

---

# Navigating Agile Estimation Challenges





**Laura Zuber,  
Lead Support Representative and Instructor, CSM, SA**

Laura Zuber has 25 years of experience in software development consulting, training, and support. She has conducted training and coaching sessions for all QSM SLIM-Suite tools and helped customers implement SLIM across a wide variety of processes and platforms. Laura has managed software development projects, served as a senior software process improvement specialist, performed process assessments, designed and implemented best practices, and authored numerous training programs. She is a Certified Scrum Master and SAFe Agilist.



## It's Complicated!

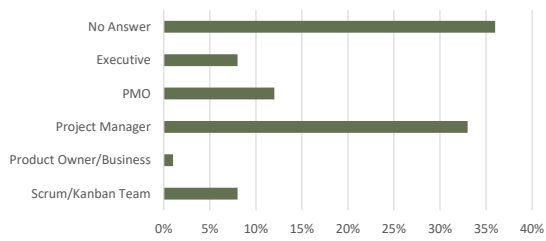


QSM<sup>®</sup> The Intelligence behind  
Successful Software Projects

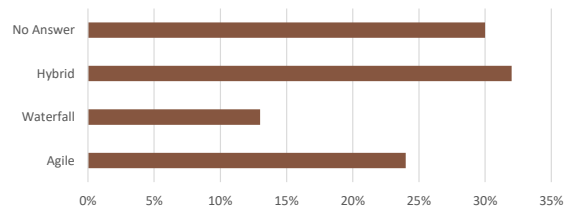
- Every business has challenges and each business probably thinks theirs are unique. Probably, but software development challenges are different mainly due to the need to create reliable products quickly in the midst of rapidly changing technologies and having to jump in, often with little information.
- Before we start to address common Agile estimation challenges, we'd like to know more about you and your world, so we have a couple of polls:
- What is your software project role?
- What category best describes the majority of your projects?

## Poll Results — 1 and 2

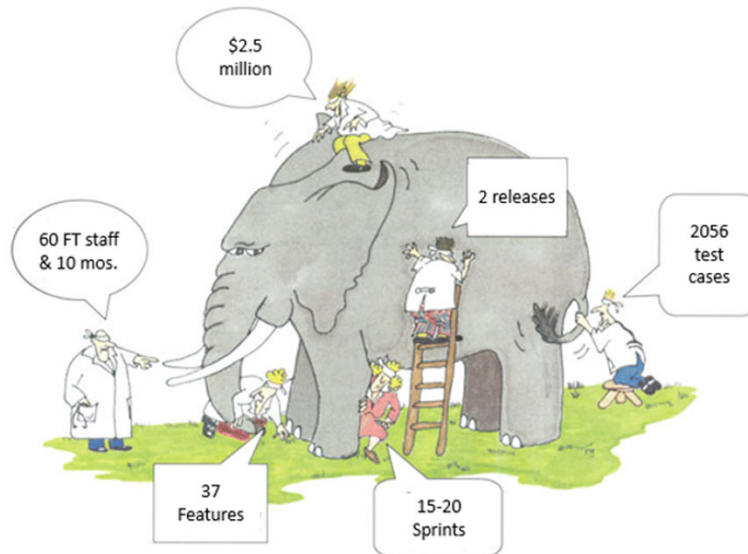
What is your software project role?



What category best describes the majority of your projects?



## Stakeholders View Product Development Differently



**QSM**<sup>®</sup>  
The Intelligence behind  
Successful Software Projects

- I asked about your role, because software projects have a variety of stakeholders – each has their own needs and perspective, and that means their challenges are varied as well.
- This fact alone is one of the biggest sources of typical estimation challenges. It is hard to get everyone on the same page. There are different goals that are sometimes in conflict – like do it fast & make sure it's high quality – especially early in the life cycle when all the details have yet to be worked out and assumptions have to be made.
- Many organizations are at various levels of Agile implementation or only use Agile for certain project types. Variability here can create challenges.

---

## Why We Estimate

- Strategic enterprise need
- Win new business
- Prioritize the backlog
- Budget
- Schedule
- Resource needs
- Technical challenges



- You may not have thought about it, but there are different reasons why we estimate.
- {List}
- These are the typical reasons we see.

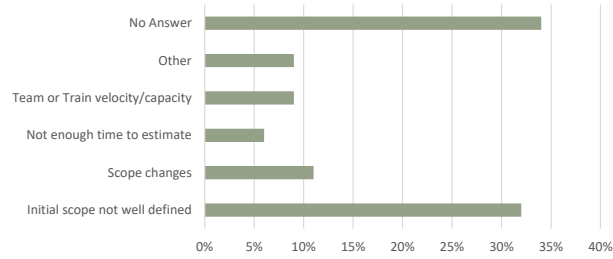
## What Are Your Estimation Challenges?



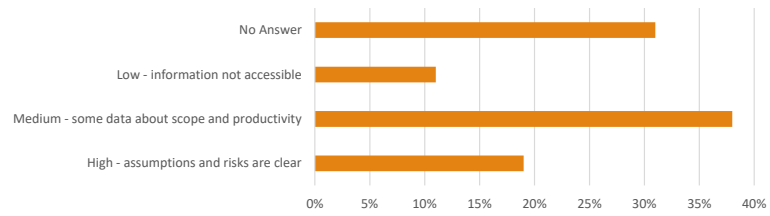
- Before we start navigating through Agile Estimation Challenges, I want to make sure I understand a little about your struggles.
- A couple more quick polls, please:
  - What is your biggest estimation challenge?
  - How would you rate your visibility of project estimates, namely the assumptions upon which they are based?

## Poll Results — 3 and 4

What is your biggest estimation challenge?



How would you rate your visibility of project estimates?



## Software Lifecycle Management Challenges

### Estimation • Tracking & Forecasting • Closeout

- Scope not well defined
- Scope / Size measurements inconsistent
- Agreement on need to estimate
- Velocity / Productivity unknown & fluctuations
- Historical Data not available
- Assumptions regarding Time & Effort Tradeoff
- Tracking & Forecasting
- Measuring Deliverable Quality

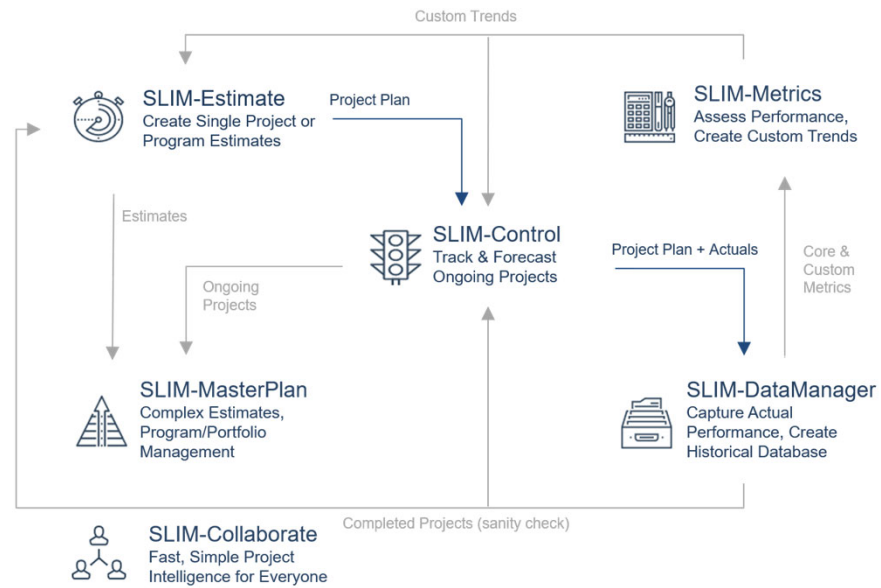
- The title of this slide is not “Agile Estimation Challenges” for two reasons:
- The software development challenges I’ll describe are not unique to Agile methods / organizations
- Estimate challenges are really lifecycle management challenges – across all stages: Estimation \* Tracking & Forecasting \* Closeout
- [List]
- The challenges highlighted in blue are more likely to be issues with Agile.

---

# SLIM APPROACH



## SLIM-Suite Tools



**QSM**<sup>®</sup>  
The Intelligence behind  
Successful Software Projects

- For those who are unfamiliar with SLIM-Suite tools, I will briefly describe each one and describe some key principles and modeling parameters embodied across the entire suite, before we address Agile Estimation Challenges.

## The Software Production Equation



Delivered System Size Is proportional to Effort over Time at some level of Productivity

Value Delivered

Resources Expended

Duration Required

Influenced by Capability and Difficulty of the task

This equation can be re-arranged to solve various estimation problems (Wizards).



12

- The software production equation is at the heart of SLIM tools (developed by our company founder Larry Putnam Sr). To deliver value to customers and/or management, we need to apply some amount of Effort over some amount of Time at some level of Productivity
- For straightforward estimate calculations, inputs are Size and Productivity.
- Outputs are Time and Effort
- Equation can be re-arranged to take what we know to solve for what we don't know. This provides different strategies for the different reasons we perform estimates. For example, our Time Boxed, Fixed Team wizard uses assumptions about Effort, Time, and Productivity to determine the amount of Size that can be delivered by a certain team for a particular application type.
- Time and Effort trade off against each other. A project with given size and productivity can be developed in less time with more effort, or more time with less effort.
- This time/effort tradeoff can be measured and modeled empirically. It is non-linear (and thus, hard for humans to predict).

## Productivity Example

Project A	Project B
Size = 300 FP	Size = 300 FP
Effort = 1038 Hrs	Effort = 1038 Hr
Duration = <b>1 Month</b>	Duration = <b>6 Months</b>
Team Size = <b>6 FTEs</b>	Team Size = <b>1 FTE</b>
Productivity = FP/Effort	
.289 FP per Hour	.289 FP per Hour

Project A	Project B
PI = <b>26.3</b>	PI = <b>16.4</b>
MTTD = <b>.755</b>	MTTD = <b>4.51</b>

SLIM's Productivity Index is not a traditional linear measure like sprint velocity. It measures the efficiency of your development environment, not just people. It is calculated from completed projects.

