2014

Tried and True - but Late! Using Lean Six-Sigma Problem Solving to Analyze ERP Software Change Management Performance

Kathleen DeBruyn
Grand Valley State University

Tried and True - but Late!
Using Lean Six-Sigma Problem Solving to Analyze ERP Software Change Management Performance

By
Kathleen DeBruyn

Tried and True – but Late!
Using Lean Six-Sigma Problem Solving to Analyze ERP Software Change Management Performance

By
Kathleen DeBruyn

A project submitted in partial fulfillment of the requirements for the degree of

Master of Science in

Computer Information Systems

at

Grand Valley State University

December, 2014

Gregory Schymik **December 6, 2014**

Table of Contents

Table of Figures

List of Tables

Appendices

Abstract

One of the most common challenges that businesses face today is the constant pressure of global competition. Technology is the tool of choice for executing excellence in the marketplace. As a result, the dilemma of needing to deliver software change at an accelerated pace is extremely normal and many organization struggle to identify what approach will work best for them.

At the same time, it is not unusual for humans to jump to conclusions before thoroughly examining a problem and taking in all available facts. After viewing a few symptoms, conclusions are made, action plans are formed and then wonderment arrives when the problem remains unsolved. The intention of this project was to partner with a large organization immersed in this struggle and use Lean Six Sigma as the mechanism for learning why their software delivery is tried and true – but late.

The goal of this specific project is to discover and define performance requirements and associated metrics by employing a problem solving method commonly used in Six Sigma initiatives referred to as the DMAIC method. The result of this research project was a surprising mix of outcomes that speak to the need to balance culture, process, and architecture.

- The project client did not have any internal service level agreements. When outlining the factors critical to quality, they were confused and struggled to define quality of process. The idea of learning how to improve software change management through the use of metrics was new for them and produced some fear mixed with hopeful excitement.
- The historical data proved incredibly difficult to acquire and convert into usable form. This was due to the architectural choices made when the software change management system was created. A short-term focus on an incomplete set of use cases rendered the system forever hostile for reporting and analysis.
- The content of the historical data was minimal in value. There was no data to use for comparing due dates with delivery dates. Effort was not being tracked. The only means of measuring work activities was to measure the time duration within work flow steps.
- The strength of DMAIC is the discipline of following the data. This helps teams avoid pursuing erroneous leads or tempting short-cuts that ultimately leave the problem unresolved.

Despite obstacles, the goal of defining software performance metrics was met. This was due to the fact that DMAIC works well as a tool for learning because it is a disciplined approach with a wide array of tools that also serve a need for flexibility under certain conditions.

Introduction

There is a prevailing opinion among business leaders at Cozy Couches[i] that it takes too much time for IT to implement software projects. A recent project history shows that this concern is well founded (Table 1). Each of the projects were initialized with proper planning and estimations provided by experienced and knowledgeable IT professionals for the areas being impacted. Yet, as found across numerous organizations, the actual implementations took longer than expected. The data in the table also illustrates that this is not a new dilemma. The concern extends over a decade into the past.

Project	Project Description	Resources Required (Size)	Original Implementation Target	Actual Implementation Reached
Project 1	Implement Baan IV c3 ERP with significant consultant support and large internal project teams. Scope includes all business systems on Mainframe.	Extra Large	Fall 1998	2008
Project 2	Implement a global application for financial consolidation	Medium	Fall 1998	Spring 2001
Project 3	Connect Baan Product Costing to Mainframe Sales to eliminate 1M+ dollars of monthly unexplainable variances in product margin performance	Small	Summer 2005	Spring 2006
Project 4	Implement the dynamic creation of routings and bills of material for product as it is actually ordered by a customer	Large	2009	2014
Project 5	Implement a system capability for customers to order sets of products related by a parenting product structure	Large	2010	Not yet started
Project 6	Modernize the Customer Contracts system	Large	Spring 2011	Spring 2012
Project 7	Implement Infor LN ERP with minimal consultant support and large internal project teams. Scope includes interfacing other systems for the entire order fulfillment cycle including cash receipts from customers	Extra Large	Fall 2013	Fall 2014
Project 8	A complete platform switch off IBM while retaining COBOL code approach	Extra Large	March 2013	August 2013

Table 1 - Project History

[i] Not the company's real name

In fact, for the past 20 years, this company has had **no** service level agreements for the delivery of internal software. As such, there had been no system to track performance. This resulted in a dilemma requiring urgent attention when legislation was passed for the Sarbanes Oxley Act of 2002. That legislation contained a requirement for public companies to provide auditors with evidence of adhering to their internal software change management processes.

The only area that had an application to manage software change was the Baan ERP team. The Baan ERP system did not provide functionality that allowed the IT staff to track the timing of software migrations being promoted across environments. To address this need, they implemented an ETQ work flow shortly after the first manufacturing site went live in 1998. This changed when the company was required to comply with SOX 404 of the Sarbanes Oxley Act (Sarbanes-Oxley Act of 2002). The ETQ system was replaced by a new application that was created on the Lotus Notes IBM platform. This system has been internally referred to as the "SCR System". All platforms of application development were required to use this new system to support **all** software change in the organization.

Fast forward to 2014 and the business leaders are asserting concern that the new multi-year ERP project for LN needs to be completed faster. The goal of the Baan implementation originally included all systems on Mainframe but the project ran over its allotted time and scope was significantly trimmed. LN ERP is the updated version of Baan and the new project picks up where the Baan ERP implementation had stopped. Several other business systems are integrated with both Baan and with LN ERP due to its foundational capabilities for normal business operations. This interdependency results in a situation where poorly managed changes can cause delays for non-ERP projects. Ultimately, the pace and quality of change for ERP sets the pace of change for integrated systems.

There has never been a project to research and evaluate the software change management process at Cozy Couches. The goal of this specific project is to discover and define performance requirements and associated metrics by employing a problem solving method commonly used in Six Sigma initiatives referred to as the DMAIC method. DMAIC is an acronym for Define, Measure, Analyze, Improve and Control. DMAIC is a Six Sigma problem solving method that is a data-driven improvement cycle used to refine, optimize and build reliability for business processes (Six Sigma DMAIC Roadmap, n.d.). If we can

determine the drivers that impede or enhance the ERP software change management process, we can seek improvements that would indirectly improve the overall delivery for software projects at Cozy Couches.

Methods/Procedure

DMAIC is a method that can be as light or intense as deemed appropriate for the situation scope and complexity. It is a problem solving method that requires a disciplined approach to follow the data and avoid leaping to conclusions based on intuition or instinct. The method was selected due to the challenge it would present within a company culture that promotes flexibility and creativity as being more valuable than stability and control. With decades of business success derived from being creative and offering break-through solutions, Cozy Couches has a Collaborate (Clan) Culture as described in Figure 1. They value teamwork and cohesion more than results and efficiency. DMAIC relies on teamwork *alongside* a structured and disciplined approach for problem solving that emphasizes following the facts as presented by the data. For a team of professionals who have historically trusted intuition equally with facts, attempting to introduce a data-driven improvement initiative would be a daunting challenge.

DEFINE

The Define phase involves building a plan for the improvement project by producing a project charter with the problem statement, goal statement, and formation of a small team of individuals with a direct role in the ERP software change management process. This included a business analyst, systems analyst, software developer, IT Manager, and system administrator. Many of the tasks within the Define phase are common to traditional forms of project management. The Define phase also requires the completion of a high-level process map, defining the customers of the ERP system, and identifying the customer requirements for quality.

Problem Statement

The problem statement from the project charter centers on the notion that the rate of delivery for software change is not satisfactory. It does not specifically identify the ERP software as being the concern, but it does acknowledge the pivotal role that ERP has as the central hub of system interfaces critical for core business functions.

> *"The ERP systems have interfaces with all other business application systems. The full process of receiving a sales order through to applying customer payments will occur within the Infor ERP LN system once the project team completes the implementation on October 13, 2014. Because this system provides foundational and critical capabilities for the organization, poor management of the software change process negatively impacts all other systems and related projects.* ***The rate of software change desired by business leaders is not being met. The business environment drivers that generate the request for software change are not predictable."***

Goal Statement

The goal statement from the project charter explicitly states that the deliverable is a set of performance requirements and associated metrics.

> *Because Cozy Couches has never formally measured any software change management process before,* ***the goal of this project is to discover and define performance requirements and associated metrics for LN ERP change management****. The strategy to achieve this goal is to employ a problem solving method commonly used in Six Sigma initiatives referred to as the DMAIC method. DMAIC is an abbreviation for Define, Measure, Analyze, Improve and Control. DMAIC is a data-driven improvement cycle used to refine, optimize and build reliability for business processes. The phases for Analyze, Improve and Control will be performed in a follow-up project with the ServiceNow project team. ServiceNow is the new application that will replace the SCR system.*

Scope Statement

The project charter also includes the scope statement that asserts both the legacy and new ERP systems are to be included in the DMAIC. The name of the DMAIC project was established as "SCM for ERP".

> *The scope for the SCM for ERP project will be limited to the software change management process for Baan IV and Infor ERP LN.*

High-Level Process Map

The highest-level process map for ERP software change at Cozy Couches closely mirrors the traditional SDLC model and offers no surprises (Figure 2). Prior to the creation of an SCR or establishment of an IT Project, the business users experience symptoms that they believe can only be solved by changing the software. Some business areas closely scrutinize what requests should be brought to IT by their managers. Other business areas will directly approach an IT Analyst to see if they agree that a software change is needed to solve the problem. Eventually the business partner or analyst for that functional area creates a request within the SCR system.

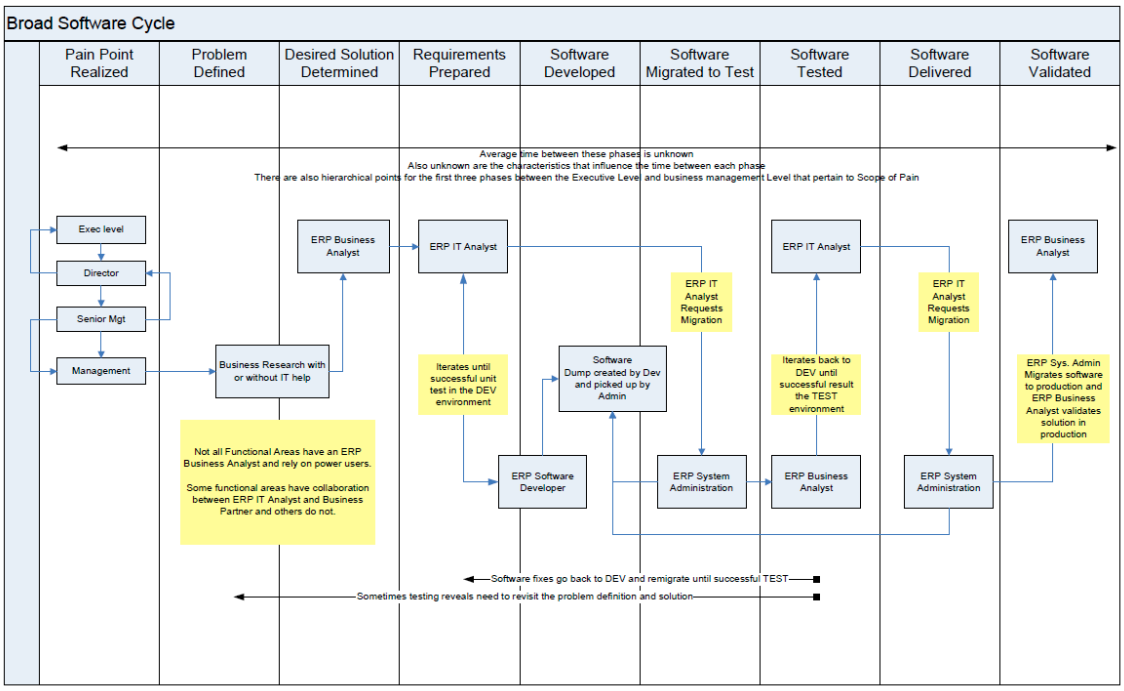


Figure 2 - High Level Process Map for Software Change

The request for a software change must be approved by IT management before any work can be started. Subsequent to the request being approved, the analyst completes the design and sends the request to a software developer. The software developer will send the request back to the analyst when it is ready for migration to the test environment. This overall process applies for all software changes made to the Baan and LN systems.

ERP Software Change Management Process Flow

There are three main process work flows for ERP software change management requests (Figures 3, 4, and 5) based on which ERP system is to be changed. Each of these work flows have a step that requires management approval to do the work followed by the traditional design, code and testing tasks performed with all software change management.

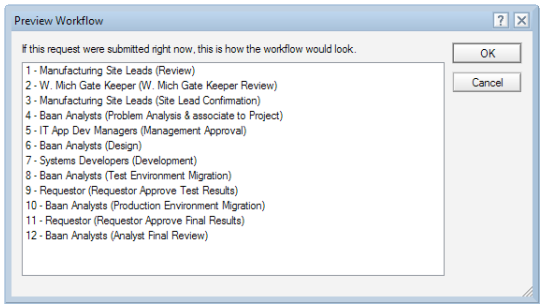


Figure 3 - Manufacturing Baan Software Change Request

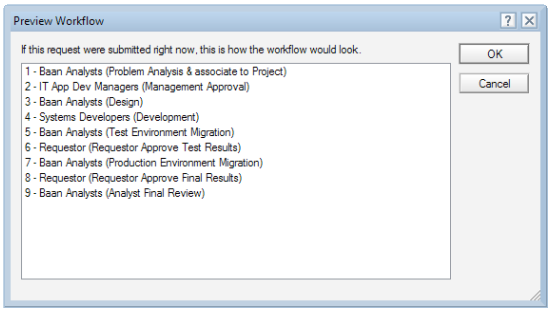


Figure 4 - Baan Software Change Request

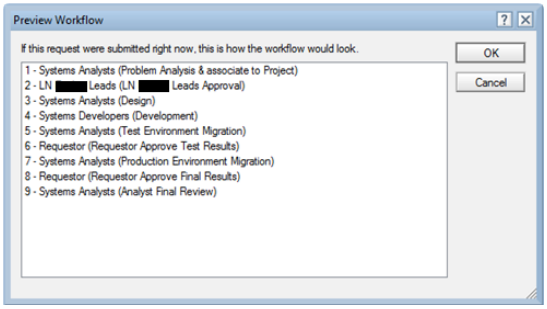


Figure 5 - LN Software Change Request

In the work flow for Manufacturing Baan Software Change Request (Figure 3), plant representatives approve the software change before it reaches an IT Manager for approval.

In the work flow for LN (Figure 5), the management approval is unique. Unlike the other processes in Figure 3 and Figure 4 where an IT Manager approves the work, the LN process work flow approval is **team based**. The leaders of the LN project discuss and approve the software change together in order to align the various sub-project teams that represent all the functional areas. This approach respects the reality that all business teams are impacted by the elimination of functional system silos as practiced in an ERP package.

Process Definition Documentation

Team interviews were conducted to document and validate the SCR processes. The SIPOC diagram (Supplier, Input, Process, Output, and Customer) was selected as the best means to illustrate the meaning within each process flow. This effort uncovered undocumented processes. For example, some of the analysts conduct additional testing in the development environment in order to avoid unnecessary migrations when the software isn't truly ready for business partner testing. This is displayed on the SIPOC diagrams (Figures 12, 13 and 14 in Appendix A) with dotted lines to convey the notion that this is an "invisible process."

All three processes have the same steps once the software design work begins. The analyst provides the design for the software change to the developer. The developer notifies the analyst when the code is ready for testing. The analyst sends a migration task request to the system administration team. The system administration team notifies the analyst when the test environment software has been updated.

Once the software change is tested and approved by the business partner, the analysts sends a task to the system administration team to migrate the software change to the production environment. The business partner gives final approval for the request results before the analyst updates the request status to "complete".

Customers of ERP

Before you can determine customer requirements in the Define phase, you must first identify and define the customer. In Table 2, the customers are described by the role they fulfill with ERP and how they are impacted by the activities of ERP. Customers who are directly involved with the software change management process also provided a representative to participate in DMAIC.

Role	Role Description	Involvement	How they are impacted by ERP
Non-ERP	Have no exposure and might not know what an ERP system is.	None	May work with other systems where initiatives are on hold due to scarcity of IT resources
ERP Users	Exposure to ERP systems is limited to their direct use for performing tasks for the business processes of their functional area	None	May find that their requests for software change to improve their internal processes are placed on hold due to organizational priorities for other areas using ERP
ERP Process Gurus	Provide direction for design requirements	Minor	Have been invited to meetings to help other people understand what functionality they

			need to have available in the system in order to be successful in their jobs.
Power Users	Provide front-line user support and contribute to designing requirements. Might be involved in testing and validation.	Indirect	Actively participate in helping IT with on-going system enhancements by explaining what functionality is needed and by performing testing
ERP Business Analysts	Work full-time on system enhancements or new implementations with IT	Direct	This role is not present for all functional areas. Where it is absent, it is filled by Power Users.
ERP Systems Analysts	Work with specific functional areas to discern functionality requirements. Also works with ERP software developers and ERP System Administrators.	Direct	Shifts in priority can impact their ability to move quickly. Often prefer to work with a specific developer so that shared knowledge can be leveraged for future work. A new requirement to avoid customization and align software design with Infor LN standard software can also impact their ability to deliver results quickly.
ERP Software Developers	Work with ERP System Analysts and ERP System Administrators to deliver solutions that meet the functional needs of the business and perform well in the system.	Direct	Errors in assumptions regarding technical feasibility can disrupt their ability to move quickly. Shifts in priority and long feedback loops can disrupt their ability to move quickly.
ERP System Administrators	Work with ERP System Analysts to migrate ERP software and setup batch jobs. Work with ERP Software Developers to resolve production issues.	Direct	They also support WebSphere and other applications in the UNIX environment. Incomplete or erroneous software dumps can disrupt their ability to migrate software quickly.
ERP Management	Approve requests for ERP software change and contribute to business discussions to define priorities	Direct	They negotiate for resources amongst all areas of the business to support current ERP priorities. Insufficient funding and resources can impact their ability to deliver results quickly.
Other IT impacted by ERP	Supportive IT roles in either infrastructure, applications, or client technologies where ERP co-exists with other systems that they support.	Indirect	Insufficient funding for equipment & training, shifts in priorities and the need to support multiple projects can impact their ability to move quickly.

Table 2 - Business Roles for Supporting ERP

Customer Critical to Quality (CTQ) Factors

While it was easy to define the customers of ERP, we hit a significant road block for the CTQ (critical to quality) measures. There were simply no metrics for defining success. This process had never been formally examined and we recognized the need to establish a baseline.

Prior to starting the DMAIC, the "metric for success" had always been to deliver working software as soon as reasonably possible. User adoption is often quite high due to the role that they play with testing and validation. This involvement gives them insight as to the pacing of progress. Through the use of observation, they often learn where complexity can cause issues. When these observations raise concerns, requirements get adjusted. These adjustments are intended to prevent future issues and represent a sort of "pseudo preventative CTQ" effort. None of this activity is recorded in detail or monitored. Progress is not measured while the work is underway beyond the feelings of progress being made generated from user involvement in testing mentioned above.

The Define phase could not be completed without defining the CTQs. Team discussions were held to determine the CTQs and the definitions for defect, unit and opportunity. This proved to be a time consuming task as the team initially struggled with the concept of measuring the process instead of the software. When they considered the speed bumps that they encounter before they can complete their portion of the overall process, it became much easier and they were satisfied with the results. An example of one CTQ is shown in Figure 6. The entire list of CTQ definitions can be located in Appendix B.

IT Developer as the Customer							
	Supplier	**Supplier Output**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**
E1	IT Analyst	Resulting Software Specification	Receipt of Functional Specification	Clear, complete, and correct documentation of functionality required.	Missing Functionality that User expected to Test	One Session	Each Session packed in the SCR

Figure 6 - Example CTQ

MEASURE

There is a saying that "what matters most gets measured." There's another saying that "what gets measured gets action." The underpinning key concept for both of these sayings is that performance concerns receive top attention and therefore are monitored. Simply stated, we measure what matters most. The purpose of the Measure phase is "To thoroughly understand the current state of the process and collect

reliable data on process speed, quality and costs that you will use to expose the underlying problems." (George, Rowlands, Price, & Maxey, 2005. p. 8) The execution of this phase is not clear cut for the inexperienced practitioner.

> "Perhaps no portion of the DMAIC process is as variable as the Measure phase and its tollgate review. The reason is simple: there is no predefined sequence or set of tools that each team *must* use. Rather, teams must apply their logic and knowledge to create their own path and select tools appropriate to their particular challenges." (George, Lean Six Sigma: Combining Six Sigma Quality with Lean Speed, 2002, p. 174.)

When the effort to define CTQs proved highly problematic, it was evident that defining measures would also be difficult. Cozy Couches did not have a specific performance goal. Their primary concern was to have a tool that provided a historical account of the sequential completion of tasks related to managing software changes. This sequential series of steps to fulfill the process is defined as the "work flow".

In this example, the names have been changed per the agreement with Cozy Couches to provide confidentiality for releasing this information. Note how the work flow is not linear and illustrates one of the LN Processes outlined in the SIPOC where the work has been sent back to the Developer from the IT Analyst.

