Improving Enterprise Architecture at West Safety Services— a 911 Emergency Telecom Company

ICT 4010 Enterprise Architecture

Brenden Hyde

University of Denver University College

November 8, 2017

Instructor: Dr. Steve Else, Ph D.

Table of Contents

Background 5

Organizational Structure of West.. 5

Architecture Vision 6

Identifying Current Architecture Issues Faced by West's GSD 7

Analysis of issues faced by the VIPER Help Desk 9

Issue 1A – Engineering Strategy.. 9

Business Case 1A.. 11

Issue 1B – Sales Strategy.. 11

Business Case 1B.. 11

Issue 2 – People 12

Business Case 2 13

Issue 3 – Integration.. 14

Business Case 3 15

Issue 4 – Information Systems 16

Business Case 4 17

Issue 5 – Knowledge and Information 17

Business Case 5 18

Implications of EA Solutions 18

West GSD Core Diagram 19

Introducing TOGAF.. 21

Recommended EA Solutions 23

Proposed Solution 1 23

Proposed Solution 2 24

Proposed Solution 3 25

Proposed Solution 4 27

Proposed Solution 5 30

Roadmap 32

Conclusion 34

References 36

West Safety Services is a 911 emergency IT company that specializes in digital 911 systems. They have police stations and fire departments across the United States and elsewhere as their customers. Their main product is a 911 system called the VIPER, and the group of technicians that provide service for it are collectively called the VIPER Help Desk.

The VIPER Help Desk has several deep-seeded problems with its Enterprise Architecture related to its strategy, personnel hiring, data integration, information systems, and knowledge base. The implications of these shortcomings are diminished customer satisfaction, longer-lasting issues in customer systems, over-allocation of personnel budget, insecure and overpriced information systems, and infighting within the larger organization.

This document identifies and analyzes each of the five problems listed above and offers potential solutions to these issues. The solutions are based on research conducted by Enterprise Architecture professionals who have amassed repositories of relevant generic architecture artifacts along with industry-specific suggestions. This document uses those EA artifacts to propose a robust solution set to the problems named hereafter.

The proposed solutions include agile system development, more precise definitions of job duties, a Discover Component for systems outside of the VIPER, Free and Open Source Software solutions as building blocks for the VIPER, following industry consortium best practices, and Mashup Applications.

Background

West Safety Services, formerly Intrado, inc, is a division of The West Corporation that provides Enhanced, digital 911 emergency services to Public Safety Answering Points (hereafter PSAPs) around the world. This includes fire stations, police stations, and emergency medical facilities. To give an idea of their market share, as of 2015, the entire 911 infrastructures of both Hawaii and Vermont were provisioned and maintained entirely by West Safety Services. Although they have no formal mission statement, their homepage offers the following sentiment *vis-à-vis* their corporate vision:

> Everything that happens after a 9-1-1 call is placed–the call routing and delivery, location determination, data management and situational data–that's us. It's essential that we execute flawlessly because every call must get through, and seconds save lives. For us, it's personal. The committed, passionate professionals who make up the West Safety Services team spend every day anticipating, developing, maintaining and improving 9-1-1 and emergency communications. While we appreciate the evolving ideas and technology innovations that make our lives more convenient, we focus on the ones that make us all safer. (West 2017)

Thematically, their statement indicates that they value quick, error-free execution of 911-related tasks; and they should– a dropped 911 call can mean an untimely end for a person in dire need of assistance.

Organizational Structure of West

West Safety Services has several top-level organizational units within it, namely: Wireless, Wireline, Government Services Division, GIS (Geographic Information Systems), TechOps (Technical Operations), and the NOC (Network Operating Center). In addition to those organizations, there are also field technicians employed by West that are deployed into the field; each technician is assigned to one or a few PSAPs. This author has personal experience working within the Government Services Division, and the viewpoints herein will largely be from the GSD perspective.

Architecture Vision

While this section does not attempt to diagnose issues in West's current Enterprise Architecture nor prescribe solutions, it is paramount that the Enterprise Architect understand the qualities of the Baseline Architecture in order to better visualize what the Target Architecture will look like.

In terms of maturity, the Baseline Architecture most closely represents what Ross, Weill, and Robertson (2006, loc 1101) refer to as "Business Silos". They state that Business Silos are the lowest level of maturity out of the 4 stages of architecture maturity. "In the Business Silos stage, companies focus their IT investments on delivering solutions for their local business problems and opportunities. These companies may take advantage of opportunities for shared infrastructure services like a data center, [but] [c]ompanies in this stage do not rely on an established set of technology standards" (Ross, Weill, and Robertson 2006, loc 1101).

The Operating Model of West Safety Services is more difficult to qualify. An Operating Model, as defined by Ross, Weill and Robertson (2006, loc 498) is "the necessary level of business integration and standardization for delivering goods and services to customers". Integration is the sharing of data and information among business units, and standardization is the level to which a process is executed in the same way for different customers by different business units or workers (Ross, Weill, and Robertson 2006, loc 525). West's operating model most closely resembles Diversification, a model characterized by low levels of both integration and standardization. The standardization is low because of the diverse service offerings they have tailor-made for different PSAPs based on budget, geographic restrictions, and the whim of PSAP directors. The integration is low because of disparate systems that were reactively created out of immediate necessity.

An ideal Enterprise Architecture at West would demonstrate a higher level of architecture maturity as well as an operating model befitting of such a large company with a market share as commanding as theirs currently is. The new West would have an Optimized Core Architecture, one in which, "companies move from a local view of data and applications to an enterprise view. IT staff eliminate data redundancy by extracting transaction data from individual applications and making it accessible to all appropriate processes" (Ross, Weill, and Robertson 2006, loc 1129-1157).

Because PSAPs are highly diverse and vary by state and county, the most sensible Operating Model for West to strive for is Coordination. The Coordination Operating Model has a high level of integration and a low level of standardization.

Identifying Current Architecture Issues Faced by West's GSD

As stated prior, this author is most familiar with the Government Services Division organizational Unit of West Safety Services. Therefore, this section is an overview of the West's EA obstacles from the perspective of the GSD.

1. **Strategy**

   West GSD’s sales team is focused on outmoded waterfall methodologies for developing the system, and they are using outmoded pay-to-play sales tactics to sell bulk, shrink-wrapped software bundles.

2. **People**

   By referring to the VIPER support team as a “Help Desk”, the management team ends up recruiting IT generalists who do not have the requisite skills to adequately support the systems.

3. **Integration**

   The infrastructural systems used by the GSD are extremely disconnected from other organizational units, despite tight coupling with them; they do very little in the way of information sharing.

4. **Information Systems**

   The VIPER architecture has a heavy dependency upon proprietary, unsupported legacy software and connection standards that make the VIPER systems hard to interface with and also potentially insecure.

5. **Knowledge and Information**

   A culture of folk wisdom combined with a lack of formal, in-depth documentation leads to a fragmentation of knowledge and information.

Analysis of issues faced by the VIPER Help Desk

Issue 1A – Engineering Strategy

In order to understand West's support strategy, more background is needed regarding their main product. The flagship product of West's GSD is a digital 911 system called the "VIPER", an acronym for Voice over IP for Emergency Response. Their VIPER fact sheet describes it as "the proven, i3 compliant, voice-over-IP (VoIP) solution purposely built from the ground up to provide PSAPs with dependable 9-1-1 call handling" (West 2016, p. 1). Essentially, the back-end of the VIPER is a system composed of several, fully redundant servers and a supporting network infrastructure in the data center of a PSAP. In the call-taking area of the PSAP where emergency responders work, each responder works on a "position", which is a Windows-based workstation with VIPER software for taking and transferring 911 calls.

The VIPER system software is developed by the VIPER engineering team; they are also the top tier of the VIPER Help Desk. The support structure is broken into tiers of support, as outlined in table 1.

Table 1. The VIPER Support Structure of West Safety Services GSD

Support Tier	Worker Qualifications	Specialization
L0	Field technician with general IT and telephony knowledge	Day-to-day PSAP operations
L1	Remote Help Desk worker with VIPER-specific training and general IT knowledge	First level of escalation when field technicians need assistance

L2	2-3 years of VIPER Help Desk work experience and troubleshooting	Second level of escalation
L3	3-5 years of VIPER Help Desk work experience and troubleshooting	Third level of escalation
L4	Software Engineers who originally wrote and tested the software. Read and write access to source code.	Fourth and Final level of escalation

Source: Personal work experience at West Safety Services GSD, formerly Intrado GSD.

The first pressing issue within the VIPER Help Desk is EA strategy. The EA strategy of the engineering team is characterized by too much focus on the long term. For example, the VIPER software only receives quarterly updates, each of which is fairly meager. The software is delivered in the form of “KBs”, which means “knowledge base[s]”. Each KB addresses a specific issue and has an accompanying description. Every one to two years, a “service pack” comprised of all the recent KBs is released. The software development lifecycle followed by the VIPER engineering team is essentially a 1990s-style waterfall strategy. Bente, Bombosch, and Langade categorize this type of grand vision as “A Deep Look into the Crystal Ball” and state that:

> [M]any EA initiatives tend to inherit the bureaucratic behaviors rooted in typical large enterprises. Therefore, today’s EA, to a large extent, sticks to the work culture of the 1990s and shows many of the traits of a waterfall software development life cycle. This means long review and approval cycles for strategy programs, favoring a strategy that looks forward many years (2012, p. 22).

Business Case 1A

This macroscopic view of strategy prevents the organizational unit from realizing short-term wins that help to move the EA towards its Target Architecture. Kotter (2012, ch. 8) demonstrates through a series of case studies that generating short-term wins is positively correlated with business transformation. Since these short-term wins may help the GSD justify its endeavors to upper management by achieving more efficient operations and therefore increased revenue, it is pivotal that the EA strategy be modified to increase efficiency.

Issue 1B – Sales Strategy

Another outmoded strategic practice employed by West is their software sales strategy. Many companies in the modern era have switched to a Software as a Service model where software has a recurring, periodic cost. Support of that software is usually an added, billable premium. West, on the other hand, sells the VIPER software at a very high, one-time cost. The VIPER then operates as-is for an indefinite period of time. Once the engineering team has written enough updates to bundle them into a service pack, West sometimes withholds those upgrades from a PSAP until they sign a new contract agreeing to purchase them along with extended support services.

Business Case 1B

Selling VIPER software as a single bundle that the customer pays for only once causes the GSD to miss out on recurring revenue, and withholding patches makes customer infrastructures unsafe.

Issue 2 – People

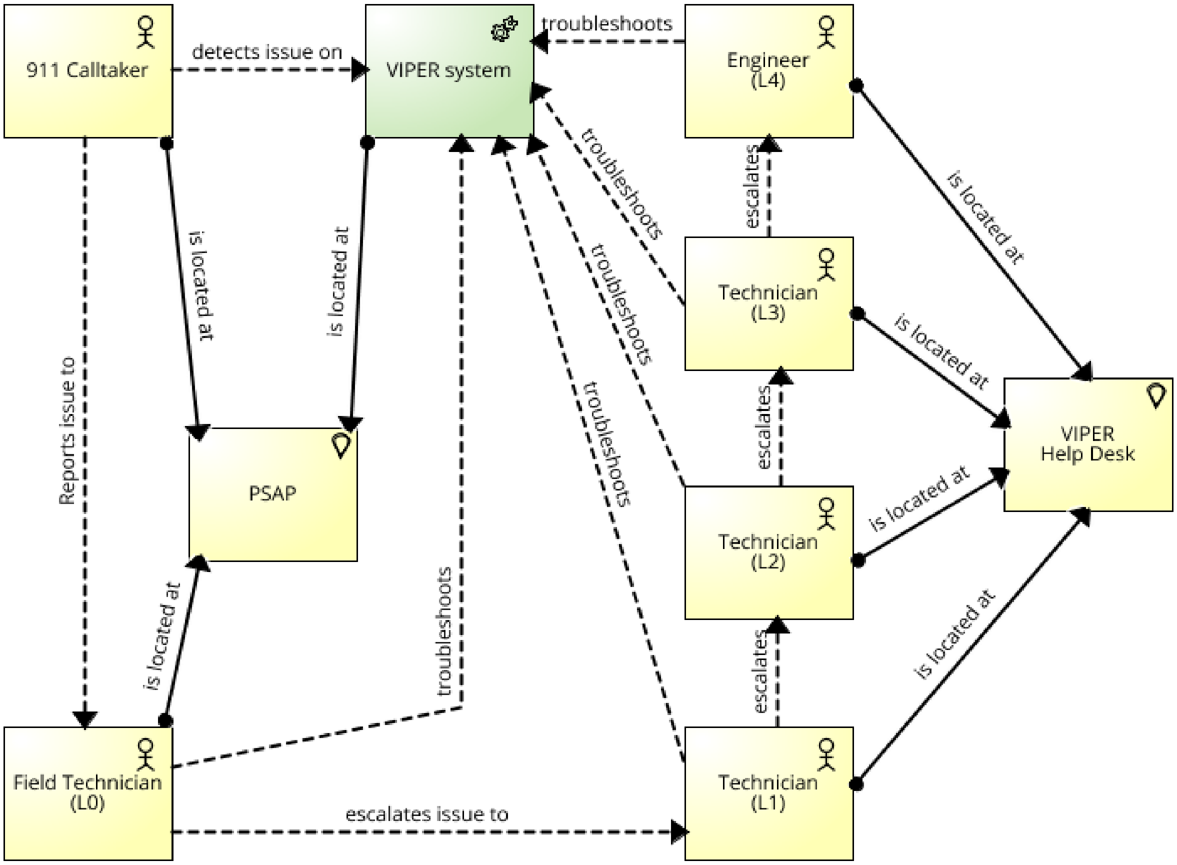
The second, pervasive issue with the GSD’s VIPER help desk has to do with people, more specifically personnel hiring. The VIPER help desk is structured as a hierarchy, from level 0 technicians to level 4 technicians, where higher-numbered tiers are the more skilled workers. The hierarchical model of the Help Desk is suitable, but is designed poorly due to the conceptualization of the L1 – L3 support roles. The job postings for the VIPER help desk seek out employees who have worked in a help desk before. Help desks are usually very basic support roles with low technical knowledge requirements. The VIPER is a proprietary hardware and software system, and the troubleshooting process of it involves testing complex systems and reading through diagnostic log files generated by that proprietary software. While this task is easy for the L4 software engineers who wrote the program, chose the hardware, and understand their inner workings, the L1 – L3 technicians are not trained in software development, cannot read code, and have most of their experience in system administration.

As a result, the overwhelming majority of trouble tickets are resolved only after being escalated to L2, L3 and even L4 technicians. Even then, L2 and L3 regularly consult with the software engineers who have very little time to act as Help Desk support while simultaneously engineering new software and hardware. Inefficiency is especially prevalent for L1 technicians who effectively serve as baseline information gatherers, copying log files and running a few rudimentary tests before handing off the task.

Business Case 2

The wrong types of candidate are being hired on as L1 technicians. Since L1 technicians are so ineffective at closing trouble tickets on their own, management is forced to hire a great number of them in to address the always-growing number of trouble tickets. Figure 1 diagrams the basic ticket flow and hierarchy of the VIPER help desk.

Figure 1. VIPER Help Desk Process Flow



Source: Brenden Hyde based on personal work experience at West Safety Services GSD, formerly Intrado GSD.

Issue 3 – Integration

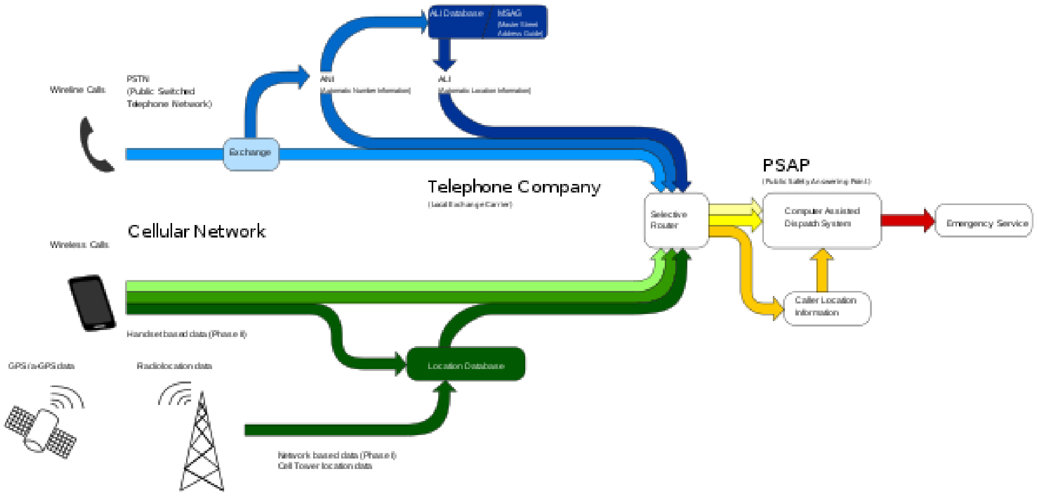
The infrastructural systems used by the GSD are extremely disconnected from other organizational units, as they do very little in the way of information sharing. As a result there are vast horizontal knowledge gaps between the GSD and other departments at West such as TechOps, Wireline, and the NOC. As with issues 1 and 2, further background is required on daily Help Desk operations to elucidate the issue at hand.

In terms of horizontal knowledge gaps, the VIPER help desk only has a view of the internal network (intranet) at a PSAP. That is to say, the technician can see all traffic between different VIPER computers and call-taker computers, but once 911 network traffic leaves the police station either on the Public Switched Telephone Network (PSTN) or on dedicated Wide Area Network (private Internet) lines provided by West, the VIPER help desk is effectively blind to what is happening. If the latter case is true, where 911 traffic goes over West's private, external network, the only department that can see the state of the network is the Network Operations Center. If the traffic goes over the PSTN, the VIPER Help Desk has a dependency on a Local Exchange Carrier like AT&T or Verizon to tell them what is going on.

In either case, the systems are highly dependent upon outside sources for their information, and yet they have very little access to the information that the outsiders control. There is regular in-fighting between the NOC and the VIPER Help Desk, who end up playing the "blame game" as a result of being blind to each other's data. Similar conflicts happen between the Help Desk and departments at the Local Exchange Carriers. Figure 2 shows the flow of an

Enhanced 911 call, i.e. the type of 911 call that West's customers receive. From left to right, the GSD is blind to 911 traffic up until the point where the arrows are yellow.

Figure 2: Enhanced 911 Call Flow


ALI Database
MSAG
(Master Street Address Guide)
Wireline Calls
PSTN
(Public Switched Telephone Network)
ANI
(Automatic Number Information)
ALI
(Automatic Location Information)
Exchange
Telephone Company
(Local Exchange Carrier)
PSAP
(Public Safety Answering Point)
Selective Router
Computer Assisted Dispatch System
Emergency Service
Cellular Network
Wireless Calls
Caller Location Information
Handset based data (Phase II)
GPS/a-GPS data
Radiolocation data
Location Database
Network based data (Phase I)
Cell Tower location data

Source: Evan Mason of Wikimedia Commons. Created for Wikipedia.

https://en.wikipedia.org/wiki/Enhanced_9-1-1

Business Case 3

This lack of integrated data which is essential to resolving customer issues prolongs the time period for which the issues exist. These prolonged system issues drain the reservoir of good will that PSAP directors have, i.e. they lower customer satisfaction and, potentially, decrease the number of contract renewals. Another consequence of prolonged system issues is that it leaves PSAP workers with suboptimal capabilities to handle 911 calls.

Issue 4 – Information Systems

In order to outline the obstacles presented by VIPER information systems, a quick outline of VIPER components is in order. The VIPER is made up of back-room servers that sit in a data center at the PSAP, as well as call-taker "positions" (workstations for answering 911 calls) in the call-taker area. The back-room servers are connected to the call-taker positions over a local network. The back-room servers consist of Voice over IP servers that allow calls to be answered, put on hold, and transferred internally. There are also Plain Old Telephony Servers (POTS for short) that know how to talk to the Public Switched Telephone Network. Lastly, there are value-added servers which are optional, such as Database servers that store data that usually has to be requested from outside.

Most of these systems depend on proprietary, pay-to-play operating systems such as Windows 7, Windows XP, Windows Server 2008, and Windows-embedded Linux. The only use of Free and Open Source software is found on the VoIP servers, which run a Linux operating system program called Asterisk. At best, the VIPER's dependency upon Windows is a monetary liability, as licensing has to be purchased for each copy. At worst, it is a security vulnerability, as Windows has officially stopped supporting Windows XP as of April, 2014 (Microsoft 2014), and it has stopped supporting Windows Server 2008 as of 2011 - 2013 (Microsoft 2017). This end-of-life announcement by Microsoft includes ending support for critical security patches that leave systems at risk.

Business Case 4

The heavy reliance upon licensed, unsupported software places a liability on all of West Safety Services because of the accompanying information security risks.

Issue 5 – Knowledge and Information

There is no single view of reality for VIPER Help Desk employees, VIPER engineers, or Help Desk managers. If data about the VoIP servers is required for a trouble ticket, it must be collected by manually connecting to a VoIP server, copying and pasting the data to a file, and delivering the file to the intended recipient. Similarly, log files on the POTS servers and call-taker positions must be tended to with the same clumsy and manual intervention. This is true for both hardware performance and software performance.

Folk wisdom plays a big role in the VIPER troubleshooting process, as each system varies and documentation is not well-maintained. There are four different systems for displaying urgent information to Help Desk employees. There are also two wiki websites, dubbed “the old wiki” and “the new wiki” that have different information but which serve redundant functions. Finally, there are Methods of Procedure, also known as “MOPs”; these MOPs are step-by-step documents for setting up a VIPER for the first time, but there are very few MOPs designed for troubleshooting a system which is already live and in production. However, technicians are told to use these MOPs as troubleshooting guides.

Business Case 5

This lack of documentation puts an undue burden on VIPER technicians to either memorize troubleshooting steps, ask others who have done it, or keep personal notes which they often fail to share with others. Fragmented knowledge undermines the already low level of standardization in the VIPER help desk, as technicians are likely to inconsistently handle issues that arise.

Implications of EA Solutions

In the preceding section, a number of Enterprise Architecture shortcomings were presented that indicate West Safety Services' GSD is in need of an EA overhaul. Each one of the highlighted shortcomings is much more than a formalized complaint from an ex-employee; rather, these issues are detrimental to daily business operations.

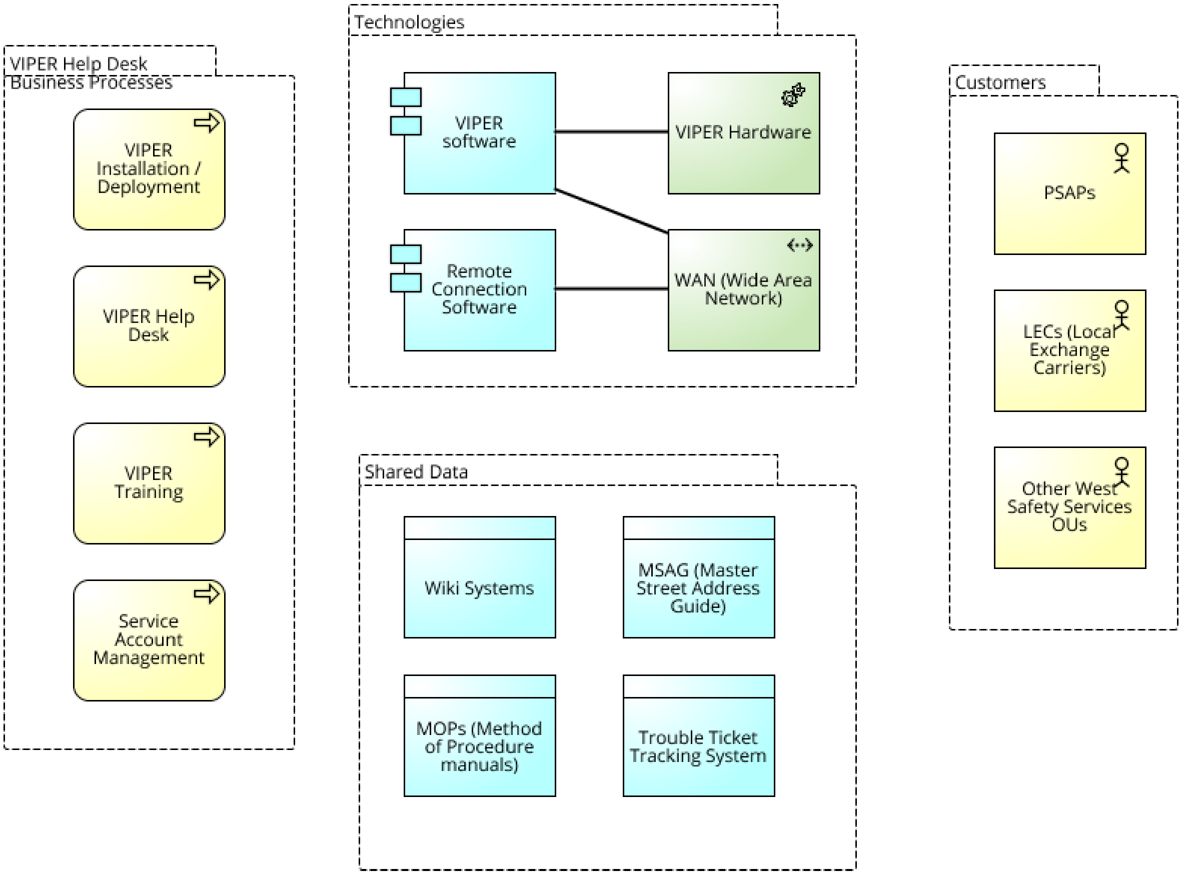
They present unnecessary complexities that stifle the GSD's Foundation For Execution. A Foundation For Execution, according to Ross, Weill and Robertson (2006, loc 223), "is the IT infrastructure and digitized business processes automating a company's core capabilities". These obstacles also have real financial implications, as a mixture of wasted man hours and high turnover rates slow the growth of the organizational unit and simultaneously lower customer satisfaction levels.

Therefore, this author proposes that the Enterprise Architecture be analyzed in its current form to create a Core Diagram. The Core Diagram will outline the core processes that the VIPER help desk is meant to execute. Those processes may not be fully agreed upon or understood by all the stakeholders, so the Core Diagram will also serve as a basis for discussion amongst stakeholders to see if it matches their view of reality. From there, portions of the The Open Group Architecture Framework, also known as TOGAF, will be utilized to plan the restructuring of core business and IT processes into a collaborative web of reusable functionality.

West GSD Core Diagram

According to Ross, Weill, and Robertson, a Core Diagram is a one-page overview of a company's core business processes, key linking and automation technologies, key customers, and shared data that drives the core processes (2006, loc 848). They also state that Core Diagrams vary depending on a business' Operating Model. The Architecture Vision earlier in this document states a desire to move toward a Coordination model. Figure 3shows the core business processes, data, systems, and customers.

Figure 3. VIPER Help Desk Core Diagram


VIPER Help Desk Business Processes
VIPER Installation / Deployment
VIPER Help Desk
VIPER Training
Service Account Management
Technologies
VIPER software
VIPER Hardware
Remote Connection Software
WAN (Wide Area Network)
Shared Data
Wiki Systems
MSAG (Master Street Address Guide)
MOPs (Method of Procedure manuals)
Trouble Ticket Tracking System
Customers
PSAPs
LECs (Local Exchange Carriers)
Other West Safety Services OUs

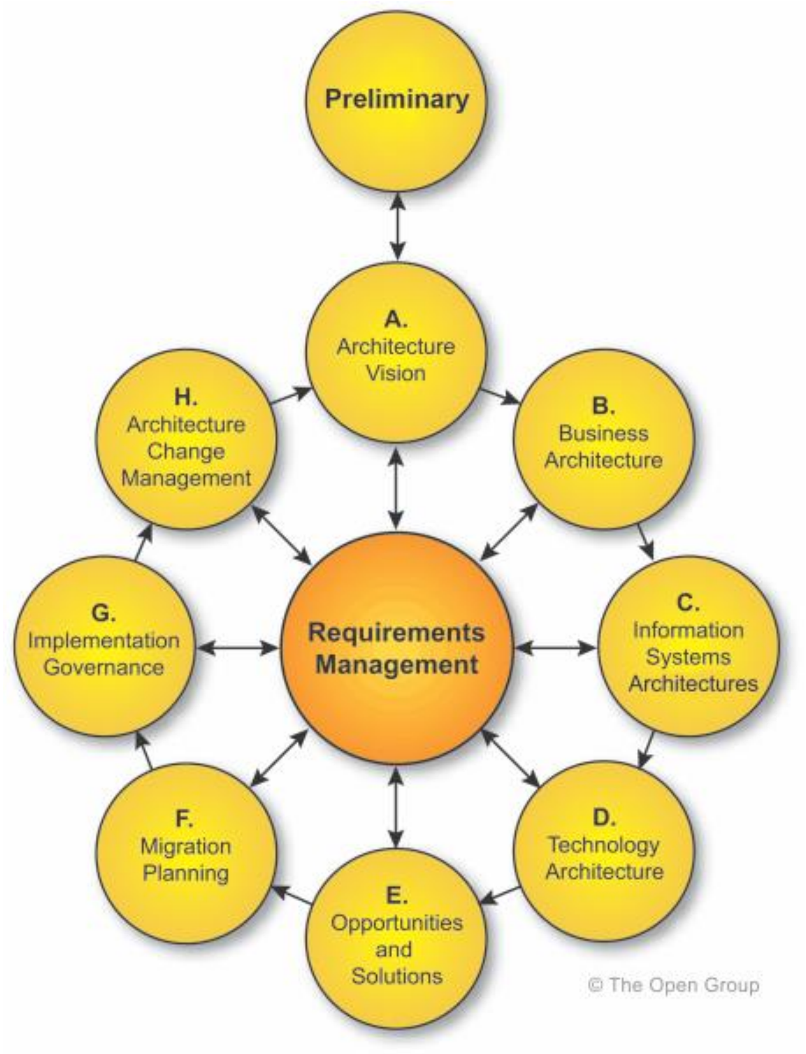
Source: Brenden Hyde based on personal work experience at West Safety Services GSD, formerly Intrado GSD.

Introducing TOGAF

The Open Group Architecture Framework is an Enterprise Architecture Framework. According to TOGAF's documentation (Harrison 2011, p. 13), an Enterprise Architecture in the context of TOGAF is, "A formal description of a system, or a detailed plan of the system at a component level to guide its implementation". The purpose of an Enterprise Architecture is "...to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy" (Harrison 2011, p. 13). This optimization is almost certainly easier said than done, which is why the main process that undergirds TOGAF is the Architecture Development Method, also known as the "ADM".

The TOGAF ADM is an iterative set of steps that act as a guide for meticulously transitioning from a Baseline Architecture to a Target Architecture. Figure 4 is a diagram that shows the TOGAF ADM. It includes multiple phases that each deal with different, important considerations that the Enterprise Architect will need in order to successfully apply the framework and change the EA of the business.

Figure 4. Diagram of the TOGAF ADM


Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
© The Open Group

Source: The Open Group. http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/adm.png

Recommended EA Solutions

Proposed Solution 1

The software release strategies employed by VIPER system engineers (who are also the L4 technicians) span far too much time in between releases. Therefore, an agile software development methodology should be used to develop software on a continual basis.

According to Pathania (2016, ch. 1), waterfall methodologies are increasingly a thing of the past. S/he states that, "[l]arge software systems in the past relied heavily on documented methodologies, such as the waterfall model. Even today, many organizations across the world continue to do so. However, as software engineering continues to evolve, there is a shift in the way software solutions are being developed and the world is going agile". Further support for agile methodologies in both EA and software development can be found in Bente, Bombosch, and Langade, who make the following observations about waterfall vs. agile:

> The waterfall approach requires strict up-front planning to avoid changes in the complex production machinery later on. Lean and agile methodologies have formulated their antithesis: welcoming change, creating and planning incrementally, and focusing on structured human interaction instead of channeled reporting lines. At the same time, lean and agile methods fully subscribe to the factory notion, too. They do not strive to abolish organizational structures and revert to the software-developing "noble savage"4working on her own. Lean and agile methods introduce a reduced and efficient organization to produce large and complex software systems (2012, p. 162).

Because agile methodologies increase efficiency in software development, they are a great replacement candidate for waterfall on the VIPER engineering team.

Proposed Solution 2

On the surface issue 2 is a personnel and human resources problem because the candidates that are being hired onto the VIPER help desk are unsuited for the job as a result of an inadequate, ill-fitting job description. In another sense, this is actually a management issue.

In Erickson's (2009) review of Jim Collins' book *Good to Great*, she discusses his notions of getting the right people on the bus. The bus is an analogy used by Collins to represent an organization, such as a business or a team. Erickson illuminates the exact type of personnel crisis the VIPER help desk is facing when she says the following:

> Jim has found that great companies have clarity around both the definition of key positions — that is, which roles are essential for success — and a focus first and foremost on making sure the right people are in those seats. This is necessary if you're trying to make a good company great — and doubly so if you're trying to excel in turbulent times. Jim emphasizes that this needs to be your highest priority — unless you're surrounded by the right people, little else will matter (Erickson 2009).

The as-is team most definitely does not have a proper definition of key positions, as L1 technicians are improperly categorized. By redefining the role of an L1 technician to "Application Support Specialist" and adding some criteria about code-level application troubleshooting to the job posting, fewer L1 technicians would need to be hired to close the same number of trouble tickets. Eventually, as L1 technicians are promoted from within, their software-centric approach would proliferate throughout the organization. The right people would then be on the bus.

Proposed Solution 3

Issue 3 describes a scenario where a lack of integration is slowing troubleshooting to a halt. Therefore, the level of integration needs to be tightened. According to Godinez et al. (2010, ch. 8) information integration can be completed by first discovering what data is available to the system at hand. Godinez et al. (2010 ch. 8.2) further break this discoverable data into two categories, namely “Technical Metadata” and “Business Metadata”. The technical metadata is information about the source of the data, the protocols needed to access it, and the format it comes in. The business metadata has to do with the flow of data through the business processes, e.g. how the data is allowed to be used in a given business process.

At the VIPER help desk, when a customer calls in with a complaint of static on the lines, the part of the system that needs troubleshooting is sometimes outside of the scope of the VIPER. That is, the “lines” themselves are Wide Area Network lines provided by another division at West– the Network Operations Center. Therefore, the technical metadata that Godinez et al. suggest gathering must come from the WAN lines provided by the NOC.

In order to collect such discovery metadata, Godinez et al. suggest a model called the Enterprise Information Integration (EII) Component Model (2010, ch. 8.1 – 8.2). The process of discovery involves creating a component called the Discover Component that discovers all the data and metadata in use for a specific process:

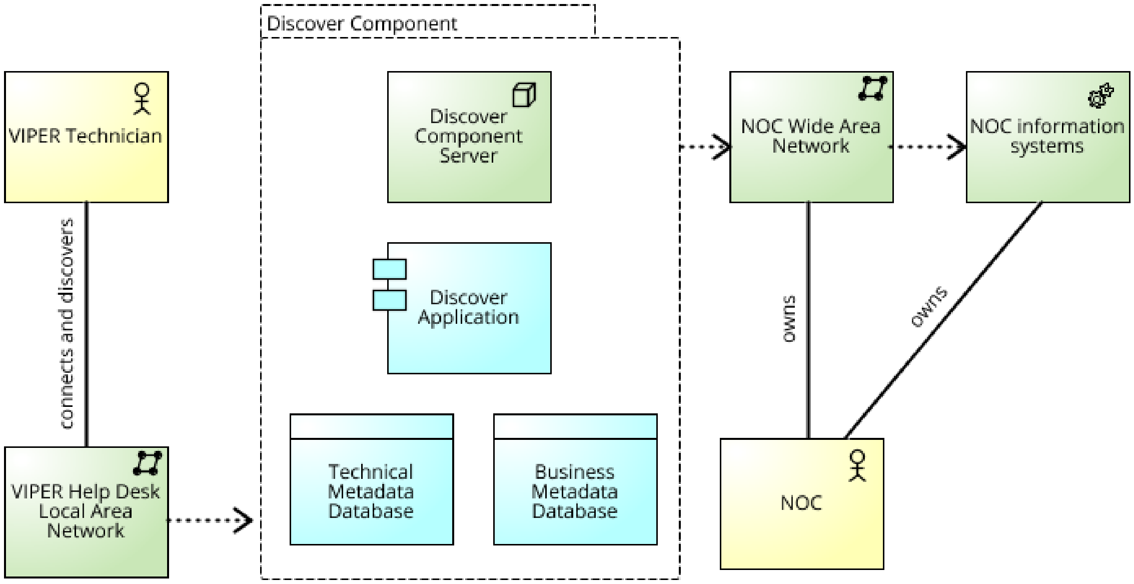
> It is useful for the Discover Component to be invoked on a regular basis as a service that is used to drive a regular auditing of the information systems that are in place. This can be accomplished by developing a set of discovery routines that build up an initial picture of the state of the existing systems and the relationships between them. The results

> from this can be stored in an analytical database for use later, or models can be captured and held in an appropriate modeling tool (Godinez et al., 2010 ch. 8.2).

Since the Wide Area Network is owned by the West NOC, a Discover Component is more viable than it might be if the lines were owned by a third party like a Local Exchange Carrier. The software engineering team could define the data requirements based on the information they currently lack and then develop a Discover Component that sweeps the network for various protocols and devices that help them to gather that data for regular use.

After the Discover Component successfully enumerates the network services in the NOC, it can be reused to regularly populate the VIPER help desk's databases with fresh network information. This effectively gets rid of the business process bottleneck because VIPER technicians no longer need to place a call to the NOC, open a ticket, and wait for investigation and subsequent response. Figure 5 shows a high-level overview of a basic Discover Component for the VIPER team.

Figure 5. Discover Component for West Safety Services Network Operations Center Systems



Source: Brenden Hyde based on personal work experience at West Safety Services GSD, formerly Intrado GSD.

Proposed Solution 4

In order to update its legacy information systems and rid itself of a dependency on licensed Microsoft software, the West GSD can utilize certain portions of the TOGAF framework in combination with a vertical industry set of best practices. Only part of the solution to issue 4 is offered in this section. The remaining portion of it is laid out in the section entitled "Roadmap".

In the as-is state of the VIPER information system architecture, it is heavily reliant upon license-based software, specifically the Windows operating system family. This is inopportune for several reasons. Firstly, two of the Windows operating systems the VIPER uses have been "end-of-lifed" by Microsoft. This means that no future updates for functionality *or security* will be released by Microsoft. This is an enormous risk for system stability reasons and business liability reasons, not to mention ethical ones. Secondly, the recurring licensure paid for by West's information systems budget could be put to a different use if something could supplement West's use of Windows.

That "something" is Free and Open Source Software or "FOSS". Free software is a term coined by GNU founder Richard Stallman, and the word "free" in this context is a word that Stallman says means free as in speech, not as in beer (Free Software Foundation 2017). This means that free software protects the user's freedom to use and modify it as s/he pleases without fear of reprisal from anyone. From a business standpoint, this ensures that customization will not be replete with red tape and restrictions and potential lawsuits for violation of a software license. Operating systems from the Linux family are also free in the monetary sense, which allows for more efficient budget use. Of course the choice of specific FOSS will need to be carefully considered over a period of time, which is where TOGAF can assist decision makers.

Utilizing the TOGAF Architecture Development Methodology, they can ensure there is a development plan, a set of steps to phase out the old systems and integrate the new ones, and a governing body to benchmark and approve the steps being carried out. In particular, TOGAF ADM Phases B, C, D, F, G, and H are very useful to the GSD for this purpose.

In phase B, the business architecture of the VIPER help desk is laid out fully. Specifically, what are the business needs of a 911 company who is re-engineering a digital emergency system? Compliance with government regulations, e.g. the FCC, understanding the stakeholders, and ensuring that the business processes align with the Architecture Vision are key points covered here.

Phases C and D deal with determining what data the new VIPER will deal with and then which suitable information systems can handle that data. Listing all of the specific data types, protocols, development methodologies needed for a VIPER system is outside the scope of this paper and could easily fill a book. However, a bird's eye view of the ideal information system for a company that wants to rid itself of license-based software while improving Security Architecture should be based Free and Open Source Software, such as the Linux operating system.

Once the data and information systems are chosen, ADM Phases F and G come into play. Phase F will allow the GSD to carefully roll out the new VIPER systems while slowly phasing out existing legacy systems. Such a plan ensures minimal downtime for important customer systems. Phase G deals with the governance of implementing the system. After all, without someone with managerial power leading the new information systems initiative, many problems could occur. For one thing, the governance board formed in phase G is in charge of benchmarking whether or not the deployment is going according to plan. Without them, the plan is more likely to fail.

In order to make sure that such a system complies with vertical industry best practices, the Enterprise Architect could also consult with NENA's documentation. NENA is the National Emergency Number Association– an industry consortium that has a vast repository of industry best practices for 911-related companies such as West Safety Services. Their standards webpage shows that they have recommended best practices for the following areas: "Accessibility, Agency Systems, Data Management, Data Structures, Interconnection [and] Security, NG9-1-1 Transition Planning, PSAP operations, Public Education and PSAP Training" (NENA 2017). Consulting with the NENA documents along each relevant phase in the TOGAF ADM will ensure that this solution is compliant, secure, and efficient.

Proposed Solution 5

The as-is state of the VIPER help desk's knowledge base is in relative disarray. As stated prior, their knowledge base consists of 2 wiki pages, several alert systems, a series of deployment documents called MOPs (Methods of Procedure), and unstructured, custom notes by each employee. In order to federate this knowledge into a system which can be viewed through a single pane of glass, the VIPER team should utilize two methods for aggregating that data-- a Mashup application to take their existing, multiple knowledge bases and incorporate them into one, as well as Enterprise 2.0 software to enable employee contributions and critiques.

Godinez et al. (2010, ch. 12.1) have the following to say in regards to Mashup applications:

> Mashups are built on a web-oriented architecture (Representational State Transfer [REST] and Hyper Text Transfer Protocol [HTTP][3]) and leverage lightweight, simple integration techniques (asynchronous JavaScript and XML [AJAX], Really Simple Syndication [RSS], and JavaScript Object Notation [JSON]). The result is the fast creation of rich, desktop-like web applications.

Essentially a Mashup application enables the VIPER team to share both structured and unstructured data in a single place. Extracting data from the technicians' personal notes must be dealt with separately, however.

To incorporate the personal notes taken by VIPER technicians, Enterprise 2.0 applications could be utilized. Bente, Bombosch, and Langade (2012, ch. 8) outline Enterprise 2.0 as a conceptual realm of software that applies techniques used in social media sites and content management systems to enterprise software. For example, Facebook is known for letting its users comment on pictures and blogs written by people in their peer group. Enterprise 2.0 software enables similar commenting, but on business-specific sites like a trouble ticket portal or a wiki.

The implication of using Enterprise 2.0 in the VIPER help desk are promising. As stated prior, technicians often have personal notes that serve as informal knowledge bases. These personal notes are often incorrect or only show a partial view of the picture. By offering a service that lets the technicians share and aggregate their viewpoints, the troubleshooting process will become clearer and more efficient. The specific Enterprise 2.0 software would have to be evaluated and chosen by management.

In terms of the architecture of a Mashup application, it would make use of what Godinez et al. refer to as "Data Service Interface[s]" (2010, ch. 12.2). A Data Service Interface uses a well-known, standard protocol like HTTP, AJAX, or RSS in combination with a standard data format like XML or JSON to bring together data from different sources (Godinez et al. 2010, ch, 12.2). Figure 6 shows what the VIPER help desk can do, on a conceptual level, to utilize Data Service Interfaces to create a Mashup application.

Roadmap

Now that we have proposed solutions to all identified issues in the VIPER Help Desk's Enterprise Architecture, we present a Roadmap to demonstrate the timeline and process that will be followed for overhauling the EA. The Roadmap is closely aligned with the TOGAF Architecture Development Method, as most of the proposals are related to it in significant ways, though some phases have been purposely skipped.

Phase	Rough Timespan	Activities
Preliminary Phase	1 - 2 Months	• Form Enterprise Architecture Working Group • Gain approval from key managers and stakeholders
Phase A	2-4 Months	• Develop the Architecture Vision • Agree on the scope of the VIPER Enterprise Architecture

Phase B	2-4 Months	• Understand business processes and business architecture • Create Key Performance Indicators that will declare a solution successful or unsuccessful • Redefine L1 Technician Role
Phase C	3-6 Months	• Hash out “KID stuff”, i.e. Knowledge, Information, and Data involved in Open Source Initiative, Enterprise 2.0 software, and Mashup applications • Understand regulations and NENA best practices
Phase D	3-6 Months	• Assess which hardware systems work best for VIPER. Test and Prototypes • Choose applications for Enterprise 2.0 software suites which are compatible with well-known protocols
Phase F	6 Months – 1 Year	• Phase out legacy VIPER systems while introducing new ones

		- Eliminate previous knowledge base deliverables and deliver new ones
Phase G	3 Months	- Assess post-implementation concerns - Review with key stakeholders
Phase H	3 Months	- Now that architecture is more stable, plan for incremental changes

Conclusion

The prospect of addressing issues that are so deeply entrenched in the corporate culture, personnel, data architecture, information systems, and business architecture of an organization is a daunting task. An almost overwhelming number of considerations need to be made at all points in an initiative aimed at aligning business goals with IT solutions.

Therefore, it is highly advisable to apply an Enterprise Architecture framework to any initiative that is of importance to the business. Doing so presents the IT or business practitioner with a repository of useful tools, framings, and guidelines which have been carefully honed over the relatively short existence of Information Technology as a profession.

West Safety Services' GSD would greatly benefit from taking a step back and analyzing its Enterprise Architecture as it stands and envisioning what the architecture could be with a well thought-out plan. The over-allocated personnel budget spent on an excessive number of L1 technicians could be more efficiently used on technicians with programming knowledge.

Disparate knowledge base systems could be connected to present a single view of reality to technicians, thereby empowering them to resolve issues more quickly. Budgets which are spent on re-licensing outdated and insecure software could be put towards new initiatives while Free and Open Source Software takes its place. Utilizing Discover Components to explore data sources within the organization promotes new and novel uses of existing data. Updated methods for developing systems keeps the company marketable and modern, qualities which are indispensable in an ever-evolving, competitive market. Finally, following industry consortium standards, in this case NENA, will ensure that the product meets customer needs and regulations.

References

Erickson, Tammy. 2009. “Who are the Right People to Invite on your Bus?”. *Harvard Business Review Online*. Originally published May 9. Accessed November 11, 2017. https://hbr.org/2009/05/who-are-the-right-people-to-in

Microsoft. 2014. “Support for Windows XP ended”. Last updated 2014. Accessed online November 11, 2017. https://www.microsoft.com/en-us/windowsforbusiness/end-of-xp-support

Microsoft. 2017. “Search Product Lifecycle”. Last updated 2017. Accessed online November 11, 2017. https://support.microsoft.com/en-us/lifecycle/search/1163

Nikhil, Pathania. 2016. *Learning Continuous Integration with Jenkins.* Packt Open Source.

West. 2017. “SAFETY SERVICES TECHNOLOGY AND COMMUNICATIONS IMPROVING EMERGENCY RESPONSE. Accessed November 9, 2017. https://www.west.com/safety-services/

Ross, Jeanie, Peter Weill, and David Robertson. 2006. *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*. Harvard Business School Press. Boston.

Bente, Stefan, Uwe Bombosch, and Shailendra Langade. 2012. *Collaborative Enterprise Architecture.* Morgan Kaufman Publishing.

Harrison, Rachel. 2011. S*tudy Guide - TOGAF 9 Foundation, 2nd Edition.* The Open Group*.*

Godinez, Mario, Eberhardt Hechler, Klaus Koenig, Steve Lockwood, Martin Oberhofer, and Michael Schroeck. 2010. *The Art of Enterprise Information Architecture: A Systems-Based Approach for Unlocking Business Insight.* IBM Press.

NENA. 2015. “NENA Standards & Other Documents”. Accessed online November 12, 2017. http://www.nena.org/?page=Standards

Free Software Foundation. 2017. “What is Free Software?”. Accessed online November 12, 2017. https://www.gnu.org/philosophy/free-sw.en.html

Kotter, John P. 2012. *Leading Change.* Harvard Business Review Press. Boston.