Out with the Old, In with the New?
An Examination of Agile Development Impacts on Traditional Software Cost Estimating Methods

Allison Hawkins
Jenna Meyers

Background/Purpose

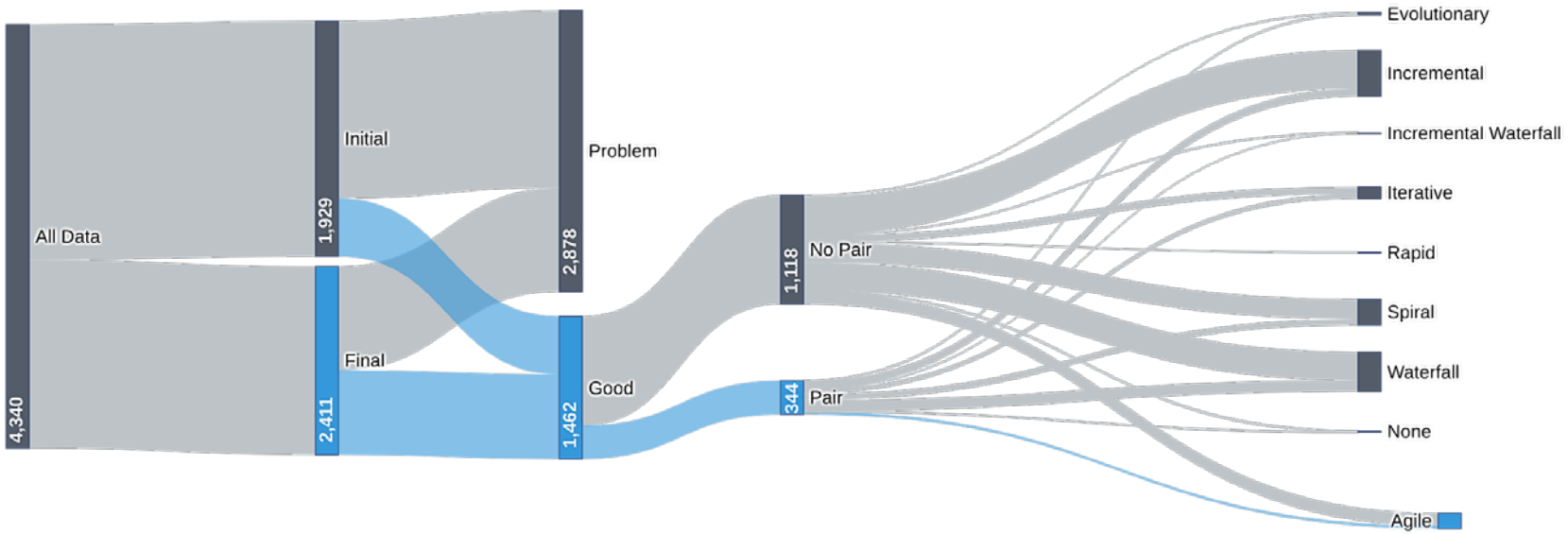
Background

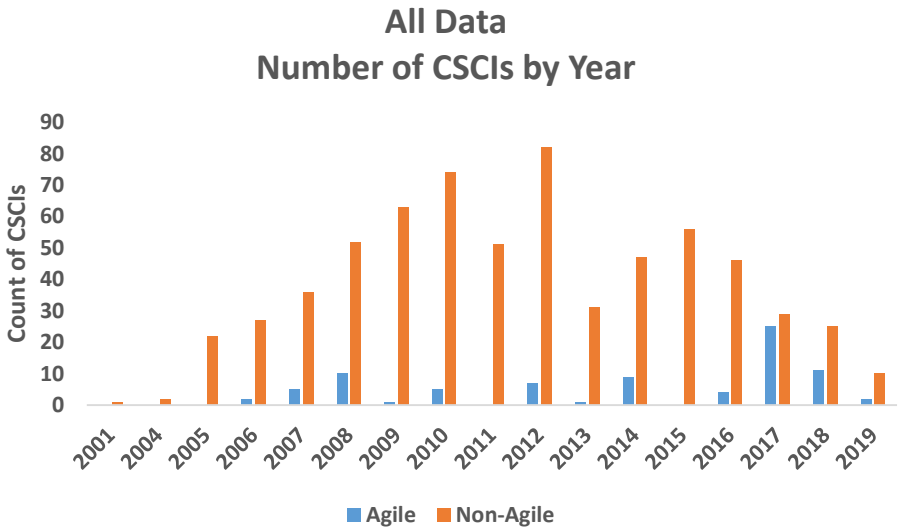
- As agile continues to become more prevalent in parts of the DoD, the cost community has been challenged to quantify the impact of pursuing agile development vs other traditional development processes
- Numerous Army programs have pushed back on providing traditional size measures such as SLOC or RICE-FW on the premise of agile development

Purpose

- ✓ Evaluate agile data points in the SRDR dataset and test traditional CER accuracy
- ✓ Compare the productivity of agile data points in the ISBSG dataset
- ✓ Review an Army case study of estimating a DBS agile development using Function Points