Productivity is affected by a host of factors, some of which can be quantified and some of which cannot:

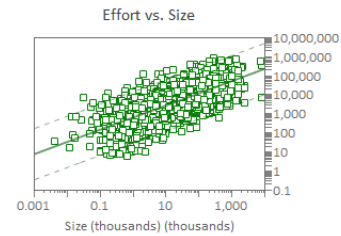
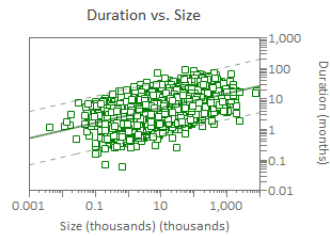
- Personnel Factors: . . . morale, experience, management capability,...
- Technology Factors: . . .tools, infrastructure, language,...
- Process Factors: . . . . . standards, maturity level, precedentedness...
- Product Factors: . . . . . size, complexity, hardware integration,...
- Reuse: . . . . . existing software, existing documentation...

[Walk through example]

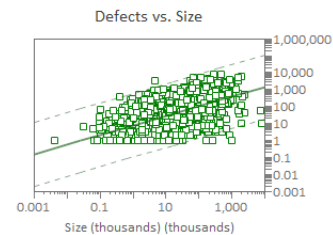
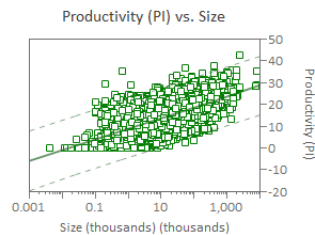
**Challenge met – measuring productivity for the entire environment at the release level gives greater predictability.**

## Estimation Expertise in a Box!

Core metrics scale non-linearly with project size. These relationships are robust and consistent over time (38+ years!) and across domains.

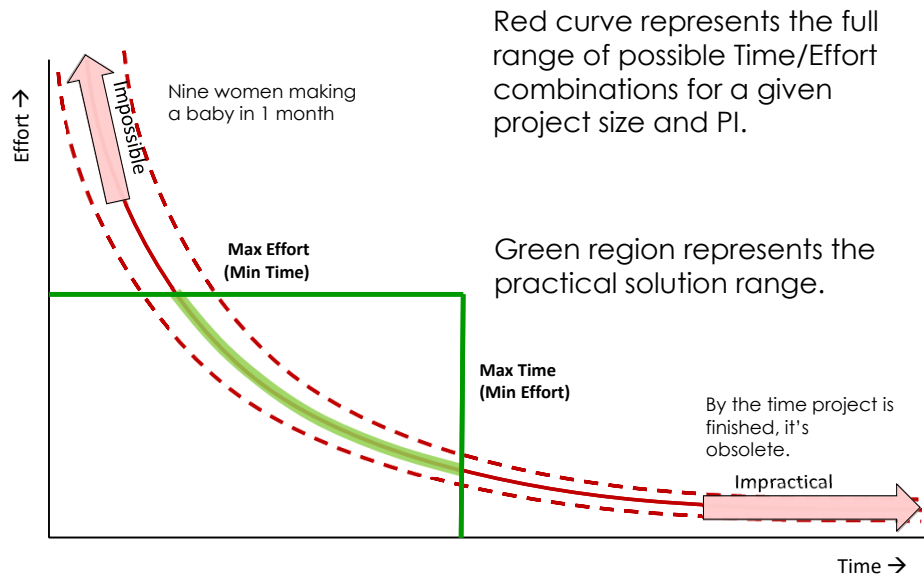


Empirical knowledge of these relationships is "built in" to SLIM-Estimate's algorithms, charts, and reports.



- Objective, Expert Judgment in a Box!
- Core Metrics increase with Size
- The increase is nonlinear
- Different trends for different application types

## Time-Effort Tradeoff



- [Describe chart]
- Projects can be scheduled and staffed in different ways but for a given project, time and effort are NEVER completely independent.
- Software development is not manufacturing!

---

# AGILE ESTIMATION CHALLENGES



## Software Sizing Challenges



- Function Points
  - Standard
  - Platform Agnostic
- Story Points
  - Customer Satisfaction
  - Inconsistent
- “Big Rock” relative sizing
- Estimates vs Actuals

“Even though we’re not trying to be consistent, it will work out that we are.”

- Andy Berner

**QSM**<sup>®</sup>  
The Intelligence behind  
Successful Software Projects

- QSM customers have shared their Agile estimation challenges over the years. We wanted to create away for them to share ideas and get help from QSM also.
- We conducted a several month-long forum with customers in 2018 called the Agile Round Table.
- The majority of discussion was related to sizing:
  - Sizing units – [bullets]
  - Consistency – Andy’s quote
  - Estimated versus what is actually delivered – often not measured

## Multiple Software Sizing Methods

### T-Shirt sizing

Size Calculator

Include	Name	Method	Expected Size	Linked Filename
Yes	T-Shirt	Sizing by History	200	
Yes	Stories and Epics	Sizing by Decomposition	184	

Double-click any existing selection to edit. Click a column header to sort the estimates.

Buttons: Add..., Edit..., Remove, Refresh, Un

Summary of Combined Sizing Estimates

Average Individual Estimates      Expected total SPTs: 192

Sum Individual Estimates      99% Range: 130 to 253

Note: Total Size includes all new, modified, and unmodified code.

Buttons: OK, Cancel, Help

### Epics & Stories

Sizing by Decomposition

Technique Name: Stories and Epics

Include	Component Name	SPTs per Component	# of Components
<input type="checkbox"/>	Very Large Epics	150.00	
<input type="checkbox"/>	Epics	55.00	
<input type="checkbox"/>	Large Stories	21.00	
<input type="checkbox"/>	Average Stories	5.00	
<input type="checkbox"/>	Small Stories	3.00	
<input type="checkbox"/>	Throwins	0.50	

Right-click on any cell for editing menu.

Results

Expected total SPTs: na      99% Range: n/a

Progress bar: Low to High

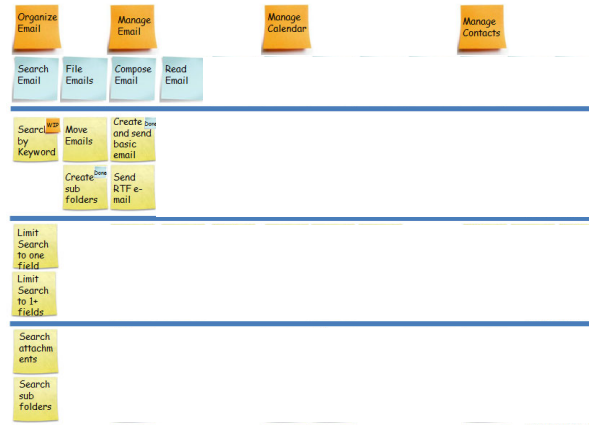
Buttons: OK, Cancel, Help



- SLIM is extremely flexible when it comes to measuring size. Different methods are appropriate for different kinds of projects and at different points in the lifecycle.
- T-shirt sizing is based on statistical trends that showed earlier, either industry data or your own, let's you create defensible estimates when you don't know much. The T-Shirt sizes are automatically calculated from the range of projects in your historical database.
- Once you start breaking out features and stories, you can quantify size by complexity. Size estimates are easily updated to account for scope changes.
- Challenge met – scope not well defined and scope changes.

## How do you get total size from Jira?

For Epics refined into stories, add the story points of the “leaf” stories



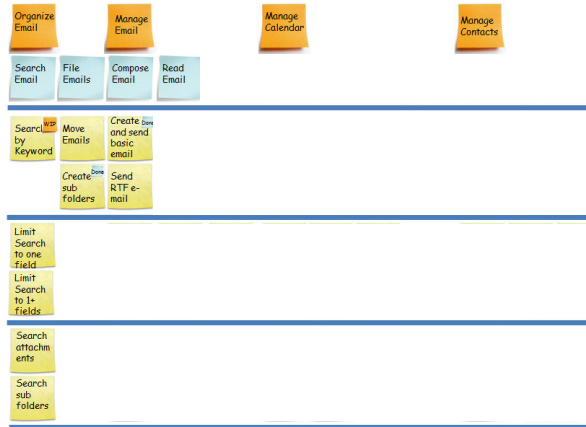
Some Stories  
not traced  
back to Epics

Steve Rogalski  
<http://winnipegagilist.blogspot.com/2012/03/how-to-create-user-story-map.htm>

- Let's take a look at how we can determine the size of your stories and epics from a tool like Jira.
- SLIM has an API used to prototype this integration; could be any other tool.
- There are many ways organizations use Jira, but QSM will work with you to map your data to what is needed.
- Example User Story map of the hierarchy of Epics and Stories – multiple levels of medium to small stories.
- Early on, only some Epics are refined.

## How do you get total size from Jira?

For Epics not yet refined, use the average of the refined epics



Steve Rogalski  
<http://winnipegagilist.blogspot.com/2012/03/how-to-create-user-story-map.htm>



And don't forget to add in the story points for stories not traced back to an epic!

- Take story points for each of the leaf stories, add them up to get the size of Epics.
- Make the assumption the Epics that have not been refined are about the same size.
- Categorize each of the Stories to a size bin based on the StPnt count.
- Add the individual Stories

## Jira Size Date Imported to Sizing Calculator

The screenshot shows the 'Sizing Calculator' application window. At the top, there is a dropdown menu for 'Apply Configuration Set' set to 'Project Specific Configuration'. Below this is an 'Add Line Item' button. The main area contains a table with the following data:

Include	Component Name	StPts/Component	# Components	
<input checked="" type="checkbox"/>	Epics (unrefined)	170.27	2	X
<input checked="" type="checkbox"/>	Epics (refined)	0.00	11	X
<input checked="" type="checkbox"/>	Throwins	1.00	21	X
<input checked="" type="checkbox"/>	Small Stories	2.76	103	X
<input checked="" type="checkbox"/>	Average Stories	7.77	202	X
<input checked="" type="checkbox"/>	Large Stories	21.00	4	X

Below the table, there are two input fields for totals: 'Calculated Total (StPts):' with the value 2299.36 and 'Current Total (StPts):' with the value 2298. A 'Calculate' button is positioned to the right of the first field. At the bottom, there are 'OK' and 'Cancel' buttons.

---

# PROJECT ESTIMATING



- Now I'd like to take what we've discussed so far and see how we apply this specifically to Agile projects in the big 3 lifecycle activities:
- Estimation
- Tracking & Forecasting
- Closeout

---

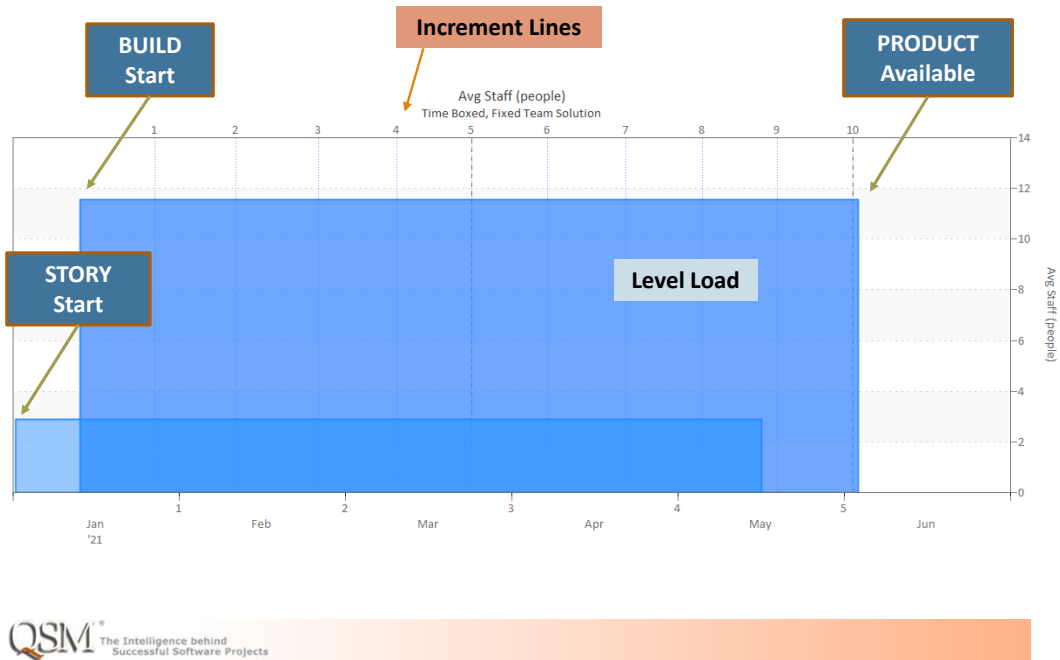
## Multiple Estimate Scenarios to Assess Risk

- What if?
  - Scope increases
  - Productivity is low
  - Staffing is increased / decreased
  - Budget is cut
  - Schedule changes
- Quickly compute multiple estimate scenarios
- Compute probability of potential outcomes
- Determine contingency needed to mitigate risks



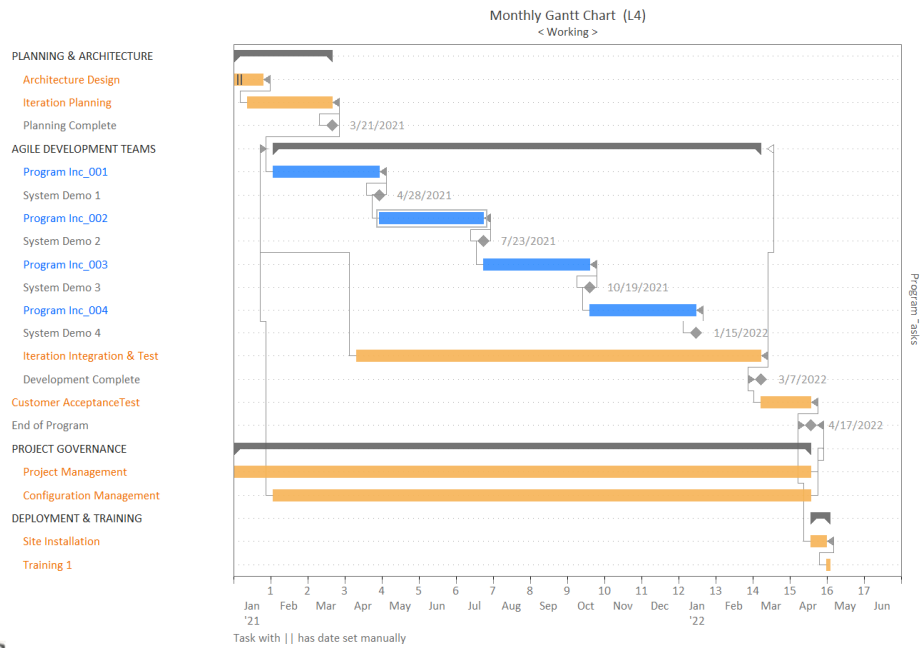
- Estimating Size is the biggest challenge. And as I've described, effort, costs, and time are driven by size.
- One of SLIM's greatest benefits is the ability to explore a range of potential outcomes [list]

## SLIM-Estimate – Agile Release Planning



- Let me show you some of the features of each of our estimation tools.
- SLIM-Estimate:
  - Models the development of a consumable release.
  - 2 major groups of activities – story writing and build & test
  - Arranged to model iterative sprint activity.
  - See impact of decisions to schedule with Agile Sprint lines

## SLIM-MasterPlan – Multiple Releases / SAFe



**QSM** The Intelligence behind Successful Software Projects

- SLIM-MasterPlan:
- Models scaled Agile methods
- Include non-development tasks
- Extremely flexible hierarchy
- Show dependencies
- Based on value produced each increment, adjustments can quickly be made with What-if analysis here as well.

## SLIM-Collaborate – Stakeholders & Workflow

**Project List**

Choose/Save Filter... Clear Filter

No Saved Filter Chosen

Project	OBS Node	Stage	Effective IU	Start Date	End Date	Active Phases	Keywords
SharePoint Integration	Singapore(and above)	Closeout	55,000	10/1/2015	3/31/2017	Concept/Planning, Require	Package Implementation, T
Care Plan	Brisbane(and above)	Closeout	22,000	6/5/2016	2/28/2017	Requirements & Design, De	Web
Training Module	Singapore(and above)	Closeout	12,000	5/16/2016	1/26/2017	Requirements & Design, De	Web
Records Migration	London(and above)	Estimation	71,262	6/1/2018	1/19/2019	Requirements & Design, De	
Patient Portal Phase 1	London(and above)	Closeout	69,000	6/1/2018	2/26/2019	Requirements & Design, De	MEDICAL RECORDS
Constant Care	London(and above)	Closeout	41,000	6/1/2018	1/16/2019	Requirements & Design, De	MEDICAL RECORDS
Agile eMed Portal	London(and above)	Closeout	54,000	6/1/2018	2/7/2019	Requirements & Design, De	AGILE, MEDICAL RECORDS
Fastfile Patient Care	London(and above)	Closeout	73,050	6/1/2018	5/9/2019	Requirements & Design, De	MEDICAL RECORDS
Agile CloudCare PT	London(and above)	Closeout	46,500	6/1/2018	3/18/2019	Requirements & Design, De	AGILE, MEDICAL RECORDS
Medical Records System		Estimation	45,128	10/1/2020	5/1/2021	Requirements & Design, De	AGILE, Fixed Price, MEDIC

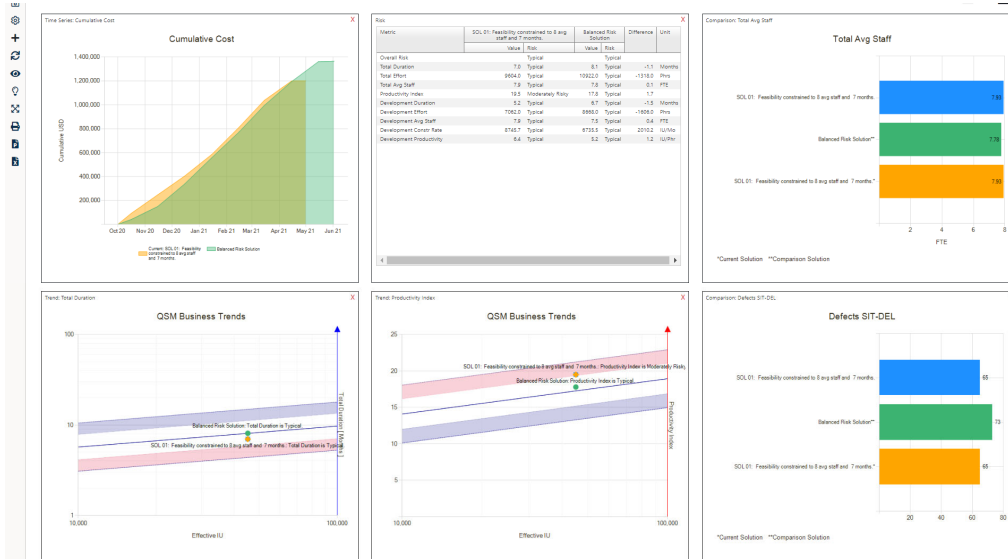
Page size: 50

31 items in 1 pages

QSM<sup>®</sup> The Intelligence behind Successful Software Projects

- SLIM-Collaborate:
- Central database; filter and group & use keywords; drill-down; list specific to user
- Estimation and Closeout projects – have our estimates and history in one place, can pick and choose relevant projects to support an estimate

# SLIM-Collaborate – Portfolio & Project Dashboards



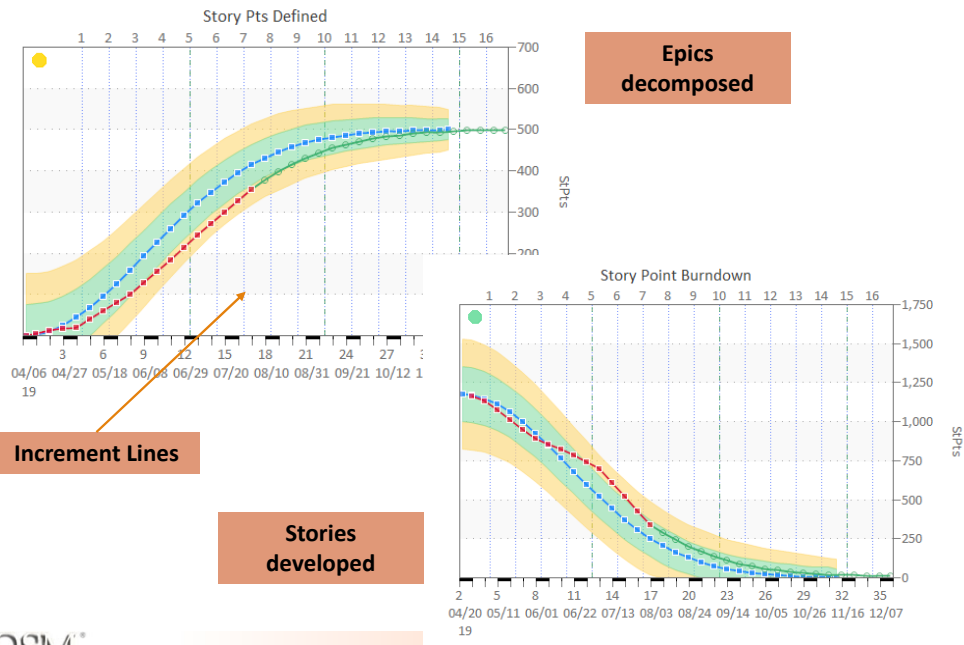
- SLIM-Collaborate specializes in providing project intelligence with 4 different dashboards.
- Portfolio dashboards show metrics for your filtered list – benchmarking performance, get range of sizes, compare to estimates!
- This is the Project dashboard [describe].
- Challenges met – productivity and other assumptions based on history, and account for fluctuations across increments; see the impact of time and effort tradeoffs in real time.

---

# PROJECT TRACKING



## SLIM-Control Agile Template



QSM<sup>®</sup>  
The Intelligence behind  
Successful Software Projects

- This is a typical time series chart used for tracking
- Plan, Actuals, Forecast (talk about in a moment)
- Control Bounds
- Release time Sprints & Program Increments

## Forecasting – An “Estimate” Based on “Actuals”

- Curve Fit Forecast – “try out” various re-planning options and assess their impact on the final cost, schedule, and quality.
- Tradeoff Forecast – assess impact of adding staff.
- Maintenance Forecast – date to meet quality goal.

Curve Fit Results

Metric	Data Pts Used	P3 Time	Goodness of Fit	QSM Weight	User Weight
Phase 3 Milestones	1	21.50	Excellent	25	25
Cum Eff SLOC	7	21.50	Excellent	75	75
Defects Found (Moderate)			Insufficient Data	0	0
Defects Found (Serious)			Insufficient Data	0	0
Defects Found (Critical)			Insufficient Data	0	0
Earned Tasks	12	21.00	Excellent	0	0
Use Cases			Metric Complete	0	0

Metric Weighting Scheme  
 Select one of the options below to weight the individual metric projections in the calculation of the final projected phase 3 time:  
 Automatic (QSM Default Weights)  
 Manual (User Supplied Weights)  
 Apply Weights to Update Forecast

Phase 3 Time Range Sampled  
 QSM Default Range: 12.0 to 27.0 Months  
 Extended Range: 12.0 to 40.5 Months  
 Rerun Forecast using Default Time Range

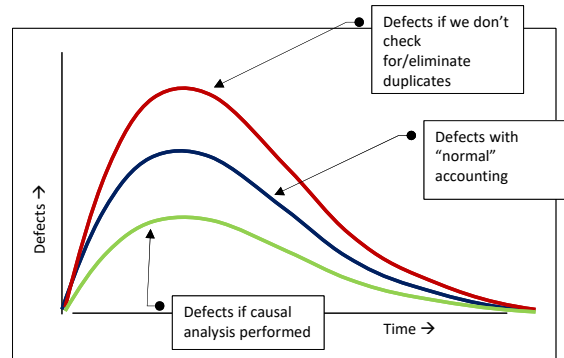
Forecast Results  
 Final Weighted Phase 3 Time: 21.5 Months  
 Life Cycle End Date: 5/21/2016  
 Implied PI: 14.6

Close Help

- When Should You Consider Replanning?
- When a project goes off track, SLIM-Control’s curve fit forecast offers a mechanism for performing “what-if” analysis.
- The curve fit forecast projects the most likely delivery date for the project using your actual progress, updated staffing assumptions, and the final values of metrics used in the forecast.
- You can run several forecasts: one based on continuing with the present plan and others based on re-planning options like deferring non-essential features or negotiating additional time.
- Tradeoff forecasts let you determine the impact of staffing changes
- Maintenance forecasts help you determine if the product quality is good enough to release.
-

## Defect Tuning Calculator

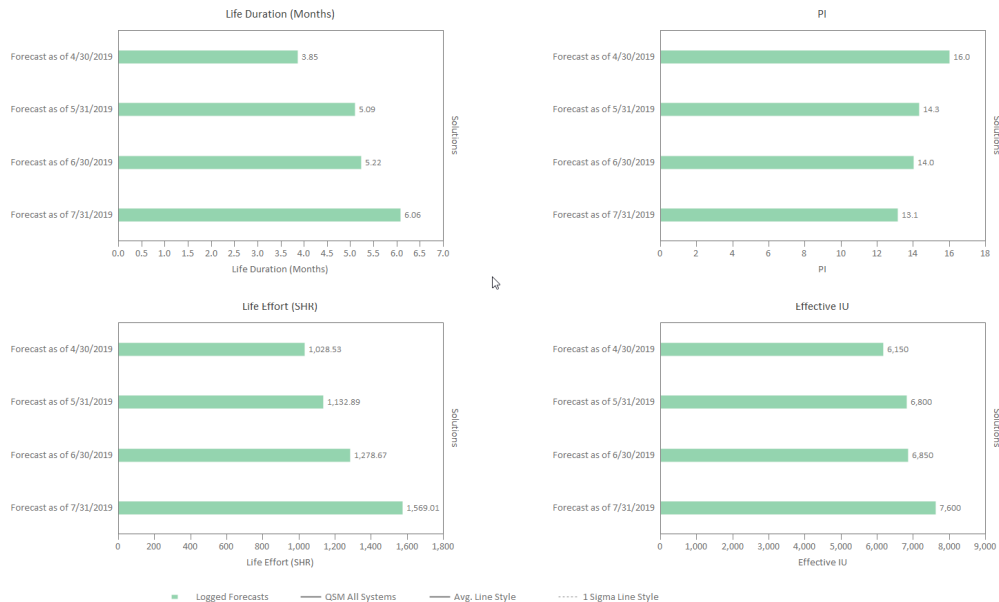
- Tunes defect projections using actual defect data from the project
- Can use all defects or exclude early



Defect tuning factors over 100 are neither good or bad; just different!

- A key metric used to forecast completion is defects. They become more important and influential late in the lifecycle.
- SLIM's defect model can be adjusted to your environment – your practices – based on defect discovery data for completed projects.

## Run Multiple Forecasts



- Here is chart that shows the comparison of multiple forecasts, including size, PI, effort and schedule results based on actual performance and forecast assumptions.
- You can take one of these forecasts and make it the current plan!
- Challenge met – visibility of project status at regular increments, scientific forecast based on multiple metrics with a choice about which metrics are the best predictors and measuring deliverable quality.

---

# PROJECT CLOSEOUT

## SLIM-DataManager & SLIM-Collaborate

- Staffing
- Cum Eff Size
- Defects
- Phase End Dates

### Core Metrics

Phase	Acronym	Plan Start	Plan End	Actual Start	Actual End
1	CD				
2	R&D	1/1/2013	7/5/2013	1/1/2013	7/29/2013
3	C&T	4/23/2013	10/9/2014	4/23/2013	11/5/2014
4	P_Mnt	10/9/2014	1/3/2015	11/5/2014	2/1/2015

- Gathering data for completed projects may seem challenging, but when you stick with 4 or 5 core metrics it isn't.
- When you can, it is important to capture the start and end dates for phases. This image is from SLIM-Control, so if you're using that, you have your history all ready and simply import it into SLIM-DataManager.

## Import SLIM-Control project into SLIM-DataManager

- Status is Completed
- Effort Totals by Phase
- Actual Size
- Defects SIT-DEL
- Defect Tuning Factor
- Actual PI - calculated

The screenshot displays the 'Project ID 1: Demo Agile Project (Record 1 of 1)' window. The 'Basic Information' tab is active, showing fields for Project Name, Status (Completed), Confidence, Preparer Name, Record Creation Date (2/28/2016), Date Last Modified (8/2/2019), and Document Links. The 'Sizing' section includes New (1758), Modified, and Unmodified values, along with Primary Function Unit (Story Points (STPts)) and Gearing Factor (85). A table lists project phases with their respective dates and effort metrics.

Phase	Start Date	End Date	Months	PHR	Peak Staff
1. CD					
2. R&D	1/1/2016	7/1/2016	6.03	8000	
3. C&T	1/15/2016	7/15/2016	6.03	12000	
4. P_Mnt					
Life Cycle	1/1/2016	7/15/2016	6.48	20,000	0

Other sections include 'Requirements', 'Defects' (System Integration to Delivery, First Month after Delivery), and 'Staffing Shape' (Unknown, Rayleigh Front Load, Rayleigh Med Front Load). The bottom right corner shows 'PI = 23.0 MBI = 4.9'.

- When you measure the actual software size and high-level capture time and effort data, PI is calculated.
- Defects discovered during and after delivery are optional, but extremely valuable.
- This data can be pulled into the estimation tools to sanity check estimates against multiple projects and used to create trend lines.

## Capture lessons learned

- **Overruns/Slippages automatically calculated – Plan vs. Actual**
- **% Growth/Reduction of Eff Size**
- **Free-form Significant Factors**

Project ID 5: Web Gateway (Record 5 of 5)

Basic Information | Application | Sizing | Accounting | Custom Metrics | Quality | Review

Overruns/Slippages Enter overrun/slippage data...  
 by % change  by planned value

Select Phase  
CD  
R&D  
C&I  
P\_Mnt

% Difference from Plan  
Time 5 %  
Effort 17 %  
Cost 17 %  
Max peak 17 %

% Growth/Reduction Enter growth/reduction data...  
 by % change  by planned value

% change from Plan  
Eff system size -18 %  
Requirements %

System Benefit/Effectiveness  
Effectiveness Rating

Significant factors  
New...  
Edit...  
Delete

0-Very Negative 5-No Impact 10-Very Positive

Delete First Prior Next Last Add Project OK Cancel Help

- Lastly, you measure your predictability with calculations of estimates vs. actuals and documents project factors, like scope changes, that impact the final outcome.
- Challenges met – more visibility, historical data as basis of estimation, measuring final delivered quality and calculating PI to support future estimates.

---

## QSM Agile Resources

- Search for “agile” on the QSM website .....
  - More than 6 pages listing white papers, blogs, webinars, and other resources
- Recommended key resources
  - Agile Estimation, Beyond the Myths (2 part webinar)
  - QSM Agile Round Table (4 published articles)
    - Big Rock Estimation, Sizing in Story Points, Changing Scope, Diff between PI and Velocity
  - Measuring Effort and Productivity in Agile Projects
  - Estimating Program Increment Capacity – one of several resources for “doing Agile at scale”
  - Measuring Software Size Webinar

---

## Questions?



[www.qsm.com](http://www.qsm.com)

[info@qsm.com](mailto:info@qsm.com)

800-424-6755

PMI PDU claim  
code: