A Method for Selecting and Sequencing Enterprise Software to Improve High-Volume New Product Introduction Performance

by

Daniel Burseth

B.S., Northwestern University, Evanston, 2007

Submitted to the MIT Sloan School of Management and the Engineering Systems Division in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Systems Engineering

in conjunction with the Leaders for Global Operations Program at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Author .... Signature redacted

MIT Sloan School of Management and the Engineering Systems Division

May 8, 2015

Certified by .... Signature redacted ....

Warren Seering, Thesis Supervisor

~~Weber-Shaughness~~ Professor of Mechanical Engineering

Department of Mechanical Engineering

Certified by .... Signature redacted ....

Thomas Roemer, Thesis Supervisor

Senior Lecturer

MIT Sloan School of Management

Approved by .. Signature redacted ....

Munther A. Dahleh

William A. Coolidge Professor of Electrical Engineering and Computer Science

Chair, ESD Education Committee

Approved by .... Signature redacted ....

Maura Herson

Director, MBA Program

MIT Sloan School of Management

A Method for Selecting and Sequencing Enterprise Software to Improve High-Volume New Product Introduction Performance

By

Daniel Burseth

Submitted to the MIT Sloan School of Management and the Engineering Systems Division on May 8, 2015, in partial fulfillment of the requirements for the degrees of
Master of Business Administration
and
Master of Science in Systems Engineering

Abstract

Powerful software exists to help consumer electronics enterprises manage complex product lifecycles and improve the speed with which they introduce new products to the market. While significant research exists to inform the organizational design for a high-performing new product introduction process, less has been done to inform strategies for selecting and implementing enterprise software that can play a key role in the market success or failure of a new product.

Studies and surveys suggest that only 10% of large enterprise software initiatives are completed on-time, within budget, and in scope. While many project management methodologies offer treatments to increase the probability of implementation success for a single software initiative, few methods exist to help inform the selection and sequence for a portfolio of initiatives. This paper proposes a method to reduce the implementation risk of large software projects within a product development organization by completing smaller projects that require the technology team to build their understanding of a complex phase-gate product development model. This method was utilized by Verizon Communications to on-board enterprise software to address time-to-market delays in the first consumer electronics produced in their new product development model.

Thesis Supervisor: Warren Seering
Title: Weber-Shaughness Professor of Mechanical Engineering and Engineering Systems

Thesis Supervisor: Thomas Roemer
Title: Senior Lecturer in the MIT Sloan School of Management

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

First and foremost I would like to thank my advisers Thomas Roemer and Professor Warren Seering for their insight, guidance, and support on this project. Additionally, I would like to thank the Leaders for Global Operations staff and friends.

I would also like to thank my colleagues at Verizon Communications specifically Gina Balunas, Jeremy Giese, and Viju Menon.

Lastly, I would like to thank my family and especially Lauren, Sonja, and Joyce for their encouragement and support during my time at MIT.

Contents

List of Figures

Chapter 1

1. Introduction to Enterprise Software in a Product Organization

Corporate spending on information technology in 2013 was estimated to be $2.2 trillion including $542 billion spent on software alone (Lunden, 2013). A portion of this software spend is by consumer electronics enterprises that purchase enterprise software and services from third parties, renew software licenses, and develop proprietary systems. Corporate surveys suggest that large enterprises perform poorly at implementing large software projects on time, in budget, and in scope relative to plans and expectations. Furthermore, the likelihood and extent of poor implementation performance is strongly correlated with the size of the software project (The Standish Group International, 2013).

Enterprise software that supports communication, process management, and asynchronous and synchronous collaboration is the backbone of the product development organization (PDO) within a consumer electronics enterprise. Investment in enterprise software is continuous and typically well-funded with the goal of supporting the PDO in making time-to-market improvements (BCG, 2012). Time-to-market is defined as the number of days it takes a PDO to conceptualize, design, produce, and deliver a desirable product to customers. Organizations that can delivery quality products quickly not only benefit from a longer sales opportunity period, but may also participate in shaping industry standards, select the best suppliers and partners, and be responsive to evolving customer preferences.

Verizon Communications ("Verizon") core business is the delivery of high quality data transmission services to customers over a network of custom and off-the-shelf hardware. Significant competition exists in the marketplace and the continuous improvement of services is a key technology objective with strategic M&A, partnerships, and internal R&D among the options available to Verizon's executive leadership to purse that goal. A recent executive decision was made to improve service quality by changing how Verizon sources data transmission hardware like in-home network routers and cable television set-top boxes. Whereas previously Verizon leveraged a large OEM network, a new Joint Development Model (JDM) internalizes to Verizon many product development functions and responsibilities that historically resided outside of the enterprise.

The transition from a model of heavy reliance on OEMs to a JDM is on-going and requires a significant amount of time, capital, and the coordination and commitment of hundreds of people to support new operations. Existing employees take on new responsibilities and new employees are hired to fill expertise gaps that cannot be met internally. Departmental reorganizations occur and strategic relationships with a design partner and a Chinese contract manufacturer form. A budget for enterprise software to support the new model has been allocated and Verizon's Supply Chain Services organization plays a leading role in software selection, implementation, and socialization throughout the product development organization.

A new product development model at one of the world's largest enterprises requires significant technology planning. Key questions whose answers will help inform the plan include:

1. Who contributes to product development efforts and what are the new processes?
2. How much would each of the processes benefit from an investment in enterprise software?
3. Given a limited budget, timeline, and technology team, what is the best method to choose among many possible enterprise software investments?
4. Is there a sequence of implementations that will deliver the most value to the enterprise and reduce the average time-to-market of the product development lifecycle?

This thesis utilizes a case study method in which these questions were addressed and resolved while selecting and implementing software at Verizon to support the new joint development model.

1.1 Problem Statement

Two hardware products have been developed at Verizon in the new joint development model. Few enterprise software investments have been made to support the complex new workflow. Examination of the development process and time-to-market of the two products informs the rate of organizational learning, employee capability gaps, sources of delays, and opportunities for technology investment. The first two hardware products missed their target launch date by an average of 207 days.

Examination of the two products led to the identification of root causes for time-to-market delays, some of which can be treated with investments in enterprise software. There is significant difficulty making optimal selections among possible technology investments. Employees that are responsible for product development sub-processes have the most informed opinions regarding the benefit of an investment in their area, but little basis to compare benefits to other investments far removed from their own. Executives have less informed opinions regarding the benefit of a investment, but a better basis to make investment comparisons across the entire product development process.

Lastly, the joint development model is still new and so processes are in flux, certain roles are vacant, and organizational understanding of all the activity is low. Few employees, if any, have an understanding of all the sub-processes that come together to define Verizon's product development lifecycle. Before any software investments can be made, a significant amount of work is needed to understand what processes are and are not occurring.

1.2 Literature Review

1.2.1 The Importance of Time-to-market

The speed with which a product development organization introduces new consumer electronics to the market is a critical factor to the success of telecommunications ("telecom") enterprises and a primary executive focus area (BCG, 2012). Folgo argues that consumer electronics like those produced by telecom enterprises begin obsolescence well before the units are available for general consumption by consumers (Folgo, 2008). A product with a shorter time-to-market affords a longer period for sales than the same product with a longer time-to-market resulting in a better opportunity for commercial success. Additionally, a company with a short time-to-market may have a "greater chance to shape industry standards, may lock up distribution channels", and can remain more nimble to changing customer demands (Chen J., 2009).

1.2.2 Organizational Design Considerations

An abundant body of literature and corporate surveys have identified best practices for improving new product introduction and reducing time to market. Griffin has nicely aggregated the research and identified key themes among the highest performing organizations including early supplier involvement, the criticality of the planning stage, and the role of enterprise architecture to connect organizations (Griffin, 1997). In addition to best practices, Hansen and Nohria conducted a survey of 107 executives that identified the barriers of "inter-unit collaboration" at large organizations:

1. Unwillingness To Seek Input and Learn From Others – "in-group bias" is the tendency to overvalue their own group and undervalue nonmembers"
2. Inability To Seek and Find Expertise – Large product organizations often require collaboration among hundreds of individuals that are globally dispersed. When coupled with frequent reorganizations and personnel changes, employees often do not find the right person when valuable expertise is needed.
3. Unwillingness To Help – Performance evaluations, competition among individuals and "turf protection" can result in an unwillingness for those with

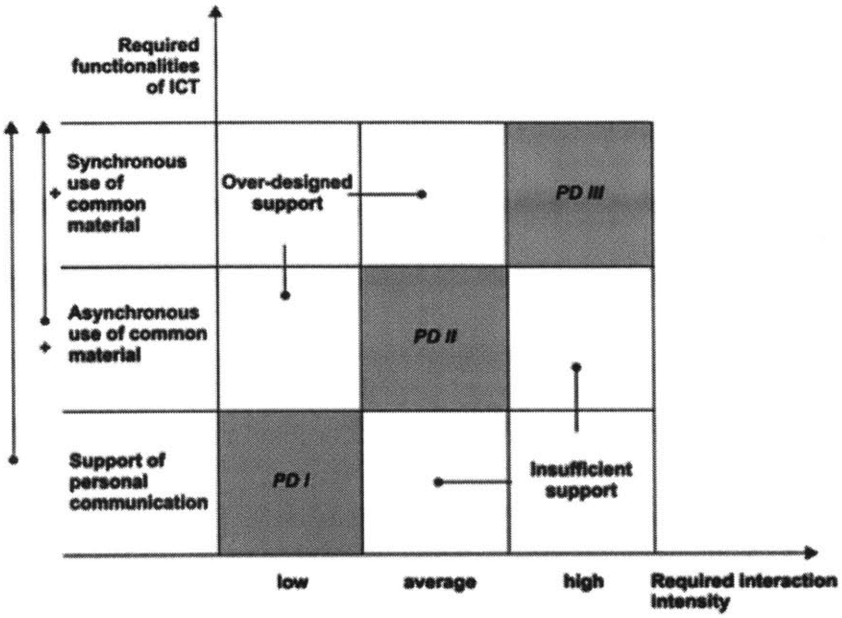
expertise to respond to or proactively seek out opportunities to improve collaboration.

4. Inability To Work Together and Transfer Knowledge – The communication that occurs in new product development is complex and difficult to standardize. While employees may be willing to help and interact, varying domain expertise and nuanced tacit knowledge makes improvement to collective organizational learning slow.

1.2.3 The Role of Enterprise Software

Extant research related to improving the performance of a product development organization primarily focus on process design and enterprise architecture topics. Less research has been done related to the role and opportunity for enterprise software to reduce time-to-market. Silva and Chathurani utilized a survey-based approach to develop an index to measure the efficacy of the enterprise software at various stages in the product development process (Silva, 2014). Kern and Kersten developed a framework for software enhanced collaboration processes within a product development organization (Kern, 2007). As figure 1 indicates, they assign processes to one of three categories based on the "interaction intensity" among employees involved in the process:

Figure 1: Kern and Kersten activity interaction types



- PD I: These are activities that involve the "exchange of explicit and routine information". Sample activities in this category include ordering a specific quantity of parts from a vendor or reporting production yields to the manufacturing organization.
- PD II: These are activities focused on "common problem solving". In addition to software supporting the "exchange of explicit and routine information", they require software that enables the "asynchronous processing of common material". Materials used in these activities are "richer" (e.g., CAD drawings) and may require many different employees to review proposed changes.
- PD III: These are activities focused on collaboration. In addition to software "supporting personal communication and asynchronous processing", they require use of software that can support synchronous processing that occurs, for example, when new product requirements are being debated and synthesized.

While the Kern and Kersten framework offers insight into required functionality given an interaction intensity level, it presupposes that design processes are mature and that the design organization is able to easily gather requirements when evaluating new enterprise software. These assumptions are not typically valid of new product development organizations like that of a start-up or a mature organization like Verizon onboarding new product capability (Mentzer, 2008). Furthermore, Kern and Kersten focus specifically on front-end applications and provide little discussion on configuring critical support layers like the data, delivery systems, and security architecture.

This research builds on Kern and Kersten by using a case study approach to develop a method to determine the implementation order of the enterprise software that supports the product development lifecycle. Unlike Kern and Kersten, the method is particularly relevant to enterprises that are new or in flux and thus processes tend to be weakly defined and the organizational understanding of what is happening is low.

1.2.4 Enterprise Software Failure Modes

An analysis of roughly 50,000 corporate technology projects by the Standish Group is the most comprehensive data set for understanding enterprise software failure modes. They have been conducting an annual survey of CTOs, CIOs, and high-level technology project managers since 1985. The survey consists of about 60% of U.S.-based projects, 25% Europe-based, and with the rest of the world making up the remainder with "more than half of the companies are considered Fortune 1000-type companies" (The Standish Group International, 2013). Projects are categorized to one of three implementation outcomes:

1. Successful: delivered on time, on budget, with required features and functions
2. Challenged: late, over budget, and/or with less than the required features and functions
3. Failed: cancelled prior to completion or delivered and never used

Figure 2 is a summary of the Standish survey data that indicates consistently poor software implementation performance:

Figure 2: Standish software implementation survey results

	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Challenged	53%	46%	44%	42%	43%
Failed	18%	19%	24%	21%	18%

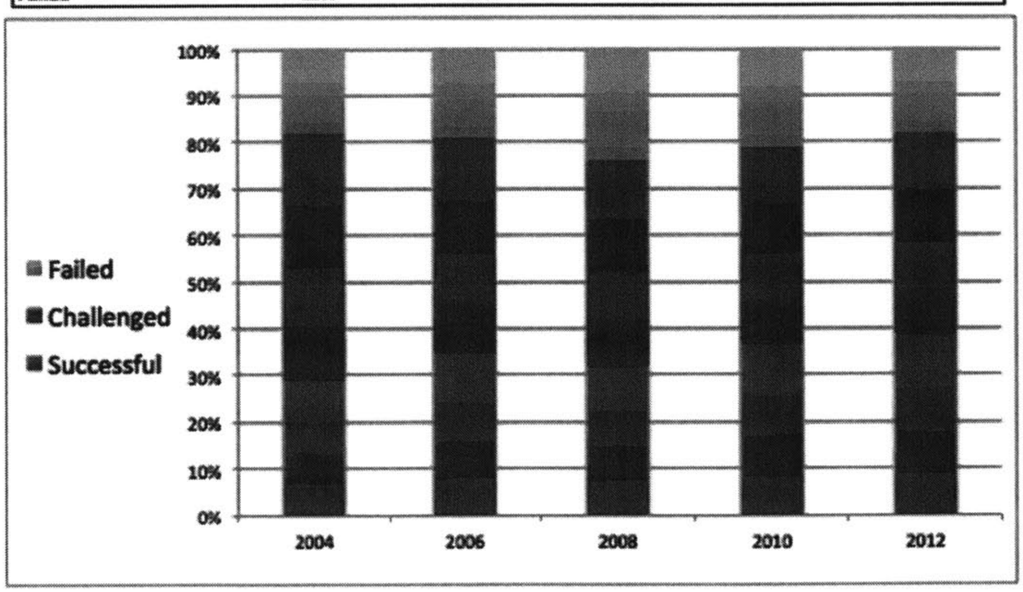
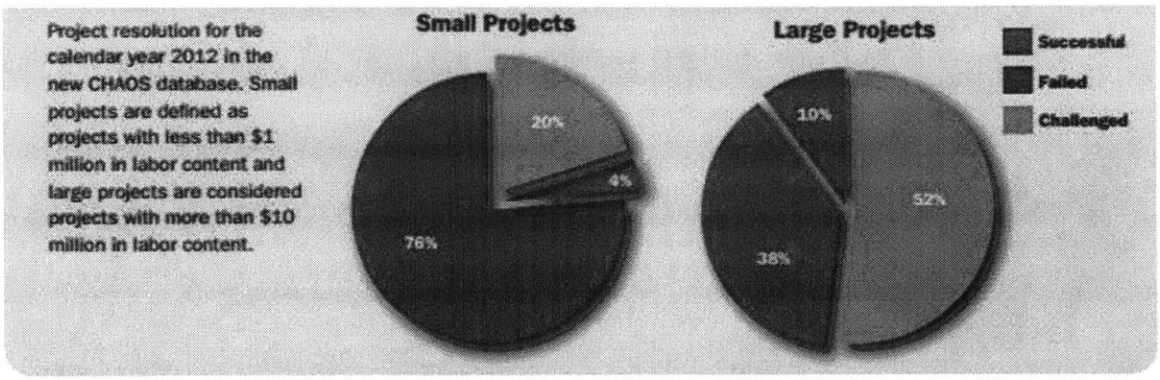


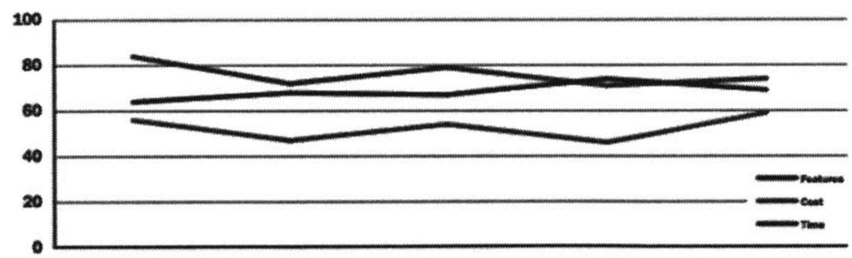
Figure 3 is also from the Standish survey and suggests that project size tends to be anti-correlated with the probability of success.

Figure 3: Implementation performance by project size



Lastly, figure 4, also from the Standish survey, provides insight into the extent to which challenged projects over-run their planned timeline, budget, and experience scope creep.

Figure 4: "Challenged" project overruns on time, budget, and features

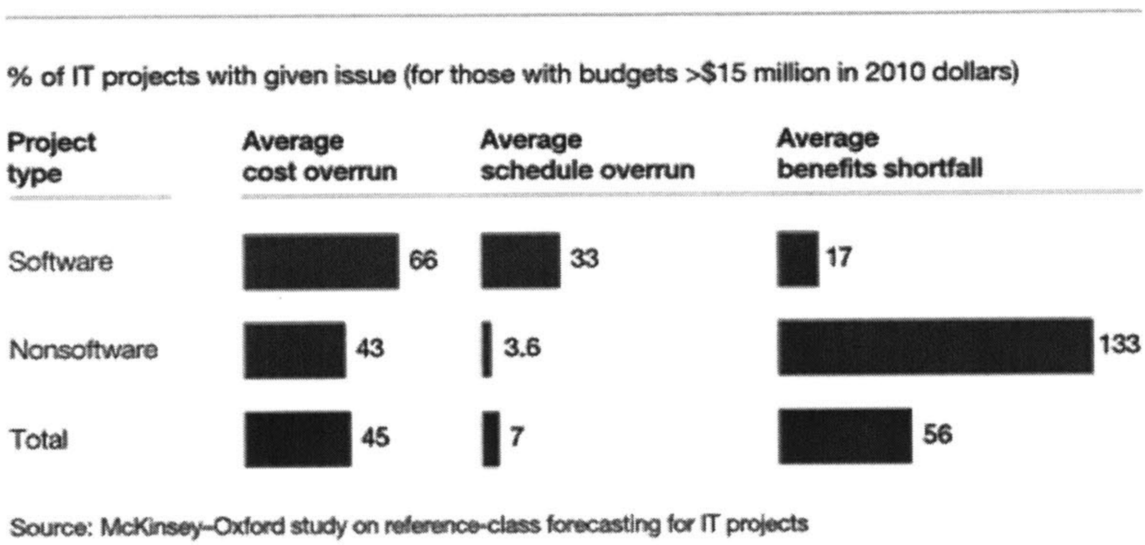


	2004	2006	2008	2010	2012
TIME	84%	72%	79%	71%	74%
COST	56%	47%	54%	46%	59%
FEATURES	64%	68%	67%	74%	69%

Example interpretation: Challenged projects in 2012 were 59% over-budget on average.

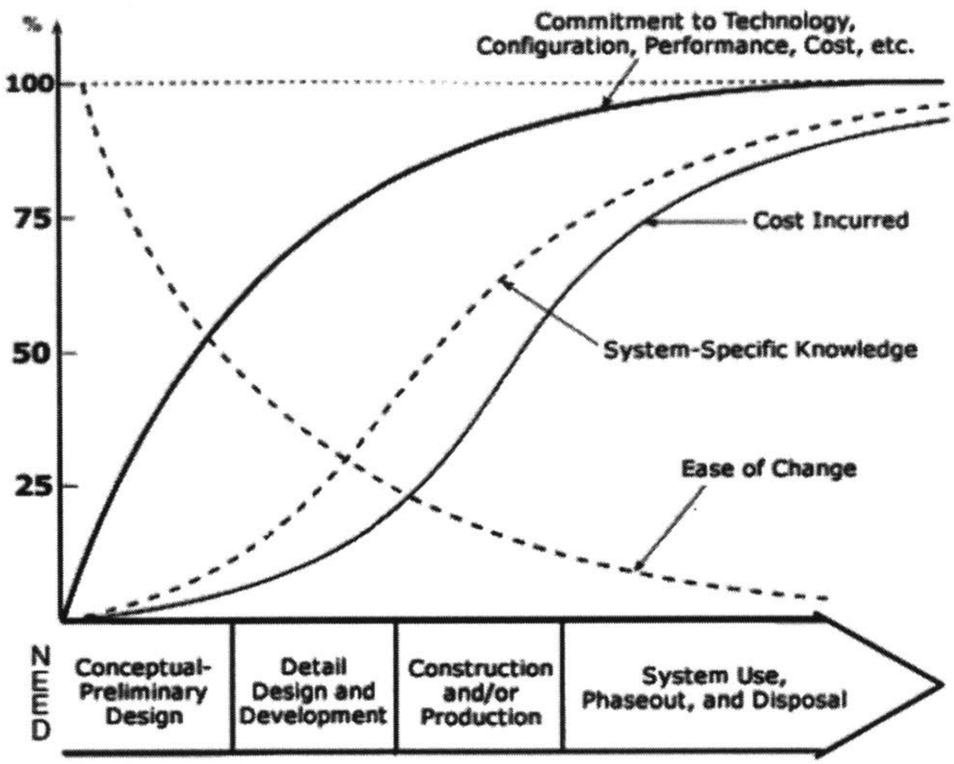
A joint study by McKinsey and the University of Oxford on the implementation performance of "large IT projects" (projects with an initial budget greater than $15 million) (Bloch, 2012) supports the Standish findings and is summarized in figure 5:

Figure 5: Additional survey data on project overruns



Failure to meet time, budget, and scope estimates is attributable to a combination of poor initial estimates as well as execution issues once a project has commenced. The cost commitment framework developed by Blanchard and Fabrycky and presented in figure 6 depicts the various relationships and tensions present in the design and implementation of a complex system like product development support technologies (Blanchard, 1998).

Figure 6: System design trade-offs and tensions



Chapter 2

2. A New Product Development Model

Historically, Verizon has purchased its customer premise equipment like Product A and Product B from OEMs. Verizon is re-designing their product development model and moving many functions in-house in an effort to:

(1) introduce new products faster

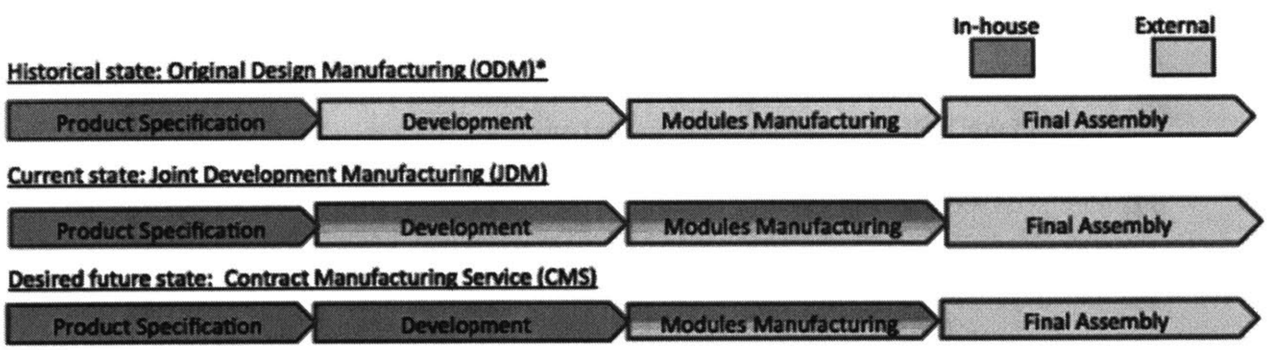
(2) reduce development and end-product costs

(3) develop an intellectual property portfolio

(4) integrate higher margin services (e.g., content, security services)

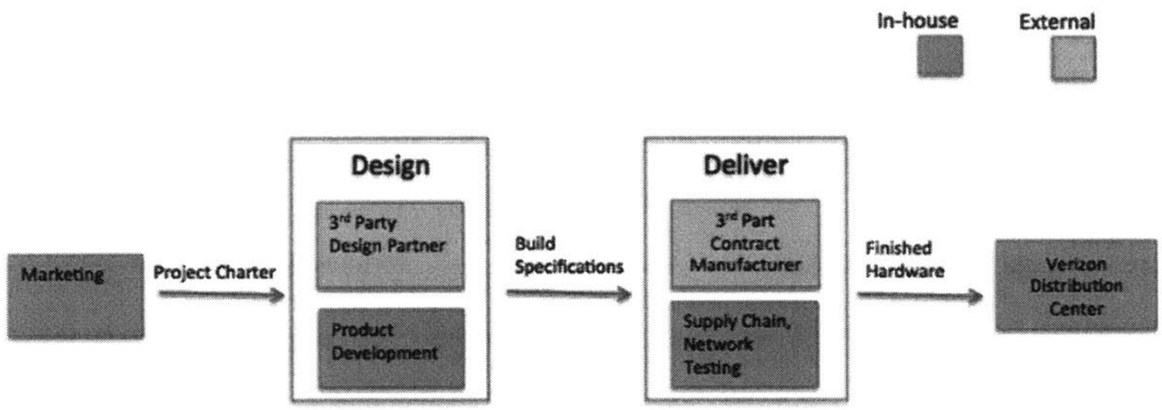
The historical, current, and desired future state are depicted in figure 7:

Figure 7: Select product development models



Verizon has partnered with an external design firm that has deep consumer electronics hardware and software development expertise. The design firm partnered with a China-based contract manufacturer to manage volume production and delivery to Verizon's warehouses in Pennsylvania and Texas. Additionally, Verizon hired a new internal technical team to help translate the project charter to product design specifications. A basic flow of information and the organizational units involved in Verizon' s product development organization is described in figure 8.

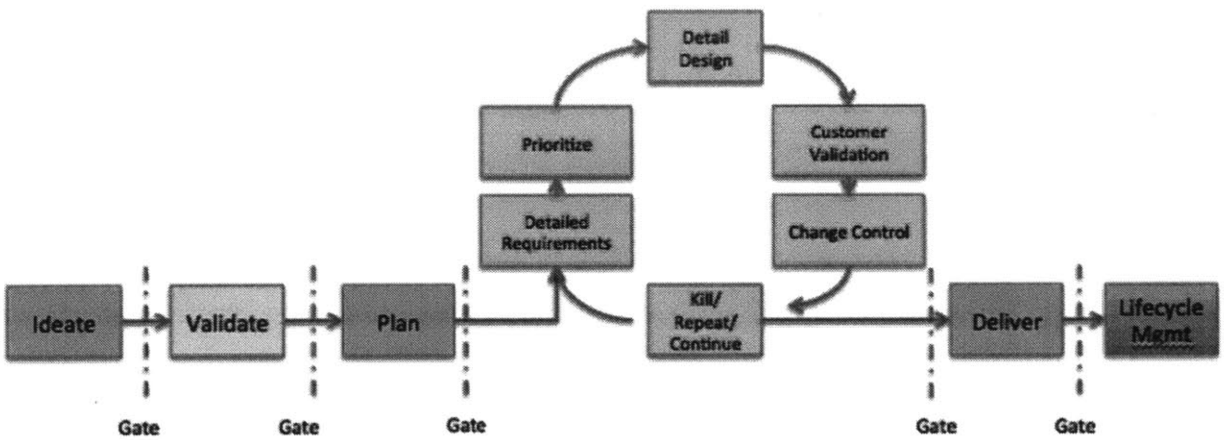
Figure 8: Verizon Product Development Organization



2.1 Process

The new JDM utilizes a phase-gate model with six phases. Sign-off by senior vice presidents is required at every gate and is based on a fresh assessment of the market demand, risk of execution, and the available resources for completion. The external design partner and the contract manufacturer are integral to all phases and their compensation is linked to the timely and successful completion of phases. For example, the design partner, in conjunction with the internal technical team, is responsible for delivering final build specifications and prototypes to the contract manufacturer to complete the "Design" phase and to proceed to the "Deliver" phase. In the "Deliver" phase, Verizon's procurement, manufacturing, and supply chain organizations coordinate the production and distribution of upwards of two millions units per product launch. Figure 9 depicts the activities associated with each phase in Verizon's phase-gate product development model:

Figure 9: Phase-gate product development model



2.2 Output

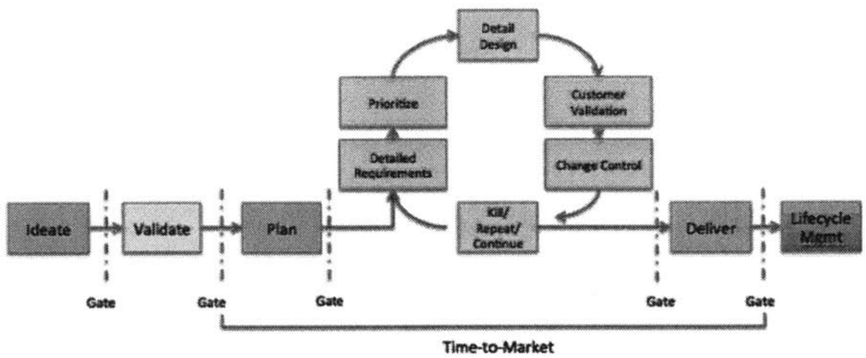
The "Product A" and "Product B" are the two pieces of hardware that have been produced in the joint development model. Both products reside in customer homes and support the transmission and reception of the cable television and broadband internet services.

2.3 Measuring Results

To better understand how and where enterprise software could support the joint development model, a postmortem analysis was conducted on the development of the

Product A and Product B. Time-to-market has the advantage of being a readily observable and easier to measure than other metrics like total development cost. For purposes of the postmortem analysis, time-to-market is defined as the number of days it took Verizon to plan, design, and deliver the Product A and Product B to customers. Figure 10 depicts the phases that are included in the time-to-market definition:

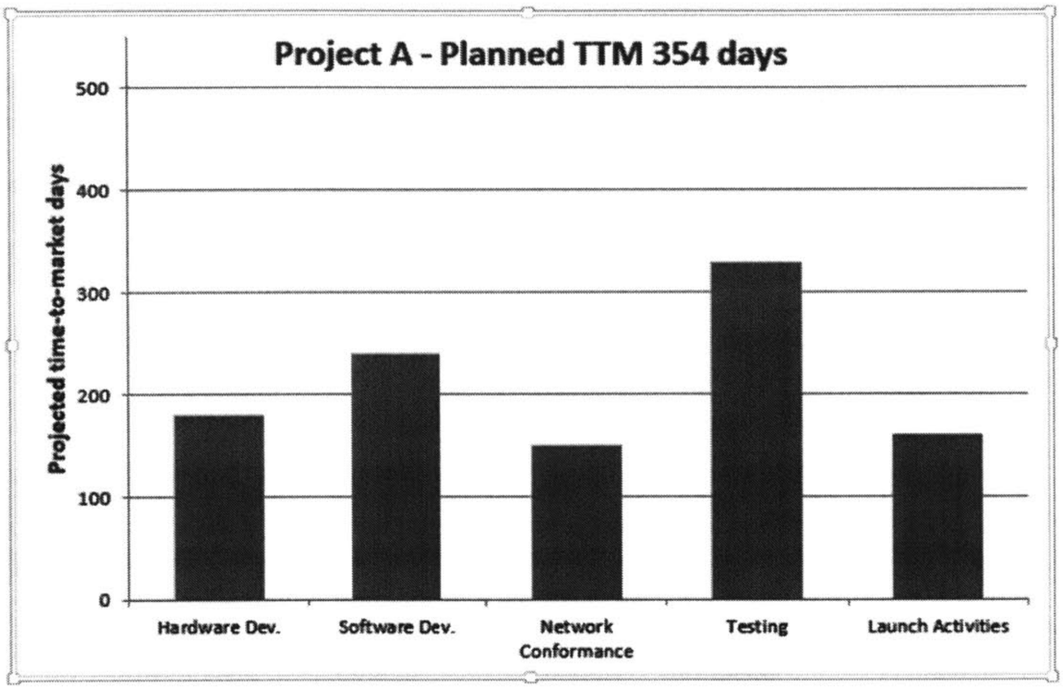
Figure 10: Time-to-Market definition

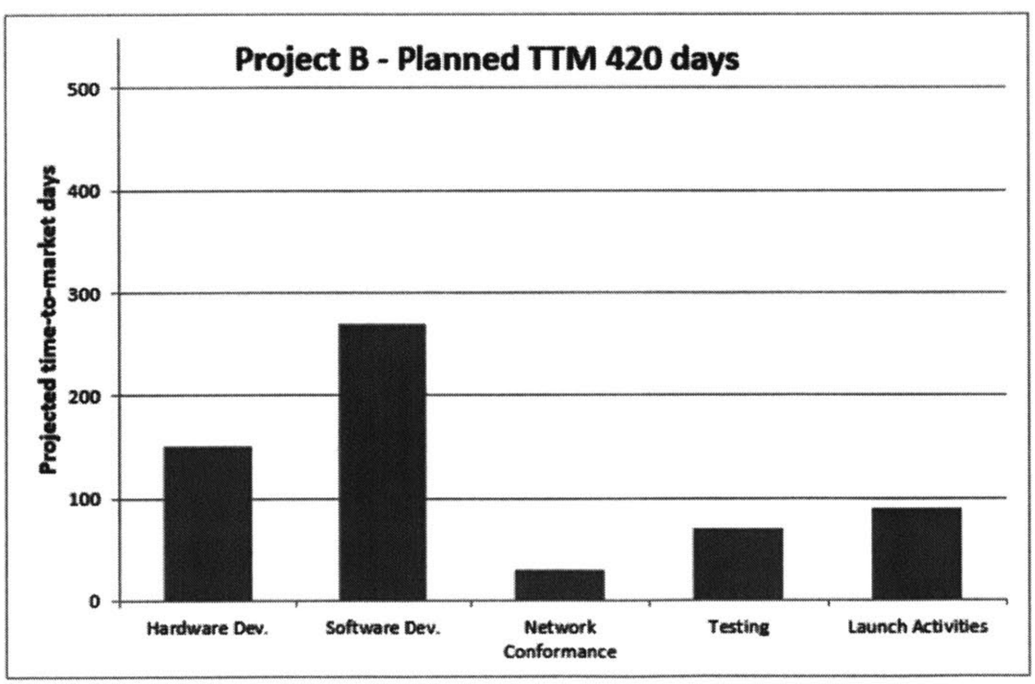


There is a clear argument for including the days spent in the "Ideate" and "Validate" stages which are excluded from the definition in figure 10, but this data was not logged. Additionally, an analysis assessing the efficacy of the joint development model should include quality data (e.g., mean time to failure), sales performance, and profitability. This data should be considered when longer product histories exist.

The original project plans that coincided with the executive-approved clearance of the "Ideate" gate for the Product A and Product B were obtained. Each plan detailed upwards of 100 individual sub-processes, decisions, and dependencies needed for the nation-wide general availability of the Product A and Product B. Figure 11 is a categorization of all the planned development activities with a corresponding aggregation of the expected day count to complete the activities:

Figure 11: Planned product development activity for the Product B and Product A



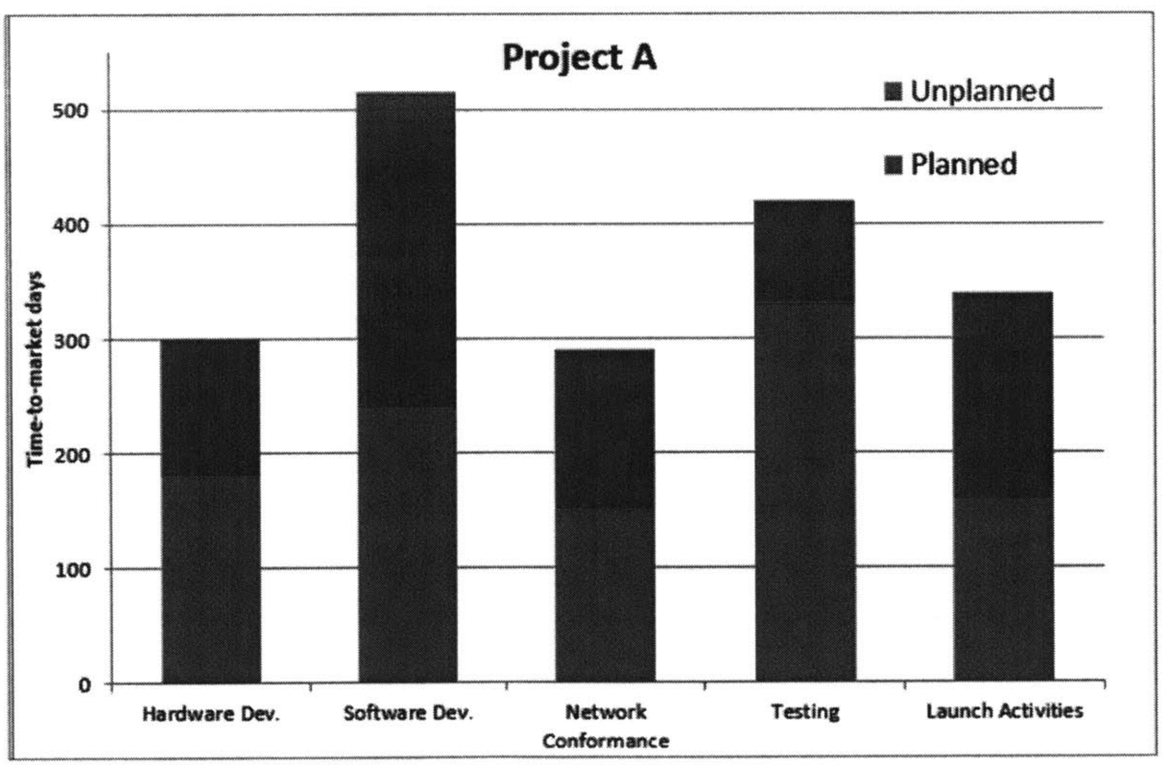


In figure 11, certain development activity occurs in parallel so column totals do not sum to the aggregated planned time-to-market.

2.3.1 Planned vs. Unplanned Activities

As the products progressed through the "Plan", "Design", and "Deliver" stages, delays began to mount and deadlines slipped. An approached used by Folgo to compare planned vs. unplanned work was used for the Product A and Product B (Folgo, 2008) and is represented in figure 12:

Figure 12: Planned vs. unplanned product development activity



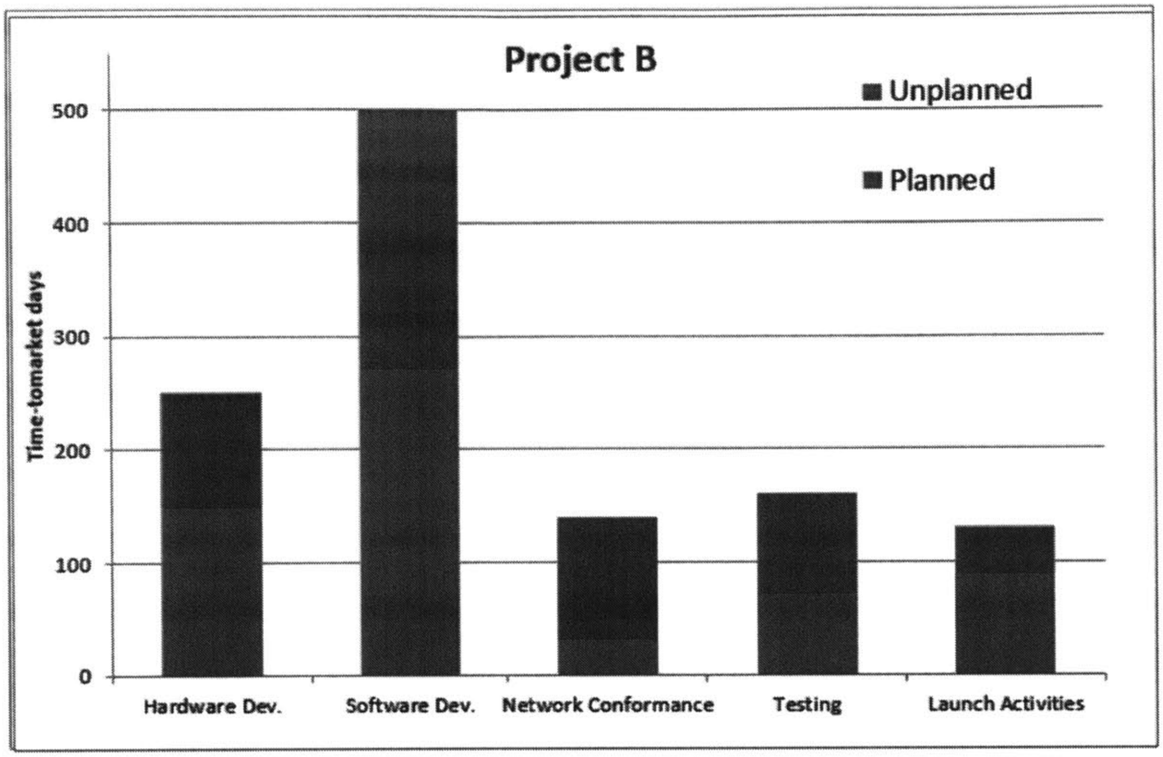


Figure 12 shows significant project slippage in every major product development activity. In both products, software development is the largest source of delay. Figure 13 calculates the difference in the planned time-to-market with the actual time-to-market and computes " timeline percentage slip" . Project slip, defined as the percentage difference between the actual vs. planned time-to-market, was greater than 50% for both the Product A and Product B.

Figure 13: Planned vs. actual time-to-market comparison

	Planned Time-to-market	Actual Time-to-market	Timeline % Slip
Product A	354	548	55%
Product B	420	640	52%

Project slip does not imply that all launch delays are attributable entirely to process execution. There is an argument to be made that perhaps the initial planned activity was incorrect and underestimated, but even with a conservative estimate of more than 1.5 years spent in the "Design" and "Deliver" phases per product, the joint development model did not deliver on the strategic goal of improving data transmission services by innovating critical hardware quickly.

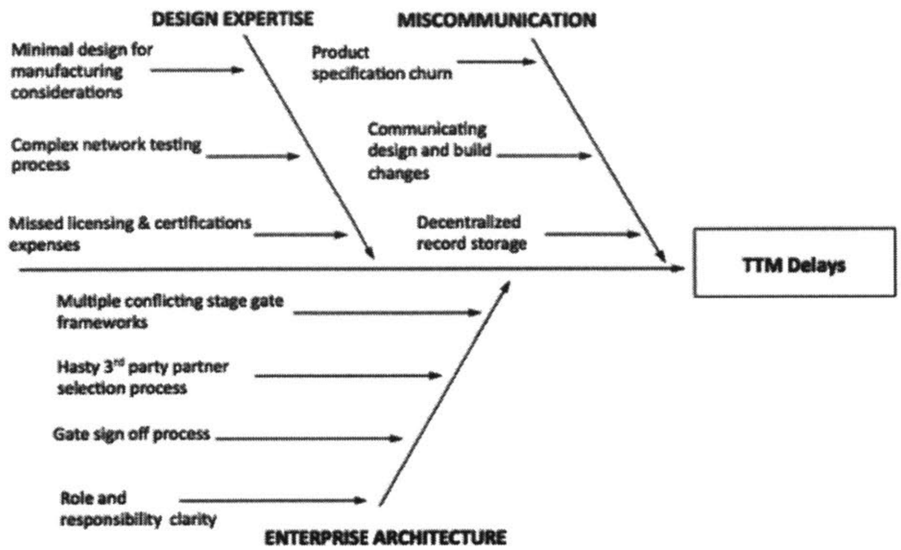
2.3.2 Root Cause Analysis

A root cause analysis was conducted to understand the drivers of the time-to-market delays. Methods and sources to inform the analysis include the technical test results, interviews with participants in all phases of the product development lifecycle, and engineering change orders (ECOs). The interviews were conducted in accordance with Committee On the Use of Human Experimental Subjects (COUHES) standards.

Given the (1) high number of people involved in the product development process, (2) a web of connected sub-processes, and (4) the iterative nature of building and evaluating prototypes, there is significant difficulty in isolating root causes to time-to-market delays. As an example, the firmware development sub-process for the Product B was delayed by the antenna selection decision for the Product A. The antenna selection delay was attributable to open issues related to desired range and performance specifications in the

Product B. Dozens of occurrences like this were studied and themes were organized into three categories in the root cause fishbone chart in figure 14.

Figure 14: Time-to-market delay root cause fishbone diagram



This root cause analysis was the critical first step for understanding the complex process that occurred to develop the Product A and Product B. Kern and Kersten's framework for software enhanced product development collaboration can lend guidance to treating events in the "miscommunication" category with the goal of implementing enterprise software to reduce time-to-market delays. While perfect attribution is difficult, an estimate of the time-to-market delay for each sub-category was assigned and is summarized in figure 15:

Figure 15: Attribution of root causes to time-to-market delay days

Root cause	Description	Estimated number of time-to-market delay days
Product specification churn	The product specifications developed during the "Plan" phase drive all downstream production activity related to procurement, supply chain, and manufacturing. Revisions to the specifications can require highly constrained troubleshooting that results in delays.	60
Communicating design and build changes	Production changes to things like suppliers, costs, and testing requirements are to be expected throughout the product development lifecycle. Communicating these changes to all impacted parties is difficult and can result in missed or duplicated work.	25
Decentralized record storage	Multiple locations exist for the storage of schematics, bill of materials, and component data which results in errors and confusion	5

Additional detail on " product specification churn" is warranted given that it is such a large driver of time-to-market delays. The product specification process is all the design activity that occurs to translate a business and use case for a new product into a product design and manufacturing specifications. Ideally, the Design and Deliver phases can occur serially, but it was common in the development of the Product A and Product B for significant changes to the product design and manufacturing specifications to occur well into Deliver phase. As a result, complex re-work related to part procurement, supply chain coordination, and manufacturing processes was commonplace as a result of the churn.

Chapter 3

3. Software Support for the New Model

Analysis of the design and delivery to the consumer market of the Product A and Product B reveals significant opportunity to improve Verizon's new product introduction process. Given Verizon's commitment to the new model and the growth of the new product pipeline, executive leadership continued to hire personnel for key positions, reorganized the design organization, and selected the Supply Chain Services group to lead an effort to improve the communication methods of employees involved in the Design and Deliver phases of the product development lifecycle.

The new mandate for Supply Chain Services was broad and afforded significant opportunity to address many areas of the product development process that contributed to time-to-market delays. The root cause analysis and delay attribution from Figures 10 and 11 form a set of options to address with technology investment. While discussing where and how to deploy technology resources, there was a natural inclination to focus on the events and delays related to "product specification churn" since it is by far the largest source of delays in both the Product A and Product B. Interviews with 14 individuals representing all the groups involved in the product specification process were interviewed and a proprietary process map was developed.

The examination of the product specification process revealed an immature and highly complex workflow. Work was communicated and completed among a dozen internal groups and the two external partners using four documents, three requirements management systems, and dozens of weekly and bi-weekly conference calls. The product hardware was fairly standard in both products consisting of antennas, integrated circuits and components on a PCB, and power supply in a basic black plastic enclosure. Product development complexity was primarily software driven and specifically related to integrating new products into the broader Verizon's fiber optic network.

There is a large opportunity to reduce time-to-market delays by improving the product specification process, but a technology-based approach is not what is needed first. There is a large need to improve the organizational understanding of how product specifications are aggregated, synthesized, and disseminated. There was a belief that any attempt to implement software to support the product specification process would fail to add value and come at a huge expense of money and time. As a result, a decision was made to sequence technology projects to start with less critical miscommunication issues that can be addressed by smaller software projects. The goal was to not only reduce time-to-market delays for subsequent new products, but also develop a technology platform for use by larger projects in the future and drive organizational understanding of the product specification process.

The next two root causes from Figure 15 - "Communicating design and build changes" and "Decentralized record storage" - were compared with respect to the potential for time-to-market improvement relative to the execution risk of any efforts. Similar to the decision to defer addressing product specification churn, Supply Chain Services determined that a technologically simpler project that addresses record storage had a better risk-return profile than a larger effort focusing on how design and build changes are communicated.

Evaluation of the root causes led to a technology sequence that favored small, low-impact projects. Supply Chain Services had relatively few planning constraints as the product development process represented a technology greenfield. A different sequence is likely had more traditional project selection considerations been incorporated like hard-to-compute expected cash flows, organizational politics and the relative status of groups affected, and an existing technology roadmap. This method is likely more applicable to a new organization (e.g., hardware start-up) or an existing organization with dedicated resources to on-board brand new capability. A mature organization or process has additional considerations to factor into the technology roadmap including disruption risks and compatibility with existing technology.

With a project sequence in place, the next step was to begin the implementation of individual software projects. The project management method consisted of four steps:

1. Requirements Gathering - With up to 100 people involved in the product development process, it is a massive undertaking to understand where and how software can support the business and where it may hinder it. For these projects, requirements gathering consumed approximately 80% of the entire project timeline.

2. Technology Planning – A skilled technical project manager will evaluate the existing technologies available in the marketplace for the ability to fulfill business requirements, meet budgetary considerations, and accommodate support and flexibility in the future.

3. Installation Process – Installing and configuring a large piece of enterprise software requires significant testing, development instances, and often many add-ons, plug-ins, and system upgrades. Verizon contracted a VAR (value-add reseller) to guide the installation process.

4. Socialization and Adoption –Large enterprises are littered with software initiatives that failed to gain traction with the intended users. These failures can be the result of poor execution of any of the steps (e.g., missed business requirements), but poor socialization and training of the completed project can drive poor adoption.

3.1 Software Project #1: Centralized record storage

Requirements Gathering

The development of a product like the Product A generates an enormous amount of information that must be searchable, easily retrievable, properly secured, and version controlled. Time-to-market delays occurred when work occurred using outdated, inaccurate, and missing information. Interviews highlighted frustration over the lack of a centralized database to seamlessly access product information.

The first step was to understand the set of critical information, who maintains it, and how frequently it is updated. An end-to-end process map of the Product A development activity (presented in figure 17) was completed to reveal that the body of critical and frequently shared information in the product development process.

Figure 16: Sample process map to understand the requirements for software project #1

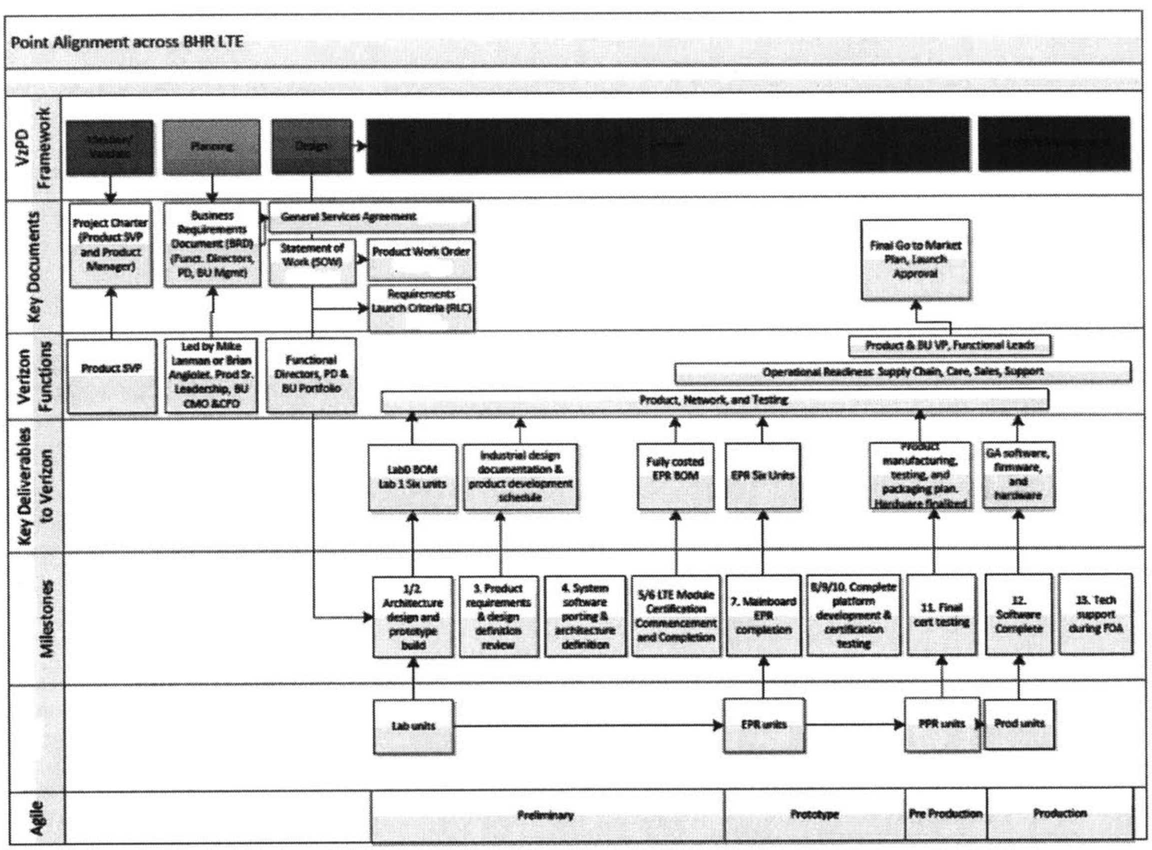
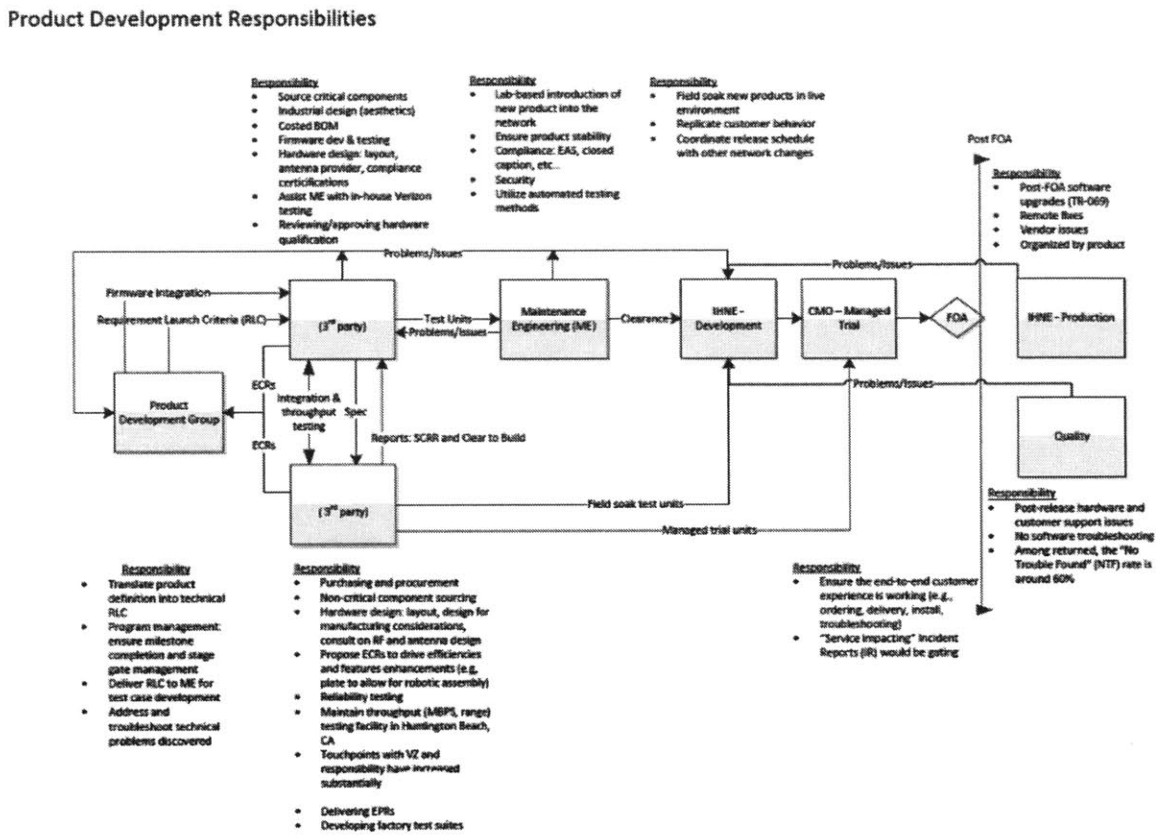


Figure 17 represents a more detailed process map of the subset of product development activities that are particularly complex and communication intensive:

Figure 17: Detailed process map of the prototyping activities



While the process maps are helpful for understanding organizational interactions, figure 18 represents a tabular view of the information that is being exchanged in the process maps. The tabular view also contains additional and critical information such as who originates or "owns" the information and who accesses it. An analysis of the ECOs for Product A provides a sense for the frequency of change of the critical information:

Figure 18: Key information flows and frequency of changes

Shared Information	Who Owns It	Who Accesses It	Major Changes during the Product A development
Component attributes	Design, Manufacturing	Supply Chain, Testing	26
Component costs	Finance, Procurement	Design	11
Structured bill of materials	Design, Manufacturing	Supply Chain, Procurement	34
Vender offerings and contact information	Procurement	Supply Chain, Design, Manufacturing	6
Assembly drawing and schematics	Design, Manufacturing	Manufacturing, Testing	8
Licenses and certifications	Legal, Testing	Design	3
Test results: failure rates, network performance, and field reports	Testing	Design, Manufacturing	44

Technology Planning

The access patterns detailed in Figure 18 lend well to a class of enterprise software called "product collaboration platforms". Record management activities are at the core of any product collaboration platform and provide a user-friendly way to contribute, access, and update product information. A handful of platforms were considered and Oracle Agile was selected based on the ability to potentially address the more complex miscommunication occurrences from the Product A and Product B and given its reputation as the gold standard among consumer electronics enterprises.

Installation Process

Once the body of information that required centralized storage was identified, the technical implementation was rather light. There was a need for a fileserver for CAD drawings and component specification sheets. A hardware taxonomy in a SQL database was defined to organize and store all the components and their attributes (e.g., a capacitor might be defined and stored as "Capacitor SMT 10pF cap. 5% tol).

All the work that went into creating an end-to-end process map served as the basis for defining and assigning access privileges to roughly 50 employees based on their roles and responsibilities.

3.2 Software project #2: Communicating Design and Build Changes

Requirements Gathering

A product development process is not afforded the luxury of finalizing every design decision ahead of the start of production activities. For example, manufacturing contracts need to be complete and negotiated well in advance of production and commitments to suppliers for long lead time parts need to be established. If a change to the product design is required after production activities have kicked off, then an engineering change order (ECO) needs to be circulated throughout the product development organization to asses the impact on work done to that point.

Time-to-market delays associated with poor communication of ECOs were common in the development of the Product A and Product B. ECO originators were uncertain who should participate in the review of the change and what pertinent information is needed. Groups complained of exclusion from ECOs that impacted their production planning sub-processes.

Technology Planning

Enterprise software was needed that could circulate an ECO template among affected parties. Ideally, an ECO originator could impart critical design change information and rely on a system to disseminate the proposal and manage the subsequent discussion and approval. A module within Oracle Agile, the technology solution selected for record management, supports exactly this type of functionality.