• Network Impact Question Set To No: Sally Evans (03/09/2012 05:18:56 PM EST),
• Request Submitted: Sally Evans (03/09/2012 05:21:52 PM EST),
• Request Routed On (1 - Systems Analysts): Sally Evans (03/09/2012 05:23:53 PM EST),
• Request Routed On (2 - IT App Dev Managers): Marcus Brown (03/09/2012 05:34:39 PM EST),
• Request Routed On (3 - Systems Analysts): Sally Evans (03/20/2012 01:42:58 PM EDT),
• Request Routed On (4 - Systems Developers): Arthur Andrews (05/18/2012 10:35:45 AM EDT),
• Request Sent Back to 4 - Systems Developers: Sally Evans (07/25/2012 05:25:58 PM EDT),
• Request Routed On (4 - Systems Developers): Arthur Andrews (07/27/2012 11:20:15 AM EDT),
• Request Routed On (5 - Systems Analysts): Sally Evans (08/21/2012 10:17:26 AM EDT),
• Request Routed On (6 - Karla Thomas /Cozy Couches): Karla Thomas (08/21/2012 11:11:41 AM EDT),
• Request Routed On (7 - Systems Analysts): Sally Evans (09/07/2012 04:49:12 PM EDT),
• Request Routed On (8 - Karla Thomas /Cozy Couches): Karla Thomas (09/10/2012 06:45:25 AM EDT),
• Request status set to Complete: Sally Evans (09/10/2012 06:46:26 AM EDT)

Sources of Data

For this study with Cozy Couches, there were few sources of data that could be used that involve software change management. The primary source of data for the Measure phase was the Software Change Request (SCR) system. Secondary sources included emails between individuals involved in managing software change coupled with various forms of project documentation. The third level of information sourcing that was available for this project involved conducting personal interviews and panel discussions.

The SCR system had historical data with disparate starting points for varying software applications from 2004 and is still building data as it is the central system for all software change management at Cozy Couches operations for North America.

Data Collection

The next step in the project was to extract documents from the SCR system. This was a lengthy process (Appendix C) that included both software change requests and software migrations for the Baan ERP system (2009-2014) and the LN ERP system (2012 – 2014). The summary level report was extracted from the Management Summary view. The detail level report is the only means for collecting the actual work flow steps where the timestamp is noted for each step of exchange. Migration tasks for LN and Baan implementations were extracted and linked to the source SCR. Lastly, observations were collated from a series of Lessons Learned brainstorming discussions with members of the LN ERP implementation team for future use within a cause and effect examination to supplement the SCR system analysis.

Data Format

Before any measuring can take place, the data must be in a state where it can be used for comparison and measurements. This proved to be the most difficult task in the project. The actual steps for generating an extract were generally not difficult. But the number of documents returned from the search for Baan and LN SCRs was too large and ultimately crashed the Notes Application. In response to this, I selected smaller date ranges to produce multiple output files. There was one year of Baan documents that had a corrupted value and would cause the extract to crash. Through trial and error, the offending date was located in the data for year 2010 and resolved by individually extracting 36 Baan SCRs.

As described in detail in Appendix C, there are only two extract options: comma separated values and structured text. The comma separated value format only includes fields available from a view. The view only has the last work flow step and is insufficient. The structured text format is the only method that will extract work flow information and unfortunately that format is a vertical list of document fields for each document within the extract.

The document fields include rich text fields where the rows wrap and can result in a variable number of rows output for each SCR. Compounding this challenge was the fact that many of the document fields had blank values or hidden characters used as break fields within the Notes program logic for custom

functionality. These appeared randomly within a series of document fields in the vertical list. The structured text also used an unreadable field to separate the SCRs. I had hoped that I could use a specific document field to indicate that the next field belonged to the next SCR. Unfortunately, the document field that precedes or follows the break field row also varied. This vertical structure proved to be a serious issue that had to be resolved before the project could continue.

To address the data structure issue, a software developer from the Notes team was asked to create a tabular extract that was supposed to include the work flow steps. Unfortunately, this output was found to be flawed during the initial review. I found a listing of SCRs that showed that I had open SCRs with a business analyst that I have not worked with for four years! Unfortunately the extraction did not allow for a variable number of work flow steps and thus the entire data set had been corrupted. The developer did not know how to correct the issue. There were no available options for a custom extract. Because of all the issues with random hidden character sets coupled with hidden character break values peppered throughout the file, I was left with no option other than manually formatting the vertical list output file in Microsoft Excel.

Converting the structured text output into tabular format also proved to be extremely cumbersome and time consuming. Numerous times the data became corrupted during the course of refinement and the process had to start over from the backup copy of the original vertical list output file. In the course of dealing with obstacles, I learned a great deal about what document fields had data values and which ones did not. After a series of more than six iterations, a repeatable manual approach was designed and determined reliable enough to accurately convert the vertical structure to the horizontal rows of SCRs (see Appendix E for the detailed formatting procedure). At a later point in time, more data was added from extracting the SCR migration tasks to show the number of migrations and the last date of migration.

Once a table of historical data concerning SCRs was ready for use, the next step was to decide what we should to measure. The logical starting point for this decision involved the CTQs developed in the Define phase.

ANALYZE

A review of the alignment between the CTQs collected during the Define phase and the SCR fields was performed in order to identify appropriate sources of data from the extracted set (Appendix F).

The initial review indicated that the data set might line up well and were color coded where green represented a potential candidate for specific CTQs (Table 3).

IT Analyst as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric**	**What can be measured from SCRs?**	**Fields from SCR**
C4	SCR updated with the management approval in timely manner.	Response time from receipt of request	Approval Date is within size threshold	The approval step date is visible and can be compared to the date submitted.	Work flow approval date
C5	SCR sent to IT Analyst.	Rationale for response is sent at time of decision	Rationale for rejection is provided within the SCR	There is a comment section that the IT Manager can use or they can update the rich text field to explain why the SCR was approved or not. They are not using this field.	Status that shows the work was not taken forward
C9	Timeliness of migration	Completion within time parameters agreed upon	Compare Request Date and Completed Date	The requested migration date and the completion date	Migration task due date migration task completion date
C10	Zero defects in migration.	Number of re-migration due to failure from IT System Admin	Sysadmin enters a note reporting the error and corrective action	A re-migration can occur using the same initial migration task and then marked as complete when the migration is stable.	New migration task could be requested, but isn't required.
IT System Administrator as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric?**	**What can be measured from SCRs?**	**Fields from SCR**
F1	Migration has been requested	Accurate packaging of component within the software dump	Request was or was not sent to Sysadmin to restore the original component.	List of components in the SCR migration tasks versus the actual components in the software dump	SCR Task SysAdmin Notes is available for this, but they are not using it.
F2	Interdependent components are migrated in the same sequential order of creation or maintenance	Only new components are packaged within the software dump	Request was or was not sent to Sysadmin to restore the original component.	List of components in the SCR migration tasks versus the actual components in the software dump	SCR Task SysAdmin Notes is available for this, but they are not using it.
F4	Request was placed in time for the Sunday Maintenance Window Meeting	SCR migration requests are sent on time to allow for effort estimation and prioritization	Compare Request Date against the Maintenance Window Dates	Comparing the migration task due date to the weekly review board meeting date. SCR doesn't show date that the migration request was sent to Sysadmin prior to the weekly review board meeting.	Due Date
F5	Change since last migration field not properly filled out	Change since last migration field accurately filled out providing IT	Sysadmin enters a note reporting the error and corrective action	The information in the changes since last migration to test should align with the tasks required to get success confirmed. Sysadmin has a place to note when this is in error but they are not using it.	SCR Task SysAdmin Notes is available for this, but they are not using it.

Table 3 - Candidate CTQs for Measures

This, however, proved to be an incorrect conclusion. The red text in Table 3 describes the subsequent discovery for each CTQ. Inconsistent use, limited enforcement regarding data entry, and the fact that the SCRs were not designed to allow for the measurement of the process resulted in a dataset that provided limited visibility into the process. This led to too many assumptions being made of the data to make any truly valid conclusions about the process. Examples of problems with the data include the fact that, out of 3,220 SCRs, only three recorded the Actual IT Effort and only four SCRs had a link to a project name. These fields are only helpful if all SCRs indicate the actual effort or source of the request. For example, knowing whether the SCR is for a specific project, to repair a production issue, to prevent future issues, or to enhance the current software functionality (data that could be entered in the project name field) would help us evaluate what ratio of overall capacity was directed towards low value or high value investment.

Consistent versus Inconsistent Usage

Missing data is but one problem. Another problem is when there are data values present that not consistent with the design purpose for the data field. The data becomes diluted when there is inconsistent usage. Decisions made from analysis that was derived from such data are often poor decisions.

Decisions made from an incorrect interpretation of precise data is also a problem. For example, although we can see the date of the last software migration and the SCR completion date, we are not able to accurately assess if delivery met the customer expectation because only 20 SCRs had a requested due date. Estimated IT Effort was a little better than these other data elements, but the lack of consistent usage significantly reduced the ability to perform historical performance tracking. Without such complete and consistent data, a proper assessment of overall process performance was essentially impossible.

The initial finding from the data extraction prior to scrubbing was that twelve fields were reliably populated. These fields (Table 4), could be used to measure subcategories of activity duration, SCR actual size, analyst preferred work style, business teams and source system.

Field	Description
RequestID:	Unique Identifier. The first four characters represent the Name of the person who created the SCR request. This value is concatenated with another random value. The first four characters allow us to map the Analyst preference for water fall or intentional
WorkflowStatus:	This value can be either IP (In Process), C (Complete), or another character value that indicates that the SCR has been cancelled, withdrawn, or rejected.
CreatedDate:	The date that the SCR was created

SubmittedDate:	The date that the SCR was submitted into the work flow
ClosedDate:	The date that the SCR was physically marked as being completed.
EditCount:	The number of times the rich text content was modified.
RequestTitle:	A somewhat short description of the work being requested
RequestTarget:	The system where the software change is being requested
RequestType:	This can be either a "data change" or a "software change". The data extraction for this project did not include SCRs for production data repairs.
RequestorName:	The individual placing the request. This is also the person who must approve test results for migration to the production environment. They are also responsible for final approval for the production result.
SteeringTeam:	A business team that would like to have a view of all the SCRs for their area so that they can track how many are yet to be completed.
WFEditLog:	This field is an array that stores the work flow steps with a timestamp for each instance when the SCR has moved from one person to another person. The difference in time equates to a duration of the process step.

Table 4 - SCR Usable Fields

This new analysis revealed that the only reliably populated data elements from Table 4 that would fit with the CTQs were the approval date and the software migration completion date. This would not be enough data to use for a proper analysis. Given the sad reality of the data, I made a decision to interrogate the data that was available in order to assess if current conditions regarding performance could be inferred. The detailed results of this analysis can be viewed in Appendix G and is summarized in the following.

Work in Progress

The first analysis was to look at how Cozy Couches managed work in progress. Work in progress is calculated as (existing open SCR requests) + (new SCR requests) – (completed SCR requests). If you complete more requests than there are new requests entering the queue, then work in progress will reduce. If new requests are coming in faster than they are being completed, then work in progress will increase. The latter condition is a signal that there is an issue in managing software change that is causing longer cycle times.

The results clearly showed how the software change requests for Baan, the legacy ERP system, were declining alongside a sharp uptick of software change requests for LN, the new ERP system. The business conditions in 2009 involved the start of a recession that led to all employees taking a pay cut with no working on Friday. These "furlough Fridays" required that no work be performed on Fridays. In 2011, the working hours were restored. Shortly after this, corporate leadership at Cozy Couches recognized the

risk of avoiding IT investment during the tough economic cycle. The LN project was assessed and approved to proceed.

The project to implement LN was started in the fall of 2011 with the first go-live (deployment), for the general ledger, slated for 2012. Following the Indirect Accounts Payable go-live in October of 2013, the test environment of LN was released to all the other project teams who were striving for a July 2014 go-live. These teams included Order Management, Inventory, Logistics, Accounts Receivable, and Cash Management. Other teams with interfaces included Order Manager, Manufacturing, and Data Warehousing. The July, 2014 target date was eventually changed to October, 2014.

Table 5 indicates that the backlog of SCRs in 2009 that were In Progress (IP) was in good shape. The recession resulted in a buildup that was relieved in 2011 when full hours were restored and a clean-up effort to prepare for the new project was completed. As the new project rolled out, backlog started to build again in 2012 for both Baan and LN and is clearly evident in the ERP Work in Progress charts (Figure 7) for 2013 and 2014.

	Status	2009	2010	2011	2012	2013	2014
Baan	Prev. IP	136	222	373	185	281	381
	New	490	605	539	424	314	193
	Completed	404	454	727	328	214	139
	New IP	222	373	185	281	381	435
LN	Prev. IP	**0**	0	0	0	21	49
	New	0	0	0	47	132	145
	Completed	0	0	0	26	104	147
	New IP	0	0	0	21	49	47
Total	Prev. IP	136	222	373	185	302	430
	New	490	605	539	471	446	338
	Completed	404	454	727	354	318	286
	New IP	222	373	185	302	430	482

Table 5 - Total SCRs in Progress (IP)

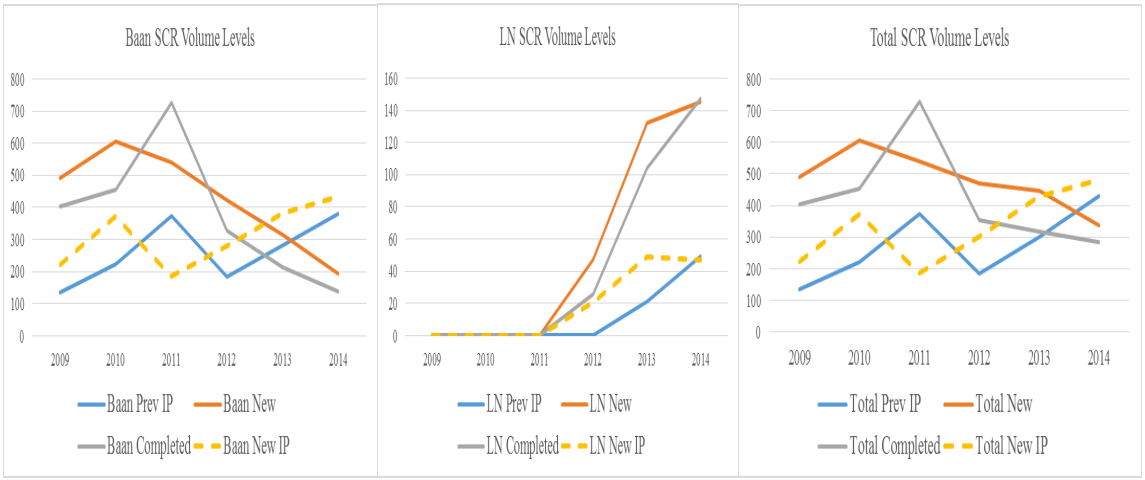


Figure 7 - ERP Work in Progress

Parsing SCR Duration

The next analysis was to review overall duration of SCRs. If we narrow the data to only include SCRs that have been completed, we can view the distribution of SCRs in search of a pattern to show the overall distribution of SCRs per range of duration (Table 6).

Total Days of Duration Range Placement			**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**Total**
<25		25	264	344	281	221	198	99	1407
>25	<=50	50	72	67	73	75	62	61	410
>50	<=75	75	34	25	15	17	21	23	135
>75	<=100	100	23	23	26	13	12	9	106
>100	<=125	125	13	15	13	10	14	11	76
>125	<=150	150	9	15	16	3	6	10	59
>150	<=175	175	5	10	10	5	3	5	38
>175	<=200	200	4	7	7	3	6	3	30
>200	<=225	225	1	11	12	3	5	1	33
>225	<=250	250	0	5	16	1	7	0	29
>250	<=275	275	2	8	11	0	5	0	26
>275	<=300	300	0	5	19	0	3	0	27
>300	<=325	325	1	8	15	0	1	0	25
>325	<=350	350	0	7	3	0	2	0	12
>350	<=375	375	1	22	1	0	1	0	25
>375	<=400	400	1	15	1	1	1	0	19
>400		>400	57	13	1	19	9	0	99

Table 6 - Counts by Duration

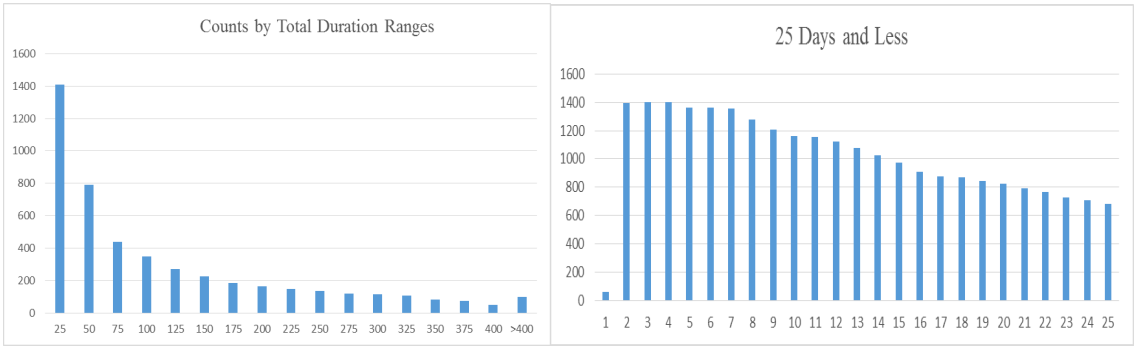


Figure 8 - Distribution of SCR by Range of Duration

Most of the SCRs were completed within 25 days. There is a great variance in duration with a small portion that are completed after 400 days. This is more than one year! In general, this does not seem to indicate a problem although we do not know what portion of these SCRs were expected to be completed in much less time.

To look at the data more closely, we can separate the valued added work flow process steps of design, develop, and testing from the non-value added steps related to "paperwork" processing to do approvals. Although we still cannot decipher the expected and actual effort, it does allow us to create a subset category of work that involves actual software change separated from managing the approvals for the work (Appendix G – Impact of Non-value Process Steps). In this analysis we refer to the approvals as being non-value added because the work doesn't directly impact the *quality* of the software product. The histogram changes dramatically when we measure the duration from the start of software design through software delivery.

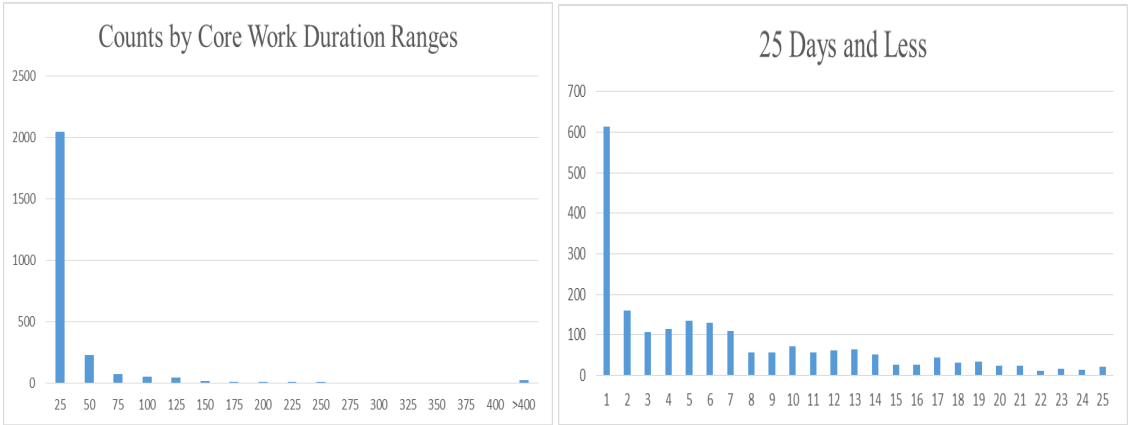


Figure 9 - Distribution of Counts (Software Work Only)

The significant portion of all SCR duration for the core work related to the software product is also 25 days. The SCRs we previously saw in Figure 7 Total Duration, where the duration was greater than 400 days, is considerably reduced. This would imply that these "long living" SCRs have either encountered a technical barrier or have been prioritized to the lowest level.

In Table 6, we can see that although the majority of SCRs are completed in 25 days or less, only 30% of the work is completed within a day. This indicates that many software changes are not fixes where "the good gets in the way of the better." Instead, 70% of the work completed within 25 days is focused on the more difficult and more valuable software changes that move the company forward towards achieving improvements and strategic goals.

Range of Days Duration		SCR Start Year							
From	To	2009	2010	2011	2012	2013	2014	Total	% of Total
<=1		136	161	136	79	70	31	613	30%
>1	<=2	22	48	31	20	24	14	159	8%
>2	<=3	17	27	23	18	12	10	107	5%
>3	<=4	16	26	36	19	9	7	113	6%
>4	<=5	24	29	29	20	22	10	134	7%
>5	<=6	20	27	27	23	19	14	130	6%
>6	<=7	24	26	23	13	15	7	108	5%
>7	<=8	14	9	11	7	8	6	55	3%
>8	<=9	9	16	11	10	6	4	56	3%
>9	<=10	15	14	18	8	11	6	72	4%
>10	<=11	16	8	12	5	12	4	57	3%
>11	<=12	11	13	15	10	7	6	62	3%
>12	<=13	10	16	10	16	3	8	63	3%
>13	<=14	12	9	14	8	3	4	50	2%
>14	<=15	6	7	6	3	4	0	26	1%
>15	<=16	4	9	3	3	2	5	26	1%
>16	<=17	3	15	11	4	6	4	43	2%
>17	<=18	7	5	5	3	7	4	31	2%
>18	<=19	6	6	6	7	4	4	33	2%
>19	<=20	4	4	7	2	5	1	23	1%
>20	<=21	5	5	6	1	0	6	23	1%
>21	<=22	1	3	2	2	3	1	12	1%
>22	<=23	2	3	5	4	1	2	17	1%
>23	<=24	7	2	2	1	2	0	14	1%
>24	<=25	5	4	7	2	2	0	20	1%

Table 7 - Number of SCRs per Days of Duration

Other Analysis

Other analysis was performed to examine if there was any pattern of impact by size (Appendix G – Impact of Size) or impact by a preference to use waterfall or intentional iteration styles (Appendix G – Impact of Style) but the results were not consequential. Samples were taken for SCRs completed from January 2013 to March 2013 to use control charts in search of variation that might indicate insight regarding current performance (Appendix G – Control Charts). Six Sigma seeks to eliminate variation. While it was an interesting experiment to conduct and review, the results were not meaningful.

Analysis examining Kanban using a rolling work in process calculation for a weekly basis showed an overall increase of bottlenecks during the last year of the LN project (Appendix G – Kanban). A great deal of SCRs were delayed in the design phase of SCR completion due to Project 8 in Table 1. Project 8 started after Project 7 and became the top priority. This resulted in a situation where the analysts for Project 7 had to pause their design work while they completed Project 8.

Predictably, a bottleneck of software development and testing arrived as design was completed after this design delay. The one set of data that did align for CTQs was the migration error rate (Appendix G – Migration Errors). Overall, this data looked very good and did not indicate any performance problems.

Ishikawa Diagram

Without detailed measures within the SCR data, we are left with extremely few data based insights for defining the drivers of duration for ERP software change management. Another tool used for the Analyze phase in a DMAIC is the Ishikawa diagram that is more commonly known as a "fishbone diagram". This tool is a cause and effect diagram that is used to spur creative thinking and is highly useful in situations where the data is sparse.

> "The fishbone will help to visually display the many potential causes for a specific problem or effect. It is particularly useful in a group setting and for situations in which little quantitative data is available for analysis." (Simon, n.d.)

With this measurement, the problem question of "Why can't we go faster?" was asked in a series of why statements to decompose ideas identified during brainstorming towards finding a root cause. The team from the LN Project are at the final state of the project and have currently conducted the last set of lessons learned.

The common theme from all the feedback is that they experience disruptions to work flow because of the decision to run multiple project teams at the same time. "Divide and conquer" leads to "Divide and wait". The other big factor was learning how to design a new way of solving problems without always creating software customizations. Customizations are when new software is created to reside alongside the ERP software. Learning takes time and having to obtain a group approval to create software customizations was frustrating because in the past they were used to having no limits on the freedom to create whatever software customization they wanted.

The other common constraint involved having to work within one environment to share amongst all the project team members. However, by bumping into one another, they also learned how their work impacted other areas. That is a key element of finding success with ERP. They simply did not coordinate with one another until integrated functional testing (IFT) was performed. This indicates a lack of coordination for providing a central hub of communication to align their development activities with one another.

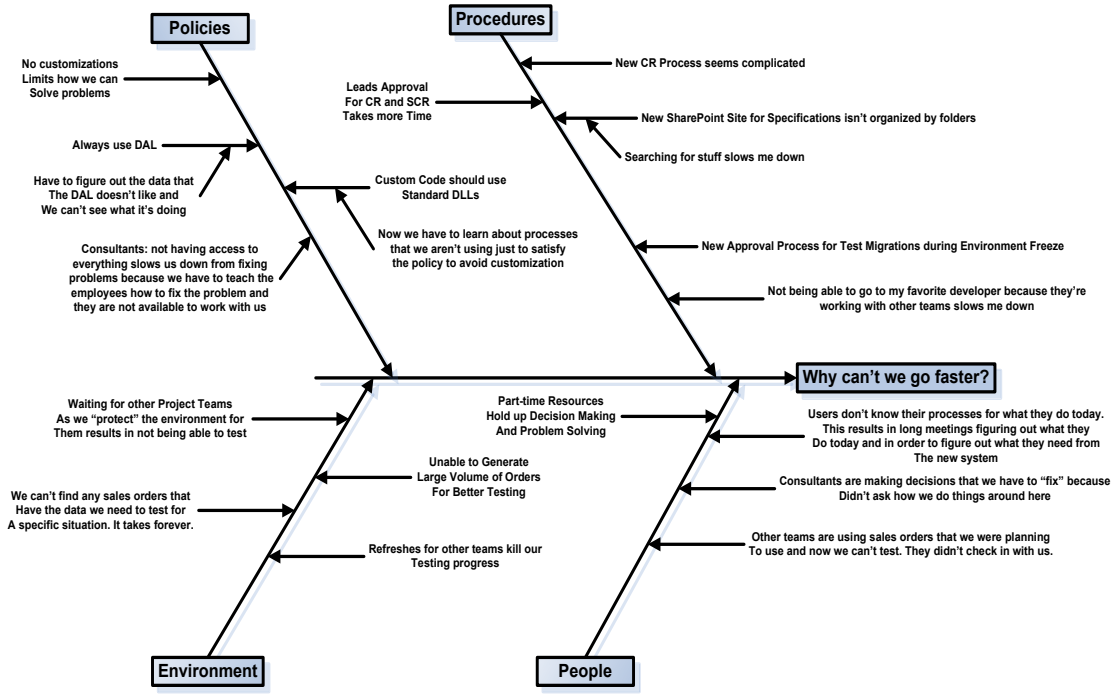


Figure 10 - LN Lessons Learned Series Collected into Ishikawa

IMPROVE

In order for the project to move into the Improve phase, a set of determinations must be selected from the previous Analyze phase. This was not possible due to the lack of data in the system that would show whether or not the software change was actually delivered to the expected date. This means that DMAIC never really left the Define phase and the problem statement is likely a symptom rather than a root issue to resolve.

Without a framework for monitoring delivery of SCRs to support a project, a project manager will have no visibility towards true project performance and will be forced to rely on heroics. Allocating resources towards managing software change will suffer the same lack of visibility towards overall demand. In order to improve, Cozy Couches must learn the current state of performance. The Improve phase will have to wait for a new round of Define and Measure to be completed. A reset is clearly required.

BACK TO DEFINE

Although the barriers for collecting the actual due dates and effort are not clearly known, one could surmise that a lack of managerial concern stemming from high levels of trust within the collaborative culture contributed to the situation. During the discussion to define CTQs, it was often said that "so long as we're working on what's important, that's what really matters." But who decides which work has priority if priority is being not tracked across all software changes? The answer is that the person who drives the SCR to completion does so based on their understanding of what has priority.

All three SIPOC charts showed that the analyst is the one who drives the SCR forward to completion, which means the analyst has been deciding the priority of SCR work. The analysts often fulfill the role of project manager if the software change resides within one functional area. The analyst is also the person who receives support calls during the day and in the middle of the night. Production environment software fixes always have top priority. What we do not know, nor can we glean from the historical SCR data, is whether or not production fixes are crowding out project work.

Knowing that there was no historical data that could help this improvement initiative to measure and analyze the current state of software change management performance, the team was unsure how they should move forward. The Ishikawa diagram was helpful but it was focused on the obstacles to project

momentum. While removing obstacles would help the ERP team move faster, it does not offer a way to learn by tracking on-going performance. The disappointment of reaching a dead-end, coupled with the recent go-live of the LN ERP, resulted in low energy and indecision.

The team needed to return to the Define phase and revisit the problem statement. Is it true that that they need to find a way to go faster? This time, a subset of the DMAIC team used the 5 Why's approach to brainstorm on the problem statement (Appendix H). This approach is not as simple as it sounds at face value because a team lacking experience can easily misdiagnose the root cause. This method has basically four steps as described by iSixSigma (Determine the Root Cause: 5 Whys, n.d.):

1. Write down the specific problem. Writing the issue helps you formalize the problem and describe it completely. It also helps a team focus on the same problem.
2. Ask Why the problem happens and write the answer down below the problem.
3. If the answer you just provided doesn't identify the root cause of the problem that you wrote down in Step 1, ask Why again and write that answer down.
4. Loop back to step 3 until the team is in agreement that the problem's root cause is identified. Again, this may take fewer or more times than five Whys.

After completing this exercise, the team decided to restate the problem as being an inability to predict performance. The key point in their decision was the realization that without any way to report the data in the SCRs, then entering the information would only result in lost data. Reporting performance is the first step towards learning the current condition. Monitoring performance is the second step towards finding the next set of questions that need to be answered in search of more insight regarding the drivers of duration. Once the drivers of duration are revealed, then refinements can be designed to help promote predictability.

As mentioned earlier in the goal statement within the project charter, Cozy Couches was planning to replace the SCR system with a new system. The ServiceNow team has started to implement the transition, but the ERP system would be the last group to move in order to avoid interfering with the focus on fulfilling the implementation date. Because there was an opportunity to influence the design within ServiceNow, the DMAIC team decided to use another tool known as Goal Question Metric (Basili, et al., n.d.) in order to define metrics that support learning the drivers of duration for managing software change. This method is similar to the 5-Why tool in that it uses a series of questions and answers to decompose the goal into tasks and metrics necessary to meet the goal.

Goal, Question, Metric (GQM)

There is no knowledge of the current cycle time at Cozy Couches because the SCR system doesn't track the expected delivery date. To be more accurate, the system provided the ability to provide this information, but it was not required that the information be entered into the system. There are many factors that can assist or impede efforts to deliver software change and having the expected delivery date only exposes a defect. The defect does not tell you why the software change was delivered late.

Before we can achieve the goal of improving predictability, we need to learn what factors act as drivers of duration. The DMAIC team held a session to brainstorm on what questions provide answers to support the necessary learning along with the associated metrics and required data elements (Table 8).

Goal: Improve Predictability of Software Delivery		
Question	**Metric**	**New Data Element(s) Needed**
How will we know that we are improving?	WIP over Time	• Accurate SCR status
Do you know how fast you are going today?	WIP	• Accurate SCR status
Do you know when the customer is expecting delivery?	Productivity Failed Deliveries	• Expected Delivery Date
Do you know the magnitude of effort required for building the product?	Productivity	• Gross Capacity Net Capacity Pre-promised Capacity
Do you know what quality concerns the customer has about the product?	Build Defects Product Defects	• Test Cases • Migration Errors
Do you know the organizational priority for the request?	Priority Hit	• IT Value (Revised) Expected Delivery Date • Priority Driver • Reset Priority Reason
Do you know if you have the necessary capacity to do the work?	Productivity	• Estimated Effort
Do you know the hidden barriers to delivery?	Build Inhibitors	• Reason for Delay • Reason for Return
Do you know that you delivered the product on the expected date?	Productivity	• Delivery Date

Table 8 - GQM: What information and metrics do we need for improving predictability?

The output was a set of Software Performance Metrics (Figure 11) that would enable learning and gain insight for performance predictability. Although it is likely that these metrics will be implemented in ServiceNow, it is not absolutely certain at this point in time. Given the fact that the team at Cozy Couches is inexperienced with defining metrics, it is highly probable that there will changes made after attempts to use them provide more learning. They will gain experience with iteration. The need for learning cannot be understated.

DMAIC Software Performance Metrics

- Productivity – Net sum of scoring SCR Delivery Points: Early (+1), On-Time (0) and Late (-1).
- Failed Deliveries – Count of SCRs still in progress that did not deliver to the expected date
- Priority Hit – Count of SCRs with changes made to priority level
- Build Inhibitors – Number and duration of factors disrupting completion of the SCR
- Build Defects – Number of SCRs with greater than 5% Failed Test Cases
- Product Defects – Sum of Migration Errors and Post-Migration Rework SCRs

Figure 11 - Metrics for Improving Delivery

Results

The result of the first three phases of DMAIC was a clear revelation that it simply would not be possible to define performance requirements and associated metrics for ERP using only historical SCR data. The DMAIC team was challenged to maintain momentum but the need to reset provided time for reflection and reassessment. After going back into the Define phase, an additional exercise was conducted to define a goal for improving the delivery of software changes and establish the metrics necessary to support the ability to learn what factors impact the ability to deliver on-time.

The struggle to define the CTQs was instructive to reveal that concerns for quality have never been formally defined at Cozy Couches. The need to learn "how to think" about quality of process, in addition to quality of product, is a new mindset that will require time and practice to acquire maturity. The idea of tracking defects is viewed negatively and stems from the absence of a quality mindset within the culture and norms of the company.

The historical data provided little value other than revealing the incredible impact that design and architectural decisions have upon the ability for extending the value of an application. The struggle to extract and normalize the data into a format that could be analyzed was met with affirmation of frustration among IT managers that they also encountered the same challenge. The data model within the SCR system had the potential to provide a baseline concerning performance for software change management. But the architecture and inability to develop a robust suite of reporting thwarted the ability to do so.

When faced with the reality that the DMAIC could not proceed any further, the team had to pause and reflect on what to do next. Many functional teams at Cozy Couches engage in continuous improvement programs but the IT department, along with other departments providing internal services, has been fully focused on serving their colleagues. Although the learning that occurred with DMAIC did

not result from its intended target, software change management performance, the learning about the Lean Six Sigma mindset was extremely valuable.

The desire to shorten software development cycles is not isolated to business leaders at Cozy Couches. The entire agile methodology is a reflection of the on-going work to improve this capability. Reaching this goal requires more than tools and methods. It requires building a firm foundation of understanding of the underlying issues and barriers to delivery. The discipline of DMAIC provides an excellent way to acquire that insight while avoiding the temptation to pursue short-cuts or false leads.

Conclusions/Discussion

This project started with a goal of using a Six Sigma problem solving approach to address a problem at Cozy Couches: the software delivery cycle needed to be shortened. I learned that although "numbers tell a story", they might not tell the true story. For example, a shortened cycle time leads to favorable work in progress. But that might or might not tell the true story. If the shortened cycle time reduces variation, then the result is an improvement. But, if the shortened cycle time increases variation, then the result is a deterioration of performance.

We must also never confuse precision with accuracy. For example, the SCRs consistently had a valid value for the work flow status. But we do not know if those SCRs that had a status of being in progress were actually put on hold. Without tracking effort, we cannot see if these SCRs are extremely difficult to deliver or if they are merely gathering dust. SCRs that are on hold should not be taken into account when calculating work in progress. By having a valid but incorrect status code, the work in progress metrics can be "precise" but they are not "accurate".

After reflecting on the results of the project, I learned that complaints without facts derived from data are just opinions. Complaints without corrective action are just observations that have not been deemed critical or actionable. There is a hidden risk for Cozy Couches that begs for attention. Success of the past is no guarantee of success for the future. Even highly competent athletes have to seek challenge in order to avoid the pitfalls of complacency. Stewards of continuous improvement recognize that complacency is the enemy of reliability and, therefore produces risk.

In a digital era where global competition has exploded with heavy reliance on technology to grant businesses a leading edge, performance matters. Culture has a huge influence on how people work together

and Cozy Couches values creativity and collaboration to a level where prescriptive processes are viewed as stifling. Concern has been raised about introducing metrics that could be used to measure individual performance. Goals and metrics are not necessarily inhibitors of flexibility. Let us not confuse good observation and tracking with micro-management or lack of trust.

High performing teams work together to help one another reach the next level of excellence (Stoner, 2013). That is true teamwork. With increased visibility to the work that is being completed, teams will have a quick view of how the project is progressing. This allows them to be better positioned to perform as we take on more work. Performance awareness helps us to reduce the uncertainty that comes with completing big projects. Although the result of this DMAIC has not reached its final state, the value of the method is well understood. Therefore, the DMAIC work will continue at Cozy Couches beyond this particular project.

The next step for this project is to bring the list of new data elements needed to support the metrics to the ServiceNow Design Team. It is important that we avoid repeating the past where fields existed for the collection of data but were not properly or consistently used. To support this effort, managers can also assist others with the data collection and status updates. Most important of all, the new reporting of the metrics needs to be developed in ServiceNow and made available to project teams so that they can use these new tools to continue supporting one another on the road to better predictability and higher performance.

There are many tools in the Six Sigma arena and this DMAIC project made use of several tools to build a richer understanding of the true problem. Going forward, this wide array of tools will prove beneficial as the DMAIC team builds skills while acquiring experience as they move forward.

Bibliography

6 Sixma DMAIC. (n.d.). Retrieved from Lean Process: http://www.leanprocess.net/dmaic-process-six-sigma/?hvid=2oSOzf

Anderson, P. (2014, March 6). *Advantages and disadvantages to using the DMAIC methodology*. Retrieved from Inside Business: http://www.insidebusiness360.com/index.php/advantages-and-disadvantages-to-using-the-dmaic-methodology-5241/

Basili, V., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., & Trendowicz, A. (n.d.). *GQM+Strategies – Aligning Business Strategies with Software Measurement*. Retrieved from University of Maryland: https://www.cs.umd.edu/~basili/publications/proceedings/P122.pdf

Bauman, C., De Heck, J., Leonard, E., & Miranda, M. (2011, July 11). *SPC: Basic control charts: theory and construction, sample size, x-bar, r charts, s charts*. Retrieved from Engineering Controls: University of Michigan: https://controls.engin.umich.edu/wiki/index.php/SPC:_Basic_control_charts:_theory_and_construction,_sample_size,_x-bar,_r_charts,_s_charts

Ben, L. (2014, July 3). *Quantifying the Impact of Agile Software Development Practices*. Retrieved from InfoQ: http://www.infoq.com/articles/quantifying-impact-agile?utm_source=infoq&utm_medium=related_content_link&utm_campaign=relatedContent_articles_clk

Berardinelli, C. F. (2012, November). *To DMAIC or Not to DMAIC?* Retrieved from ASQ: http://asq.org/quality-progress/2012/11/back-to-basics/to-dmaic-or-not-to-dmaic.html

Berardinelli, C. F. (n.d.). *A Guide to Control Charts*. Retrieved from i Six Sigma: http://www.isixsigma.com/tools-templates/control-charts/a-guide-to-control-charts/

Brussee, W. (2012). *Statistics for Six Sigma Made Easy!* (2nd ed.). McGraw-Hill.

Determine the Root Cause: 5 Whys. (n.d.). Retrieved from iSixSigma: http://www.isixsigma.com/tools-templates/cause-effect/determine-root-cause-5-whys/

DMAIC. (n.d.). Retrieved from DMAIC Tools: http://www.dmaictools.com/

DMAIC – The 5 Phases of Lean Six Sigma. (n.d.). Retrieved from GoLeanSixSigma: http://www.goleansixsigma.com/dmaic-five-basic-phases-of-lean-six-sigma/

DMAIC Tools: Six Sigma Training Resources. (n.d.). Retrieved from http://www.dmaictools.com/

George, M. L. (2002). *Lean Six Sigma: Combining Six Sigma Quality with Lean Speed.* McGraw-Hill.

George, M. L., Rowlands, D., Price, M., & Maxey, J. (2005). *The Lean Six Sigma Pocket Toolbook.* McGraw-Hill.

Jones, S. H. (n.d.). *To Use DMEDI or to Use DMAIC? That Is the Question*. Retrieved from i Six Sigma: http://www.isixsigma.com/new-to-six-sigma/design-for-six-sigma-dfss/dmedi-or-dmaic/

Jorgensen, M. (2014, August 29). *What We Do and Don't Know about Software Development Effort Estimation*. Retrieved from infoQ: http://www.infoq.com/articles/software-development-effort-estimation

Linders, B. (2014, Nov 19). *Lean Thinking Applied for Organizational Change*. Retrieved from infoQ: http://www.infoq.com/news/2014/11/lean-thinking-change

McMahon, P. E. (2014). *15 Fundamentals for Higher Performance in Software Development.* PEM Systems.

Peeters, S. (2013, December 5). *Applying Lean Thinking to Software Development*. Retrieved from InfoQ: http://www.infoq.com/articles/applying-lean-thinking-to-software-development?utm_source=infoq&utm_medium=related_content_link&utm_campaign=relatedContent_articles_clk

Peterson, D. (2009). *What is Kanban?* Retrieved from Kanban Blog: http://kanbanblog.com/explained/

Sarbanes-Oxley Act of 2002, Pub. L. No. 107-204, § 404, 116 Stat. 745 (2002).

Simon, K. (n.d.). *Customer CTQs – Defining Defect, Unit and Opportunity*. Retrieved from i Six Sigma: http://www.isixsigma.com/new-to-six-sigma/sigma-level/customer-ctqs-defining-defect-unit-and-opportunity/

Simon, K. (n.d.). *The Cause and Effect (a.k.a. Fishbone) Diagram*. Retrieved from i Six Sigma: http://www.isixsigma.com/tools-templates/cause-effect/cause-and-effect-aka-fishbone-diagram/

SIPOC Suppliers-Inputs-Process-Outputs-Customers. (n.d.). Retrieved from Six Sigma Material: http://www.six-sigma-material.com/SIPOC.html

Six Sigma DMAIC Roadmap. (n.d.). Retrieved from i Six Sigma: http://www.isixsigma.com/new-to-six-sigma/dmaic/six-sigma-dmaic-roadmap/

Stoner, J. L. (2013, 15 April). *The 6 Benchmarks of High Performance Teams*. Retrieved from Seapoint Center for Collaborative Leadership: http://seapointcenter.com/benchmarks-of-team-excellence/

Tharp, B. (n.d.). *Four Organizational Culture Types*. Retrieved from Academia: https://www.academia.edu/1588713/Four_Organizational_Culture_Types

Valuestream Mapping. (n.d.). Retrieved from i Six Sigma: http://www.isixsigma.com/tools-templates/value-stream-mapping/

Appendix A - SIPOC

When we examine the roles within each process, we can immediately see the customers earlier defined in Table 2 as having a direct role in a segment of the ERP software change request. Each person involved in the process is both the customer and supplier at times. The output from a supplier is the complete work with value added by the step that has been completed. For all three processes, the roles have been color coded to make it visually easier to compare the processes.

The SIPOC for a Manufacturing Baan Software Change Request has several initial steps involving approval from site leads. A site lead is an individual who speaks for the needs of one or more manufacturing sites when working with IT. The intention of having these approvals up front was to avoid work cancellation due to ideas not being fully vetted across sites. It was quickly discovered that the additional steps were clumsy and resulted in redundant approvals as each site lead also was involved in testing and approving test results.

In Figure 14, we see the role of the LN Leads Team in granting approval for the custom work in the SIPOC for a LN Software Change Request. What we don't see is another outside process where a customization request is posted onto the LN Project SharePoint site in a list with work flow for doing the approval and tracking the work in a duplicate fashion. This is invisible to the SCR system. I decided not to add this invisible process to the SIPOC because it is merely duplicate tracking of the work in another system. The only "connection" between SharePoint and the SCR system is that many analysts started inserting hyperlinks for the SharePoint customization request instead of attaching or outlining the requirements within the SCR itself.

In all three SIPOCs, we also see a loop of rework where the testing reveals a software bug or missing functional requirements. However, in short, there really aren't any significant differences between the three processes outside of the approval portion.

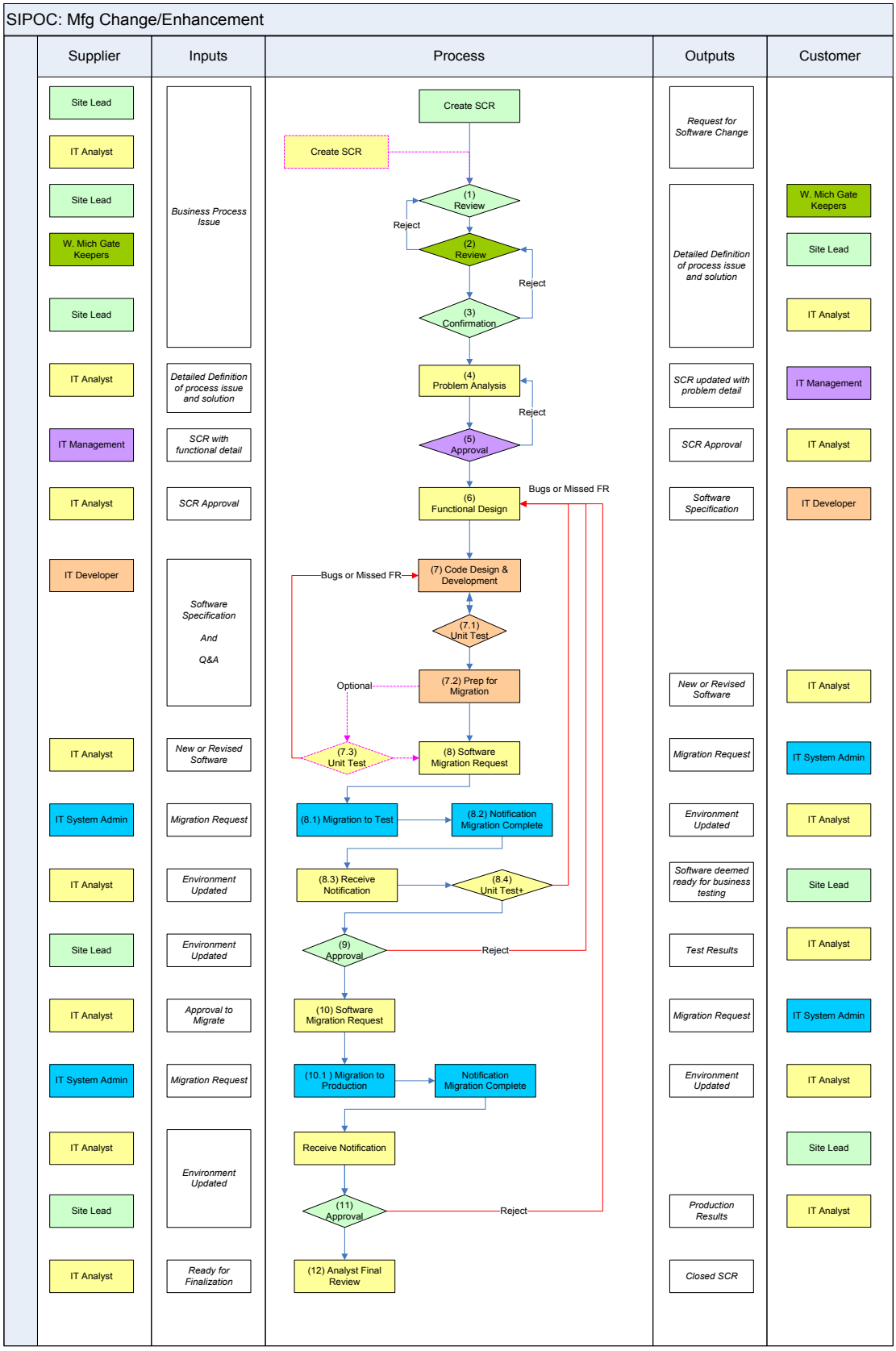


Figure 12 - Manufacturing Enhancement SIPOC

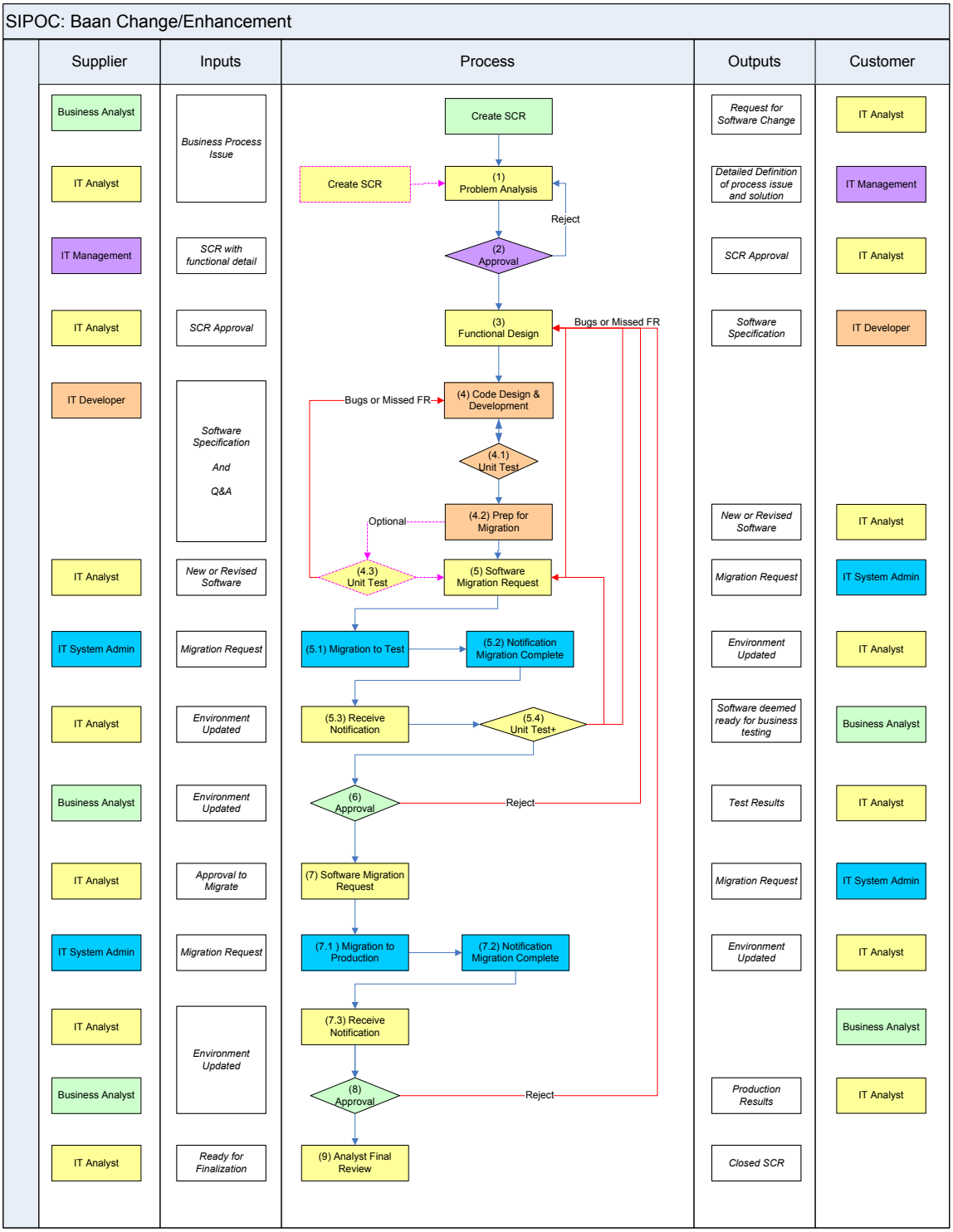


Figure 13 - Baan ERP SIPOC

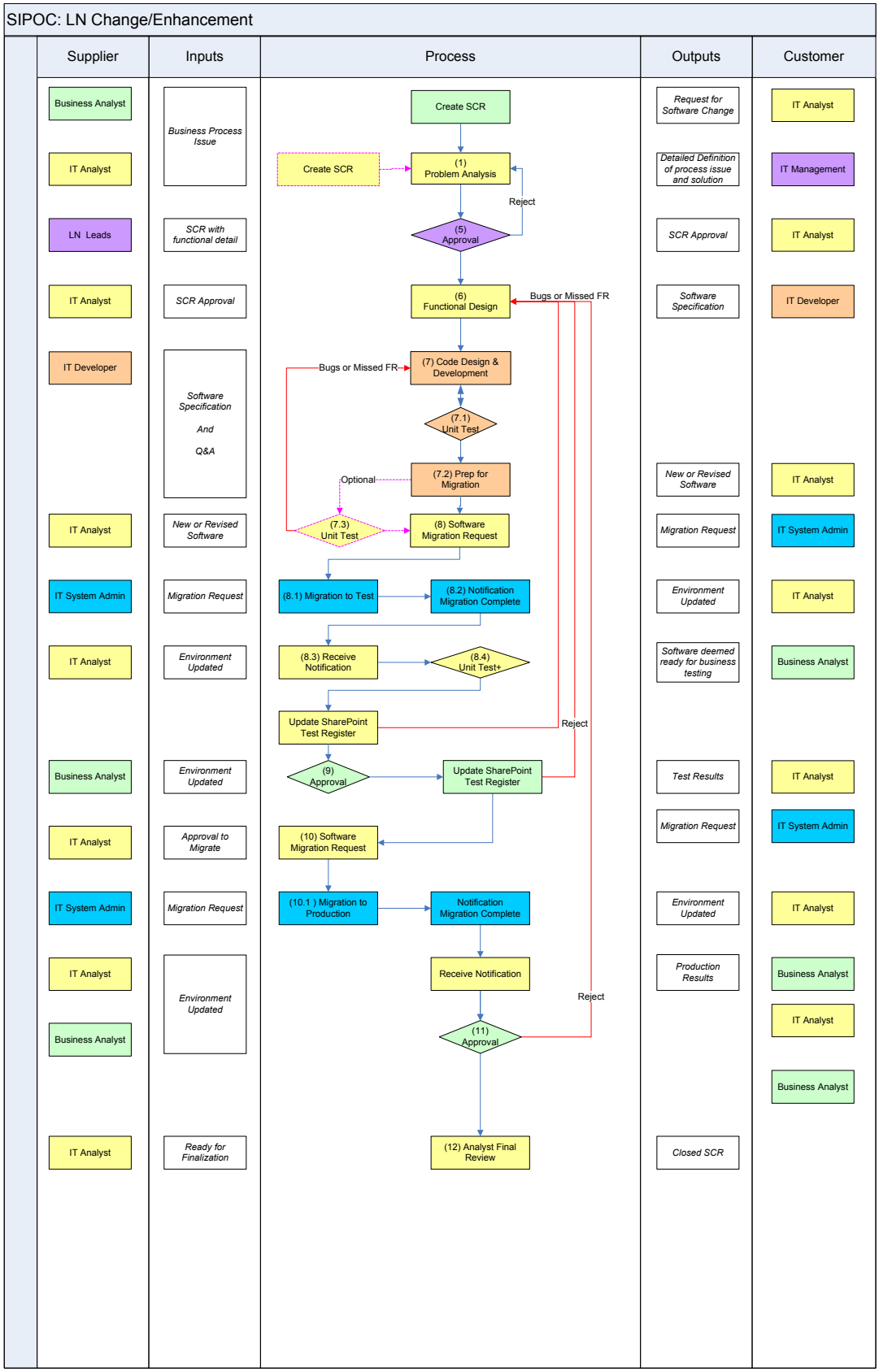


Figure 14 - LN ERP SIPOC

Appendix B - Factors Critical to Quality (CTQ)

One of the most challenge elements within the project was to work with direct customers of ERP to outline the factors that they deem **critical to quality**. This was a foreign concept and required more than one instance of providing an explanation alongside examples taken from isixsigma.com (Simon, Customer CTQs – Defining Defect, Unit and Opportunity, n.d.). The brainstorming and iterative cycles of comparison were very difficult as the team struggled to understand the concept of quality defined as measuring the process quality. Nevertheless, the team prevailed and eventually reached consensus regarding the CTQs that they need as a customer in the software change management process. These CTQs are listed below and organized by each customer role.

Note - Size of change when development effort is:

- Small Change is < 8 hours
- Medium Change is > 1 day and < 2 months with no more than 1 system integration
- Large Change is > 6 months and impacts other systems

Site Lead as the Customer							
ID	**Supplier**	**Supplier Output**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**
A1	West Michigan Gate Keepers	Detailed Definition of process issue and solution	Timing considerations	Length of advance notice aligns with the size of remedy needed	Insufficient engagement with IT Analyst when business process problem first appears	The degree of severity regarding business process in need of software remedy	Initial contact with Site Lead
A2	IT Analyst	Detailed Definition of process issue and solution	Communication confidence regarding requirements	Number of functional design errors	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
A3	IT Analyst	Detailed Definition of process issue and solution	Setting and meeting agreed due date	Instances of communication exchange regarding progress status	Instance of surprises due to lack of communicatio n	One surprise	Set of time slated for defining specifications
A4	IT Analyst	Software deemed ready for business testing	Notification that Software is Ready for Testing	Number of test cases that fail.	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
A5	IT Analyst	Production Results	Production Reliability	Number of production issues at first usage	Any single issue	One Session	Each Session packed in the SCR

Business Analyst as the Customer							
ID	**Supplier**	**Supplier Output**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**

B1	IT Analyst	Detailed Definition of process issue and solution	Communication confidence regarding requirements	Number of functional design errors	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
B2	IT Analyst	Detailed Definition of process issue and solution	Setting and meeting agreed due date	Instances of communication exchange regarding progress status	Instance of surprises due to lack of communicatio n	One surprise	Set of time slated for defining specifications
B3	IT Analyst	Software deemed ready for business testing	Notification that Software is Ready for Testing	Number of test cases that fail.	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
B4	IT Analyst	Production Results	Production Reliability	Number of production issues at first usage	Any single issue	One Session	Each Session packed in the SCR
B5	IT Analyst	Software deemed ready for business testing	Functionality meets specifications	Number of test cases failed due to incomplete functionality	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
B6	IT Analyst	Software deemed ready for business testing	Meets Performance Requirements	Session functions within specified transaction time	Session runs 10% longer than required	1 transaction	Each transaction

IT Analyst as the Customer							
ID	**Supplier**	**Supplier Output**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**
C1	Business Analyst	Detailed definition of process issue and solution	Documentation on current condition/ metrics	Number of occurrences where data provided was found to be inaccurate	Current business process situation is incorrectly represented	Current business process situation is incorrectly represented more than 3 times	Initial contact with IT Analyst
C2	Business Analyst	Detailed definition of process issue and solution	Target condition/ metric	Number of changes to target condition over the course of the project	Future business process remedy is incorrectly represented	The outline of changes to be provided in the software remedy	Final confirmation of business need with IT Analyst
C3	Business Analyst	Detailed definition of process issue and solution	Timing considerations	Length of advance notice aligns with the size of remedy needed	Insufficient engagement with IT Analyst when business process problem first appears	The degree of severity regarding business process in need of software remedy	Initial contact with IT Analyst
C4	IT Mgt	Email notification that approval process has completed	SCR updated with the management approval in timely manner.	Response time from receipt of request	Response of Approval or Rejection exceeds measure	One SCR	One SCR

C5	IT Mgt	Email notification that approval process has completed	SCR sent to IT Analyst.	Rationale for response is sent at time of decision	Response of Approval or Rejection is not sent at time of decision	One SCR	One SCR
C6	IT Developer	New or revised Software	Timeliness of software completion	On or before agreed upon date	Completion is after agreed upon date	One SCR	One SCR
C7	IT Developer	New or revised Software	Quality of software	Number of test cases that fail.	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
C8	IT System Admin	Updated Environment	Back-up copy of existing components	Backup Exists	Backup does not exist	One Component	One Migration of that Component
C9	IT System Admin	Updated Environment	Timeliness of migration	Completion within time parameters agreed upon	Migration is late	One Migration	One Migration
C10	IT System Admin	Updated Environment	Zero defects in migration.	Number of re-migrations due to failure from IT System Admin	Remigration is required	One Migration	One Migration
C11	Business Analyst	Approval Results	Timeliness of testing	Completion within time parameters agreed upon	Testing is completed late	One Migration	One Migration
C12	Business Analyst	Approval Results	Quality of testing	Number of defects found post implementation	Software bug or missing functionality is found after production migration	One Migration	One Migration
C13	Business Analyst	Approval Results	SCR updated with the Business Analyst approval.	Rationale is documented in the SCR	Approval is not completed in agreed upon time	One SCR	One SCR
C14	Business Analyst	Approval Results	SCR sent to IT Analyst	SCR sent at time of pass/fail decision	Success or Fail is not communicated to IT Analyst at time of finding	One SCR	One SCR

IT Manager as the Customer							
ID	**Supplier**	**Supplier Output**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**
D1	IT Analyst	SCR approval decision	Request for SCR approval	Sufficient information to make proper decision	SCR doesn't explain what problem is to be solved, what value it adds, and what effort it will require. All elements must be included.	One SCR	One SCR

IT Developer as the Customer							
ID	**Supplier**	**Supplier Output**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**
E1	IT Analyst	Resulting Software Specification	Receipt of Functional Specification	Clear, complete, and correct documentation of functionality required.	More than 5 % of test cases fail	One Session	Each Session packed in the SCR
E2	IT Analyst	Resulting Software Specification	Proper Data Model included in Specification	Number of table design changes during testing	Any single change	One Table	Each Table packed in the SCR
E3	IT Analyst	Resulting Software Specification	Business Process Flow Diagram included in Specification	Number of missing process tasks	Any single task related issue	One Program Script or DLL	Each Script or DLL packed in the SCR
E4	IT Analyst	Resulting Software Specification	All scenarios of exception handling included in Specification	Number of surprise exceptions	Any single unexpected exception	One Script or DLL	Each Script or DLL packed in the SCR
E5	IT Analyst	Resulting Software Specification	Examples of successful test results	Number of examples match number of test cases	One unidentified test case	One Test Case	Each Test Case packed in the SCR
E6	IT Analyst	Resulting Software Specification	Future Business Process Change fully communicated to other stakeholders	Number of requests that are in conflict with SCR business process	Any single request	One SCR	One SCR

IT System Administrator as the Customer							
ID	**Supplier**	**Outputs**	**CTQ**	**CTQ Measure**	**Defect**	**Unit**	**Opportunity**
F1	IT Analyst	Migration Request	Migration has been requested	Accurate packaging of component within the software dump	Old component overwrite newer version	One Component	One SCR Task
F2	IT Analyst	Migration Request	Interdependent components are migrated in the same sequential order of creation or maintenance	Only new components are packaged within the software dump	Old component overwrites newer version	One Component	One SCR Task
F3	IT Analyst	Migration Request	Software dump is clean of known risks	Software dump passes script validation	Error identified by the script	One Scan	One Scan
F4	IT Analyst	Migration Request	Request was placed in time for the Sunday Maintenance Window Meeting	SCR migration requests are sent on time to allow for effort estimation and prioritization	Request is sent after the schedule has been finalized	One SCR Migration Task	One SCR Migration Task

F5	IT Analyst	Migration Request	Change since last migration field not properly filled out	Change since last migration field accurately filled out providing IT	Change since last migration field not properly filled out.	One SCR Migration Task	One SCR Migration Task
F6	IT Analyst	Migration Request	Migration request was for proper environment	Migrations are requested into environments in the proper order, with proper approvals	Migration is being requested to incorrect environment (Production before test).	One SCR Migration Task	One SCR Migration Task
F7	IT Analyst	Migration Request	Components manually migrated by wrong personnel	Migrations are only to be done by IT System Administrators using approved process	Migration manually completed by unauthorized person	One SCR Migration Task	One SCR Migration Task

Table 9 - CTQs for Cozy Couches

Appendix C - Data Extraction Plan

The SCR system doesn't have a relational database. Applications developed in Lotus Notes have a document database. Creating a view of documents is a relatively easy task for a Notes developer and there are several views in the SCR system. However, there is no functionality for producing an output report. Although users submitted requests for the ability to produce output reports, management declined approval. The rationale for declining those requests was that the Notes developers needed to complete their backlog of project work supporting software changes for other Notes applications. The requests simply were not deemed a priority sufficient for allocating resources.

There are two methods for extracting data from the SCR system. In the first method, high level content can be extracted into a file with comma separated values. The columns of the output file match the columns of the view. Unfortunately, all the views in the SCR system only show the most current work flow step in the process. For the purpose of this DMAIC, we also needed to have the actual work flow steps where value is added at each step of software change or creation.

To acquire an output file that contained the work flow information, it was necessary to use the search features and select the structured text format while performing the extract tasks. This is the second method and the structured text format is a vertical list of document fields. When performing these extracts, I discovered that the output sequential sequence of document fields was inconsistent with no recognizable pattern to explain why this was the case.

When performing a search selection, each field in the list is for all Notes Applications (Figure 15). There is no error message to guide you if you could make selections that are inappropriate for the SCR system. The search returns a listing of documents that meet the search requirements (Figure 16). The user then completes the tasks to trigger the extract process (Figure 17). Once the extract process is completed, the user can view the output result using any text editor (Figure 18). The structured text output is a vertical list. An example of the non-readable field that did not consistently separate SCRs properly can be seen in Figure 19.

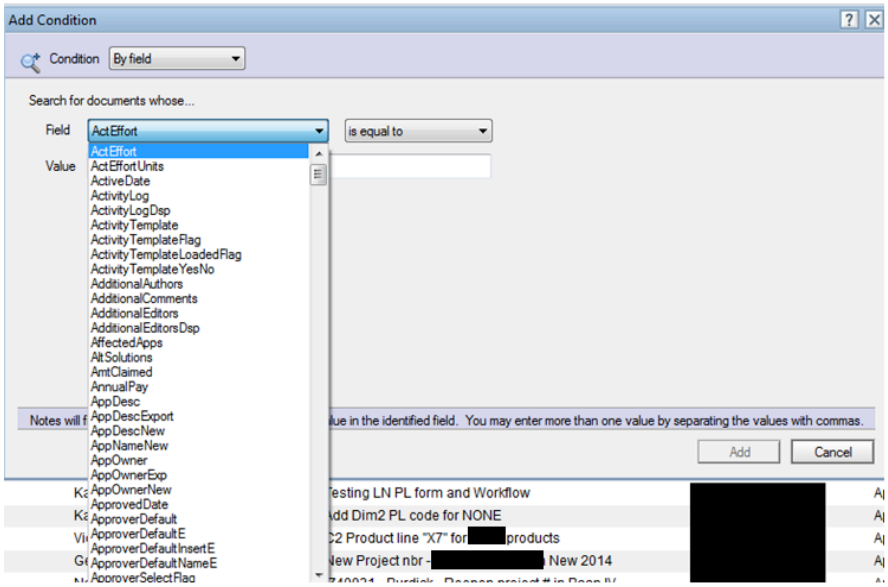


Figure 15 - SCR Search Fields

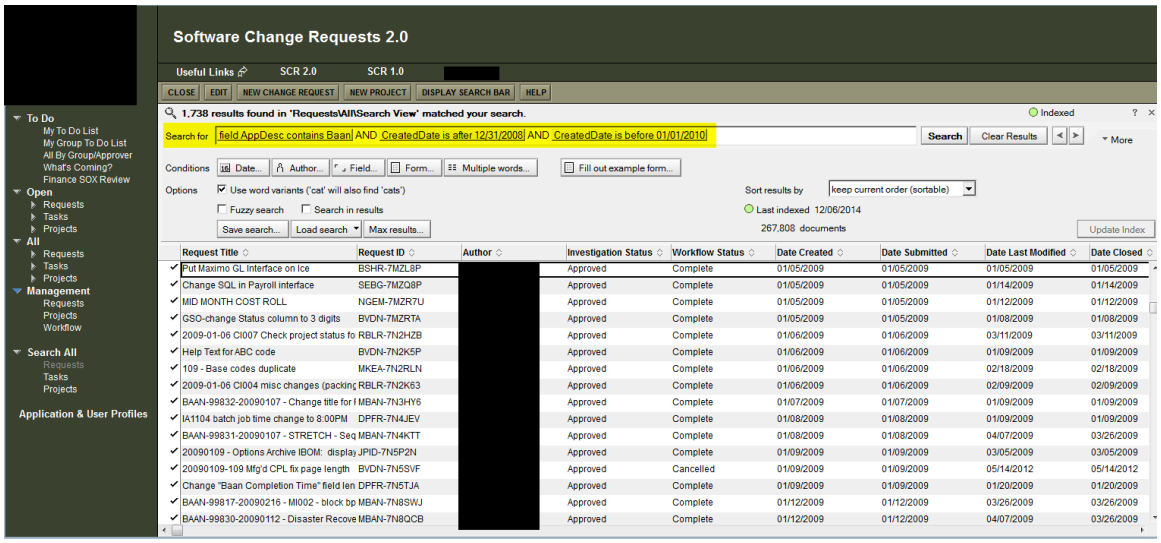


Figure 16 - Notes Search Result Set

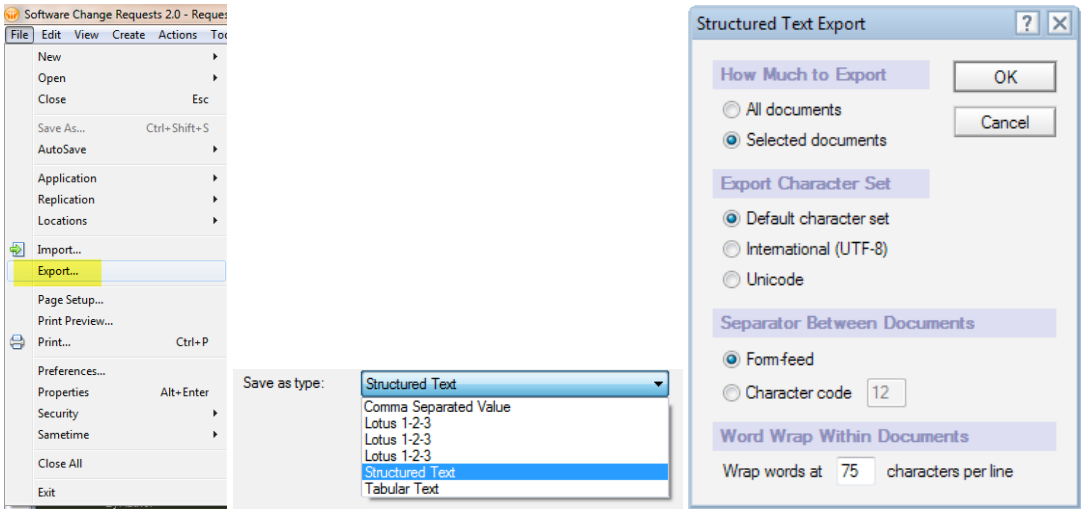


Figure 17 - Steps to Create Structured Text Export

```
$FILE:
BUDGET:
LastScanOID:  31b7ddca65ff4e21be4b062f79178258c3318a063e02aed1f77e7b6ec429210d5a4e5b1121424770be93d6b9955ab15c
LastScanOIDCheck:  8f187caf071e5793e0d6de16744b0b629775feef600efccc7d9c8925bb2b153590e82746078e66bdad6a0eb5baf7d5d4
LastScannedVersion:  d41406d5a40f482685663da0f5654b6b3916b6bd11a6ee91f15da4851578d5bd2546e7a70b07882fcc984ffeebce6d2e
LastScannedVersionCheck:  c79e1419d25e7fbee57d37b63c93d0f7f8021708f3818e4c735deae4d63414d2b4835823c37d068ef5f4209aeb3541cd
SOXReviewYesNo:  N
FinanceReviewer:  CN=William Jones/O=Cozy Couches
FinanceReviewDate:  08/13/2014 02:52:18 PM
Process:
ProcessOwner:
ProcessCoach:
$WebFlags:  V
From:  CN=Bob Madison/O=Cozy Couches
ModifiedDT:  08/29/2014 09:17:23 AM
AdditionalAuthors:
SendBackFlag:
RequestIDHistory:
RequestID:  BMCN-9MHPKX
DocID:  1DE4DD2CEC32520485257D2500651062
WorkflowStatus:  IP
InvestigationStatus:  A
CreatedDate:  07/30/2014 02:23:53 PM
SubmittedDate:  07/30/2014 02:24:50 PM
ClosedDate:
WhatsComing:  Carl Brown/Cozy Couches Systems Analysts
```

Figure 18 - Sample Structured Output File Screenshot

```
36043
36044
36045
36046
36047
36048
36049
36050
36051
36052
36053 FF
36054 SOXReviewYesNo:  N
36055 FinanceReviewer:
36056 FinanceReviewDate:
36057 Process:
36058 ProcessOwner:
36059 ProcessCoach:
36060 $WebFlags:  V
```

Figure 19 - Example of Non-Readable Character (line 36053)

Appendix D - Available Data Elements

Data in the SCR system has never been purged and the system is now tattered with performance issues. The extract was restricted to SCRs starting in 2009 up until the go-live for LN in October 2014. This extract included the three SCR processes described earlier in the Define phase and was to be converted into an Excel file format in order to perform filtering and create calculations for measuring performance. The Management View in the SCR system has the most set of fields available. Many of these fields show the designers intention of using the information for planning future projects and monitoring allocation of resources. There were no fields for linking the SCR to a project within the Management view. However, the SCR itself can be linked to a project. But the only way to "see" this relationship is to where the SCRs are grouped by a specific Project. Very few teams are using the projects feature. There were no SCRs for Baan or LN linked to a project.

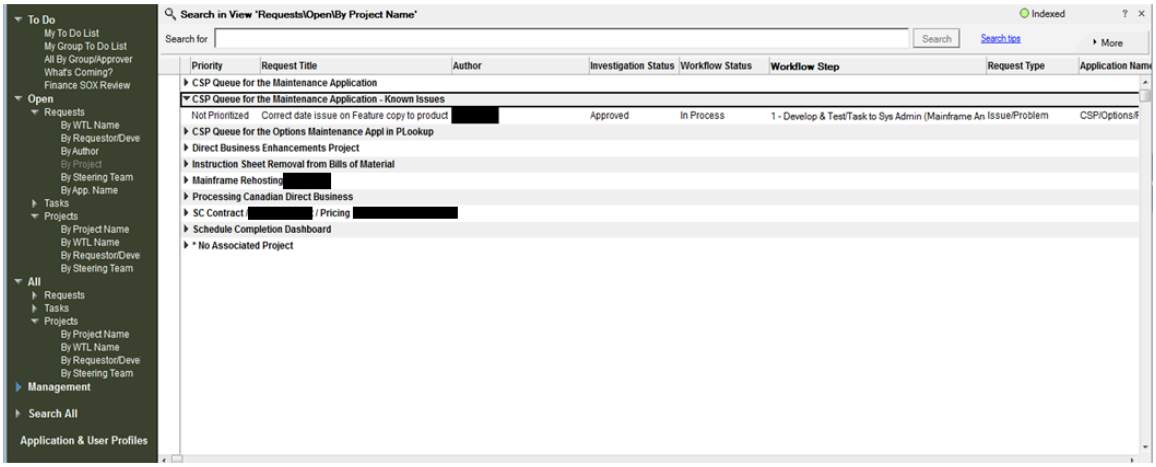


Figure 20 - SCR System Project View

Here is a list of the available data elements from the Management View:

Workflow Status	Request Type	Department	In Budget Notes
Date Created	Affected Applications	Business Impact	Percent Complete
Date Submitted	Primary Analyst	Strategic Alignment	Requested Due Date
Date Last Modified	Secondary Analyst	Short Description	IT Due Date
Mod-Sub	Requestor	Estimated IT Effort	Project Name
Request Title	Area(s) Affected	Actual IT Effort	SOX doc. chng.
Request ID	Business Lead	Estimated Non-IT Effort	SOX Process
Author	Steering Team	Capital	Finance Reviewer
Investigation Status	Steering Team	Expense	Review Date
Application Name	Priority	In Budget	Developer Priority
			Developer Comments

Table 10 - SCR Available Fields

Appendix E –Data Extraction Scrub

The following process steps describe the manual, approach to format the eventual structured text extract file:

1. Open the SCR extract file in MS Excel with NO parsing after creating a backup copy.
2. Insert a new column and fill it with sequential numbers. This will serve as an index that allows for sorting that returns the rows to the original sequence.
3. Insert a worksheet that lists a table of SCR data fields to be retained given that most of the data elements do not have values and lack a consistent sequence of order.
4. Go back to the source data and add a column to house a formula for locating the colon character. Manually inspect one row for accuracy.
5. Add another column and add a formula that uses a vertical lookup to search the table of data fields to retain. Embed the formula for identifying the colon character to properly parse the value that serves as the search value within the overall vertical lookup formula. Include in the formula the logical test for whether or not a value was found.
6. Use data filtering to hide the rows that are not relevant
7. Select visible cells and then copy the data to a new worksheet by pasting values.
8. On this new worksheet, find the beginning and ending data element for a given SCR. There are instances where the Request ID will not be placed in the proper order and you will need to cut and paste the Request ID to the proper place. When this occurs, log into the SCR system and verify that the document fields are correctly groups with the Request ID that as not in the proper sequence. Request ID should be the first field. The Work Flow Edit Log should be the last field. These rows serve as endcaps.
9. Sort the rows within the endcaps. Then, select the all rows with in the endcaps and paste special transpose. Delete the extra rows. Repeat the process for the rows belonging to the next SCR. Repeat until all SCR fields have been properly sequenced and transposed.
10. Once the vertical list of rows has been properly normalized into a table of SCR, parse the last column containing the work flow steps.
11. Add columns to count the number of work flow steps overall, for the user, the analyst and the developer. Copy and paste as values.
12. Scrub the work flow steps to properly show the initial onset of design and development by removing any subsequent work flow steps for repairing design or code. Removing these rework steps will mean that this duration is reflected within the testing effort.
13. Add two more columns for the total number of SCR migrations and the last migration date. This will eliminate non-testing duration from the testing duration created in step 12.

Appendix F - Examine data elements for CTQ evaluation

Each CTQ was reviewed and cross compared with data available within the SCRs (Table 11). Those that cannot be measured are highlighted with pink. Yellow was used for those that are risky and green for those we can measure.

Because there has never been any internal service level agreement that is reported and actively monitored, Cozy Couches has fields within the SCR system that simply are not used while there are other data elements that should be added to align with the CTQs.

This situation means that the measurement of data in the SCR system cannot be totally based on the CTQs. The data will need to be examined to find out what is possible to measure and should be measured as a beginning baseline to build an understanding of the current condition.

Site Lead as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric**	**What can be measured from SCRs?**	**Fields from SCR**
A1	Timing considerations	Length of advance notice aligns with the size of remedy needed	SCR Core Duration exceeds window of notification from desired delivery date (estimated before detail design)	The rich text section of the detailed description should contain the information needed to determine the potential solution and related development effort. The due date is not being provided.	Est Effort (In Hours) Due Date Submitted Date
A2	Communication confidence regarding requirements	Number of functional design errors	Number of Instances of Missing Functionality within Test Package	Rework following development could be inferred from migrations and hops, but it's possible that this could be incorrect information if they are testing in the dev environment. To be accurate, the test cases and status (pass/fail) should be in the SCR.	None available. There are no test cases in the SCR.
A3	Setting and meeting agreed due date	Instances of communication exchange regarding progress status	Compare Due Date and Completed Date	The due date isn't filled out.	Due Date Completion Date
A4	Notification that Software is Ready for Testing	Number of test cases that fail.	Any instance of delay due to software not being ready in the environment.	This seems like it really should be a comparison between the migration date and when the analyst sent the SCR to the Site Lead	
A5	Production Reliability	Number of production issues at first usage	A new SCR to fix the problem is required	New SCR gets created that identifies it is a fix. Analysts are not required to write this in the SCR but some might do it.	Description that mentions the need to fix. Fixes might not be labeled as such as it is not required.

Business Analyst as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric**	**What can be measured from SCRs?**	**Fields from SCR**
B1	Communication confidence regarding requirements	Number of functional design errors	Number of Instances of Missing Functionality within Test Package	Rework following development could be inferred from migrations and hops, but it's possible that this could be incorrect information if they are testing in the dev environment. To be accurate, the test cases and status (pass/fail) should be in the SCR.	None available. There are no test cases in the SCR.
B2	Setting and meeting agreed due date	Instances of communication exchange regarding progress status	Compare Due Date and Completed Date	The due date isn't filled out, which is why this comparison didn't get listed as the CTQ measure.	Due Date Completion Date
B3	Notification that Software is Ready for Testing	Number of test cases that fail.	Any instance of delay due to software not being ready in the environment.	Rework following development could be inferred from migrations and hops, but it's possible that this could be incorrect information if they are testing in the dev environment. To be accurate, the test cases and status (pass/fail) should be in the SCR.	None available. There are no test cases in the SCR.
B4	Production Reliability	Number of production issues at first usage	New SCR to fix the problem is required	New SCR gets created that identifies it is a fix. Analysts are not required to write this in the SCR but many do.	Description that mentions the need to fix. Fixes might not be labeled as such as it is not required.
B5	Functionality meets specifications	Number of test cases failed due to incomplete functionality	Number of Instances of Missing Functionality within Test Package	Rework following development could be inferred from migrations and hops, but it's possible that this could be incorrect information if they are testing in the dev environment. To be accurate, the test cases and status (pass/fail) should be in the SCR.	None available. There are no test cases in the SCR.
B6	Meets Performance Requirements	Session functions within specified transaction time	Transaction time is or is not within acceptable range	This is performance related and the only way to see this in a SCR is if a new SCR gets created that states its purpose is to address a performance issue. Analysts are not doing this.	Description that mentions the need to fix. Fixes might not be labeled as such as it is not required.

IT Analyst as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric**	**What can be measured from SCRs?**	**Fields from SCR**

C1	Documentation on current condition/ metrics	Number of occurrences where data provided was found to be inaccurate)	A new SCR is required for building missing functionality	The rich text section of the detailed description would list the previous SCR and outlines what is missing. Most analysts do not include this information, so this is not a reliable measure	Description that mentions the need to fix. Fixes might not be labeled as such as it is not required.
C2	Target condition/metri c	Number of changes to target condition over the course of the project	Number of design edits is more than the annual average	The number of edits in the SCR could help, but the Analyst could use a hyperlink to house all the changes. This is the best information we have available. Iterations could be higher.	Number of edits
C3	Timing considerations	Length of advance notice aligns with the size of remedy needed	SCR Core Duration exceeds window of notification from desired delivery date (estimated before detail design)	The rich text section of the detailed description should contain the information needed to determine the potential solution and related development effort. Analysts are not documenting this.	Est Effort (In Hours) Due Date Submitted Date
C4	SCR updated with the management approval in timely manner.	Response time from receipt of request	Approval Date is within size threshold	The approval step date is visible and can be compared to the date submitted.	Work flow approval date
C5	SCR sent to IT Analyst.	Rationale for response is sent at time of decision	Rationale for rejection is provided within the SCR	There is a comment section that the IT Manager can use or they can update the rich text field to explain why the SCR was approved or not. They are not using this field.	Status that shows the work was not taken forward
C6	Timeliness of software completion	On or before agreed upon date	Compare Due Date and Completed Date	The date for expected initial release is nowhere in the SCR	None available
C7	Quality of software	Number of test cases that fail.	Test Case failures / total number of test cases	Rework following development could be inferred from migrations and hops, but it's possible that this could be incorrect information if they are testing in the dev environment. To be accurate, the test cases and status (pass/fail) should be in the SCR.	None available. There are no test cases in the SCR.
C8	Back-up copy of existing components	Backup Exists	Component was or was not available for request to revert.	There is nothing in the SCR to show this happened. Backups are not noted in the SCR. It is assumed to have happened.	None available
C9	Timeliness of migration	Completion within time parameters agreed upon	Compare Request Date and Completed Date	The requested migration date and the completion date	Migration task due date migration task completion date

C10	Zero defects in migration.	Number of re-migration due to failure from IT System Admin	Sysadmin enters a note reporting the error and corrective action	A re-migration can occur using the same initial migration task and then marked as complete when the migration is stable.	New migration task could be requested, but isn't required.
C11	Timeliness of testing	completion within time parameters agreed upon	Compare Due Date and Completed Date	The due date for testing completion is nowhere in the SCR	None available
C12	Quality of testing	Number of defects found post implementation	A new SCR is required for building missing functionality	The rich text should indicate that the SCR is for correcting issues due to poor testing. Analysts are not documenting this.	Description that mentions the need to fix. Fixes might not be labeled as such as it is not required.
C13	SCR updated with the Business Analyst approval.	Rationale is documented in the SCR	Compare Due Date and Completed Date	There is a comment section that the IT Manager can use or they can update the rich text field to explain why the SCR was approved or not. Managers are not doing this.	None available
C14	SCR sent to IT Analyst	SCR sent at time of pass/fail decision	User or IT Analyst send a follow-up communication to inquire about the status	This would only be possible if we had a date where we could record the actual test completion date and then compare it to the date when the analyst was notified of the results	None available

IT Manager as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric?**	**What can be measured from SCRs?**	**Fields from SCR**
D1	request for SCR approval	Sufficient information to make proper decision	Item is check listed as sufficient	This could be a subjective area, but a checklist could be created for minimum required information. Some of the fields in use today could help with this measure, but they are not populated since they are not required	Rich Text Detailed Description. Rich Text cannot be extracted for reporting.

IT Developer as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric?**	**What can be measured from SCRs?**	**Fields from SCR**
E1	Receipt of Functional Specification	Clear, complete, and correct documentation of functionality required.	Number of Instances of Missing Functionality within Test Package	Rework following development could be inferred from migrations and hops, but it's possible that this could be incorrect information if they are testing in the dev environment. To be accurate, the test cases and status (pass/fail) should be in the SCR.	None available. There are no test cases in the SCR.
E2	Proper Data Model included in Specification	Number of table design changes during testing	Item is check listed as sufficient	Content or attachments that would have the data model. Analysts are not providing this information in the SCR.	Rich Text Detailed Description. Rich Text cannot be extracted for reporting.

E3	Business Process Flow Diagram included in Specification	Number of missing process tasks	Item is check listed as sufficient	Content or attachments that would have the business process model. Analysts are not providing this information.	Rich Text Detailed Description. Rich Text cannot be extracted for reporting.
E4	All scenarios of exception handling included in Specification	Number of surprise exceptions	Item is check listed as sufficient	Content or attachments that would list the appropriate error handling needed as baseline of comparison. Analysts are not providing this information.	Rich Text Detailed Description. Rich Text cannot be extracted for reporting.
E5	Examples of successful test results	Number of examples match number of test cases	Compare the number of examples with the number of test cases	Request for additional functionality results in editing the SCR. Sometimes the analyst and developer do this verbally but do not record it.	Rich Text Detailed Description. Rich Text cannot be extracted for reporting.
E6	Future Business Process Change fully communicated to other stakeholders	Number of requests that are in conflict with SCR business process	A new SCR to fix the problem is required	Content that indicates that we need to reverse or repair a previous software change due to conflicting requirements that were not known during the original SCR work. Analysts are not reporting this.	Rich Text Detailed Description. Rich Text cannot be extracted for reporting.

IT System Administrator as Customer					
ID	**CTQ**	**CTQ Measure**	**Metric?**	**What can be measured from SCRs?**	**Fields from SCR**
F1	Migration has been requested	Accurate packaging of component within the software dump	Request was or was not sent to Sysadmin to restore the original component.	List of components in the SCR migration tasks versus the actual components in the software dump	SCR Task SysAdmin Notes is available for this, but they are not using it.
F2	Interdependent components are migrated in the same sequential order of creation or maintenance	Only new components are packaged within the software dump	Request was or was not sent to Sysadmin to restore the original component.	List of components in the SCR migration tasks versus the actual components in the software dump	SCR Task SysAdmin Notes is available for this, but they are not using it.
F3	Software dump is clean of known risks	Software dump passes script validation	Zero errors returned from script results	The SCR is not used for this validation task	SCR Task SysAdmin Notes is available for this, but they are not using it.
F4	Request was placed in time for the Sunday Maintenance Window Meeting	SCR migration requests are sent on time to allow for effort estimation and prioritization	Compare Request Date against the Maintenance Window Dates	Comparing the migration task due date to the weekly review board meeting date. SCR doesn't show date that the migration request was sent to Sysadmin prior to the weekly review board meeting.	Due Date
F5	Change since last migration field not properly filled out	Change since last migration field accurately filled out providing IT	Sysadmin enters a note reporting the error and corrective action	The information in the changes since last migration to test should align with the tasks required to get success confirmed. Sysadmin has a place to note when this is in error but they are not using it.	SCR Task SysAdmin Notes is available for this, but they are not using it.

F6	Migration request was for proper environment	Migrations are requested into environments in the proper order, with proper approvals	Request was or was not sent to Sysadmin to restore the original component.	There are no fields or SCRs for this production issue created by generating a defect.	SCR Task SysAdmin Notes is available for this, but they are not using it.
F7	Components manually migrated by wrong personnel	Migrations are only to be done by IT System Administrators using approved process	SOX Script reports violation	This CTQ is for when the SCR process was circumvented. By definition, it means there was no SCR when a component change is discovered. Better to lock down the environment and no longer define as a CTQ.	None available

Table 11 - CTQ Fit-Gap-Analysis

Appendix G - Impact of Non-value Process Steps

The SCR process flow includes steps for acquiring approvals and wrapping up final documentation tasks when the analyst closes the SCR. These steps do not add direct value to the work product: ERP software. To see if there is an impact that the non-value added steps had on overall duration, a comparison was made between total duration and work duration where work duration only includes tasks directly related to software change. A count of SCRs using both types of duration showed that the SCRs completed within 25 days had a dramatic increase when viewed from the value-add perspective. When the non-value added process steps are removed, the throughput is much higher.

Value Add Impact to Duration by Range of Days

From	To	Total All Duration		Total Work Duration	
<=25		1407	29%	2047	80%
>25	<=50	793	17%	232	9%
>50	<=75	441	9%	73	3%
>75	<=100	350	7%	51	2%
>100	<=125	273	6%	44	2%
>125	<=150	225	5%	19	1%
>150	<=175	187	4%	14	1%
>175	<=200	166	3%	14	1%
>200	<=225	150	3%	11	0%
>225	<=250	138	3%	8	0%
>250	<=275	121	3%	2	0%
>275	<=300	117	2%	5	0%
>300	<=325	107	2%	3	0%
>325	<=350	84	2%	1	0%
>350	<=375	75	2%	7	0%
>375	<=400	51	1%	1	0%
>400		99	2%	26	1%

Table 12 - Impact of Non-Value Steps

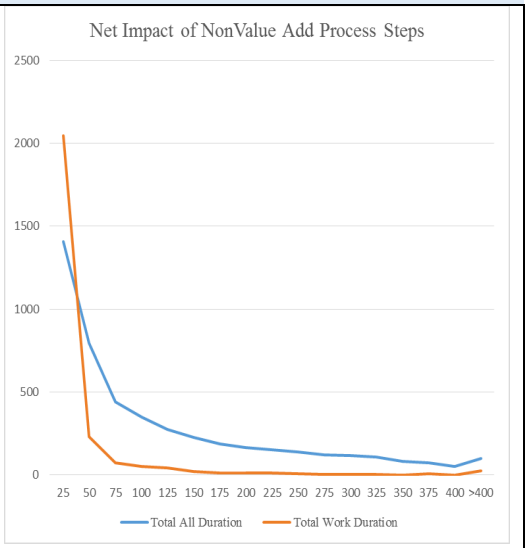


Figure 21 - Total Duration vs Work Duration

Appendix G - Impact of Size

Because there are no estimates in the SCRs, it is not possible to gauge the actual impact of non-equal effort being required for the software change. If we look at actual duration to imply size of effort, we risk enormous distortion because an SCR that took more than 400 days could really be a situation where a low priority easy enhancement was merely delayed until there was no other priority work to be completed on that day. It is equally possible that the SCR involved a large complex interface that resulted in numerous surprises and rework. There simply is no data in the SCR to help us.

But we can ask the question about the distribution of duration size on the overall work load. Given the enormous range of durations, we might reasonably assume that SCRs completed within less than a day were for production fixes and indirectly prove the presence of rework.

For this experiment, each completed SCR was tagged as a "Fix" if the duration was less than or equal to one day. If it was more than one day but less than or equal to five days, the SCR was tagged as "Small". When the duration was more than "Small", but less than or equal to 30 days, the SCR was tagged as "Medium". The remaining SCRs were tagged as "Large".

		2009	2010	2011	2012	2013	2014	Total
Baan	**Fix**	136	162	135	74	54	17	578
	Small	79	130	119	73	51	24	476
	Medium	197	222	211	135	93	50	908
	Large	75	86	55	42	31	15	304
LN	**Fix**	0	0	0	5	16	14	35
	Small	0	0	0	4	16	17	37
	Medium	0	0	0	8	31	45	84
	Large	0	0	0	30	64	42	136
Total	**Fix**	136	162	135	79	70	31	613
	Small	79	130	119	77	67	41	513
	Medium	197	222	211	143	124	95	992
	Large	75	86	55	72	95	57	440

Table 13 - Duration by System, Size and Year Started

In the Baan "Fix" duration SCRs, the total count is more than half the "Medium" durations. Both exceed the counts for "Small" and "Large". The case for LN is the opposite. The "Large" durations far outnumber the "Fix" durations. When we consider that 25 days duration is the significant portion of the total dataset, this means we can infer that the Baan SCRs are the majority. The Baan system has been fully

established since 2008 and the 25 duration pattern is likely a reflection of its maturity within the system lifecycle than the software change management performance we are trying to glean from the SCRs.

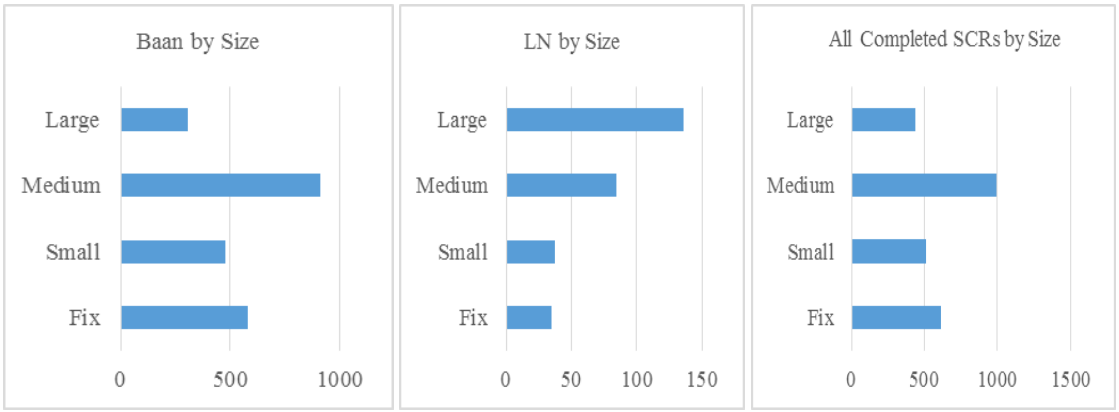


Figure 22 - SCR Counts by System and Duration Size

If we look at this same data year over year between the systems, we see the leveling and reduction of SCR size for Baan. It would be interesting to see if LN has the same pattern as Baan after it has been in production for six years. If the pattern emerges, it would be reasonable to assert that system skills, knowledge, and experience are the primary driver of duration. In DMAIC, you cannot assume. Data is required.

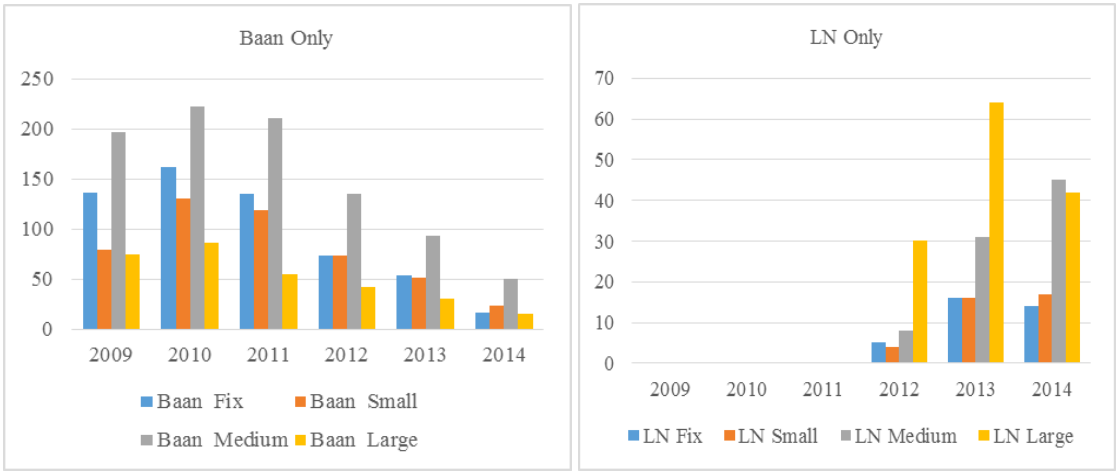


Figure 23 - SCR Size Duration by Year Started

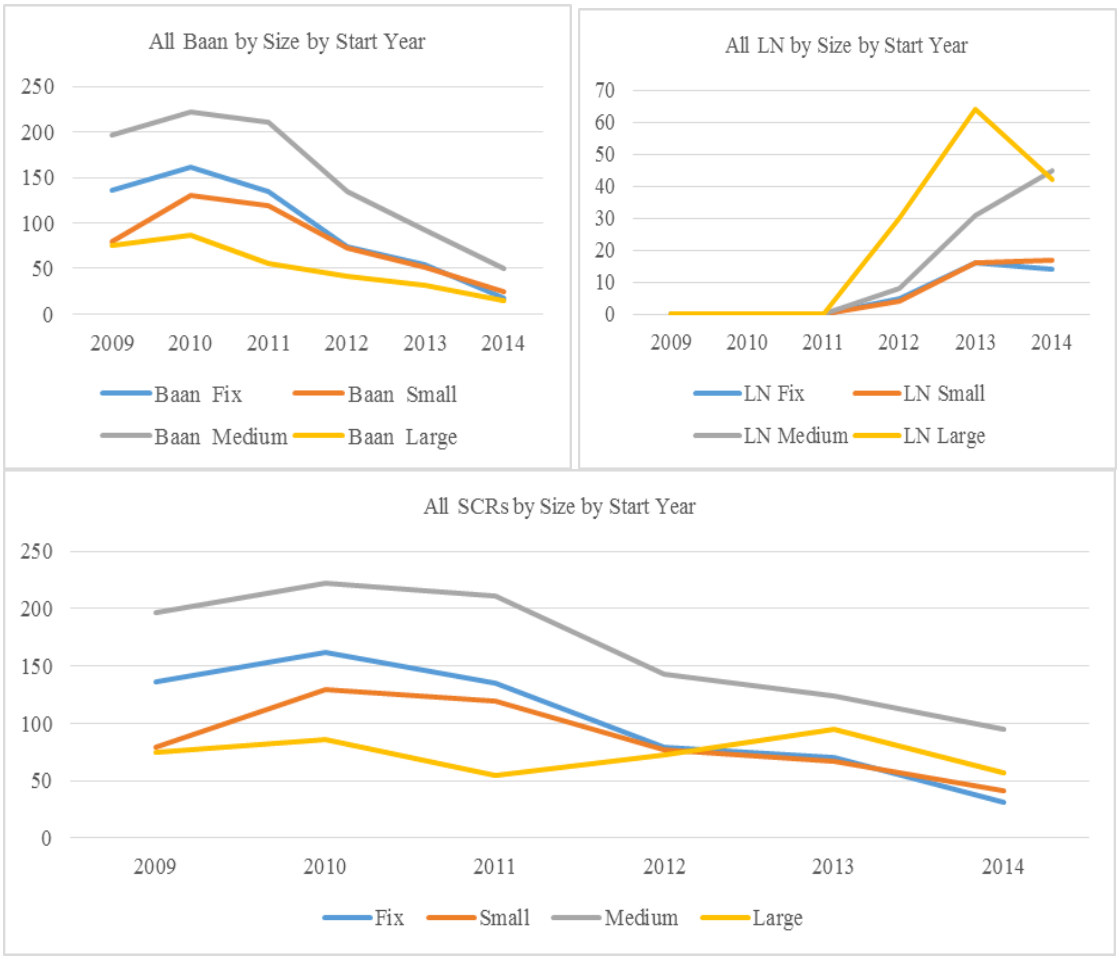
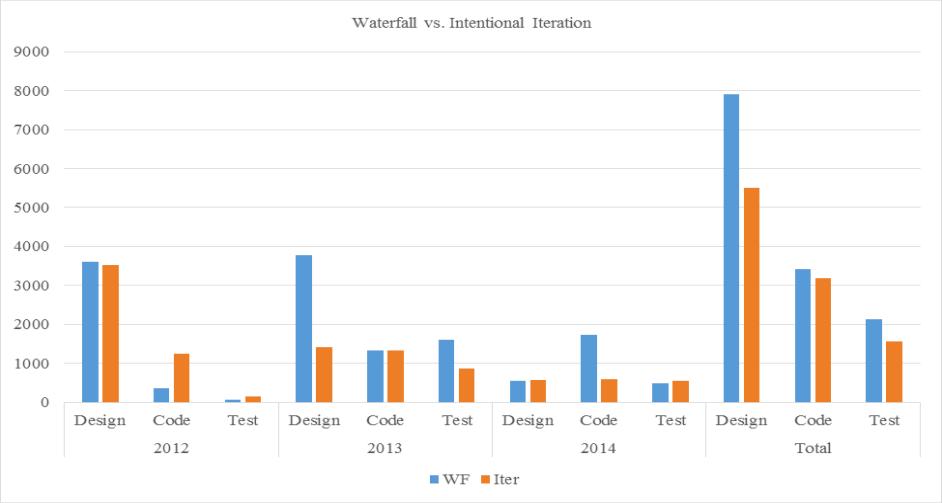


Figure 24 - SCR Duration by Size by Year

Appendix G - Impact of Style

Certain analysts are known for their preference to use the tradtional waterfall approach to managing their SCRs. Other analysts prefer the use of iteration to incrementally build out the functionality. The preference for iteration has been the case for Baan for quite some time. In this experiment, each SCR was tagged for the work style so that we could run a comparison to see if the preference for either style had an impact on SCR duration. The dataset was limited to LN completed SCRs where the mix of style preference exists.

Figure 25 - Waterfall versus Intentional Iteration



When viewed in total duration, only the design work is significantly impacted. The effort to code is nearly equivalent. Testing bears a small impact where waterfall has the greater duration. It is important to note that the duration pertaining to code activity is for first time delivery. Once testing begins, code activity is counted as part of the testing work. It is also important to note that the implementations for 2012 and 2013 were managed by an analyst who prefers intentional iteration. Other analysts worked primarily in the development environment with a mix of preference for waterfall and iteration during 2012 and 2013.

Based on this sample, we can say that a waterfall preference could drive duration to a longer period of time. We cannot assert this as absolute because the actual effort of the work is not tracked. The dates give us a duration of how long the SCR remained in a given phase of work. What we can say is that the preference for waterfall or intentional iteration has no impact on the SCR duration for the software code activity for the duration of first time delivery and minimal impact on the duration of testing.

Appendix G - Control Charts

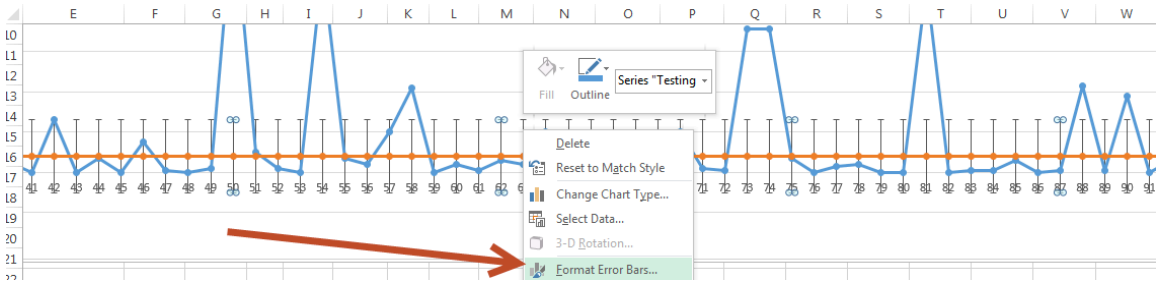
The next experiment involved using the SCR duration tag to see if this subgroup of SCRs have any signifcant variation with one another or if the process (pertaining to duration size) is stable. The sample of data for this work was for all completed SCRs where the design work started between January 2013 and March 2013. The core development of design, code, and test were taken for the duration in order to eliminate the role of "paperwork" and focus on software change only. The SCRs were also separated by size from the previous experiment.

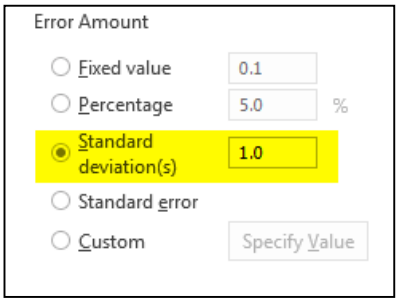
Figure 26 - Format Error Bars for Standard Deviation

The analysis of this experiment relied on the common use of four rules for evaluation (Bauman, De Heck, Leonard, & Miranda, 2011) that pin a process as not being in control:

- Rule 1: Any point falls beyond 3σ from the centerline
- Rule 2: Two out of three consecutive points fall beyond 2σ on the same side of the centerline.
- Rule 3: Four out of five consecutive points fall beyond 1σ on the same side of the centerline.
- Rule 4: Nine or more consecutive points fall on the same side of the centerline.

In the dataset, a column was added to calulate the mean value. This was only necessary for providing a visual way to show the data points above or below the mean when evaluating the rules. Microsoft Excel automatically calcuates the mean within the graphing capability when using the "Format Error Bars" feature within Excel graphs.





Because I was initally not familiar with this functionality, I had spent a fair amount of time researching how to calcuate standard deviation within Microsoft Excel. It's actually quite easy because the Format Error Bars function allows you to specifically set the level of standard deviation. No additional formulas were needed.

Figure 27 - Setting Level of Standard Deviation

SMALL SCRs - The small SCRs pass all four rules.

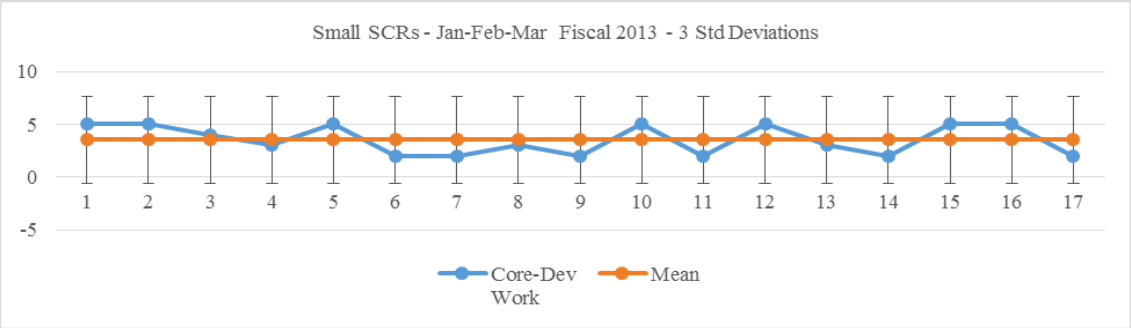


Figure 28 – Control Chart: Small SCRs (3 Standard Deviations)

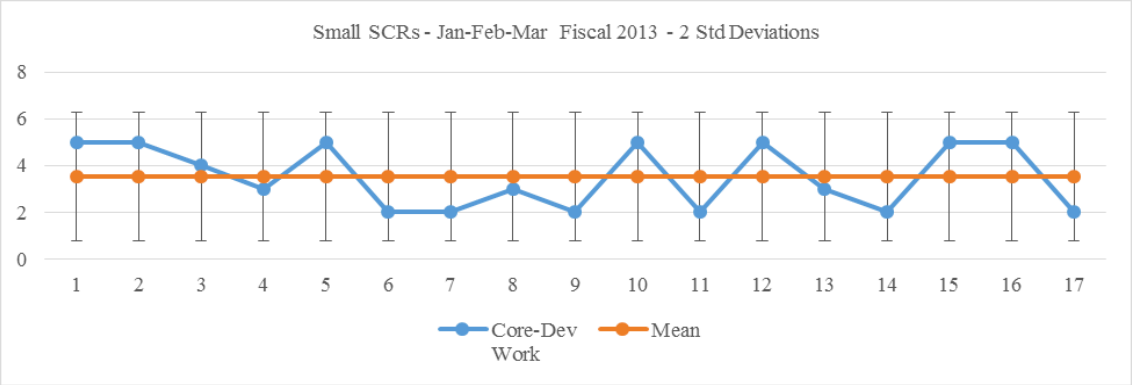


Figure 29 - Control Chart: Small SCRs (2 Standard Deviations)

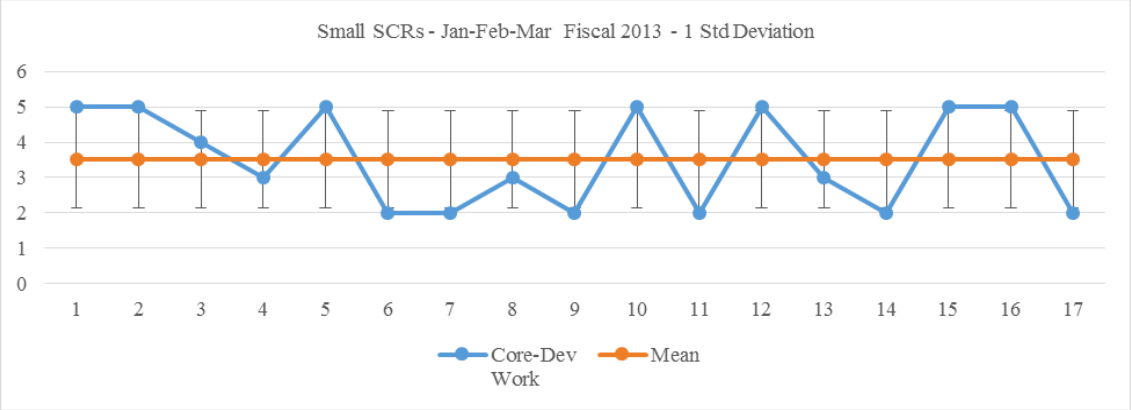


Figure 30 - Control Chart: Small SCRs (1 Standard Deviation)

MEDIUM SCRs - Although there is more variation, the “Medium” SCRs passed all four rules.

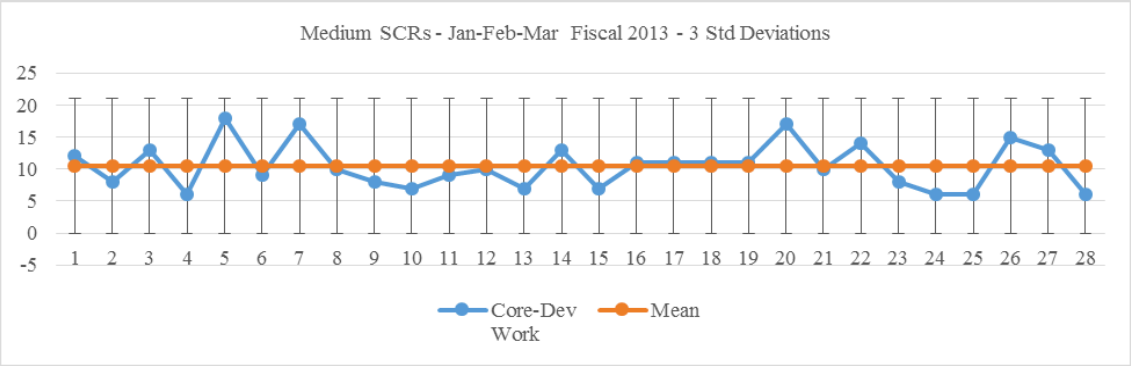


Figure 31 - Control Chart: Medium SCRs (3 Standard Deviations)

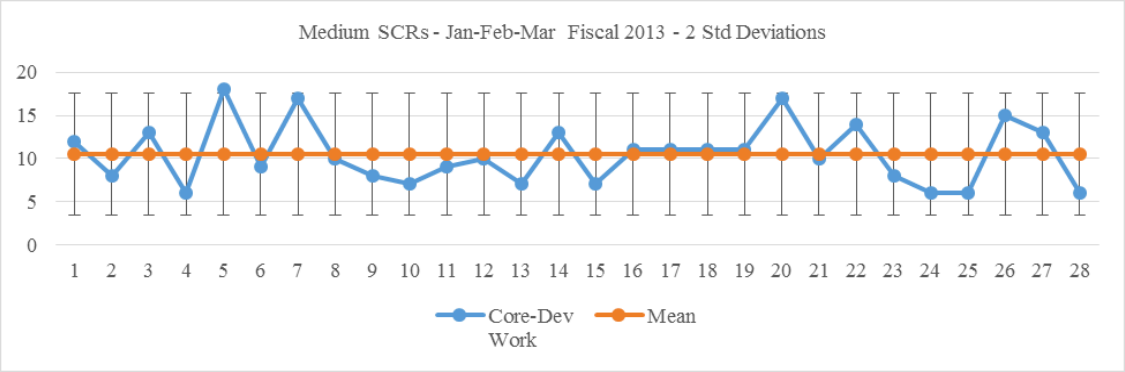


Figure 32 - Control Chart: Medium SCRs (2 Standard Deviations)

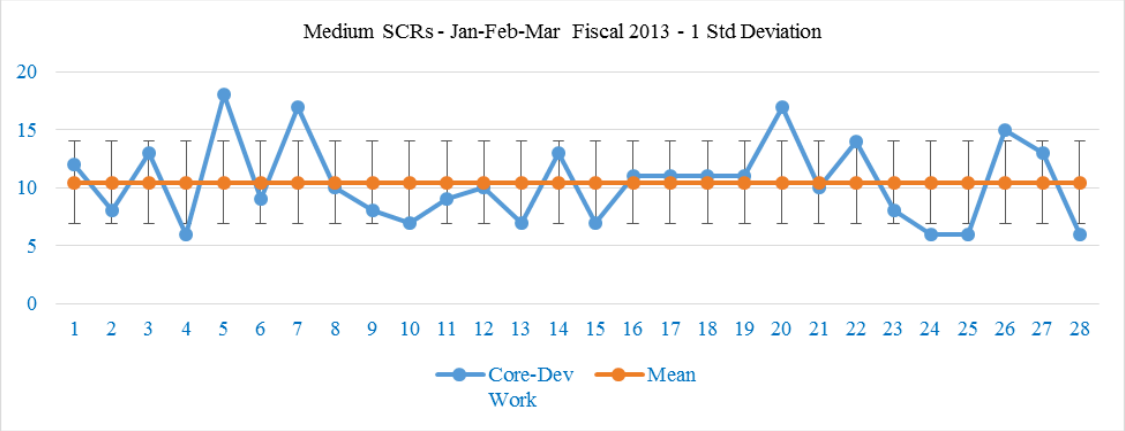


Figure 33 - Control Chart: Medium SCRs (1 Standard Deviation)

LARGE SCRs - The pattern of passing the four rules did not hold out for the “Large SCRs” by failing rule number 4.

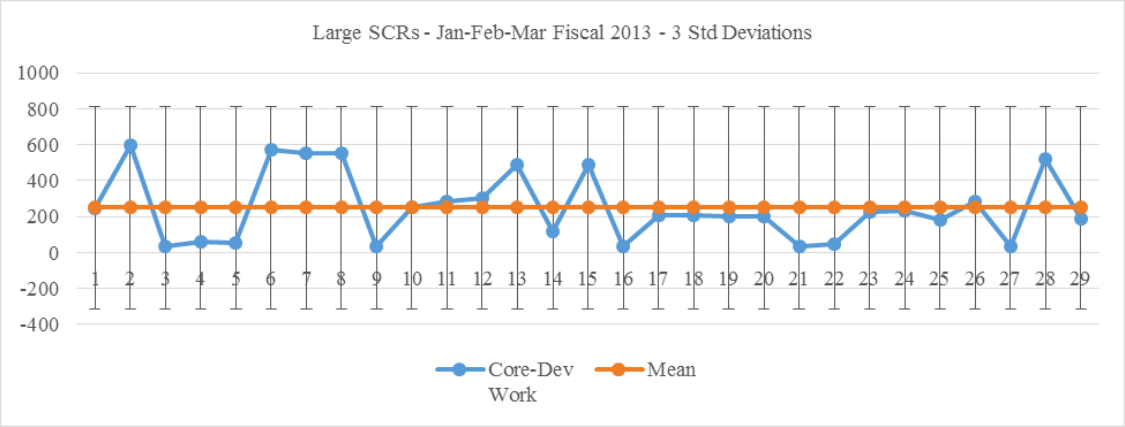


Figure 34 - Control Chart: Large SCRs (3 Standard Deviations)

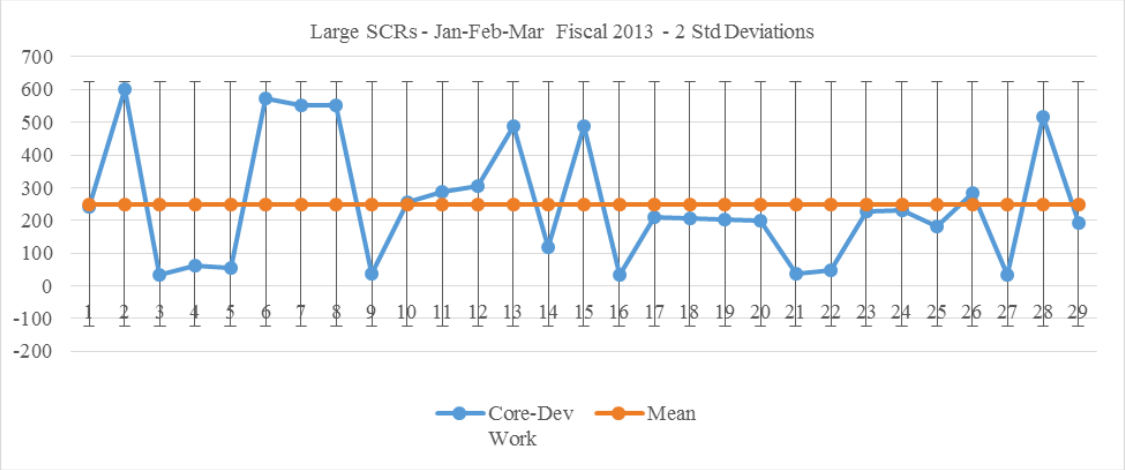


Figure 35 - Control Chart: Large SCRs (2 Standard Deviations)

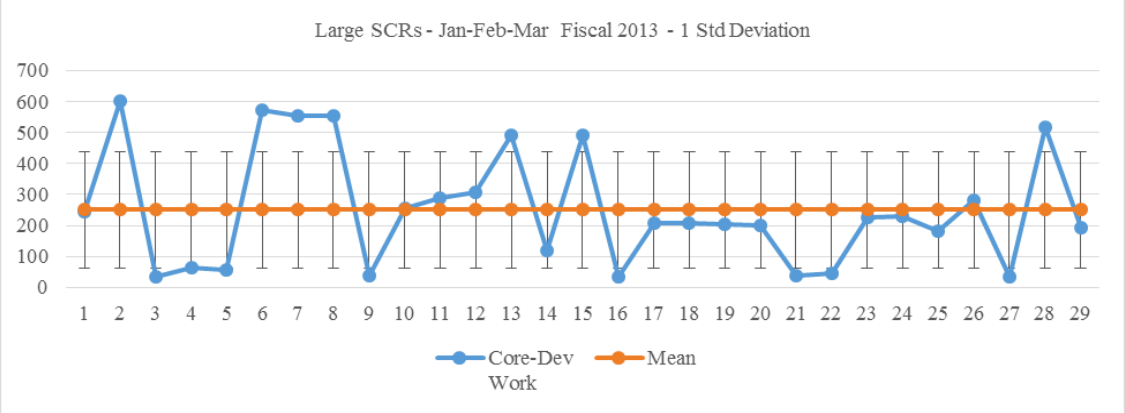


Figure 36 - Control Chart: Large SCRs (1 Standard Deviation)

There are ten points below the centerline for sequences 16 to 25. The SCRs involved in this variation are a mix of Baan and LN. There are two analysts involved and one prefers iteration (DPFR) and

other prefers waterfall (JPID). Per the developers for these SCRs, those that have zero code duration involve code changes that took less than 15 minutes each to complete.

ERP	SCR	Team	Design	Code	Test	Description
Baan	DPFR-94PMM2	MFG	32	0	1	MBDC - dfe request for a new report. Print Materials by Manufacturer.
LN	JPID-94QQD9	CIP	128	76	4	20130208 - LN CIP/FA: new custom tables
LN	JPID-94WQW6	CIP	126	76	4	20130214 - LN CIP/FA: display & maintain session for zhfam901 (Template Master)
LN	JPID-952LCN	CIP	129	69	4	20130218 - LN CIP/FA: new entry session for zhfam902 & zhfam903
LN	JPID-952R4P	CIP	121	75	4	20130218 - LN CIP/FA: load tipcs tables from zhfam tables
Baan	JPID-956LNZ	MFG	6	4	27	20130222 - AlphaCAM: chgs for new Canvas mfg line
Baan	DPFR-95AQE3	MFG	9	0	38	108 - Live - NCM by Production Order Geiger Fulton - tcqmscc227m000 - Logic change to include the Edgeband item and quantity from the PBOM when processing the NCM for the piece name.
Baan	JPID-95KPVG	MFG	39	151	37	20130307 - Pattern Fabric Cutting: database changes
Baan	JPID-95LJUB	MFG	38	192	0	20130308 - Pattern Fabric Cutting: Cutter.exe replacement in Baan
LN	JPID-95HMGW	CIP	105	72	4	20130305 - LN CIP/FA: new file extract from LN to Sage (FI043)

Table 14 - Rule 4 Failures

The work for the CIP team was part of the Accounts Payable go-live for LN. The analyst was brought in to cover for a shortage of analyst capacity and struggled with understanding the requirements due to a lack of functional knowledge. Eventually the analyst was taken off the work with design work subsequently reassigned to a consultant. This extension of design duration is more an indication of an issue with resource staffing within the project rather than an issue with the design process itself.

For the remaining SCRs that failed rule 4, two of the MFG were compliance related and the other two were not urgent. The zero day duration for testing came about because the testing was performed in the development environment off the "SCR clock". This is yet another example of where the duration data is not reliable for analysis use. The coding for the Pattern Fabric Cutting SCR also included some time for retrofit to reduce technical debt. While retrofit development isn't always performed in the interest of decreasing the development cycle time, ideally it would always be included.

Appendix G - Migration Errors

Although there wasn't sufficient data to measure and evaluate most of the CTQs, the customers of the System Administration team did have a CTQ for timely and accurate migrations. While the SCR doesn't directly contain migration information, each SCR has a task for each migration request. By extracting migration tasks, we can see and measure how many were accurate and on time. In general, the CTQs for migrations are being met.

System	Environment	Total Number of SCRs	Complete	Cancelled
Baan	**Test**	4757	4718	39
	Prod	2524	2504	20
LN	**Test**	860	1225	24
	Prod	726	234	6

Table 15 - Migration Count by Status

- Baan Test Migration: 2 cancels noted by Sysadmin as proactive
- Baan Production Migration: No explanation was provided for the cancellations
- LN Test Migration: 2 issues with migration requests caused cancellation
- LN Production Migration: 6 were cancelled and 2 had errors

System	Environment	Has Defect	No Defect
Baan	**Test**	0.40%	99.60%
	Prod	0.87%	99.13%
LN	**Test**	0.48%	99.52%
	Prod	1.67%	98.33%

Table 16 - Migration Defects

- Four migrations for LN had issues with the software dump
- Three involved a problem with a domain component
- One was a "leapfrog" where analysts were not in sync with one another
- Two of these four were cancelled and two were repaired

System	Environment	On Time	Late	Early	No Due Date
Baan	**Test**	76.28%	2.22%	21.50%	11.81%
	Prod	78.51%	4.99%	16.49%	0.80%
LN	**Test**	0.00%	0.00%	100.00%	36.27%
	Prod	87.61%	5.13%	7.26%	2.50%

Table 17 - Migration CTQ for Timeliness

- Production Migrations REQUIRE a due date.
- Counts with No Due Date reflect failures to migrate due to defect or proactive detection of a problem

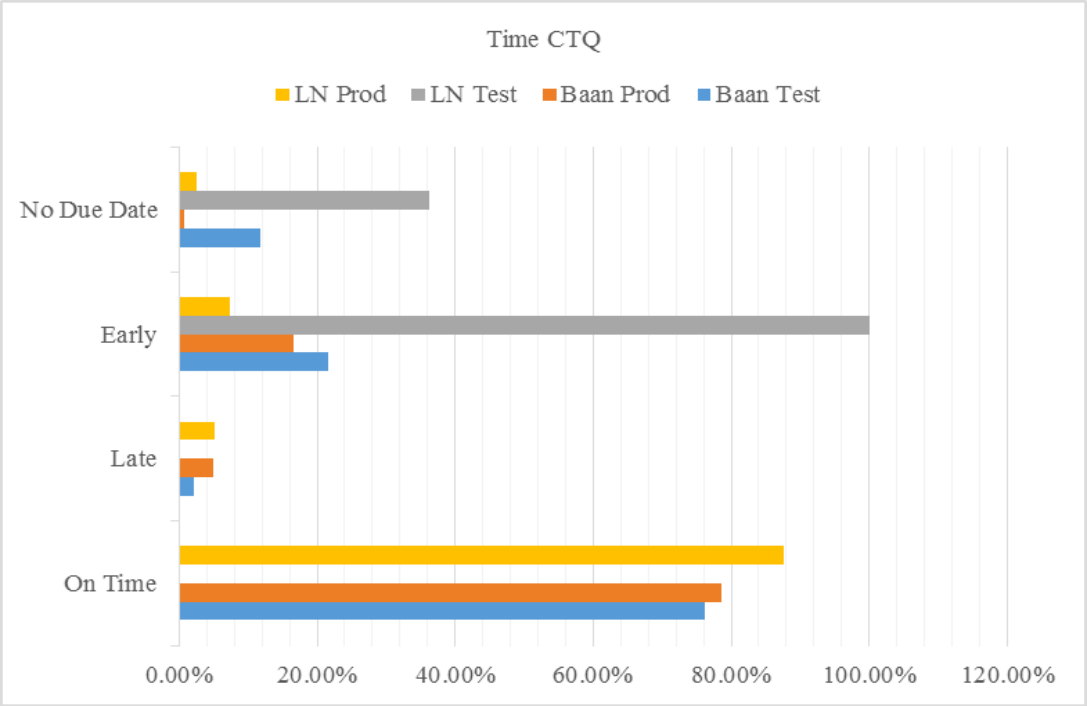


Figure 37 - Migration CTQ (On-Time)

Appendix G - Kanban

Given that the focus of the SCR system design and usage is to manage work flow for the software development process, it is fair to say that that the process more closely mimics the Kanban approach. A key flaw when using this approach is to push demand forward instead of pull demand. In pull demand, the downstream process reaches back for more work. If there is no work, then that process has wait time. With push demand, the upstream process pushes demand to the next step regardless of readiness. This can easily lead to the formation of bottlenecks.

In this experiment, ignore the SCRs in process prior to 2009 because we are concerned about flow. Specifically, this is a view of the weekly flow of SCRs between work stages. We can count how many SCRs entered the phase of work for a given week and how many exited in the previous week. When we add the previous work in process (WIP), we essentially have a view that is similar to very first chart of WIP from Figure 7 that takes us from viewing an annual measure to viewing a weekly measure of flow. This allows us to show whether or not there are bottlenecks causing delay of delivery.

Each of the following charts show us the relationship of new and WIP SCRs. The line charts also have bars to show one standard deviation from the centerline so that we can cross-check the search for bottlenecks. With the chart for SCR approvals, we see the influence of team based approvals used for LN.

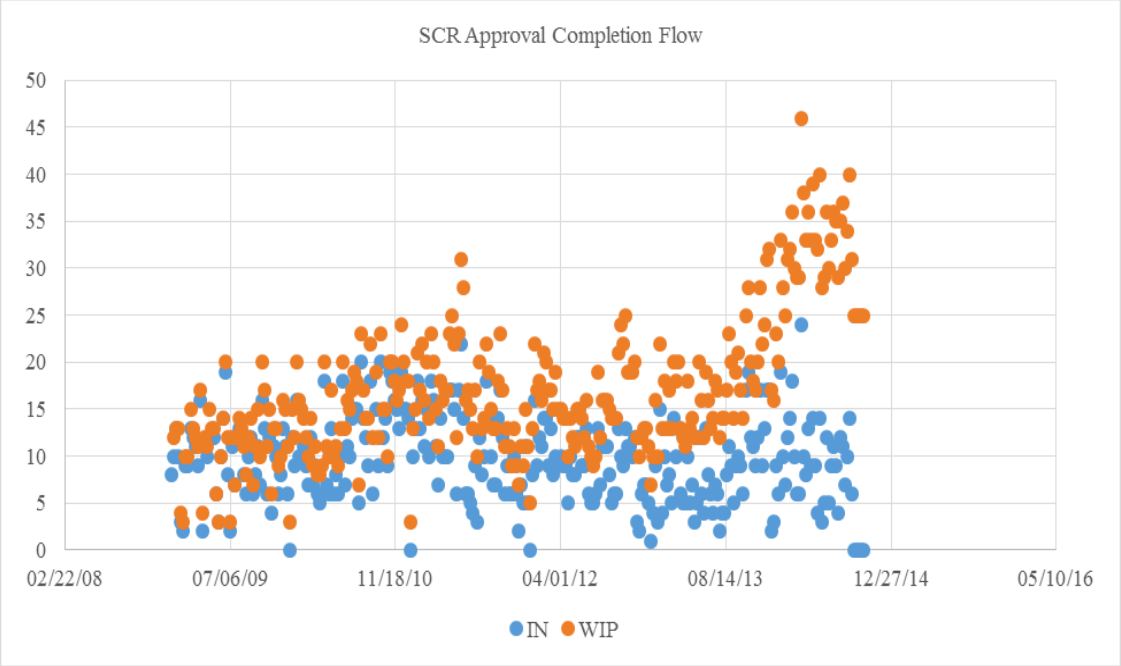


Figure 38 - Weekly WIP with New SCRs Flowing In

As in previous charts, we can see that SCRs in the design phase build a large set of WIP which is clearly a bottleneck. More research outside of the SCR data is needed in order to discern if this was due to resource constraints, delays in design decisions by business leaders, or uncertainly about how to technically approach the new functionality.

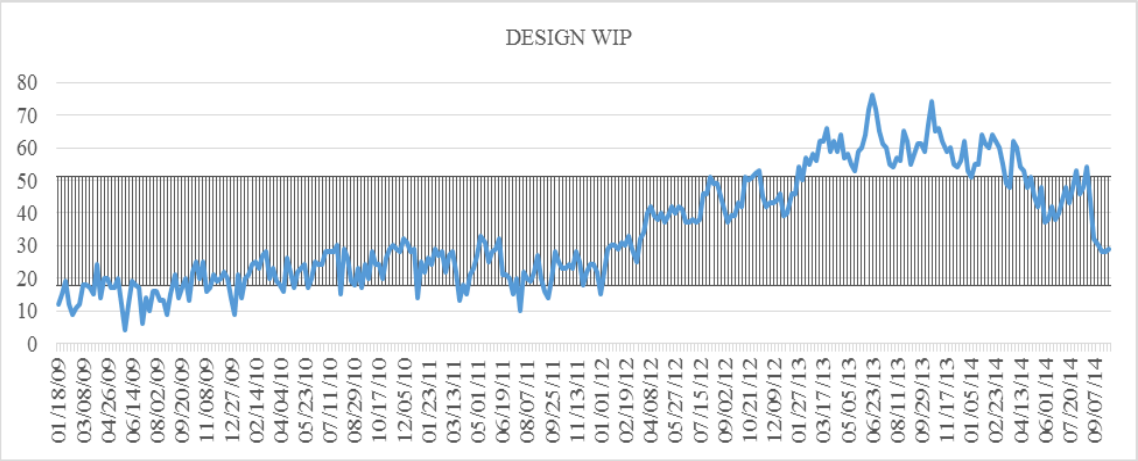


Figure 39 - SCRs Work in Process for Design Phase

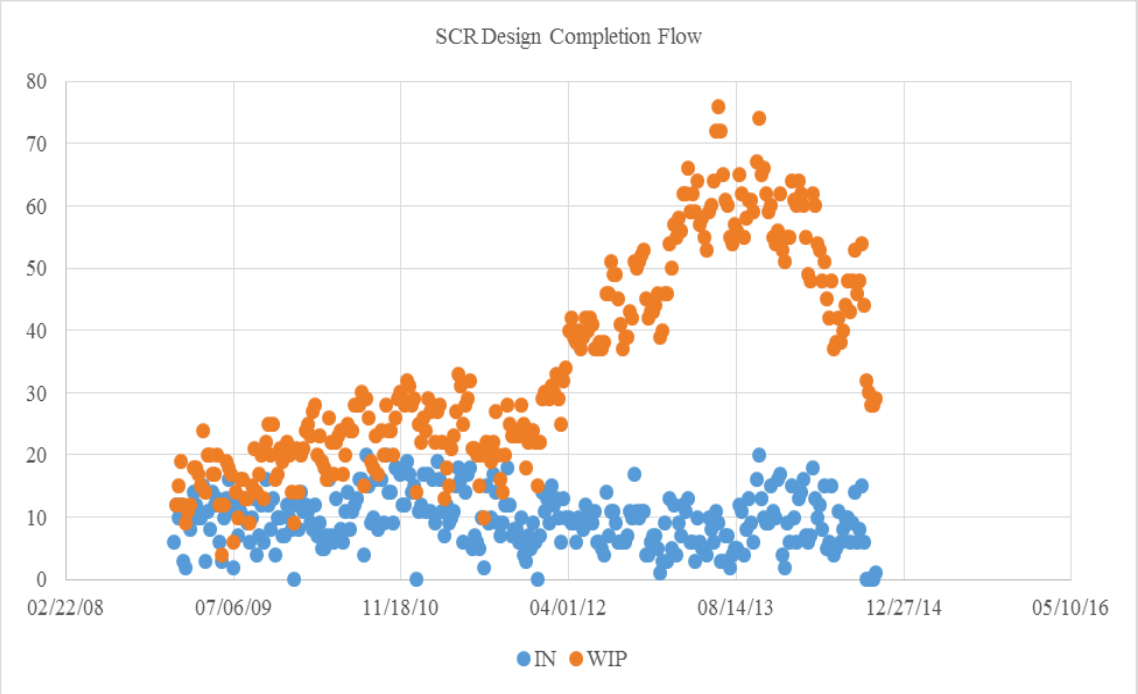


Figure 40 - Weekly WIP with New Design SCRs Flowing In

While the design bottleneck started in the spring of 2012, bottlenecks for code development have consistently peppered the timeline since the summer of 2009. Two additional developers were hired in 2012 and we see a brief decline that is unable to keep pace with design catching up in August. A more detailed analysis might find that the design spike causing the development bottleneck is a classic waterfall approach result. The design spike could also be due to a shortage of analysts on the LN project or from the priority reset for Project 8 from Table 1. We simply cannot confirm because there is no data to support defining a specific cause.

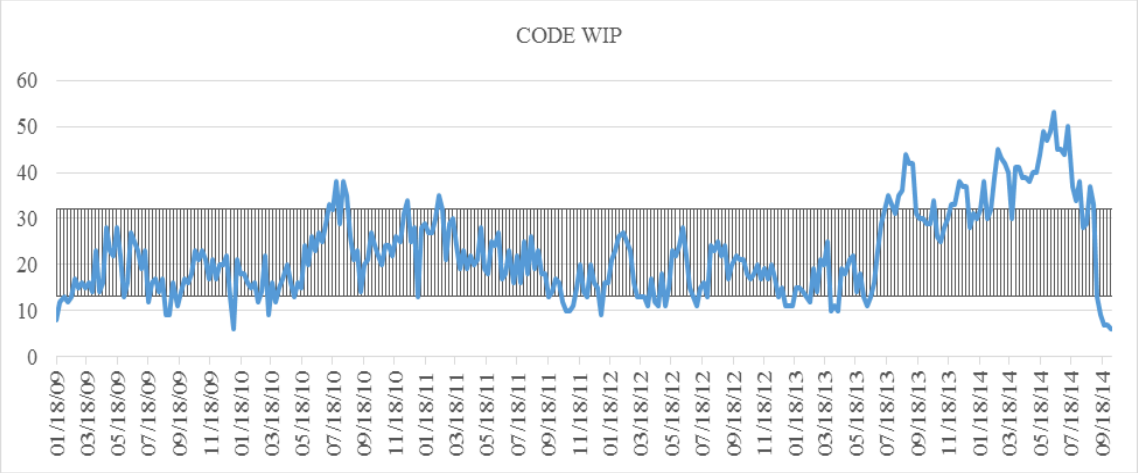


Figure 41 - SCRs Work in Process for Coding Phase

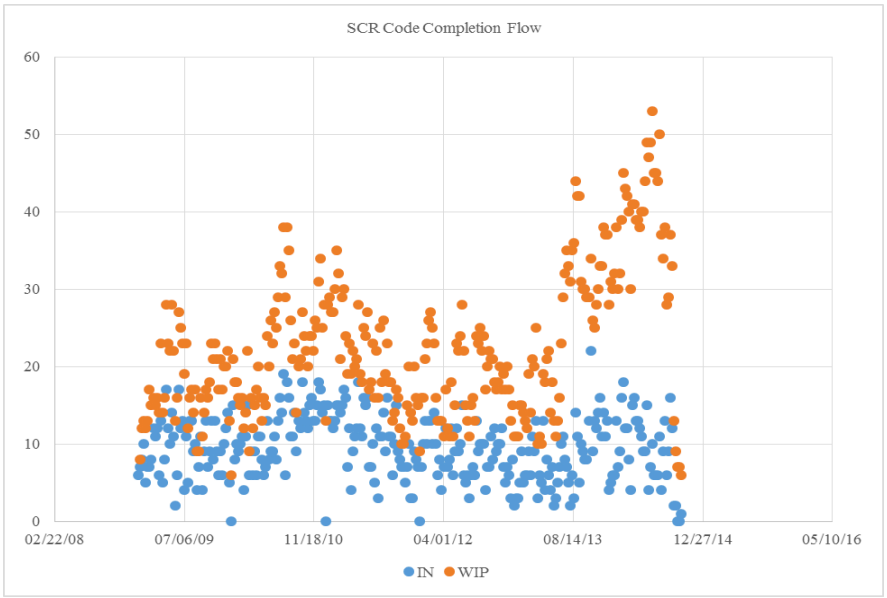


Figure 42 - Weekly WIP with New Coding SCRs Flowing In

The testing phase, combined with code fixes, shows steady and predictable growth since January 2012. The LN project started testing the third go-live segment in January of 2012 and was never able to keep pace with the steady stream of SCRs entering the testing phase in order to stabilize the work in progress. In fact, the testing SCRs continue to flow all the way to the October go-live date as analysts and users seek to make just "one more" change to tweak the software into perfection.

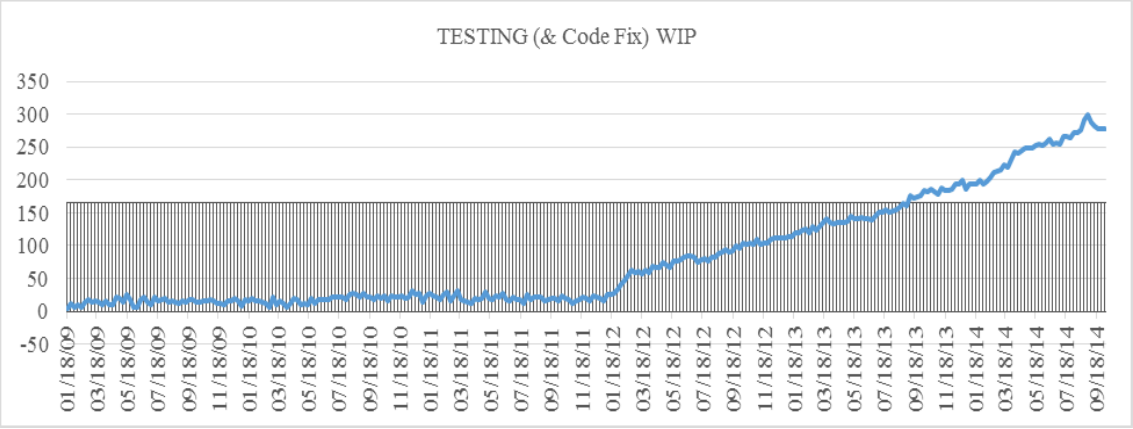


Figure 43 - SCRs Work in Process for Testing Phase

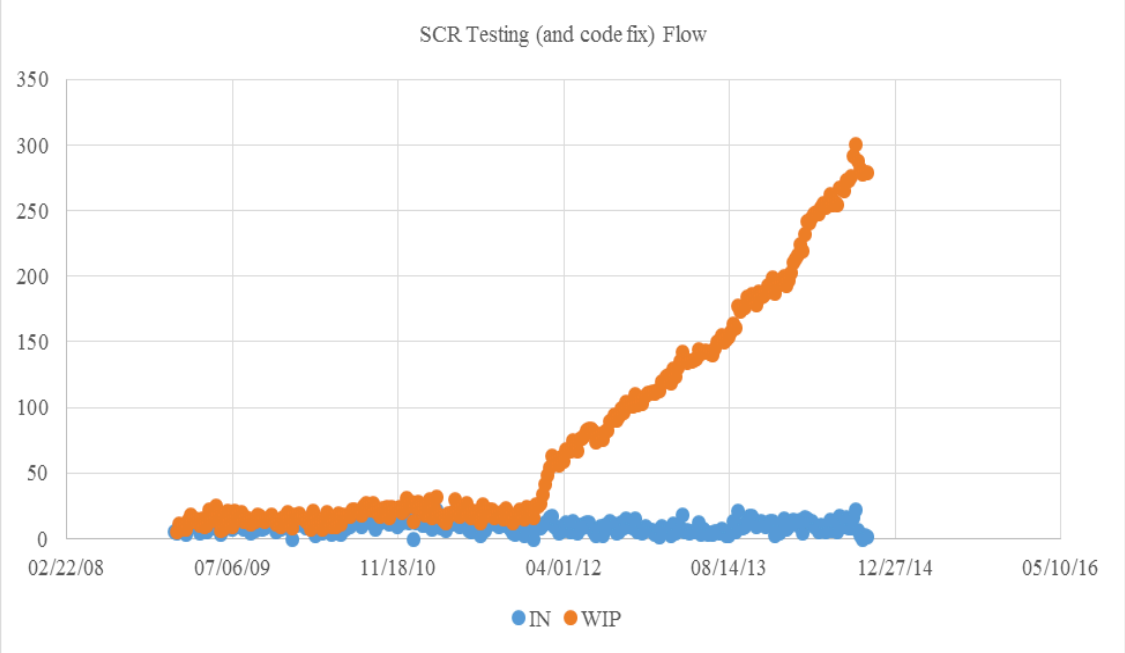


Figure 44 - Weekly WIP with New Testing SCRs Flowing In

It should be no surprise that software migrations for the production environment would experience bottlenecks slightly before a go-live date. September of 2013 and 2014 are for the second and third implementations for LN. However, the peak of bottleneck occurred in April of 2011.

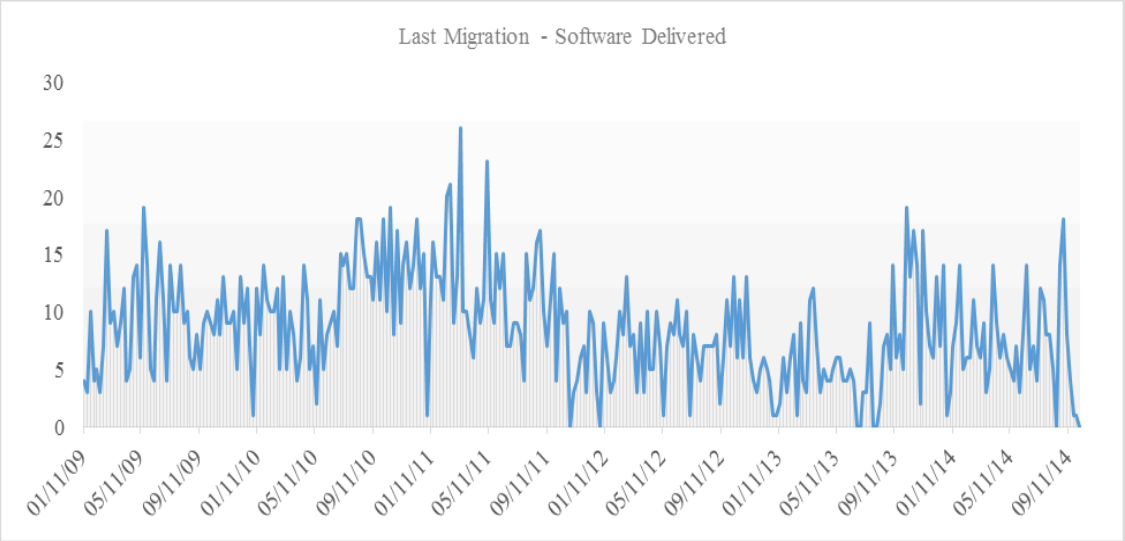


Figure 45 - Count of SCRs by Last Migration Date

To see if this spike was related to another project implementation, the specific SCRs were identified and are listed below. We see a combination of normal fixes and maintenance alongside a project for International Inventory Management. But the project is only four of the twenty-six migrations. It is possible that news of a looming LN project that led to a decision to hire two more developers may have indirectly incentivized the analysts to finish up existing SCRs before LN became a higher priority. However, there is nothing in the data to support this conjecture.

Request ID	Request Title
JPID-8ESP9Q	20110309 - BP Table Update Utility: fix for related field updates
MBAN-8EHS4T	BAAN-99591-20100228 - Receipt Delete - tdpur4320m000 - add logic for when to adjust inventory in prior to doing the receipt delete
MBAN-8EQPFV	BAAN-99587-20110307 - tdpurc290m000 - ASN cleanup – unreceived ASNs - hang on to for extended period for company 112
MBAN-8ERJNX	BAAN-99585-20110308 - MI005 - tdpscc119m000 - 2FO logic when checking for due date > current date - and running on a Sunday
BVDN-8D8PZD	20110118-Lam Line Optimizer extract for Phantoms
BVDN-8DHQF4	20110127-Veneer Intent Manifest - barcodes

BVDN-8ELPTS	20110301-171st Manifest fix timestamp link
BVDN-8EMLLF	20110304-NCM table fix missed sessions
BVDN-8ESPDQ	20110309-Intent Veneer Manifest report fix
DRUR-8E7KHH	Process Central Pricing Updates
DPFR-8EHS6B	nonul14 - label format change. tex1 - tiitm901 - chge to len 50.
DPFR-8EJQXB	Add more Order Types to the new Manifest logic - Fulton Manifest V2
DPFR-8EMLQM	ASN Session - tdpurc199m000 - need to add new conditions from the log to the report.
DPFR-8EQTR6	Add S10 and S12 to the summary sheet logic - for SOLIDS - tisfcc447m000.
JPID-8DPRVJ	20110202 - Supplier IBOM: modify processing to work for static items too
JPID-8EHR8V	20110228 - STD->Cust COPY (tipcfc229m000 & tipcs2230m000): warehouse & operation in copied BOM
JPID-8EKTGT	20110302 - IBOM: Signal Code code processing
JPID-8EMRQG	20110304 - Give Item Search Criteria: items with Signal Code = OBS to be treated as item Status "Obsolete - to be deleted"
MBAN-8ELQNP	BAAN-99610-20101109 - International Inventory Mgmt - Phase 18b - Visibility to the At Port Date - tdpurc423m000
MBAN-8ELQUS	BAAN-99610-20101109 - International Inventory Mgmt - Phase 20a - ASN manipulation - Copy ASN - default ASN header input field to current values
MBAN-8ELQXW	BAAN-99589-20110302 - tdpurc142m000 - reset ASN Header status when using the copy function
MBAN-8EMHTN	BAAN-99610-20101109 - International Inventory Mgmt - Phase 12e - tdpurc210m000/tdpur4130m000
MBAN-8EST2V	BAAN-99610-20101109 - International Inventory Mgmt - Phase 18c - Visibility to tiitm901.sxrf and tdpur949 PO xref - tdpscc420m000
RBLR-8EQR83	2011-03-07 MI066 - remove hard coded supplier logic and fix PPID technique
RBLR-8ERP67	2011-03-08 Cfg Request Maint
SEBG-8ESN2H	VOR -Update script to initialize variables before checking integration transactions

Table 18 - SCRs from Migration Top Spike

Appendix H - 5 Why's

One of the most powerful yet most difficult problem solving tools to use is the "5 Whys". The premise is that a team will brainstorm a particular question where the answer will lead to another question. The team has to be careful not to prejudge whether or not the answer will lead to the eventual root cause. The series of questions below represent the 5 Why session that was held with a subset of the DMAIC team. Not all team members were available or comfortable with performing the exercise.

Q: Prior to this DMAIC project, what methods have been used to solve the problem of needing to implement software projects faster?

A: There has been no formal effort to find a way to implement software projects faster.

Q: Why is that?

A: Nobody knows where to start. Also, the executives are satisfied if we make sufficient progress and want to make sure we don't generate risk from trying to go too fast.

Q: Then, why do business leaders complain that software delivery isn't fast enough?

A: The reason is that they are frustrated when their software changes are deemed lower priority for the organization and take longer to complete. We can't promise when we'll be able to get their software changes completed because we have to focus on priorities.

Q: Why can't we estimate the non-priority work? It's not unusual to have wait times where we could do some other work.

A: We don't really know how long the effort will be. Development is usually easy to estimate within a week or two, but design and testing are unpredictable. We might end up wasting our time.

Q: Why can't we estimate the design and testing work?

A: We have always struggled with it. Because so many things can change that make the estimate wrong. Priorities get shuffled around and resources are shifted. There's really no good way to predict when we'll be able to finally finish the SCR.

Q: Why isn't there a good way to predict delivery?

A: We simply can't "see" what's on the horizon let alone be able to see how we're progressing with the priority SCRs. The system doesn't have a way for us to do any reporting.