Summer 2019

Too Agile? - DevOps Software Development Challenges in a Military Environment

Burl T. Kenner III

TOO AGILE? - DEVOPS SOFTWARE DEVELOPMENT CHALLENGES
IN A MILITARY ENVIRONMENT

by

Burl T. Kenner III

Bachelor of Science
Middle Tennessee State University, 1986

Master of Aeronautical Science
Embry Riddle Aeronautical University, 2007

Master of Business Administration
University of South Carolina, 2017

_____

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Engineering Management

College of Engineering and Computing

University of South Carolina

2019

Accepted by:

Gregory Gay, Director of Thesis

Jason D. Bakos, Reader

John R. Rose, Reader

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

DEDICATION

To Nestor, the model of an effective, professional Air Force staff officer.

ACKNOWLEDGEMENTS

ABSTRACT

Agile and DevOps software development processes now represent proven, commonly accepted models in commercial industry. For DoD military environments however, empowering teams of software developers and end user operators to build, test and release solutions without direct oversight and management of leadership is a brand new experience. In 2017, a DoD software acquisitions revolution occurred when approval was granted to scrap the former ineffective, cost prohibitive process and initiate the agile DevOps model. After one year since its introduction, this paper may represent the very first look at how well this nontraditional process is working out in a military command and control (C2) environment. More specifically, the study attempts to answer concerns by an USAF air operations center (AOC) headquarters that the process may not be delivering value to end user operators.

PREFACE

Any large organization will utilize a wide range of software applications and data sources to complete required processes, yet few if any of those applications were originally designed to work in concert with each other to achieve the organization's specific, custom process tasks. Until recently, military DoD organizations were subjected to an acquisitions process that forced software change requests to occur in lethargic 5-year recurring event (RE) cycle. When new applications were purchased, they tended to remain indefinitely, translating to antiquated, decades old software designs and creating a multitude of cross functional inefficiencies. As a result of this process, air operations center (AOC) active duty end user/operators were forced to develop customized workarounds - manual grease boards and Microsoft Excel macros served as 'band aids' to patch together the highly complex system of systems that was required to command and control as many as two thousand combat aircraft missions, or sorties, daily.

While this approach did work, it certainly wasn't efficient, and the agile DevOps software development processes that were revolutionizing the civilian business world weren't seen as compatible for use in military environments. Eventually, the AOC software acquisitions problem reached a boiling point, and in 2017, the DoD adopted agile. It's now been nearly one year running with the new DevOps process in place. This study seeks to determine if agile DevOps processes in a complex, high risk, combat command and control (C2) military environment actually delivers expected efficiencies

and effectiveness in a unique culture that many cautioned is not a fit for such a process

model. The study also seeks to determine if recently raised concerns by military

leadership are valid, or simply the result of the discomfort associated with any such non-

traditional change to a legacy process.

The contents of this thesis are unclassified, and were conducted with approval

from the military organization in which the studies and data were collected. For

proprietary reasons, the names of vendor companies and specific military organizations

were kept to a minimum whenever possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AOC .... Air Operations Center

AOC-WS... Air Operations Center Weapons System

AOR .... Area of Responsibility

ATO (AOC context) .... Air Tasking Order

ATO (software development context) .... Authority to Operate

C2...Command and Control

DevOps .... Development Operations

GAO... General Accounting Office

IC...Intelligence Community

ISR .... Intelligence Surveillance and Reconnaissance

ISRD ....Intelligence Surveillance and Reconnaissance Division

KREL .... Kessel Run Experimental Lab

MAAP-TK .... MAAP Tool Kit

MAAP... Master Air Attack Plan

TBMCS ....Theater Battle Management Core Systems

USAFCENT...United States Air Forces Central Command

USCENTCOM...United States Central Command

CHAPTER 1

INTRODUCTION

Military air operations center (AOC) operations, planning and intelligence

personnel have been frustrated for decades with an antiquated software applications

acquisition and development approach which provided blunt, inefficient software tools to

accomplish the complex mission planning and execution of air combat missions. A

cumbersome five-year long military software acquisitions process cycle afforded little

capability for improving applications such as the AOC's primary weapon system

application engine, Theater Battle Management Core Systems (TBMCS). At the heart of

all AOC system data integration, TBMCS and its AOC environment of other

applications, was best articulated in 2004 by a MITRE Corporation engineering case

study, which assessed designs as being severely hampered by the complexity of legacy

applications [TBMCS] and of the immaturity and complexity of commercial and third-

party off-the-shelf products. It also cited the arduous process of incorporating new

software products into the AOC system of systems (Collens, 2004).

In early 2003, AOC major combat planning personnel in the Middle East had

enjoyed a brief period of ad hoc agile DevOps process relief when the vendor for the

primary AOC plans application, Master Air Attack Plans Tool Kit, or MAAP-TK, was

allowed to seat code writers directly with operational plans end users, to improve the user

friendliness and combat efficiency and effectiveness of the air combat mission planners'

primary combat flight planning and scheduling application. The results were impressive,

1

and the application became legendary in the years to follow for its brilliant user interface and well-designed range of features. The accomplishment was even more impressive when the myriad of additional AOC applications and electronic system interfaces they were tied to is taken into account. Unfortunately, after major combat operations ceased in the summer of 2003, the lethargic DoD acquisitions process again took over and the AOC went right back to the ineffective, inefficient five-year 'spiral development' RE process - the 2003 agile DevOps software development anomaly that made MAAP-TK legendary for its intuitive usability and improved performance seemed to be all but forgotten. Additionally, MAAP-TK's expansive capabilities were severely limited by the core TBMCS system, which had been designed in the 1980's MS-DOS era.

Even still, the TBMCS core system application remains in DoD AOC's to this day, mixed with a range of old and new applications that were never originally designed to work together. This represented another aspect of the critical need for agile DevOps processes becoming commonplace – so that organizations can pull data from older legacy applications to a higher *cloud*, and thus refrain from rebuilding a ground-up suite of all new applications for a given organization.

In the following years, United States Air Forces Central Command (USAFCENT) AOC operators made up for such software inadequacies through creative manual workarounds. One such 'band aid' fix was jokingly called, the *real* system of record; *Microsoft Excel.* MS 'macros' saved the day in many AOC divisions, providing the capability to build unique, customized, macro-powered tools for planners and intelligence divisions. One of the more well-known products was informally named, 'The Generic Flow.' This Excel product contained dozens of macro functions designed in by those rare

operators who were familiar with macro creation. The generic flow pulled data from multiple outside sources, calculated complex air combat scheduling solutions, and produced power point briefing presentations for leadership. The problem was, those who understood macro function development were few and far between. Active duty personnel were on constant rotational deployments, and the customized tool had no formal vendor to make new modifications or fix features that 'broke.' If the generic flow product needed adjustment or updates, there was rarely anyone qualified to accomplish and the USAF C2 community certainly didn't provide any formal macro training. Such MS Excel products were abundant in the AOC, and arguably represented a 'poor man's' approach to agile DevOps software development.

The need for an agile DevOps solution was most pronounced in the AOC aerial refueling tanker plans cell, where highly complex mental calculations were required to establish fuel efficient solutions for a wide range of various airborne tanker and receiver aircraft. Attempts had been made over the years from outside vendors, to produce software that would solve the problem set, but since AOC software development was limited to the five year acquisition cycle, tanker cell planners suffered the use of wall sized manual grease boards in order to visualize and calculate solutions (Wallace, 2018). And even once those solutions were attained, the resulting data had to be manually input, or 'thumped,' into MAAP-TK. Worse, the mission data then had to be 'dumbed down,' in order to push it through the antiquated TBMCS air tasking order (ATO) application.

According to the FY15 Air Force Programs AOC weapon system (AOC-WS) review, AOC's were operating with, 'commercial off-the-shelf hardware, multiple core third-party software applications, web-based applications and additional third-party

systems that accept, process, correlate, and fuse command and control data from multiple sources and share them through multiple communications systems.' The software applications are designed by a wide range of outside vendors and pull data from a multitude of outside sources. To fully appreciate the complexity of the AOC systems, one must understand they operate on multiple LANs. These include the Secret Internet Protocol Router Network (SIPRnet), the Joint Worldwide Intelligence Communications System (JWICS), the non-classified internet protocol router network (NIPRnet) and multiple coalition LANs, as required (AOC-WS Overview, 2015).

In the highly complex, never ending cyclic processes of an AOC, multi-source, big data predominates, with a plethora of applications and data sets bound together in a mish-mash of code and process architecture that was never originally designed to operate in unison. The *air tasking order* (ATO) cycle creates an ATO, which is essentially a glorified flying schedule, and an *airspace control order* (ACO), that directs de-confliction of airborne assets in time and space. A *targeting cycle* creates a joint prioritized targeting list (JPTL), which is effectively a list of highly developed target sets that are sent to teams of air combat planners who apportion available strike assets to service the targets while other planners such as fighter, bomber, surveillance, naval, marine and coalition officers build missions into the larger system. These missions are then de-conflicted by combat airspace experts, sent into TBMCS for production, delivered and disseminated out to the operations floor of the AOC and sent to flying squadron aircrews and other joint combat teams for highly complex, coordinated mission operations. Once aircraft are airborne, missions are conducted, weapons are delivered,

assessments collected and the cycle repeats all over again with strategic guidance by leadership initiating the process.

The intelligence community follows a *collection management cycle* (for collection of intelligence imagery and surveillance) that creates a *reconnaissance surveillance targeting annex* (RSTA), which is effectively a de-conflicted schedule for what the intelligence surveillance and reconnaissance (ISR) assets *look* at, and *listen* to. Above all of these cycles is an overarching *intelligence* (PCPAD) *cycle* (Figure 1.1). These process cycles constantly feed into each other, relying on virtually too many data sources, systems and applications and web-based applications to count.

In this super-complex, big data, multi-system of systems environment, the need for an agile DevOps solution to achieve better ways of doing business was reaching fever pitch. Major combat operations require as many as 2,000 air sorties per day, with tens of thousands of calculations and data sets being required, and the need for a modernized process method to upgrade the associated range of software was significant.

Meanwhile, agile DevOps processes, which have been commonplace in the commercial world, were being steadily resisted in the traditional DoD military culture. Agile concepts involve effectively removing traditional management oversight, allowing end user operators to request changes directly with software developers, and this was not a fit for risk averse DoD processes and policies. DoD conversations *were* occurring, but it was beginning to seem unlikely that such a cultural shift would ever happen (Broadus, 2013).

Intelligence Cycle (PCPAD)

Evaluation and Feedback

Planning and Direction

Collection

Dissemination and Integration

Analytics and Production

Processing and Exploitation

CC's Objectives

End State/CC's Objectives

Assessment

Joint Collection Requirements (CR's)

Target Development & Prioritization

Assessment

CC's Guidance

Mission Planning & Force Execution

Joint Integrated Prioritized Target List (JIPTL)

Mission Planning & Force Execution

Joint Prioritized Collection List (JPCL)

Component Coordination

Target Development

CC's Decision & Force Assignment

Capabilities & Analysis

Strategic Assessment

Recon Surveillance & Targeting Acquisition Annex (RSTA)

Combat Assessment (BDA)

Joint Integrated Prioritized Target List (JIPTL)

Targeting Cycle

Collection Management Cycle

Weaponeering Allocation

Force Execution

Air Tasking Order (ATO/ACO) Development

Master Air Attack Plan (MAAP)

Air Tasking Order Cycle

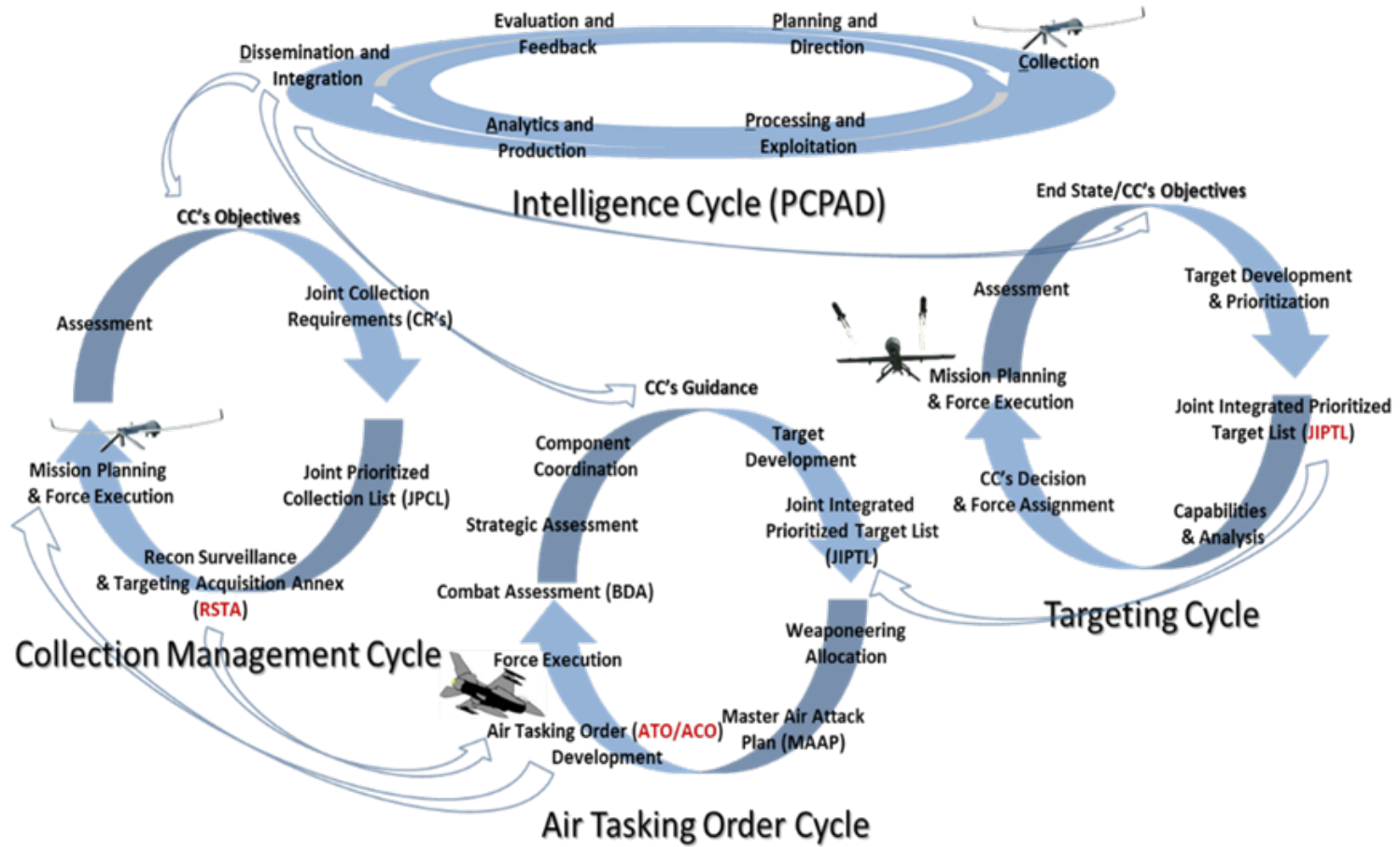Figure 1.1 Air Operations Center and IC ISR Processes (Kenner, 2019)

CHAPTER 2

PROBLEM BACKGROUND - SEARCHING FOR A SOFTWARE
DEVELOPMENT SILVER BULLET

Air Force AOC weapon systems (AOC-WS's), otherwise known as AN/USQ-163 'Falconers,' have historically operated using software development acquisition contracts that service numerous third-party software applications with 'spiral upgrade,' Recurring Event (RE) test cycle phases. Following the traditional, linear sequential *waterfall* software development design model approach, these RE phases require five years to complete – that's five full years from the time an operator requests a change to an application, to the time he/she may, or may not, experience the upgrade.

Meanwhile, active duty personnel continuously rotate out of their respective tours of duty at AOC's in anywhere from one year to as little as 3 months. Additionally, in the world's only fully engaged combat AOR, USCENTCOM, the war-fight moves, changes and morphs, with continual process and tactical evolutions occurring on a daily pace – five years was entirely unacceptable for modern command and control requirements. AOC-WS's represent highly complex system of systems that receive, process and fuse command and control (C2) data from many sources and communications systems, and the RE test phase process delivered software development solutions that were largely unsatisfactory on average, taking five years between change request and delivery and realization of very few meaningful upgrades (AOC-WS Overview, 2013-2016).

Additionally, fielding inadequacies translated to unsuitable operational deliveries that couldn't be configured, maintained or troubleshot (Collens, 2004).

Such fundamental flaws in acquisition strategy, combined with stifling bureaucratic guidelines, translated to a severe lack of software innovation. The 2008 Weapon System Software Management Guidebook, produced by the Secretary of the Air Force for Acquisition (SAF/IAQ) revealed a culture that was highly risk averse. Concerns over potentials for software risk of failure drove a priority on retaining legacy applications. Control and reduction of risk was deemed the only priority. The guidebook even went so far as to state that the 'raising of expectations' was considered negative, as such an approach also raised risk! *Continuous oversight* was demanded of program offices in design, development testing and review. Verification of all activities via metrics and formal technical review meetings was required, in order to, 'manage risks and address problems.' Highly complex war-gaming simulations, integration studies, source code rights reviews, vulnerability assessments, ethics standards and a plethora of defense and mil-standard regulations added even more layers to the problem (Weapon System Software Management Guidebook, 2008).

2.1 Origins of the DoD Software Development Agile Acquisitions Revolution

While the commercial world had begun to appreciate the innovative value of agile software development in the early 2000's, the DoD was an entirely different culture. Traditional military mindset was a looming barrier to change, with vastly different hierarchical structures, individual vs. team reward systems, high demand for activity and process documentation and top down communications with external regulations and policies driving work efforts (Mergel, 2016). Most significant of all was the culture of

oversight, formal test and review for development, which used the classic *waterfall* method (Mergel, 2016). Waterfall models represent the very first traditional method and consist of a linear, sequential design approach, using phases such as conception, initiation, analysis, design, construction, test, production or release and maintenance. It's simple to understand and to use, but each phase has to be completed before the next, it's not desirable for complex projects that have constantly changing requirements, end user feedback can only be inserted at the beginning phases and documentation is cumbersome. Each phase depends on the deliverable of the previous, so it's relatively slow, inefficient and expensive. It's also considered effective in mitigating risk, but that aspect is debatable. Since iteration loops between phases are typically essential, the model isn't efficient and in some cases, isn't effective. Errors aren't discovered until very late in the cycle and fast changing environments can only make change requests at the start of the cycle (Lapham, 2014).
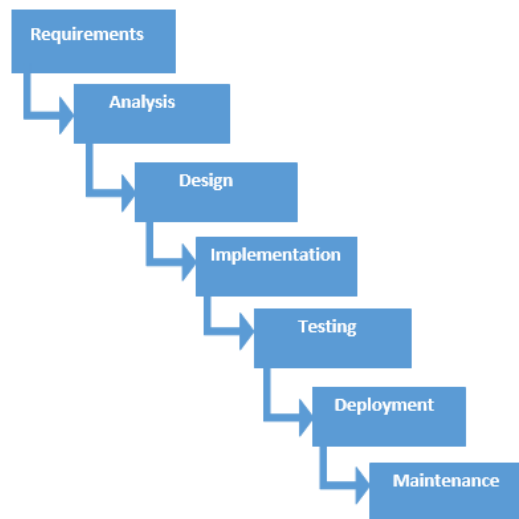
Figure 2.1, Waterfall Software Development Model

As DoD acquisitions began to understand that software development was its biggest problem, waterfall methods were seen as the primary cause. Agile software developmental models, which make rapid, consistent updates and eliminate bugs along the way, was seen as the answer (Maucione, 2018). Agile methods are *iterative*, as opposed to linear, and solutions evolve using cross-functional teams who collaborate. The process allows continuous improvement and rapid, flexible responses by developers to change requests. Deliveries are early, continuous and adaptive, and while the model is often understood to require no formal documentation, process or plan, such notions are a myth. Agile does require documentation, it is a formal process and it certainly requires planning (Lapham, 2014). Other myths include; hard choice between agile or waterfall - but not both, 'agile is a fad,' 'agile won't work in DoD military environments' and, agile is a 'cowboy' approach (Lapham, 2014).

Another mistaken concept is that agile is the same as *incremental*, *spiral* or *iterative* and that it's simply renamed. Reality is that agile is a formal process with a roadmap to a final delivered capability, just like waterfall. The big difference is that agile allows delivered capability *along the way*, as well as providing opportunity for continual improvement (Lapham, 2014). The Titanic was large, complex and did have forward speed, but the big ship couldn't turn quickly. An *agile* Titanic would accelerate, stop and turn very quickly, and, *if* it had been built using the agile processes, the substandard steel issue could have been more easily detected, reported and resolved during the build process. Ultimately, agile is summed up best in what's called, the *agile manifesto*, a list of twelve concepts guiding the movement. It emphasizes individuals and interactions over processes and tools, working software over comprehensive documentation, customer

collaboration over contract negotiation and responding to change over following a plan –

radical concepts in a traditionally highly regulated government environment.
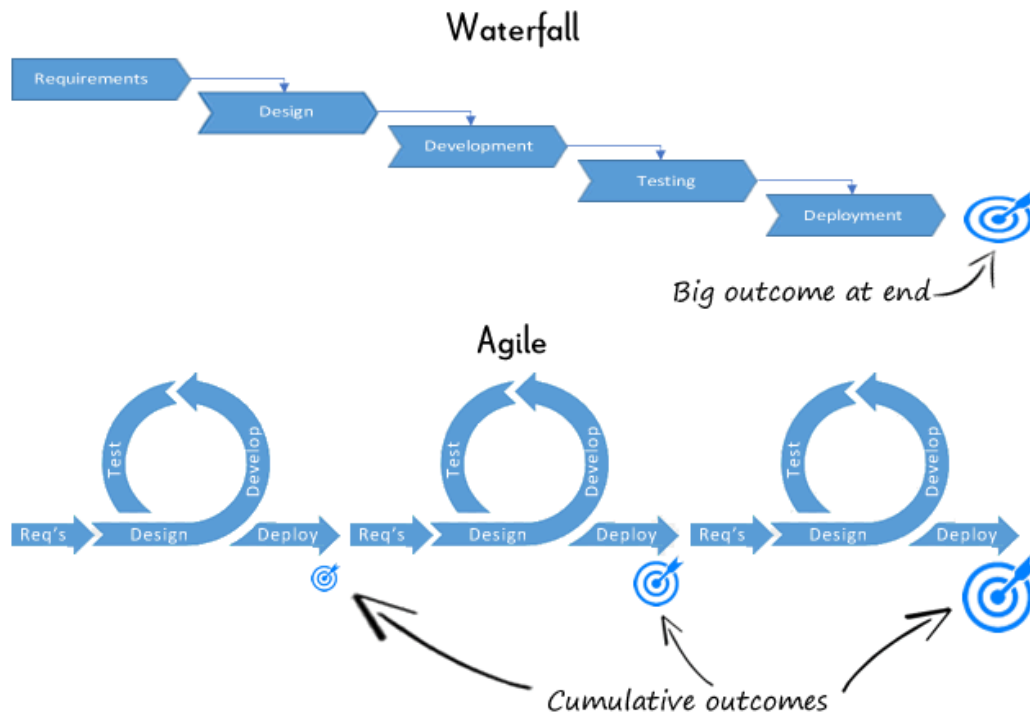

Figure 2.2, Waterfall vs. Agile. Source: Schaeffer, 2019.

The *DevOps* software development model, on the other hand, can be added to the

agile process and adds the innovative concept of allowing direct collaboration and

communication between code writing developers and end user operators. This

combination enhances the organization's ability to produce delivery even faster, and with

innovative solutions (Ott, Pham, Saker 2015). DevOps also involves high levels of

automation. This allows for programming to occur very rapidly, with the update and test

of code for millions of lines in drastically reduced periods. Raytheon discovered 80

hours of coding could be accomplished in as little as 3 hours, 'shrinking release cycles from years to weeks' (Insinna, 2017). In order for DevOps practices to work best however, it assumes identification of qualified, experienced operations subject matter experts. Proper identification of these operational SMEs is known as, *crowd sourcing*.

Unlike agile processes, DevOps benefits are less understood, but the results of proper implementation are impressive (Riungu-Kalliosaari, 2016). The downside, if there is one, is that for a culture embedded with a traditional waterfall process to transform can be challenging.
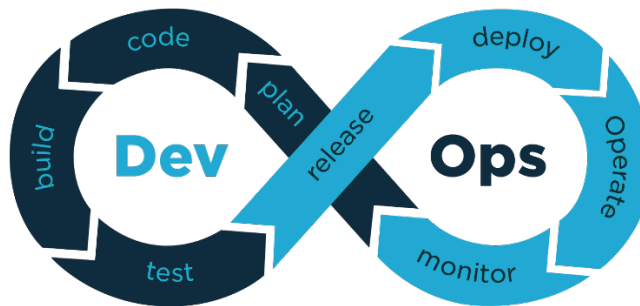

Figure 2.3, Development-Operations (DevOps) Software Development Model

In the case of C2 environments, there was tremendous resistance to change, due to the highly complex nature of the critical warfighting system of systems infrastructure. *DoD Agile Adoption: Necessary Considerations, Concerns and Changes*, addressed the hard reality of such road blocks in 2012, but emphasized the need for agile methods that would translate to the, 'iterative and incremental approach to software development in a highly collaborative environment with just enough ceremony to produce high quality software that meets the needs of its stakeholders,' (Lapham, 2012).

Lapham emphasized DoD adoption wouldn't be easy but that it didn't translate to *total abandonment* of traditional methods. It stressed *neither* approach solved *all* problems but that an agile DoD, with team focus on rewards and '*just enough*' documentation is urgent, and that the DoD has no choice but to accept the process. It concluded military staff must become accustomed to the new practices and that they cannot just *adopt* agile, they must *become* agile, and accept it into their culture.

Later that same year, a DoD-systems engineering conference in San Diego countered the heated debate move towards agile, by warning that the DoD might just be moving towards another, 'management best seller' (Paparone, 2009). The USAF and DoD had learned the hard way over the decades with concepts that had been touted as a silver bullet to success but had instead become embarrassing failures. TQM, reinventing government, management by objectives, balanced scorecard, lean, Six Sigma®, were all buzzwords in the DoD that didn't produce the desired effects. The San Diego conference stressed that *balance* should guide any such transition. Further, the DoD should not confuse the need for *warfighting* agility with agility in the acquisition of software developments - faster acquisition of newer and better is not always the needed solution, they emphasized. The analogy of the Titanic was highlighted, in that it had speed, but not agility, and so neither should the DoD rush its software development. The conference also highlighted the profound importance of a foundational military concept; trade-offs between speed (quantity) vs. quality, which has always been a strategic point of debate.

Meanwhile, another key to the debate is stakeholder interests. Identification of the *actual* user, the customer and the software developer is essential for mission success, which is defined as; 'the achievement by an acquired system to meet not only specified

performance requirements but also expectations of users and operators in terms of safety, operability, suitability, and supportability.' Further, 'mission success is evaluated *after* operational turnover (Guarro, 2007). In the case of the AOC-WS, the actual user is the *combatant commands* and the actual customer is, *congress*. DoD 5000.02 acquisitions regulation and joint capabilities integration and development system (JCIDS) at that time, was far removed from the actual user and customer (DoD 5000.02, 2007).

Whatever process that was chosen, it was clear that some level of flexibility was critical. For example, SCRUM processes, which are similar to agile, but instead have a pre-determined launch and completion date, could also be considered, as long as *flexibility* was built into the process. As for the primary reasons for adopting agile, proponents listed; acceleration of delivery, enhancement of management's changing priorities, increase in productivity, quality enhancement, simplification of the development process, cost reduction and improvement of the alignment between IT and objectives. On the other hand, top causes for concern were; loss of managerial control, lack of upfront planning, leadership opposed to change, lack of documentation, lack of predictability, lack of engineering discipline, regulatory compliance and, interestingly, reduced software quality. Another concern expressed was a lack of consistent, reliable metric data to verify if the reasons for implementing agile were being met.

So, while agile may be the current buzz word in commercial industry, it was put into doubt if it was necessarily the right path for the DoD. Choosing a development method that fits the military C2 cultural environment, that prefers order over chaos, and that operates in highly critical [life threatening] 'project' scenarios, with steady state requirements, might indicate *anything* other than an agile solution (Boehm, 2004).

Interestingly, the *Agile Manifesto*, which was coined in the 2000's, included several primary tenets that traditional waterfall DoD software acquisitions guidelines also *claimed* to adhere to; iterative life cycle processes, intermediate release with evolving levels of details and demo based approaches to assess intermediate results, all point to the need for agility (Royce, 1998).

So, was agile methodology the *silver bullet* that could fix what had become known as a broken acquisitions system (Brooks, 1987)? The 2006 Defense Acquisitions Performance Assessment (DAPA) recommended a *two-year* recurring process – much quicker than the then current five year cycle, and a 2011 GAO study clearly communicated that Agile wasn't the answer for *all* acquisitions, but only for resolving immediate *budgetary* funding issues (Chaplain 2011).

Ultimately, the San Diego conference summarized; 1) software needs are typically, poorly understood and there is no silver bullet, 2) agile software development still does not seem to have all the answers. The biggest surprise was a stark reminder that the [then current] cumbersome, ineffective, inefficient spiral development five year acquisitions process, which had been introduced in 1988, and which became the DoD 5000.2 standard in 2003, had *also* been touted as *the* premiere revolutionary software development solution of the era (Hantos, 2012). Advocating largely against the cultural changes agile development would bring, the conference emphasized agile methods represent an extreme adaptive process model, and there was strong resistance to incorporating the change into the military environment. At the time, it seemed unlikely agile methodology would be introduced.

Figure 2.4, The 12 Agile Principles and The Agile Manifesto

2.2 The DoD Agile Revolution Begins

In contrast to the San Diego conference's implication that adopting Agile might

not be the right choice, a Deputy Secretary of Defense for Systems Engineering study

leaned more towards ensuring that *if* Agile was accepted, then customized efforts must be

made to ensure its success. Arguments against the change began to collapse in early

2017, due to glaring financial negatives. The AOC-WS 10.2 contract was suffering costs

ballooning from $374 to $745 million and delivery upgrades were years behind schedule

(Insinna, 2017). In April, 2017, the Senate Armed Services Committee cancelled the

16

AOC-WS 10.2 RE upgrade and a recommendation was made to collaborate with the office of the secretary of defense to implement industry standard agile software development processes into the AOC-WS (AOC-WS Overview, 2018).

Things rolled quickly after that, and by July of 2018, the Air Force authorized 'alternative approaches,' which resulted in the 'AOC Pathfinder' project to utilize a software development technique called, DevOps, short for development-operations, to replace the former RE process. The move authorized the approach in order to achieve faster development, but only after a cybersecurity Cooperative Vulnerability and Penetration Assessment (CVPA) had been completed. Eventually, AOC-WS Mods Block 20 began execution by the Kessel Run Experimentation Lab (AOC-WS Overview, FY18). Project Kessel Run Experimentation Lab (KREL) is managed by the Air Force Life Cycle Management Center (AFLCMC) at Hanscom AFB, which oversees sustainment and upgrade of the AOC weapon system combat software.

KREL's primary initial focus was on the AOC that oversees air combat in the Middle East, and our research focuses on their efforts there, as it represents the only fully engaged combat area of responsibility (AOR) since 1991. Located in Boston, their 90+ work stations and processes are modeled after another prominent agile development and innovation company, Pivotal Labs. Currently utilizing an eight world-region AOC cloud platform, they are constantly growing, and active duty airmen work alongside code writers, to modify and build the next generation of air combat C2 software (Newel, 2018).

2.3 Defining the Problem and Hypothesis

Since KREL startup, almost one year ago at the time of this writing, many AOC

C2 applications were designed, developed and delivered using a pure agile methodology.

KREL code writers worked with an authority to operate (ATO) that held virtually

unlimited powers. 'ATO,' in this instance, reflects the ability by KREL to design,

modify and deliver virtually any request made by an active duty AOC end user -

immediately. The result has been a fast-moving software production line capability that

has delivered many new applications designed to be innovative and to simplify end users'

application functionality and work load and increase AOC process efficiency and

effectiveness. Predominate application offerings relevant to this research that KREL

developed include;

- o ARC – The intelligence surveillance and reconnaissance (ISR) planning tool designed to eliminate manual, error prone processes, streamline critical planning products and enable collection managers to rapidly respond to last minute, dynamic tasking requests.
- o ATLAS FULCRUM – A production environment hub for all deployments to the operational enterprise providing environment parity and application up time for war fighters.
- o BLAZON – A target system analysis tool designed to modernize the monolithic, months long process into a more responsive capability to provide up to date information to the intelligence community (IC).
- o CHAINSAW – A dynamic targeting tool designed to replace manual data transfers between multiple tools, providing single source of ground truth on dynamic targeting for the AOC operations floor.
- o DINGO – An in-theater aircraft status tool designed to confirm an air asset's location and capability. Dingo provides immediate updates for AOC planners, leadership, flying squadrons and maintainers.
- o HOLOCRON – A geo-spatial data distribution tool designed to improve the efficiency of geospatial intelligence (GEOINT) data distribution.
- o HOME ONE – A digital strategy to task tool designed to provide the AOC strategy division an ability to create digital products to track changes across ATO cycles.

- o JIGSAW – This application was one of the very highest priorities. Jigsaw is a tanker planner tool that allows tanker planners in the aerial refueling cell, to utilize an application solution, as opposed to their former manual process, that offered visualization of the highly complex algorithms required to make air refueling receiver requests and mission feasibility both efficient and effective. Jigsaw allowed solutions without the use of the infamous, manually cumbersome, *tanker whiteboard* – a large grease board that spanned an entire wall and required a wide range of color coding, trial and error and weapon system bartering to resolve daily mission planning scheduling.
- o KRUXE – A software development environment automation and process tool on a cloud native platform designed to provide consistent, sustainable services for KREL's developer community.
- o MARAUDER – A dynamic, modern intelligence mission report analysis application for collection, consumption and analysis of mission data to assess operations and drive future mission planning.
- o PAPERCUT – A digital air operations directive (AOD) creation tool for use by the AOC strategy team to enhance collaboration with other AOC teams.
- o RAVEN – Designed to decrease target development workload, Raven is a target production/electronic target folder manager designed to be easy to consume, with a cradle-to-grave target workflow management capability.
- o RIPSAW – Designed to serve the target development community, Ripsaw pulls data from many sources such as the modernized integrated database (MIDB), then stores the data for long term event stream visualization.
- o SLAPSHOT – Built as a replacement for the iconic 'Generic Flow,' used by the master air attack plans (MAAP) cell, this application visualizes the pairing of air combat missions with respective aircraft and guides the creation of the air tasking order (ATO). It automates many manual tasks to include data pull, data entry and quality checks, some of which are performed by weapon system operators, and even pilots in the field, adhering to arduous constraints.
- o SPACER – Designed to alleviate the cumbersome, manually intensive combat airspace management systems in the AOC, allowing intuitive requests, allocation and use of joint airspace.
- o TRITON – This tool dynamically generates the air tasking order (ATO) and is designed to greatly enhance the formerly manual intensive process.
- o YETI – A toolset designed to enhance the efficiency and effectiveness of space based overhead persistent infrared (OPIR) battlespace awareness center intelligence products.

While acceleration of delivery aspects had been resolved, perceptions by some USAF leadership almost immediately began to pour in with regards to loss of managerial controls. AOC leadership seemed to have no mechanism to track changes or to ensure quality control to the very processes they were responsible to maintain. There were even

rumblings that questioned the quality of the software itself. Agile DevOps authors

Boehm, Guarro, Royce, Chaplain and Hantos had all but predicted this eventuality

(Boehm, 2004, Gurrro, 2008, Royce, 1998, Chaplain, 2011, Hantos, 2012). However,

were leadership's concerns valid? Were the KREL code writers' and project managers

granted authority now become the equivalent of Billy the Kid's deputized Regulators

running wild in the AOC? Are the software improvement metrics being measured? Are

end user AOC operators being serviced with timely responses to their requests, or was

KREL simply launching out on its own, free to build with no experienced C2 oversight?

Where did the process need to be adjusted, if at all? The questions being asked

represented a text book concerns that many authors had warned of previously that would

happen in a military environment, and the combat risks associated with a fully engaged

combat AOC demanded answers.

The old RE cycle ways of doing business were certainly ineffective, but the new

process seemed to lack balance. A hypothesis developed during conversations with

innovation offices, leadership and senior analysts that pointed to the fact that the new

process might just be *too agile*. While indeed innovative, it seemed to lack the flexibility

spoken of by the experts. It seemed to require a customized tweak to inject oversight,

and possibly even *management*. Additionally, stakeholders, customers and end users

didn't seem to have been clearly defined, with proper roles understood. The Middle

Eastern AOC's higher headquarters is USAFCENT, however, the entity calling the shots

for agile DevOps acquisition was Air Combat Command (ACC), at Langley Air Force

Base, which is a force provider, not a war fighting combat command. Was their role

being abused, with no apparent avenue for the war fighter to shape the development of

their own AOC? Most discomforting of all, AFCENT higher headquarters leadership

didn't seem to be receiving any timely metrics on status, changes, quality. In short,

communication was not occurring between KREL and arguably the most important

stakeholder.

This study sought to collect data, via both interviews and surveys, from subject

matter experts in a full spectrum of roles, that would help validate if the war fighting

headquarters leadership concerns are valid, and if there are solutions to adjust a new,

possibly immature process.

CHAPTER 3

ADDITIONAL LITERATURE REVIEW

Since agile DevOps processes were only introduced into military environments recently, little to no literature currently exists that has studied the results of its introduction, but robust analysis exists that was preparing and recommending it's inclusion. *Thinking About Agile in the DoD*, a 2013 government summit study by the Deputy Assistant Secretary of Defense for systems engineering, emphasized that agility is demanded and that current regulations must be waived and modified by congress itself. Current process challenges were completely unsatisfactory and waterfall and spiral models locked down growth. The study highlighted risk has to be balanced with urgency and customer performance priorities must be identified. It also understood that vendor incentives have to be tailored and that active stakeholder involvement and feedback, after release and delivery must be accomplished. Bottom line statements were made emphasizing program management attention is required to address the challenges. It also pointed that agile does not translate to no documentation or artifacts, as some have errantly believed (Welby, 2013).

Another study that looked at the challenges of being agile in the DoD, was a 2013 quarterly publication with an article by the same name. It also clearly identified governmental agencies as needing to adopt agile, but admitted it's new territory, full of unfamiliar processes and unprepared stakeholders to adapt. Like the Welby analysis, Broadus emphasized user feedback as essential for agile value (Broadus, 2013).

Agile Innovation Management in Government, a 2016 quarterly publication article

also echoed the aforementioned patterns required for governmental agile success:

outdated systems demanded acquisitions process changes to agile due to over budget,

behind schedule projects. Interestingly, it pointed to the failed Healthcare.gov as a

catalyst for U.S. government change (Mergel, 2016). Ultimately, and nearly

unanimously, almost every study on agile in government identified that success will

come not just from adhering to a rigid agile concept, but from stakeholder's flexible

feedback into the process. If agile is going to be successful in the AOC military

environment, not only does industry best practices equate to airmen communicating

software requirements directly to developers throughout the life of the system, but

stakeholders must also retain some degree of flexible form oversight feedback and

communication.

According to DORA Research and Investment, agile has, 'won the methodology

wars.' In DORA's *2016 State of DevOps Report*, over 25,000 world-wide surveys were

conducted confirming the agile DevOps process delivers phenomenal increases in

productivity, savings and value (Puppet + DORA, 2016). However, it emphasizes

software quality is notoriously difficult to measure. Software quality is highly subjective

and it's highly contextual. One end user may have a glowing opinion while another a

glaring criticism. One may feel the process is revolutionary and another that it is a sham.

DORA discovered the only way to accurately measure the software process is by a

perception of *value,* from its end users.

No literature review of this topic would be complete without an in-depth mention

of the 2018 *Kessel Run* overview by the Air Force lead data scientist for Kessel Run at

the Air Force Lifecycle Management Center (AFLCMC). The presentation consists of a 42-slide briefing prepared to articulate the requirement and value of agile DevOps, as well as user-centered design methodology, that was happening in the Air Force software acquisition world (AFLCMC, 2018). The presentation states a mission to continuously deliver war-winning software their Airmen love. It conveys a refreshing, passionate, excited tone of agile DevOps and user-centered design being introduced into the once stodgy, waterfall RE cycle world. It emphasizes *true* agile software development principles being brought into the USAF as revolutionary. It blasts the use of buzzwords, and emphasizes a real effort to change. It touts the highly successful tanker JIGSAW application as saving tremendous amounts of money, due to fuel savings for combat missions, and the soon arrival of MARAUDER and other anticipated applications.

One of its core themes demonstrates minimum viable products (MVP), which is doing the smallest thing you can do to start a next step. Think; skateboard to steerable scooter, then bike, then motorcycle, then car. The concept is to build on what is available, not re-invent from ground up. Combined with short feedback loops where you 'build it,' 'measure it,' 'learn from it,' and combined with accelerated code test development tools and co-located teams, a recipe for success is created. KREL applications are demoed to include Raven targeting folder, Chainsaw dynamic target developer, Marauder mission analysis planner, Arc ISR collections deck manager, Ripsaw integral database and applications organizer, Slapshot mission and air combat sortie flow visualizer and Jigsaw aerial refueling tanker mission planner.

JIGSAW, arguably KREL's most impressive achievement, is perhaps best represented by a 2018 article entitled, *The U.S. Air Force Learned to Code – And Saved*

the Pentagon Millions (Wallace, 2018).. The aerial refueling planning cell in the AOC

plans as many as 250 air refueling missions per day during major combat operations,

seeing offloads of as much as 9 million pounds of jet fuel, each day. The calculations are

extremely complex, with multiple tanker and receiver aircraft receptacle systems and

refueling speed and altitude requirements and due to this complexity, no software

application to date had ever been created that would allow visualization of the daily

problem set.

       To resolve daily fuel efficient solutions, the planners were forced to utilize a

white tanker grease board, the size of an entire wall. The board was basically a flying

schedule with moveable magnetic pucks representing receiver and tanker aircraft that

looked somewhat like playing a Tetris game. The board was required in order for

planners see the entire picture, moving pucks and bartering with receiver aircraft in order

to resolve the ever changing plan. Once resolved, a 'gonker' places missions manually

into a custom built Excel-macro 'gonkulator.' These mission would then be manually

'thumped' into the MAAP-TK air attack planning application, pushed on to TBMCS, de-

conflicted by airspace planners, de-classified into multiple coalition versions and

distributed to the flying wings.

       The process was a complex nightmare that was finally addressed and resolved by

KREL. In 2017, KREL developers worked directly, face to face, with seasoned, highly

experienced tanker operations planners, just as the agile DevOps manifesto directs. The

result was the JIGSAW application that saves the USAF millions in personnel and fuel

savings every day (Wallace, 2018).

CHAPTER 4

STUDY

This study sought to collect data via both interviews and surveys from subject matter experts in a wide range of roles, that would help validate if the war fighting headquarters leadership concerns were valid. It also sought solutions to adjust what appeared to be a process that might be *too agile*. In order to attain data that would validate if the AOC DevOps process required some refinements, it was determined that surveys alone might not provide the open ended, free flow of ideas needed. Additionally, since the AOC is geographically separated from the higher headquarters by half a world away (AFCENT HQ is located in the eastern U.S. and the AOC is in the Middle East), a survey was developed for send to the AOC. Questions for the surveys and interviews were very similar and were designed to allow consideration for any issues KREL might be experiencing – a one sided study, representing only the experiences of the end users or of AFCENT leadership was seen as a threat to validity.

A total of fourteen qualitative interviews took place over the course of approximately eight months, using a general interview guided approach (Turner, 2010). Although open ended questions and discussion was the interview pattern, a series of questions was sent to the interviewee prior to the interview for review. Subject matter experts from USAFCENT and Kessel Run personnel were initially desired, to ensure a balanced result. However, due to operational concerns (OPSEC), the survey was directed to be sent out over the military SIPRnet system and this system did not allow convenient

access to KREL personnel in Boston. This lack of KREL personnel access represented a strong threat to data validity, so an effort *was* made to ensure interviews contacted KREL personnel. Regardless, the final data results did not allow for robust KREL representation.

Other personnel included AOC leadership, the AFCENT innovation division, AFCENT/A6 and ACOMS communications directorate, AOC A32 ISRD operators, personnel in the AOC master air attack plans (MAAP) cell, AOC operations floor personnel, 609 AOC Detachment 1, Kessel Run leadership and software developers, Air Combat Command (ACC) and the MITRE Corporation.

Interviews used the following questions for structure, but due to operational time constraints, exhaustive response captures were impractical – interviewees provided contextual offerings that enhanced awareness of the subject matter based on a preview of the questions. The same questions for the unclassified *survey* was distributed on a classified network by the innovation office to participants. Also found in Appendix A, questions included; relationship to AOC DevOps, specific DevOps software application/s and if the applications are reliable or not. It also sought to determine if the development process had provided applications that make tasks more efficient and effective, if users still rely on Microsoft Excel to manage associated war fighting tasks, if the process is providing transparent, relevant updates to AOC leadership and if leadership should have the capability to inject more direction into the process.

Others were; did the process provide timely responses to end users and were Kessel Run priorities correctly aligned with AOC warfighting priorities? Not wanting the survey to come across hostile towards KREL, it was asked, are AOC rotations

exasperating the development process by creating an environment with too many inputs for change coming from too many end users? In order to ensure capture on crowd sourcing, we asked, do you feel the process is correctly identifying the full range of required war fighting experts to build resilient and flexible applications? Reliability and user friendliness was addressed with; once operating within a Kessel Run application, do you experience upgrades that happen behind the scenes and do not require retraining or troubleshooting?

Finally, it was asked, if feedback loops between users and development teams allowed provision of inputs to product decisions, do you believe Kessel Run applications represent the next most valuable thing, the simplest way, as fast as possible and if participants felt Kessel Run user-centered design approaches ensures delivery of value to users. Totaling sixteen questions, the last asked if participants felt interaction with Kessel run developmental teams was exciting and allowed user testing to spark ideas for additional innovation. A seventeenth question was open ended for any desired statement and all questions utilized a standard, clear response matrix of, strongly disagree, disagree, neutral, agree, strongly agree and not applicable.

CHAPTER 5

ANALYSIS AND THREATS TO VALIDITY

With a primary focus on the entering hypothesis that the new AOC DevOps process

might just be *too* agile, we analyze the data results in light of the original set of questions

that AFCENT higher headquarters proposed.

- o Are AFCENT higher headquarters leadership concerns valid?
- o Have KREL code writers and project managers been granted too much authority?
- o Are software improvement metrics being measured?
- o Are end user AOC operators being serviced with timely responses to requests?
- o Where did the process need to be adjusted, if at all?
- o Was the role of the (ACC) force provider being appropriately followed?

5.1 Interview Results

A total of fourteen interviews were conducted over the period of late summer

2018 to early summer 2019. While not necessarily considered hard data, as per a survey,

the results provided free thinking, candid responses in the spirit that agile manifesto

concepts might dictate – *individuals and interactions over processes and tools*, is a

fundamental manifesto concept. Open ended, informal interviews therefore would

represent a good fit. Highlights from those interviews follow.

1. An AOC ISRD leadership role was identified who provided a thorough, hands-on end user perspective. This individual was a tactical weapons school graduate and represented the cutting edge of USAF tactical and operational ISR and AOC C2 knowledge. He stated, end users are providing inputs to KREL developers, but never receive status updates of the request and no mechanism exists to provide end user operators with updates on their change requests. KREL customer service is nearly non-existent and creates moderate levels of frustration and a perception exists that the KREL contract has no obligation to report any feedback to the

customer. His take was that KREL perceived agile to mean full control and no responsibility to report outside KREL. Went so far as to say that internal KREL decision making process apparently isn't in line with AFCENT AOC *combat* priorities. Estimated time to get fixed (ETIG) is non-existent, KREL operates on their own schedule with no AFCENT or other customer oversight at all. Reported that TBMCS is 'going away' soon – this represents a huge development. However, AFCENT higher headquarters is not aware and there is no KREL responsibility to communicate significant developments that effect the war fighting processes to the very higher headquarters responsible for the war fight! There's also no knowledge or perception of metrics showing improved cost, personnel or time savings.

2. Another interview captured the thoughts of a higher headquarters AFCENT intelligence training division leader who echoed all of the aforementioned weapons and tactics patch individual's concerns. He added that KREL is actually *refusing* to provide metrics, status reports or measurements of success to AFCENT leadership.

3. A software engineer who resides at AFCENT higher headquarters, but who has direct lines of responsibility into AOC KREL development equities, offered that the core of the KREL problem is not KREL contractual misunderstandings or even cultural failures. Instead, the constant rotation of USAF active duty personnel at the AOC exasperate the process with a never-ending flow of bright ideas and who expect immediate answers. Also, many of these constantly rotating personnel do not represent subject matter experts in their field – their inputs may not be valid or based on experienced knowledge of the application, and definitely do not represent major combat operational (MCO) based experience, which is of significant importance to an AOC capability. KREL personnel are flooded with too many ideas from temporary end users from other commands and AORs, who typically are anticipating rotation out of the AOR and don't have full ownership equity in the outcome.

4. A second systems (process) engineer echoed and concurred with all previous statements as they were interviewed together. He offered there should be weekly VTC's to provide updates on this fast-moving developmental process and that rapid 'spiral tests' are now conducted, 'on the fly,' to validate the code. Both of their take was that the overall DevOps concept is a phenomenal development that is revolutionizing AOC processes and applications. They provided a demo of SLACK, a chat tool that Bostonian KREL developers use to collaborate with AOC personnel.

5. A senior operations duty officer (SODO), with robust AOC background experience offered positives on the newly released SLAPSHOT application. SLAPSHOT replaced the MS Excel 'Generic Flow' and is very user friendly. It allows for clear viewing and interpretation of complex sortie flows on his SODO

work station screens. His feel was that agile DevOps process is allowing for positive software application changes – no complaints.

6. The commander of a C2 squadron had negative views of SLAPSHOT. He said SLAPSHOT is only suited for current operations, but is not designed, nor is capable, to provide data interpretation for major combat operations, which experience as many as 100 times the sortie rate as typical current ops. This statement revealed something important, that had not been addressed by any other source up to this point - the inclusion of major combat operational environments.

7. A seasoned operations trainer who resides at the AOC Detachment 1, was very happy with the DevOps products, stating end user inputs were being respected and that functionality and user friendliness of the new operations floor applications were impressive. He did counter however, that KREL refuses to offering user's manuals – KREL touts that the software is, 'intuitively obvious,' and requires, 'no manual or checklist.' He spoke very highly of CHAINSAW, the dynamic targeting tool used by the operations floor personnel to track changes in real time, in that it has proven to be very effective. He even went so far as to say that the application was instrumental in directly enhancing successful results during a specific military operational campaign. The same individual then stated that others in the dynamic targeting cell held an opposing view, and that they held that the application was considered, 'half baked.' Important to note; the individual revealed he had no actual experience using the applications, since the Det 1 doesn't have them installed, so his comments may not translated to value validity.

8. Interview with a 609 AOC Det 1 systems engineer revealed that TBMCS will in fact remain in the background, although others have stated it will be removed. His perception was that young, inexperienced code writers, with no military background, are creating applications on the fly, with very little 'adult supervision.' He has observed that they are not working as closely with the operational end users as they should. He brought up the concept of a 'pain point' -- EG; the infamous tanker 'white board.' Pain points are mission areas that are glaringly obvious to require a DevOps solutions. He pointed out that JIGSAW, considered KREL's most famous success, is a success only because the KREL developers worked with seasoned, experienced subject matter experts and 'legacy' personnel. This sentiment has been echoed by others.

9. An interview with the highest ranking government civilian intelligence directorate analyst at USAFCENT occurred, revealing that, 'we may have become so agile that we've outrun oversight.' Not seeming to be a fan, he stated we may need to inject some form of control mechanism. His experience was based on the MARAUDER application, which had 'failed miserably' in the recent past, and CHAINSAW, which, 'has been shelved' -- conflicting opinions on CHAINSAW obvious. He asked, are we even tracking application use? If AOC users are even using the applications, are the apps more effective? He stated All applications are hosted and available via web app access at a central platform operations center

and seemed agitated at ACC. Much like USCENTCOM shoved the half-baked CPN-X system at components, then ordered them to use it, ACC is doing the same with agile. If ACC is going to shove agile at us and say, 'here's your agile,' then it goes against what agile is supposed to stand for. AFCENT needs to have some say in the process.

10. Interview was conducted with a seasoned NTI cell SME. He was passionate to not mandate crowd sourcing and asked, 'where are the *undiscovered* experts?' He felt transparency is lacking and therefore, frustration is building. He spoke will of SLACK collaborative chat tool and the test and evaluation tools. He offered that Air Force Life Cycle Management Center (AFLCMC) is the SPO for DevOps and that end users, 'love the apps,' but that those who don't use the apps seem not to like the apps - interesting.

11. An interview with senior ranking enlisted chief for a critical AOC functional area spoke to leadership discomfort with the process. 'Who is making all of these decisions?' 'Is it ACC/CC?' USAFCENT HHQ, the *warfighter*, is being left entirely out of the process cycle – no transparency, no metrics, no visibility, no oversight and no ability to track what's happening in our own AOC.

12. An AOC chief of training was interviewed as he had just returned from an orientation trip to Boston KREL, so the timing was perfect. The reason he traveled to Boston was because there had been very little transparency. He met with KREL JIGSAW, MARAUDER, DISCO, BLAZON, ARC, RAVESAW team leads, as well as with the KREL lead project manager. He didn't like that no training material was being offered at this time but stated KREL is working to produce it in the future. As good as that sounded, he said they didn't consider it a priority as KREL is now working with other AOCs, not just AFCENT – they're getting busy.

13. An interview with a KREL project manager occurred and he had healthy levels of enthusiasm and positive energy - excitement for making intuitive, user friendly apps that deliver *now*. He was frustrated with constant active duty rotations and bright ideas from end users who are not always experts in their field and who expect immediate results from a brand new process. He offered KREL is operating under the contractual direction that *agile* equates to; no documentation, no metric reporting and very limited leadership interaction.

14. Interview with an action officer level at AFCENT ACOMS A6 communications and systems office occurred. This organization tracks and maintains all AOC systems in the AOR. They are tracking operational status of all new applications, but it's only a red/yellow/green status fidelity – no further end user metric is available. He provided an impressive list of apps that is revolutionizing the AOC but are not necessarily directly related to KREL. ATOMS – ATO management system that replaces TAP and EMR ops floor tools, AXIS – air execution information system – replaces ABIM, IRIS, IAMD Planner – integrated air

missile defense planner, AirSTAT – replace ESTAT and FSTAT FrOB tools, ASMA-ASIS – airspace management/airspace info services de-confliction tool that replaces the tired legacy app WebAD and RISC2 – replaces WARP, RAMP mission trackers and air support request (ASRs). Again, it wasn't clear if these applications were KREL built or not, but they are new, and they are replacing older legacy programs.

5.2 Interview Analysis

The interviews were conducted prior to the surveys, and therefore provided a foundational understanding of the realities of the situation. One of the more important interviews was from a weapons school graduate officer, but many other seasoned subject matter experts, in a wide range of roles were contacted. Software and systems engineers, an operations floor veteran, an intelligence directorate trainer, an experience senior operations duty officer, a command and control squadron commander, AFCENT's highest ranking civilian analyst, an AOC training chief, an intelligence division chief and a KREL project manager. From these interviews, we gathered the following information highlights;

- o No mechanism for status updates to AFCENT leadership or end users.
- o KREL customer service was virtually non-existent.
- o KREL priorities were not in line with the war fighting AOC.
- o No metrics for showing improvements in time savings or personnel.
- o KREL offers no user manuals or training of any kind.
- o Frustration levels are high that end users have no face to face contact with developers.
- o Constant rotation of active duty personnel frustrates KREL developers.
- o KREL is flooded with too many inputs from too many personnel who may not even be experts in their field.
- o Major combat operations (MCO) aspects are definitely not being introduced into the algorithms.
- o Some applications receive both rave reviews and accusations of being 'half-baked.'
- o Successful applications such as JIGSAW are apparently so successful and popular due to the face to face interaction of seasoned experts with developers.

- o ACC has control of the funding and therefore the process, leaving AFCENT higher headquarters, and possibly the AOC itself, with no stakeholder consideration.
- o Tremendous amounts of energy and enthusiasm are present but applications appear to be, in many instances, half-baked.

The predominant take away from the interviews was that there appears to be many opposing opinions. There was no concrete sense that any of the statements represented the full story. It should be noted that none of these interviews spoke directly to an end user from the AOC, but rather, from the CONUS based headquarters and training AOC in the United States. The weapons and tactics officer was an exception, with actual, recent, relevant and robust experience in the AOC, but the study required surveys taken directly from end users at the forward AOC, to attain a more concrete understanding of the problem. It would have been unfair to KREL to draw conclusions without direct results from personnel actively using those applications and directly experiencing the KREL DevOps process on a daily basis, at the AOC.

5.3 Survey Results

The survey was distributed over the classified SIPRnet system, although actual questions were all unclassified. This was required to decrease the risk of recipient responses that may reveal sensitive information. The roles of each respondent was primarily AOC end users. Actual numbers of AOC personnel are classified, but the response sample represented approximately 4% of total AOC personnel. 63% of respondents were AOC end users, 14% AOC leadership and 23% AFCENT higher headquarters leadership. No KREL developers were contacted in the survey but they were represented in the interviews. The Marauder, Jigsaw and Slapshot applications were represented by the majority of users.
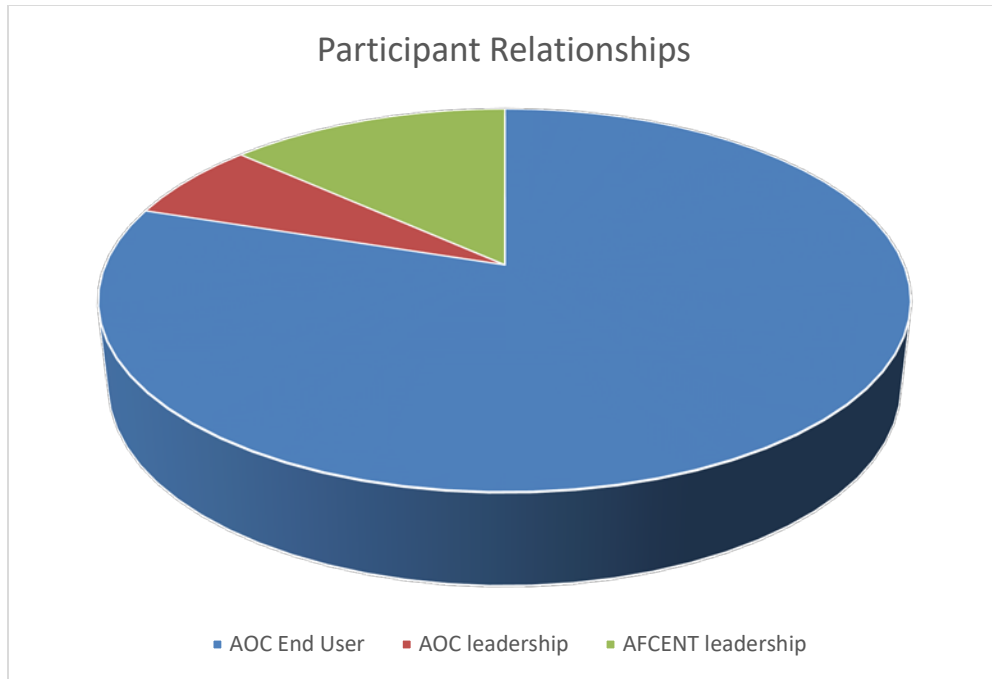
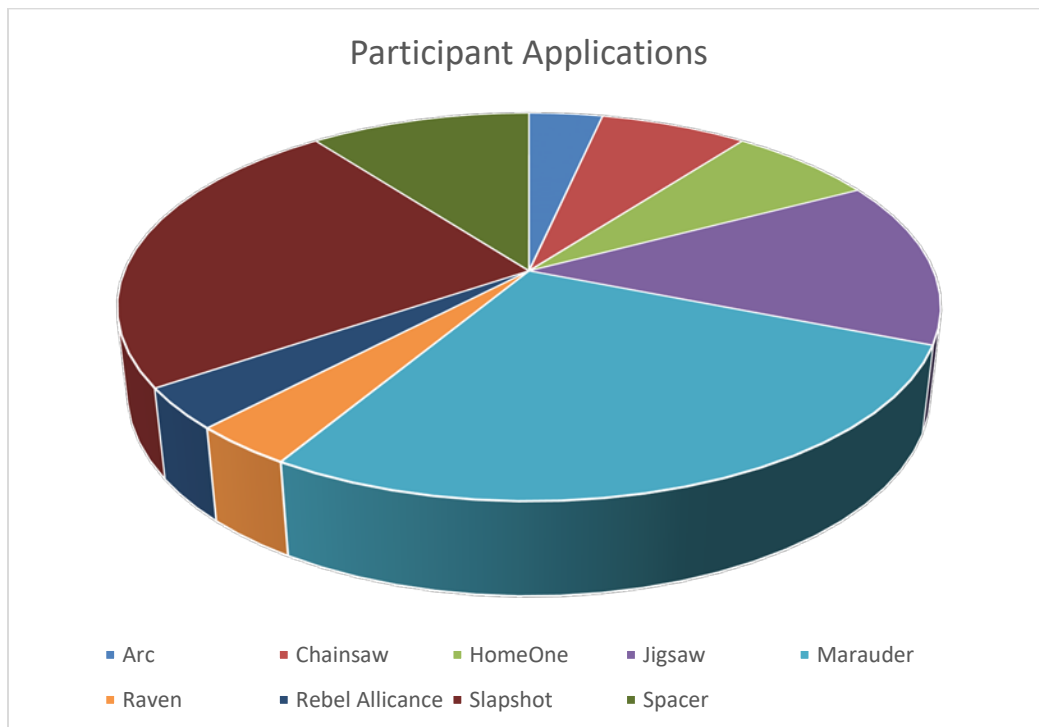Figure 5.1, Question 1. Participant Relationships

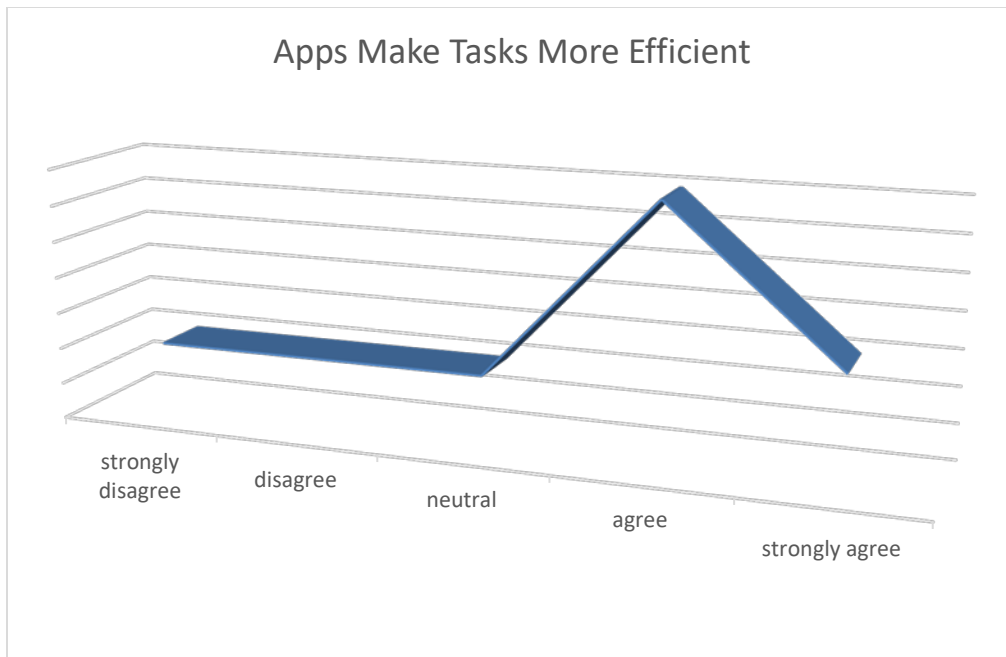Figure 5.2, Question 2. Participant Applications

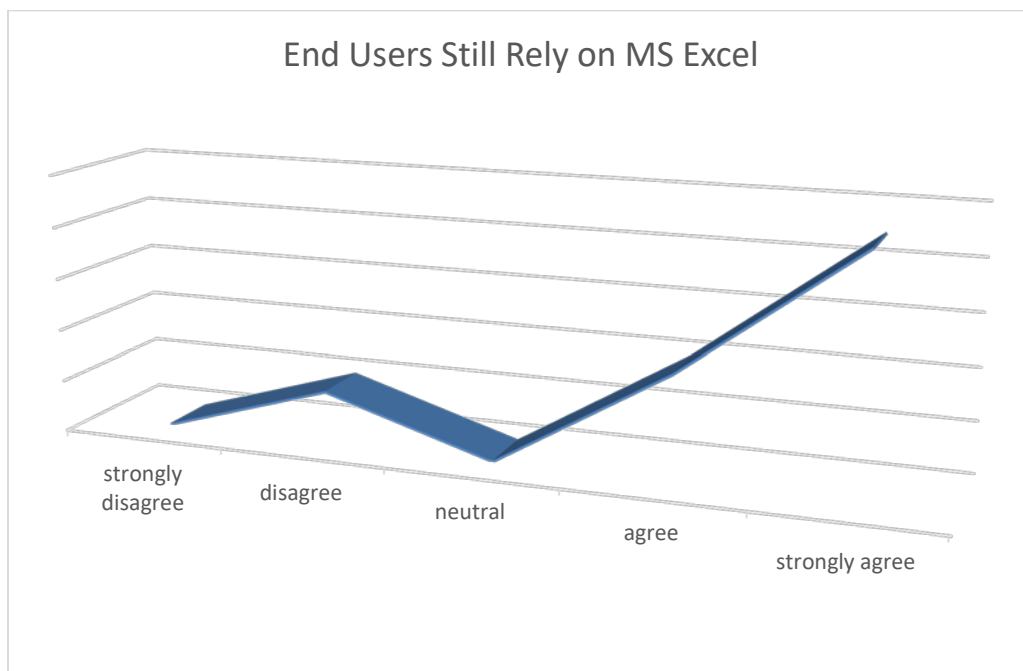Figure 5.3, Question 3. Applications Make Tasks More Efficient

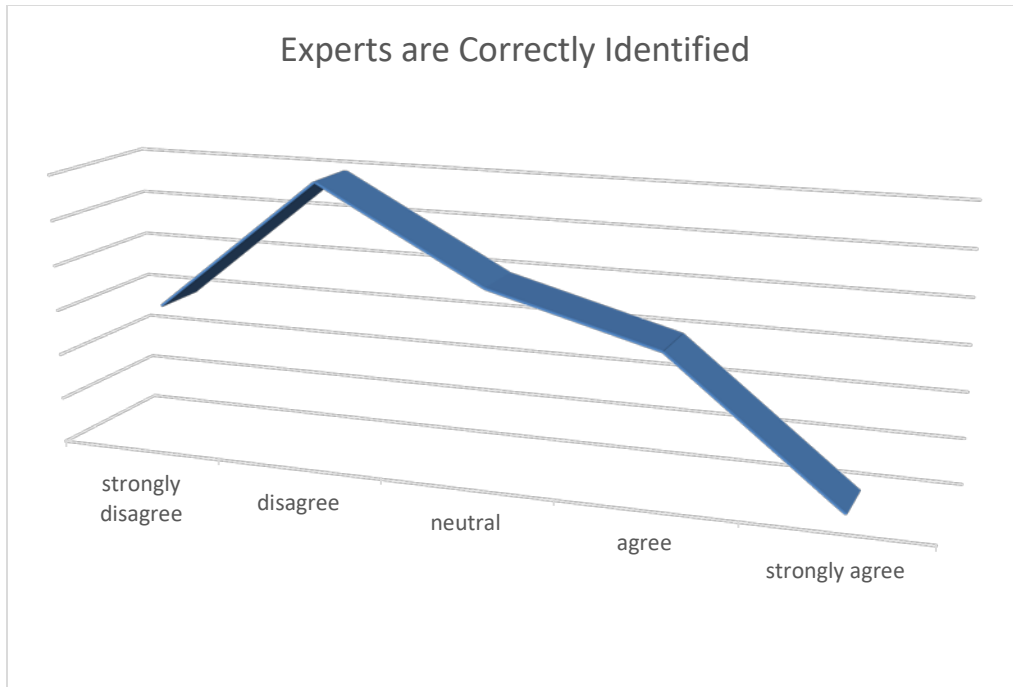Figure 5.4, Question 4. Still Rely on Excel

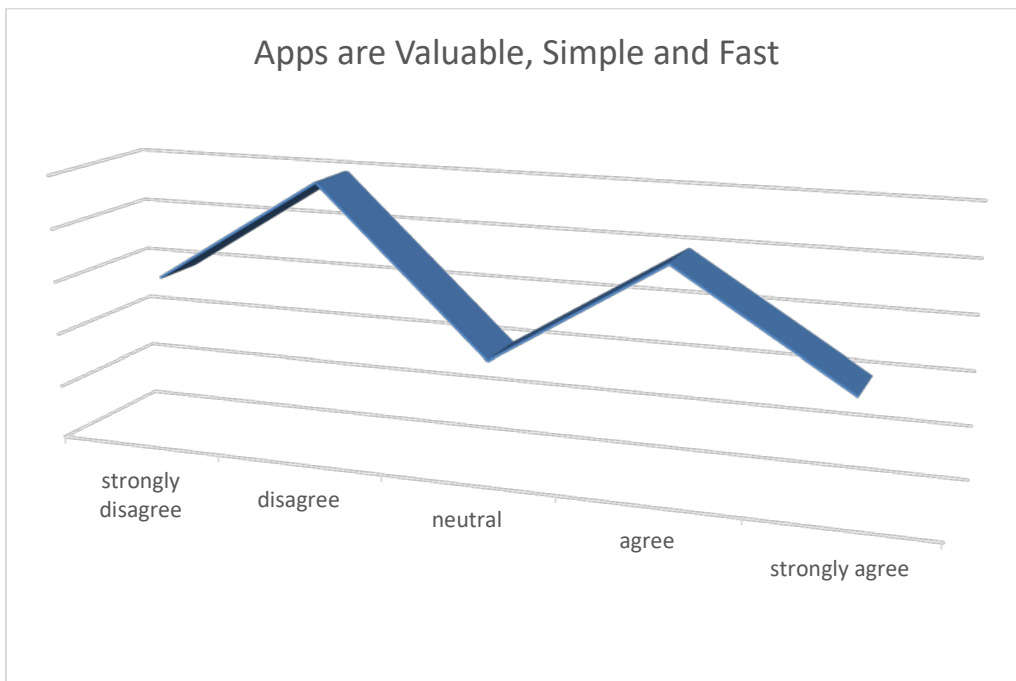Figure 5.5, Question 5. Correctly ID Experts

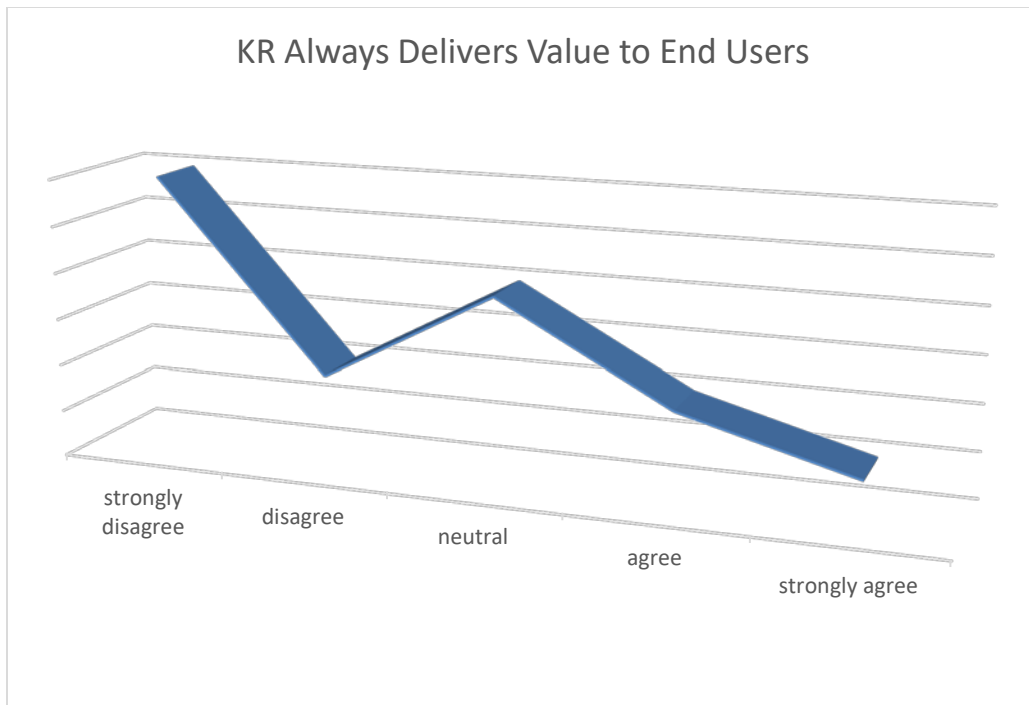Figure 5.6, Question 6. KR Applications are Valuable, Simple and Fast

Figure 5.7, Question 7. KR Always Delivers Value to End Users
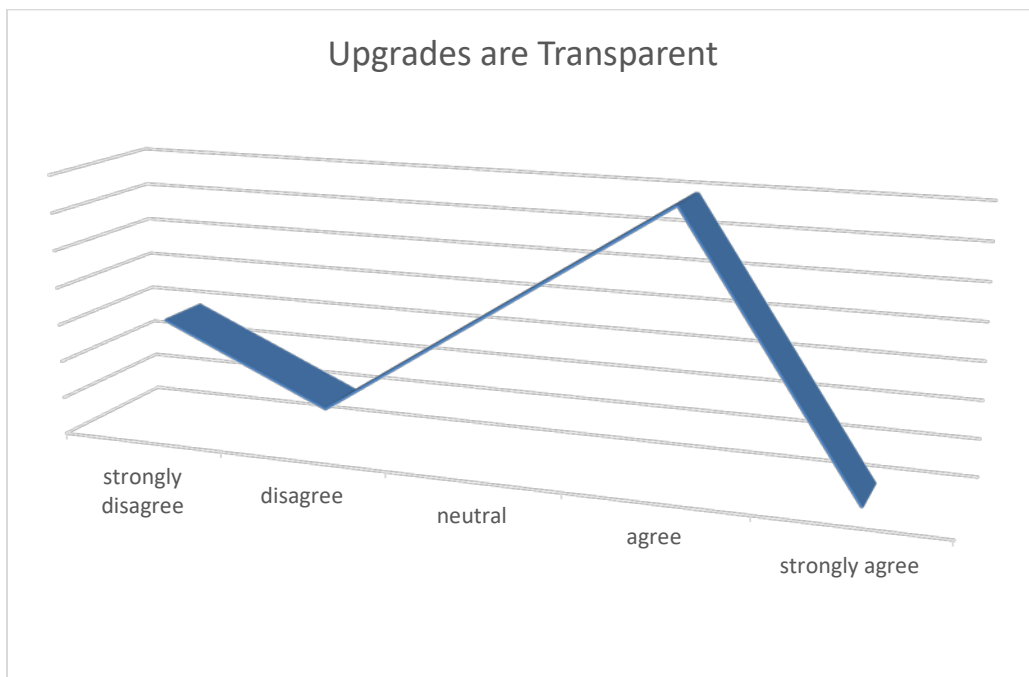

Figure 5.8, Question 8. Upgrades are Transparent

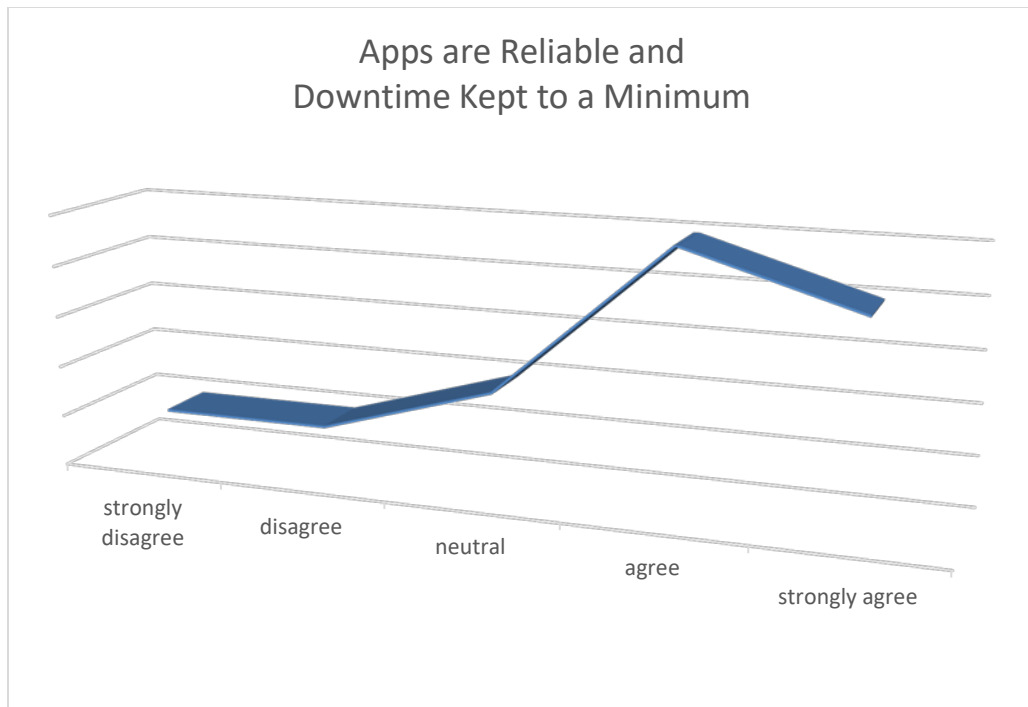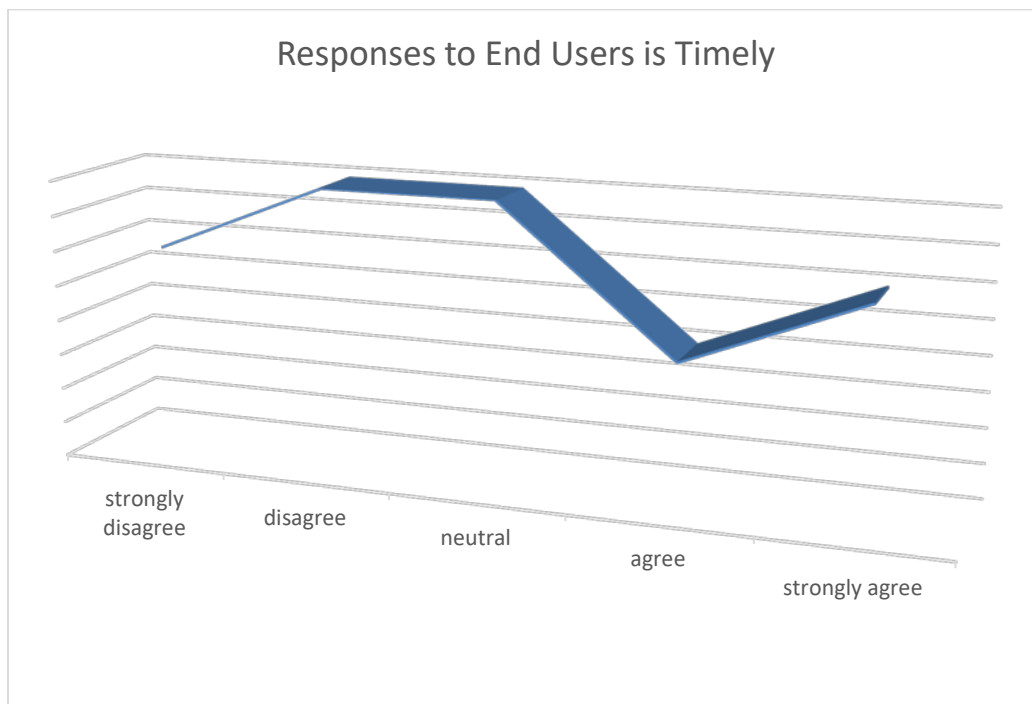Figure 5.9, Question 9. Downtime Kept to a Minimum

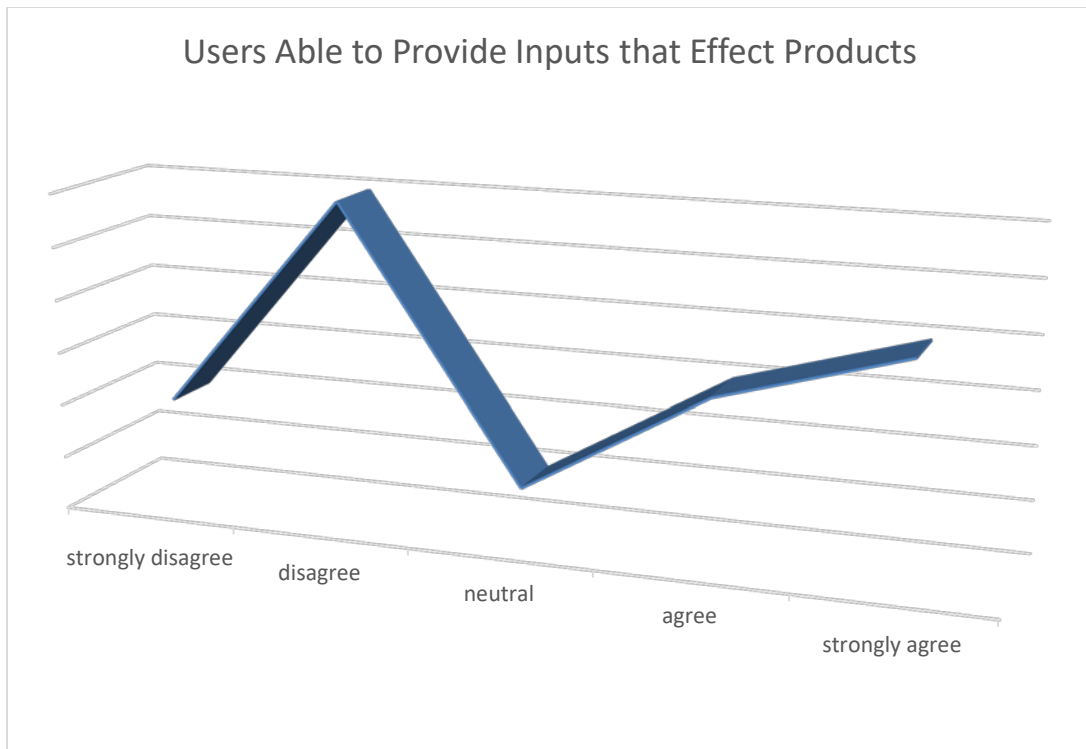Figure 5.10, Question 10. Timely Responses to End Users

Figure 5.11, Question 11. Users Able to Provide Inputs that Effect Products
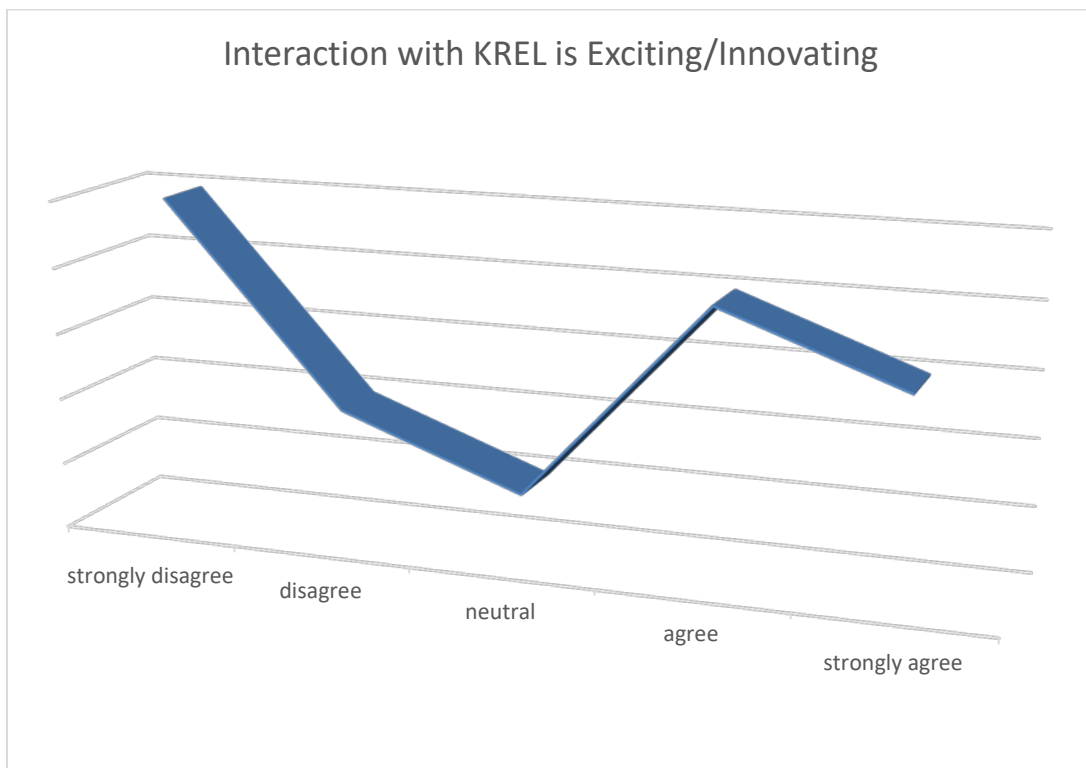
Figure 5.12, Question 12. KREL Interaction is Exciting/Innovating
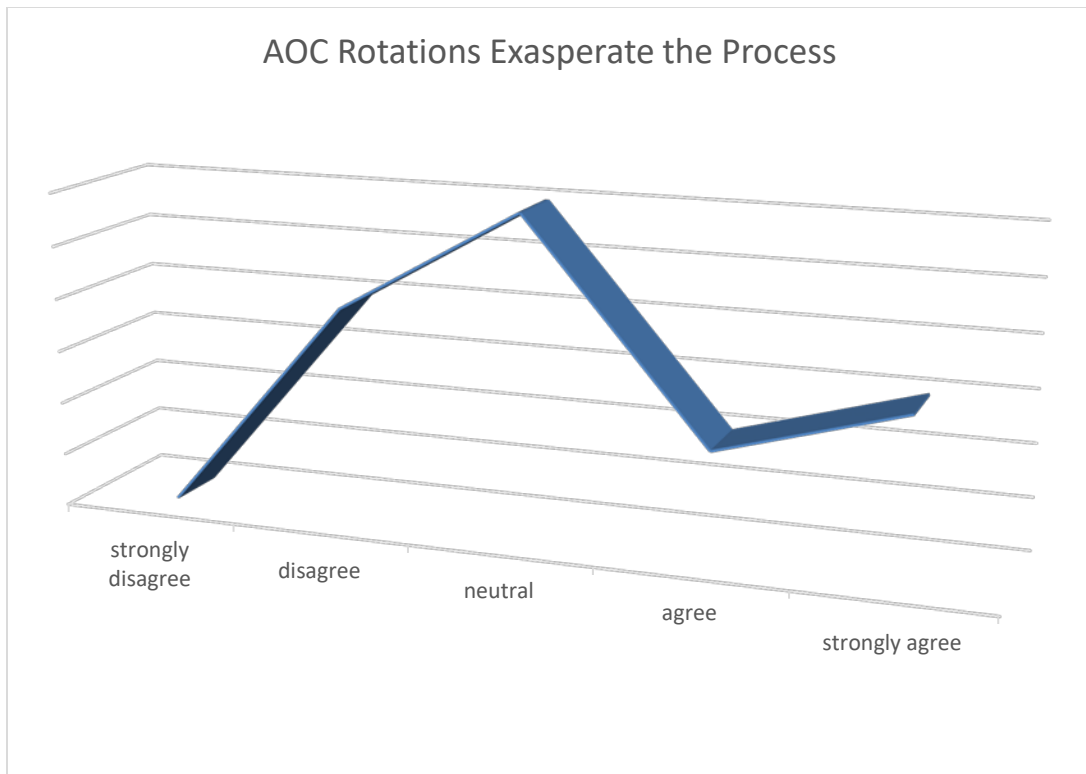
Figure 5.13, Question 13. AOC Rotations Exasperate the Process
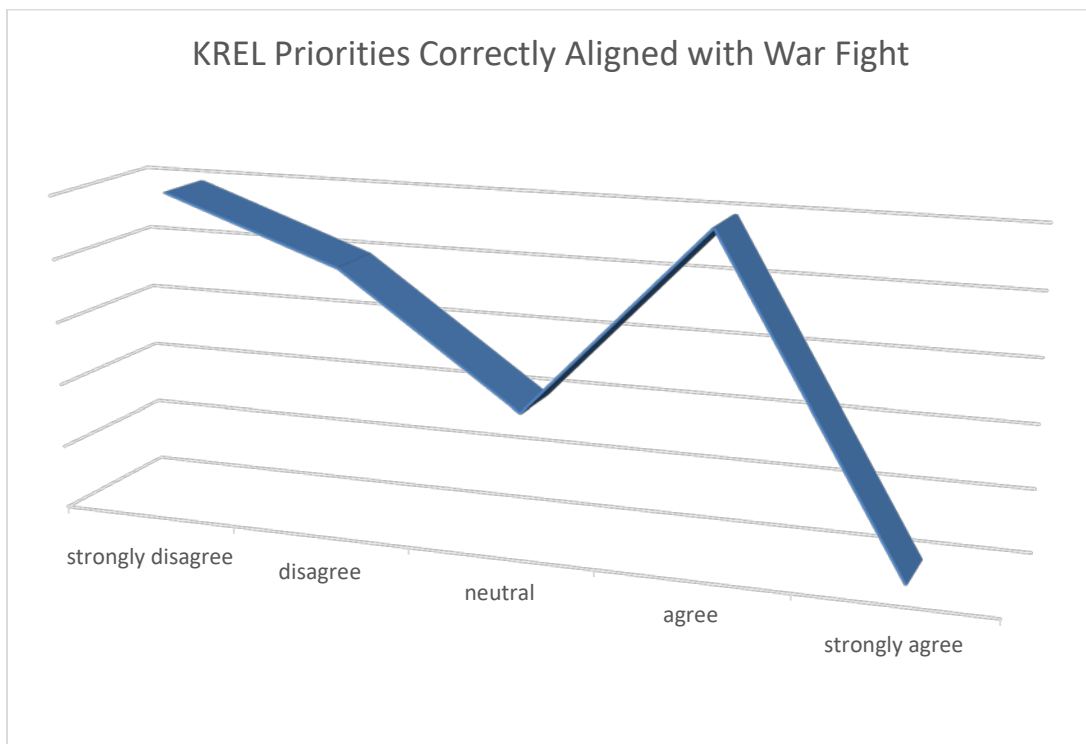
Figure 5.14, Question 14. KREL Priorities Correctly Aligned with War Fight
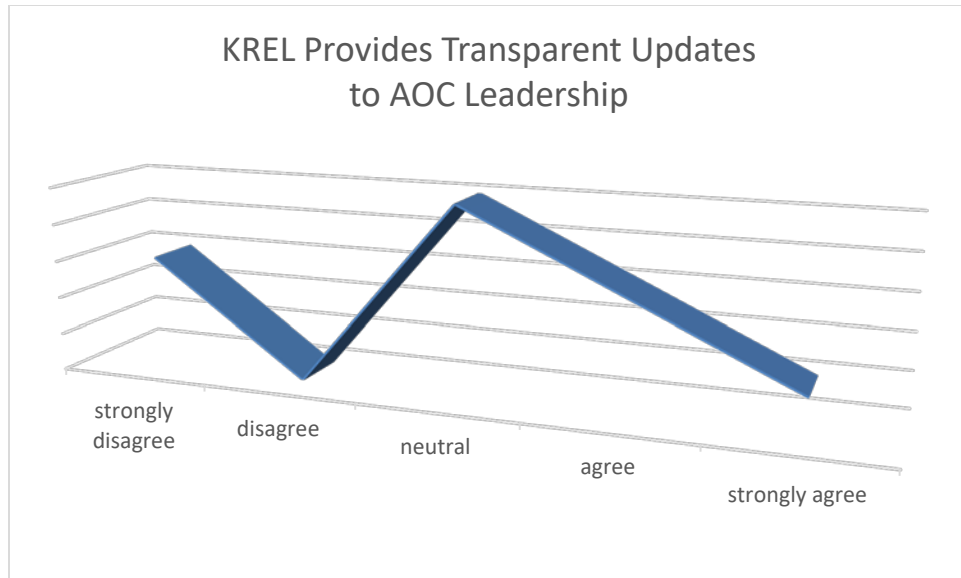
Figure 5.15, Question 15. Process Provides Transparent Updates to AOC Leadership
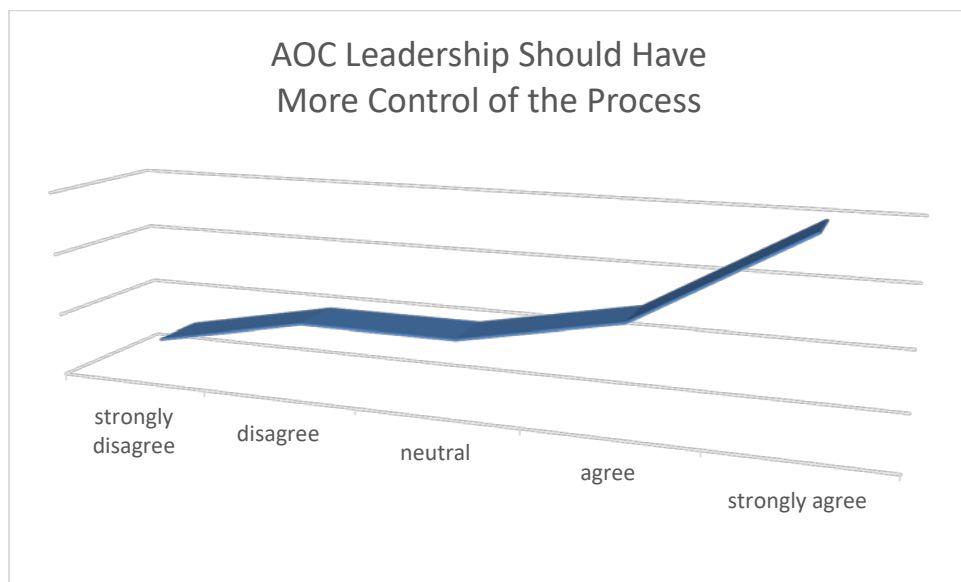
Figure 5.16, Question 16. AOC Leadership Should Have More Control into Process

The last question was open ended, in order to allow for unknowns. Responses were passionate and reflected the tone of what might be expected with the introduction of an entirely new process being introduced, just as the literature reviews predicted. A sampling of the comments are transcribed below, with some paraphrasing accomplished to save space.

o There should be actual KR/MARAUDER rep physically at AOC, that users can go to when issues arise. It's difficult and very time consuming to have to communicate to the KR team over unclassified system through Slack chat. When the system is not working properly, we have to use Slack chat and be very detailed and specific when describing issues which could be better and faster resolved if there was an actual KS/rep here. It seems that the KR team in the states does not have a good understanding of what the needs of AOC users are - especially our constant mission changes. They also don't seem to prioritize updates that users need to accomplish tasks. Having a KR body at AOC would be very beneficial so they would know how an AOC operates.

o For the last 16 months, there has been no reliable way to get data out of Marauder. Both the CSV export and API are broken. There is also no way to view historic data from MAT n Marauder. Both of those are critical to my role and without those features, I can't use Marauder to replace my workflow going forward. DevOps process seems fine, but KR priorities are not aligned with my needs.

o Knowing the whole cycle/problem set would better guide developers and make a better end product. User – centered design is great but with CAOC turnover rates it's hard to finish anything. KR needs to be more agile in development and make smaller updates. App API's need to be written in a way that others apps can connect and share data.

o User-centered design is a good concept but currently it's experiencing some problems. Total lack of reporting status, metrics, user manual and such, to the very client that is directly affected by its work, the warfighting HHQ and end user, is not in line with what user-centered design is supposed to be about. It's really good that the old RE cycle is gone, but the new way of doing business needs some changes – maybe just better customer oriented service.

o So many started apps, but nothing is finished. Why don't any KR apps have guest accounts? RAVEN is essentially a glorified share drive to move info from upstairs to downstairs. It does not have any publish capability to existing databases, so that function is still manual. It also does not allow file output by MAAPTK. We have had 2-3 periods over great than 1 week of downtimes with no feedback other than, 'we're working on it.' One of my targeteers was used as a prop during a CSAF visit to KR to give the impression of being tied into the AOR. It was the first time in 6 months that communication was anything other than a phone call. The VTC was set up to coincide with the CSAF visit and the outgoing VTC was pointed out to the CSAF during a walk around. As soon as he moved on, the KR side wrapped up the call.. MARAUDER is nowhere near full functionality and was declared 'mature' and lost most of its dev team, so now even basic updates take much longer.   It still does not ingest the ATO. Nothing is being done with MAT data to populate more than about 2 years' worth of reports. The change of servers to EPOC [edit] cancelled all user accounts and a month-long process to get accounts was not fruitful. Search functionality is limited – not good. When KREL reps visited CAOC, the O-6 didn't understand our feedback on how half-baked apps are not helping our efforts. Perception: senior leaders think KR is doing great things, but it is not the case for users.

o MARAUDER team is [edited]. [edited]. You should have a classified phone where people can seed assistance or provide feedback instead of having us do it for you at CAOC when it's not our job. Additionally, the team has no clearance. This is ridiculous. You're working with secret data but have no clearance. [approximately 1 page of detailed clearance account issues edited] MARAUDER team is [edited] because they refuse to take responsibility for their task and communicate with users. This process will be easy to fix if we could simply have you here, at the AOC. Doing this remote does not work. It also does not work if the KR team does not really actually care. [name and number deleted] [classified mission details and MARAUDER specific data source issues deleted] So it doesn't free up the analysts to actually think, analyze and provide assessments or update reporting tools on how it displays data back to consumers. MARAUDER needs lots of work.

o Kessel Run responds quickly to irrelevant questions and provides little to no useful responses to items that should be actioned upon quickly. They take lots of feedback, and listen to none of it. They are the least useful DevOps program I've ever worked with.

o Some clear problems with prioritization, communication and development keeps it from living up to being as good as it could be. Many discussions with other users that have similar responses. When we provide feedback, it's shut down, even when constructive and professional. KR mission definitely worthwhile but lacking proper execution and transparency. Frustration is due to lack of progress, appears like nothing is being done and no one is heard. Developers clearly don't understand how the 609th AOC operates, what we do or how their products meet our needs.

o New app rollouts are well publicized and create a lot of headaches from the data extraction standpoint. Very little care is given to how we get the info from the app. The lack of insight into if/when issues will be fixed or if they are even on a story board is frustrating. Pushing system breaking updates like the Marauder GeoAxis sign-in swap is not satisfactory ever, especially right before going home for the day in the States and forcing the theater to find a work-around. Finally, when program managers tell AOC users that are forced to use these apps while fighting multiple active campaigns that they aren't the Dev priority, there appears to be a large disconnect on what the DevOps priorities should actually be.

5.4 Survey Analysis

The survey represents AOC end users, AOC leadership and AFCENT leadership only, with no KREL or ACC respondents. The majority of applications represented were Marauder, Slapshot and Jigsaw. Question 3 asked if applications make tasks more efficient and the distribution shows a majority opinion of agreement. However, just as

seen in the interviews, it appears that it depends on who you ask. Many either love the applications or think they're half baked. The interview data also revealed some apps are better than others. Ultimately however, it does appear KREL applications have value in creating more efficiency.

Question 4 asked if users are still engaged with MS Excel products. The distribution was clear – a resounding response that Excel is still used by most all players in their daily routines.

Question 5 asked if end user experts are being correctly identified. This aspect was identified as being critical for the JIGSAW application success. The results mixed, but most felt the process had *not* identified the best subject matter experts. This functional duty would not fall in KREL's responsibility lanes however. Such a function would fall on the customer.

Question 6 asked if the applications had value, as defined by being simple and fast to use. A mixed response, with strong opinions on either side was attained, so the answer appears to depend on the application. Question 7 was similar, and asked if generic 'value' was delivered to end users. Again, we see a mixed response that would indicate it depends on the application. However, there were many strong disagreements to value being delivered. Question 8 seemed to confirm that upgrades are in fact transparent and question 9 indicated that downtime and reliability are solid.

Timely response to end users, question 10, showed again, mixed results, with many strong opinions that it KREL was not delivering timely responses. This is more evidence to suggest all applications are not the same, with some users getting frustrated and some experiencing good response times. Question 11 also shows a mixed

distribution, for users being able to provide inputs that positively affect products, so we continued to see strong opinions, seemingly based on application specific.

Question 12 asked if interaction with the KREL teams was exciting and innovation inspiring. There were many strong disagreements, but, again, we observe mixed response distribution curves. Question 13 asked if the active duty rotations of personnel are causing frustrations for the process. This question was designed to be fair to KREL. The majority of opinions was neutral, but we do see a strong response of agreement that rotations do cause a problem for the process.

Question 14 asked if KREL priorities are correctly aligned with the war fighting customer and although we do observe some mixed results, there is strong evidence to suggest KREL priorities are *not* properly aligned. Finally, question 16 asked if KREL is providing transparent updates to leadership and since most users are end users, we observe many neutral responses, with strong disagreement, likely from leadership, that it does not. That said, there is also injects that KREL does. This likely translates that the AOC is in fact receiving updates, but that it's not distributing it to other divisions – a common problem in the AOC vs. higher headquarters organizations. Should AOC leadership have more control over the process was the last Likert scale question. The response was again mixed.

So, were the primary questions answered? Are AFCENT higher headquarters' leadership concerns valid? It would appear some of them are, but it's application dependent. Does leadership need to inject more controls? This doesn't appear to be the issue and it might be dangerous to run with a conclusion to inject controls before having a conversation with ACC and KREL. Too much control could damage the agile concept.

46

It's a moot point regardless, since AFCENT doesn't own any part of this process, at least for now.

Do KREL code writers and project managers have too much authority? It appears they may, but again, this doesn't appear to be the issue, since we have evidence to suggest it's an application specific problem. Are development improvement metrics being measured? It appears they are being created, but it's not being disseminated to all stakeholders. Are end user AOC operators being serviced with timely responses to requests? Answer was mixed and it's application dependent apparently. Where does the process need to be adjusted, if at all? It appears from one [very sharply worded] open-ended survey response that KREL could solve all of these problems with one easy solution: acquire classified SIPRnet phones in KREL shops and initiate travel to, or establish residence at the forward AOC. Few enjoy working in the middle-east environment, but if KREL is going to make real progress towards resolving issues, they'll have to follow tenet #1 of the Agile Manifesto*; individuals and interactions over processes and tools*. This equates to face to face DevOps, not chat tool to chat tool DevOps.

Possibly the most glaring negative analytic was the responses to question 6 and 7, which asked the level of perceived value in KREL applications and processes. The DORA Research and Investment firm emphasized that software quality is notoriously difficult to measure as it's highly subjective, highly contextual. The only way to accurately measure the software process, is by measuring a perception of *value,* from its *end users* (Puppet + DORA, 2016). Question 6 and 7 responses were overwhelmingly

negative. Does this translate to KRE applications having low value? Not enough data to determine, but it does raise a red flag.

Finally, is the role of the (ACC) force provider being misused? Again, there's no data from this study, but it would appear that the role of who is the customer is not clear. Hopefully, a conversation can occur between ACC and the combat AOC and an agreement can occur that will help alleviated some of the concern. As for the primary hypothesis, it would appear the question of being *too agile* is application dependent.

5.5 Threats to Validity

The predominant threat to the validity of this study was that neither Air Combat Command (ACC), nor the KREL developers' interests, were represented in the interviews or surveys, with the exception of one interview with a KREL program team lead. However, the questions were designed to field responses to represent KREL's process interests as best could be understood. One could also argue that KREL's and ACC's responses are not required for this study, since the task was to internally determine the validity of the questions asked by AFCENT leadership.

A second threat to validity could be the total number of survey results. If the answers are application specific, we'd need to survey each application end user cadre and KREL teams to fully identify the problem areas.

CHAPTER 6

RECOMMENDATIONS

As the literature reviews predicted, there are challenges being experienced within the unique military environment as it relates to the agile DevOps revolution. Like any other problem set, there are ways stakeholders and vendors can improve. As the agile manifesto guides, conversations need to continually occur between stakeholders, and there's evidence to suggest that's not happening. It's a brand new process injected into a culture that was known to have challenges incorporating it and no one entity is to blame for any current shortcomings.

USAFCENT higher headquarters and their associated AOC leadership should understand that active duty rotations and overwhelming injects of change requests is likely a challenge for KREL to manage. KREL should understand physical, face-to-face collaboration is required for quality DevOps and their lack of presence in the AOC is not in line with the spirit and intent of agile concepts. Second, it should be the war fighting AOC's job to ensure qualified, *experienced* subject matter experts, who understand the larger AOC picture, to participate in the process.

Possibly, the hire of a seasoned program manager to interface, from the AOC, with KREL and ACC, to ensure big picture MCO continuity and capability, accurate crowd sourcing, metrics distribution and status reporting would mitigate some of the problems. Such an positional role would ideally require a multi-faceted AOC C2 resume;

combat plans and operations, MAAP cell, intelligence, airspace, air refueling, major combat operations experience and heavy AOC systems and project management and/or engineering management skills.

As for any eagerness on USAFCENT leadership to inject process controls, it should be remembered that the RE five-year nightmare cycle that was just removed was in large part due to too much control. AFCENT leadership also needs to initiate conversations with ACC and KREL. Of course, the priority is always the war fight, but this project is affecting the current war fight and shaping the future war fight. Tenet #1 of the agile manifesto has strong, proven value*; individuals and interactions over processes and tools* - conversations between these stake holders appear to be long overdue.

KREL should understand many end users in this combat AOC are still using MS Excel, feels some KREL applications are not valuable, simple or fast, feel KREL doesn't deliver value to end users and that KREL priorities are not aligned with the customers. KREL also should work harder to provide updates, metrics, user manuals and training. They need to improve timely response and seriously discuss a road ahead for funding travel and permanent party to the AFCENT AOC. Developers should at least be working over the phones and acquiring SIPR VOSIP lines in Boston will be required to resolve. If this isn't possible, then the facility in Boston is likely not the correct location for such a mission. Ultimately, the agile concept dictates interaction with end users in *their* environment, not on a cold chat tool.

ACC, the force provider, needs to have a conversation with USAFCENT, to better understand who primary customer and stakeholders are, and adjust the ownership to streamline results. The agile concepts were written to address processes assuming clear lines of customer/client – contract developer/vendor roles, but the reality of the DoD military organizational structure is often a confused mish-mash of solid and dotted command organizational lines. The good news is, the software development process achievements of KREL to date *have* been revolutionary, and represent a much better process than the former RE cycle model. It simply appears the process is not fully mature and needs some adjustments. Challenges were expected and this study should by no means fault KREL for the imperfections of the insertion of a brand new, revolutionary process.

JIGSAW is an example of strong success, and there appear to be others. One survey respondent provided a glowing assessment of the SPACER (combat airspace) application. Apparently, not every KREL application team produces the same results. Should KREL therefore provide bonus incentives for teams who deliver better products – as measured by end users? This process has the potential to reach an impressive state of AOC C2 combat software excellence, but if we assume its first iteration equates to perfection, this study indicates we'd be fooling ourselves. Lines of communication between stakeholders should open up, emphasizing focus on individuals and interactions over processes and tools. Staying true to the agile spiral tenet, it's built. Now let's measure it, and learn from it.

Synopsis of Recommendations

- o AFCENT and ACC:
  - ➢ Initiate regular stakeholder conversations to include KREL and KREL's parent group, Pivotal Labs
  - ➢ Acknowledge war fighting AOC and associated higher headquarters are customers
  - ➢ Identify experienced subject matter experts (crowd sourcing) to ensure the same quality of experienced end users that resulted in the highly successful JIGSAW application
  - ➢ Place a uniquely qualified program manager to track the AFCENT AOC KREL process, providing status updates to leadership and managing stakeholder equities and outcomes
  - ➢ Communicate clearly to KREL that chat tools represent an ineffective substitute to the face to face communications required for successful agile DevOps software development
  - ➢ Discuss funding for permanent party KREL AOC developers to include bonus incentives for applications that end users designate as sucessful
  - ➢ Augment or replace the primary chat tool concept with robust SIPRnet VOSIP and VTC capability at Boston facility
  - ➢ Assist KREL in mitigating the frustrating reality of continual rotational personnel and their associated overload of change request injections from end users who may not be experienced enough to provide valuable inputs

- o KREL:
  - ➢ Acknowledge the perception of a lack of value, lack of timely response and inadequate customer support by AOC end users
  - ➢ Provide financial incentives to development teams who produce the better products, as measured by end users
  - ➢ Be prepared to step away from chat tools and replace with the direct, developer-to-end user, interaction that successful agile DevOps software development demands
  - ➢ Prioritize delivery of user manuals
- o AFCENT, ACC and KREL:
  - ➢ Retain a focus that this project scenario represents one of the very first iterations of the inclusion of a process that carried with it significant cautions that unique challenges *would* occur in the military environment
  - ➢ Retain focus that individuals and interactions should always take precedence over process and tools

REFERENCES

OSD. 2018. *Air Operations Center - Weapon System (AOC-WS) Overview*, FY13-FY18. 2013-2018. Department of Operational Test and Evaluation, Office of the Secretary of Defense.

Boehm, B., Turner, R., *Balancing Agility and Discipline – A Guide for the Perplexed*, Addison-Wesley, 2004.

Broadus, W., *The Challenges of Being Agile in DoD*, January-February, 2013. Defense AT&L.

Brooks, F., *No Silver Bullet – Essence and Accidents of Software Engineering*, IEEE Computer 20(4):10-19, Apr 1987.

Chaplain, C.T., *DoD Delivering New Generations of Satellites, but Space System Acquisition Challenges Remain*, Government Accountability Office Report GAO-11-590T, May 2011.

Collens, Josiah R. 2004. *TBMCS Systems Engineering Case Study*. The MITRE Corporation.

DoD Instruction 5000.02. *Operation of the Defense Acquisition System*, 2007-2019.

Guarro, S.B. and Tosney, W.F. (editors), *Mission Assurance Guide, The Aerospace Corporation Technical Report* TOR-2007(8546)-6018, 1 Jul 2007.

Guarro, S.B. and Hecht, M., *Risk and Reliability Assessment of Software-Intensive Systems*, Space Systems Engineering and Risk Management Symposium, Los Angeles, CA, 26 Feb 2008

Hantos. P., *Agile Software Development in Defense Acquisition – A Mission Assurance Perspective*, October 22, 2012. NDIA 15th Annual Systems Engineering Conference, San Diego, CA.

Insinna, Valerie. 2017. *Air Force Cancels Air Operations Center 10.2 Contract, Starts New Pathfinder Effort*. Defense News. Retrieved from, https://www.defensenews.com/air/2017/07/13/air-force-cancels-air-operations-center-10-2-contract-starts-new-pathfinder-effort, 2019.

Kenner, B. 2019. AOC and ISR Enterprise Processes.

Lapham, M. 2012. *DoD Agile Adoption: Necessary Considerations, Concerns and Changes*, January/February, 2012. CrossTalk.

Lapham, M. 2014. *Agile Mythbusting*. Software Engineering Institute, Carnege Mellon Univ.

Maucione, S. 2018. *Software is Air Force Acquisition's Biggest Problem*, Federal News Network, Apr 30, 2018.

Mergel, I., *Agile Innovation Management in Government: A Research Agenda*, Government Information Quarterly, 33, 2016. *Defense AT&L.*

Newel, Benjamin. 2018. *Kessel Run Hits Hyperdrive*, Hanscom Air Force Base News, 66th Air Base Group Public Affairs. Retrieved from, https://www.af.mil/News/Article-Display/Article/1518722/kessel-run-hits-hyperdrive/, 2019.

Ott B., Pham J., Saker H. 2017. *Enterprise DevOps Playbook – A Guide to Delivering at Velocity*. Booze Allen Hamilton and O'Reilly Media, First Ed., Nov 2016.

Paparone, C.R. 2009. *From Not-So-Great to Worse – The Myth of Best Practice Methodologies*, Defense AT&L, Jul-Aug 2009.

Puppet Inc. + DORA Research and Investments. 2016. *The 2016 State of DevOps Report*.

Riungu-Kalliosaari, L. 2016. *DevOps Adoption Benefits and Challenges in Practice: A Case Study.* Springer International Publishing, University of Helsinki.

Royce, W. 1998. *Software Project Management – A Unified Framework*, Addison-Wesley.

SECAF. 2008. *Weapon Systems Software Management Guidebook*, Version 1 (Abridged), 15 August 2008. Secretary of the Air Force for Acquisition (SAF/IAQ).

Schaeffer, C. 2019. *Agile Versus Waterfall for CRM Implementation Success*. CRM Search. Retrieved from, http://www.crmsearch.com/agile-versus-waterfall-crm.php, June 5, 2019.

Turner, Daniel. 2010. *Qualitative Interview Design: A Practical Guide for Novice Investigators*, TQR, The Qualitative Report Vol. 15, No. 3, Nova Southeastern University, May 2010.

Wallace, M. 2018. *The U.S. Air Force Learned to Code – And Saved the Pentagon Millions.* The News Articles, July 5, 2018. Retrieved from,

https://thenewsarticles.com/the-u-s-air-force-learned-to-code-and-saved-the-pentagon-millions, June, 2019.

Welby, S., 2013. *Thinking About Agile in the DoD*, November 21, 2013. AFEI Agile for Government Summit, Deputy Assistant Secretary of Defense for Systems Engineering.

APPENDIX A

QUALITATIVE INTERVIEW AND SURVEY QUESTIONS

The following list of questions was used as an outline for the qualitative

interviews and as specific questions in the innovation survey. Due to limited time

available during qualitative interviews, the questions were not specifically recorded, but

instead were sent out in advance for individuals to review, to facilitate a more structured

conversational feedback approach.

1. What is your relationship to AOC DevOps?

       AOC 'end user'

       AOC leadership

       Kessel Run software developer

       Kessel Run program management

       AFCENT leadership

       A6 or ACOMS

       ACC

       Other

2. My specific DevOps software application/s are;

       Papercut, Ghost, Ripsaw, Slapshot, Blazon, Jigsaw, Spacer, Mai Tai (FrOB),

       Triton, Arc, Yeti, Chainsaw, Marauder, Skyhook, Pythagoras, Rebel Alliance,

       HomeOne

3. My AOC DevOps software application/s is/are reliable and 'downtime' is kept to a minimum.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

4. The AOC DevOps software development process has provided applications that make my processes and tasks more efficient and effective.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

5. I still rely on Microsoft Excel to manage my associated war fighting processes and tasks.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

6. The AOC DevOps software development process is providing transparent, relevant updates to AOC active duty leadership.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

7. AOC active duty leadership should have the capability to inject more direction into the AOC DevOps software development process.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

8. The AOC DevOps software development process provides timely responses to end users.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

9. Kessel Run priorities are correctly aligned with AOC warfighting priorities.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

10. AOC rotations exasperate the agile software development process by creating an environment with too many inputs for changing coming from too many end users.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

11. I feel that the AOC DevOps software development process is correctly identifying the full range of required war fighting experts to build resilient and flexible applications.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

12. Once operating within a Kessel Run application, I experience upgrades that happen behind the scenes and do not require me to retrain or troubleshoot.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

13. Feedback loops between users and development teams allow me to provide inputs to the product decisions.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

14. I believe the Kessel Run applications represent the next most valuable thing, the simplest way, as fast as possible.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

15. Kessel Run user-centered design approach ensures they always deliver value to users.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]

16. Interaction with Kessel run developmental teams is exciting and allows user testing to spark ideas for additional innovation.

[Strongly disagree, Disagree, Neutral, Agree, Strongly Agree, N/A]