3-2021

An Analysis of Application Type, Super Domain, and Productivity in Software Intensive Defense Acquisition

Evan P. Amato

AN ANALYSIS OF APPLICATION TYPE, SUPER DOMAIN, AND PRODUCTIVITY IN SOFTWARE INTENSIVE DEFENSE ACQUISITIONS

THESIS

March 2021

Evan P. Amato, 1st Lieutenant, USAF

AFIT-ENV-MS-21-M-202

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENV-MS-21-M-202

AN ANALYSIS OF APPLICATION TYPE, SUPER DOMAIN, AND PRODUCTIVITY IN SOFTWARE INTENSIVE DEFENSE ACQUISITIONS

THESIS

Presented to the Faculty

Department of Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Cost Analysis

Evan P. Amato, BS

1st Lieutenant, USAF

March 2021

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENV-MS-21-M-202

AN ANALYSIS OF APPLICATION TYPE, SUPER DOMAIN, AND PRODUCTIVITY IN SOFTWARE INTENSIVE DEFENSE ACQUISITIONS

Evan P. Amato, 1st Lieutenant, USAF

Committee Membership:

Lieutenant Colonel Scott T. Drylie, PhD
Chair

Dr. Jonathan D. Ritschel, PhD
Member

Dr. Edward D. White, PhD
Member

AFIT-ENV-MS-21-M-202

Abstract

The productivity of software developers has become an area of interest in the private industry in an endeavor to more appropriately interpret pertinent software development cost drivers. In an effort to better predict the cost of developing software, many have focused on Application Type as an important factor. The Department of Defense (DoD) has recently adopted a similar focus. Studies on categorization schemes of Application Types have shown increasingly relevant cost and productivity driving characteristics recently in the DoD. Identification of factors associated with distinct productivity effects is important for the defense acquisition and software cost estimation domains. Distinct factors can provide insight into what drives software productivity and cost. This research attempts to investigate the significance of Application Type and Super Domain in predicting productivity in software intensive defense programs. This current study analyzed 655 Software Resource Data Reports of DoD projects spanning the years 2001 to 2019. The analysis indicates the significance of Application Type in predicting productivity to be overstated for DoD. Only two of seventeen Application Types adopted by the DoD and one of four larger environmental settings called "Super Domains" proved significant. Regarding those, this study was able to identify characteristics that may prove more useful for understanding drivers of software cost and productivity.

To my wife.

Thank you for your undying love and support through this arduous effort.

Acknowledgments

I would like to express my sincere appreciation to my research advisor, Lt Col Scott Drylie, for his invaluable guidance and knowledge throughout the research and writing process. I would also like to thank Dr. Jonathan Ritschel and Dr. Edward White for their extensive support.

Evan P. Amato

Table of Contents

List of Figures

List of Tables

Page

List of Equations

Page

AN ANALYSIS OF APPLICATION TYPE, SUPER DOMAIN, AND PRODUCTIVITY IN SOFTWARE INTENSIVE DEFENSE ACQUISITIONS

I. Introduction

Background

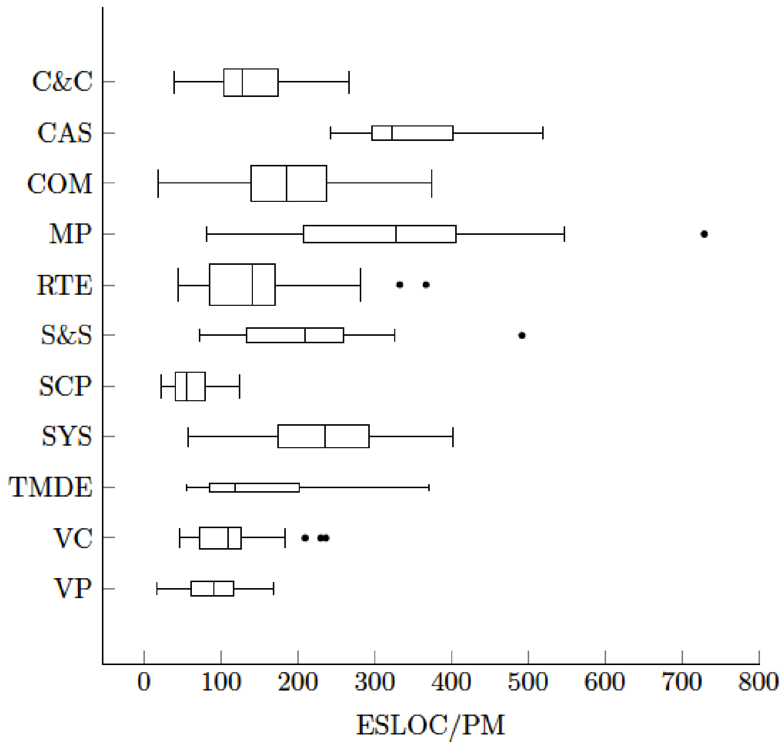
Software is an integral part of weapon system acquisitions and the national security enterprise. Software drives program risk on roughly 60% of new defense acquisition programs and is often evaluated as a critical challenge (Defense Science Board, 2018). The DoD is behind the private industry in current software development practices (Defense Acquisition University, 2019). To address this problem, the DoD is implementing a software development venture called the DoD Enterprise DevSecOps Initiative. DevSecOps is an agile-based software engineering practice that encompasses software development (Dev), security (Sec), and operations (Ops) (Department of Defense, Chief Information Officer, 2019). DevSecOps focuses on keeping information technology, IT, security as a forefront objective during every part of the software development process by automating core security tasks and controls from the beginning of the lifecycle rather than adding them at the end of the process. It is believed that modern DevSecOps development practices will enable different results: rapid, constant, and secure application delivery. Thus, how the DoD tracks and operates in the software development realm is immensely important to reach the desired productivity of the private industry.

While the DoD may achieve the desired benefits of shifting towards the DevSecOps paradigm, the benefits themselves require validation. One way to measure benefits is through software productivity. Software productivity is often considered one of the most important

cost-related aspects of software development. Productivity is commonly defined as the ratio between the amount of product produced and the number of resources used to generate it (Morasca & Russo, 2001). Understanding productivity is important as it establishes a connection between technical and economic concerns (Durate C. H., 2019). Software productivity has become a key focus area in the private industry in an endeavor to more appropriately interpret pertinent software development cost drivers. Attempts to model productivity have driven the need for better ways to track productivity in software development processes. Recognizing characteristics associated with higher levels of productivity is valuable to discerning the path forward for the DoD.

The DoD has taken a particular interest in the study of Application Type, and productivity. In 2015, Clark and Madachy, published the Software Cost Estimation Metrics Manual, preposing 11 software productivity Cost Estimating Relationships (CERs), and 17 different Application Types (Clark & Madachy, 2015). The Software Cost Estimation Metrics Manual focused on 317 Computer Software Configuration Items (CSCIs) from the Software Resource Data Reports database ranging from 2004 to 2013. CERs and Application Types are currently being used by the DoD, and Application Type has become part of SRDR database standard tracking. This study is motivated by the CERs and productivity benchmarks outlined in the Software Cost Estimation Metrics Manual. The Software Cost Estimation Metrics Manual presented a boxplot distribution of productivity for each Application Type studied. This boxplot can be found in Figure 1.

Figure 1: Application Type Productivity Boxplot (Clark & Madachy, 2015)



While Application Types are proposed to be important to software cost estimation and productivity in the Software Cost Estimation Metrics Manual, it appears to lack empirical evidence that affirms Application Types are distinctly different from one another without convergence. Many of the Application Type productivity means displayed in Figure 1 appear to be homogenous. Application Types VP, VC, TMDE, RTE, and COM appear to have nearly indistinguishable productivity mean values. Empirical testing such as an Analysis of Variance (ANOVA) must be employed to analyze the differences among Application Types to prove a distinction between productivity mean values. This empirical analysis was left out of the Software Cost Estimation Metrics Manual and led to the idea that the developed CERs for productivity from the Software Cost Estimation Metrics Manual would not be statistically

distinct from one another. If the CERs for productivity are not statistically distinct, the derived CERs may hold as less viable as cost predictors than initially thought. This study mainly focuses on empirically analyzing the significance of using Application Type as a predictor of productivity, which is a proxy for software development cost. This study also looks at Super Domain as a predictor of productivity. While Super Domain was not in the scope of the Software Cost Estimation Metrics Manual analysis, it has also become an area of interest parallel with Application Type having to become part of SRDR database standard mapping.

Problem Statement

Empirical analysis and research specific to defense productivity within Super Domain and Application Type appears to lack empirical statistical evidence that these groupings are distinctly different from each other. The current notion of Application Type being predictive of software productivity in the Software Cost Estimation Metrics Manual (2015) and the associated CERs, may be premature. Clark and Madachy have set a baseline by having done the important foundational work of identifying plausible categories of Application Types. Clark and Madachy suggest these categories *should* make a difference in productivity, and they appear to be well situated within the literature on the subject. Thus, their initial CERs and Application Types emerge as inherently viable. This research employs the methods used by Clark and Madachy to assess the Application Types and CERs adopted by the Software Cost Estimation Metrics Manual. This research intends to build upon Clark and Madachy's findings and inspects for relationships between Application Types, Super Domains, and characteristic predictors of both. The primary objective of this study is to empirically test

Application Type and Super Domain's significance in predicting productivity and highlight significant characteristics of distinctly productive Application Types and Super Domains.

Research Objective and Questions

The objective of this research is to analyze the significance of Application Type and Super Domain in predicting productivity in software intensive defense acquisition programs. Application Type and productivity have been asserted to be of considerable importance in the Software Cost Estimating Metric Manual for Defense Systems (2015). The Software Cost Estimating Metric Manual for Defense Systems (2015) provides several CERs, organized by Application Type, to determine the relationship between equivalent source lines of code (ESLOC) and manpower effort (Clark & Madachy, 2015). The relationship signifies what "productivity" should be expected from coders for a given kind of project. While an important tool for cost estimators, the publication stops short of identifying which conditions may underlie productivity – rather it implies that the category of the project is the driver. The current study seeks, therefore, to identify which conditions, and characteristics of software intensive acquisition projects may correspond to the distinctly productive programs.

This research is focused on addressing the following major research questions:

1. How predictive of productivity are Application Types in defense software intensive programs?
2. How predictive of productivity are Super Domains in defense software intensive programs?
3. Among Application Types and Super Domains that show statistically distinct productivity effects, which characteristics are statistically significant?

Scope and Limitations

The primary data source for this research is derived from the Defense Automated Cost Information Management System (DACIMS) using the Cost Assessment Data Enterprise (CADE). From DACIMS, Software Resource Data Reports (SRDR) are analyzed to analyze historical productivity data in DoD software intensive acquisition programs. Limitations include the availability and accuracy of the SRDR data in the CADE repository.

Summary

This research addresses Application Type and its impact on productivity in defense software intensive acquisition programs. Chapter II explores the literature on software productivity, and program critical success factors. Chapter III explains the data source and methodology for this research. Chapter IV contains the results and analyses of this research. Chapter V discusses the practical implications of the research and presents potential considerations for future research.

II. Literature Review

Overview

The purpose of this chapter is to provide a recapitulation of peer-reviewed literature on software productivity and its scope. This literature review focuses on four key areas: the purpose of productivity in software cost estimating, the different available software productivity measures, the perceived influences of productivity, and the relevance of Super Domain and Application Type in software development. This chapter provides the foundation upon which subsequent research and chapters will be built.

Software Productivity Introduction

Software productivity is often considered one of the most important aspects of software development. It connects the engineering and production of software to cost (Durate C. , 2019). It is a foundational way to measure the quality of a production process and allows leaders a way to evaluate the effort needed for a project (Morasca & Russo, An Empirical Study of Software Productivity, 2001). Software productivity has been a major area of study in both industry and the DoD for the past 60 years to better understand and quantify the drivers of cost in the software development process.

Attempts to model productivity have driven the need for better ways to measure productivity in software development processes. However, measuring and tracking software productivity is difficult due to the complex interaction between productivity factors (Maxwell, Van Wassenhove, & Dutta, 1996). Thus, the software engineering industry has struggled with developing a proper way to consistently and accurately measure productivity and the factors that influence it. Previous software productivity research has been split into

two major categories: determining which factors have a significant impact on software productivity, and the best way to measure software productivity (Maxwell, Van Wassenhove, & Dutta, 1996). Both areas of productivity research and their relevance will be discussed in-depth in subsequent sections.

Software Productivity Factors

Managing software productivity is essential in software development projects in the DoD and civilian industry. Many factors influence productivity output (Morasca & Russo, An Empirical Study of Software Productivity, 2001). Due to software development being primarily knowledge work, productivity factors can be sorted into two primary lanes: technical factors and soft factors (Wagner & Ruhe, 2018). The current study will consider factors from each. It is important to note the sources used to compile the subsequent list of factors are predominantly from surveys of the literature and meta-analyses.

Technical factors encompass the mechanical and tangible aspects influencing software development process productivity (Oliveira, Conte, Cristo, & Valentim, 2013). Technical factors primarily are further subdivided into product, process, and development environment factors (Wagner & Ruhe, 2018; Sadowski & Zimmerman, 2019; Trendowicz & Münch, 2009). Product factors are all the factors that are in direct relation to the software product itself (Wagner & Ruhe, 2018). These product factors include aspects such as software size, the complexity of the project, quality, storage requirements, and many more. Table 1 provides a list of the product factors compiled from multiple sources. A great deal of DoD and industry research and daily estimating places these types of factors front and center (Clark & Madachy, 2015; Boehm, et al., 2000).

Table 1: Product Factors

Product Factor	Description	Sources
Developed for reusability	To what extent should the components be reusable?	[1, 4, 5]
Development flexibility	How strong are the constraints on the system?	[3, 4, 5]
Execution time constraints	How much of the available execution time is consumed?	[1, 2, 3, 4, 5]
Main storage constraint	How much of the available storage is consumed?	[1, 2, 3, 4, 5]
Precedentedness	How similar are the projects?	[3, 4, 5]
Product complexity	The complexity of the function and structure of the software.	[1, 2, 3, 4, 5]
Product quality	The quality of the product influences motivation and hence productivity.	[2, 3, 4, 5]
Required software reliability	The level of reliability needed.	[1, 2, 3, 4, 5]
Reuse	The extent of reuse.	[3, 4, 5]
Software size	The amount of code in the system.	[1, 2, 3, 4, 5]
User interface	The degree of complexity of the user interface.	[2, 3, 4, 5]
Technical dependencies	Data-related or functional dependencies such as call graphs or coupled changes.	[3]

The process category of technical factors relates to the technical aspects of the software development process rather than the product itself (Wagner & Ruhe, 2018). These process factors include aspects such as project length, project type, development method, and many more. A list of the process factors compiled from multiple sources is in Table 2.

Table 2: Process Factors

Process Factor	Description	Sources
Agile	Is an agile development process used?	[3, 4]
Architecture risk resolution	How are the risks mitigated by architecture?	[3, 4, 5]
Completeness of design	The amount of the design that is completed when coding starts.	[3, 4, 5]
Early prototyping	Early in the process prototypes are built.	[3, 4, 5]
Effective and efficient V&V	The degree to which defects are found and the required effort therein.	[3, 4, 5]
Hardware concurrent development	Is the hardware developed concurrently?	[2, 3, 4, 5]
Outsourcing and global distribution	Degree of outsourcing of the work of the project.	[3]
Platform volatility	Timespan between major changes.	[1, 2, 3, 4, 5]
Process maturity	The well-definedness of the process.	[3, 4, 5]
Project duration	Length of the project.	[2, 3, 4, 5]
Project type	Integration or development project.	[3, 4]

The development environment category of technical factors relates to tools a programmer or developer may use in the software development project (Wagner & Ruhe, 2018). These development environment factors include aspects such as programming language used, Application Type, and many more. A list of the process factors compiled from multiple sources can be seen in Table 3. The work of Clark and Madachy which inspires the current study has served as a launching point for DoD to consider such aspects of a project more carefully.

Table 3: Development Environment Factors

Development Environment Factor	**Description**	**Sources**
Documentation match to life cycle needs	How well the documentation fits the needs	[1, 2, 3, 4, 5]
Domain/Type	Application Types such as embedded software, management information system, or web application	[2, 3, 4]
Programming language	The programming language used	[2, 3, 4, 5]
Use of software tools	The degree of tool use	[1, 2, 3, 4, 5]
Use of modern development practices	For example, continuous integration, automated testing, or configuration management	[2, 3, 4, 5]

Soft factors encompass the non-technical and intangible aspects influencing software development teams and overall work environments; soft factors are often referred to as human factors (Oliveira, Conte, Cristo, & Valentim, 2013). Soft factors that influence software productivity can be further subdivided into corporate culture, team culture, developer capabilities, environmental aspects, and other environmental factors (Sadowski & Zimmerman, 2019; Trendowicz & Münch, 2009; Wagner & Ruhe, 2018). These soft factors play an important role in software productivity outside of just the technical factors. Corporate culture factors are all the factors that are in direct relation to the company-wide cultural environment that have a significant impact on software productivity (Wagner & Ruhe, 2018).

These corporate culture factors include aspects such as credibility, fairness, and respect at the organizational level. A list of the product factors compiled from multiple sources can be found in Table 4. It is such factors that appear more and more relevant as one considers the massive changes possible through agile software development.

Table 4: Corporate Culture Factors

Corporate Culture Factor	**Description**	**Sources**
Credibility	Open communication and competent organization.	[3, 4, 5]
Fairness	Fairness in compensation and diversity.	[3, 4, 5]
Respect	Opportunities and responsibilities.	[3, 4, 5]

The team culture category of soft factors is similar to the corporate culture factors but on the team level (Wagner & Ruhe, 2018). These factors can vary largely from team to team even in the same parent corporation (Sadowski & Zimmerman, 2019). These team culture factors include aspects such as camaraderie amongst team members, communication, overall team cohesion, and many more. A list of the team culture factors compiled from multiple sources can be found in Table 5.

Table 5: Team Culture Factors

Team Culture Factor	**Description**	**Sources**
Camaraderie	Social and friendly atmosphere.	[3, 4, 5]
Clear goals	How clearly defined are the group goals?	[3, 4, 5]
Communication	The degree and efficiency of which information flows in the team.	[3, 4, 5]
Psychological safety	The atmosphere is safe for risk-taking.	[3]
Sense of eliteness	The feeling in the team that they are superior.	[3, 4, 5]
Support for innovation	To what degree assistance for new ideas is available.	[3, 4, 5]
Team cohesion	The cooperativeness of the stakeholders.	[3, 4, 5]
Team identity	A common identity of the team members.	[3, 4, 5]
Turnover	The amount of change in the personnel.	[1, 3, 4, 5]

The capabilities and experience category of soft factors relate to the individual developers and team members on a software development project (Wagner & Ruhe, 2018).

There are varied results to how individual productivity relates to productivity. Experience and skill level are considered important in interviews but tend to test insignificant in empirical studies; suggesting that experience does not necessarily correlate to higher levels of productivity (Trendowicz & Münch, 2009; Sadowski & Zimmerman, 2019). These capabilities and experience factors include aspects such as individual's capability, programming language experience, level of happiness, platform experience, and many more. A list of the capabilities and experience factors compiled from multiple sources can be found in Table 6.

Table 6: Capabilities and Experience Factors

Capabilities and Experience Factor	Description	Sources
Analyst capability	The skills of the system analyst.	[1, 2, 3, 4, 5]
Application Type experience	The familiarity with the Application Type.	[1, 2, 3, 4, 5]
Developer personality	Individual personality and the mix of different personalities on the team.	[3, 4]
Developer happiness	Positive experiences leading to positive emotions.	[3]
Language and tool experience	The familiarity with the programming language and tools.	[1, 2, 3, 4, 5]
Manager Application Type experience	The familiarity of the manager with the application.	[3, 4, 5]
Manager capability	The control of the manager over the project.	[3, 4, 5]
Platform experience	The familiarity with the hardware and software platforms.	[1, 3, 4, 5]
Programmer capability	The skills of the programmer.	[1, 2, 3, 4, 5]

The environment category of soft factors relates to the properties of the working environment for the developers and team (Wagner & Ruhe, 2018). These environmental factors could come from the team or organizational level. Environment factors include aspects such as the office layout, workplace suitability, uninterrupted work hours, and many more. A list of the environmental factors compiled from multiple sources can found in Table 7.

Table 7: Environment Factors

Environment Factor	Description	Sources
E-factor	This environmental factor describes the ratio of uninterrupted hours and body-present hours.	[3, 4, 5]
Office layout	Private or open-plan office layout.	[3]
Physical separation	The team members are distributed over the building or multiple sites.	[1, 3, 4, 5]
Proper workplace	The suitability of the workplace to do creative work.	[3, 4, 5]
Time fragmentation	The amount of necessary "context switches" of a person.	[3, 4, 5]
Telecommunication facilities	Support for work at home, virtual teams, video conferencing with clients.	[3, 4, 5]

The final soft factor category is the project category factors. The project category of factors relates to the human influence on the software development project rather than the technical factors (Wagner & Ruhe, 2018). These project factors include aspects such as team size, required stability, and schedule. A list of the project factors compiled from multiple sources can be found in Table 8.

Table 8: Project Factors

Project Factor	Description	Sources
Average team size	Number of people on the team.	[2, 3, 4, 5]
Requirements stability	The number of requirements changes.	[3, 4, 5]
Schedule	The appropriateness of the schedule for the development task.	[1, 3, 4, 5]

An overview of the sources with their accompanying index used to compile the software productivity factors can be found in Table 9.

Table 9: Productivity Factor Sources

Source Index	Source
1	Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B., Horowitz, E., . . . Steece, B. (2000). COCOMO II. USC Center for Software Engineering.
2	Maxwell, K., Van Wassenhove, L., & Dutta, S. (1996). Software Development Productivity of European Space, Military, and Industrial Applications. *IEEE Transactions on Software Engineering*.
3	Sadowski, C., & Zimmerman, T. (2019). *Rethinking Productivity in Software Engineering.* Apress Media LLC.
4	Trendowicz, A., & Münch, J. (2009). Factors Influencing Software Development Productivity. Fraunhofer Institute for Experimental Software Engineering.
5	Wagner, S., & Ruhe, M. (2018). A Systematic Review of Productivity Factors in Software Development. Germany.

The list of factors influencing software productivity is extremely diverse. The list of factors discussed in this section is far from comprehensive but rather a compilation of the most commonly cited software productivity factors. As described, both technical and soft factors are believed to have a significant impact on software productivity. As challenging as software productivity is, it is believed these factors can go a long way to explaining variance between ESLOC and effort. Productivity of a software development team is decisive for a software project to be considered successful (Wagner & Ruhe, 2018). However, controlling the productivity of a software development program is only possible if the factors influencing the project are known (Trendowicz & Münch, 2009). This knowledge is crucial for program managers to control these factors in their software development projects to improve overall productivity (Oliveira, Conte, Cristo, & Valentim, 2013).

Measuring Software Productivity

Software productivity metrics have varying definitions, but they all come back to the same generally agreed upon principle; that productivity is the amount of product produced for

a given amount of resources (Sadowski & Zimmerman, 2019). In Table 10 below is a compiled list of definitions of productivity concerning software productivity.

Table 10: Productivity Definitions

Productivity Definition	Equation	Source
The ratio of outputs to inputs.	$Productivity = \frac{Outputs}{Inputs}$	(Clark & Madachy, 2015)
The ratio of the amount of product to the amount of resources used to generate it.	$Productivity = \frac{Product}{Resources\ Used}$	(Morasca & Russo, An Empirical Study of Software Productivity, 2001)
The functional size of software developed divided by the amount of effort employed in the development process.	$Productivity = \frac{Software\ Functional\ Size}{Effort}$	(Lavazza, Morasca, & Tosi, 2018)
Source lines of code (SLOC) per man-month. It is a measure of the amount of product produced per unit of human effort.	$Productivity = \frac{SLOC}{Manmonths\ of\ Effort}$	(Maxwell, Van Wassenhove, & Dutta, 1996)
Productivity describes the ratio between output and input	$Productivity = \frac{Outputs}{Inputs}$	(Sadowski & Zimmerman, 2019)

The input metric used as the denominator by software productivity equations is often the effort required by a development project commonly measured in hours, months, or person-months. The COCOMO II Model Definition Manual defines a person-month as the amount of time a person spends working on the software development project for one month, which is generally accepted as 152 hours of work or effort per person-month (Boehm, et al., 2000). The equation used by the COCOMO II for deriving effort estimation in person-months can be seen in Equation.

Equation 1: Person-Months (Boehm, et al., 2000)

$$Person - Month = A * Size^{E} * \prod_{i=1}^{n} EM_i$$

The measure of productivity appears to be straightforward. Yet, much controversy exists in productivity metrics using SLOC. Bill Gates, for example, once complained, “Measuring software productivity by lines of code is like measuring progress on an airplane by how much it weighs.” Commentary within software productivity research has been similar. Sadowski and Zimmermann said that no single measurement (SLOC) can adequately capture developer productivity, instead, a set of metrics tailored to specific programs would be better suited (Sadowski & Zimmerman, 2019).

The key elements here are size and effort, using certain measures. Size is measured in function points, story points, and source lines of code (SLOC). SLOC has long been the workhorse of productivity research. Within DoD, SLOC remains the king of software size measurement. SLOC is generally defined as any software statement that is designed, documented, and tested (Clark & Madachy, 2015). Not all statements require the same amount of effort. There are generally four categories of statements recognized. Logical, non-commented source, physical and total. The implication is that the count of statements must vary based on the category. This problem is commonly addressed by normalizing SLOC into *equivalent* source lines of code (ESLOC). ESLOC normalizes for the additional resource demands attributed to different sources of code along with the software engineering, testing, and software integration required (Clark, et al., 2017). In this manner, it is believed to reliably translate statement count into a measure of effort. As effort, then, it may be compared to another level of effort, hours contributed to the task, producing a unique

measure of productivity for a program. SLOC is identical to physical source lines of code and is one of the most commonly used measures in the productivity literature. The COCOMO II model describes the goal of SLOC as the measure of the amount of intellectual work put into program development (Boehm, et al., 2000). Counting functions points, in contrast, is vastly more complicated and can result in misleading measures of output and functionality for a software project (Maxwell, Van Wassenhove, & Dutta, 1996). Counting feature points is another method that expands on function point measurements; however, this is not a widely used method (Maxwell, Van Wassenhove, & Dutta, 1996).

Notionally, the problem with measuring productivity by way of the ratio of outputs and inputs is that it is a false paradigm, one adopted from the industrial or manufacturing environment. This does not translate well into software development. Software development is much more a human-based skill embedded within a dynamic organizational process than a replicable mechanical process. This human element leads to major uncertainties, inconsistencies, and reliability issues in software development (Trendowicz & Münch, 2009). A single productivity metric, therefore, is imperfect. Productivity tends to be too broad of a concept to be put into a single metric form, and confounding factors add additional complexity (Sadowski & Zimmerman, 2019). Jensen shows the human element impact on software development and productivity in his People-Project-Process Triad in Figure 2.

Figure 2: Software Development Triad: People-Process-Project (Jensen, 2015)

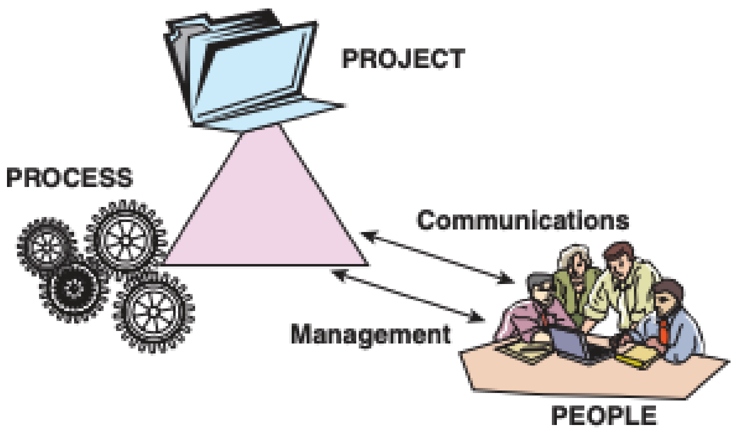


Figure 2 shows the connection of people and processes in software development projects. The people element, especially the continued reliance on communication and management, is a crucial connection to understanding why software productivity is difficult to measure and keep consistent. Since the software development arena is primarily knowledge work as opposed to physical labor, the main output of the software is dependent on the programmer's knowledge (Sadowski & Zimmerman, 2019). Thus, productivity—measured with the tools traditionally available, and those available for DoD measurements—tends to be highly variable in software development. Although the best metric to gauge productivity is subject to debate, the ubiquitous use of SLOC as a measurement of software productivity provides potentially meaningful comparisons amongst software development projects (Clark & Madachy, 2015). SLOC is arguably the most widespread method of measuring the size of a software development project and what will be used as a baseline for meaningful comparison in this research.

Super Domain and Application Type

Many factors of productivity have been examined in previous literature. Among those factors specific to "environment," Super Domain and Application Type have recently become of particular interest to the DoD. But Application Type has long been considered an important aspect of software productivity and cost estimation. Measures of Excellence: Reliable Software on Time, within Budget (1992) discussed the implications of sorting software development projects by Application Type. Putnam discusses that when estimating software projects and productivity, estimators must be aware of the type of application. Due to highly variable productivity rates in software development, departures from constant productivity become apparent and productivity in software development appears to be a function of the size and complexity of the software system (Putnam, 1992). Putnam's study which consisted of 1600 projects from the Quantitative Software Management Incorporated database suggested that Application Type is the second biggest cost driver following the size of the software product (Putnam, 1992). Putnam's study suggests that Application Type is one of the largest cost drivers affecting overall software productivity and cost growth. However, the data was proprietary and left out descriptive statistics and empirical details.

Several other studies beginning in 2003 have attempted to bolster the Application Types implications to productivity within the DoD software cost estimating community. A 2003 study by Harmon and Om of the Institute for Defense Analysis on ground-based ballistic missile command, control, and communications (C^3) is 2 to 2.5 times more productive in development than other ground software Application Types when holding software size and staff level ceteris paribus (Harmon & Om, 2003). Harmon and Om's study suggests Application Type is a relevant predictor of productivity within the DoD type

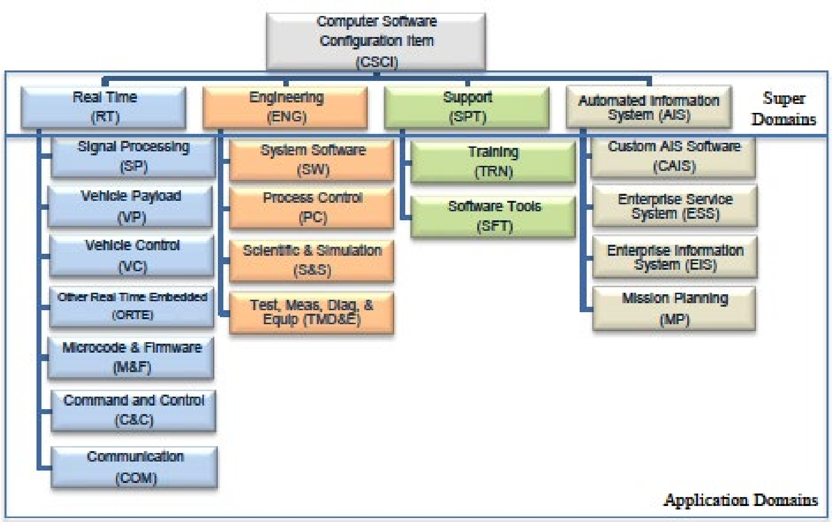
software development programs. A 2011 study suggested software cost parameters including size, effort distribution, and productivity are imprecise due to variations by Application Type. To better predict productivity, the authors chose to segment the DoD dataset according to Application Type (Madachy, Bohem, Clark, Tan, & Rosa, 2011). Tan employed Application Type to improve the COCOMO II effort distribution and more accurately allocate resources for software projects (Rosa, et al., 2014). The results from Tan's study supported the use of Application Types in effort distribution estimation (Tan, 2012). Tan's results show promise for the acceptance of Application Type validity in software cost estimating and productivity efforts. Another study by Rosa et al. looked at 317 DoD programs to improve the method used for predicting software and schedule effort. Rosa et al. found that the Mission Planning and Intelligence & Information Application Types ought to be the most productive (Rosa, et al., 2014). Also, Sensor Control and Vehicle Payload, are the least productive Application Types (Rosa, et al., 2014). Rosa et al. showed promising results but yet again failed to empirically test the distinction of productivity values.

The effect of these DoD-centric studies has been to officially adopt the concept of Application Type as a meaningful construct for analysis within DoD. In 2015, Application Type was adopted by the DoD in the Software Cost Estimation Metrics Manual for Defense Systems as a significant factor influencing software productivity. Under new guidance, projects must be mapped into Super Domains and Application Types in the central SRDR repository for widespread cost analysis across the DoD. Application Type is now used by the DoD to develop cost estimating relationships (CERs) and benchmarks in software acquisition projects (Clark & Madachy, 2015). Clark and Madachy have laid the significant groundwork in this realm by providing readymade CERs per Application Type in the Software Cost

Estimation Metrics Manual for Defense Systems, attempting to bolster the importance of these relationships. Since productivity varies greatly across Application Types, the grouping of similar software projects allows for an organization to more accurately benchmark and compare results.

The SRDR Verification and Validation Guide Version 4.0, outlines the process used for mapping individual DoD software intensive programs. The purpose of the SRDR Verification and Validation Guide Version 4.0 is to provide general directions, key focus areas, and possible solutions to cost community members tasked with inspecting SRDR data submissions. Each Computer Software Configuration Item (CSCI) is first mapped by Super Domain, then by Application Type using the SRDR Data Dictionary and supporting functional description using the overall SRDR contractor submission (Lanham, et al., 2018). The mapping tree used can be found in Figure 3.

Figure 3: Super Domain and Application Type Map (Lanham, et al., 2018)



The SRDR Verification and Validation Guide Version 4.0 defines 4 accepted Super Domains. The descriptions of these Super Domains can be found in Table 11.

Table 2: Super Domain Descriptions (Lanham, et al., 2018)

Super Domain	Description
Real Time (RT)	RT is the most constrained type of software. These are specific solutions limited by system characteristics such as memory size, performance, or battery life. These projects take the most time and effort due to constraints e.g., • May have guaranteed execution requirements i.e., missed deadline means catastrophic results • May have to be compact and efficient due to limited storage capacity and high throughput requirements • Could have very high reliability requirements (life critical, manned mission) • Might have tightly coupled interfaces • Program code may be imprinted on hardware devices • May process sensor inputs and directs actuator outputs • Sometimes executed on special-purpose processors
Engineering (ENG)	Engineering software operates under less severe constraints than real-time software. This software may take the outputs of real-time software and further process them to provide human consumable information or automated control of devices. Or the software may perform transformation and aggregation / distribution of data. These projects take more time and effort due to multiple factors, e.g., • May have a fast response time requirement • May have more storage capacity • Might need to be highly reliable but not life critical • May have multiple interfaces with other systems • May implement complex algorithms, models or protocol • Program code can be modified or uploaded • Executes on general purpose processors that may be embedded in special purpose hardware
Automated Information Systems (AIS)	Automated Information System software provides information processing services to humans or software applications. These applications allow the designated authority to exercise control and have access to typical business / intelligence processes and other types of information access. These systems also include software that facilitates the interface and control among multiple COTS / GOTS software applications. This software has few constraints, e.g., • Must have acceptable response time • Fewer storage and throughput constraints • Must be reliable enough to prevent data loss • May consist of a single COTS / GOTS solution or multiple products coordinated with customer software • Algorithms, models, and protocols are well understood • Code may not be available for modification • Software restarts are acceptable • Executes on commercial processing hardware
Support (SPT)	Support software assists with operator training and software testing. This software has few constraints, e.g., • Has to have an acceptable response time most of the time • Less limited by storage or throughput • Less stringent reliability requirement • Software restarts are acceptable • Fewer interfaces • Relatively low complexity algorithms, models or protocols • Program code can be modified and uploaded • Executes on general purpose processors on general purpose computer boards

Super Domains are a higher-level grouping of similar software Application Types, as can be seen in Figure 3. Within each Super Domain are Application Type possessing similar functional attributes. The DoD Software Factbook (2015) studied the production rate of the Super Domains listed in Table 11. For an average-sized project, Real Time showed the highest productivity rate at 1.5 months per KESLOC. Engineering and Mission Support followed with a productivity rate at 1.3 months per KESLOC. Automated Information systems showed the lowest rate of productivity at 1.1 months per KESLOC. The DoD

Software Factbook mentions that care must be taken into account when applying these productivity rate expectations to projects of other sizes as production rates can vary depending on the size of the project (Clark, McCurley, & Zubrow, 2015). The DoD Software Factbook (2015) also extended this research to create staffing and cost factors derived from Super Domain productivity rates (Clark, McCurley, & Zubrow, 2015). Automated Information Systems was found to have the highest average production rate at 1.1 months per KESLOC. Mission Support and Engineering software fell in the middle with an average production rate of 1.3 months per KELSOC for both, respectively. The least productive Super Domain is Real Time at an average production rate of 1.5 months per KESLOC. The DoD Software Factbook (2015) is in line with the Software Cost Estimation Metrics Manual for Defense Systems further providing evidence that Super Domain and Application Type are pertinent when it comes to estimating software productivity and development costs.

The Software Cost Estimation Metrics Manual for Defense Systems defines Application Types as groups of software applications with similar functions and attributes (Clark & Madachy, 2015). Table 12 lists the Application Types as described by Clark and Madachy.

Table 3: Application Type Definitions (Clark & Madachy, 2015)

Application Type	Description
Microcode and Firmware (M&F)	Microcode and Firmware is software stored on target hardware devices that do not have hard disks and use programmable logic devices. It is a combination of persistent memory and the program code and data stored in it. Examples: Field Programmable Gate Arrays (FPGAs); Microwave controllers; Field Programmable Logic (FPL); Electronic Programmable Logic Device (EPLD); Application Specific Integrated Circuit (ASIC); Programmable Read-Only Memory (PROM); Erasable Programmable Read-Only Memory (EPROM); Electrically Erasable Programmable Read-Only Memory (EEPROM); Complex Programmable Logic Device (CPLD); Programmable Array Logic (PAL).
Sensor Control and Signal Processing (SCP)	Software that requires timing-dependent device coding to enhance, transform, filter, convert, or compress data signals. Examples: Lasers; Sonar; Acoustic; Electromagnetic; Signal Processor; Radar Altimeter; Photographic Sensors; Motion Sensors; Infrared Sensors; Sensor Assembly; Electronic Sensors; Seeker Assembly; Signal Electronics; Optical Assembly; Tracking Sensors; Antenna Assembly.
Vehicle Control (VC)	Software necessary for the control of vehicle primary and secondary mechanical devices and surfaces. Examples: Flight Control; Electrical Power; Hydraulic; Fuel Subsystem; Propulsion; Attitude Control System; Structures & Mechanisms; Bus Flight Software; Thermal Control; Landing Gear; Controls software; Thrust Vector Actuation; Executive.
Vehicle Payload (VP)	Software which controls and monitors vehicle payloads and provides communications to other vehicle subsystems and payloads. Examples: Fire Control; Mine Warfare; Electronic Attack subsystem controller; Weapons Delivery and Control; Gun fire control system; Missile fire control systems; Antisubmarine warfare fire control and torpedo fire control systems; Pointing; Command & Control Interface; Payload Flight Software; Armament; Survivability Payload
Real Time Embedded (RTE)	Interrupt-driven, embedded software in military and consumer appliances, devices, and products, possibly directing and processing sensor inputs/outputs, generally with a very small executive for an operating system interface to basic processor(s). Examples: Embedded Electronics/ Appliance; Robotics; PDAs; Telemetry; Tracking & Command (TT&C); Guidance; Navigation and Control; Controls and Displays; Data Links; Radios (device); Remote Control; Receiver; Transmitter; Exciter; Bombing Computer; Video and recorders; Telephones (device); Built-in-Test.
Command and Control (C&C)	Software that allows humans to manage a dynamic situation and respond in real time. Examples: Mission Management; Mission Computer Processing; Mission Control; Command processing; Air traffic control; Data reduction/ analysis; Telemetry Processing; Battlefield command; Battle management.
Communications (COM)	The transmission of information, e.g., voice, data, commands, images, and video across different mediums. Primarily software systems that control or manage transmitters, receivers and communications channels. Examples: Switches; Routers; Integrated circuits; Multiplexing; Encryption; Broadcasting; Transfer modes; Radios (networks). Network management; Network Operations; Satellite communications; Telecommunications; Networks (WAN/LAN); Protocols (VOIP, TCP/IP, PKI, etc.).
System Software (SYS)	Layers of software that sit between the computing platform and applications. Examples: Operating Systems; Infrastructure; Framework; Middleware; Device Driver; Display Drivers; File management; Image Processing; Interface Driver; Utilities.
Process Control (PC)	Software that manages the planning, scheduling and execution of a system based on inputs, generally sensor driven. Examples: Temperature control; Manufacturing process control; Device or instrument control.
Scientific and Simulation (S&S)	Non real time software that involves significant computations and scientific analysis. Examples: System Integration Lab (SIL) Simulation; Simulators; Offline Data Analysis; Expert Systems; Math & Algorithm Intensive; Graphics; Statistical Analysis; Artificial Intelligence; Simulation & Modeling; Engineering & Science; 3D Modeling & Animation; Trainer Simulations; Computer Aided Design (CAD). Weather models.
Test, Measurement, and Diagnostic Equipment (TMDE)	Software used for testing, measuring, diagnosing, emulating, and evaluating operational hardware and software systems. Software necessary to operate and maintain systems and subsystems which are not consumed during the testing phase and are not allocated to a specific phase of testing. This does not include built-in-test (BIT). Examples: Test equipment software; Test driver; Maintenance and Diagnostic; Fault Tolerance; Diagnostic; Equipment emulators.
Training (TRN)	Software used for educational and training purposes. Examples: Computer Based Training (CBT); Computer Aided Instruction (CAI); Tutorial Applications; Courseware.
Software Tools (TOOL)	Software that is used for analysis, design, construction, or testing of computer programs. Examples: Compilers; Linker/loaders; Debuggers; Editors; Assemblers; Requirements analysis & design tool aids; Code generators; Programming aids; Report generators; Code auditors; Test case data recording; Test case data reduction/analysis; Test case generation.
Mission Planning (MP)	Provides the capability to maximize the use of the platform. The system supports all the mission requirements of the platform and may have the capability to program onboard platform systems with routing; targeting; and performance, map, and Intel data. Examples: Scenario generation; Planning & Analysis; Target planning; Route planning; Fuel planning; Cargo load planning.

Custom AIS Software (CAS)	Software needed to build a custom software application to fill a capability gap not captured by COTS/GOTS software packages. Examples: Glue code; External system interfaces; Data transformation; Inter-COTS/GOTS data exchange; Graphical User Interface; Internet Server Applet; Website.
Enterprise Service Systems (ESS)	Software needed for developing functionality or a software service that are unassociated, loosely coupled units of functionality that have no calls to each other embedded in them. Examples: Enterprise service management; Machine-to-machine messaging; Service discovery; People and device discovery; Metadata discovery; Mediation service; Service security; Content discovery and delivery; Federated search; Enterprise catalog service; Data source integration; Enterprise content delivery network; Session management; Presence and awareness; Text collaboration; White boarding and annotation; Application sharing.
Enterprise Information Systems (EIS)	Software needed for building an enterprise information system that uses an integrated database to support typical business processes within business/functional areas and consistent information access across areas and systems. COTS/GOTS attributed to a specific software service or bundle of services. Examples: Enterprise resource planning; Enterprise data warehouse; General ledger; Accounts payable; Revenue and accounts receivable; Funds control and budgetary accounting; Cost management; Financial reporting; Real property inventory and management; Document management; Logistic or Supply Planning & Control; Transaction Processing; Management Performance Reporting; Office Information System; Reservation System; Geographic or spatial information system; Financial Transactions; Database management; Data Warehousing

The equation used to create the CERs accepted by the Software Cost Estimation Metrics Manual for Defense Systems can be seen in Equation 2. It is important to note the use of the exponent, B, in the CER equation. The exponent, B, being less than one indicates economies of scale, while an exponent greater than one indicates diseconomies of scale. Marginal productivity is greater than average productivity if B is less than one, and marginal productivity is less than average productivity if B is greater than one (Banker & Kemerer, 1989).

Equation 2: Application Type Cost Estimating Relationship (Clark & Madachy, 2015)

$$Effort(PersonMonths) = A * KESLOC^{B}$$

Application Type has been studied by Clark and Madachy and these groupings of similar software programs has a *suggested* impact on software productivity. However, upon closer inspection, it appears that the studies have done more to establish a theoretical basis, than an empirical basis for these CERs by failing to empirically test for unique effects of these Application Types. To confirm that each category has unique productivity requires specific empirical testing. A category cannot be applied theoretically in the establishment of a CER but must be affirmed as predictive of a CER that is distinct from any other CER. That

is, a CER must be *determined* to be unique through rigorous testing such as ANOVA or moderation. Clark and Madachy showed no indication of testing the proposed CERs or benchmarks through such testing but their language suggests the CERs, and benchmarks were predetermined to be distinct. The current list of Application Type CER accepted by the Software Cost Estimation Metrics Manual for Defense Systems can be seen in Table 13.

Table 4: Application Type Cost Estimating Relationships (Clark & Madachy, 2015)

Application Type	**Effort Equation**	**N**	R^2
CAS	$PM = 2.64 * KESLOC^{1.02}$	16	97%
COM	$PM = 7.30 * KESLOC^{0.91}$	47	88%
C&C	$PM = 6.60 * KESLOC^{1.05}$	33	88%
MP	$PM = 6.14 * KESLOC^{0.86}$	20	77%
RTE	$PM = 13.20 * KESLOC^{0.84}$	57	83%
SCP	$PM = 26.5 * KESLOC^{0.87}$	36	91%
S&S	$PM = 7.43 * KESLOC^{0.91}$	17	85%
SYS	$PM = 5.06 * KESLOC^{0,98}$	27	93%
TMDE	$PM = 7.42 * KESLOC^{1.00}$	11	92%
VC	$PM = 9.05 * KESLOC^{1.02}$	27	92%
VP	$PM = 22.27 * KESLOC^{0.81}$	18	89%

Clark and Madachy have done the important foundational work of identifying plausible categories of Application Types and their impact on software productivity. These categories *should* make a difference in productivity, and they appear to be well situated within the literature on the subject. Moreover, their initial CERs seem intuitively credible. However, they are lacking empirical distinction between suggested productivity benchmarks and CERs between the Application Types.

This research aims to build upon Clark and Madachy's important work within the software cost estimating arena. This research borrows from Clark and Madachy's methods by evaluating their accepted Application Types and CERs while seeking to determine which we may confidently apply. This research looks for relationships between Application Types, and

common predictors of Application Type. This gap in research of Application Type empirical affirmation and the proposed effects on software productivity will be the major area of focus for this thesis.

Conclusion

This literature review presented previous research of relevant literature on software productivity and its scope. This literature review focused on four key areas: the purpose of productivity in software cost estimating, the different available software productivity measures, the perceived influences on productivity, and the relevance of Super Domains and Application Type in software development. With the literature covered in this chapter, we are able to identify the starting point of this theses theory and strategy moving forward into Chapter III, research methodology.

III. Methodology

Overview

The purpose of this chapter is to describe the procedures used to analyze the significance of Application Type and Super Domain to predict productivity in defense software intensive programs and to identify distinct characteristics of predictive Application Types and Super Domains. This chapter discusses the data and methodology used to analyze the proposed research questions. The data source, collection process, and inclusion/exclusion criteria will be discussed. The steps used to normalize the data and calculations used before the analysis, statistical testing of the data will be examined, and limitations of the data. This chapter aims to summarize the key points of the methodological components used to conduct this study.

Data

The data for this study consists of Software Resource Data Reports (SRDR). SRDRs are a mechanism used by the Office of the Secretary of Defense (OSD) Cost Assessment and Program Evaluation (CAPE) to collect cost and technical data on software development, maintenance, and enterprise resource planning efforts (OSD CAPE, 2019). The SRDR is a mandatory contractor reporting item for all major software development contracts in Acquisition Category I and IA programs, pre-major Defense Acquisition Programs (MDAP), and pre-Major Automated Information System (MAIS) programs following Milestone A approval for any software development project greater than $20 million (OSD CAPE, 2019). SRDR reporting events align with MDAP cost reporting events. The SRDR dataset used for this study was collected from the Defense

Automated Cost Information Management System (DACIMS), which is nested within the Cost Assessment Data Enterprise (CADE) repository.

Data Exclusions and Characteristics

This study focuses on the analysis of final SRDR reports, which are composed of actual project result data. The SRDR database as of the July 2020 update, contains 1908 final reports (DD Form 2630-3 and 3026-1). A complete list of SRDR data exclusions can be found in Table 14.

Table 54: SRDR Data Exclusions

Category	**Number Removed**	**Number Remaining**
Final Reports (DD Form 2630-3 & 3026-1) Only	2591	1908
Missing, Negative, or Zero Reported for Software Development Effort (Hours, Days, Months) and Equivalent Source Lines of Code (ESLOC)	366	1542
Impossible Schedule Tags	132	1410
V&V Quality Tags (Good Only)	753	657
Application Type Small Sample (PC and M&F)	2	655
Final SRDRs for Analysis		**655**

It is important to note that during the data exclusion, 4 of the 17 SRDR Application Types fell out of the final dataset through the exclusion process. These four Application Types were Microcode & Firmware (M&F), Process Control (PC), Enterprise Service System (ESS), and Enterprise Information System (EIS). This left the final dataset after exclusions with 13 Application Types for analysis. Also, due to

reporting inconsistencies and missing data, this study could not use all 1908 final reports for analysis. As discussed in Chapter II, the main aspects when calculating software productivity are Size and Effort. Therefore, records with missing, zeroed, and blank reporting in size and effort data were excluded. Similarly, records with known deficiencies or suspected of error, as determined by the owners of the database, were excluded in accordance with the SRDR Verification & Validation (V&V) guide (Lanham, et al., 2018).

In addition, two more data points were excluded. As that Application Type and Super Domain are the key moderating variables of concern, it was important to consider the sample size of each. The Application Types of Process Control (PC) and Microcode & Firmware (M&F) had only a single data point each, thus rendering any analysis of them meaningless. All other categories had considerably varied N counts but were sufficient for analysis. No further exclusions were necessary. A total of 655 SRDR spanning from 2001 to 2019 were used for analysis in this study. Table 15 provides an overview of the final SRDR dataset's characteristics used for this research.

Table 15: SRDR Dataset Characteristics

Category	**Total**	**% of Data**
SRDRs	655	100%
Super Domain		
Real Time (RT)	481	73.5%
Engineering (ENG)	113	17.3%
Automated Information System (AIS)	41	6.2%
Mission Support (MS)	20	3.0%
Application Type		
Real Time Embedded (RTE)	133	20.4%
Command & Control (C&C)	115	17.5%
Signal Processing (SP)	71	10.8%
Communications (COM)	68	10.3%
Vehicle Control (VC)	54	8.3%
System Software (SS)	54	8.2%
Vehicle Payload (VP)	40	6.1%
Custom AIS Software (CAS)	16	2.4%
Scientific & Simulation (S&S)	35	5.4%
Mission Planning (MP)	25	3.8%
Test, Measurement, & Diagnostic Equipment (TMDE)	25	3.8%
Software Tools (TOOL)	10	1.5%
Training (TRN)	10	1.5%
Service		
Navy (includes USMC)	266	40.2%
Army	220	33.5%
Air Force	164	25.2%
Missile Defense Agency (MDA)	4	>0.1%
Department of Defense (DoD)	1	>0.1%

As can be derived from the table above, the proportions of SRDRs in each respective Super Domain, Application Type, and service are not balanced. For example, there is an overwhelming majority of SRDRs categorized into the Real Time (RT) Super

Domain compared to Engineering (ENG), Automated Information Systems (AIS), or Mission Support (MP). This disproportionate distribution of SRDRs can also be found in the Application Type categorizations in which Real Time Embedded (RTE), Command & Control (C&C), Signal Processing (SP), and Communications (COM) encompass the majority of SRDRs selected for analysis.

Dependent Variables

Thousands of Equivalent Source Lines of Code (KESLOC)

For the sake of robustness, two different measures of Thousands of Equivalent Source Lines of Code (KESLOC) were used in the determination of the relationship between effort and size. The first measure of KESLOC tested was the reported Naval Air Systems Command (NAVAIR) measure of ESLOC in the SRDR database. NAVAIR KESLOC was calculated using Equation 3.

Equation 3: NAVAIR KESLOC

$$NAVAIR\ KESLOC = \frac{((New\ SLOC * New\ CF) + (Modified\ SLOC * Modified\ CF) + (Reuse\ SLOC * Reuse\ CF) + (Autocode\ SLOC * Autocode\ CF))}{1000}$$

The NAVAIR KESLOC equation takes lines of New, Modified, Reuse, and Autocode and multiplies them by conversion factors (CF) associated with the primary coding language used in the software development project. The NAVAIR KESLOC equation is derived from the premise that software development language is a primary aspect when calculation KESLOC size. The logic behind such an equation is that each language has a different efficiency in accomplishing a given task. This method of sizing software by language has been rooted in software productivity studies as an accurate technique (Stephenson, 1976).

The programming language used has long been considered a relevant factor (Wagner & Ruhe, 2018). An additional merit of using the NAVAIR equation is that it is the equation generating the KESLOC measure within the SRDR database, and it is readily available to all who access the database. The complete list of KESLOC software development language conversion factors can be found in Appendix A.

The second measure of KESLOC was developed using a conversion method outlined in Software Cost Estimation Metrics Manual (2015). Clark and Madachy used an Adapted SLOC Adjustment Factor (AAF). This measure is frequently discussed in the literature and is one of the primary textbook models. It is pulled from the base model of equivalent KSLOC from the COCOMO II Model Definition Manual (Boehm, et al., 2000). The AAF equation can be seen in Equation 4.

Equation 4: Adapted SLOC Adjustment Factor (AAF)

$$AAF = (.4 * DM) + (.3 * CM) + (.3 * IM)$$

Where,

DM = Design Modified
CM = Code Modified
IM = Integration Required

The idea behind the AAF is to account for productivity differences associated with the kind of coding task. Therefore, a measure of .4 indicates a larger effort requirement for design modification than that of code modification and integration which use a measure of .3. The AAF is applied to the size of adapted software to get the "equivalent" size (Clark & Madachy, 2015). However, due to constraints of the SRDR dataset, exact percentages for Design Modified, Code Modified, and Integration Required were unattainable. Thus, the

proxy DM, CM, and IM values which the manual identified from a subset of SRDR data that had values for each factor (Clark & Madachy, 2015). The Proxy DM, CM, and IM values derived by Clark and Madachy can be seen in Table 16.

Table 16: Proxy DM, CM, and IM Values (Clark & Madachy, 2015)

Code Type	#	DM Range	DM Mdn.	CM Range	CM Mdn.	IM Range	IM Mdn.	AAF
New		N/A		N/A		N/A		1.0
Modified	101	0-100%	31%	1-100%	44%	3-100%	72%	0.47
Reused	145	0%		0%		0-100%	10%	0.03
Auto-Gen	6	0%		0%		0-100%	50%	0.15

Going forward this variant of KESLOC will be called “AAF KESLOC,” and the equation can be seen in Equation 5.

Equation 5: AAF KESLOC

$$AAF\ KESLOC = \frac{((\#New\ SLOC * 1.0) + (\#Modified\ SLOC * .47) + (\#Reuse\ SLOC * .03) + (\#Autocode\ SLOC * .15))}{1000}$$

With two measures of KESLOC, this study is able to generate two variants of the important measure of productivity.

Productivity

In a 2014 study, Application Type was found to be the second most influential cost driver in software intensive acquisition programs behind size (Putnam, 1992). Productivity is often used as a proxy for software costs due to materials being relatively insignificant in software development, while fixed overhead costs logically relate to the time of effort and manpower costs (Lavazza, Morasca, & Tosi, 2018; Maxwell, Van Wassenhove, & Dutta, 1996). Thus, consistent with most of the literature on cost, productivity is used as the Dependent Variable in this analysis for its ability to convey the cost of a program. Since the

research regarded two different KESLOC measures, two productivity measures were thus calculated and employed. The NAVAIR KESLOC Productivity and the AAF KESLOC Productivity and can be found in Equations 6 and 7.

Equation 6: Productivity (NAVAIR)

$$Productivity\ (NAVAIR) = \frac{NAVAIR\ KESLOC}{PersonMonths}$$

Equation 7: Productivity (AAF)

$$Productivity\ (AAF) = \frac{AAF\ KESLOC}{PersonMonths}$$

Both productivity measures were initially tested. This was done to eliminate the possibility that results depended considerably on the choice of productivity measures.

Independent Variables

The independent variables chosen for our analysis vary between research questions. Independent variables tested for each corresponding research question can be found in Table 17. The five independent variable technical factors chosen for Research Question #3 were chosen based upon data availability in the SRDR dataset and relevance from previous literature as discussed in Chapter II.

Table 17: Independent Variables

Independent Variable(s)	Variable Type	Test(s)	Description
RQ#1			
Application Type	Nominal	ANOVA	Indicates the Application Type mapped to the software development contract.
RQ#2			
Super Domain	Nominal	ANOVA	Indicates the Super Domain mapped to the software development contract.
RQ#3			
Primary Coding Language Conversion Factor	Nominal	Contingency Analysis, & Test for Proportions	Indicates the primary conversion factor of the programing language for the code that is being developed for the contract.
Development Process	Nominal	Contingency Analysis, & Test for Proportions	The primary software development process used in the contract (e.g., Agile, Waterfall, etc.)
Upgrade/New	Nominal	Contingency Analysis, & Test for Proportions	Indicates whether the primary development was for a software upgrade or new software development.
Peak Staff	Continuous	Contingency Analysis, & Test for Proportions	The estimated peak team size measured in full-time equivalent (FTE) staff.
Service	Nominal	Contingency Analysis, & Test for Proportions	The DoD Service (Navy, Air Force, Army) for which the contract is developing software.

For Research Question #1, the 13 Application Types presented in Table 15 were used as the Independent variable to conduct the ANOVA test. For Research Question #2, the four Super Domains presented in Table 15 were used as the Independent variable to conduct the ANOVA test. For Research Question #3, five chosen characteristics were used to conduct contingency analyses and two sample tests for proportions. The five characteristics chosen are described in more detail in Chapter IV.

Statistical Tests

This study's analysis was conducted using parametric statistics. The primary test is to determine if Application Type is associated with significantly different levels of productivity. This same test is used to determine if Super Domain is associated with different levels. For each, an Analysis of Variance (ANOVA) test will be employed. For any Application Type or Super Domain that is different, the study then continues with an exploratory attempt to identify characteristics that may be unique to those different Application Types and Super Domains. To that end, ANOVA and contingency tables are employed.

Test for Normality

In prior research, the relationship between effort and size is not linear. Moreover, we expect the relationship to be noisy due to the large range of the projects' size and effort. We employ the Shapiro-Wilk test on our data to check for normality. The Shapiro-Wilk hypothesis tests were conducted using an alpha level of 0.05. Rejecting the null hypothesis would require transforming the data into logarithmic space to normalize the data. The test of this data was expected to reject the null hypothesis, and indeed, it did. Therefore, the Dependent Variable data in this study was transformed using natural logarithm to achieve approximate normalization and conducting parametric statistical analysis.

Test of Constant Variance

To conduct parametric statistical analysis, the data used in RQ#1 and RQ#2 ANOVA analysis is required to satisfy the assumption of constant variance. To verify the assumption

of constant variance, the Levene test is applied. The Levene test is conducted using an alpha level of .05. The hypothesis test for the Levene test is:

$$H_0: \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2$$

$$H_a: \sigma_i^2 \neq \sigma_j^2, for\ at\ least\ one\ pair\ (i, j)$$

The null hypothesis of the Levene test states that the variance of the data is approximately constant. Failing to reject the null hypothesis indicates there is insufficient evidence to conclude the variances are constant. The assumption of constant variance must be satisfied to test our data using parametric statistical analysis.

One-Way Analysis of Variance (ANOVA)

One-way Analysis of Variance (ANOVA) is used to compare the productivity mean values of independent groups to determine if there is a statistically significant difference in the mean values of various attributes of given cohorts. In the first instance, mean productivity differences between different Application Types and Super Domains will be tested. The one-way ANOVA test is conducted using an alpha level of 0.05. The hypothesis test for the ANOVA test is:

$$H_0: \mu_1^2 = \mu_2^2 = \cdots = \mu_k^2$$

$$H_a: \mu_i^2 \neq \mu_j^2, for\ at\ least\ one\ pair\ (i, j)$$

The null hypothesis of the ANOVA test states that the mean values of the data is approximately equal. Failing to reject the null hypothesis would give insufficient evidence that the data's productivity means are not equal.

Pairwise Means Comparison

A rejection of the null hypothesis during the one-way ANOVA analysis indicates at least one pair of productivity means between groups are significantly different. Following this rejection, it is necessary to test all combinations of pairs within the groups to determine where the differences in means are significant. To test these combinations, we employ the Tukey-Kramer Honestly Significant Difference (HSD) test to determine which groups have distinctly different productivity means. The initial run set the standard to an alpha level of 0.05, but as that the analysis is exploratory, and observing findings just outside the 0.05 bound, we expanded the range to 0.10. The hypothesis test for the Tukey-Kramer HSD test is:

$$H_0: \mu_1 - \mu_2 = 0$$

$$H_a: \mu_1 - \mu_2 \neq 0$$

The null hypothesis of the Tukey-Kramer HSD test states that the mean productivity values of associated groups are approximately equal. Rejecting the null hypothesis would give evidence that mean productivity values between the associated groups are significantly different.

Contingency Tables and Two Sample Test for Proportions

For those tests in which both the IV and the DV are categorical, contingency tables are used to identify if there is a relationship between two categorical variables. In the first set of tests, Application Type will be explored to determine if those which were distinct in terms of productivity had distinctly different characteristics. In the second set of tests, Super domains will be tested. The contingency tables provide insight into the frequency distribution

of our significant Application Types and Super Domains that differ based on chosen independent variables. The independent variables chosen for testing are listed in Table 14. These independent variables were chosen for their relevance as software productivity factors discussed in Chapter II and what was available in the SRDR dataset. We use the Pearson Chi-Squared test to evaluate that the likelihood of the resulting frequencies differing more than by chance alone. For the sake of exploration, the observed significance is at alpha level .10. All specific results will be disclosed. The null hypothesis for the Pearson Chi-Squared test is that no relationship exists between the categorical variables tested. Rejecting the null hypothesis for the Pearson Chi-Squared test indicates that a relationship exists between variables. Due to the small sample size of some Application Types and Super Domains, it also makes sense to reconsider the contingency table findings using the more conservative two sample test for proportions. The two sample test of proportions test the difference between two categorical variables, similar to the contingency analysis. The hypothesis test for the two sample test for proportions is:

$$H_0: p_1 - p_2 = 0$$

$$H_a: p_1 - p_2 \neq 0$$

The two sample test for proportions results at determined at an alpha level of .05 and will be presented after each corresponding contingency table.

Data Limitations

The major limitations of the SRDR dataset include the availability and accuracy of the reported contractor data in the CADE repository. Program offices and contractors use different definitions and standards when reporting SRDRs. These differences were

particularly noticeable when finalizing the SRDR empirical data set for analysis. Some variables had questionable and inconsistent reporting by contractors with potential for human error when translating SRDR data into the CADE dataset. Reporting inconsistencies in the SRDR database could cause potential underlying correlations that will not be able to be accurately accounted for due to data limitations. However, in this study, the data limitations were accounted for by carefully selecting accurate and relevant variables for statistical testing. The selection and elimination methodology used ensured that this study did not capture spurious results stemming from poor data or selection bias.

Summary

This chapter outlined the methodology used to answer the research questions proposed in Chapter I. The purpose of this chapter was to describe the methods used to analyze the significance of Application Type and Super Domain to predict productivity in DoD software intensive programs and identify distinct characteristics of predictive Application Types and Super Domains. This chapter discussed the data source, inclusion/exclusion process, statistical tests used, and data limitations. Chapter IV will provide the results and analysis of this study.

IV. Results and Analysis

Overview

This chapter presents the results of the data analysis applying the methodology covered in Chapter III. The results are presented in three parts correlated with the research questions presented in Chapter I. The first part presents the Application Type analysis for Research Question #1. The second part presents the Super Domain analysis for Research Question #2. The third part presents the Contingency Table analysis for Research Question #3. The final section of the chapter will conclude the results and analysis.

Application Type Analysis

The purpose of this section is to review the results of the analysis used to answer Research Question #1:

How predictive of productivity are Application Types in defense software intensive programs?

Descriptive Statistics

Tables 18 and 19 present the basic descriptive statistics for both NAVAIR and AAF productivity by Application Type.

Table 18: LN Productivity (NAVAIR) Descriptive Statistics by Application Type

Level	Number	Mean	Median	Std Dev	Std Err Mean
RTE	133	6.6867293	6.607	1.5719312	0.1363037
C&C	115	7.1721826	7.07	1.5800624	0.1473416
SP	71	6.349662	6.423	1.5398124	0.1827421
COM	68	6.4046324	6.2915	1.6173865	0.1961369
VC	54	6.1330185	6.328	1.3846203	0.188423
SS	53	6.7090755	6.795	1.4580935	0.2002845
VP	40	6.3866	6.1245	1.5940481	0.2520411
S&S	35	7.1440571	7.131	1.2233693	0.2067871
MP	25	8.1424	8.286	1.1211893	0.2242379
TMDE	25	6.2288	5.94	1.494631	0.2989262
CAS	16	6.9939375	6.55	1.8104904	0.4526226
TOOL	10	6.7753	6.8495	0.9844328	0.311305
TRN	10	6.4657	6.66	1.235295	0.3906346

Table 19: LN Productivity (AAF) Descriptive Statistics by Application Type

Level	Number	Mean	Median	Std Dev	Std Err Mean
RTE	133	6.6298421	6.553	1.6400869	0.1422136
C&C	115	7.1016435	6.997	1.5580142	0.1452856
SP	71	6.4020986	6.436	1.5089794	0.1790829
COM	68	6.3575882	6.2195	1.6504766	0.2001497
VC	54	6.1250926	6.4355	1.375831	0.1872269
SS	53	6.6672642	6.747	1.4289676	0.1962838
VP	40	6.31375	6.2395	1.6538067	0.2614898
S&S	35	7.1168857	7.177	1.254468	0.2120438
MP	25	8.08332	8.022	1.1482566	0.2296513
TMDE	25	6.07332	5.94	1.6301966	0.3260393
CAS	16	6.9708125	6.4945	1.9784372	0.4946093
TOOL	10	6.7463	6.978	1.0812368	0.3419171
TRN	10	6.3618	6.5445	1.0871681	0.3437927

As can be deduced from Tables 18 and 19, there does not appear to be any distinct difference between the descriptive statistics of the NAVAIR and AAF productivity measurements by Application Type. To confirm this assertion, unpaired two-tailed t tests were employed between the Application Type mean values of LN Productivity (NAVAIR) and LN Productivity (AAF). The results of the t tests can be found in Table 20.

Table 20: Application Type Unpaired Two-Tailed t test Results

Level	**Number**	**P-Value**	**Significant or Not Significant**
RTE	133	.7608	Not Significant
C&C	115	.7348	Not Significant
SP	71	.8168	Not Significant
COM	68	.8583	Not Significant
VC	54	.9699	Not Significant
SS	53	.8862	Not Significant
VP	40	.8473	Not Significant
S&S	35	.9194	Not Significant
MP	25	.8519	Not Significant
TMDE	25	.7356	Not Significant
CAS	16	.9763	Not Significant
TOOL	10	.9488	Not Significant
TRN	10	.8490	Not Significant

Based on the results of the unpaired two-tailed t test, it is confirmed that there is no statistically significant difference between the NAVAIR and AAF productivity values by Application Type.

Verification of Assumptions

Before a valid one-way ANOVA and Tukey-Kramer HSD test can be performed, it is necessary to ensure the assumptions of normality and constant variance are satisfied. Both Productivity (NAVAIR) and Productivity (AAF) had leftward skewed distributions when left in unit space. Both distributions overwhelmingly reject the null hypothesis of the Shapiro-Wilk Test confirming that the data comes from non-normal distributions. As discussed in Chapter III, the data for both Productivity (NAVAIR) and Productivity (AAF) was then normalized using natural logarithmic transformation to fit an approximately normal distribution. Figures 4 and 5 show the approximately normal distribution of LN Productivity (NAVAIR) and LN Productivity (AAF) after transformation that are used for the Application

Type analysis. After normalization by way of natural logarithm, the distributions in Figures 4 and 5 still reject the null hypothesis of the Shapiro-Wilk test. However, since both distributions are relatively symmetrical with minimal outliers in the tails, we continue with parametric statistical analysis assuming approximate normality.

Figure 4: LN Productivity (NAVAIR) Distribution

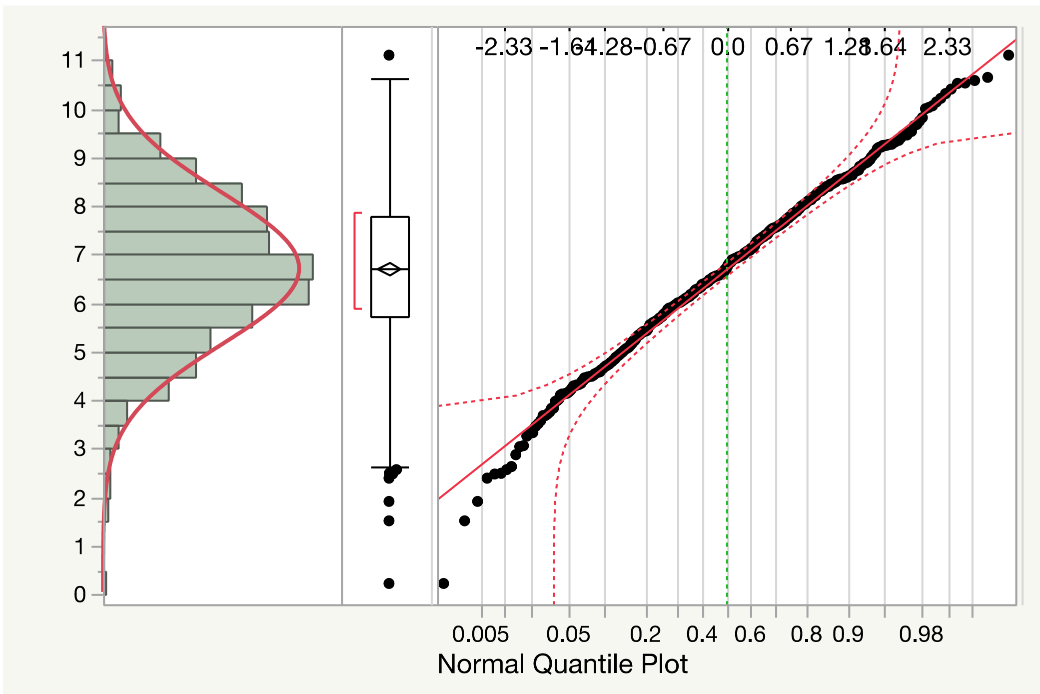
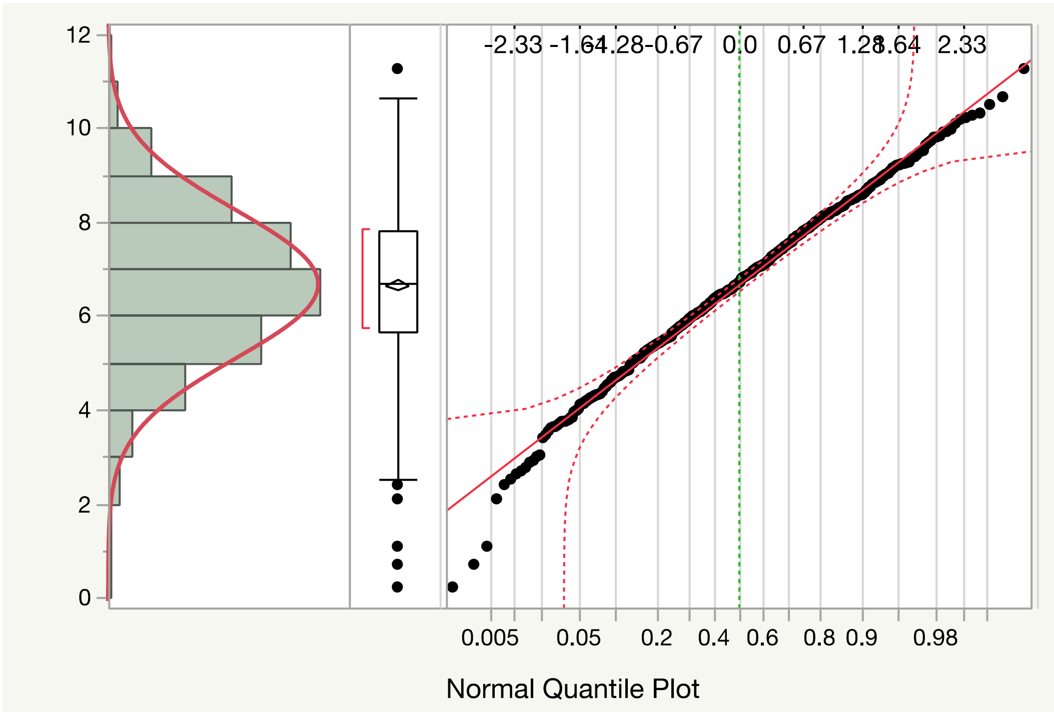


Figure 5: LN Productivity (AAF) Distribution



The second assumption addressed is the constant variance in the one-way ANOVA analysis. As discussed in Chapter III, the Levene test is used to check for constant variance in the one-way ANOVA model. Figure 6 shows the variance chart of LN Productivity (NAVAIR) by Application Type and Table 21 shows the results of the Levene test.

Figure 6: LN Productivity (NAVAIR) by Application Type Variance Distribution

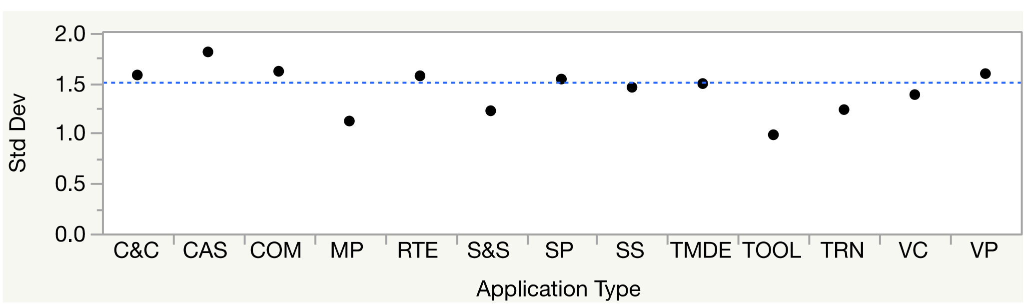


Table 21: LN Productivity (NAVAIR) by Application Type Levene Test

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.7373	12	642	0.7153
Brown-Forsythe	0.7544	12	642	0.6979
Levene	0.8201	12	642	0.6298
Bartlett	1.0837	12	.	0.3687

In Table 21 the Levene test affirms that the data does satisfy the constant variance assumption necessary for ANOVA analysis by failing to reject the null hypothesis with a p-value of 0.6298. LN Productivity (AAF) is also tested for constant variance. Figure 7 shows the variance chart of LN Productivity (AAF) by Application Type and Table 22 shows the results of the Levene test.

Figure 7: LN Productivity (AAF) by Application Type Variance Distribution

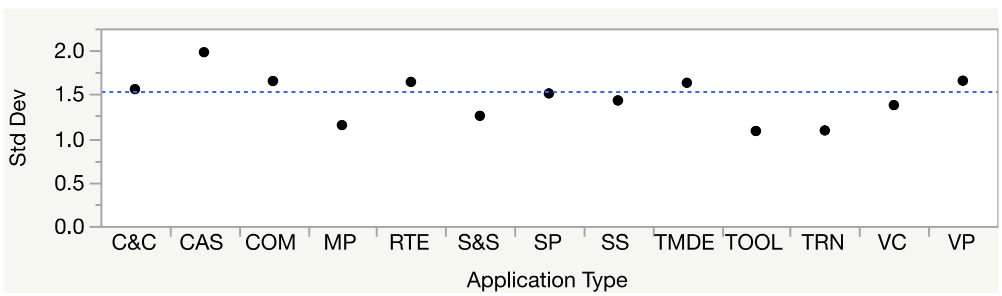


Table 22: LN Productivity (AAF) by Application Type Levene Test

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.9144	12	642	0.5320
Brown-Forsythe	0.8676	12	642	0.5802
Levene	0.9242	12	642	0.5219
Bartlett	1.3119	12	.	0.2033

In Table 22 the Levene test again affirms that the data does satisfy the constant variance assumption necessary for ANOVA analysis by failing to reject the null hypothesis with a p-value of 0.5219. Thus, both LN Productivity (NAVAIR) and LN Productivity

(AAF) satisfy the necessary assumptions of normality and constant variance required to conduct the ANOVA and Tukey-Kramer HSD analysis.

One-way ANOVA & Tukey-Kramer HSD

The one-way ANOVA analysis was first conducted using the NAVAIR productivity measure normalized by natural logarithm as the dependent variable. Application Type categories were used as the response variable. Table 23 shows the results of the one-way ANOVA means comparison.

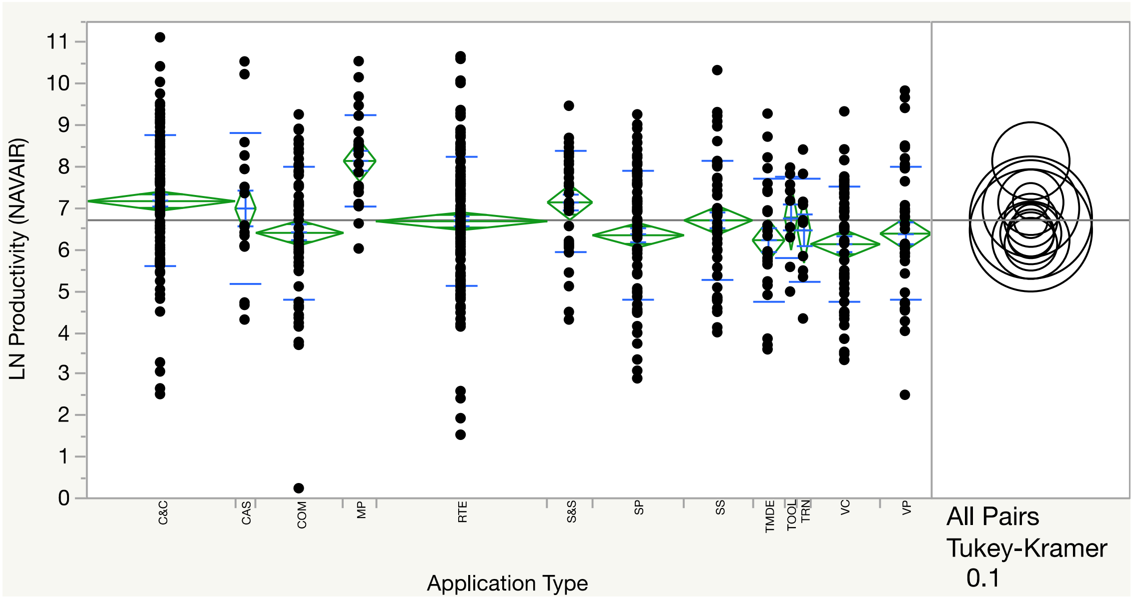
Table 23: LN Productivity (NAVAIR) ANOVA Results

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Application Type	12	127.9620	10.6635	4.6585	<.0001*
Error	642	1469.5749	2.2891		
C. Total	654	1597.5369			

It can be seen in Table 23 that the p-value for the one-way ANOVA is <0.0001 which is less than our alpha level of 0.05. The null hypothesis for the LN Productivity (NAVAIR) ANOVA is rejected indicating that at least one pair of the productivity means between Application Types are not equal. Now that it is apparent that at least one pair of the productivity means of the Application Types are different, we move to the Tukey-Kramer HSD test to determine which of these Application Types differ from each other. Figure 8 shows a graphical representation of the one-way ANOVA and Tukey-Kramer HSD test.

Figure 8: LN Productivity (NAVAIR) ANOVA and Tukey-Kramer HSD Graphic



Following the rejection of the null hypothesis for the one-way ANOVA test for the Application Type's LN Productivity (NAVAIR) means, the next test to be run was the Tukey-Kramer HSD. The Tukey-Kramer HSD shows where the significant differences in means between Application Types are. Table 24 summarizes the significant Tukey-Kramer HSD results for LN Productivity (NAVAIR).

Table 24: LN Productivity (NAVAIR) Tukey-Kramer HSD Pairwise Comparison Results

LN Productivity (NAVAIR) Tukey-Kramer HSD Test $\{\alpha = 0.10\}$		
Level	**Level**	**P-Value**
Mission Planning (MP)	Signal Processing (SP)	<0.0001
Mission Planning (MP)	Communications (COM)	<0.0001
Mission Planning (MP)	Vehicle Control (VC)	<0.0001
Mission Planning (MP)	Vehicle Payload (VP)	0.0005
Mission Planning (MP)	Test, Measurement, & Diagnostic Equipment (TMDE)	0.0007
Mission Planning (MP)	Real Time Embedded (RTE)	0.0009
Mission Planning (MP)	System Software (SS)	0.0068
Command & Control (C&C)	Vehicle Control (VC)	0.0025
Command & Control (C&C)	Signal Processing (SP)	0.0203
Command & Control (C&C)	Communications (COM)	0.0514

The results of the Tukey-Kramer HSD test produced 10 out of 78 individual Application Type means comparisons showing as significant. The null hypothesis is rejected for the 10 Application Type matchups listed in Table 24. This rejection gives evidence that the mean LN Productivity (NAVAIR) values between the Application Type matches are significantly different. The first major finding from this test shows that the Mission Planning (MP) Application Type is significantly different than seven other Application Types in terms of the NAVAIR Productivity. The second major finding from this test shows that the Command & Control (C&C) Application Type's LN Productivity (NAVAIR) mean value is

significantly different than three other Application Types from the SRDR dataset. This is noticeable in Figure 8, where MP and C&C appear to have higher mean values. It is important to emphasize the limited significance found for the Application Type's distinct productivity predictability. With only MP and C&C showing distinct mean productivity values compared to other Application Types in the data, it alludes to Application Type not being distinctly predictive of productivity. However, that is not to say that Application Type is completely unimportant, in general, but it does not appear to be important in productivity prediction. The complete ANOVA and Tukey-Kramer Analysis for LN Productivity (NAVAIR) on Application Types can be found in Appendix B.

Next, the one-way ANOVA analysis was conducted using the AAF productivity measure normalized by natural logarithm as the dependent variable. Application Type categories were used as the response variable. Table 25 shows the results of the one-way ANOVA means comparison.

Table 25: LN Productivity (AAF) ANOVA Results

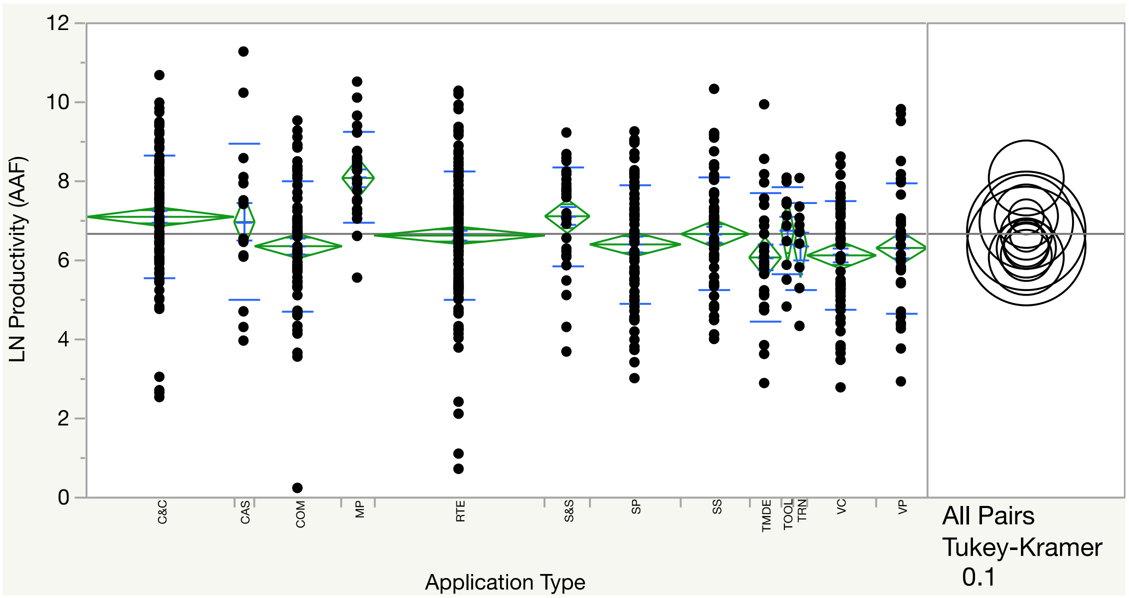
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Application Type	12	122.7651	10.2304	4.3334	<.0001*
Error	642	1515.6697	2.3609		
C. Total	654	1638.4348			

It can be seen in Table 25 that the p-value for the one-way ANOVA is <0.0001 which less than our alpha level of 0.05. The null hypothesis for the LN Productivity (AAF) ANOVA is rejected indicating that at least one pair of the productivity means between Application Types are not equal. It is now apparent that at least one pair of the productivity means of the Application Types are different, we move to the Tukey-Kramer HSD test to

determine which of these Application Types differ from each other. Figure 9 shows a graphical representation of the one-way ANOVA and Tukey-Kramer HSD test

Figure 9: LN Productivity (AAF) ANOVA and Tukey-Kramer HSD Graphic



Following the rejection of the null hypothesis for the one-way ANOVA test for the Application Type's LN Productivity (AAF) means, the next test ran was the Tukey-Kramer HSD. The Tukey-Kramer HSD shows where the significant differences in means between Application Types are. Table 26 summarizes the significant Tukey-Kramer HSD results for LN Productivity (AAF).

Table 26: LN Productivity (AAF) Tukey-Kramer HSD Pairwise Comparison Results

LN Productivity (AAF) Tukey-Kramer HSD Test $\{\alpha = .10\}$		
Level	**Level**	**P-Value**
Mission Planning (MP)	Vehicle Control (VC)	<0.0001
Mission Planning (MP)	Communications (COM)	0.0001
Mission Planning (MP)	Test, Measurement, & Diagnostic Equipment (TMDE)	0.0003
Mission Planning (MP)	Signal Processing (SP)	0.0002
Mission Planning (MP)	Vehicle Payload (VP)	0.0005
Mission Planning (MP)	Real Time Embedded (RTE)	0.0012
Mission Planning (MP)	System Software (SS)	0.0101
Command & Control (C&C)	Vehicle Control (VC)	0.0083
Command & Control (C&C)	Signal Processing (SP)	0.0203
Command & Control (C&C)	Communications (COM)	0.0802

The results of the Tukey-Kramer HSD test produced 10 out of 78 individual Application Type means comparisons showing as significant. The null hypothesis is rejected for the 10 Application Type matchups listed in Table 26. This rejection gives evidence that mean LN Productivity (AAF) values between the Application Type matches are significantly different. The first major finding from this test shows that the Mission Planning (MP) Application Type LN Productivity (AAF) mean value is significantly different than seven other Application Types from the SRDR dataset. The second major finding from this test shows that the Command & Control (C&C) Application Type LN Productivity (AAF) mean

value is significantly different than three other Application Types from the SRDR dataset. This is noticeable in Figure 9, where MP and C&C appear to have higher mean values. Like the NAVAIR productivity measure ANOVA results, it is again important to emphasize the limited significance found for Application Type's distinct productivity predictability for productivity AAF. With only MP and C&C showing as having distinct mean productivity values compares to other Application Types in the data, it again alludes to Application Type not being distinctly predictive of productivity. However, that is not to say that Application Type is completely unimportant, in general, but it does not appear to be important in productivity prediction. The complete ANOVA and Tukey-Kramer Analysis for LN Productivity (AAF) on Application Types can be found in Appendix B.

The one-way ANOVA analysis was conducted twice using both LN Productivity (NAVAIR) and LN Productivity (AAF) for robustness and to rule out any significant differences in results that may come from using either productivity measure. The results from the one-way ANOVA and Tukey-Kramer HSD tests for both productivity measures are not distinctly different. Thus, for this analysis using the NAVAIR or AAF productivity measure when comparing productivity means between Application Types does not have any significant impact on the pairwise means comparison or ANOVA results.

Super Domain Analysis

The purpose of this section is to review the results of the analysis used to answer Research Question #2:

> *How predictive of productivity are Super Domains in defense software intensive programs?*

Descriptive Statistics

Tables 27 and 28 give present the basic descriptive statistics for both NAVAIR and AAF productivity by Super Domain

Table 27: LN Productivity (NAVAIR) Descriptive Statistics by Super Domain

Level	Number	Mean	Median	Std Dev	Std Err Mean
RT	481	6.6260374	6.591	1.5888249	0.07045
ENG	113	6.7375487	6.848	1.424752	0.14535
AIS	41	7.6942195	7.518	1.5182693	0.24131
MS	20	6.6205	6.66	1.098679	0.34550

Table 28: LN Productivity (AAF) Descriptive Statistics by Super Domain

Level	Number	Mean	Median	Std Dev	Std Err Mean
RT	481	6.5875842	6.585	1.6020187	0.07138
ENG	113	6.6751239	6.83	1.4618836	0.14728
AIS	41	7.6491707	7.509	1.6002492	0.24450
MS	20	6.55405	6.7705	1.0735644	0.35007

As can be deduced from Tables 27 and 28, there does not appear to be any significant difference between the descriptive statistics of the NAVAIR and AAF productivity measurements by Super Domain. To confirm this assertion, unpaired two-tailed t tests were employed between the Super Domain mean values of LN Productivity (NAVAIR) and LN Productivity (AAF). The results of the t tests can be found in Table 29.

Table 29: Super Domain Unpaired Two-Tailed t test Results

Level	Number	P-Value	Significant or Not Significant
RT	481	0.6965	Not Significant
ENG	113	0.7544	Not Significant
AIS	41	0.8847	Not Significant
MS	20	0.8387	Not Significant

Based on the results of the unpaired two-tailed t test, it is confirmed that there is no statistically significant difference between the NAVAIR and AAF productivity values by Super Domain.

Verification of Assumptions

Similar to the verification of assumptions section for the Application Type analysis, it is necessary to ensure the assumptions of normality and constant variance are satisfied for the Super Domain ANOVA and Tukey-Kramer HSD analysis. Since both Productivity (NAVAIR) and Productivity (AAF) had leftward skewed distributions when left in unit space. Both of these non-normal distributions overwhelmingly rejected the null hypothesis of the Shapiro-Wilk Test confirming that the data comes from non-normal distributions. As discussed in Chapter III, the data for both Productivity (NAVAIR) and Productivity (AAF) was normalized using natural logarithmic transformation to fit an approximately normal distribution. Refer to Figures 4 and 5 for the approximately normal distribution of LN Productivity (NAVAIR) and LN Productivity (AAF). These approximately normal distributions are used for the Super Domain analysis. The second assumption addressed is that of constant variance. As discussed in Chapter III, the Levene test is used to check for constant variance in the one-way ANOVA model. Figure 10 shows the variance chart of LN Productivity (NAVAIR) by Application Type and Table 30 shows the results of the Levene test.

Figure 10: LN Productivity (NAVAIR) by Application Type Distribution

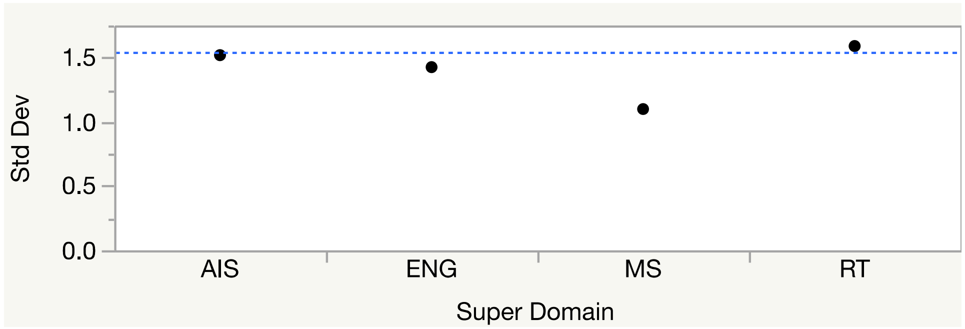


Table 30: LN Productivity (NAVAIR) by Super Domain Levene Test

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.2904	3	651	0.2767
Brown-Forsythe	1.1816	3	651	0.3159
Levene	1.1633	3	651	0.3229
Bartlett	1.8972	3	.	0.1276

In Figure 10 the Super Domains do not initially appear to show constant variance. However, in Table 30 the Levene test affirms that the data does satisfy the constant variance assumption necessary by failing to reject the null hypothesis with a p-value of .3229. LN Productivity (AAF) is also tested for constant variance. Figure 11 shows the variance chart of LN Productivity (AAF) by Super Domain and Table 31 shows the results of the Levene test.

Figure 11: LN Productivity (AAF) by Super Domain Distribution

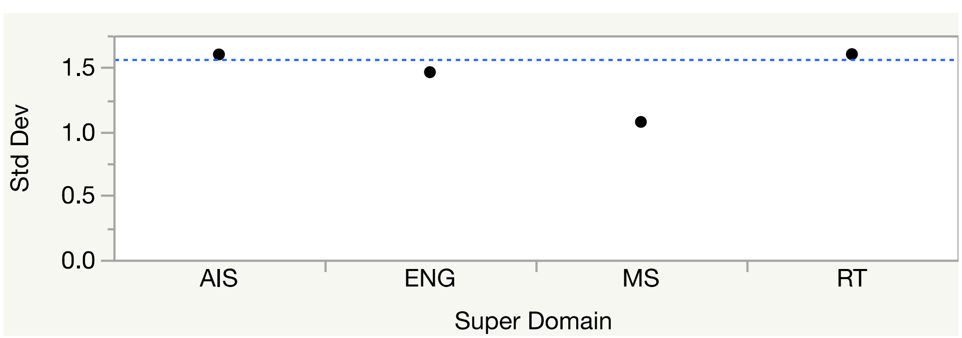


Table 31: LN Productivity (AAF) by Super Domain Levene Test

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.1678	3	651	0.3212
Brown-Forsythe	1.1929	3	651	0.3116
Levene	1.1142	3	651	0.3426
Bartlett	1.9251	3	.	0.1231

In Figure 11 the Application Types do not initially appear to show constant variance. But, in Table 31 the Levene test affirms that the data does satisfy the constant variance assumption necessary by failing to reject the null hypothesis with a p-value of 0.3426. Thus, both LN Productivity (NAVAIR) and LN Productivity (AAF) satisfy the necessary assumptions of normality and constant variance required to conduct the ANOVA and Tukey-Kramer HSD analysis.

One-way ANOVA & Tukey-Kramer HSD

The one-way ANOVA analysis was first conducted using the NAVAIR productivity measure normalized by natural logarithm as the dependent variable. Super Domain categories were used as the response variable. Table 32 shows the results of the one-way ANOVA means comparison.

Table 32: LN Productivity (NAVAIR) ANOVA Results

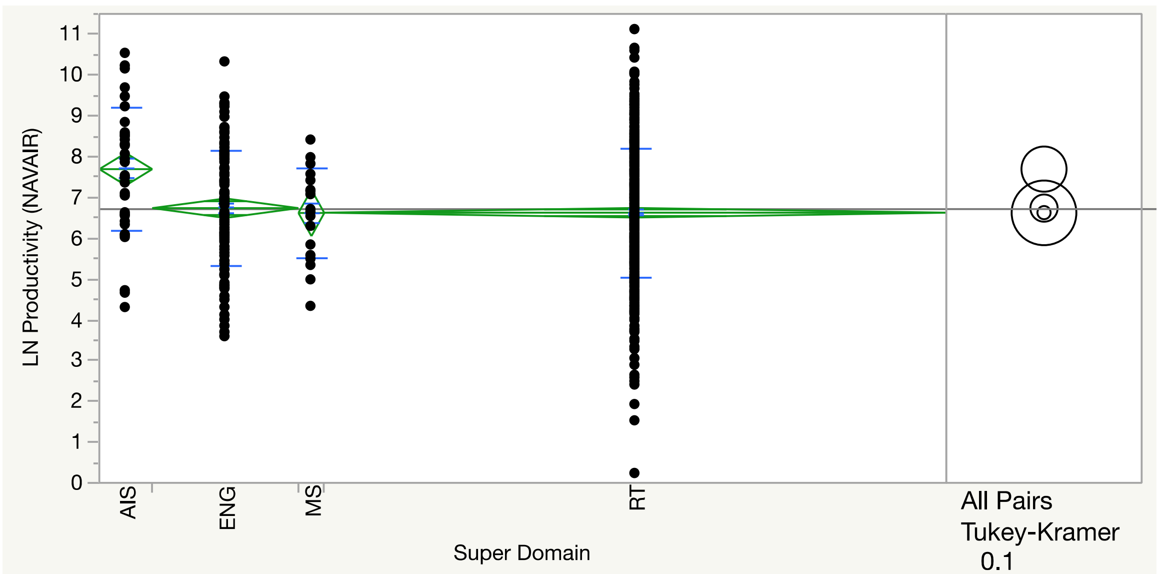
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Super Domain	3	43.3505	14.4502	6.0527	0.0005*
Error	651	1554.1864	2.3874		
C. Total	654	1597.5369			

It can be seen in Table 32 that the p-value for the one-way ANOVA is 0.0005 which less than our alpha level of 0.05. The null hypothesis for the LN Productivity (NAVAIR)

ANOVA is rejected indicating that at least one pair of the productivity means between Super Domains are not equal. It is apparent that at least one pair of the productivity means of the Super Domains are different, next we move to the Tukey-Kramer HSD test to determine which of these Super Domains differ from each other. Figure 12 shows a graphical representation of the one-way ANOVA and Tukey-Kramer HSD test.

Figure 12: LN Productivity (NAVAIR) ANOVA and Tukey-Kramer HSD Graphic



Following the rejection of the null hypothesis for the one-way ANOVA test for the Super Domain's LN Productivity (NAVAIR) means, the nest test ran was the Tukey-Kramer HSD. The Tukey-Kramer HSD shows where the significant differences in means between Super Domains are. Table 33 summarizes the significant Tukey-Kramer HSD results for LN Productivity (NAVAIR).

Table 33: LN Productivity (NAVAIR) Tukey-Kramer HSD Pairwise Comparison Results

LN Productivity (NAVAIR) Tukey-Kramer HSD Test $\{\alpha = .10\}$		
Level	**Level**	**P-Value**
Automated Information Systems (AIS)	Real Time (RT)	0.0001
Automated Information Systems (AIS)	Engineering (ENG)	0.0040
Automated Information Systems (AIS)	Mission Support (MS)	0.0538

The results of the Tukey-Kramer HSD test produced 3 out of 6 individual Super Domain means comparisons showing as significant. The null hypothesis is rejected for the three Application Type matchups listed in Table 33. This rejection gives evidence that the mean LN Productivity (NAVAIR) values between the Super Domain matches are significantly different. The major finding from this test shows that the Automated Information Systems (AIS) Super Domain LN Productivity (NAVAIR) mean value is significantly different than three other Application Types from the SRDR dataset. This is noticeable in Figure 12, where AIS appears to have a higher mean value than the other Super Domains. It is important to emphasize the limited significance found for the Super Domains distinct productivity predictability. With only AIS showing distinct mean productivity values compares to other Super Domains in the data, it alludes to Super Domain not being distinctly predictive of productivity. However, that is not to say that Super Domain is unimportant, in general, but it does not appear to be important in productivity prediction. It is also important to note that AIS may be showing as distinctly predictive due to the MP, which is an Application Type within AIS, being 25 of the 41 data points in the AIS sample. The majority

of the data being MP could be driving all of AIS to be significant compared to the other Super Domains. Additional testing, that was not in the scope of this study, would be required to determine if AIS is being controlled by way of MP. The complete ANOVA and Tukey-Kramer Analysis for LN Productivity (NAVAIR) on Super Domains can be found in Appendix B.

Next, the one-way ANOVA analysis was conducted using the AAF productivity measure normalized by natural logarithm as the dependent variable. Super Domain categories were used as the response variable. Table 34 shows the results of the one-way ANOVA means comparison.

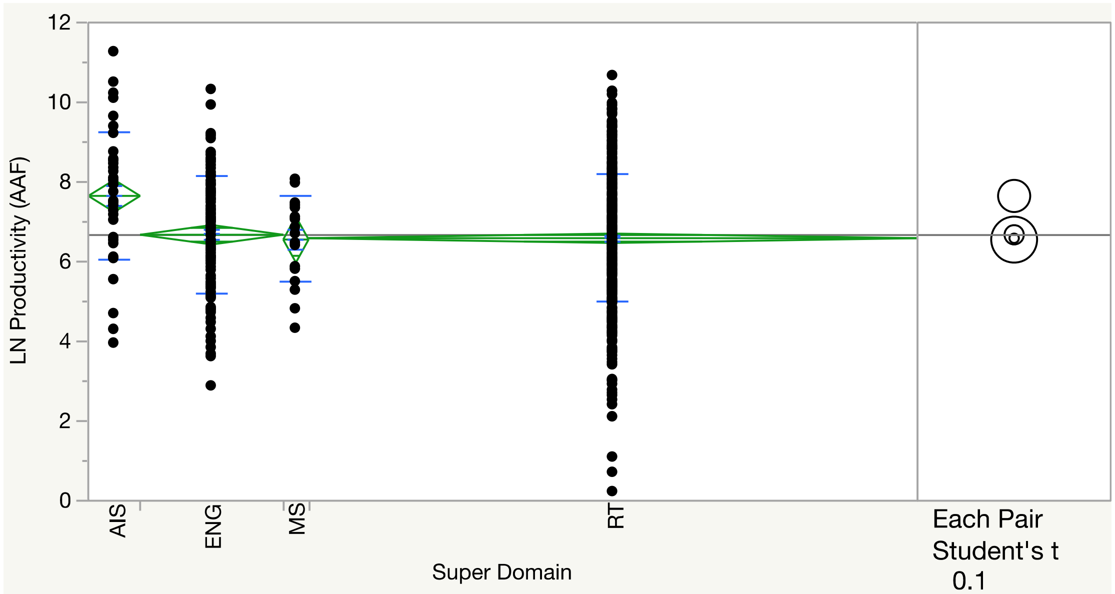
Table 34: LN Productivity (AAF) ANOVA Results

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Super Domain	3	42.8464	14.2821	5.8271	0.0006*
Error	651	1595.5884	2.4510		
C. Total	654	1638.4348			

It can be seen in Table 34 that the p-value for the one-way ANOVA is 0.0006 which less than our alpha level of 0.05. The null hypothesis for the LN Productivity (AAF) ANOVA is rejected indicating that at least one pair of the productivity means between Super Domains are not equal. It is now apparent that at least one pair of the productivity means of the Super Domains are different, we move to the Tukey-Kramer HSD test to determine which of these Super Domains differ from each other. Figure 13 shows a graphical representation of the one-way ANOVA and Tukey-Kramer HSD test.

Figure 13: LN Productivity (AAF) ANOVA and Tukey-Kramer HSD Graphic



Following the rejection of the null hypothesis for the one-way ANOVA test for the Super Domain's LN Productivity (AAF) means, the nest test ran was the Tukey-Kramer HSD. The Tukey-Kramer HSD shows where the significant differences in means between Super Domains are. Table 35 summarizes the significant Tukey-Kramer HSD results for LN Productivity (AAF).

Table 35: LN Productivity (AAF) Tukey-Kramer HSD Pairwise Comparison Results

LN Productivity (AAF) Tukey-Kramer HSD Test $\{\alpha = .10\}$		
Level	**Level**	**P-Value**
Automated Information Systems (AIS)	Real Time (RT)	<0.0001
Automated Information Systems (AIS)	Engineering (ENG)	0.0007
Automated Information Systems (AIS)	Mission Support (MS)	0.0106

The results of the Tukey-Kramer HSD test produced 3 out of 6 individual Super Domain means comparisons showing as significant. The null hypothesis is rejected for the three Super Domain matchups listed in Table 35. This rejection gives evidence that the mean LN Productivity (AAF) values between the Super Domain matches are significantly different. The major finding from this test shows that the Automated Information Systems (AIS) Super Domain LN Productivity (AAF) mean value is significantly different than three other Super Domains from the SRDR dataset. This is noticeable in Figure 13, where AIS appears to have a higher mean value than the other Super Domains. It is important to again emphasize the limited significance found for the Super Domains distinct productivity predictability in AAF as with NAVAIR. Again, only AIS showed distinct mean productivity values compares to other Super Domains in the data, it alludes to Super Domain not being distinctly predictive of productivity. It is also important to note that again AIS may be showing as distinctly predictive due to the MP, which is an Application Type within AIS, being 25 of the 41 data points in the AIS sample. The majority of the data being MP could be driving all of AIS to be significant compared to the other Super Domains. Additional testing, that was not in the scope of this study, would be required to determine if AIS is being controlled by way of MP. The complete ANOVA and Tukey-Kramer Analysis for LN Productivity (AAF) on Super Domains can be found in Appendix B.

The one-way ANOVA analysis was conducted twice using both LN Productivity (NAVAIR) and LN Productivity (AAF) for robustness and to rule out any significant differences in results that may come from using either productivity measure. The results from the one-way ANOVA and Tukey-Kramer HSD tests for both productivity measures are not distinctly different. Thus, for this analysis using the NAVAIR or AAF productivity measure

when comparing productivity means between Super Domains does not have any significant impact on the pairwise means comparison or ANOVA results.

Contingency Table Analysis and Two Sample Test for Proportions

The purpose of this section is to review the results of the analysis used to answer Research Question #3:

> *Among Application Types and Super Domains that show statistically distinct productivity effects, which characteristics are statistically significant?*

Model Overview

As discussed in Chapter III, our analysis for Research Question #3 utilizes the contingency tables and two sample tests for proportions to help identify potentially distinct characteristics of Application Type and Super Domain categories that tested as significant for productivity means in Research Questions #1 and #2. MP, C&C, and the Super Domain of AIS were distinct. The question, then, was what underlying characteristics may be driving the distinction.

The contingency tables provide insight into how MP, C&C, and AIS may be different than the rest of the Application Types and Super Domains. Namely, Mission Planning (MP) is compared to the group of Application Types that the ANOVA was significantly different from in Research Question #1. Next, Command and Control (C&C) is compared to the group of Application Types that the ANOVA was significantly different from in Research Question #1. Lastly, Automated Information Systems is compared to the group of Super Domains that

the ANOVA was significantly different from in Research Question #2. The remaining Application Types and Super Domains that did not fit into Group 1 or Group 2 were excluded from the contingency table analysis. The exclusion of the other Application Types and Super Domains directs the focus of the analysis on the distinct differences between the significant pairings recognized in the ANOVA analysis. Table 36 provides the methodology for how Application Type and Super Domain were parsed.

Table 36: Dependent Variable Groupings

Dependent Variables	Cohort Description
Mission Planning (MP)	*MP:* Programs with an MP Application Type designation (N=25) *VC, TMDE, VP, SP, COM, RTE, SS:* Programs with a VC, TMDE, VP, SP, COM, RTE, SS Application Type designation (N=444) Exclusions: Programs with a C&C, CAS, S&S, TOOL Application Type designation (N=186)
Command & Control (C&C)	*C&C:* Programs with a C&C Application Type designation (N=115) *COM, VC, VP:* Programs with a COM, VC, SP Application Type designation (N=193) Exclusions: Programs with an RTE, TMDE, S&S, VP, COM, RTE, SS, TOOL, TRN Application Type designation (N=347)
Automated Information System (AIS)	*AIS:* Programs with an AIS Super Domain designation (N=41) *ENG, MS, RT:* Programs with an ENG, MS, RT Super Domain designation (N=614) Exclusions: None

Table 37 provides the methodology for how the five chosen characteristics were bucketed.

Table 37: Independent Variable Groupings

Independent Variables	Cohort Description
Primary Coding Language Conversion Factor	Group 1: Programs with primary languages using 100% New, 50% Modified, 5% Reuse, and 32% Autocode conversion factors (See Appendix A) Group 2: Programs with primary languages using 100% New, 15% Modified, 7% Reuse, and 32% Autocode conversion factors (See Appendix A) Exclusions: None
Development Process	Agile: Programs referencing Agile in the primary software development method Non-Agile: Programs not referencing Agile in the primary software development method Exclusions: None
Upgrade/New	Upgrade: Programs listed as an Upgrade software development project New: Programs listed as a New software development project Exclusions: None
Peak Staff	Over: Programs with a peak staff level over the designated mean for the sample Under: Programs with a peak staff level under the designated mean for the sample Exclusions: None
Service	Air Force: Programs with an Air Force service designation Army: Programs with an Army designation Navy: Programs with a Navy (includes USMC) service designation Exclusions: Programs with an MDA or DoD service designation

The Primary Coding Language Conversion Factor Independent Variable cohorts were created using a natural break between conversion factors used in the SRDR to create the NAVAIR ESLOC measure. Figure A-2 in Appendix A shows the two groupings.

Group 1 utilized the following effort conversion factors for each type of code: 100% New, 50% Modified, 5% Reuse, and 32% Autocode. Group 2 utilized the following effort conversion factors for each type of code: 100% New, 50% Modified, 5% Reuse, and 32% Autocode. The Development Process Independent Variable cohorts were created using the contractor-provided primary development process description. Development Process was focused on Agile versus non-agile development methods. The first cohort, Agile, included any CSCI that referenced "Agile" in its primary development process. The second cohort, Non-Agile, included any CSCI that did not reference "Agile" in its primary development process. The Upgrade/New Independent Variable cohorts were created based on if the CSCI was categorized as either a software upgrade or a new software development project. The Peak Staff Independent Variable cohorts were created based on the mean peak staff level of the sample. The first cohort included all CSCIs that were over the mean peak staff level. The second cohort included all CSCIs that were under the mean peak staff level. Lastly, the Service Independent Variable cohorts were created based on which military service branch the software contract was developed for. The focus of the service Independent Variable was the three biggest players: Air Force, Army, and Navy (including USMC). CSCIs with an MDA or DoD designation were excluded due to the small sample size.

In the subsequent sections, the Dependent and Independent nominal variables are tested to find out where the difference between significantly different Application Types and Super Domains are prevalent. All contingency table analyses are performed using an alpha level of 0.10.

Application Type: MP versus VC, TMDE, VP, SP, COM, RTE, and SS

Table 38 summarizes the key results from the MP contingency analysis for the five variables of interest. Complete results can be found in Appendix C.

Table 38: Mission Planning (MP) Contingency Analysis Results

Mission Planning (MP) Contingency Analysis Results $\{\alpha = .10\}$			
Independent Variable	**Pearson Chi-Squared Test P-Value**	**Significant or Not Significant**	**N**
Primary Coding Language Conversion Factor	0.0396	Significant	469
Development Process	0.0011	Significant	469
Upgrade/New	0.1241	Not Significant	469
Peak Staff	0.2773	Not Significant	469
Service	<0.0001	Significant	465[1]

Primary Coding Language Conversion Factor

Table 39 shows the mosaic plot of Application Type MP cohorts by Primary Coding Language Conversion Factor groups.

[1] The Service independent variable testing focuses on Air Force, Navy, and Army. The four excluded were comprised of Missile Defense Agency (MDA) and Department of Defense (DoD) programs.

Table 39: Application Type MP by Primary Coding Language Conversion Factor Contingency Table

Primary Coding Language	Count Total % Col % Row %	Application Type MP Groups: MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
	Group 1	7	59	66
		1.49	12.58	14.07
		28.00	13.29	
		10.61	89.39	
	Group 2	18	385	403
		3.84	82.09	85.93
		72.00	86.71	
		4.47	95.53	
	Total	25	444	469
		5.33	94.67	

Primary Coding Language Conversion Factor rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0396. The rejection indicates that the relationship between MP Application Type groups and Primary Coding Language Conversion Factor groups differ more than expected by chance alone. From the contingency analysis, it can be deduced that the VC, TMDE, VP, SP, COM, RTE, SS Application Type group uses a larger rate of Primary Coding Language Conversion Factors in Group 2 than Group 1. Also, it can be deduced that the MP Application Type group uses a larger rate of Primary Coding Language Conversion Factors in Group 2 than Group 1. Programs in the MP Application Type group use a larger rate of Group 1 languages at 28% than Programs in the VC, TMDE, VP, SP, COM, RTE, SS Application Type group at 13.29%. Programs in the VC, TMDE, VP, SP, COM, RTE, SS Application Type group use a larger rate of Group 2 language at 86.71% than Programs in the MP Application Type group at 72%. Table 40 shows the two sample test of proportions results for MP by Primary Coding Language Conversion Factor.

Table 40: Application Type MP by Primary Coding Language Conversion Factor Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(MP|Group 1)-P(MP|Group 2)	0.061396	-0.00857	0.150032

Adjusted Wald Test (Null Hypothesis)	Prob
P(MP|Group 1)-P(MP|Group 2) ≤ 0	0.0402*
P(MP|Group 1)-P(MP|Group 2) ≥ 0	0.9598
P(MP|Group 1)-P(MP|Group 2) = 0	0.0804

The difference in proportions for Group 1 and Group 2 Primary Coding Language Conversion Factors for the MP Application Type fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.0804. This failure to reject the null indicates the proportions are approximately equal.

Development Process

Table 41 shows the mosaic plot of Application Type MP groups by development process groups.

Table 41: Application Type MP by Development Process Contingency Table

Development Process	Application Type MP Groups			
	Count Total % Col % Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
	Agile	4 0.85 16.00 22.22	14 2.99 3.15 77.78	18 3.84
	Non-Agile	21 4.48 84.00 4.66	430 91.68 96.85 95.34	451 96.16
	Total	25 5.33	444 94.67	469

Development process rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0011. The rejection indicates that the relationship between MP Application Type groups and development process groups differ more than expected by chance alone. From the contingency analysis, it can be deduced that the VC, TMDE, VP, SP, COM, RTE, SS Application Type group uses a larger rate of Non-Agile development processes than Agile development processes. Also, it can be deduced that the MP Application Type group uses a larger rate of Non-Agile development processes than Agile development processes. Programs in the MP Application Type group use a larger rate of Agile development processes at 16% than programs in the VC, TMDE, VP, SP, COM, RTE, SS Application Type group at 3.15%. Programs in the VC, TMDE, VP, SP, COM, RTE, SS Application Type group use a larger rate of Non-Agile development processes at 96.85% than programs in the MP Application Type group at 84%. Table 42 shows the two sample test of proportions results for MP by Development Process.

Table 42: Application Type MP by Development Process Two Sample Test for Proportions

Description	**Proportion Difference**	**Lower 95%**	**Upper 95%**
P(MP|Agile)-P(MP|Non-Agile)	0.175659	0.010633	0.392237

Adjusted Wald Test (Null Hypothesis)	**Prob**
P(MP|Agile)-P(MP|Non-Agile) ≤ 0	0.0193*
P(MP|Agile)-P(MP|Non-Agile) ≥ 0	0.9807
P(MP|Agile)-P(MP|Non-Agile) = 0	0.0385*

The difference in proportions for Agile and Non-Agile Development Processes for the MP Application Type rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.0385. This rejection of the null indicates the proportions are not equal.

Upgrade/New

Table 43 shows the mosaic plot of Application Type MP groups by upgrade/new groups.

Table 43: Application Type MP by Upgrade/New Product and Development Descriptions

Contingency Table

Upgrade/New - Product and Development Description	Count Total % Col % Row %	Application Type MP Groups: MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
New	Count	16	214	230
	Total %	3.41	45.63	49.04
	Col %	64.00	48.20	
	Row %	6.96	93.04	
Upgrade	Count	9	230	239
	Total %	1.92	49.04	50.96
	Col %	36.00	51.80	
	Row %	3.77	96.23	
Total	Count	25	444	469
	Total %	5.33	94.67	

Upgrade/New fails to reject the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.1241. The failure of rejection indicates that the relationship between MP Application Type groups and upgrade/new product and development description groups do not differ more than expected by chance alone. From the contingency analysis, it can be deduced that there no significant difference in upgrade/new product development descriptions between the VC, TMDE, VP, SP, COM, RTE, SS Application Type group, and MP Application Type group. Table 44 shows the two sample test of proportions results for MP by Upgrade/New Product and Development Descriptions.

Table 44: Application Type MP by Upgrade/New Product and Development Descriptions

Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(MP|New)-P(MP|Upgrade)	0.031908	-0.01015	0.073715

Adjusted Wald Test (Null Hypothesis)	Prob
P(MP|New)-P(MP|Upgrade) ≤ 0	0.0687
P(MP|New)-P(MP|Upgrade) ≥ 0	0.9313
P(MP|New)-P(MP|Upgrade) = 0	0.1374

The difference in proportions for New and Upgrade Upgrade/New Product and Development Descriptions for the MP Application Type fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.1374. This failure to reject the null indicates the proportions are approximately equal.

Peak Staff

Table 45 shows the mosaic plot of Application Type MP groups by peak staff groups.

Table 45: Application Type MP Groups by Peak Staff Contingency Table

Peak Staff Groups by Mean	Application Type MP Groups			
	Count Total % Col % Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
	Over 21.65	9 1.92 36.00 7.20	116 24.73 26.13 92.80	125 26.65
	Under 21.65	16 3.41 64.00 4.65	328 69.94 73.87 95.35	344 73.35
	Total	25 5.33	444 94.67	469

The peak staff mean value breakpoint used for this contingency analysis is 21.65. Peak staff fails to reject the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.2773. The failure of rejection indicates that the relationship between MP Application Type groups and peak staff mean groups do not differ more than expected by chance alone. From the contingency analysis, it can be deduced that there no significant difference in peak staff means between the VC, TMDE, VP, SP, COM, RTE, SS Application Type group, and MP Application Type group. Table 46 shows the two sample test of proportions results for MP by Peak Staff.

Table 46: Application Type MP by Peak Staff Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(MP|Over 21.65)-P(MP|Under 21.65)	0.025488	-0.02248	0.081692

Adjusted Wald Test (Null Hypothesis)	Prob
P(MP|Over 21.65)-P(MP|Under 21.65) ≤ 0	0.1326
P(MP|Over 21.65)-P(MP|Under 21.65) ≥ 0	0.8674
P(MP|Over 21.65)-P(MP|Under 21.65) = 0	0.2652

The difference in proportions for Over 21.65 and Under 21.65 Peak Staff mean for the MP Application Type fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.2652. This failure to reject the null indicates the proportions are approximately equal.

Service

Table 47 shows the mosaic plot of Application Type MP groups by Service groups.

Table 47: Application Type MP Groups by Service Contingency Table

Service - Report Context	Application Type MP Groups		
Count Total % Col % Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
Air Force	21 4.52 84.00 17.65	98 21.08 22.27 82.35	119 25.59
Army	2 0.43 8.00 1.28	154 33.12 35.00 98.72	156 33.55
Navy	2 0.43 8.00 1.05	188 40.43 42.73 98.95	190 40.86
Total	25 5.38	440 94.62	465

Service rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of <0.0001. The rejection indicates that the relationship between MP Application Type groups and service groups differs more than expected by chance alone. From the contingency analysis, it can be deduced that the Air Force, Army, and Navy have a larger rate of VC, TMDE, VP, SP, COM, RTE, SS Application Type projects than MP Application Type projects. Also, it can be deduced that the Air Force has the highest rate of MP Application Type group projects at 84% versus the Army at 8% and the Navy at 8%. Table 48 shows the two sample test of proportions results for MP by Service. For this test, the services were consolidated to two cohorts: Air Force and Non-Air Force.

Table 48: Application Type MP by Service Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(MP|Air Force)-P(MP|Non-Air Force)	0.16491	0.0976	0.237301

Adjusted Wald Test (Null Hypothesis)	Prob
P(MP|Air Force)-P(MP|Non-Air Force) ≤ 0	<.0001*
P(MP|Air Force)-P(MP|Non-Air Force) ≥ 0	1.0000
P(MP|Air Force)-P(MP|Non-Air Force) = 0	<.0001*

The difference in proportions for Air Force and Non-Air Force Services for the MP Application Type rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of <0.0001. This rejection of the null indicates the proportions are not equal.

Application Type: C&C versus COM, VC, and SP

The purpose of this section is to analyze the results of our Command & Control Application Type contingency table analysis. Table 49 summarizes the key results from the C&C contingency analysis for the five variables of interest. Complete results can be found in Appendix C.

Table 49: Command & Control (C&C) Contingency Analysis Results

Command & Control (C&C) Contingency Analysis Results $\{\alpha = .10\}$			
Independent Variable	**Pearson Chi-Squared Test P-Value**	**Significant or Not Significant**	**N**
Primary Coding Language Conversion Factor	.0003	Significant	308
Development Process	.0135	Significant	308
Upgrade/New	.0040	Significant	308
Peak Staff	.4006	Not Significant	308
Service	.0664	Significant	306[2]

[2] The Service independent variable testing focuses on Air Force, Navy, and Army. The two excluded were comprised of Missile Defense Agency (MDA) and Department of Defense (DoD) programs.

Primary Coding Language Conversion Factor

Table 50 shows the mosaic plot of Application Type C&C groups by Primary Coding Language Conversion Factor groups.

Table 50: Application Type C&C Groups by Primary Coding Language Conversion Factor Contingency Table

Primary Coding Language	Count / Total % / Col % / Row %	C&C	COM, VC, SP	Total
	Group 1	35	26	61
		11.36	8.44	19.81
		30.43	13.47	
		57.38	42.62	
	Group 2	80	167	247
		25.97	54.22	80.19
		69.57	86.53	
		32.39	67.61	
	Total	115	193	308
		37.34	62.66	

Primary Coding Language Conversion Factor rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0003. The rejection indicates that the relationship between C&C Application Type groups and Primary Coding Language Conversion Factor groups differ more than expected by chance alone. From the contingency analysis, it can be deduced that the COM, VC, SP Application Type group uses a larger rate of Primary Coding Language Conversion Factors in Group 2 than Group 1. Also, it can be deduced that the C&C Application Type group uses a larger rate of Primary Coding Language Conversion Factors in Group 1 than Group 2. Projects in the C&C Application Type group use a larger rate of Group 1 language at 30.43% than

Programs in the COM, VC, SP Application Type group at 13.47%. Programs in the COM, VC, SP Application Type group use a larger rate of Group 2 language at 86.53% than Programs in the C&C Application Type group at 69.57%. Table 51 shows the two sample test of proportions results for C&C by Primary Coding Language Conversion Factor.

Table 51: Application Type C&C by Primary Coding Language Conversion Factor Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(C&C|Group 1)-P(C&C|Group 2)	0.249884	0.11078	0.381474

Adjusted Wald Test (Null Hypothesis)	Prob
P(C&C|Group 1)-P(C&C|Group 2) ≤ 0	0.0002*
P(C&C|Group 1)-P(C&C|Group 2) ≥ 0	0.9998
P(C&C|Group 1)-P(C&C|Group 2) = 0	0.0004*

The difference in proportions for Group 1 and Group 2 Primary Coding Language Conversion Factors for the C&C Application Type rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.0004. This rejection of the null indicates the proportions are not equal.

Development Process

Table 52 shows the mosaic plot of Application Type C&C groups by development process groups.

Table 52: Application Type C&C Groups by Development Process Contingency Table

	Application Type C&C Groups			
Development Process	Count Total % Col % Row %	C&C	COM, VC, SP	Total
	Agile	8	3	11
		2.60	0.97	3.57
		6.96	1.55	
		72.73	27.27	
	Non-Agile	107	190	297
		34.74	61.69	96.43
		93.04	98.45	
		36.03	63.97	
	Total	115	193	308
		37.34	62.66	

Development process rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0135. The rejection indicates that the relationship between C&C Application Type groups and development process groups differ more than expected by chance alone. From the contingency analysis, it can be deduced that the COM, VC, SP Application Type group uses a larger rate of Non-Agile development processes than Agile development processes. Also, it can be deduced that the C&C Application Type group uses a larger rate of Agile development processes than Non-Agile development processes. Programs in the C&C Application Type group use a larger rate of Agile development processes at 6.96% than programs in the COM, VC, SP Application Type group at 1.55%. Programs in the COM, VC, SP Application Type group use a larger rate of Non-Agile development processes at 98.45% than programs in the C&C Application Type group at 93.04%. Table 53 shows the two sample test of proportions results for C&C by Development Process.

Table 53: Application Type C&C by Development Process Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(C&C|Agile)-P(C&C|Non-Agile)	0.367003	0.074373	0.587834

Adjusted Wald Test (Null Hypothesis)	Prob
P(C&C|Agile)-P(C&C|Non-Agile) ≤ 0	0.0057*
P(C&C|Agile)-P(C&C|Non-Agile) ≥ 0	0.9943
P(C&C|Agile)-P(C&C|Non-Agile) = 0	0.0115*

The difference in proportions for Agile and Non-Agile Development Processes for the C&C Application Type rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.0115. This rejection of the null indicates the proportions are not equal.

Upgrade/New

Table 54 shows the mosaic plot of Application Type C&C groups by upgrade/new groups.

Table 54: Application Type C&C Groups by Upgrade/New Product and Development Descriptions Contingency Table

Upgrade/New - Product and Development Description	Count / Total % / Col % / Row %	Application Type C&C Groups: C&C	COM, VC, SP	Total
New	Count	76	95	171
	Total %	24.68	30.84	55.52
	Col %	66.09	49.22	
	Row %	44.44	55.56	
Upgrade	Count	39	98	137
	Total %	12.66	31.82	44.48
	Col %	33.91	50.78	
	Row %	28.47	71.53	
Total	Count	115	193	308
	Total %	37.34	62.66	

Upgrade/New rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0040. The rejection indicates that the relationship between C&C Application Type groups and upgrade/new product and development description groups differ more than expected by chance alone. It is important to note that Upgrade/New only shows as significant for the C&C cohort. From the contingency analysis, it can be deduced that the COM, VC, SP Application Type group has a larger rate of upgrade development projects than new development projects. Also, it can be deduced that the C&C Application Type group has a larger rate of new development projects than upgrade development projects. The C&C Application Type group has a larger rate of new development projects at 66.09% than programs in the COM, VC, SP Application Type group at 49.22%. Programs in the COM, VC, SP Application Type group has a larger rate of upgrade development projects processes at 50.78% than programs in the C&C Application Type group at 33.91%. Table 55 shows the two sample test of proportions results for C&C by Upgrade/New Product and Development Descriptions.

Table 55: Application Type C&C by Upgrade/New Product and Development Descriptions

Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(C&C|New)-P(C&C|Upgrade)	0.159773	0.05173	0.262904

Adjusted Wald Test (Null Hypothesis)	Prob
P(C&C|New)-P(C&C|Upgrade) ≤ 0	0.0017*
P(C&C|New)-P(C&C|Upgrade) ≥ 0	0.9983
P(C&C|New)-P(C&C|Upgrade) = 0	0.0035*

The difference in proportions for New and Upgrade Upgrade/New Product and Development Descriptions for the C&C Application Type rejects the null hypothesis of the

two-tailed Adjusted Wald test at a p-value of 0.0035. This rejection of the null indicates the proportions are not equal.

Peak Staff

Table 56 shows the mosaic plot of Application Type C&C groups by peak staff groups.

Table 56: Application Type C&C Groups by Peak Staff Contingency Table

Peak Staff Groups by Mean	Count / Total % / Col % / Row %	Application Type C&C Groups: C&C	COM, VC, SP	Total
Over 22.36	Count	38	55	93
	Total %	12.34	17.86	30.19
	Col %	33.04	28.50	
	Row %	40.86	59.14	
Under 22.36	Count	77	138	215
	Total %	25.00	44.81	69.81
	Col %	66.96	71.50	
	Row %	35.81	64.19	
Total	Count	115	193	308
	Total %	37.34	62.66	

The peak staff mean value breakpoint used for this contingency analysis is 22.36. Peak staff fails to reject the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.4006. The failure of rejection indicates that the relationship between C&C Application Type groups and peak staff mean groups do not differ more than expected by chance alone. From the contingency analysis, it can be deduced that there no significant difference in peak staff means between the COM, VC, SP Application Type group, and C&C Application Type group. Table 57 shows the two sample test of proportions results for C&C by Peak Staff.

Table 57: Application Type C&C by Peak Staff Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(C&C|Over 22.36)-P(C&C|Under 22.36)	0.050463	-0.06665	0.168813

Adjusted Wald Test (Null Hypothesis)	Prob
P(C&C|Over 22.36)-P(C&C|Under 22.36) ≤ 0	0.1976
P(C&C|Over 22.36)-P(C&C|Under 22.36) ≥ 0	0.8024
P(C&C|Over 22.36)-P(C&C|Under 22.36) = 0	0.3951

The difference in proportions for Over 22.36 and Under 22.36 Peak Staff mean for the C&C Application Type fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.3951. This failure to reject the null indicates the proportions are approximately equal.

Service

Table 58 shows the mosaic plot of Application Type C&C groups by service groups.

Table 58: Application Type C&C Groups by Service Contingency Table

Service - Report Context	Count / Total % / Col % / Row %	Application Type C&C Groups: C&C	COM, VC, SP	Total
Air Force	Count	31	32	63
	Total %	10.13	10.46	20.59
	Col %	26.96	16.75	
	Row %	49.21	50.79	
Army	Count	38	62	100
	Total %	12.42	20.26	32.68
	Col %	33.04	32.46	
	Row %	38.00	62.00	
Navy	Count	46	97	143
	Total %	15.03	31.70	46.73
	Col %	40.00	50.79	
	Row %	32.17	67.83	
Total	Count	115	191	306
	Total %	37.58	62.42	

Service rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0664. The rejection indicates that the relationship between C&C Application Type groups and service groups differs more than expected by chance alone. From the contingency analysis, it can be deduced that the Army and Navy have a larger rate of COM, VC, SP Application Type projects than C&C Application Type projects. Air Force is nearly evenly split between the two Application Type groups. The Navy has the highest rate of C&C Application Type group projects at 40% versus the Army at 33.04% and the Air Force at 26.96%. Table 59 shows the two sample test of proportions results for C&C by Service. For this test, the services were consolidated to two cohorts: Air Force and Non-Air Force.

Table 59: Application Type C&C by Service Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(C&C|Air Force)-P(C&C|Non-Air Force)	0.146384	0.010003	0.280735

Adjusted Wald Test (Null Hypothesis)	Prob
P(C&C|Air Force)-P(C&C|Non-Air Force) ≤ 0	0.0177*
P(C&C|Air Force)-P(C&C|Non-Air Force) ≥ 0	0.9823
P(C&C|Air Force)-P(C&C|Non-Air Force) = 0	0.0353*

The difference in proportions for Air Force and Non-Air Force Services for the C&C Application Type rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.0353. This rejection of the null indicates the proportions are not equal.

Super Domain: AIS versus ENG, MS, RT

The purpose of this section is to analyze the results of our Automated Information Systems Super Domain contingency table analysis. Table 60 summarizes the key results from the AIS contingency analysis for the five variables of interest. Complete results can be found in Appendix C.

Table 60: Automated Information Systems (AIS) Contingency Analysis Results

Automated Information Systems (AIS) Contingency Analysis Results $\{\alpha = .10\}$			
Independent Variable	**Pearson Chi-Squared Test P-Value**	**Significant or Not Significant**	**N**
Primary Coding Language Conversion Factor	.0168	Significant	655
Development Process	.0731	Significant	655
Upgrade/New	.4990	Not Significant	655
Peak Staff	.6628	Not Significant	655
Service	<.0001	Significant	650[3]

[3] The Service independent variable testing focuses on Air Force, Navy, and Army. The five excluded were comprised of Missile Defense Agency (MDA) and Department of Defense (DoD) programs.

Primary Coding Language Conversion Factor

Table 61 shows the mosaic plot of Super Domain Groups by Primary Coding Language Conversion Factor groups.

Table 61: Super Domain Groups by Primary Coding Language Conversion Factor Contingency Table

Primary Coding Language	Count / Total % / Col % / Row %	Super Domain Groups: AIS	Super Domain Groups: ENG, MS, RT	Total
Group 1	Count	13	104	117
	Total %	1.98	15.88	17.86
	Col %	31.71	16.94	
	Row %	11.11	88.89	
Group 2	Count	28	510	538
	Total %	4.27	77.86	82.14
	Col %	68.29	83.06	
	Row %	5.20	94.80	
Total	Count	41	614	655
	Total %	6.26	93.74	

Primary Coding Language Conversion Factor rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0168. The rejection indicates that the relationship between Super Domain groups and Primary Coding Language Conversion Factor groups differ more than expected by chance alone. From the contingency analysis it can be deduced that the ENG, MS, RT Super Domain group uses a larger rate of Primary Coding Language Conversion Factors in Group 2 than Group 1. Also, it can be deduced that the AIS Super Domain group uses a larger rate of Primary Coding Language Conversion Factors in Group 2 than Group 1. Programs in the AIS Super Domain group use a larger rate of Group 1 language at 31.71% than Programs in the ENG, MS, RT

Super Domain group at 16.94%. Programs in the ENG, MS, RT Super Domain group use a larger rate of Group 2 language at 83.06% than Programs in the AIS Super Domain group at 68.29%. Table 62 shows the two sample test of proportions results for Super Domain AIS by Primary Coding Language Conversion Factor.

Table 62: Super Domain AIS by Primary Coding Language Conversion Factor Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(AIS|Group 1)-P(AIS|Group 2)	0.059067	0.003013	0.124874

Adjusted Wald Test (Null Hypothesis)	Prob
P(AIS|Group 1)-P(AIS|Group 2) ≤ 0	0.0198*
P(AIS|Group 1)-P(AIS|Group 2) ≥ 0	0.9802
P(AIS|Group 1)-P(AIS|Group 2) = 0	0.0397*

The difference in proportions for Group 1 and Group 2 Primary Coding Language Conversion Factors for the AIS Super Domain rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.0397. This rejection of the null indicates the proportions are not equal.

Development Process

Table 63 shows the mosaic plot of Super Domain groups by development process groups.

Table 63: Super Domain Groups by Development Process Contingency Table

Development Process	Count / Total % / Col % / Row %	Super Domain Groups: AIS	ENG, MS, RT	Total
Agile	Count	4	24	28
	Total %	0.61	3.66	4.27
	Col %	9.76	3.91	
	Row %	14.29	85.71	
Non-Agile	Count	37	590	627
	Total %	5.65	90.08	95.73
	Col %	90.24	96.09	
	Row %	5.90	94.10	
Total	Count	41	614	655
	Total %	6.26	93.74	

Development process rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.0731. The rejection indicates that the relationship between Super Domain groups and development process groups differ more than expected by chance alone. From the contingency analysis, it can be deduced that the ENG, MS, RT Super Domain group uses a larger rate of Non-Agile development processes than Agile development processes. Also, it can be deduced that the AIS Super Domain group uses a larger rate of Non-Agile development processes than Agile development processes. Programs in the AIS Super Domain group use a larger rate of Agile development processes at 9.76% than programs in the ENG, MS, RT Super Domain group at 3.91%. Programs in the ENG, MS, RT Super Domain group use a larger rate of Non-Agile development processes at 96.09% than

programs in the AIS Super Domain group at 90.24%. Table 64 shows the two sample test of proportions results for Super Domain AIS by Development Process.

Table 64: Super Domain AIS by Development Process Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(AIS|Agile)-P(AIS|Non-Agile)	0.083846	-0.0284	0.240905

Adjusted Wald Test (Null Hypothesis)	Prob
P(AIS|Agile)-P(AIS|Non-Agile) ≤ 0	0.0610
P(AIS|Agile)-P(AIS|Non-Agile) ≥ 0	0.9390
P(AIS|Agile)-P(AIS|Non-Agile) = 0	0.1220

The difference in proportions for Agile and Non-Agile Development Processes for the AIS Super Domain fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.1220. This failure to reject the null indicates the proportions are approximately equal.

Upgrade/New

Table 65 shows the mosaic plot of Super Domain groups by upgrade/new groups.

Table 65: Super Domain Groups by Upgrade/New Product and Development Descriptions

Contingency Table

Upgrade/New - Product and Development Description		Super Domain Groups		
	Count Total % Col % Row %	AIS	ENG, MS, RT	Total
	New	19	318	337
		2.90	48.55	51.45
		46.34	51.79	
		5.64	94.36	
	Upgrade	22	296	318
		3.36	45.19	48.55
		53.66	48.21	
		6.92	93.08	
	Total	41	614	655
		6.26	93.74	

Upgrade/New fails to reject the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.4990. The failure of rejection indicates that the relationship between Super Domain groups and upgrade/new product and development description groups do not differ more than expected by chance alone. From the contingency analysis, it can be deduced that there no significant difference in upgrade/new product development descriptions between the ENG, MS, RT Super Domain group, and AIS Super Domain group. Table 66 shows the two sample test of proportions results for Super Domain AIS by Upgrade/New Product and Development Descriptions.

Table 66: Super Domain AIS by Upgrade/New Product and Development Descriptions Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(AIS|New)-P(AIS|Upgrade)	-0.0128	-0.05069	0.024936

Adjusted Wald Test (Null Hypothesis)	Prob
P(AIS|New)-P(AIS|Upgrade) ≤ 0	0.7478
P(AIS|New)-P(AIS|Upgrade) ≥ 0	0.2522
P(AIS|New)-P(AIS|Upgrade) = 0	0.5045

The difference in proportions for New and Upgrade Upgrade/New Product and Development Descriptions for the AIS Super Domain fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.5045. This rejection of the null indicates the proportions are approximately equal.

Peak Staff

Table 67 shows the mosaic plot of Super Domain groups by peak staff groups.

Table 67: Super Domain Groups by Peak Staff Contingency Table

Peak Staff Groups by Mean	Count / Total % / Col % / Row %	Super Domain Groups: AIS	ENG, MS, RT	Total
Over 21.65	Count	10	169	179
	Total %	1.53	25.80	27.33
	Col %	24.39	27.52	
	Row %	5.59	94.41	
Under 21.65	Count	31	445	476
	Total %	4.73	67.94	72.67
	Col %	75.61	72.48	
	Row %	6.51	93.49	
Total	Count	41	614	655
	Total %	6.26	93.74	

The peak staff mean value breakpoint used for this contingency analysis is 21.65. Peak staff fails to reject the null hypothesis of the Pearson Chi-Squared test at a p-value of 0.6628. The failure of rejection indicates that the relationship between Super Domain groups and peak staff mean groups do not differ more than expected by chance alone. From the contingency analysis, it can be deduced that there no significant difference in peak staff means between the ENG, MS, RT Super Domain group, and AIS Super Domain group. Table 68 shows the two sample test of proportions results for Super Domain AIS by Peak Staff.

Table 68: Super Domain AIS by Peak Staff Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(AIS|Over 21.65)-P(AIS|Under 21.65)	-0.00926	-0.04757	0.035222

Adjusted Wald Test (Null Hypothesis)	Prob
P(AIS|Over 21.65)-P(AIS|Under 21.65) ≤ 0	0.6150
P(AIS|Over 21.65)-P(AIS|Under 21.65) ≥ 0	0.3850
P(AIS|Over 21.65)-P(AIS|Under 21.65) = 0	0.7701

The difference in proportions for Over 21.65 and Under 21.65 Peak Staff mean for the AIS Super Domain fails to reject the null hypothesis of the two-tailed Adjusted Wald test at a p-value of 0.7701. This failure to reject the null indicates the proportions are approximately equal.

Service

Table 69 shows the mosaic plot of Super Domain groups by service groups.

Table 69: Super Domain Groups by Service Contingency Table

		Super Domain Groups		
Service - Report Context	Count Total % Col % Row %	AIS	ENG, MS, RT	Total
	Air Force	25	139	164
		3.85	21.38	25.23
		60.98	22.82	
		15.24	84.76	
	Army	12	208	220
		1.85	32.00	33.85
		29.27	34.15	
		5.45	94.55	
	Navy	4	262	266
		0.62	40.31	40.92
		9.76	43.02	
		1.50	98.50	
	Total	41	609	650
		6.31	93.69	

Service rejects the null hypothesis of the Pearson Chi-Squared test at a p-value of <0.0001. The rejection indicates that the relationship between Super Domain groups and service groups differs more than expected by chance alone. From the contingency analysis, it can be deduced that the Air Force, Army, and Navy have a larger rate of ENG, MS, RT Super Domain projects than AIS Super Domain projects. The Air Force has the highest rate of AIS Super Domain group projects at 60.98% versus the Army at 29.27% and the Navy at 9.76%. Table 70 shows the two sample test of proportions results for Super Domain AIS by Service. For this test, the services were consolidated to two cohorts: Air Force and Non-Air Force.

Table 70: Super Domain AIS by Service Two Sample Test for Proportions

Description	Proportion Difference	Lower 95%	Upper 95%
P(AIS|Air Force)-P(AIS|Non-Air Force)	0.119517	0.064158	0.179423

Adjusted Wald Test (Null Hypothesis)	Prob
P(AIS|Air Force)-P(AIS|Non-Air Force) ≤ 0	<.0001*
P(AIS|Air Force)-P(AIS|Non-Air Force) ≥ 0	1.0000
P(AIS|Air Force)-P(AIS|Non-Air Force) = 0	<.0001*

The difference in proportions for Air Force and Non-Air Force Services for the AIS Super Domain rejects the null hypothesis of the two-tailed Adjusted Wald test at a p-value of <0.0001. This rejection of the null indicates the proportions are not equal.

Summary

This chapter discussed the results and analysis performed in this research. The results were presented in three parts correlated with the research questions presented in Chapter I. The first part presents the Application Type analysis for Research Question #1 identifying statistically distinct Application Types. The second part presented the Super Domain analysis for Research Question #2 identifying statistically distinct Application Types. The third part presented the Contingency Table analysis and Two Sample Tests for Proportions for Research Question #3 identifying statistically significant characteristics of distinct Application Types and Super Domains. Chapter V will discuss the results as they apply to the proposed research questions and the software cost estimation field of study.

V. Conclusion

Overview

The purpose of this chapter is to summarize the results and findings presented in Chapter IV. The results for each research question are discussed in context to the significance within the software cost estimating field of study. Finally, recommendations for future research stemming from the findings of this study will be discussed.

Research Questions Answered

This research focused on Application Type and Super Domain productivity in defense software intensive projects. Using the results from Chapter IV, the three research questions from Chapter I can be answered. Results pertaining to the relevance of each research question follow.

Research Question #1: How predictive of productivity are Application Types in defense software intensive programs?

The first research question was answered through two one-way ANOVA tests using the 13 SRDR Application Types by Productivity (NAVAIR) and Productivity (AAF). The results of both Productivity (NAVAIR) and Productivity (AAF) indicated at least one pair of productivity means between Application Types were not equal. The Tukey-Kramer HSD test produced results (Table 24 and Table 26) showing that only 10 of 78 individual Application Type mean comparisons are significant for both measures of productivity. The two distinct Application Types were Mission Planning (MP), and Command & Control (C&C). These results are meaningful in that, broadly speaking,

Application Type does not appear to be statistically predictive of distinct productivity amongst software intensive programs. This is a pertinent finding to the field of software cost estimation as it contradicts the commonly accepted notion that Application Type matters to productivity and is a cost driver. The Software Cost Estimating Metric Manual for Defense Systems which also studied Application Type productivity developed 11 CERs which are currently accepted by the DoD. This current study indicates that CERs should not be imposed upon the data.

Nonetheless, two Application Types did appear distinct from others. Mission Planning's (MP) productivity mean value is significantly different than *seven* other Application Types from the SRDR dataset. Mission Planning (MP) is uniquely distinct from VC, TMDE, VP, SP, COM, RTE, and SS in productivity means. Command & Control's (C&C) productivity mean value is significantly different than *three* other Application Types from the SRDR dataset. Command & Control (C&C) is uniquely distinct from COM, VC, and SP in productivity means. Now that the statistically distinct Application Types have been recognized, we look to see what is specifically distinct about them in Research Question #3.

Research Question #2: How predictive of productivity are Super Domains in defense software intensive programs?

The second research question was also answered through two one-way ANOVA tests using the four SRDR Super Domains by Productivity (NAVAIR) and Productivity (AAF). The results of both Productivity (NAVAIR) and Productivity (AAF) indicated at least one pair of productivity means between Super Domains were not equal. The Tukey-

Kramer HSD test produced results (Table 33 and Table 35) showing that only 3 of 6 individual Super Domain means comparisons were significant for both measures of productivity. The major finding from this test shows that the Automated Information Systems (AIS) productivity mean value is significantly different than all three other Super Domains from the SRDR dataset. These results are meaningful in that it appears that Super Domain is not broadly predictive of productivity amongst software intensive programs. Outside of the three significant Super Domain mean comparisons recognized in the Tukey-Kramer HSD results, only one out of four Super Domain means, Automated Information Systems (AIS), was distinct. From this we can conclude individual Super Domain Cost Estimating Relationships (CERs) for Engineering (ENG), Mission Support (MS), and Real Time (RT) are likely homogenous and thus not uniquely predictive of productivity. Now that the statistically distinct Super Domain has been recognized, we look to see what is specifically distinct about them in Research Question #3.

AIS does appear to be distinct from the other three Super Domains and appears to be uniquely predictive of productivity. This is not proof that Super Domain is not relevant in predicting productivity but rather implies Super Domains may not be universally applicable in software cost estimating. The Software Cost Estimating Metric Manual for Defense Systems does not study Super Domain using the SRDR dataset, but productivity effects were studied in the DoD Software Factbook (2015). Our findings affirm the DoD Software Factbook's findings that AIS has the highest average production rate of 1.1 months per KESLOC (Clark, McCurley, & Zubrow, 2015). The AIS Super Domain is a larger super grouping of like software projects, one of which being Mission Planning (MP). MP appears to be so significantly different than other Application Types that it

made the entire AIS Super Domain productivity mean show as statistically significant in the ANOVA analysis. Since the DoD is mapping Super Domains to individual CSCIs within the SRDR database, it will be important to further study Super Domain and its significance as a cost driver or predictor of productivity.

Research Question #3: Among Application Types and Super Domains that show statistically distinct productivity effects, which characteristics are statistically significant?

The third research question was answered using contingency table analysis to identify distinct characteristics of Application Type and Super Domain categories that tested as significant in Research Questions #1 and #2. Contingency table analysis was used due to both the dependent and independent variables being categorical. The contingency tables show the frequency distribution in a matrix form of two categorical variables. In this case, the categorical variables are the Application Types and Super Domains cohorts and the five chosen independent variables. The significance of this analysis is to imply potential underlying variables influencing distinct productivity effects. Characteristics identified through the contingency analysis are summarized in Table 71.

Table 71: Significant Characteristics Contingency Analysis Results

Significant Characteristics Contingency Analysis Results $\{\alpha = .10\}$			
Mission Planning (MP) Application Type			
Independent Variable	**Pearson Chi-Squared Test P-Value**	**Significant or Not Significant**	**N**
Primary Coding Language Conversion Factor	0.0396	Significant	469
Development Process	0.0011	Significant	469
Service	<0.0001	Significant	465
Command & Control (C&C) Application Type			
Primary Coding Language Conversion Factor	0.0003	Significant	308
Upgrade/New	0.0040	Significant	308
Development Process	0.0135	Significant	308
Service	0.0664	Significant	306
Automated Information Systems (AIS) Super Domain			
Primary Coding Language Conversion Factor	0.0168	Significant	655
Development Process	0.0731	Significant	655
Service	<0.0001	Significant	650

The contingency table results were compiled using an alpha level of 0.10. This elevated alpha level allowed for a more robust analysis of the significant variables as some were borderline at alpha 0.05. The Primary Coding Language Conversion Factor, Development Process, and Service characteristics were significant for Mission Planning (MP), Command & Control (C&C), and Automated Information Systems (AIS). Command & Control (C&C) also showed significance for the upgrade/new product and

development description characteristics. In other words, each has a disproportionate occurrence of these characteristics.

To start, in the MP Application Type Primary Coding Language Conversion Factor analysis, overall MP showed a disproportionate rate of using Group 1 Primary Coding Language Conversion Factors at 28% versus the significantly different Application Type group of VC, TMDE, VP, SP, COM, RTE, SS which only had a Group 1 Primary Coding Language Conversion Factor usage rate of 13.29%. In the MP development process analysis, MP showed a disproportionate rate of using Agile development at 16% versus the significantly different Application Type group of VC, TMDE, VP, SP, COM, RTE, SS which only had an Agile development usage rate of 3.15%. In the MP military service analysis, MP showed a disproportionate rate of Air Force as the service designation at 84% versus the significantly different Application Type group of VC, TMDE, VP, SP, COM, RTE, SS which only had Air Force as the service designation at a rate of 22.7%.

Next, in the C&C Application Type Primary Coding Language Conversion Factor analysis, C&C showed a disproportionate rate of using Group 1 Primary Coding Language Conversion Factors at 30.43% versus the significantly different Application Type group of COM, VC, SP which only had a Group 1 Primary Coding Language Conversion Factor usage rate of 13.47%. In the C&C development process analysis, C&C showed a disproportionate rate of using Agile development at 6.96% versus the significantly different Application Type group of COM, VC, SP which only had an Agile development usage rate of 1.55%. In the C&C Upgrade/New product development description analysis, C&C showed a disproportionate rate of New product development at

66.09% versus the significantly different Application Type group of COM, VC, SP which only had New product and development descriptions rate of 49.22%. In the C&C military service analysis, C&C showed a disproportionate rate of Air Force as the service designation at 26.96% versus the significantly different Application Type group of COM, VC, SP which only had Air Force as the service designation at a rate of 16.75%.

Finally, in the AIS Super Domain Primary Coding Language Conversion Factor analysis, overall AIS showed a disproportionate rate of using Group 1 Primary Coding Language Conversion Factors at 31.71% versus the significantly different Super Domain group of ENG, MS, RT which only had a Group 1 Primary Coding Language Conversion Factor usage rate of 16.94%. In the AIS development process analysis, AIS showed a disproportionate rate of using Agile development at 9.76% versus the significantly different Super Domain group of ENG, MS, RT which only had an Agile development usage rate of 3.91%. In the AIS military service analysis, AIS showed a disproportionate rate of Air Force as the service designation at 60.98% versus the significantly different Super Domain group of ENG, MS, RT which only had New product and development descriptions rate of 22.82%.

These significant characteristics of the Application Types and Super Domains provided shed light on underlying attributes which may be driving the distinct productivity effects of MP, C&C, and AIS rather than the grouping itself being the catalyst. Software categories such as Application Type and Super Domain have been historically touted as relevant cost drivers in software intensive projects; however, the characteristics which make these categories relevant are empirically lacking. This lack is unfortunate because Application Type is ultimately a proxy used to discover what

common characteristics each software grouping has. This research question is intended to spark future study and interest in what is driving productive Application Type and Super Domains rather than just affirming that the categories themselves are representative of increased productivity.

Recommendations for Future Research

The statistical analysis presented in this study should provoke further study on Application Type and Super Domain's productivity effects. It would be of use to investigate the reasons why statistically distinct Application Types and Super Domains are unique beyond the scope of this study. This study was limited to looking at only five characteristics: primary coding language conversion factor, development process, upgrade or new development, peak staff level, and military service branch. The productivity of individual projects may be heavily influenced by soft factors that were not accounted for in the SRDR dataset. The soft factors of individual contractor environments that influence the productivity of projects would also be an influential undertaking that could help expand areas are not in the scope of this study. Furthermore, the mapping of Super Domain and Application Type to defense projects should be studied. The current Application Type and Super Domain mapping method relies heavily on analyst interpretation of subjective SRDR submission details, supporting comments, and data dictionaries given for individual CSCIs.

Summary

The objective of this study was to analyze the significance of Application Type and Super Domain productivity predictability in software intensive defense acquisition programs.

Clark and Madachy's work within the defense software cost estimating domain and the Software Cost Estimating Metric Manual for Defense Systems indicated that Application Type was predictive of productivity and cost (Clark & Madachy, 2015). This study does not corroborate their findings. The disagreement, unfortunately, may not be one of different data sets but of standards of the statistical tests employed. There is no indication that Clark and Madachy conducted the necessary ANOVA testing to validate their CERs and productivity benchmarks, and it appears that they needed to.

The findings of this current study explore the unique predictability of software groupings by way of Application Type and Super Domain and recognized characteristics that underlie distinctly productive software project groupings. The findings also suggest that the previously accepted software groupings such as Application Type and Super Domain may not be as relevant to predicting the software productivity and cost as previous literature states. But, rather, the factors and characteristics behind such software groupings may have a larger influence on productivity than the grouping itself. The findings presented suggest that the current use of Application Type CERs may be less accurate than initially thought. These CERs based on Application Type appear to not be broadly predictive of software development cost by way of associated productivity. The contingency analysis also shed light on relevant productivity drivers behind the distinct Application Type and Super Domain. Software productivity cost estimation remains a complex and ever-evolving arena, this study is a small piece of the puzzle aimed at building upon the existing established literature, and empirical analysis in benchmarking and building productivity CERs by way of Application Type and Super Domain.

Appendix A

Figure A-1: KESLOC Software Development Language Conversion Factors

	NEW	Modified	Reuse	Autocode
Ada	100%	15%	7%	32%
Ada, C, C++, VHDL	100%	15%	7%	32%
VHDL	100%	15%	7%	32%
Ada 95	100%	15%	7%	32%
Ada 83	100%	15%	7%	32%
Ansi C	100%	15%	7%	32%
ANSI C/C++	100%	15%	7%	32%
Assembly	100%	50%	5%	32%
Basic	100%	50%	5%	32%
C	100%	15%	7%	32%
C (ANSI C)	100%	15%	7%	32%
C and FORTRAN	100%	15%	7%	32%
C#	100%	50%	5%	32%
C#, .Net	100%	50%	5%	32%
C, C#	100%	15%	7%	32%
C/Assembly	100%	15%	7%	32%
C/C++	100%	15%	7%	32%
C/C++/C#	100%	15%	7%	32%
C++	100%	15%	7%	32%
CPP	100%	50%	5%	32%
EC++	100%	15%	7%	32%
COMET	100%	50%	5%	32%
FORTRAN	100%	50%	5%	32%
Eiffel	100%	50%	5%	32%
HLB	100%	50%	5%	32%
INNO	100%	50%	5%	32%
J2EE	100%	50%	5%	32%
Java	100%	50%	5%	32%
Java, C/C++	100%	50%	5%	32%
Java/JavaScript	100%	50%	5%	32%
Javascript	100%	50%	5%	32%
Jovial	100%	50%	5%	32%
LabWindows CVI	100%	50%	5%	32%
LUCOL	100%	50%	5%	32%
MATLAB	100%	50%	5%	32%
MDA/C++	100%	50%	5%	32%
Other	100%	50%	5%	32%
PLC	100%	50%	5%	32%
PRL	100%	50%	5%	32%
PV Wave	100%	50%	5%	32%
Python	100%	50%	5%	32%
Scala	100%	50%	5%	32%
Script	100%	50%	5%	32%
Simulink	100%	50%	5%	32%
UML	100%	50%	5%	32%
UML/C++	100%	50%	5%	32%
UNIX/Windows Scripts	100%	50%	5%	32%
VB.Net	100%	50%	5%	32%
Visual Basic	100%	50%	5%	32%
XML	100%	50%	5%	32%
Blank (SUMPRODUCT)	100%	15%	7%	32%

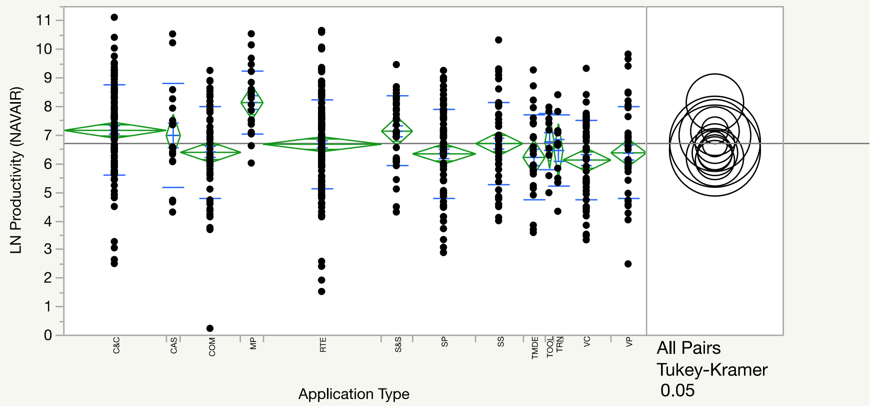
Figure A-2: KESLOC Software Development Language Groups (Contingency Analysis)

		NEW	Modified	Reuse	Autocode
Group 1	Assembly	100%	50%	5%	32%
	Basic	100%	50%	5%	32%
	C#	100%	50%	5%	32%
	C#, .Net	100%	50%	5%	32%
	CPP	100%	50%	5%	32%
	COMET	100%	50%	5%	32%
	FORTRAN	100%	50%	5%	32%
	Eiffel	100%	50%	5%	32%
	HLB	100%	50%	5%	32%
	INNO	100%	50%	5%	32%
	J2EE	100%	50%	5%	32%
	Java	100%	50%	5%	32%
	Java, C/C++	100%	50%	5%	32%
	Java/JavaScript	100%	50%	5%	32%
	Javascript	100%	50%	5%	32%
	Jovial	100%	50%	5%	32%
	LabWindows CVI	100%	50%	5%	32%
	LUCOL	100%	50%	5%	32%
	MATLAB	100%	50%	5%	32%
	MDA/C++	100%	50%	5%	32%
	Other	100%	50%	5%	32%
	PLC	100%	50%	5%	32%
	PRL	100%	50%	5%	32%
	PV Wave	100%	50%	5%	32%
	Python	100%	50%	5%	32%
	Scala	100%	50%	5%	32%
	Script	100%	50%	5%	32%
	Simulink	100%	50%	5%	32%
	UML	100%	50%	5%	32%
	UML/C++	100%	50%	5%	32%
	UNIX/Windows Scripts	100%	50%	5%	32%
	VB.Net	100%	50%	5%	32%
	Visual Basic	100%	50%	5%	32%
	XML	100%	50%	5%	32%
Group 2	Ada	100%	15%	7%	32%
	Ada, C, C++, VHDL	100%	15%	7%	32%
	VHDL	100%	15%	7%	32%
	Ada 95	100%	15%	7%	32%
	Ada 83	100%	15%	7%	32%
	Ansi C	100%	15%	7%	32%
	ANSI C/C++	100%	15%	7%	32%
	C	100%	15%	7%	32%
	C (ANSI C)	100%	15%	7%	32%
	C and FORTRAN	100%	15%	7%	32%
	C, C#	100%	15%	7%	32%
	C/Assembly	100%	15%	7%	32%
	C/C++	100%	15%	7%	32%
	C/C++/C#	100%	15%	7%	32%
	C++	100%	15%	7%	32%
	EC++	100%	15%	7%	32%
	Blank (SUMPRODUCT)	100%	15%	7%	32%

Appendix B

Figure B-1: One-Way ANOVA of LN Productivity (NAVAIR) by Application Type

Oneway Analysis of LN Productivity (NAVAIR) By Application Type



Oneway Anova

Summary of Fit

Rsquare	0.0801
Adj Rsquare	0.062905
Root Mean Square Error	1.512963
Mean of Response	6.711969
Observations (or Sum Wgts)	655

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Application Type	12	127.9620	10.6635	4.6585	<.0001*
Error	642	1469.5749	2.2891		
C. Total	654	1597.5369			

Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
C&C	115	7.17218	0.14108	6.8951	7.4492
CAS	16	6.99394	0.37824	6.2512	7.7367
COM	68	6.40463	0.18347	6.0444	6.7649
MP	25	8.14240	0.30259	7.5482	8.7366
RTE	133	6.68673	0.13119	6.4291	6.9443
S&S	35	7.14406	0.25574	6.6419	7.6462
SP	71	6.34966	0.17956	5.9971	6.7022
SS	53	6.70908	0.20782	6.3010	7.1172
TMDE	25	6.22880	0.30259	5.6346	6.8230
TOOL	10	6.77530	0.47844	5.8358	7.7148
TRN	10	6.46570	0.47844	5.5262	7.4052
VC	54	6.13302	0.20589	5.7287	6.5373
VP	40	6.38660	0.23922	5.9169	6.8563

Std Error uses a pooled estimate of error variance

Means and Std Deviations

Level	Number	Mean	Std Dev	Std Err Mean
C&C	115	7.1721826	1.5800624	0.1473416
CAS	16	6.9939375	1.8104904	0.4526226
COM	68	6.4046324	1.6173865	0.1961369
MP	25	8.1424	1.1211893	0.2242379
RTE	133	6.6867293	1.5719312	0.1363037
S&S	35	7.1440571	1.2233693	0.2067871
SP	71	6.349662	1.5398124	0.1827421
SS	53	6.7090755	1.4580935	0.2002845
TMDE	25	6.2288	1.494631	0.2989262
TOOL	10	6.7753	0.9844328	0.311305
TRN	10	6.4657	1.235295	0.3906346
VC	54	6.1330185	1.3846203	0.188423
VP	40	6.3866	1.5940481	0.2520411

Means Comparisons

Comparisons for all pairs using Tukey-Kramer HSD

Confidence Quantile

q*	Alpha
3.32536	0.05

HSD Threshold Matrix

Abs(Dif)-HSD

	MP	C&C	S&S	CAS	TOOL	SS	RTE	TRN	COM	VP	SP	TMDE	VC
MP	-1.4230	-0.1400	-0.3191	-0.4623	-0.5154	0.2126	0.3589	-0.2058	0.5610	0.4731	0.6227	0.4906	0.7923
C&C	-0.1400	-0.6635	-0.9431	-1.1642	-1.2618	-0.3722	-0.1552	-0.9522	-0.0021	-0.1380	0.0632	-0.1668	0.2092
S&S	-0.3191	-0.9431	-1.2027	-1.3682	-1.4353	-0.6608	-0.4985	-1.1257	-0.3072	-0.4070	-0.2447	-0.4022	-0.0807
CAS	-0.4623	-1.1642	-1.3682	-1.7788	-1.8095	-1.1503	-1.0241	-1.4999	-0.8086	-0.8809	-0.7480	-0.8456	-0.5711
TOOL	-0.5154	-1.2618	-1.4353	-1.8095	-2.2500	-1.6684	-1.5611	-1.9404	-1.3333	-1.3901	-1.2737	-1.3360	-1.0898
SS	0.2126	-0.3722	-0.6608	-1.1503	-1.6684	-0.9773	-0.7949	-1.4912	-0.6174	-0.7313	-0.5539	-0.7404	-0.3967
RTE	0.3589	-0.1552	-0.4985	-1.0241	-1.5611	-0.7949	-0.6170	-1.4287	-0.4679	-0.6071	-0.4024	-0.6388	-0.2581
TRN	-0.2058	-0.9522	-1.1257	-1.4999	-1.9404	-1.4912	-1.4287	-2.2500	-1.6429	-1.6997	-1.5833	-1.6456	-1.3994
COM	0.5610	-0.0021	-0.3072	-0.8086	-1.3333	-0.6174	-0.4679	-1.6429	-0.8628	-0.9845	-0.7987	-1.0009	-0.6454
VP	0.4731	-0.1380	-0.4070	-0.8809	-1.3901	-0.7313	-0.6071	-1.6997	-0.9845	-1.1250	-0.9577	-1.1249	-0.7960
SP	0.6227	0.0632	-0.2447	-0.7480	-1.2737	-0.5539	-0.4024	-1.5833	-0.7987	-0.9577	-0.8444	-1.0492	-0.6918
TMDE	0.4906	-0.1668	-0.4022	-0.8456	-1.3360	-0.7404	-0.6388	-1.6456	-1.0009	-1.1249	-1.0492	-1.4230	-1.1213
VC	0.7923	0.2092	-0.0807	-0.5711	-1.0898	-0.3967	-0.2581	-1.3994	-0.6454	-0.7960	-0.6918	-1.1213	-0.9682

Positive values show pairs of means that are significantly different.

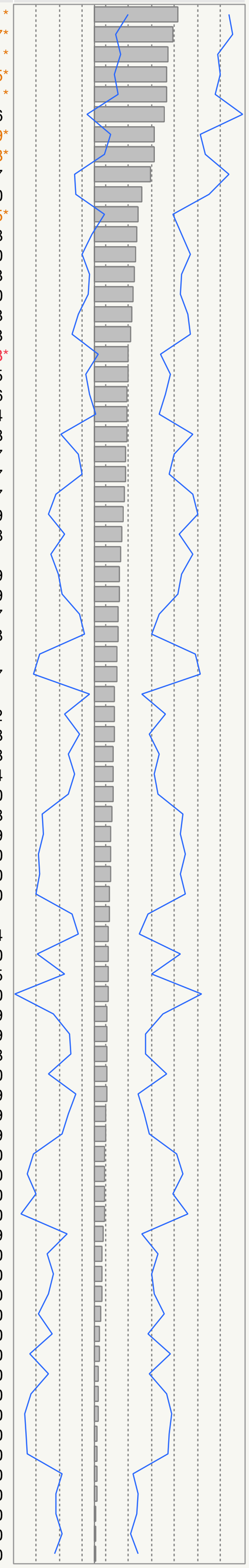
Connecting Letters Report

Level				Mean
MP	A			8.1424000
C&C	A	B		7.1721826
S&S	A	B	C	7.1440571
CAS	A	B	C	6.9939375
TOOL	A	B	C	6.7753000
SS		B	C	6.7090755
RTE		B	C	6.6867293
TRN	A	B	C	6.4657000
COM		B	C	6.4046324
VP		B	C	6.3866000
SP			C	6.3496620
TMDE		B	C	6.2288000
VC			C	6.1330185

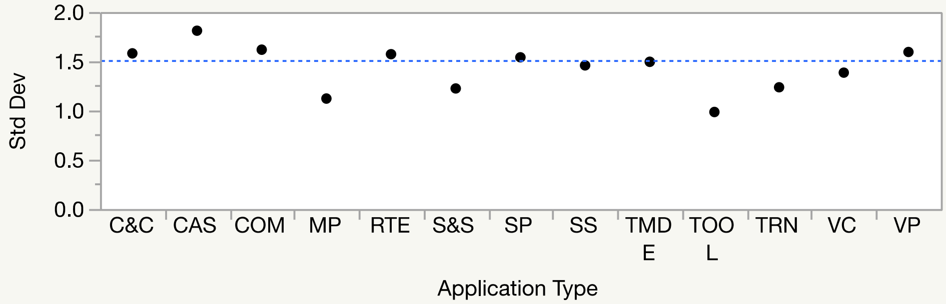
Levels not connected by same letter are significantly different.

Ordered Differences Report

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
MP	VC	2.009381	0.3659949	0.79232	3.226446	<.0001*
MP	TMDE	1.913600	0.4279306	0.49058	3.336624	0.0007*
MP	SP	1.792738	0.3518559	0.62269	2.962786	<.0001*
MP	VP	1.755800	0.3857314	0.47310	3.038496	0.0005*
MP	COM	1.737768	0.3538713	0.56102	2.914517	<.0001*
MP	TRN	1.676700	0.5660990	-0.20578	3.559183	0.1386
MP	RTE	1.455671	0.3298079	0.35894	2.552401	0.0009*
MP	SS	1.433325	0.3670859	0.21263	2.654018	0.0068*
MP	TOOL	1.367100	0.5660990	-0.51538	3.249583	0.4347
MP	CAS	1.148462	0.4843845	-0.46229	2.759216	0.4660
C&C	VC	1.039164	0.2495893	0.20919	1.869138	0.0025*
S&S	VC	1.011039	0.3283163	-0.08073	2.102809	0.1018
MP	S&S	0.998343	0.3961867	-0.31912	2.315807	0.3630
MP	C&C	0.970217	0.3338670	-0.14001	2.080445	0.1593
C&C	TMDE	0.943383	0.3338670	-0.16685	2.053611	0.1930
S&S	TMDE	0.915257	0.3961867	-0.40221	2.232721	0.5103
CAS	VC	0.860919	0.4306461	-0.57113	2.292973	0.7333
C&C	SP	0.822521	0.2283530	0.06316	1.581877	0.0203*
S&S	SP	0.794395	0.3124770	-0.24470	1.833494	0.3485
C&C	VP	0.785583	0.2777252	-0.13795	1.709119	0.1916
C&C	COM	0.767550	0.2314464	-0.00209	1.537193	0.0514
CAS	TMDE	0.765138	0.4843845	-0.84562	2.375891	0.9358
S&S	VP	0.757457	0.3501829	-0.40703	1.921942	0.6187
S&S	COM	0.739425	0.3147447	-0.30721	1.786065	0.4817
C&C	TRN	0.706483	0.4988092	-0.95224	2.365203	0.9717
S&S	TRN	0.678357	0.5425010	-1.12565	2.482369	0.9899
CAS	SP	0.644276	0.4186960	-0.74804	2.036591	0.9468
TOOL	VC	0.642281	0.5208605	-1.08977	2.374331	0.9911
CAS	VP	0.607338	0.4475405	-0.88090	2.095571	0.9799
CAS	COM	0.589305	0.4203911	-0.80865	1.987257	0.9739
SS	VC	0.576057	0.2925401	-0.39674	1.548858	0.7527
RTE	VC	0.553711	0.2441330	-0.25812	1.365541	0.5413
TOOL	TMDE	0.546500	0.5660990	-1.33598	2.428983	0.9991
CAS	TRN	0.528238	0.6098949	-1.49988	2.556358	0.9997
C&C	RTE	0.485453	0.1926546	-0.15519	1.126099	0.3631
SS	TMDE	0.480275	0.3670859	-0.74042	1.700969	0.9852
C&C	SS	0.463107	0.2511864	-0.37218	1.298393	0.8268
RTE	TMDE	0.457929	0.3298079	-0.63880	1.554660	0.9758
S&S	RTE	0.457328	0.2874241	-0.49846	1.413117	0.9324
S&S	SS	0.434982	0.3295321	-0.66083	1.530795	0.9840
TOOL	SP	0.425638	0.5110245	-1.27370	2.124979	0.9998
C&C	TOOL	0.396883	0.4988092	-1.26184	2.055603	0.9999
TOOL	VP	0.388700	0.5349132	-1.39008	2.167479	1.0000
TOOL	COM	0.370668	0.5124142	-1.33329	2.074630	1.0000
S&S	TOOL	0.368757	0.5425010	-1.43525	2.172769	1.0000
SS	SP	0.359413	0.2746452	-0.55388	1.272708	0.9851
RTE	SP	0.337067	0.2223763	-0.40241	1.076549	0.9524
TRN	VC	0.332681	0.5208605	-1.39937	2.064731	1.0000
SS	VP	0.322475	0.3168852	-0.73128	1.376233	0.9985
TOOL	TRN	0.309600	0.6766177	-1.94040	2.559598	1.0000
CAS	RTE	0.307208	0.4003462	-1.02409	1.638504	0.9999
SS	COM	0.304443	0.2772226	-0.61742	1.226308	0.9969
RTE	VP	0.300129	0.2728322	-0.60714	1.207395	0.9968
CAS	SS	0.284862	0.4315737	-1.15028	1.720000	1.0000
RTE	COM	0.282097	0.2255517	-0.46794	1.032138	0.9899
COM	VC	0.271614	0.2757763	-0.64544	1.188669	0.9989
VP	VC	0.253581	0.3156206	-0.79597	1.303134	0.9999
SS	TRN	0.243375	0.5216277	-1.49122	1.977976	1.0000
TRN	TMDE	0.236900	0.5660990	-1.64558	2.119383	1.0000
RTE	TRN	0.221029	0.4961015	-1.42869	1.870746	1.0000
CAS	TOOL	0.218638	0.6098949	-1.80948	2.246758	1.0000
SP	VC	0.216643	0.2731853	-0.69180	1.125083	0.9999
C&C	CAS	0.178245	0.4036966	-1.16419	1.520682	1.0000
COM	TMDE	0.175832	0.3538713	-1.00092	1.352582	1.0000
VP	TMDE	0.157800	0.3857314	-1.12490	1.440496	1.0000
S&S	CAS	0.150120	0.4565827	-1.36818	1.668422	1.0000
SP	TMDE	0.120862	0.3518559	-1.04919	1.290910	1.0000
TRN	SP	0.116038	0.5110245	-1.58330	1.815379	1.0000
TMDE	VC	0.095781	0.3659949	-1.12128	1.312846	1.0000
TOOL	RTE	0.088571	0.4961015	-1.56115	1.738287	1.0000
TRN	VP	0.079100	0.5349132	-1.69968	1.857879	1.0000
TOOL	SS	0.066225	0.5216277	-1.66838	1.800825	1.0000
TRN	COM	0.061068	0.5124142	-1.64289	1.765030	1.0000
COM	SP	0.054970	0.2567155	-0.79870	0.908642	1.0000
VP	SP	0.036938	0.2991098	-0.95771	1.031586	1.0000
C&C	S&S	0.028125	0.2920727	-0.94312	0.999373	1.0000
SS	RTE	0.022346	0.2457656	-0.79491	0.839605	1.0000
COM	VP	0.018032	0.3014781	-0.98449	1.020556	1.0000



Tests that the Variances are Equal



Level	Count	Std Dev	MeanAbsDif to Mean	MeanAbsDif to Median
C&C	115	1.580062	1.253530	1.251122
CAS	16	1.810490	1.420930	1.367313
COM	68	1.617387	1.250177	1.239191
MP	25	1.121189	0.891744	0.886000
RTE	133	1.571931	1.178621	1.176406
S&S	35	1.223369	0.972059	0.971686
SP	71	1.539812	1.288493	1.285817
SS	53	1.458093	1.165999	1.164377
TMDE	25	1.494631	1.190672	1.179120
TOOL	10	0.984433	0.801500	0.801500
TRN	10	1.235295	0.985360	0.952700
VC	54	1.384620	1.157942	1.150944
VP	40	1.594048	1.234030	1.221000

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.7373	12	642	0.7153
Brown-Forsythe	0.7544	12	642	0.6979
Levene	0.8201	12	642	0.6298
Bartlett	1.0837	12	.	0.3687

Welch's Test

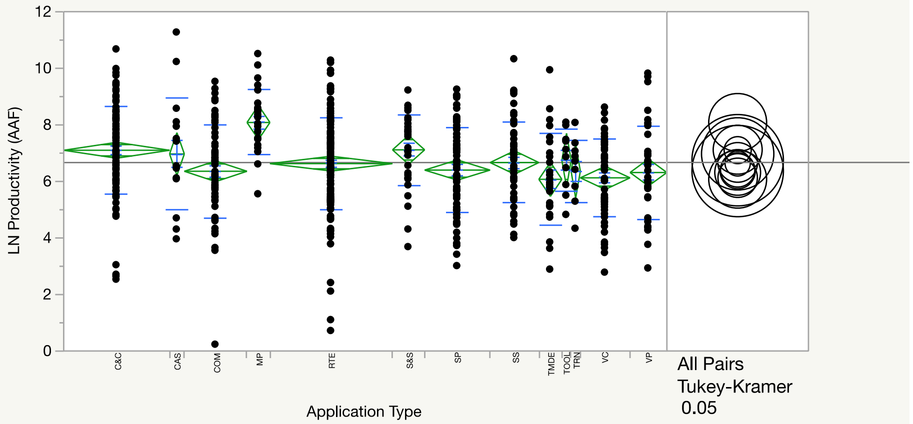
Welch Anova testing Means Equal, allowing Std Devs Not Equal

F Ratio	DFNum	DFDen	Prob > F
5.9085	12	120.54	<.0001*

Excluded Rows 2

Figure B-2: One-Way ANOVA of LN Productivity (AAF) by Application Type

Oneway Analysis of LN Productivity (AAF) By Application Type



Oneway Anova

Summary of Fit

Rsquare	0.074928
Adj Rsquare	0.057637
Root Mean Square Error	1.536508
Mean of Response	6.668113
Observations (or Sum Wgts)	655

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Application Type	12	122.7651	10.2304	4.3334	<.0001*
Error	642	1515.6697	2.3609		
C. Total	654	1638.4348			

Means for Oneway Anova

Level	Number	Mean	Std Error
C&C	115	7.10164	0.14328
CAS	16	6.97081	0.38413
COM	68	6.35759	0.18633
MP	25	8.08332	0.30730
RTE	133	6.62984	0.13323
S&S	35	7.11689	0.25972
SP	71	6.40210	0.18235
SS	53	6.66726	0.21106
TMDE	25	6.07332	0.30730
TOOL	10	6.74630	0.48589
TRN	10	6.36180	0.48589
VC	54	6.12509	0.20909
VP	40	6.31375	0.24294

Std Error uses a pooled estimate of error variance

Means and Std Deviations

Level	Number	Mean	Std Dev	Std Err Mean
C&C	115	7.1016435	1.5580142	0.1452856
CAS	16	6.9708125	1.9784372	0.4946093
COM	68	6.3575882	1.6504766	0.2001497
MP	25	8.08332	1.1482566	0.2296513
RTE	133	6.6298421	1.6400869	0.1422136
S&S	35	7.1168857	1.254468	0.2120438
SP	71	6.4020986	1.5089794	0.1790829
SS	53	6.6672642	1.4289676	0.1962838
TMDE	25	6.07332	1.6301966	0.3260393
TOOL	10	6.7463	1.0812368	0.3419171
TRN	10	6.3618	1.0871681	0.3437927
VC	54	6.1250926	1.375831	0.1872269
VP	40	6.31375	1.6538067	0.2614898

Means Comparisons

Comparisons for all pairs using Tukey-Kramer HSD

Confidence Quantile

q*	Alpha
3.32536	0.05

HSD Threshold Matrix

Abs(Dif)-HSD

	MP	S&S	C&C	CAS	TOOL	SS	RTE	SP	TRN	COM	VP	VC	TMDE
MP	-1.4452	-0.3715	-0.1458	-0.5233	-0.5748	0.1764	0.3397	0.4930	-0.1903	0.5307	0.4669	0.7222	0.5648
S&S	-0.3715	-1.2214	-0.9711	-1.3959	-1.4615	-0.6632	-0.4836	-0.3405	-1.0770	-0.3036	-0.3795	-0.1170	-0.2944
C&C	-0.1458	-0.9711	-0.6738	-1.2325	-1.3292	-0.4139	-0.1788	-0.0716	-0.9447	-0.0376	-0.1500	0.1337	-0.0992
CAS	-0.5233	-1.3959	-1.2325	-1.8065	-1.8352	-1.1539	-1.0110	-0.8453	-1.4507	-0.8065	-0.8543	-0.6086	-0.7383
TOOL	-0.5748	-1.4615	-1.3292	-1.8352	-2.2850	-1.6826	-1.5589	-1.3816	-1.9005	-1.3418	-1.3739	-1.1378	-1.2388
SS	0.1764	-0.6632	-0.4139	-1.1539	-1.6826	-0.9925	-0.7926	-0.6623	-1.4561	-0.6265	-0.7166	-0.4458	-0.6457
RTE	0.3397	-0.4836	-0.1788	-1.0110	-1.5589	-0.7926	-0.6266	-0.5232	-1.4073	-0.4895	-0.6053	-0.3197	-0.5573
SP	0.4930	-0.3405	-0.0716	-0.8453	-1.3816	-0.6623	-0.5232	-0.8575	-1.6855	-0.8224	-0.9218	-0.6456	-0.8595
TRN	-0.1903	-1.0770	-0.9447	-1.4507	-1.9005	-1.4561	-1.4073	-1.6855	-2.2850	-1.7263	-1.7584	-1.5223	-1.6233
COM	0.5307	-0.3036	-0.0376	-0.8065	-1.3418	-0.6265	-0.4895	-0.8224	-1.7263	-0.8763	-0.9743	-0.6988	-0.9108
VP	0.4669	-0.3795	-0.1500	-0.8543	-1.3739	-0.7166	-0.6053	-0.9218	-1.7584	-0.9743	-1.1425	-0.8772	-1.0622
VC	0.7222	-0.1170	0.1337	-0.6086	-1.1378	-0.4458	-0.3197	-0.6456	-1.5223	-0.6988	-0.8772	-0.9833	-1.1842
TMDE	0.5648	-0.2944	-0.0992	-0.7383	-1.2388	-0.6457	-0.5573	-0.8595	-1.6233	-0.9108	-1.0622	-1.1842	-1.4452

Positive values show pairs of means that are significantly different.

Connecting Letters Report

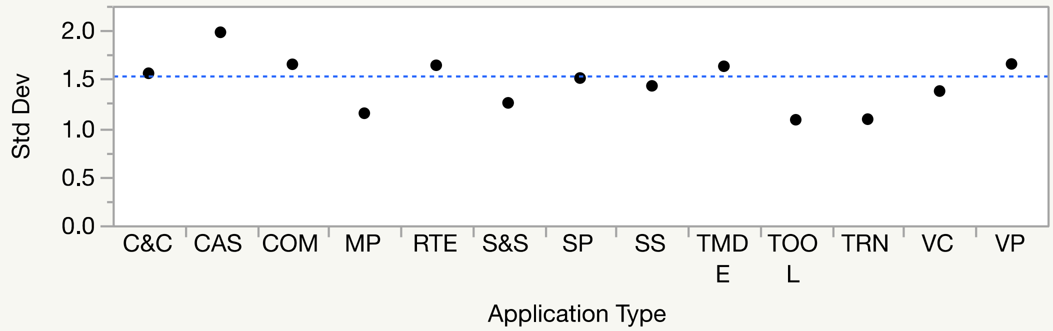
Level				Mean
MP	A			8.0833200
S&S	A	B	C	7.1168857
C&C	A	B		7.1016435
CAS	A	B	C	6.9708125
TOOL	A	B	C	6.7463000
SS		B	C	6.6672642
RTE		B	C	6.6298421
SP		B	C	6.4020986
TRN	A	B	C	6.3618000
COM		B	C	6.3575882
VP		B	C	6.3137500
VC			C	6.1250926
TMDE		B	C	6.0733200

Levels not connected by same letter are significantly different.

Ordered Differences Report

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
MP	TMDE	2.010000	0.4345900	0.56483	3.455169	0.0003*
MP	VC	1.958227	0.3716905	0.72222	3.194232	<.0001*
MP	VP	1.769570	0.3917342	0.46691	3.072227	0.0005*
MP	COM	1.725732	0.3593782	0.53067	2.920794	0.0001*
MP	TRN	1.721520	0.5749086	-0.19026	3.633298	0.1275
MP	SP	1.681221	0.3573314	0.49297	2.869477	0.0002*
MP	RTE	1.453478	0.3349404	0.33968	2.567275	0.0012*
MP	SS	1.416056	0.3727985	0.17637	2.655745	0.0101*
MP	TOOL	1.337020	0.5749086	-0.57476	3.248798	0.4990
MP	CAS	1.112508	0.4919225	-0.52331	2.748327	0.5461
S&S	TMDE	1.043566	0.4023522	-0.29440	2.381532	0.3162
C&C	TMDE	1.028323	0.3390626	-0.09918	2.155829	0.1153
S&S	VC	0.991793	0.3334255	-0.11697	2.100553	0.1342
MP	C&C	0.981677	0.3390626	-0.14583	2.109182	0.1635
C&C	VC	0.976551	0.2534734	0.13366	1.819441	0.0083*
MP	S&S	0.966434	0.4023522	-0.37153	2.304400	0.4439
CAS	TMDE	0.897492	0.4919225	-0.73833	2.533312	0.8369
CAS	VC	0.845720	0.4373478	-0.60862	2.300059	0.7749
S&S	VP	0.803136	0.3556325	-0.37947	1.985742	0.5485
C&C	VP	0.787893	0.2820471	-0.15002	1.725802	0.2077
S&S	COM	0.759297	0.3196428	-0.30363	1.822225	0.4628
S&S	TRN	0.755086	0.5509434	-1.07700	2.587171	0.9782
C&C	COM	0.744055	0.2350482	-0.03756	1.525675	0.0802
C&C	TRN	0.739843	0.5065716	-0.94469	2.424377	0.9640
S&S	SP	0.714787	0.3173398	-0.34048	1.770056	0.5529
C&C	SP	0.699545	0.2319066	-0.07163	1.470718	0.1204
TOOL	TMDE	0.672980	0.5749086	-1.23880	2.584758	0.9944
CAS	VP	0.657062	0.4545051	-0.85433	2.168456	0.9667
TOOL	VC	0.621207	0.5289661	-1.13780	2.380211	0.9942
CAS	COM	0.613224	0.4269332	-0.80648	2.032931	0.9684
CAS	TRN	0.609012	0.6193861	-1.45067	2.668695	0.9989
SS	TMDE	0.593944	0.3727985	-0.64575	1.833634	0.9318
CAS	SP	0.568714	0.4252117	-0.84527	1.982696	0.9822
RTE	TMDE	0.556522	0.3349404	-0.55728	1.670320	0.9091
SS	VC	0.542172	0.2970926	-0.44577	1.530112	0.8367
RTE	VC	0.504750	0.2479322	-0.31971	1.329213	0.7087
S&S	RTE	0.487044	0.2918970	-0.48362	1.457706	0.9065
C&C	RTE	0.471801	0.1956527	-0.17881	1.122417	0.4371
S&S	SS	0.449622	0.3346603	-0.66324	1.562488	0.9815
C&C	SS	0.434379	0.2550954	-0.41390	1.282663	0.8932
TOOL	VP	0.432550	0.5432375	-1.37391	2.239011	0.9999
TOOL	COM	0.388712	0.5203884	-1.34177	2.119191	0.9999
TOOL	TRN	0.384500	0.6871472	-1.90051	2.669512	1.0000
S&S	TOOL	0.370586	0.5509434	-1.46150	2.202671	1.0000
C&C	TOOL	0.355343	0.5065716	-1.32919	2.039877	1.0000
SS	VP	0.353514	0.3218165	-0.71664	1.423670	0.9969
TOOL	SP	0.344201	0.5189770	-1.38158	2.069987	1.0000
CAS	RTE	0.340970	0.4065763	-1.01104	1.692983	0.9998
SP	TMDE	0.328779	0.3573314	-0.85948	1.517034	0.9994
RTE	VP	0.316092	0.2770780	-0.60529	1.237476	0.9956
SS	COM	0.309676	0.2815367	-0.62654	1.245887	0.9969
SS	TRN	0.305464	0.5297453	-1.45613	2.067058	1.0000
CAS	SS	0.303548	0.4382898	-1.15392	1.761020	1.0000
TRN	TMDE	0.288480	0.5749086	-1.62330	2.200258	1.0000
COM	TMDE	0.284268	0.3593782	-0.91079	1.479331	0.9999
SP	VC	0.277006	0.2774366	-0.64557	1.199583	0.9988
RTE	COM	0.272254	0.2290617	-0.48946	1.033967	0.9936
RTE	TRN	0.268042	0.5038218	-1.40735	1.943431	1.0000
SS	SP	0.265166	0.2789193	-0.66234	1.192673	0.9992
VP	TMDE	0.240430	0.3917342	-1.06223	1.543087	1.0000
TRN	VC	0.236707	0.5289661	-1.52230	1.995711	1.0000
COM	VC	0.232496	0.2800679	-0.69883	1.163822	0.9998
RTE	SP	0.227744	0.2258369	-0.52325	0.978733	0.9986
CAS	TOOL	0.224513	0.6193861	-1.83517	2.284195	1.0000
VP	VC	0.188657	0.3205323	-0.87723	1.254543	1.0000
S&S	CAS	0.146073	0.4636880	-1.39586	1.688003	1.0000
C&C	CAS	0.130831	0.4099789	-1.23250	1.494159	1.0000
TOOL	RTE	0.116458	0.5038218	-1.55893	1.791847	1.0000
SP	VP	0.088349	0.3037646	-0.92178	1.098475	1.0000
TOOL	SS	0.079036	0.5297453	-1.68256	1.840630	1.0000
VC	TMDE	0.051773	0.3716905	-1.18423	1.287777	1.0000
TRN	VP	0.048050	0.5432375	-1.75841	1.854511	1.0000
SP	COM	0.044510	0.2607105	-0.82245	0.911467	1.0000
COM	VP	0.043838	0.3061697	-0.97429	1.061963	1.0000
SP	TRN	0.040299	0.5189770	-1.68549	1.766084	1.0000
SS	RTE	0.037422	0.2495902	-0.79256	0.867399	1.0000
S&S	C&C	0.015242	0.2966180	-0.97112	1.001604	1.0000
TRN	COM	0.004212	0.5203884	-1.72627	1.734691	1.0000

Tests that the Variances are Equal



Level	Count	Std Dev	MeanAbsDif to Mean	MeanAbsDif to Median
C&C	115	1.558014	1.233633	1.230000
CAS	16	1.978437	1.517539	1.461813
COM	68	1.650477	1.289358	1.285235
MP	25	1.148257	0.901373	0.898920
RTE	133	1.640087	1.231776	1.230226
S&S	35	1.254468	0.979438	0.974800
SP	71	1.508979	1.273897	1.272775
SS	53	1.428968	1.144448	1.142943
TMDE	25	1.630197	1.254198	1.248680
TOOL	10	1.081237	0.874040	0.853300
TRN	10	1.087168	0.846440	0.839200
VC	54	1.375831	1.156123	1.137574
VP	40	1.653807	1.280600	1.280600

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.9144	12	642	0.5320
Brown-Forsythe	0.8676	12	642	0.5802
Levene	0.9242	12	642	0.5219
Bartlett	1.3119	12	.	0.2033

Welch's Test

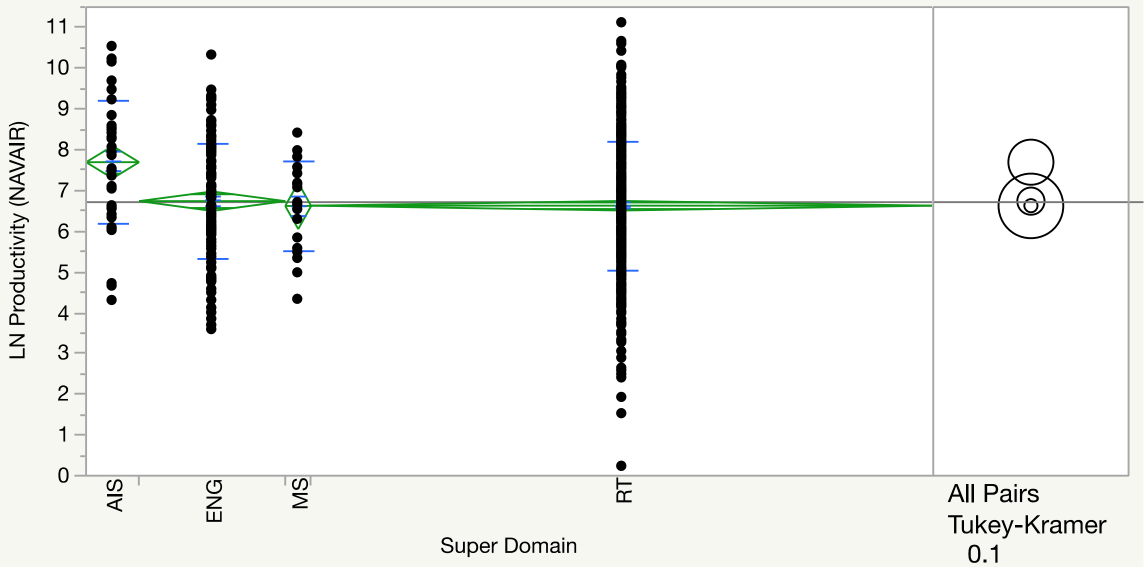
Welch Anova testing Means Equal, allowing Std Devs Not Equal

F Ratio	DFNum	DFDen	Prob > F
5.5234	12	120.47	<.0001*

Excluded Rows 2

Figure B-3: One-Way ANOVA of LN Productivity (NAVAIR) by Super Domain

Oneway Analysis of LN Productivity (NAVAIR) By Super Domain



Oneway Anova

Summary of Fit

Rsquare	0.027136
Adj Rsquare	0.022653
Root Mean Square Error	1.545116
Mean of Response	6.711969
Observations (or Sum Wgts)	655

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Super Domain	3	43.3505	14.4502	6.0527	0.0005*
Error	651	1554.1864	2.3874		
C. Total	654	1597.5369			

Means for Oneway Anova

Level	Number	Mean	Std Error
AIS	41	7.69422	0.24131
ENG	113	6.73755	0.14535
MS	20	6.62050	0.34550
RT	481	6.62604	0.07045

Std Error uses a pooled estimate of error variance

Means and Std Deviations

Level	Number	Mean	Std Dev
AIS	41	7.6942195	1.5182693
ENG	113	6.7375487	1.424752
MS	20	6.6205	1.098679
RT	481	6.6260374	1.5888249

Means Comparisons

Comparisons for all pairs using Tukey-Kramer HSD

Confidence Quantile

q*	Alpha
2.29593	0.1

HSD Threshold Matrix

Abs(Dif)-HSD

	AIS	ENG	RT	MS
AIS	-0.7835	0.3099	0.4910	0.1062
ENG	0.3099	-0.4720	-0.2593	-0.7435
RT	0.4910	-0.2593	-0.2288	-0.8040
MS	0.1062	-0.7435	-0.8040	-1.1218

Positive values show pairs of means that are significantly different.

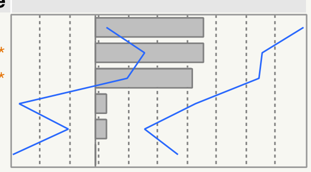
Connecting Letters Report

Level			Mean
AIS	A		7.6942195
ENG		B	6.7375487
RT		B	6.6260374
MS		B	6.6205000

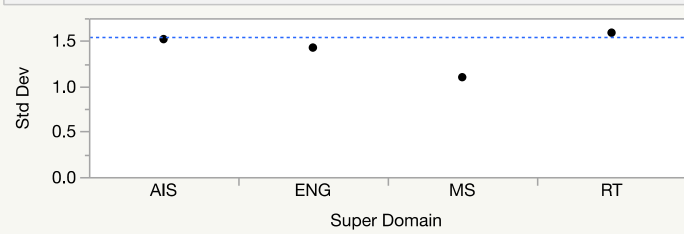
Levels not connected by same letter are significantly different.

Ordered Differences Report

Level	- Level	Difference	Std Err Dif	p-Value
AIS	MS	1.073720	0.4214238	0.0538
AIS	RT	1.068182	0.2513806	0.0001*
AIS	ENG	0.956671	0.2817022	0.0040*
ENG	MS	0.117049	0.3748285	0.9894
ENG	RT	0.111511	0.1615260	0.9008
RT	MS	0.005537	0.3526082	1.0000



Tests that the Variances are Equal



Level	Count	Std Dev	MeanAbsDif to Mean	MeanAbsDif to Median
AIS	41	1.518269	1.195200	1.190902
ENG	113	1.424752	1.165789	1.160611
MS	20	1.098679	0.877950	0.877100
RT	481	1.588825	1.248637	1.248356

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.2904	3	651	0.2767
Brown-Forsythe	1.1816	3	651	0.3159
Levene	1.1633	3	651	0.3229
Bartlett	1.8972	3	.	0.1276

Welch's Test

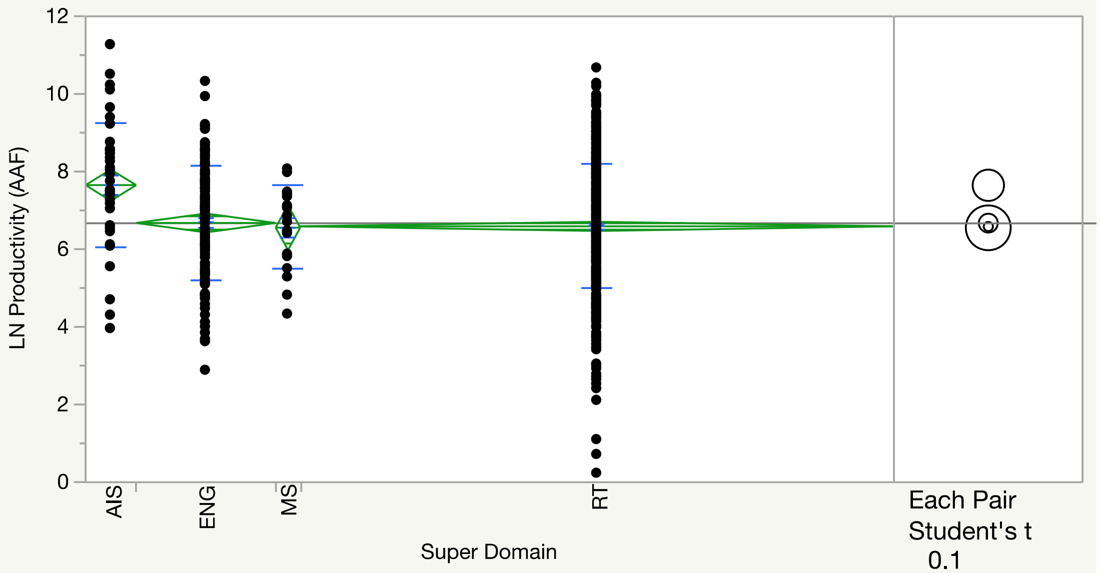
Welch Anova testing Means Equal, allowing Std Devs Not Equal

F Ratio	DFNum	DFDen	Prob > F
6.1247	3	67.131	0.0010*

Excluded Rows 2

Figure B-4: One-Way ANOVA of LN Productivity (AAF) by Super Domain

Oneway Analysis of LN Productivity (AAF) By Super Domain



Oneway Anova

Summary of Fit

Rsquare	0.026151
Adj Rsquare	0.021663
Root Mean Square Error	1.565561
Mean of Response	6.668113
Observations (or Sum Wgts)	655

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Super Domain	3	42.8464	14.2821	5.8271	0.0006*
Error	651	1595.5884	2.4510		
C. Total	654	1638.4348			

Means for Oneway Anova

Level	Number	Mean	Std Error
AIS	41	7.64917	0.24450
ENG	113	6.67512	0.14728
MS	20	6.55405	0.35007
RT	481	6.58758	0.07138

Std Error uses a pooled estimate of error variance

Means and Std Deviations

Level	Number	Mean	Std Dev
AIS	41	7.6491707	1.6002492
ENG	113	6.6751239	1.4618836
MS	20	6.55405	1.0735644
RT	481	6.5875842	1.6020187

Means Comparisons

Comparisons for each pair using Student's t

Confidence Quantile

t	Alpha
1.64720	0.1

LSD Threshold Matrix

Abs(Dif)-LSD

	AIS	ENG	RT	MS
AIS	-0.56956	0.50389	0.64203	0.39177
ENG	0.50389	-0.34308	-0.18205	-0.50451
RT	0.64203	-0.18205	-0.16629	-0.55497
MS	0.39177	-0.50451	-0.55497	-0.81548

Positive values show pairs of means that are significantly different.

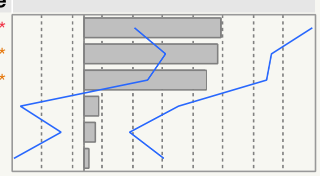
Connecting Letters Report

Level			Mean
AIS	A		7.6491707
ENG		B	6.6751239
RT		B	6.5875842
MS		B	6.5540500

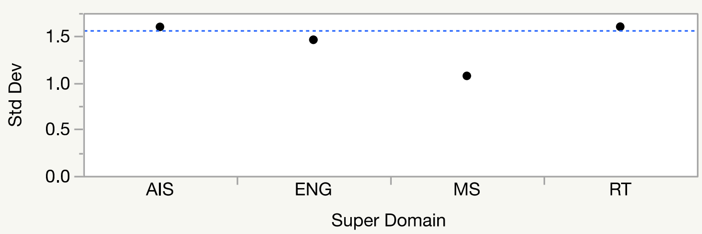
Levels not connected by same letter are significantly different.

Ordered Differences Report

Level	- Level	Difference	Std Err Dif	p-Value
AIS	MS	1.095121	0.4270001	0.0106*
AIS	RT	1.061587	0.2547069	<.0001*
AIS	ENG	0.974047	0.2854297	0.0007*
ENG	MS	0.121074	0.3797883	0.7500
ENG	RT	0.087540	0.1636633	0.5929
RT	MS	0.033534	0.3572739	0.9252



Tests that the Variances are Equal



Level	Count	Std Dev	MeanAbsDif to Mean	MeanAbsDif to Median
AIS	41	1.600249	1.220029	1.216610
ENG	113	1.461884	1.188543	1.183442
MS	20	1.073564	0.875845	0.862150
RT	481	1.602019	1.257712	1.257707

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.1678	3	651	0.3212
Brown-Forsythe	1.1929	3	651	0.3116
Levene	1.1142	3	651	0.3426
Bartlett	1.9251	3	.	0.1231

Welch's Test

Welch Anova testing Means Equal, allowing Std Devs Not Equal

F Ratio	DFNum	DFDen	Prob > F
5.5075	3	67.252	0.0019*

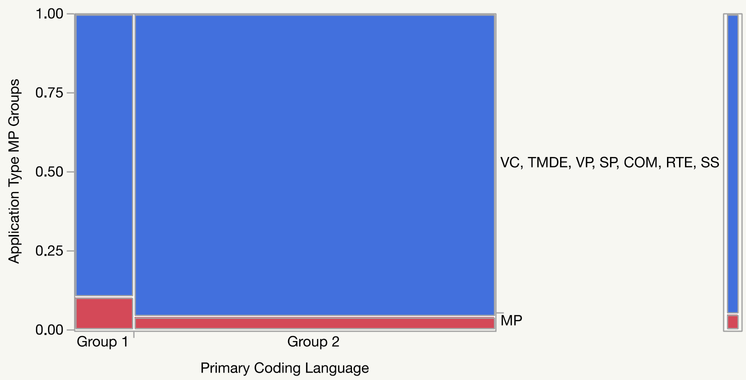
Excluded Rows 2

Appendix C

Figure C-1: Contingency Analysis of Application Type MP Groups by Primary Coding Language Conversion Factor

Contingency Analysis of Application Type MP Groups By Primary Coding Language

Mosaic Plot



Contingency Table

Application Type MP Groups (columns) by Primary Coding Language (rows)

Count / Total % / Col % / Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
Group 1	7	59	66
	1.49	12.58	14.07
	28.00	13.29	
	10.61	89.39	
Group 2	18	385	403
	3.84	82.09	85.93
	72.00	86.71	
	4.47	95.53	
Total	25	444	469
	5.33	94.67	

Tests

N	DF	-LogLike	RSquare (U)
469	1	1.7475052	0.0179

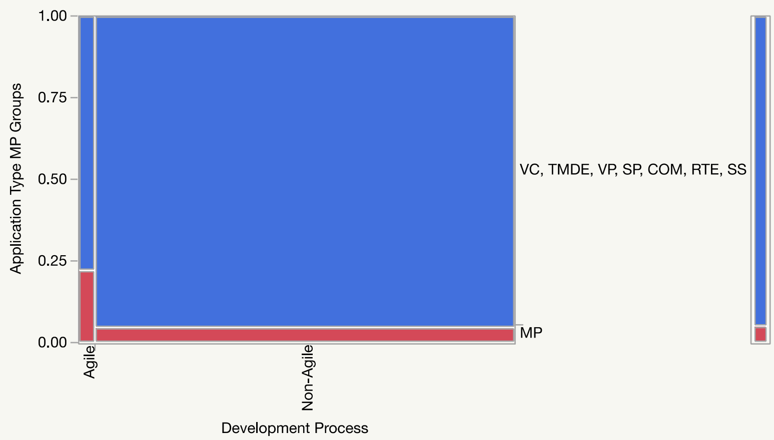
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	3.495	0.0616
Pearson	4.236	0.0396*

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9849	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Primary Coding Language=Group 1 than Group 2
Right	0.0473*	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Primary Coding Language=Group 2 than Group 1
2-Tail	0.0675	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is different across Primary Coding Language

Figure C-2: Contingency Analysis of Application Type MP Groups by Development Process

Contingency Analysis of Application Type MP Groups By Development Process

Mosaic Plot



Contingency Table

Application Type MP Groups

Development Process	Count / Total % / Col % / Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
Agile	Count	4	14	18
	Total %	0.85	2.99	3.84
	Col %	16.00	3.15	
	Row %	22.22	77.78	
Non-Agile	Count	21	430	451
	Total %	4.48	91.68	96.16
	Col %	84.00	96.85	
	Row %	4.66	95.34	
Total	Count	25	444	469
	Total %	5.33	94.67	

Tests

N	DF	-LogLike	RSquare (U)
469	1	3.1708263	0.0325

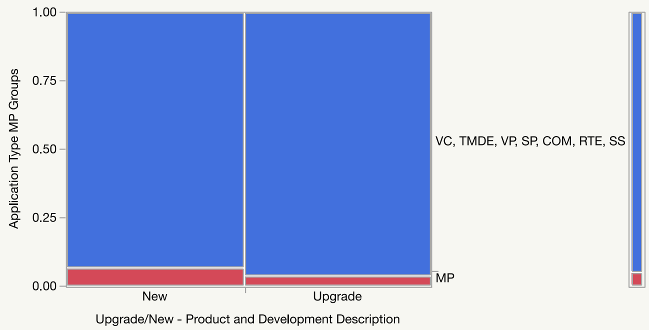
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	6.342	0.0118*
Pearson	10.584	0.0011*

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9985	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Development Process=Agile than Non-Agile
Right	0.0116*	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Development Process=Non-Agile than Agile
2-Tail	0.0116*	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is different across Development Process

Figure C-3: Contingency Analysis of Application Type MP Groups by Upgrade/New Product Development Description

Contingency Analysis of Application Type MP Groups By Upgrade/New - Product and Development Description

Mosaic Plot



Contingency Table

Application Type MP Groups

Upgrade/New - Product and Development Description	Count / Total % / Col % / Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
New	Count	16	214	230
	Total %	3.41	45.63	49.04
	Col %	64.00	48.20	
	Row %	6.96	93.04	
Upgrade	Count	9	230	239
	Total %	1.92	49.04	50.96
	Col %	36.00	51.80	
	Row %	3.77	96.23	
Total	Count	25	444	469
	Total %	5.33	94.67	

Tests

N	DF	-LogLike	RSquare (U)
469	1	1.1952161	0.0122

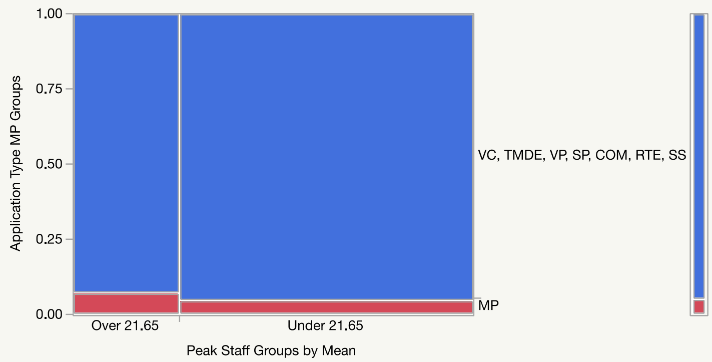
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	2.390	0.1221
Pearson	2.365	0.1241

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9601	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Upgrade/New - Product and Development Description=New than Upgrade
Right	0.0911	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Upgrade/New - Product and Development Description=Upgrade than New
2-Tail	0.1511	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is different across Upgrade/New - Product and Development Description

Figure C-4: Contingency Analysis of Application Type MP Groups by Peak Staff Mean

Contingency Analysis of Application Type MP Groups By Peak Staff Groups by Mean

Mosaic Plot



Contingency Table

Application Type MP Groups

Peak Staff Groups by Mean (Count / Total % / Col % / Row %)	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
Over 21.65	9	116	125
	1.92	24.73	26.65
	36.00	26.13	
	7.20	92.80	
Under 21.65	16	328	344
	3.41	69.94	73.35
	64.00	73.87	
	4.65	95.35	
Total	25	444	469
	5.33	94.67	

Tests

N	DF	-LogLike	RSquare (U)
469	1	0.55611623	0.0057

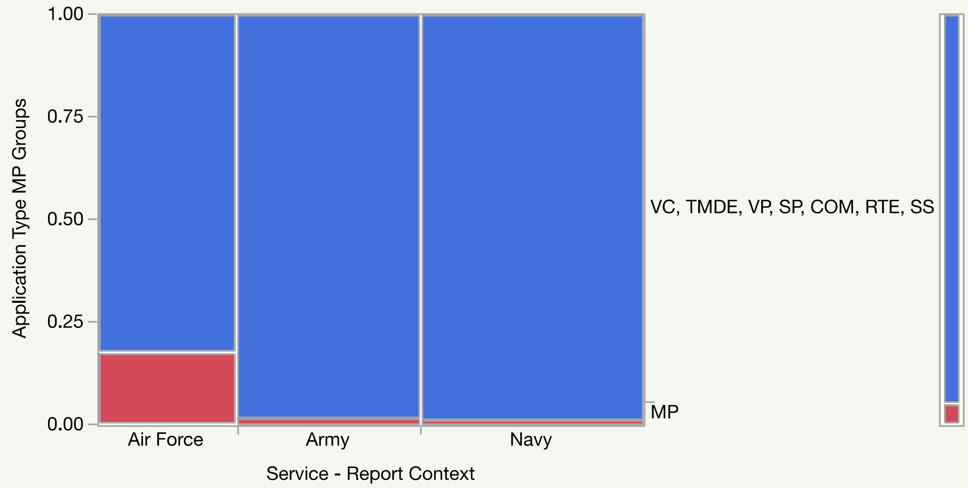
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	1.112	0.2916
Pearson	1.180	0.2773

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9034	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Peak Staff Groups by Mean=Over 21.65 than Under 21.65
Right	0.1941	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is greater for Peak Staff Groups by Mean=Under 21.65 than Over 21.65
2-Tail	0.3510	Prob(Application Type MP Groups=VC, TMDE, VP, SP, COM, RTE, SS) is different across Peak Staff Groups by Mean

Figure C-5: Contingency Analysis of Application Type MP Groups by Service

Contingency Analysis of Application Type MP Groups By Service - Report Context

Mosaic Plot



Contingency Table

Application Type MP Groups

Service - Report Context	Count / Total % / Col % / Row %	MP	VC, TMDE, VP, SP, COM, RTE, SS	Total
Air Force	Count	21	98	119
	Total %	4.52	21.08	25.59
	Col %	84.00	22.27	
	Row %	17.65	82.35	
Army	Count	2	154	156
	Total %	0.43	33.12	33.55
	Col %	8.00	35.00	
	Row %	1.28	98.72	
Navy	Count	2	188	190
	Total %	0.43	40.43	40.86
	Col %	8.00	42.73	
	Row %	1.05	98.95	
Total	Count	25	440	465
	Total %	5.38	94.62	

Tests

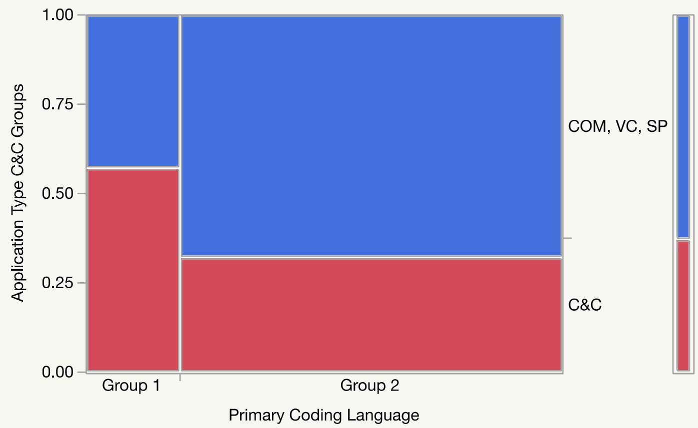
N	DF	-LogLike	RSquare (U)
465	2	20.142974	0.2068

Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	40.286	<.0001*
Pearson	47.343	<.0001*

Figure C-6: Contingency Analysis of Application Type C&C Groups by Primary Coding Language Conversion Factor

Contingency Analysis of Application Type C&C Groups By Primary Coding Language

Mosaic Plot



Contingency Table

Application Type C&C Groups

Primary Coding Language	Count / Total % / Col % / Row %	C&C	COM, VC, SP	Total
Group 1	Count	35	26	61
	Total %	11.36	8.44	19.81
	Col %	30.43	13.47	
	Row %	57.38	42.62	
Group 2	Count	80	167	247
	Total %	25.97	54.22	80.19
	Col %	69.57	86.53	
	Row %	32.39	67.61	
Total	Count	115	193	308
	Total %	37.34	62.66	

Tests

N	DF	-LogLike	RSquare (U)
308	1	6.3368967	0.0311

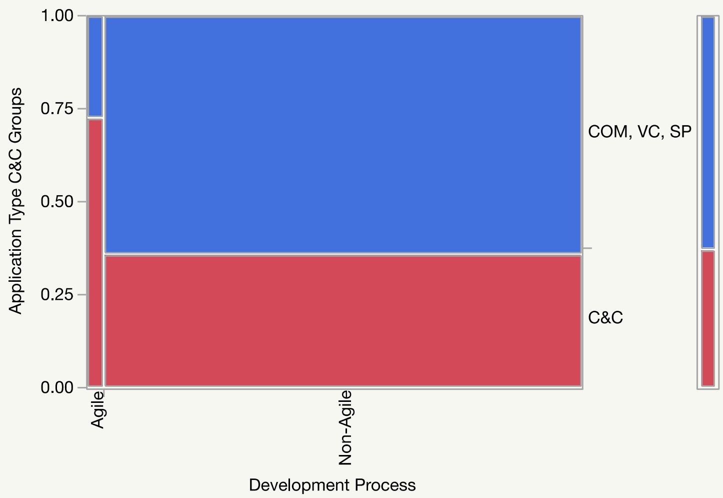
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	12.674	0.0004*
Pearson	13.056	0.0003*

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9999	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Primary Coding Language=Group 1 than Group 2
Right	0.0003*	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Primary Coding Language=Group 2 than Group 1
2-Tail	0.0006*	Prob(Application Type C&C Groups=COM, VC, SP) is different across Primary Coding Language

Figure C-7: Contingency Analysis of Application Type C&C Groups by Development Process

Contingency Analysis of Application Type C&C Groups By Development Process

Mosaic Plot



Contingency Table

Application Type C&C Groups

Development Process	Count / Total % / Col % / Row %	C&C	COM, VC, SP	Total
Agile	Count	8	3	11
	Total %	2.60	0.97	3.57
	Col %	6.96	1.55	
	Row %	72.73	27.27	
Non-Agile	Count	107	190	297
	Total %	34.74	61.69	96.43
	Col %	93.04	98.45	
	Row %	36.03	63.97	
Total	Count	115	193	308
	Total %	37.34	62.66	

Tests

N	DF	-LogLike	RSquare (U)
308	1	2.9476669	0.0145

Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	5.895	0.0152*
Pearson	6.106	0.0135*

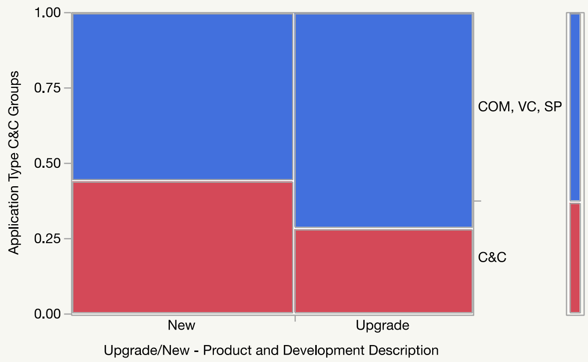
Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9971	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Development Process=Agile than Non-Agile
Right	0.0170*	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Development Process=Non-Agile than Agile
2-Tail	0.0223*	Prob(Application Type C&C Groups=COM, VC, SP) is different across Development Process

Figure C-8: Contingency Analysis of Application Type C&C Groups by Upgrade/New

Product Development Description

Contingency Analysis of Application Type C&C Groups By Upgrade/New - Product and Development Description

Mosaic Plot



Contingency Table

Application Type C&C Groups

Upgrade/New - Product and Development Description	Count Total % Col % Row %	C&C	COM, VC, SP	Total
	New	76 24.68 66.09 44.44	95 30.84 49.22 55.56	171 55.52
	Upgrade	39 12.66 33.91 28.47	98 31.82 50.78 71.53	137 44.48
	Total	115 37.34	193 62.66	308

Tests

N	DF	-LogLike	RSquare (U)
308	1	4.2022326	0.0206

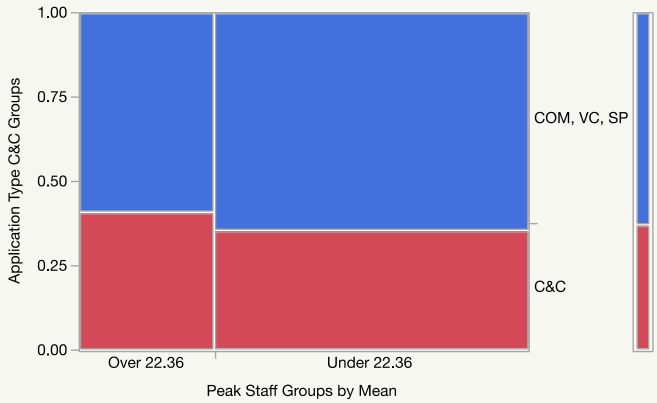
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	8.404	0.0037*
Pearson	8.299	0.0040*

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9987	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Upgrade/New - Product and Development Description=New than Upgrade
Right	0.0027*	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Upgrade/New - Product and Development Description=Upgrade than New
2-Tail	0.0045*	Prob(Application Type C&C Groups=COM, VC, SP) is different across Upgrade/New - Product and Development Description

Figure C-9: Contingency Analysis of Application Type C&C Groups by Peak Staff Mean

Contingency Analysis of Application Type C&C Groups By Peak Staff Groups by Mean

Mosaic Plot



Contingency Table

Application Type C&C Groups

Peak Staff Groups by Mean	Count / Total % / Col % / Row %	C&C	COM, VC, SP	Total
Over 22.36	Count	38	55	93
	Total %	12.34	17.86	30.19
	Col %	33.04	28.50	
	Row %	40.86	59.14	
Under 22.36	Count	77	138	215
	Total %	25.00	44.81	69.81
	Col %	66.96	71.50	
	Row %	35.81	64.19	
Total	Count	115	193	308
	Total %	37.34	62.66	

Tests

N	DF	-LogLike	RSquare (U)
308	1	0.35102832	0.0017

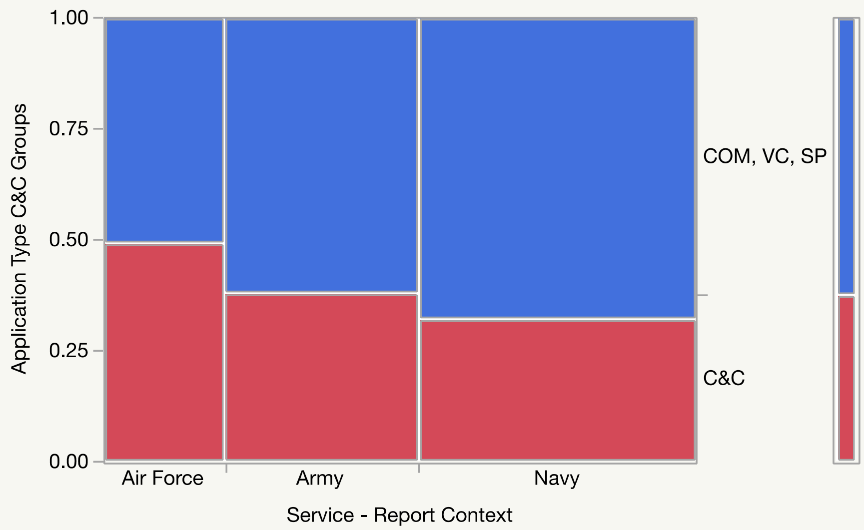
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	0.702	0.4021
Pearson	0.707	0.4006

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.8338	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Peak Staff Groups by Mean=Over 22.36 than Under 22.36
Right	0.2375	Prob(Application Type C&C Groups=COM, VC, SP) is greater for Peak Staff Groups by Mean=Under 22.36 than Over 22.36
2-Tail	0.4420	Prob(Application Type C&C Groups=COM, VC, SP) is different across Peak Staff Groups by Mean

Figure C-10: Contingency Analysis of Application Type C&C Groups by Service

Contingency Analysis of Application Type C&C Groups By Service - Report Context

Mosaic Plot



Contingency Table

Application Type C&C Groups

Service - Report Context	Count Total % Col % Row %	C&C	COM, VC, SP	Total
Air Force	Count	31	32	63
	Total %	10.13	10.46	20.59
	Col %	26.96	16.75	
	Row %	49.21	50.79	
Army	Count	38	62	100
	Total %	12.42	20.26	32.68
	Col %	33.04	32.46	
	Row %	38.00	62.00	
Navy	Count	46	97	143
	Total %	15.03	31.70	46.73
	Col %	40.00	50.79	
	Row %	32.17	67.83	
Total	Count	115	191	306
	Total %	37.58	62.42	

Tests

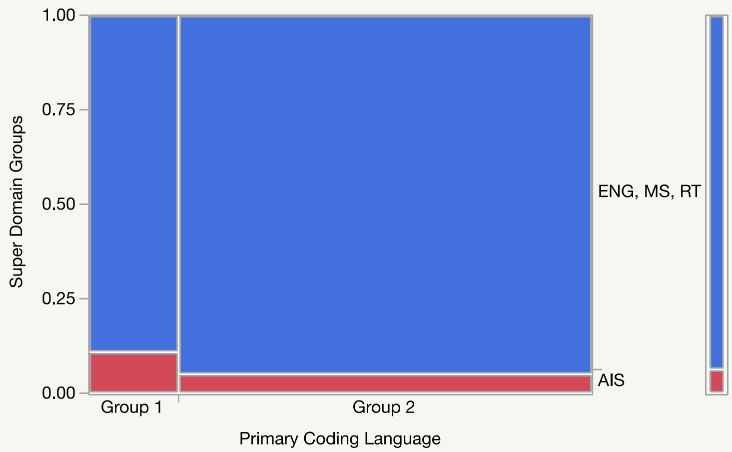
N	DF	-LogLike	RSquare (U)
306	2	2.6765605	0.0132

Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	5.353	0.0688
Pearson	5.423	0.0664

Figure C-11: Contingency Analysis of Super Domain AIS Groups by Primary Coding Language Conversion Factor

Contingency Analysis of Super Domain Groups By Primary Coding Language

Mosaic Plot



Contingency Table

Super Domain Groups (columns) by Primary Coding Language (rows)

Count / Total % / Col % / Row %	AIS	ENG, MS, RT	Total
Group 1	13	104	117
	1.98	15.88	17.86
	31.71	16.94	
	11.11	88.89	
Group 2	28	510	538
	4.27	77.86	82.14
	68.29	83.06	
	5.20	94.80	
Total	41	614	655
	6.26	93.74	

Tests

N	DF	-LogLike	RSquare (U)
655	1	2.4726745	0.0161

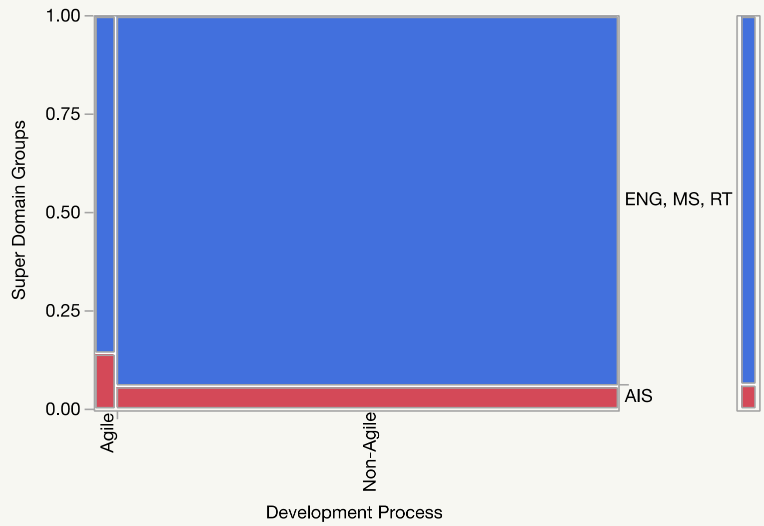
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	4.945	0.0262*
Pearson	5.714	0.0168*

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9927	Prob(Super Domain Groups=ENG, MS, RT) is greater for Primary Coding Language=Group 1 than Group 2
Right	0.0192*	Prob(Super Domain Groups=ENG, MS, RT) is greater for Primary Coding Language=Group 2 than Group 1
2-Tail	0.0322*	Prob(Super Domain Groups=ENG, MS, RT) is different across Primary Coding Language

Figure C-12: Contingency Analysis of Super Domain AIS Groups by Development Process

Contingency Analysis of Super Domain Groups By Development Process

Mosaic Plot



Contingency Table

Super Domain Groups

Development Process: Count / Total % / Col % / Row %	AIS	ENG, MS, RT	Total
Agile	4	24	28
	0.61	3.66	4.27
	9.76	3.91	
	14.29	85.71	
Non-Agile	37	590	627
	5.65	90.08	95.73
	90.24	96.09	
	5.90	94.10	
Total	41	614	655
	6.26	93.74	

Tests

N	DF	-LogLike	RSquare (U)
655	1	1.2222604	0.0080

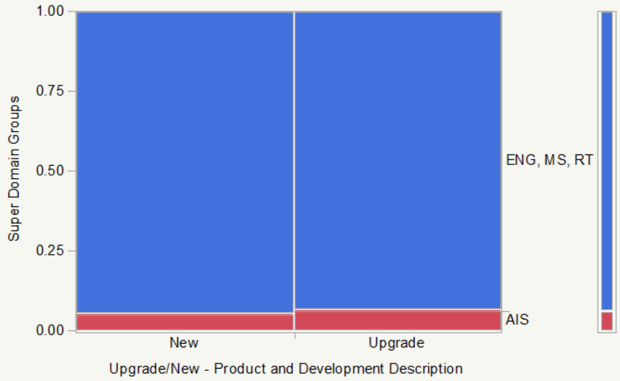
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	2.445	0.1179
Pearson	3.211	0.0731

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.9748	Prob(Super Domain Groups=ENG, MS, RT) is greater for Development Process=Agile than Non-Agile
Right	0.0904	Prob(Super Domain Groups=ENG, MS, RT) is greater for Development Process=Non-Agile than Agile
2-Tail	0.0904	Prob(Super Domain Groups=ENG, MS, RT) is different across Development Process

Figure C-13: Contingency Analysis of Super Domain AIS Groups by Upgrade/New Product Development Description

Contingency Analysis of Super Domain Groups By Upgrade/New - Product and Development Description

Mosaic Plot



Contingency Table

Super Domain Groups (columns); Upgrade/New - Product and Development Description (rows)

Count Total % Col % Row %	AIS	ENG, MS, RT	Total
New	19 2.90 46.34 5.64	318 48.55 51.79 94.36	337 51.45
Upgrade	22 3.36 53.66 6.92	296 45.19 48.21 93.08	318 48.55
Total	41 6.26	614 93.74	655

Tests

N	DF	-LogLike	RSquare (U)
655	1	0.22846425	0.0015

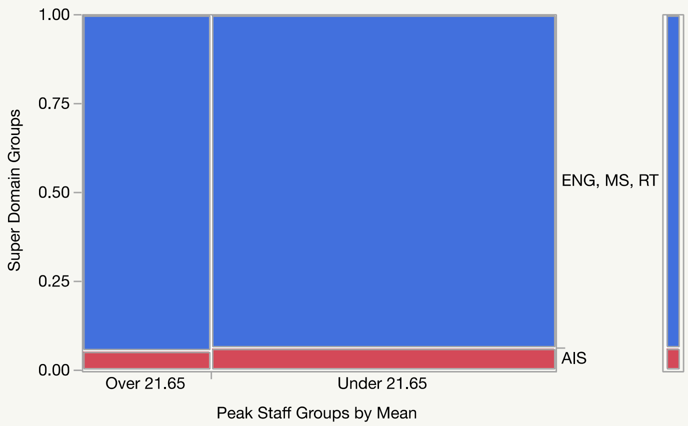
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	0.457	0.4991
Pearson	0.457	0.4990

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.3033	Prob(Super Domain Groups=ENG, MS, RT) is greater for Upgrade/New - Product and Development Description=New than Upgrade
Right	0.7988	Prob(Super Domain Groups=ENG, MS, RT) is greater for Upgrade/New - Product and Development Description=Upgrade than New
2-Tail	0.5224	Prob(Super Domain Groups=ENG, MS, RT) is different across Upgrade/New - Product and Development Description

Figure C-14: Contingency Analysis of Super Domain AIS Groups by Peak Staff Mean

Contingency Analysis of Super Domain Groups By Peak Staff Groups by Mean

Mosaic Plot



Contingency Table

Peak Staff Groups by Mean	Count / Total % / Col % / Row %	AIS	ENG, MS, RT	Total
Over 21.65	Count	10	169	179
	Total %	1.53	25.80	27.33
	Col %	24.39	27.52	
	Row %	5.59	94.41	
Under 21.65	Count	31	445	476
	Total %	4.73	67.94	72.67
	Col %	75.61	72.48	
	Row %	6.51	93.49	
Total	Count	41	614	655
	Total %	6.26	93.74	

(Super Domain Groups)

Tests

N	DF	-LogLike	RSquare (U)
655	1	0.09717352	0.0006

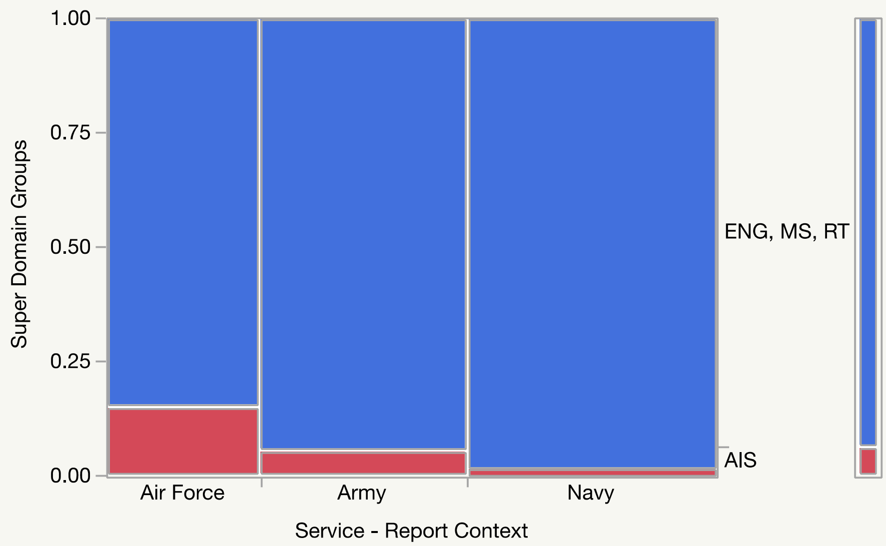
Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	0.194	0.6593
Pearson	0.190	0.6628

Fisher's Exact Test	Prob	Alternative Hypothesis
Left	0.4084	Prob(Super Domain Groups=ENG, MS, RT) is greater for Peak Staff Groups by Mean=Over 21.65 than Under 21.65
Right	0.7261	Prob(Super Domain Groups=ENG, MS, RT) is greater for Peak Staff Groups by Mean=Under 21.65 than Over 21.65
2-Tail	0.7215	Prob(Super Domain Groups=ENG, MS, RT) is different across Peak Staff Groups by Mean

Figure C-15: Contingency Analysis of Super Domain AIS Groups by Service

Contingency Analysis of Super Domain Groups By Service - Report Context

Mosaic Plot



Contingency Table

Super Domain Groups (rows: Service - Report Context)

Count / Total % / Col % / Row %	AIS	ENG, MS, RT	Total
Air Force	25	139	164
	3.85	21.38	25.23
	60.98	22.82	
	15.24	84.76	
Army	12	208	220
	1.85	32.00	33.85
	29.27	34.15	
	5.45	94.55	
Navy	4	262	266
	0.62	40.31	40.92
	9.76	43.02	
	1.50	98.50	
Total	41	609	650
	6.31	93.69	

Tests

N	DF	-LogLike	RSquare (U)
650	2	15.634094	0.1022

Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	31.268	<.0001*
Pearson	32.819	<.0001*

References

Banker, R., & Kemerer, C. (1989, October). Scale Economies in New Software Development. IEEE Transactions on Software Engineering.

Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B., Horowitz, E., . . . Steece, B. (2000). COCOMO II. USC Center for Software Engineering.

Chaillan, N. (2019). DoD Enterprise DevSecOps Initiative (Software Factory).

Clark, B., & Madachy, R. (2015). *Software Cost Estimation Metrics Manual for Defense Systems.* Haymarket, VA, VA: Software Metrics Inc.

Clark, B., McCurley, J., & Zubrow, D. (2015, December). DoD Software Factbook Version 1.1. Software Engineering institute.

Clark, B., Miller, C., McCurley, J., Zubrow, D., Brown, R., & Zuccher, M. (2017, July). Department of Defense Software Factbook. *Software Engineering Institute*. Carnegie Mellon University.

Defense Acquisition University. (2019, May 9). DEVSECOPS ACADEMY TRANSFORMING DOD’S WORKFORCE: WINNING THE FIGHT WITH DEVSECOPS AND DIGITAL INNOVATION. Defense Acquisition University.

Defense Science Board. (2018). *Design and Acquisition of Software For Defense Systems.* Washington, D.C.: Department of Defense.

Department of Defense, Chief Information Officer. (2019, August 12). DoD Enterprise DevSecOps Reference Design.

Durate, C. (2019). The Quest for Productivity in Software Engineering: A Practitioners Systematic Literature Review. IEEE/ACM International Conference on Software and System Processes.

Durate, C. H. (2019). The Quest for Productivity in Software Engineering: A Practitioners Systematic Literature Review. *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)* (pp. 145-154). IEEE Xplore Digital Library.

Harmon, B., & Om, N. (2003, August). Schedule Assessment Methods for Ballistic Missile Defense Ground-Based Software Development. Institute for Defense Analysis.

Jensen, R. (2015). *Improving Software Development Productivity.* Prentence Hall.

Lanham, N., Russo, M., Strickland, D., Cipressi, R., Palmer, S., & Rudloff, C. (2018, February). DEPARTMENT OF DEFENSE SOFTWARE RESOURCE DATA REPORT (SRDR) VERIFICATION AND VALIDATION (V&V) GUIDE VERSION 4.0.

Lavazza, L., Morasca, S., & Tosi, D. (2018). An Empirical Study on the Factors Affecting Software Development Productivity. *e-Informatica Software Engineering Journal*.

Madachy, R., Bohem, B., Clark, B., Tan, T., & Rosa, W. (2011). US DoD Application Domain Empirical Software Cost Analysis. International Symposium on Empirical Software Engineering and Measurement.

Maxwell, K., Van Wassenhove, L., & Dutta, S. (1996). Software Development Productivity of European Space, Military, and Industrial Applications. *IEEE Transactions on Software Engineering*.

Morasca, S., & Russo, G. (2001). An Empirical Study of Software Productivity. *25th Annual International Computer Software and Applications Conference* (pp. 317-322). IEEE Xplore Digital Library.

Morasca, S., & Russo, G. (2001). An Empirical Study of Software Productivity. *IEEE*.

Oliveira, E., Conte, T., Cristo, M., & Valentim, N. (2013). Influence Factors in Software Productivity: A Tertiary Literature Review. Brazil.

OSD CAPE. (2019, February). THE SOFTWARE RESOURCES DATA REPORT (SRDR) IMPLEMENTATION GUIDANCE.

Putnam, L. (1992). Measures of Excellence: Reliable Software on Time, within Budget.

Rosa, W., Madachy, R., Bohem, B., Clark, B., Jones, C., McGarry, J., & Dean, J. (2014). Improved Method for Predicting Software Effort and Schedule.

Sadowski, C., & Zimmerman, T. (2019). *Rethinking Productivity in Software Engineering.* Apress Media LLC.

Stephenson, W. E. (1976). AN ANALYSIS OF THE RESOURCES USED IN THE SAFEGUARD SYSTEM SOFTWARE DEVELOPMENT. Bell Laboratories.

Tan, T. (2012). DOMAIN - BASED EFFORT DISTRIBUTION MODEL FOR SOFTWARE COST ESTIMATION. UNIVERSITY OF SOUTHER N CALIFORNIA.

Trendowicz, A., & Münch, J. (2009). Factors Influencing Software Development Productivity . Fraunhofer Institute for experimental Software Engineering.

Wagner, S., & Ruhe, M. (2018). A Systematic Review of Productivity Factors in Software Development. Germany.

REPORT DOCUMENTATION PAGE				*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 23-03-2021	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* Sep 2019 – March 2021
TITLE AND SUBTITLE **4.** An Analysis of Application Type, Super Domain, and Productivity in Software Intensive Defense Acquisitions	**5a. CONTRACT NUMBER**	
	5b. GRANT NUMBER	
	5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Amato, Evan P., 1st Lt, USAF	**5d. PROJECT NUMBER**	
	5e. TASK NUMBER	
	5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENV) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865		**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENV-MS-21-M-202
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Norwegian Defence Research Establishment (FFI) FFI, Postboks 25, 2027 Kjeller, Norway ATTN: Helene Berg Phone: 63 80 77 82; E-mail: Helene.Berg@ffi.no		**10. SPONSOR/MONITOR'S ACRONYM(S)** NATO
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The productivity of software developers has been an area of interest in the private industry in an endeavor to more appropriately interpret pertinent software development cost drivers. In an effort to better predict the cost of developing software, many have focused on Application Type as an important factor. The Department of Defense (DoD) has recently adopted a similar focus. Studies on categorization schemes of Application Types have shown increasingly relevant cost and productivity driving characteristics recently in the DoD. Identification of factors associated with distinct productivity effects is important for the defense acquisition and software cost estimation domains. Distinct factors can provide insight into what drives software productivity and cost. This research attempts to investigate the significance of Application Type and Super Domain in predicting productivity in software intensive defense programs. This current study analyzed 655 Software Resource Data Reports of DoD projects spanning the years 2001 to 2019. The analysis indicates the significance of Application Type in predicting productivity to be overstated for DoD. Only two of seventeen Application Types adopted by the DoD and one of four larger environmental settings called "Super Domains" proved significant. Regarding those, this study was able to identify characteristics that may prove more useful for understanding drivers of software cost and productivity.

15. SUBJECT TERMS

Cost, Software, Productivity, Application Type, Super Domain, Analysis of Variance, Contingency Table

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Lt Col Scott T. Drylie, Ph.D., AFIT/ENV
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, x4441 scott.drylie@afit.edu
U	U	U	UU	151	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18