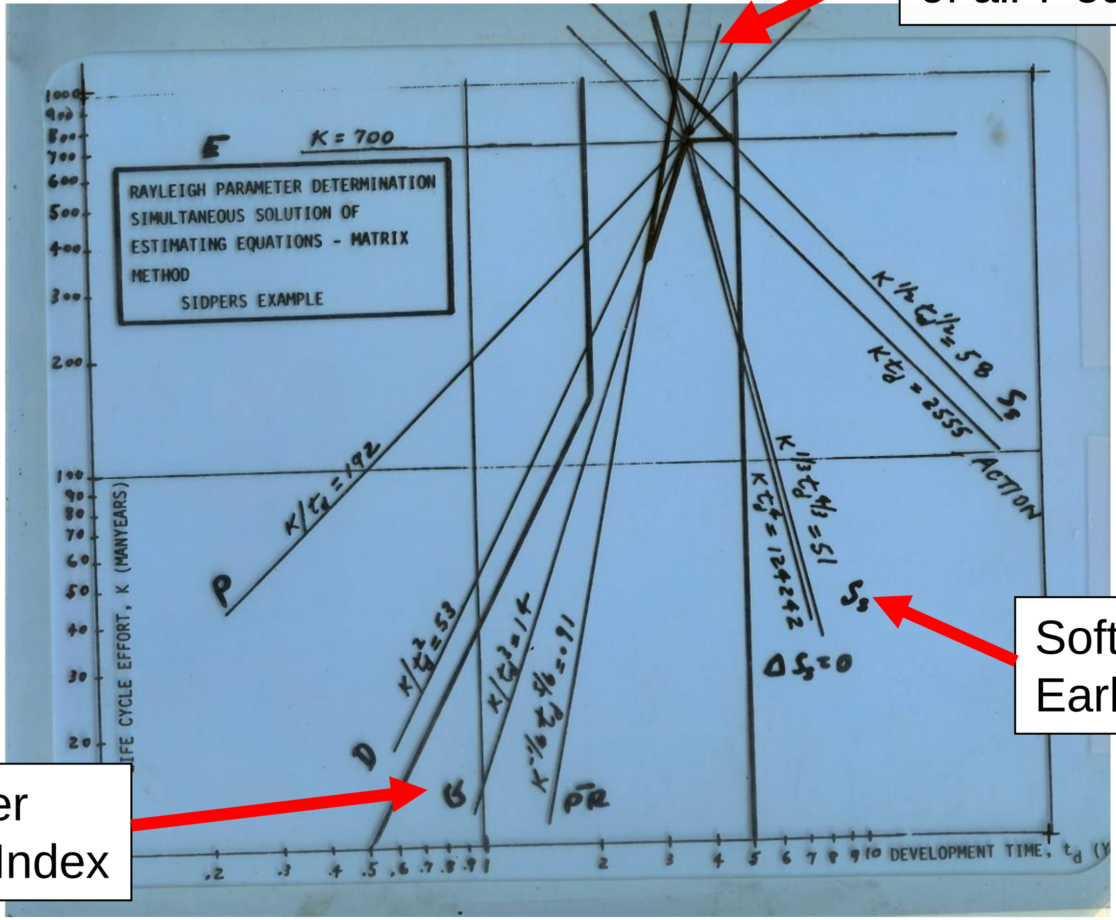
Matrix math to solve simultaneous equations

Goodness of Fit

5 equations relating the number of reports and application programs to time and effort….the parameters needed to generate the Rayleigh staffing model

Graphical Curve Fitting and Some of the Equations We Still Use Today

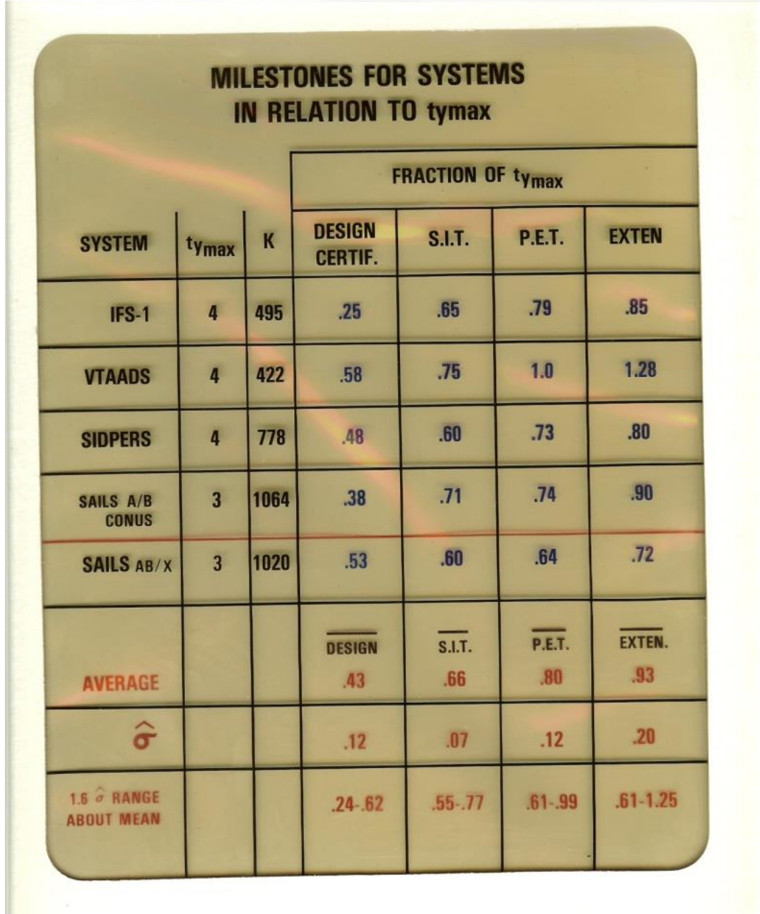
Mathematical solution of all 7 equations



Software Equation Early Version

Manpower Buildup Index

Schedule Milestone Determination



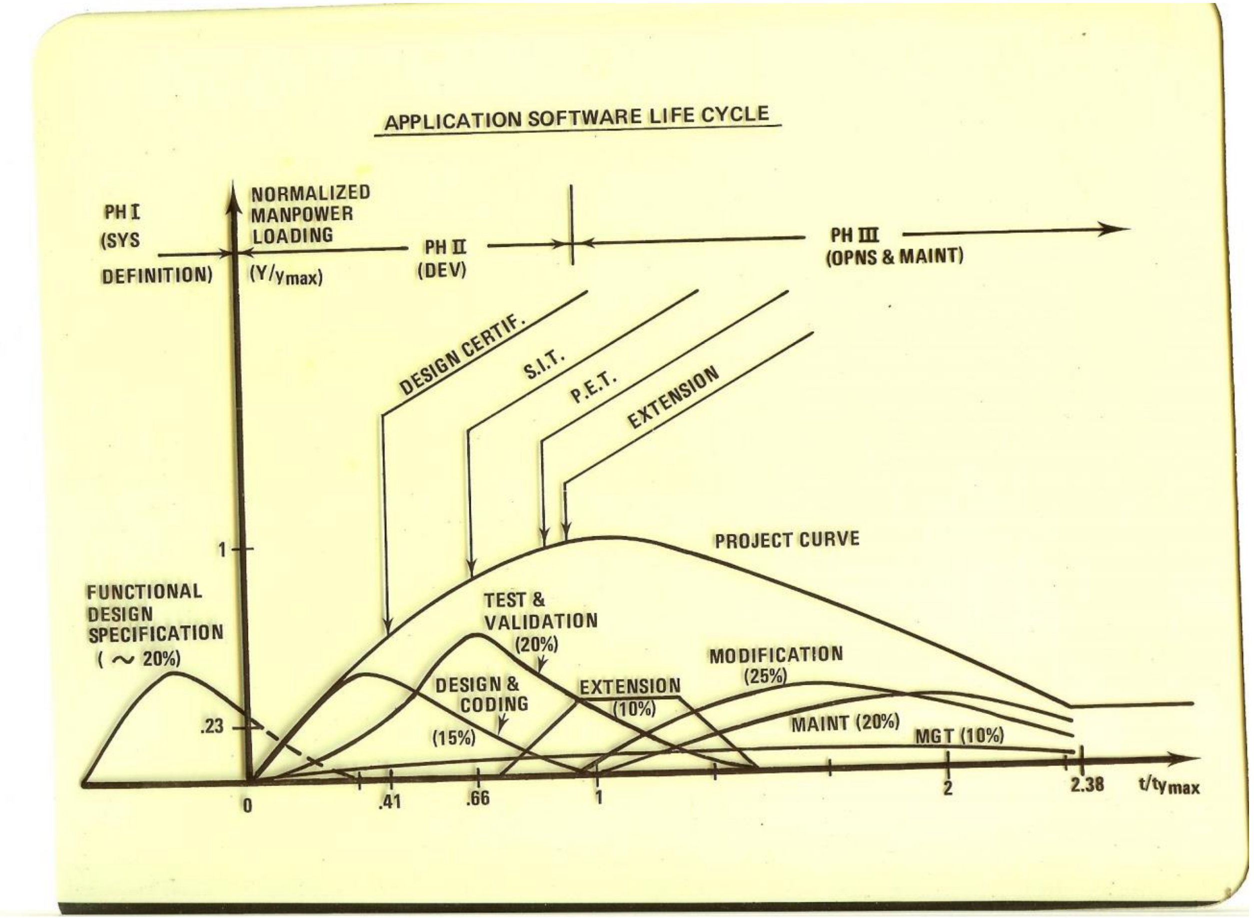
MILESTONES FOR SYSTEMS
IN RELATION TO tymax

SYSTEM	$t_{y_{max}}$	K	FRACTION OF $t_{y_{max}}$			
			DESIGN CERTIF.	S.I.T.	P.E.T.	EXTEN
IFS-1	4	495	.25	.65	.79	.85
VTAADS	4	422	.58	.75	1.0	1.28
SIDPERS	4	778	.48	.60	.73	.80
SAILS A/B CONUS	3	1064	.38	.71	.74	.90
SAILS AB/X	3	1020	.53	.60	.64	.72
AVERAGE			$\overline{\text{DESIGN}}$ .43	$\overline{\text{S.I.T.}}$ .66	$\overline{\text{P.E.T.}}$ .80	$\overline{\text{EXTEN.}}$ .93
$\hat{\sigma}$			.12	.07	.12	.20
1.6 $\hat{\sigma}$ RANGE ABOUT MEAN			.24-.62	.55-.77	.61-.99	.61-1.25

Milestone determination based on the average of these 5 Army Systems

The Putnam-Norden Rayleigh Model

End Product Estimate (Schedule, Effort and Milestones)



QSM®

Key Thoughts

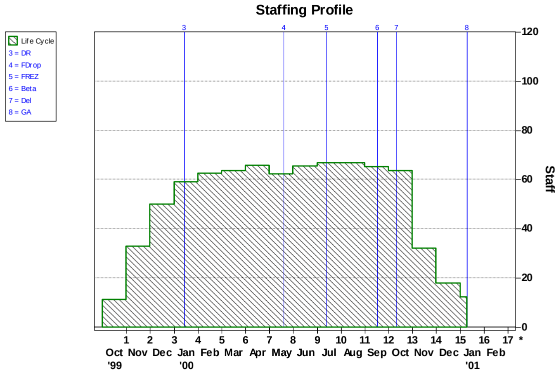
- Rayleigh Model is an excellent model of human design processes
- Software equation captures the time effort trade-off relationship
- The algorithms have withstood the test of time
- This suggests we must be close to the fundamental behavior of software development
  - How humans solve complicated design problems
  - Can be applied to other domains that exhibit similar behavior
    - Hardware, firmware, infrastructure, etc.

In Software (Design Processes) Everything is Non-linear

- Schedule & effort as a function of size
  - As the size and scope changes there is a non-linear change in the schedule and effort required to develop it
- Time effort trade-off
  - Models human communication complexity
    - Exponential increase in unique communication paths as we add people

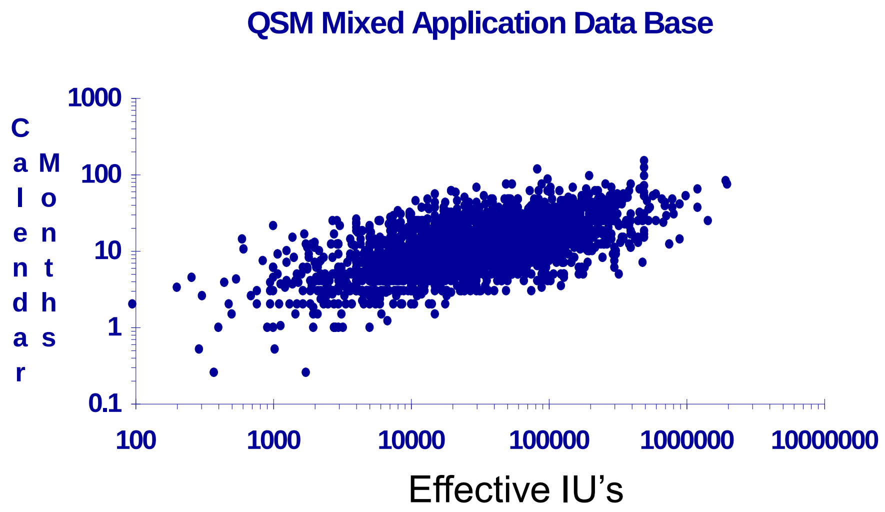
$$\frac{\text{Functionality}}{\text{Efficiency}} = \text{Effort}^{x} \times \text{Time}^{y}$$

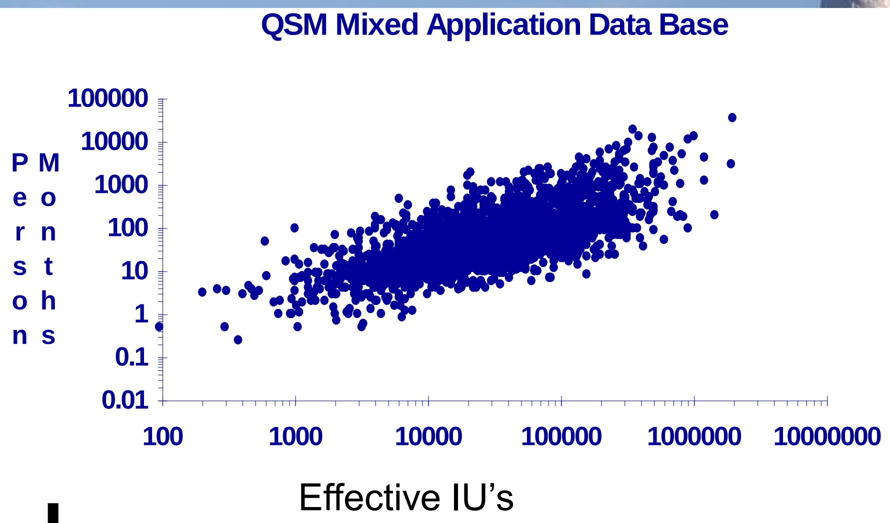
x=1/3
y=4/3

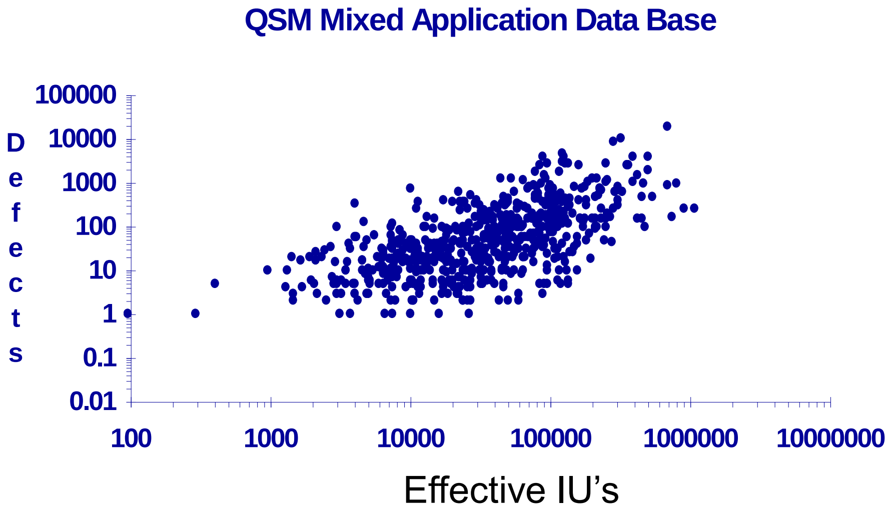


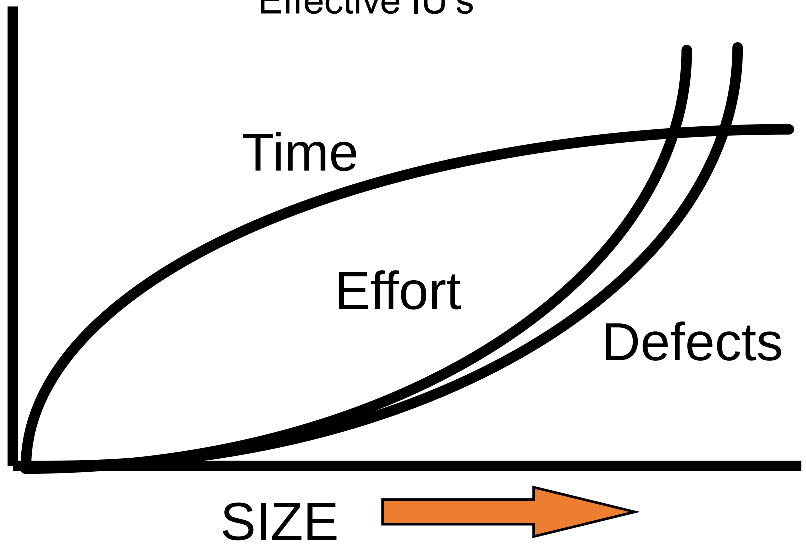
QSM®

Software’s Non-linear Behavior with Respect to Size



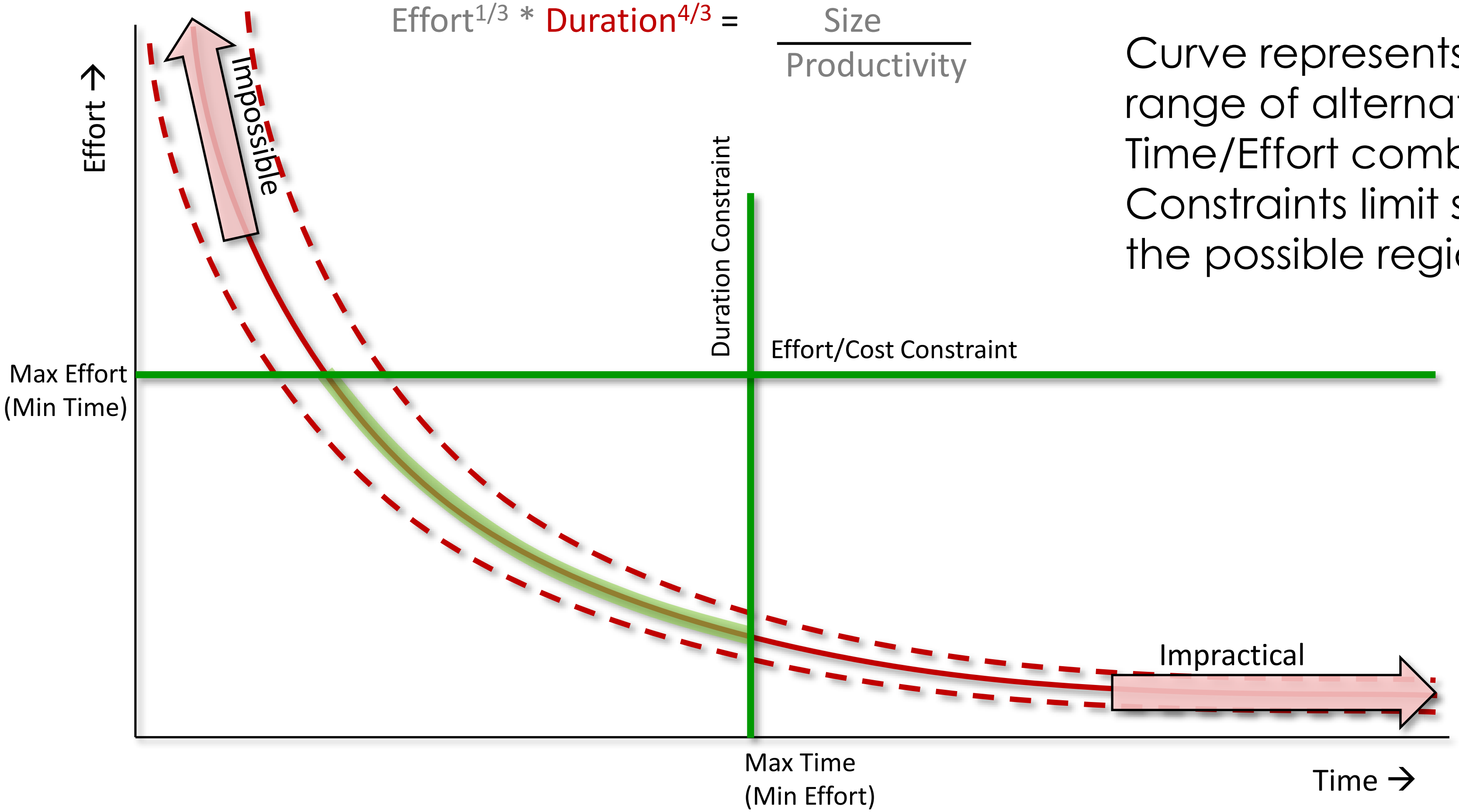






Time/Effort Trade-Off & Constraints

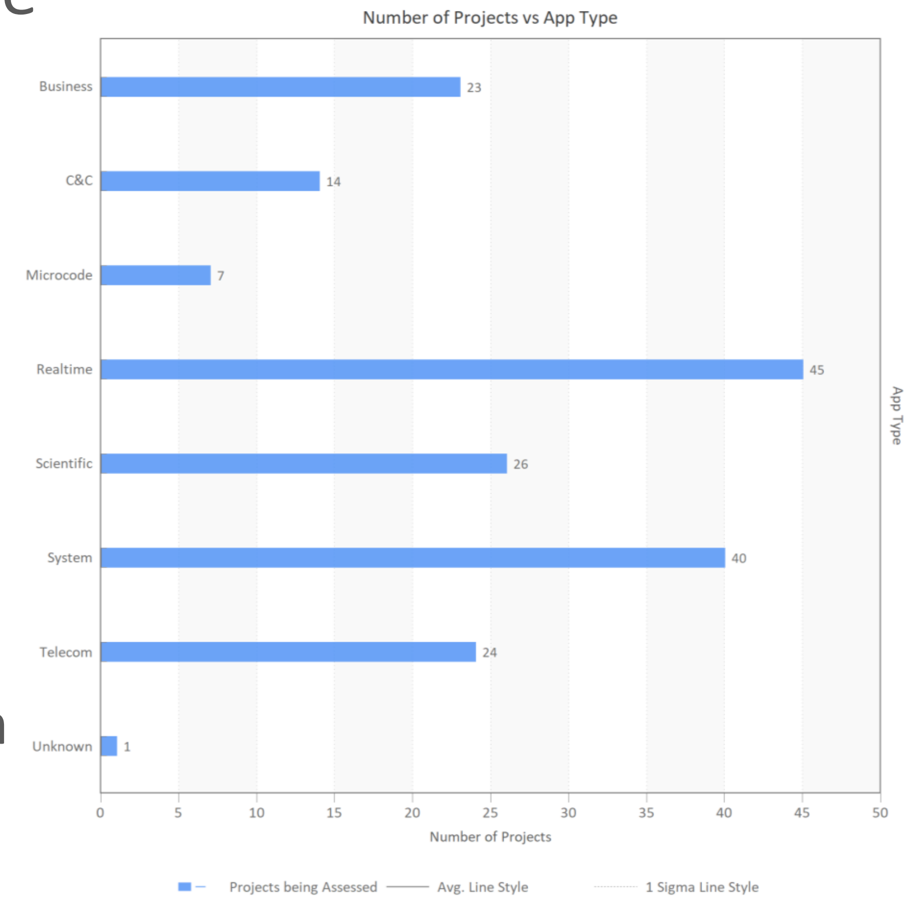
$$\text{Effort}^{1/3} * \text{Duration}^{4/3} = \frac{\text{Size}}{\text{Productivity}}$$



Curve represents the full range of alternative Time/Effort combinations. Constraints limit solutions to the possible region

Confirmation with RADC Data

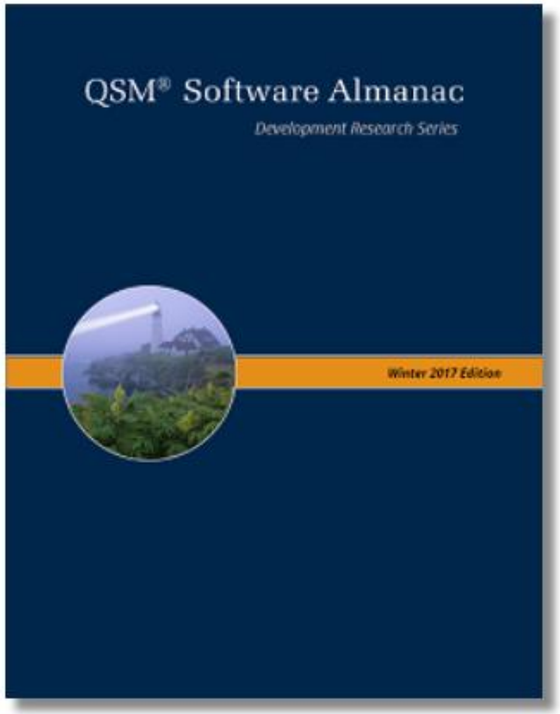
- Additional data became available from the Rome Air Development Center
  - Larger sample of data (180 projects) spanning multiple organizations
  - More diverse application complexity
    - Included Business/IT, Engineering & Realtime systems
  - Data covered a larger size range with multiple languages
- Similar Behavior Exhibited
  - Non-linear change in schedule & effort as a function of size/scope
  - Non-linear schedule/effort trade-off of different



QSM®

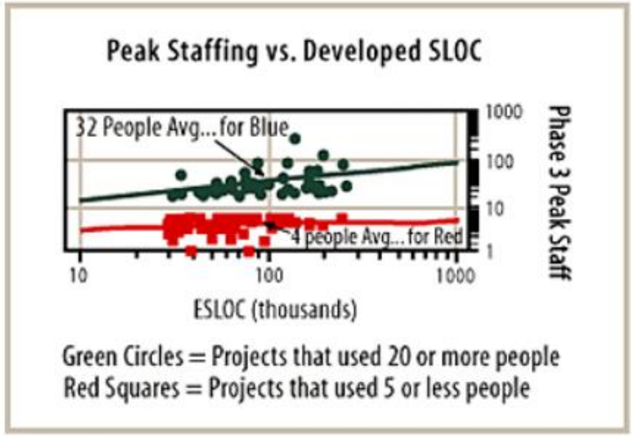
Do These Trends Stand the Test of Time

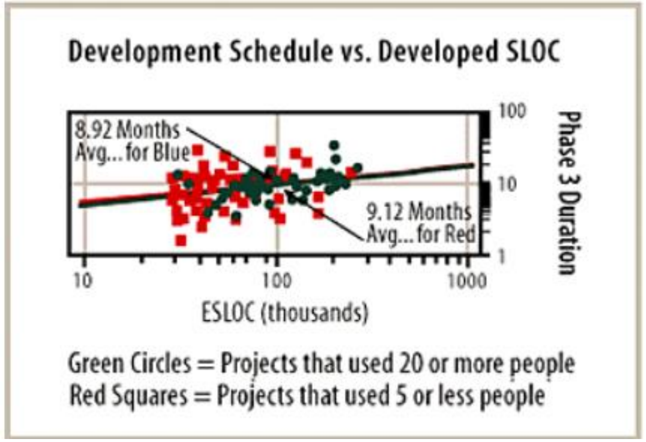
- QSM continues to collect industry data and revalidate the trade-off relationships
- We regularly publish new research in the form of QSM Software Almanac
    - http://www.qsm.com/resources/research/qsm-almanacs
- Work with clients to incorporate their own data
    - Custom sizing models
    - Productivity calibration
    - Methodologies used
        - Phase tuning
        - Staff loading
        - Skill allocations

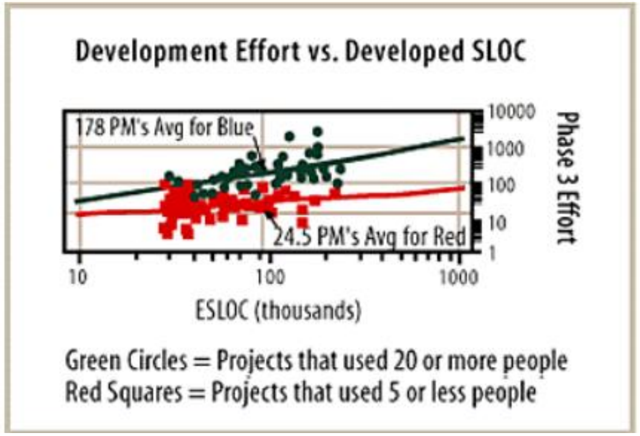


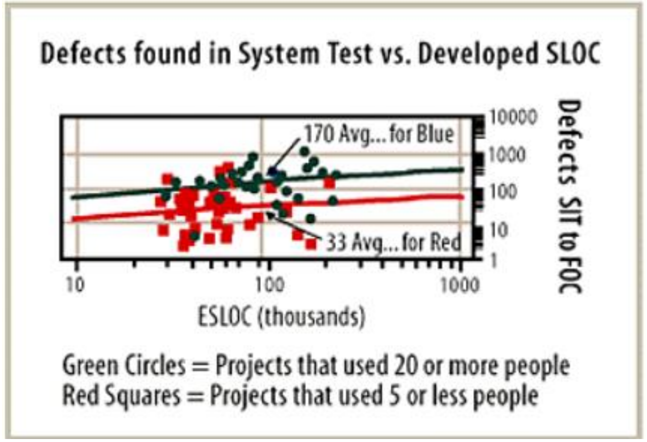
Revalidated the Trade-Off Relationship in the Mid-1990’s

- “Haste Makes Waste” article
  - http://www.qsm.com/risk_02.html





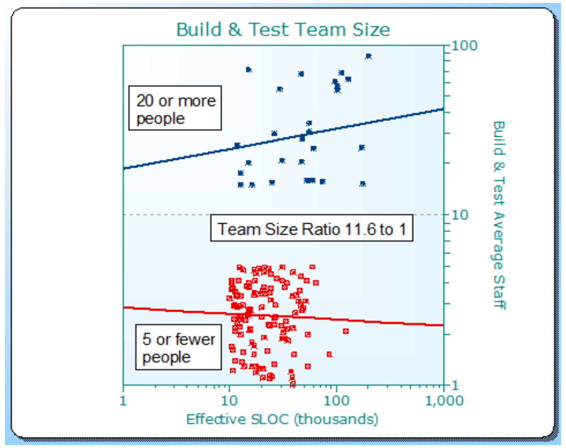


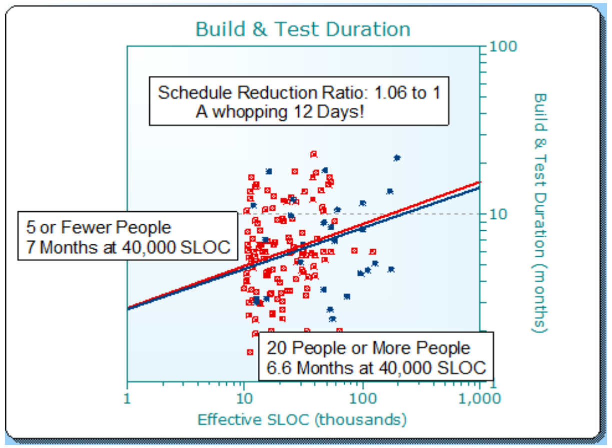


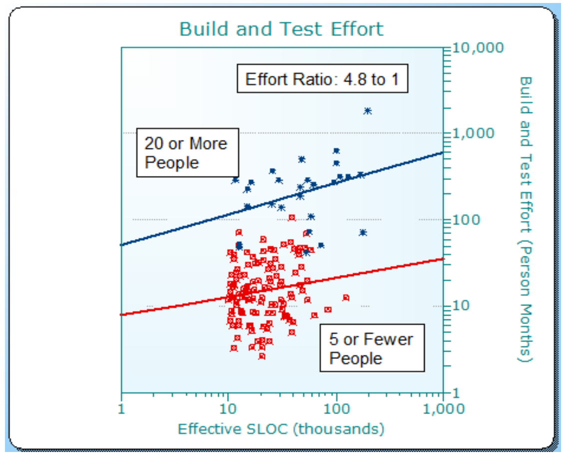
QSM®

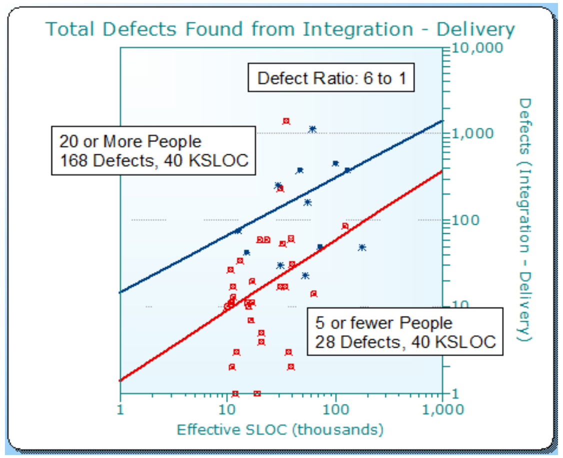
Revalidated the Trade-Off Relationship in the Mid-2000's

- 2006 QSM Almanac
  - http://www.qsm.com/resources/research/qsm-almanacs





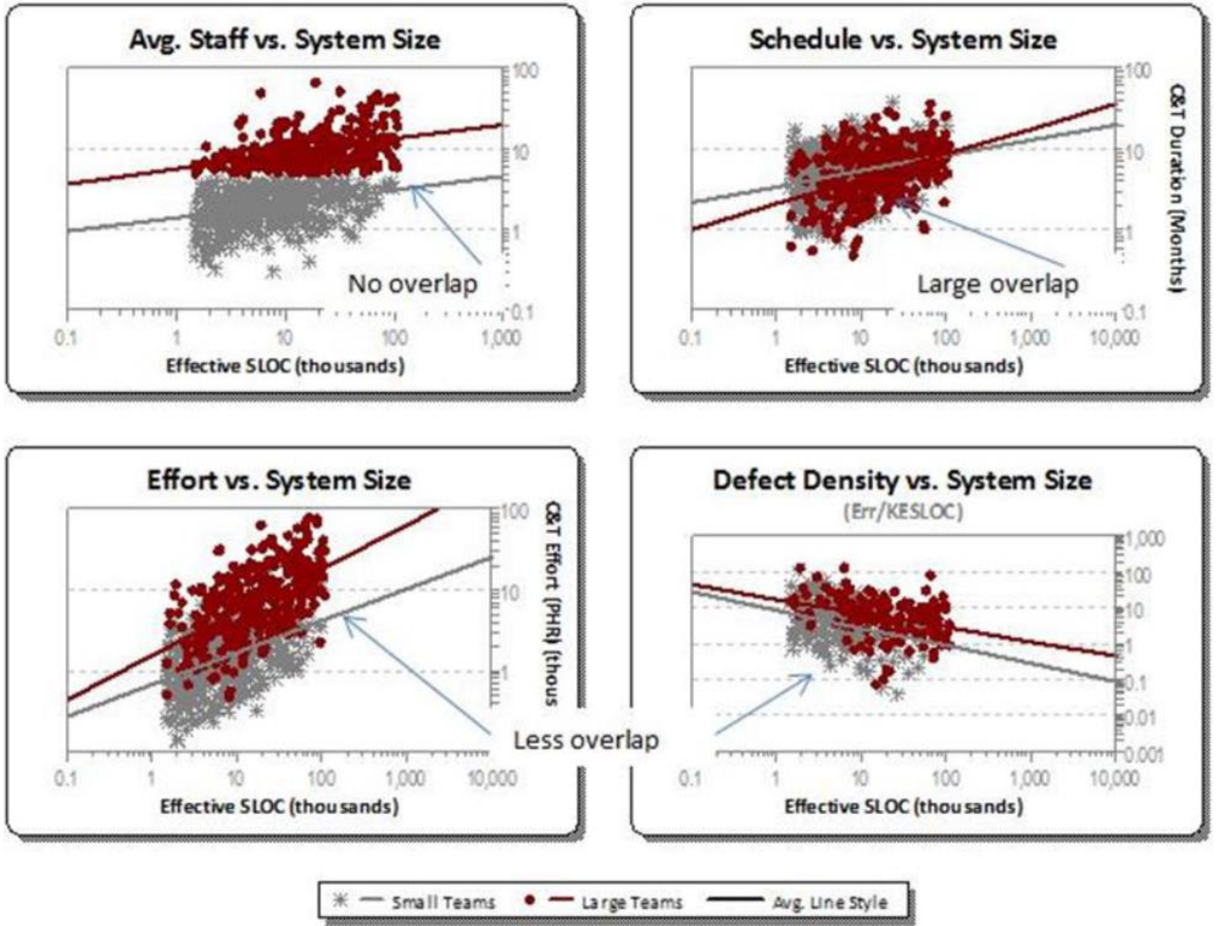




QSM®

Revalidated the Trade-Off Relationship in 2017

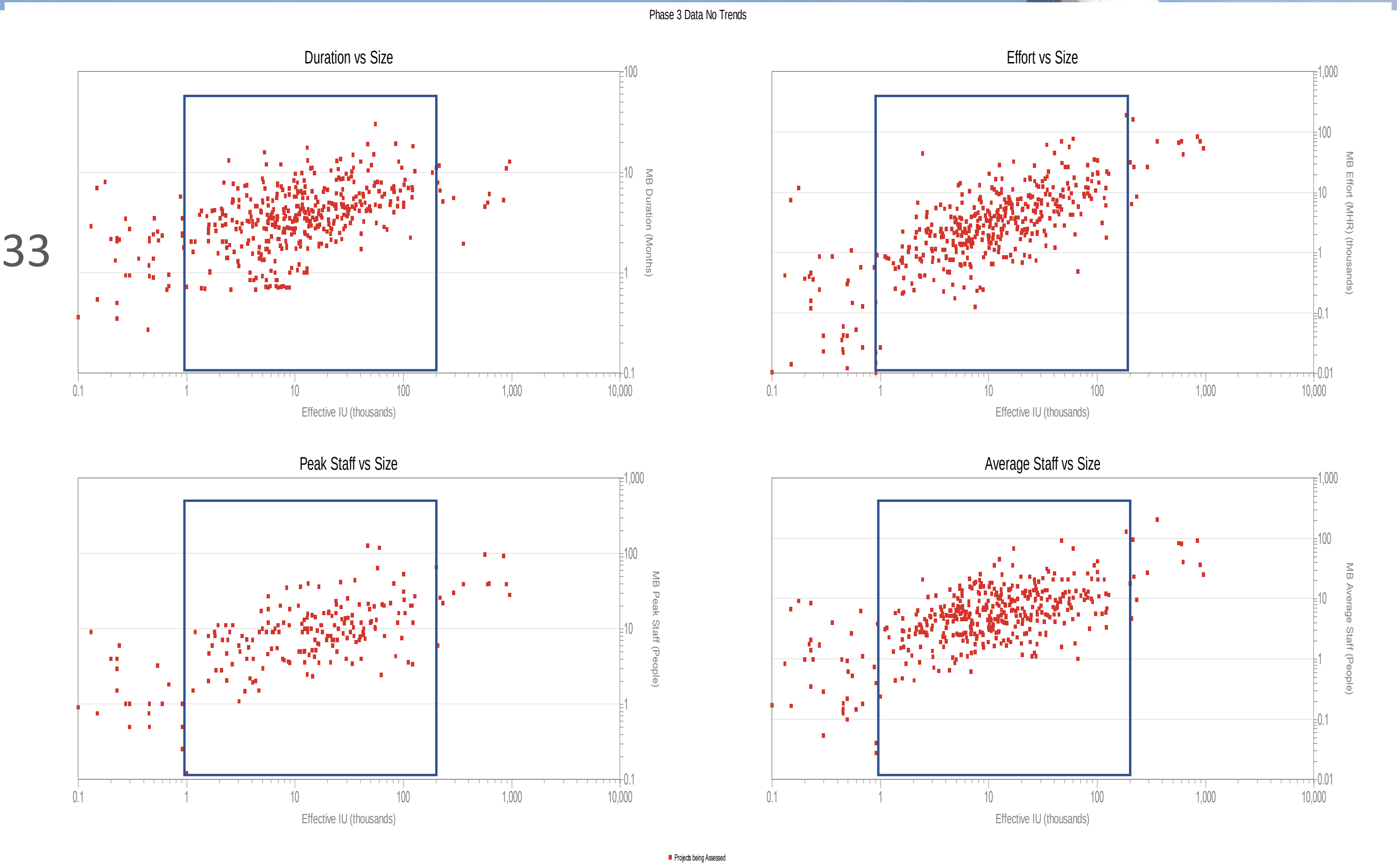
- 2017 QSM Almanac
  - http://www.qsm.com/resources/research/qsm-almanacs



QSM®

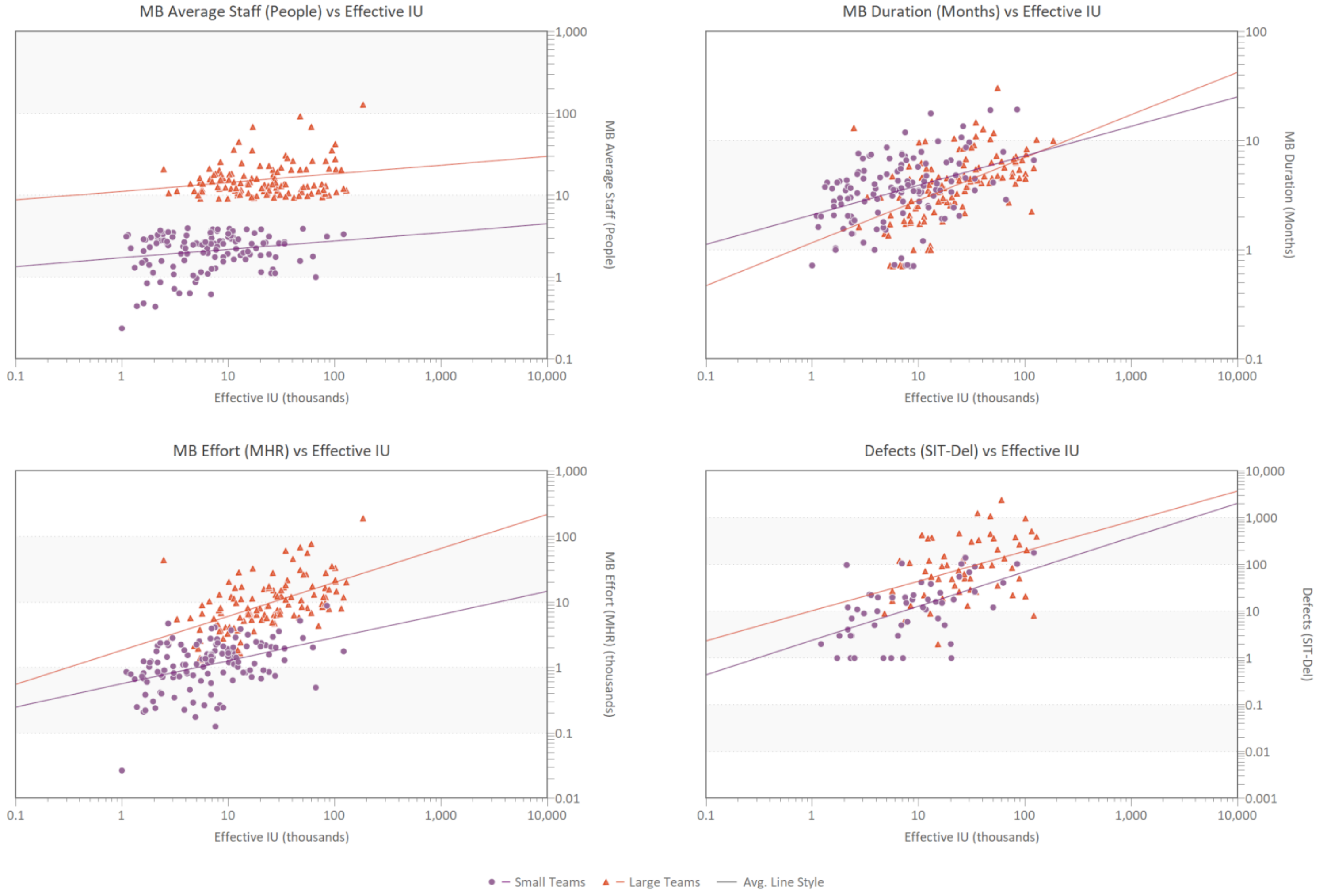
Do We See a Similar Pattern in Current Agile Data?

- Created two datasets with overlapping size data
  - Overall Sample
    - Avg staff median 5.36, mean 9.33
    - Size 100 IU and 953k IU
    - Number of projects - 436
  - Large Team
    - Avg staff >= 9 people
    - Size >=1k IU and <= 200k IU
    - Number of projects - 119
  - Small Team
    - Avg staff <= 4 people
    - Size >=1k IU and <= 200k IU
    - Number of projects - 129



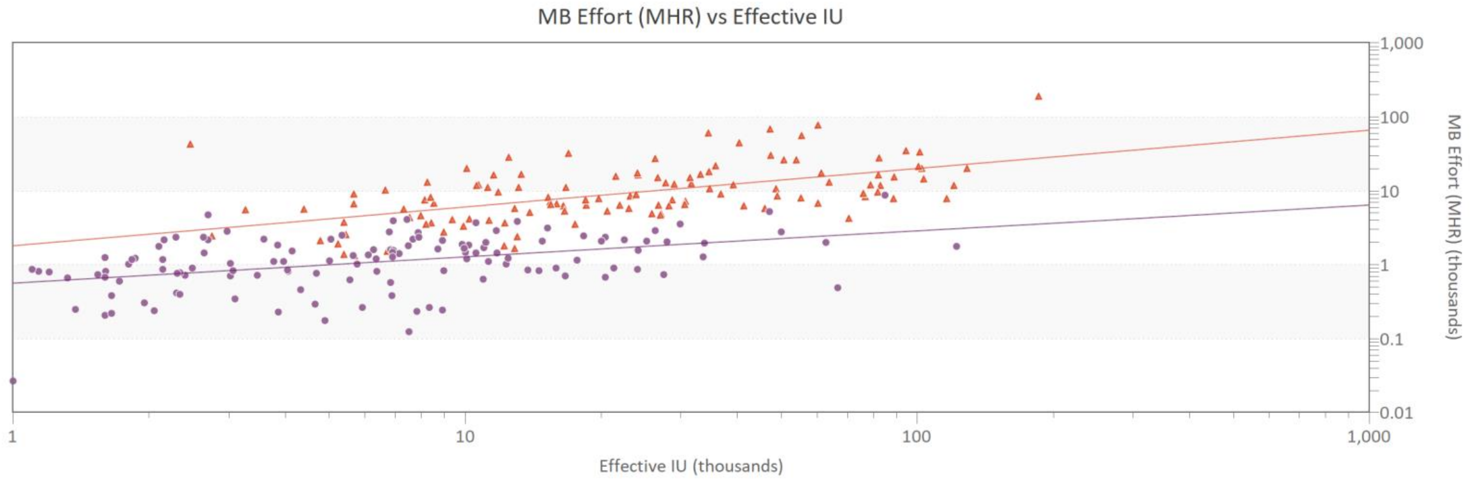
Trade-Off Relationship with Agile Data

Phase 3 Small-Large Team Trends



Effort Trade-Off Relationship with Agile Data

Dataset Comp - Phase 3 Effort



Comparison of Large Teams to Small Teams
MB Effort (MHR) vs. Effective IU

	MB Effort (MHR) Values				
	at Min Effective IU: 2464	at 25% Quartile Effective IU: 10600	at Median Effective IU: 23045	at 75% Quartile Effective IU: 47461	at Max Effective IU: 185500
Benchmark Data Set:					
Small Teams	778.70	1303.05	1713.83	2211.45	3577.41
Comparison Data Set:					
Large Teams	2924.72	6238.94	9337.63	13587.96	27577.17
Difference From Benchmark	2146.02	4935.90	7623.80	11376.51	23999.77

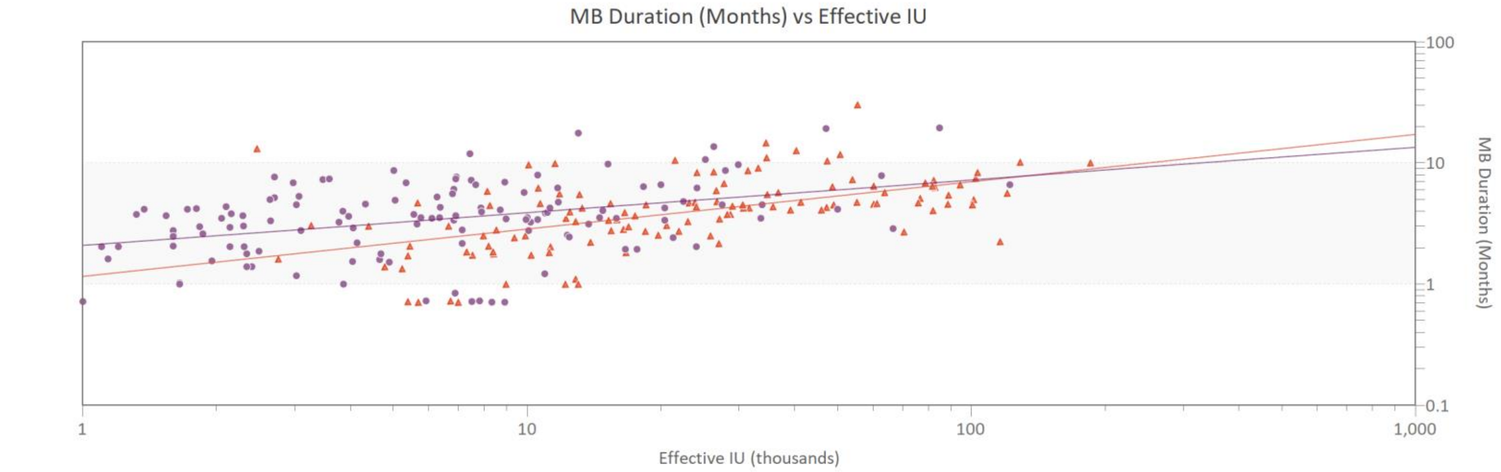
On average 4-8 times more effort expended by large teams for the same amount of functionality than the small team approach

Comparison breakpoints based on min, max, median and quartile values for the data set: Large Teams

● – Small Teams ▲ – Large Teams — Avg. Line Style

Schedule Trade-Off Relationship with Agile Data

Dataset Comp - Phase 3 Duration



Comparison of Small Teams to Large Teams
MB Duration (Months) vs. Effective IU

	MB Duration (Months) Values				
	at Min Effective IU: 1000	**at 25% Quartile Effective IU: 2670**	**at Median Effective IU: 6840**	**at 75% Quartile Effective IU: 11743**	**at Max Effective IU: 122000**
Benchmark Data Set: Large Teams	1.16	1.70	2.46	3.04	7.59
Comparison Data Set: Small Teams	2.09	2.72	3.51	4.07	7.66
Difference From Benchmark	0.93	1.02	1.05	1.03	0.07

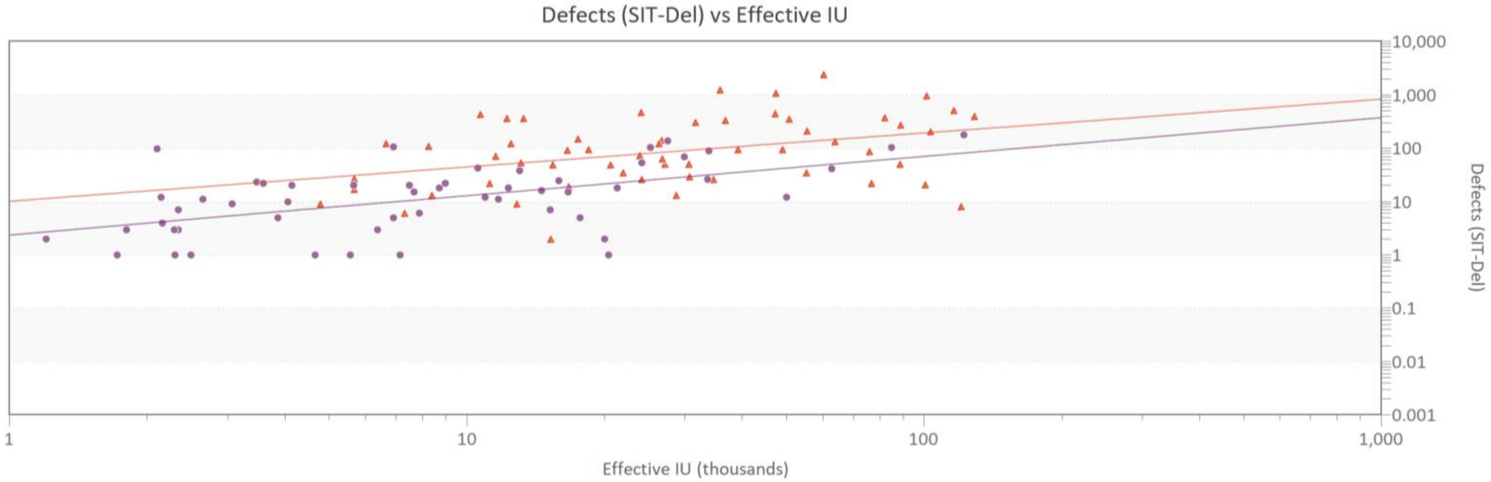
A month or less difference in schedule between large and small team data sets

Comparison breakpoints based on min, max, median and quartile values for the data set: Small Teams

● – Small Teams ▲ – Large Teams —— Avg. Line Style

QSM®

Defect Trade-Off Relationship with Agile Data



Comparison of Small Teams to Large Teams
Defects (SIT-Del) vs. Effective IU

	Defects (SIT-Del) Values				
	at Min Effective IU: 1100	at 25% Quartile Effective IU: 3263	at Median Effective IU: 7672	at 75% Quartile Effective IU: 15557	at Max Effective IU: 122000
Benchmark Data Set: Large Teams	10.80	21.66	37.42	58.81	219.57
Comparison Data Set: Small Teams	2.55	5.67	10.62	17.85	80.99
Difference From Benchmark	-8.25	-15.99	-26.80	-40.97	-138.58

3-4 times as many defects created by large teams than small team projects

Comparison breakpoints based on min, max, median and quartile values for the data set: Small Teams

● – Small Teams ▲ – Large Teams — Avg. Line Style

QSM®

Trends in Modern Development Methods

Modern agile & lean development methods seem to embrace these principles exploiting these positive economic trends - **Finally!**

- There is an emphasis on using small cross function teams
- Constant review and re-prioritization of functional content
  - Recognizes that change is inevitable
  - Emphasis on concept of “content must add value”
- Deliberately plan for short cycle times and strive for continuous delivery
  - Recognize that there is some minimum value that the customer will accept
    - Minimum Marketable Features (MMF) or Minimum Viable Product (MVP)

Trends in Team Size

- From the Scrum Guide Today

Development Team Size

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. Fewer than three Development Team members decrease interaction and results in smaller productivity gains. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Having more than nine members requires too much coordination. Large Development Teams generate too much complexity for an empirical process to be useful. The Product Owner and Scrum Master roles are not included in this count unless they are also executing the work of the Sprint Backlog.

- From QSM Research in 1997

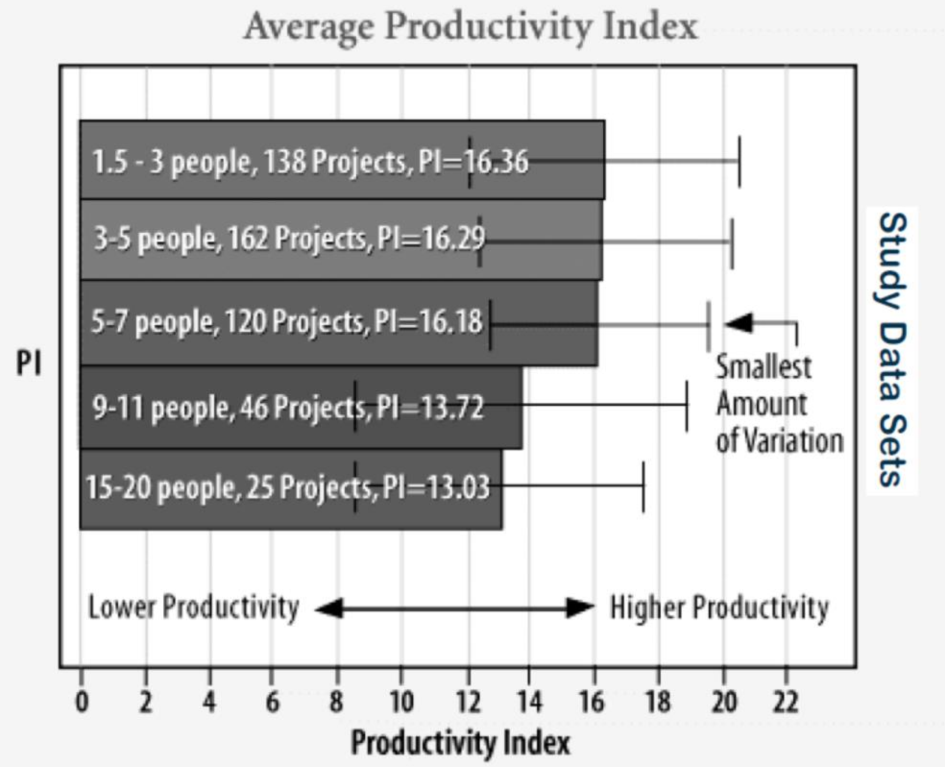
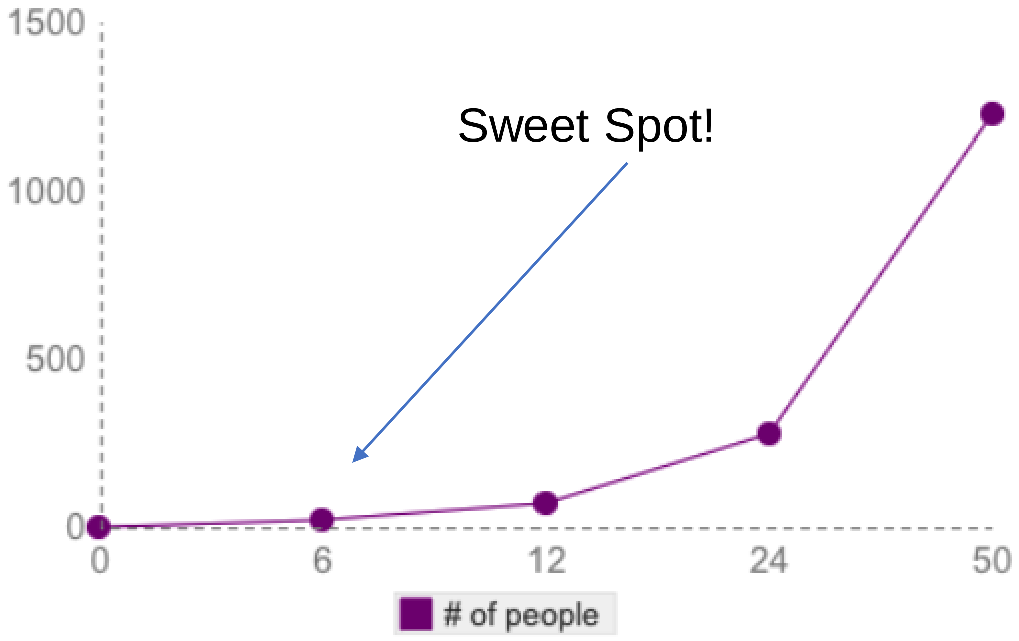


Figure 2. Average Productivity Index for each project staffing group with variation bounds.

QSM®

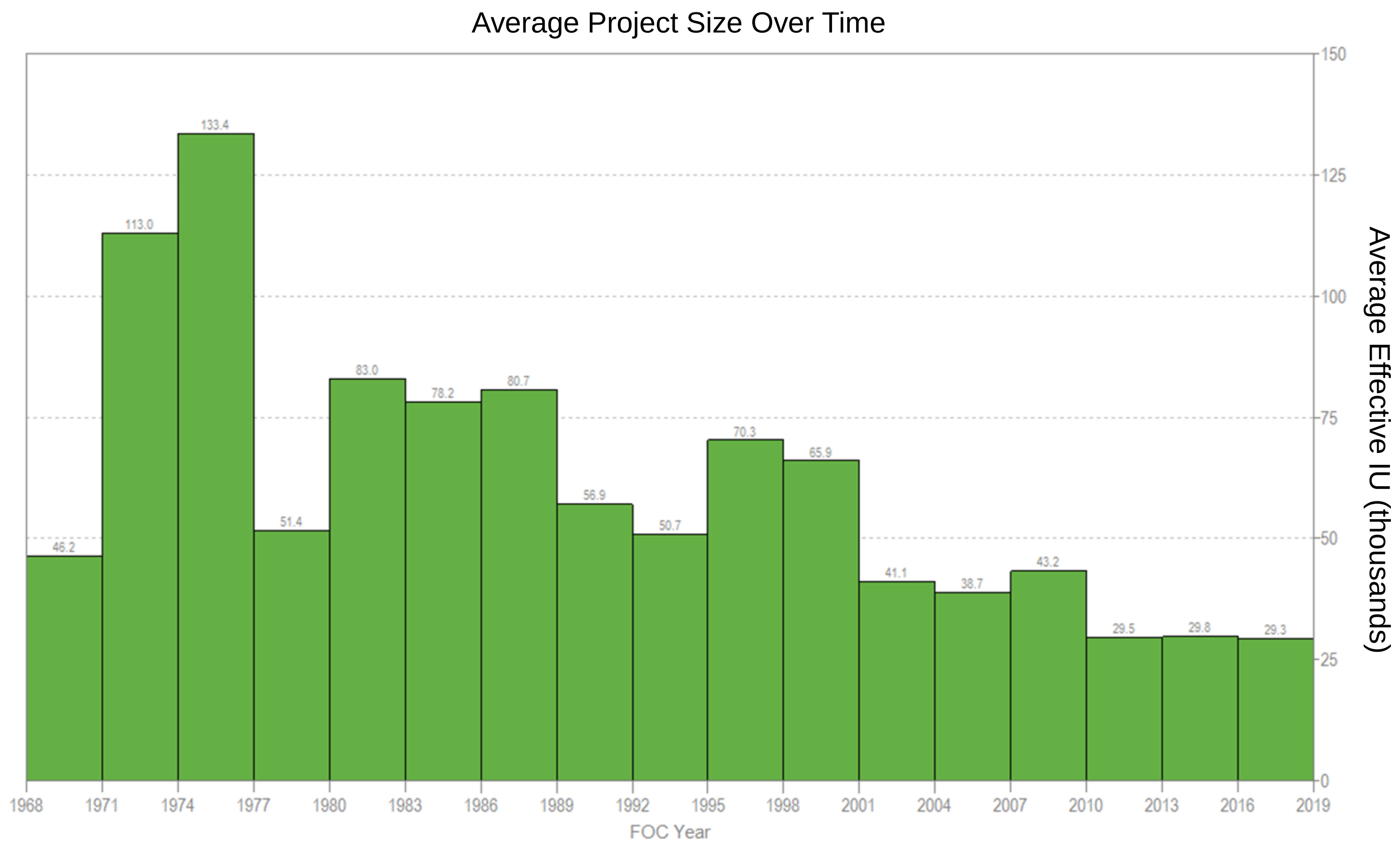
Two Pizza Rule

- Developed by Jeff Bezos to keep meetings more productive
- The science behind it matches what we see in the software data
- As we add people to the team we exponentially increase potential communication paths



QSM®

Trends in Project Size



General trend is that size has been declining over time. Early on this may have been more due to more powerful languages and technology.

More recently we believe this size reduction is more driven by reuse and a conscious effort to keep project size as small as possible.

Summary

- Larry's original research and estimating models are as relevant today as they were close to 40 years ago
- They have been adapted and improved to make them applicable to modern development practices and methodologies
  - More configuration and fundamental changes to the model
- Current agile data shows very similar non-linear behavior
- Current agile development methodologies seem to be exploiting these behaviors
  - Promotes small cross functional teams
  - Promotes keeping scope as small as possible to fit short iterations

QSM®