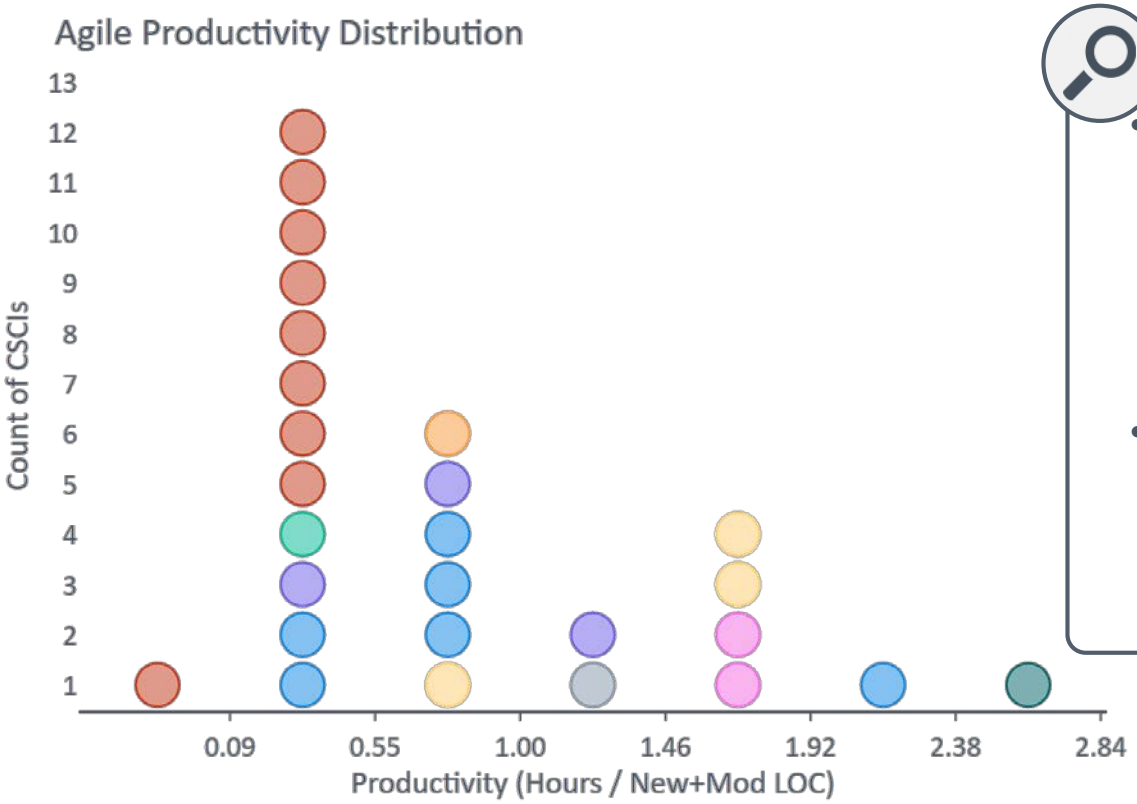
Agile in the SRDR Data Demographics





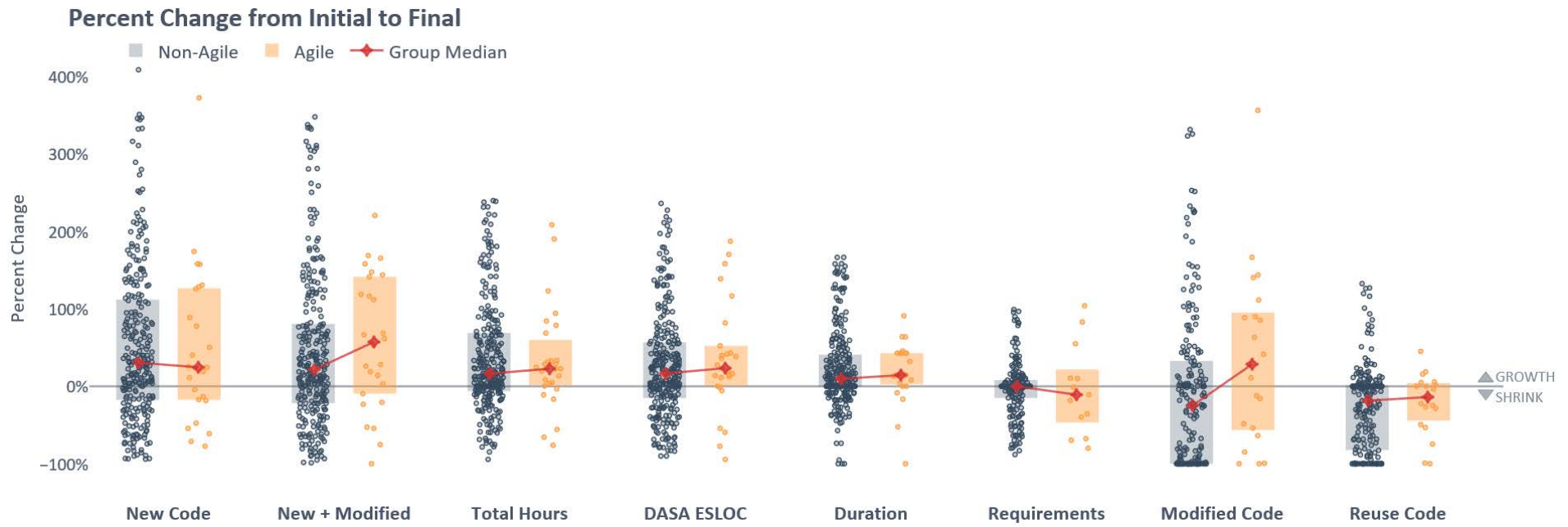
Agile has become more prevalent in the SRDR dataset in recent years, however the data is largely dominated by a few programs

Agile in the SRDR Productivity



- Agile data in the SRDR dataset is dominated by a few programs. In the chart to the left, each circle represents a CSCI which is color-coded by program
- A single program that has multiple CSCIs with the same productivity can influence general observations about agile productivity

Agile in the SRDR
Initial to Final Metrics



One concern when using agile is the utility of the initial metrics. In comparing agile vs non-agile data points in the SRDR dataset, the median percent change for metrics such as new code, total hours, and duration does not seem to differ significantly

Estimating Agile
Do Traditional Methods Work For Agile?

Traditional Software Estimating

Metrics

- Source Lines of Code
- Function Points
- Requirements
- RICE-FW

Methods

- Primary: CERs derived from Initial size vs Final Hours or Final Size vs Final Hours
- Cross-check/Secondary: Commercial Models (SEER, SLIM, True Planning)

Agile Test Cases

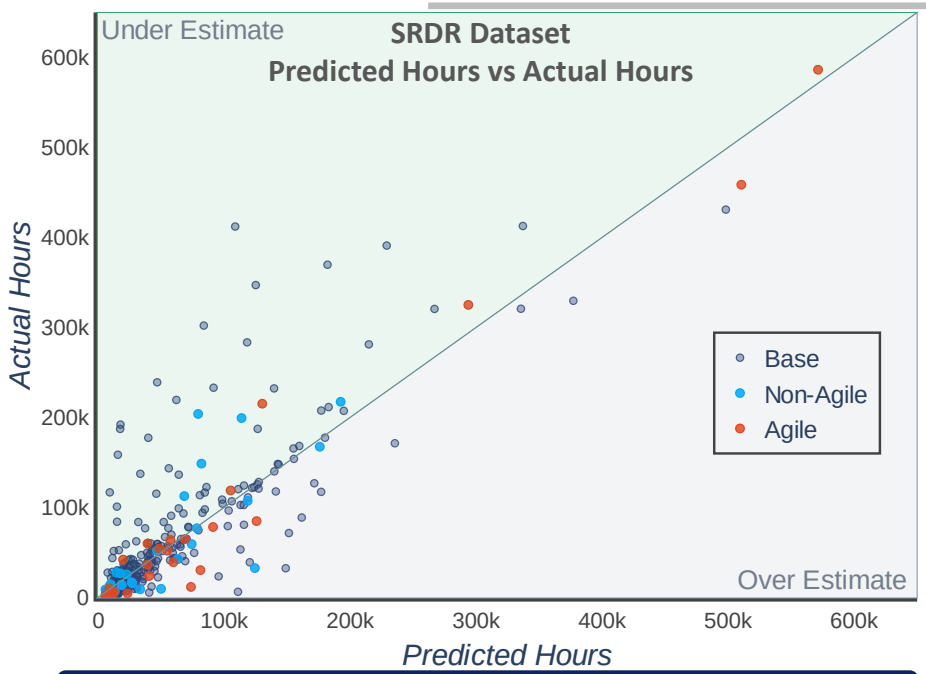
Case 1

- Utilize “Good” SRDR Pairs; withhold agile data points and a test set of non-agile
- Develop a CER using initial new + modified code and initial hours as inputs
- Test CER accuracy using agile data points and test set of non-agile

Case 2

- Utilize SRDR ERP data and withhold agile data points
- Develop a CER using final RICE-FW as input
- Test CER accuracy using agile data points

Estimating Agile
Test Case Results



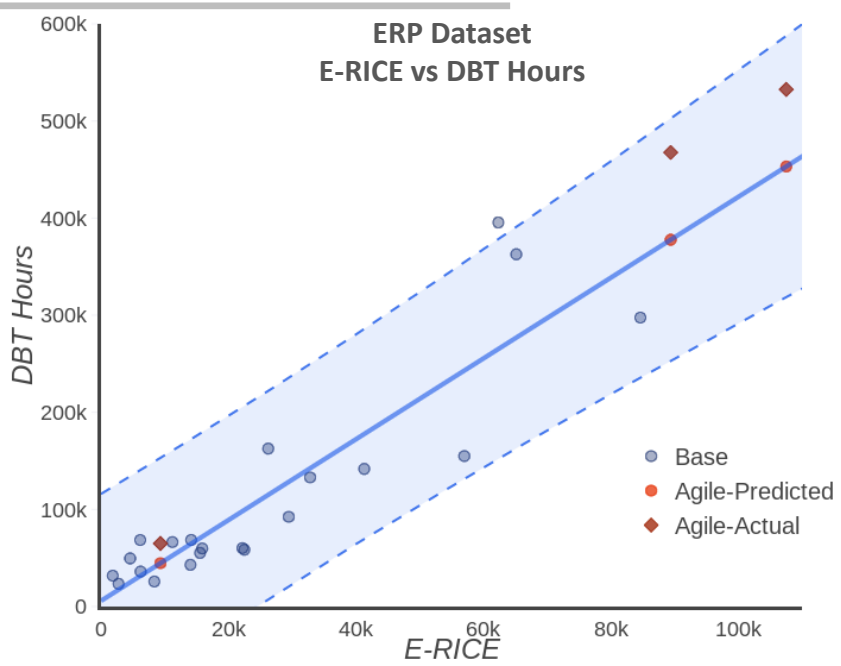
Formula

$$\boldsymbol{Hours} = 0.14\sum(new, mod) + 0.98(initial\_hours) + 4843.12$$

Dataset	Observations
Non-Agile (Held Back)	28
Agile	28

R^2 : 64.6%
CV (MAD) : 78.6%

Dataset	R^2	MAPE	SEE	PRED30
Paired SRDR Base	64.6%	66.03%	47,674	76.84%
Non-Agile (Held Back)	66.76%	48.52%	38,764	67.86%
Agile	96.05%	42.94%	27,982	89.29%



Formula $\boldsymbol{Hours} = 4.1593(erice) + 5855.59$

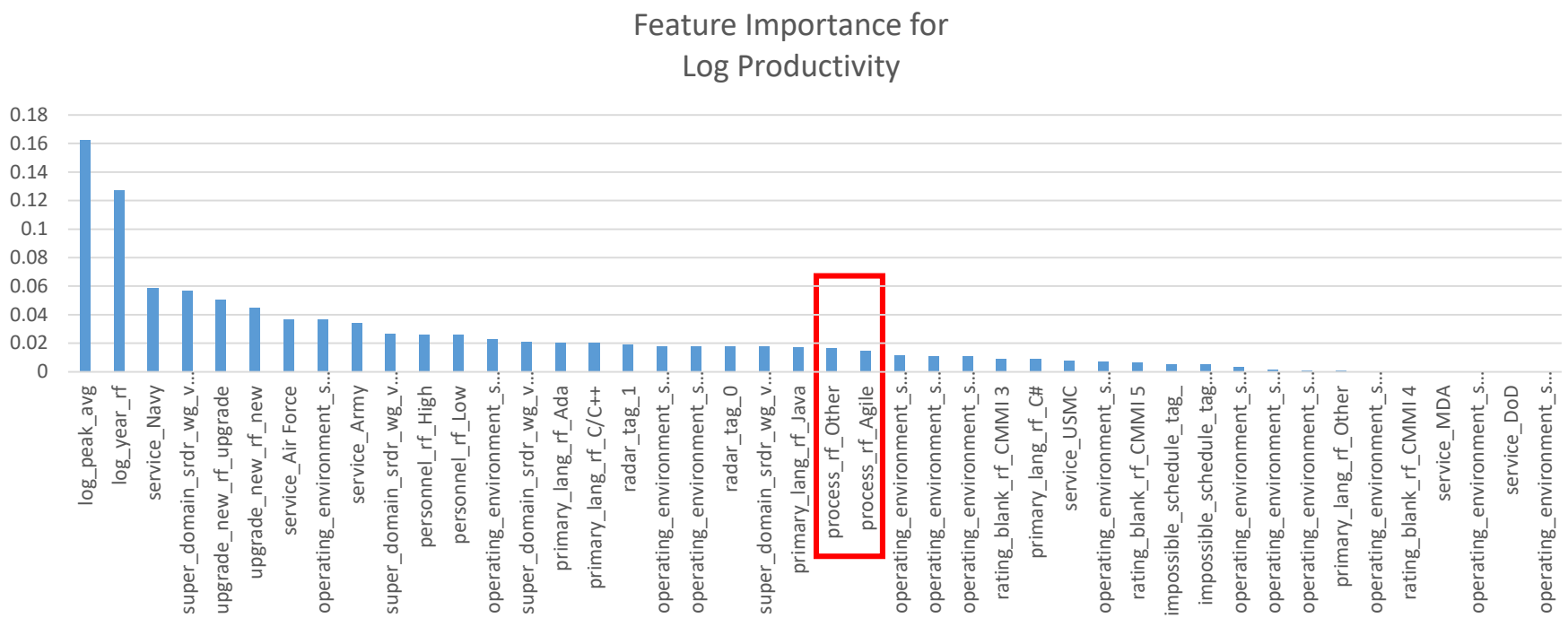
Dataset	Observations
	3

R^2 : 80.0%
CV (MAD) : 41.80%

Conclusion

While the set of agile data points is small as compared to the SRDR dataset, DASA-CE's initial results suggest that agile programs can be effectively estimated using traditional CERs

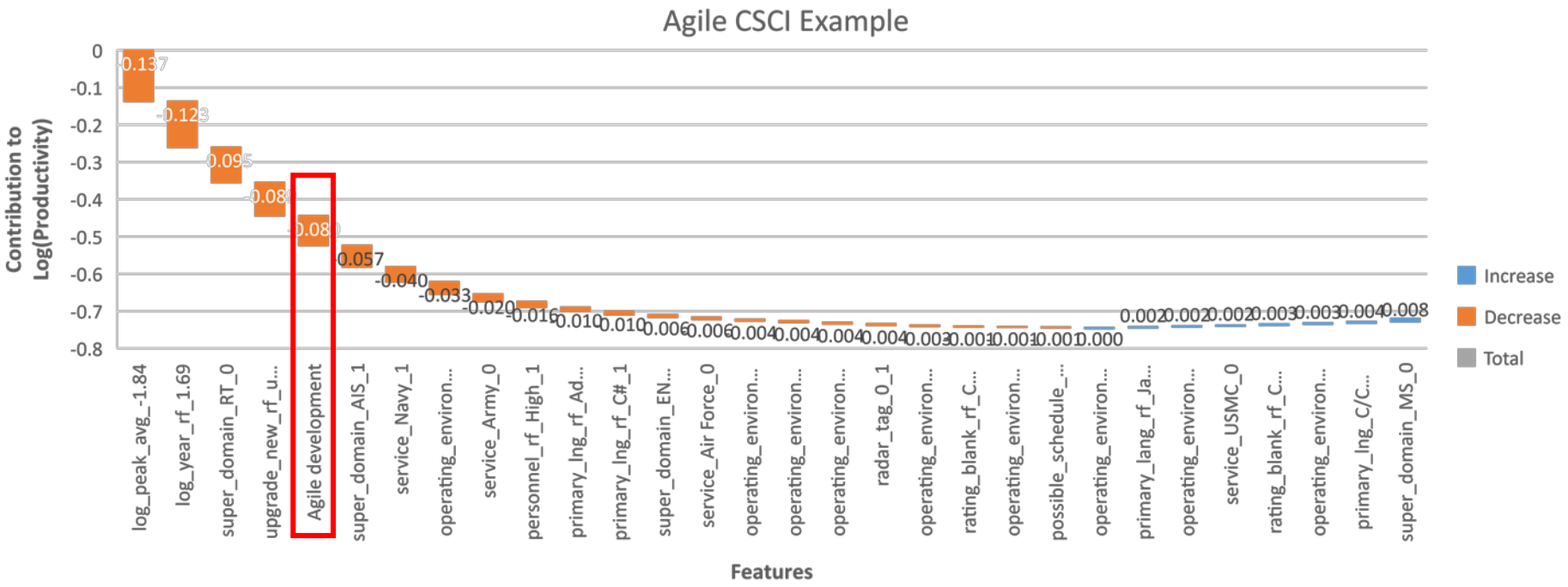
SRDR Productivity
Feature Importance



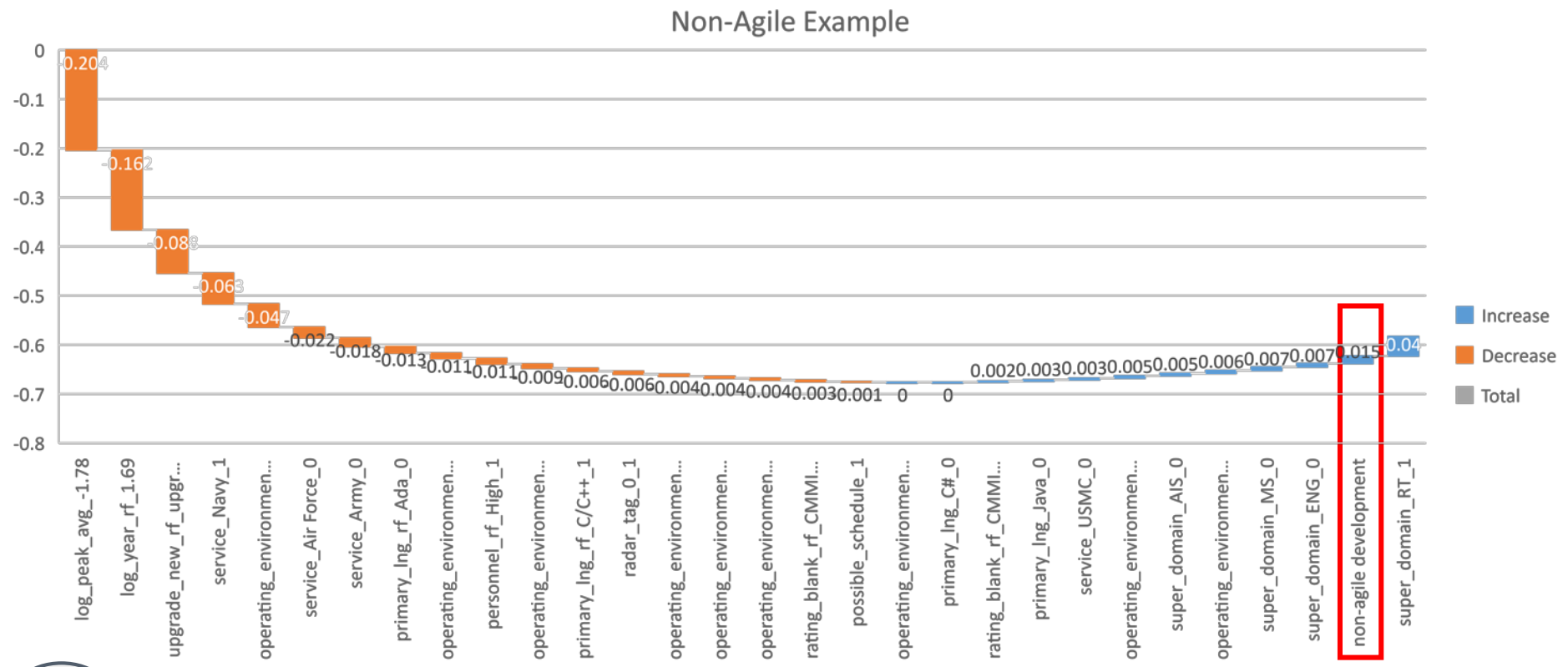
A Random Forest model was created to estimate log productivity and to provide insight into feature importance, i.e., how useful each independent variable is at predicting the dependent variable. Overall, development methodology (agile vs non-agile) did not rank as a significant feature which contributes to productivity.

SRDR Productivity Feature Importance
Agile CSCI Example

The significance of SW development methodology on log productivity was further investigated by examining an agile and non-agile data point in the SRDR dataset. In the particular agile example, SW development methodology does show to increase productivity.



SRDR Productivity Feature Importance
Non-Agile CSCI Example

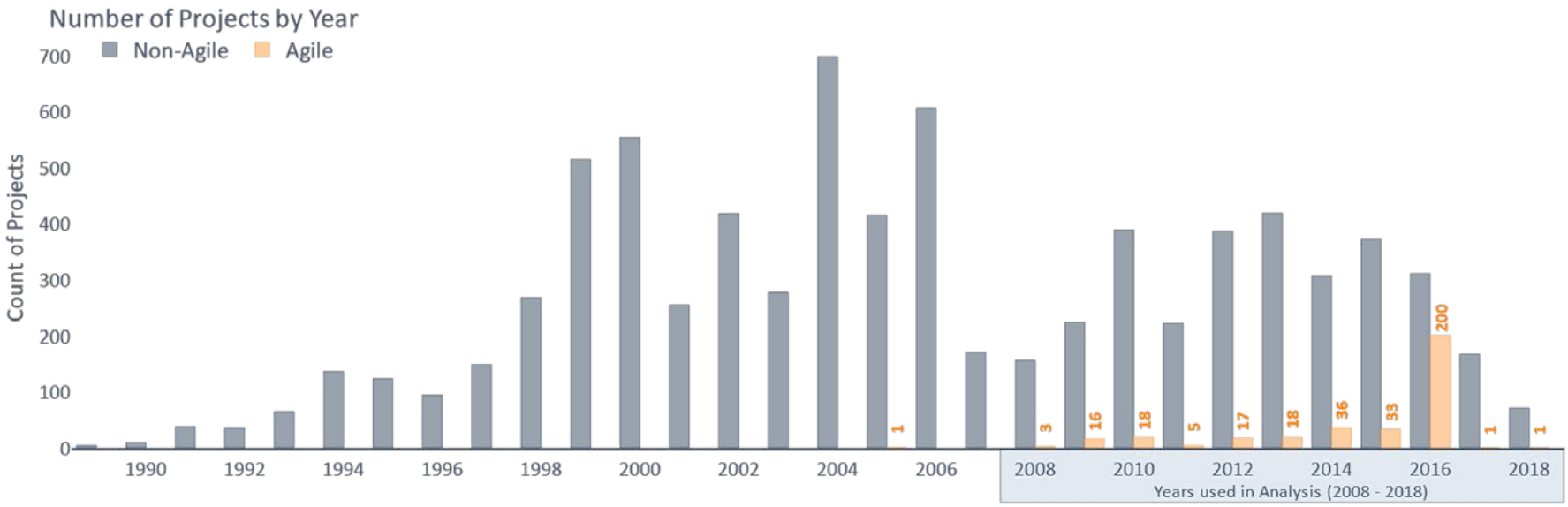


In the particular non-agile example, SW development methodology does show to decrease productivity.

Although the overall importance of development method did not come out high, in select agile vs non-agile data-points, it does have an impact.

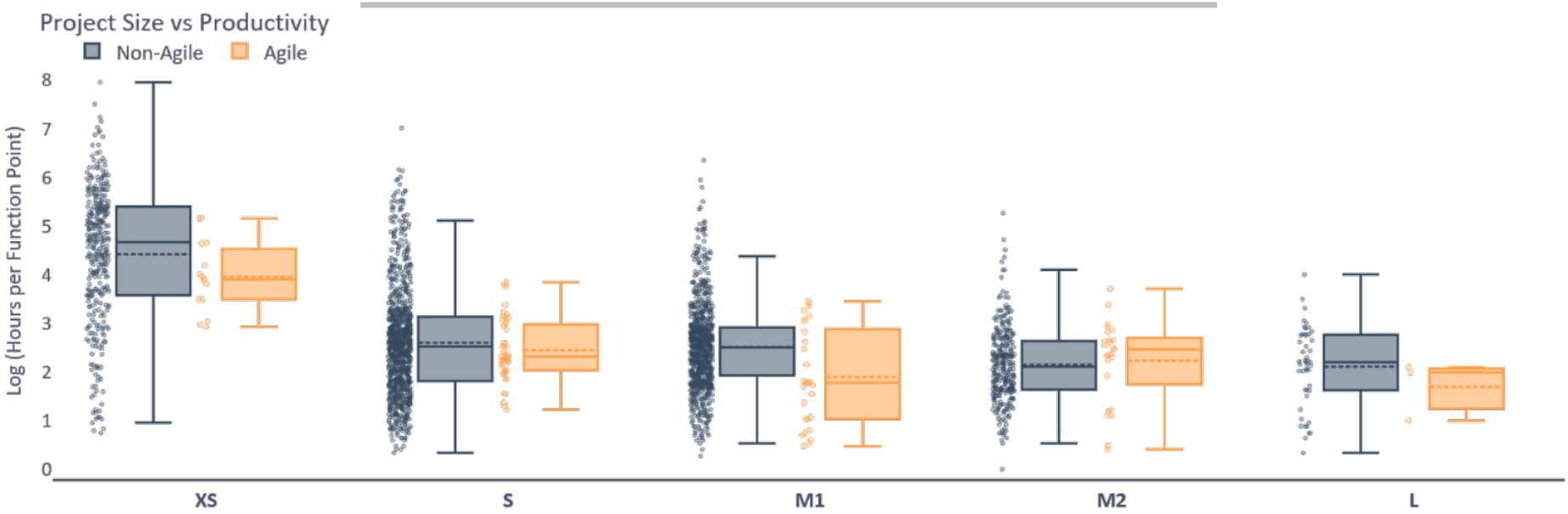
ISBSG
Demographics

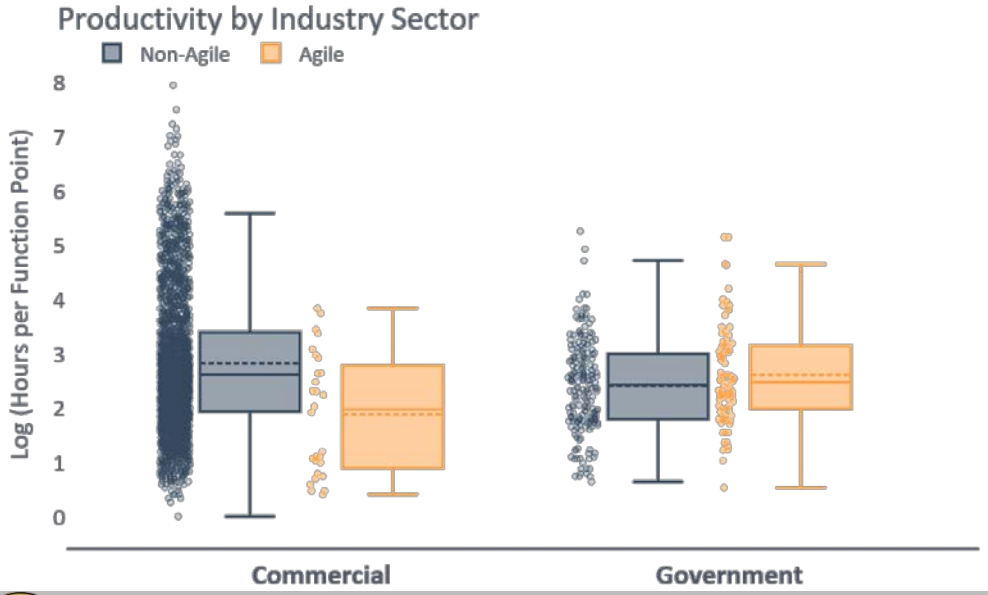
- The International Software Benchmarking Standards Group (ISBSG) dataset is a large dataset of projects (approximately 9,178 projects) from both the commercial and government sector
  - The majority of the projects within the dataset use Function Points as the measure of size
- In analyzing the ISBSG data, the following filters were applied:
  - Development Type: Development/Enhancement
  - Data quality rating: A/B
  - Year: 2008+
  - Counting Approach: IFPUG, NESMA, MARK II



ISBSG

Productivity in Agile vs Non-Agile Projects

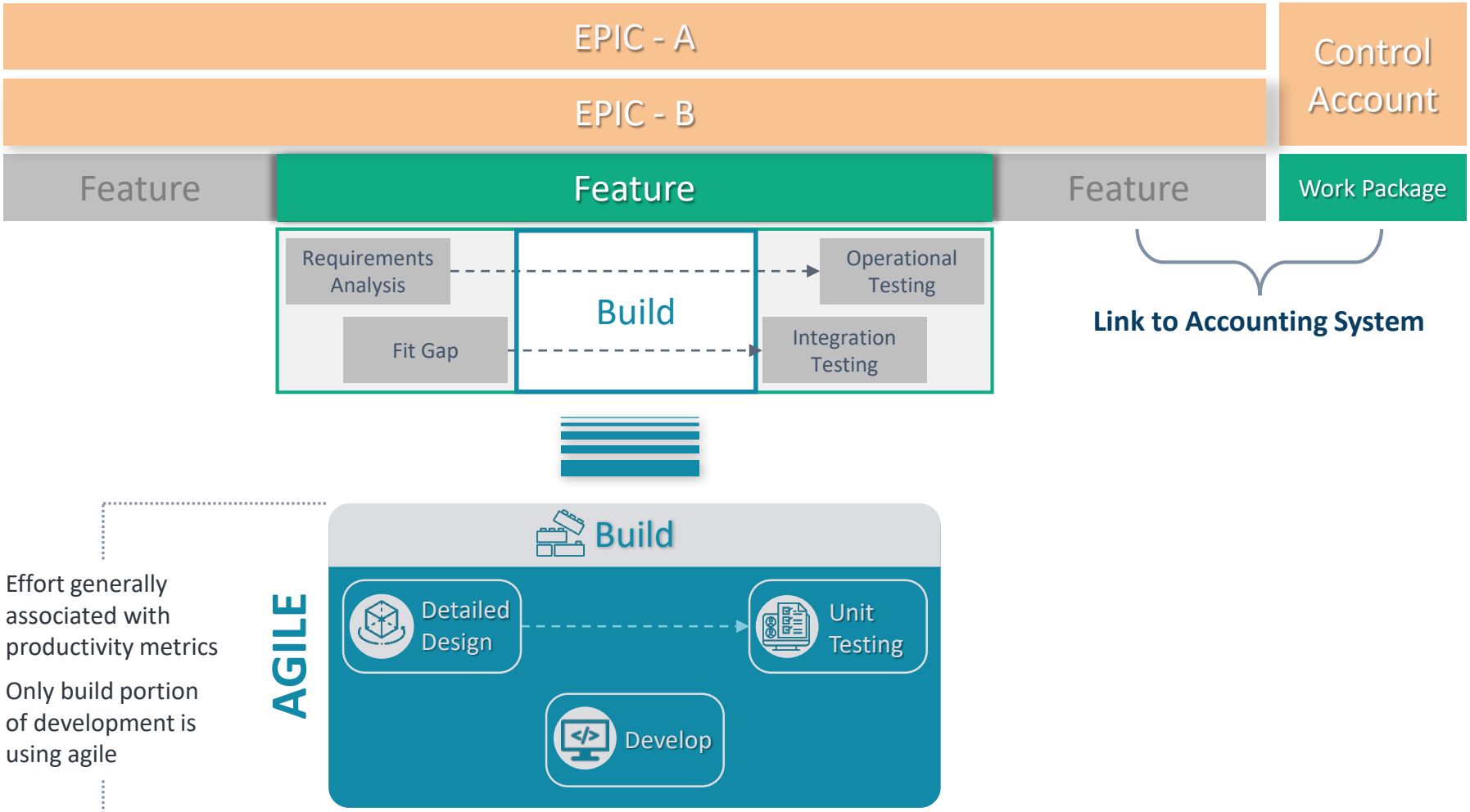




Agile projects appear to have slightly better productivity on average as compared to non-agile projects when the ISBSG data set is grouped by project size. However, when the data set is broken out by Government vs Commercial, the Government agile projects do not exhibit better productivity

Agile Development and EVM

Epics are a set of Features that define a 'to be' (sub- business process) within the COTS products



- Effort generally associated with productivity metrics
- Only build portion of development is using agile

Agile Development and the SRDR-ERP Form

Snapshot of Initial SRDR

Release Map 3.3.5.3.1		Planned and Achieved Development (by Feature per Epic) 3.3.5.3.2					
Epic/ Capability Identifier	Feature Identifier	Planned Stories per Feature by Epic	Actual Stories	Planned Story Points per Feature by Epic	Actual Story Points	Planned Hours per Feature by Epic	Actual Feature Hours
EPIC A	Feature X	*NA*	*NA*	*27*		*136*	
EPIC A	Feature Y	*NA*	*NA*	*87*		*432*	
EPIC A	Feature Z	*NA*	*NA*	*39*		*126*	
EPIC B	Feature R	*NA*	*NA*	*19*		*201*	
EPIC B	Feature P	*NA*	*NA*	*76*		*99*	

- Initially Program A only reported features and story points (instead of RICE-FW) in the SRDR which made it very difficult to use historical DBS data to estimate cost
- In support of a DASA-CE cost review, the vendor was able to map each feature into RICE-FW which will now be completed in subsequent SRDRs for Program A

Agent Dashboard → Med Complexity Extension

Data Reconciliation → High Complexity Report

Allows for understanding of how capabilities tie to RICE-FW and cost (since features are tracked as WPs)

Agile DBS Case Study Using Function Points

Overview

Using Function Points for Agile Development

Background:

- Typically, DASA-CE uses total object count or a RICE-FW object breakout to estimate DBS development
- All of Program A's release sizes were initially sized by the LSI using agile metrics. While two of the program's releases were also sized in RICE-FW using the defined features (R2 and R3), the last release (R4) was sized with a large number of configurations since detailed blueprinting had not yet been completed. DASA-CE was skeptical of the high configuration count and obtained a Function Point count on all releases to have an alternative sizing metric

Process:

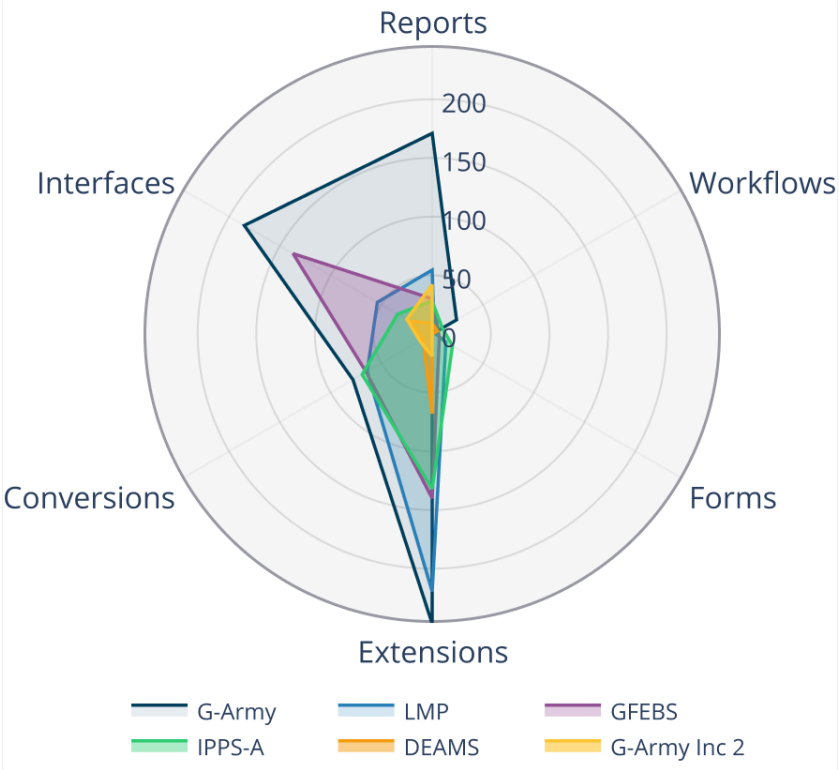
- Since the SRDR dataset does not contain Function Point data and DASA-CE does not have an internal FP dataset, DASA-CE explored three potential methods for estimating using FPs:

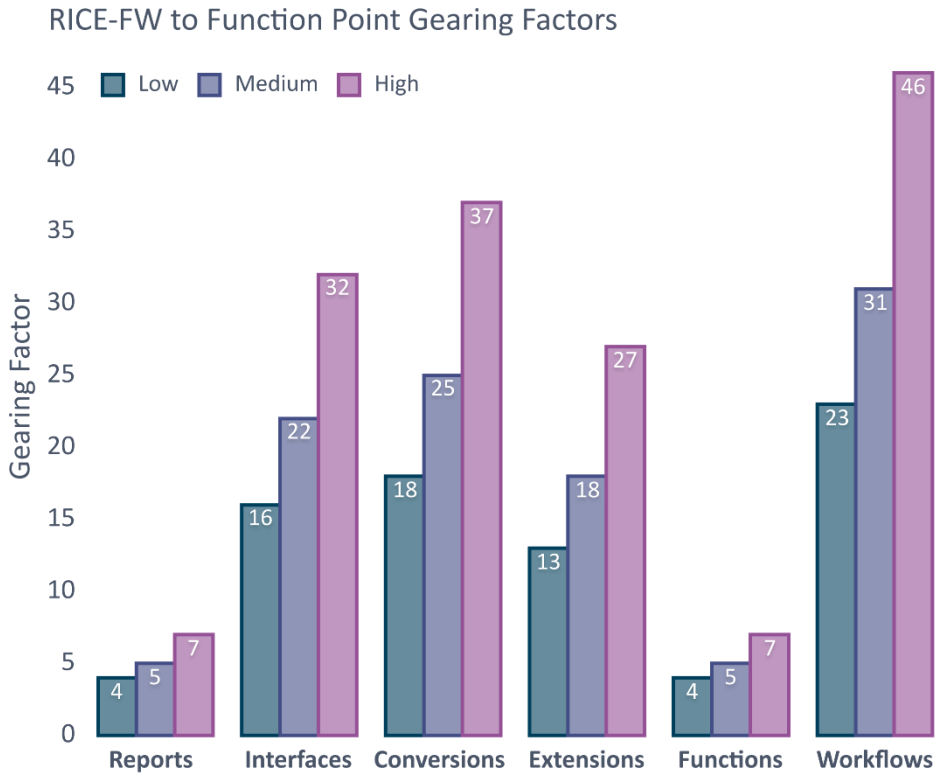
  1) Backfire to FP from ERP RICE-FW data and regress FP to Hours

  2) Use of Parametric Model such as SLIM, True Planning, or SEER (FP as direct input)

  3) Use of ISBSGs dataset (FP as direct input)

Function Point Use Cases
(1 of 3)



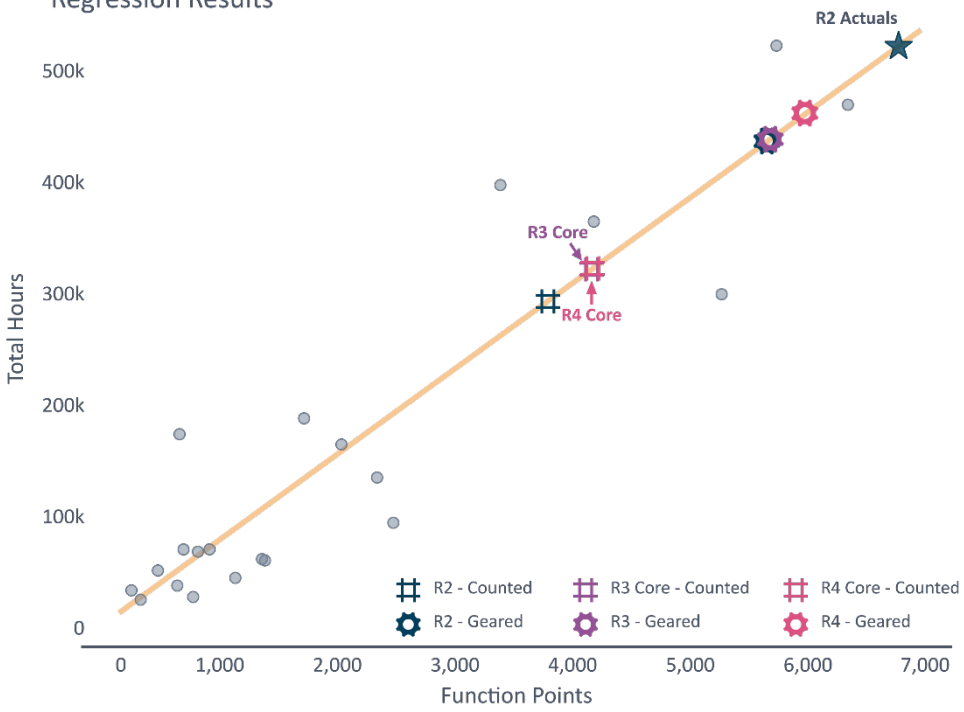


- The chart above shows the composition of the aggregated DBS data that DASA-CE typically uses to estimate development
- Extensions, Reports, and Interfaces are the most common objects in the release level data

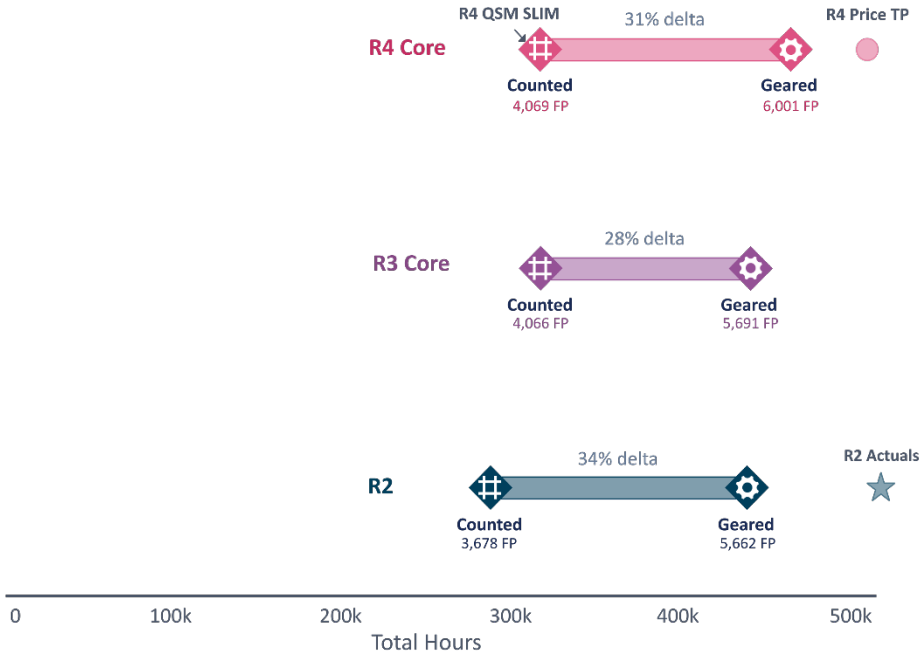
- Since the parametric model weighting factors for RICE-FW closely align to the DASA-CE ERICE coefficients, gearing factors from SEER were used to backfire the RICE-FW data into FP counts
- Other gearing factors from True Planning (PRICE) were also analyzed in a sensitivity analysis

Function Point Use Cases
(2 of 3)

Regression Results



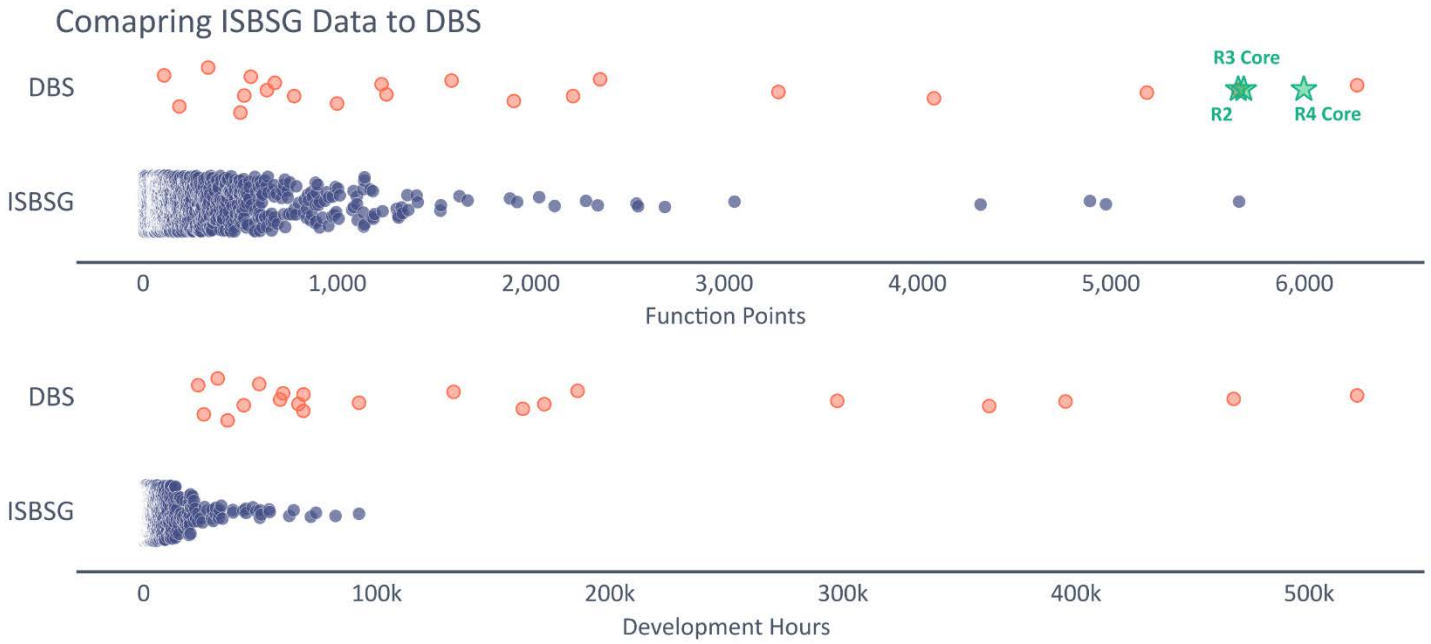
Estimate Comparisons



- The above chart depicts the historical releases geared to FPs plotted against the development hours (design, build, and test only)
- The other points on the line show the results for Program A associated with counted FP and geared FP
- Since Program A R2 development was complete, the actual for R2 is depicted

- Parametric models (SLIM, True Planning, and SEER) were each calibrated using the historical DBS data. The calibrated models were then used to estimate Program A's releases using the counted FP as input
- There was significant variability in the regression output using the counted vs geared FPs. There was also significant variability in the parametric model output for R4

Function Point Use Cases
(3 of 3)



- Due to DASA-CE's lack of historical FP data, DASA-CE attempted to use the ISBSG data to estimate Program A's releases as an additional cross check. The ISBSG dataset contains projects from both Defense and Commercial sectors sized in Function Points
- As shown in the above chart, the projects within the ISBSGs dataset are very small when compared to Program A's releases
- When using a simple productivity metric derived from the ISBSGs data, the resulting development hours for Program A R2 were significantly under-estimated when compared to the actuals

Conclusion/Next Steps

Conclusion

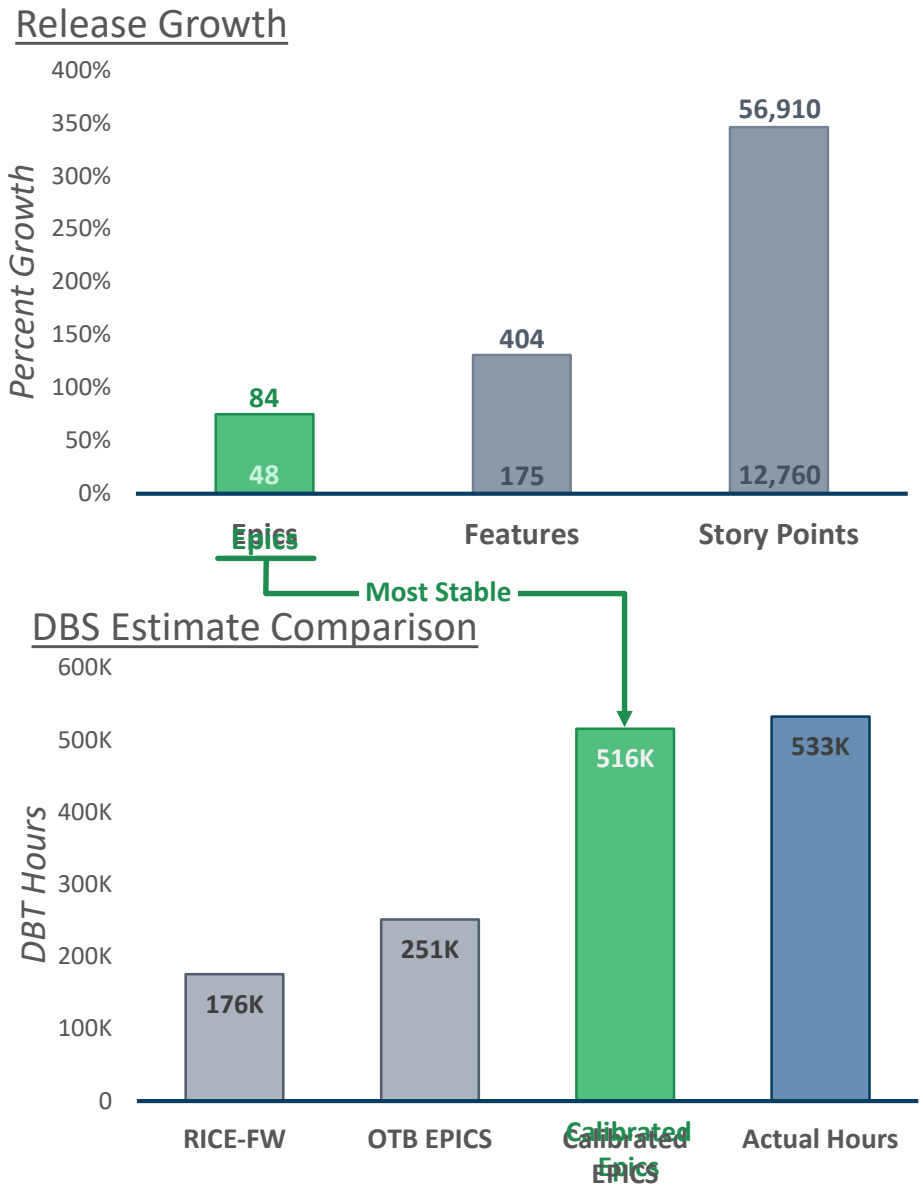
- Initial analysis suggests that traditional data and CERs can still be utilized to estimate agile data points
  - Agile programs may need an alternative way for initially sizing the release
- Caution should be used when utilizing the SRDR dataset to draw conclusions on the agile development process
  - Agile data is dominated by few programs
  - Conversion factors utilized for normalizing to logical code or calculating ESLOC may skew productivity conclusions
- A lack of function point data hinders DASA-CE's ability to effectively use Function Points as an alternative sizing method

Next Steps

- Continue to investigate the importance of agile in predicting development effort/productivity using advanced techniques (i.e. machine learning)
- Evaluate the sensitivity of conclusions based on factors such as the grouping of agile vs non-agile and the factors used to normalize code counts
- Evaluate utility of Function Points for estimating software development for a weapon system

Backup

Estimating Agile
Agile Metrics using QSM SLIM



- Arguments are often made for the use of metrics such as story points/hour or velocity to estimate agile software development effort
- One of the major challenges for estimating agile using cross-program data is inconsistent terminology. Epics, features, and stories may be used to represent different units of work across programs
- The chart on the left depicts Program A's release growth using Agile metrics. Epics are more stable as they tie directly to the program's requirements, while story points grew significantly

- Due to a lack of agile metric data in the DoD, the commercial model SLIM was used to estimate the program's release effort using both final RICE-FW and final Epics to evaluate the model's accuracy using each metric

Conclusion

Using "out-of-the-box" Epics resulted in a more accurate estimate as compared to traditional RICE-FW. Adjusted Epics, derived from the program-specific relationship between Features and Epics (calibrated Epics), was even more accurate