The process maps and centralized record system developed for software project #1 were invaluable for codifying the ECO sign-off process. An employee defining a new ECO can link it directly to an affected part or sub-assembly to avoid any confusion. New drawings and support documents can be included to help articulate the design change and link to existing documents that would replaced or impacted. Lastly, the system allows for the definition of a sign-off procedure such that critical parties like procurement, manufacturing, and network testing are aware of production altering change.

Installation Process

Transitioning the formal workflows and informal communication of a product development process to Oracle Agile's collaboration module is a significant organizational commitment with Verizon's estimate being approximately 500 hours of employee time. Additionally, hiring a Value Added Reseller (VAR) to manage the on-boarding process is the norm. Between VAR fees, licenses, software cost, and employee time, Oracle Agile installations can easily cost an enterprise $1 million. Specific work completed includes:

1. Codified Workflows - Defined user groups and workflows associated with engineering change orders (e.g., which groups need to review and sign off on a request to change the operating frequency of the 802.11 antenna from 2.4ghz to 5ghz). Assign permissions and notification triggers for all system users.
2. Administration - Defined and configured a security and administration model. Develop procedures and policy related to environment changes, upgrades, and patches.
3. Validate - Confirm the environment set-up by way of extensive scenario testing.
4. Support - Develop documentation and training materials for the extended user group.

Socialization and Adoption Activities

While the technological component is the bulk of the implementation activity, an equally critical component is the socialization and adoption of the new technology by the user-base. This effort starts early on in the requirements gathering and stakeholder analysis stage by identifying mutually agreed upon target workflows and responsibilities within the new information system. As Agile progressed closer to launch date, training sessions were held to simulate day-to-day workflows. The goal of the sessions was to reduce any opposition associated with having to learn a new and complex piece of software. Inevitably, conflicts arise and user adoption is not guaranteed, but clearly articulated and demonstrable benefits ultimately "pulled" a few holdout groups in the new workflows.

System Launch and Expected Gains

Oracle Agile was successfully implemented on time, within budget, and delivered on the mission statement to improve data management and information flows within the product development organization. Historical problem areas that drive time-to-markets delays that are addressed with Oracle Agile include (1) role clarity confusion, (2) ambiguous sign off procedures, (3) gate sign off process, (4) high geographic dispersion of expertise, and (5) minimal knowledge transfer from product to product. Additionally, Agile is expected to provide additional benefits as Verizon products proliferate. For example, a parts database will help inform part reuse decisions and help drive intelligent purchasing decision in subsequent product designs.

3.3 Software Project #3: Address product specification churn

Requirements Gathering

Product specification churn was the single largest root cause of time-to-market delays in the development of the Product A and Product B. Despite spending more than a year on a process intended to capture and solidify product specifications, the Product A incurred 15 revisions to the product specifications. These changes occurred in the midst of hardware and software prototyping and trigger costly restarts to development activities like hardware and software testing. In turn, the testing processes prevent further development activity until successful completion.

Additionally, critical downstream functions that manage the "Deliver" phase of the product development lifecycle were provided little opportunity to participate in the product specification process. As a result there were no opportunities to include design for manufacturing (DfM) or design for supply chain (DfSC) considerations in the specification. Furthermore, the product development team was not aware of all the business requirements. For example, the "Deliver" organization in charge of managing customer installations and troubleshooting was missed as a potential requirements contributor. As a result, a finalized build plan erroneously excluded a method for on-site installation technicians to interface with the product. This was discovered and addressed after the first units were already manufactured resulting in time-to-market delays and highly constrained troubleshooting.

A high-performing product specification process is difficult to achieve and is often what distinguishes the world's best PDOs from their competition. Specific challenges include:

1. Ideally evoke a "skunkworks" environment but solicit input from a wide and diverse set of groups to capture design for DfM and DfSC considerations
2. Allow for iterative and fast prototyping while maintaining transparency for management and adhering to network technical standards (e.g., Verizon's fiber optic network incorporation standards)
3. Define a stable product specification but also build flexibility into downstream functions to facilitate design changes.

Technology Planning

Perhaps the biggest benefit associated with the implementation of Oracle Agile is that it highlighted many missing processes and communication linkages within Verizon's product development organization. A bottom-up and incremental project selection strategy allowed Verizon to complete enterprise software initiatives to potentially mitigate 30 days of project slip for the next product. These initiatives were completed on-time and in-budget and provide a platform to address the production specification churn responsible for the most time-to-market delay. Specific work that can be leveraged from the product collaboration module and centralized record storage includes:

1. Standardized component references – The product collaboration platform formalized how employees communicate about part and manufacturing decisions. The component taxonomy developed for centralized record storage can be used by future design activity and hopefully alleviate any confusion that results from the dissemination of build specifications throughout the organization.
2. Provided personnel and role clarity – More than 100 employees scattered across six offices are involved in Verizon's product development process. Given the new JDM model and personnel reorganizations, there was significant confusion regarding who did what activities and how organizations were organized. The installation of the product collaboration module mapped all critical personnel to a role. Roles contain information about process responsibility and co-workers with similar roles. Future products can ideally solicit input for DfM and DfSC considerations and know exactly who to approach for this information.
3. Documented decision histories to avoid repeat mistakes – ECOs were needed to make dozens of fixes to the designs of both the Product A and Product B. Simply archiving the nature of those changes, the parties involved, and updated plans was capability that did not exist. For future products, activities in the Design phase

can review past ECOs with the goal of avoiding design mistakes as opposed to troubleshooting them during the production phase.

The engagement with Verizon completed before there was an opportunity to design and implement a system to address product specification churn. Verizon intends to explore the opportunity for enterprise software to help improve the product specification process. A system would draw heavily on previous work completed and consist of:

1. A web-based portal that serves as centralized storage for all product specifications and requirements
2. A notification system to prompt key personnel to review and approve changes to product specifications
3. A method for organizations who work in the "Deliver" phase of the product lifecycle to transmit operational, business, and manufacturing requirements into the planning process for consideration
4. A method to publish and version control finalized product specifications

Chapter 4

4. Generalized Software Selection Strategy for Product Organizations

Building a globally competitive, consumer electronics product development organization is a massive challenge. Product strategy, organizational design, partner selection, and the hiring of key personnel are just some of the areas that require attention from executive leadership.

In many organizations, enterprise software is an afterthought to support business process and is not included in initial strategy discussions. Corporate technology operating modes are often highly reactive and address software shortcomings after an error has occurred or when it is clear the existing system can no longer support the business. These modes lead to the selection of large, critical initiatives on expedited timelines and ultimately to the astounding failures rates.

Planning frameworks do exist with net present value (NPV) methods and real options being among the more popular and roadmaps are maintained within the CTO's organization. Product enterprises have limited success with these frameworks citing the subjective and highly uncertain nature of the inputs. To further complicate matters, product enterprises are often managing the concurrent development of many products across the product lifecycle. These realities coupled with a web of existing infrastructure make software overhauls difficult and add confusion to the business as legacy software often operates in parallel until it can be cleanly phased out.

In contrast to reactive and traditional project selection methodologies, the enterprise software strategy employed by Verizon in a new product development organization can best be characterized as one where a low-risk and often low-utility project is completed first. The goal is to develop business process expertise within the technology organization that translates to higher probabilities of success for subsequent and more complex. In addition to organizational learning advances, there are significant technology-related benefits. Knowing that a software project exists in a global sequence informs local decisions by placing a higher emphasis on architecture and implementation flexibility. For example, coding and structuring libraries, APIs, and platforms for a current project with some notion of future needs helps mitigate a common IT pitfall of rework due to incompatibility.

The subsequent discussion relates to the role that software projects can play in helping define and improve a complex process like new product development and a nomenclature to assess relatedness or " carryover" from one software project to the next with the goal of de-risking the technology roadmap.

4.1 Software Implementation as a Driver of Organizational Improvement

Enterprise software implementation drives discovery about how people work and who they work with. For an organization in flux, this information can be as valuable as the gains resulting from the implemented technology itself. This discovery is driven by a skilled technical project manager working closely with the process operators to understand the complex inter-organizational interactions that are likely undocumented and obfuscated to senior management. As was the case at Verizon, simply understanding the opportunities for enterprise software to support the business is the first step of any technology planning and can often consume the majority of the software project timeline.

Requirements gathering is the collection of activities that help inform how technology can add value to the business. There are dozens of different requirements gathering approaches that vary according to factors like industry and organization size, but the basic components are (1) stakeholder identification, (2) stakeholder interviews, and (3) synthesizing requirements. Process maps connecting the various stakeholders are a key tool for understanding how work is getting done as well as impediments to workflows and even missing functions. The process maps developed in the context of requirements gathering may be of interest to senior management outside of the technology function as it may highlight deviations of the current organizational implementation relative to executive strategy.

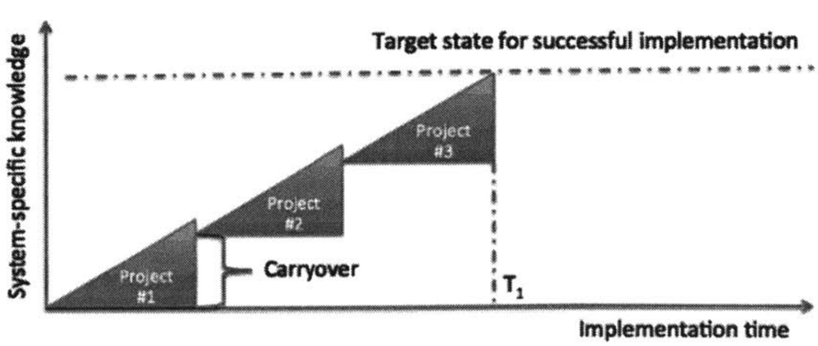
4.2 Carryover of System-Specific Knowledge

In the context of an IT project, "system-specific knowledge" refers to the organizational and procedural understanding of how the product development process works. An organization's ability to arrive at a higher level of system-specific knowledge at a faster rate reduces the risk that the IT project fails to deliver with respect to time, budget, and scope. Thankfully organizations do not need to start building their system-specific knowledge from scratch with every project since knowledge *carryover* exists. For example, an organization that has recently completed a project to centralize document storage can draw on work done in the requirements gathering stage to inform the stakeholder identification for a new and more complex project.

A software selection strategy that maximizes carryover from one project to the next commences each successive project at a higher level of system-specific knowledge. The amount of time it takes to reach the target state for the successful implementation of the technology roadmap is represented as T_i. What follows are three scenarios to illustrate the relationship between carryover and implementation time:

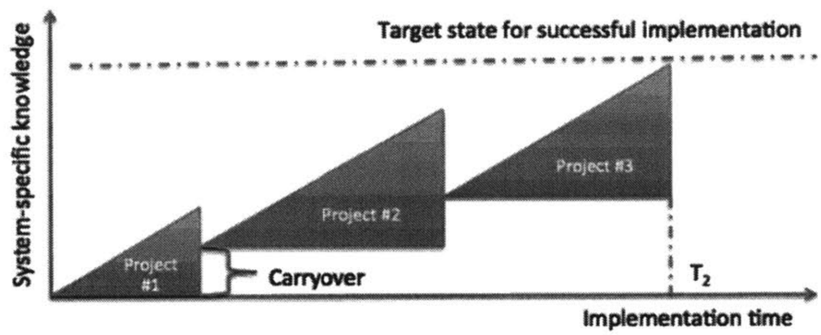
> Scenario 1: High carryover from one project to the next exists as stakeholder analysis, technology platforms, and socialization activities are partially transferable from one project to the next.

Figure 19: Scenario 1 - High carryover among projects delivers shorter implementation time



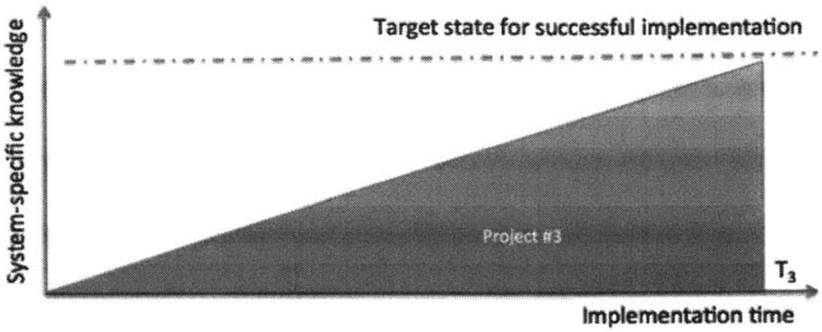
Scenario 2: Low carryover from one project to the next means there are few opportunities where previous work can be leveraged. System-specific knowledge has not been transferred requiring a " fresh start" for each project.

Figure 19: Scenario 2 - Low carryover among project delivers longer implementation time



Scenario 3: A technology strategy that chooses not to select low-risk projects and instead attempts to complete a large, high-utility project at once. In this scenario, risk of project failure with respect to the Standish Group criteria of time, budget, and scope is highest.

Figure 20: Scenario 3 - Longer implementation time associated with big projects



Implementation time is inversely proportional to the carryover that exists in the sequence. In notational form:

$$\text{Carryover}_{\text{Scenario 1}} > \text{Carryover}_{\text{Scenario 2}} > \text{Carryover}_{\text{Scenario 3}}$$

There is risk that a project selection strategy that optimizes for carryover underperforms an approach in which the organization attempts a large and high-value project initially. If the carryover from project to project is small, then time and resources are being consumed to implement technology that does not support higher impact investment areas. This may unfold because the degree to which one project can carryover to another is often known only in hindsight.

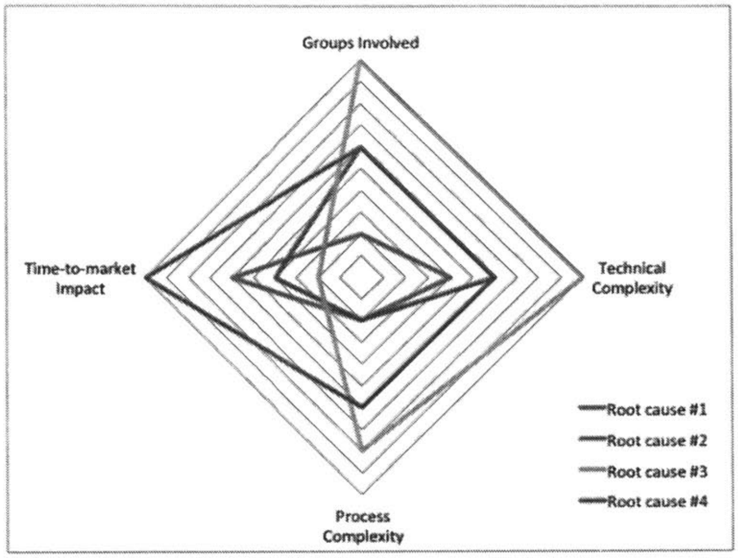
As was the case at Verizon, the enterprise should conduct a root cause analysis of the time-to-market delays resulting from their product development process. The root causes can be organized by their contribution to (1) time-to-market delays, (2) process complexity, (3) technical complexity, and (4) the number of groups that participate in the activity. The project selection process should look to de-risk big projects by identifying smaller projects that will increase system-specific knowledge and contribute carryover to higher impact project areas. A cross-functional team can create a scorecard like the one in figure 22 to help evaluate each root cause according to the four dimensions listed above. The root causes are scored on a scale of 1 to 5 indicating their intensity for a particular attribute. For example, "root cause #3" scores a 5 for "Groups involved" because it has the highest number of people and organizational groups involved contributing to the time-to-market delays.

Figure 22: Root cause scorecard

	Groups Involved	Technical Complexity	Process Complexity	Time-to-market Impact
Root cause #1	1	2	1	3
Root cause #2	3	3	1	2
Root cause #3	5	5	4	1
Root cause #4	3	3	3	5
......				

It may be helpful to plot the root cause scoreboard from figure 22 onto a radar map like the one in figure 23. The radar map visually indicates the relative complexity that should be expected with addressing each root cause relative to one another. For example, root cause #3 was scored as high intensity with respect to the number of groups involved, the technical complexity of the process, and the time-to-market impact. However, root cause #1, as evident by the smaller surface area in the radar map, can be viewed as a fairly manageable area for improvement by way of a software project.

Figure 23: Radar map for complexity assessment



When viewed as a radar chart, the area formed by each project is a heuristic for the challenge that should be expected to implement the project on time, in budget, and in scope relative to plans and expectations. Process complexity risk is the factor that can be managed by a project selection method that emphasizes carryover from one project to the next. A skilled IT team that has completed a number of requirements gathering processes for small software projects may have the strongest understanding of the end-to-end product development process. The documentation that is produced to inform technology selection should be shared with senior management to help strengthen their understanding of the operation and identify departures from the intended strategy.

Chapter 5

5. Conclusions and Future Work

The software solutions discussed addressed the root causes associated with miscommunication in a large, new, and complex product development organization. Two other categories, enterprise architecture and design expertise, were identified in the root cause analysis as significant drivers of time-to-market delays. While there is significant opportunity to improve the transparency and speed of the product specification process with enterprise software, it represents first and foremost an enterprise architecture challenge. Inclusion of the manufacturing and supply chain functions will need to be initiated by the design team, the current owners of the process. There may be opportunities to rotate a small number of individuals through the critical functions in the product development process and look to them to streamline the cross-functional process.

The creation of the process maps for the development of the Product A and Product B highlighted missing expertise and underdeveloped process controls. There is a large need at Verizon for more hardware engineering expertise to independently test and validate the work done by the 3rd party design partner and the contract manufacturer. Currently, there is a heavy reliance on the network testing process to serve in a design validation capacity. Network testing occurs late in the product development lifecycle and failures can result in expensive, highly-constrained, and extremely time consuming workarounds. Catching design flaws while still in the prototype stage will result in significant time-to-market improvements.

These recommendations were developed while working on the implementation of the product collaboration software (software project #2) and shared with Verizon senior management. While the objective of IT is to support critical business functions, byproducts of traditional software implementation work are insights that might normally originate from outside consultants or an internal audit team. While software may directly address only a small subset of process improvements, activities like process mapping, stakeholder interviews, and error logging are valuable in making almost any organizational decision.

An organization should employ a healthy skepticism that it has the resources and system-specific knowledge to tackle the big problems early on. The data suggests that implementation failure rates increase with the ambition of the project and that corporate IT has consistently failed to meet the expectations of senior leadership. These problems are particularly acute in product development organizations where the speed of innovation is often the differentiator between competitive and languishing enterprises.

Bibliography

BCG. (2012, February 31). *The Most Innovative Companies 2012.* Retrieved February 1, 2015, from The Boston Consulting Group: http://www.bcg.com/documents/file125210.pdf

Blanchard, B. a. (1998). *Systems Engineering & Analysis.* Prentice Hall.

Bloch, M. a. (2012 , October). *Delivering large-scale IT projects on time, on budget, and on value.* Retrieved Janurary 4, 2015, from McKinsey & Company: http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value

Chen J., D. F. (2009). Understanding antecedents of new product development speed: A meta-analysis. *Journal of Operations Management*, 17.

Folgo, E. J. (2008). *Accelerating Time-to-market in the Global Electronics Industry.* DSpace@MIT. MIT.

Griffin, A. (1997, 11). PDMA Research on New Product Development Practices: Updating Trends and Benchmarking Best Practices. *Journal of Product Innovation Management.*

Kern, E. a. (2007). Framework For Internet-Supported Inter-Organizational Product Development Collaboration. *Journal Of Enterprise Information Management 20.5*, 562-577.

Lunden, I. (2013, July 15). *Forrester: $2.1 Trillion Will Go Into IT Spend In 2013.* Retrieved February 18, 2015, from TechCrunch: http://techcrunch.com/2013/07/15/forrester-2-1-trillion-will-go-into-it-spend-in-2013-apps-and-the-u-s-lead-the-charge/

Maria, M. (2008). *Supply Chain Impact of Implementing a Forecast Driven Demand Pull Methodology.* Cambridge: MIT LGO Thesis.

Silva, C. C. (2014). A Framework to Evaluate the Impact of ICT Usage on Collaborative Product Development Performance. *25th Australasian Conference on Information Systems*, (p. 34). Auckland.

The Standish Group International. (2013). *CHAOS Manifesto 2013. Think Big, Act Small*. Retrieved 1 15, 2015, from CHAOS Report - The Standish Group: http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf