HERREN SYSTEMS THINKING. SMARTER SOLUTIONS.

Program Analysis & Cost Engineering

Dynamic Help Desk Modeling

A SCEA Case Study

Steve Sheamer, Nolin Huddleston, Sujoy Roy

11

Dynamic Help Desk Resource Modeling

Abstract

This paper discusses the concept of dynamic help desk resource modeling for a major Information Technology Enterprise system. The methodology discussed in this paper utilizes regression analysis to project the resource requirements for a three-tiered help desk system based on a number of factors including system user population and maturity. This study utilizes data and direct observations from the implementation of a help desk system serving a large Enterprise Resource Planning (ERP) system within the Department of Defense (DoD). The ERP system was deployed to a number of client organizations over time, which resulted in a discrete increase in ERP system user populations with each deployment.

The help desk system studied utilizes a three-tier structure for ticket routing and processing. The following establishes the typical escalation process for ticket items entering the help desk; Tier 1 answers the initial call from the user, creates a ticket for tracking, performs the initial triage, and escalates the issue to the appropriate next level (Tier 2 or Tier 3) as required. This paper argues that help desk staffing requirements are a function of more than just total number of system users – other factors include user activity level, user maturity, system maturity, system complexity, and implementation stage.

The paper explains a methodology to understand the shifts in help desk ticket volumes and resource requirements over time as the users, system, and help desk staff mature. The model discussed in this paper projects help desk requirements with demonstrated certainty, giving decision makers sufficient information to project future costs and ensure optimum funding staffing levels.

1.0 Introduction and Background

This paper discusses the development and practical application of a model that was developed to project the number of resources required to support the help desk operations for a DoD Enterprise Resource Planning (ERP) Program. As is the case with many large IT implementations, the implementation strategy utilized "rolling wave" deployments where the system was deployed in multiple phases spanning multiple years. With each deployment additional users were added to the system, requiring sufficient help desk support staff to respond to user issues in a reasonable amount of time. In order to ensure that the help desk was appropriately staffed and funded, the program had to develop resource estimates over 2 years in advance of each deployment. To make things more challenging, the estimates had to be developed with limited historical data and frequently changing program requirements. The objective of the model was to estimate the number of resources required by fiscal year, rather than by hour or day, so a macro-level modeling approach was used.

The help desk used to support this system consists of multiple tiers with varying levels of expertise. The first tier of the help desk, Tier 1, is a 24x7 call center that answers the initial call from the users and resolves simple issues such as password resets. If the issue is too complex for the Tier 1 help desk to resolve, a ticket is created and is routed to either the Tier 2 or Tier 3 help desk. The Tier 2 help desk responds to "how-to" questions and technical issues such as server resets, system outages and application errors. The Tier 3 help desk responds to the most complex issues that may require changes to the underlying application, modifications to data in the system, or business process changes. Figure 1 shows the schedule of each Go-Live, with the number of new users being added and cumulative numbers of system users over time.

Deployment	# of New Users	Cumulative Users	Go-Live Date
1	15,989	15,989	12/1/2007
2	10,385	26,374	12/1/2008
3	10,135	36,509	12/1/2009
4	7,667	44,176	10/1/2010
5	22,382	66,558	10/1/2011
6	2,753	69,311	10/2/2012

Figure 1: System Go-Live Schedule

The traditional approach to this problem would involve using either a step function (e.g. 1 additional help desk FTE for every additional increment of 500 users) or a parametric estimate (e.g. total number of system users multiplied by x%). For the system that was studied in this example, both of these approaches would have underestimated the number of resources required soon after implementation when help desk call volumes are higher, and overestimated the number of resources required after implementation when users have adjusted to the new system and the system itself has matured. The result of such an approach would be insufficient resources in the near-term and overinflated sustainment costs in the long-term. This fixed capacity approach is shown in Figure 2.

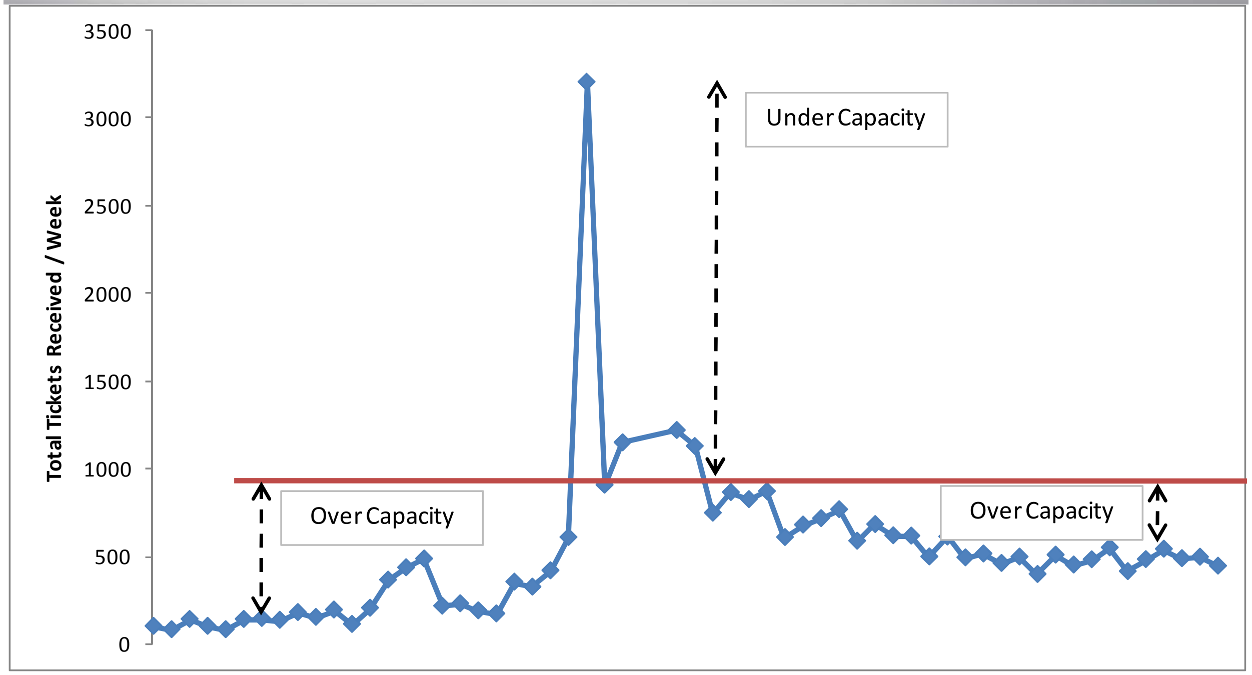

Figure 2: Fixed Capacity versus Help Desk Tickets Received

2.0 Data Observations

Figure 3 shows the number of help desk tickets that were submitted each week for the 1st of the 6 planned system deployments. The system went "live" in December of 2007 and the spike in help desk tickets occurred the week of system go-live. In analyzing the data from the initial deployment, it was clear that there were 3 distinct periods that characterized the behavior of the data:

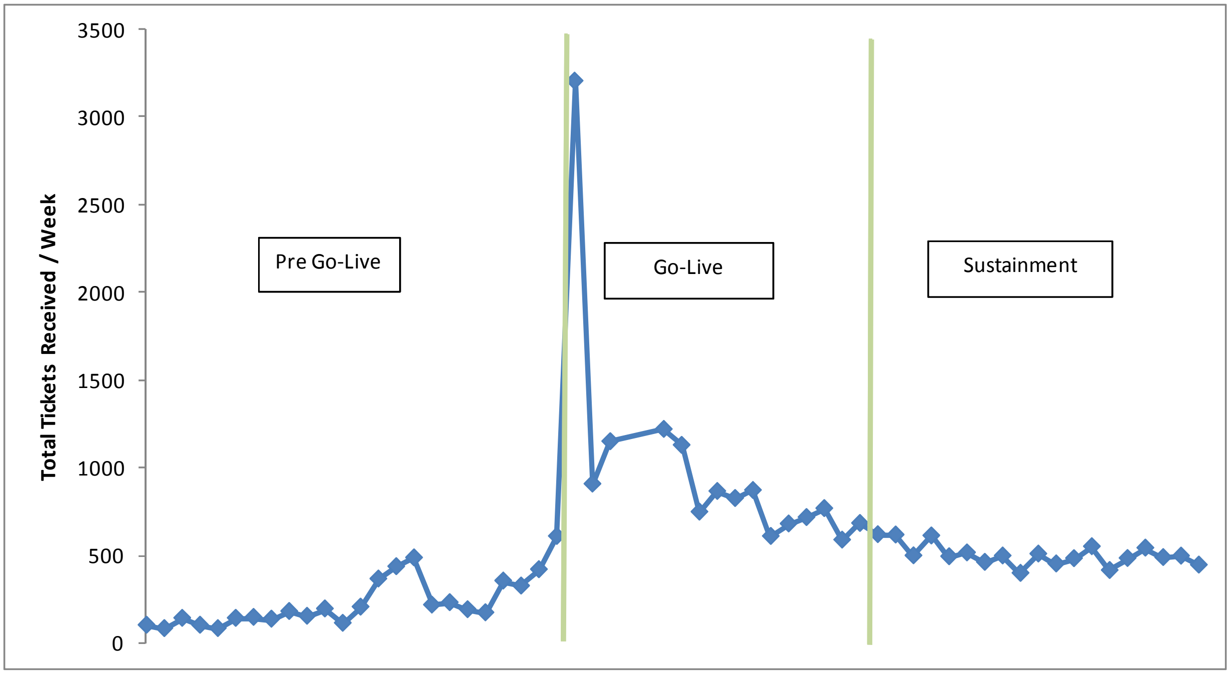

Figure 3: Data Observations

Phase I: Pre-Go Live

During this phase of the deployment, the volume of help desk tickets is relatively low but increases as the go-live date draws closer. It might seem odd that help desk tickets would be submitted prior to system go-live but there are several reasons why this occurred for the system being studied. User accounts were established prior to go-live and users were required to log into the system and validate that they had the proper access and roles. Any changes to roles or access required a help desk ticket. Many users also had to complete web-based training and had issues either accessing the web-based training or receiving credit for completing training. Finally, there were some help desk tickets that were submitted by the implementation team to resolve technical issues prior to go-live.

Phase II: Go Live

The go-live phase is characterized by a large spike in help desk tickets the week of go-live, followed by a steep reduction in the number of tickets. As we will discuss later in the paper, the first deployment had the most dramatic spike in tickets during the week of go-live due to the lack of overall system maturity.

Phase III: Sustainment

The final phase of the deployment is characterized by a gradual but continued reduction of help desk tickets over time as the users adapt to the system and the system itself matures over time. The observed data from the first 4 system deployments demonstrated that the transition from the high ticket volume go-live phase to the lower volume sustainment phase occurred somewhere between 12 to 16 weeks after the system go-live. The exact timing of this transition does not have a substantive impact on the resource projections from the help desk model.

2.1 Data Analysis and Modeling Techniques

After plotting the data and identifying the three distinct deployment stages, the team needed to develop a method for estimating the required help desk staffing for the five remaining deployments. At the time, the only data available about the remaining deployments was the number of system users and the planned go-live dates. The team also interviewed technical subject matter experts who believed that many system defects and training issues experienced during the first deployment would be addressed prior to the following deployments and would reduce the overall number of help desk tickets. These SMEs also believed that the help desk resources would become more efficient at closing tickets over time as they gained familiarity with common system issues and the appropriate resolution.

In order to determine the number of resources required to support the help desk, the model had to predict the number of incoming tickets and the expected closure rate for each ticket. The number of incoming tickets was believed to be primarily driven by the number of users, the maturity of the system, and the deployment phase (pre go-live, go-live, or sustainment).

Since the number of incoming help desk tickets is largely dependent on the number of users in the system, the first step in the analysis was to normalize the data based on the total user count. This was accomplished by dividing the total number of tickets submitted by the total number of users to obtain the number of tickets submitted per week per user. Figure 4, Figure 5, and Figure 6 show the incoming tickets at the three different stages, Pre Go-Live, Go-Live, and Sustainment.

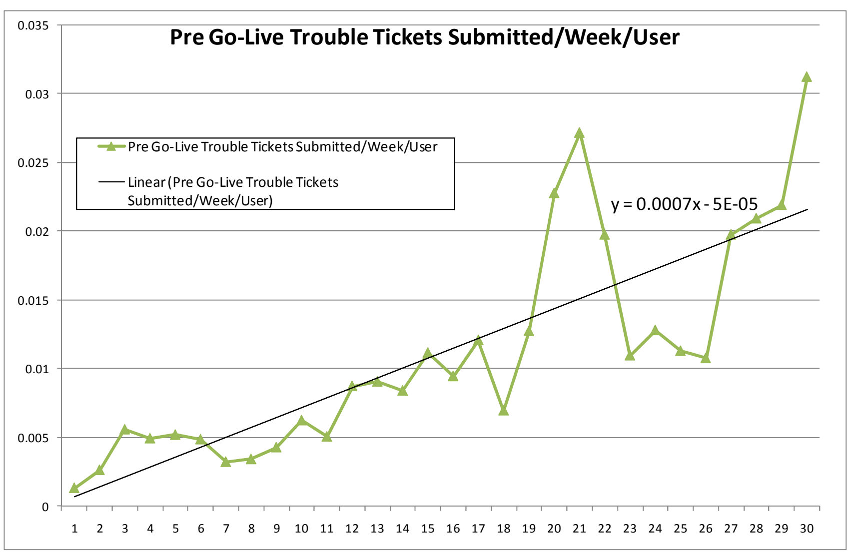

Figure 4: Pre Go-Live Ticket Behavior

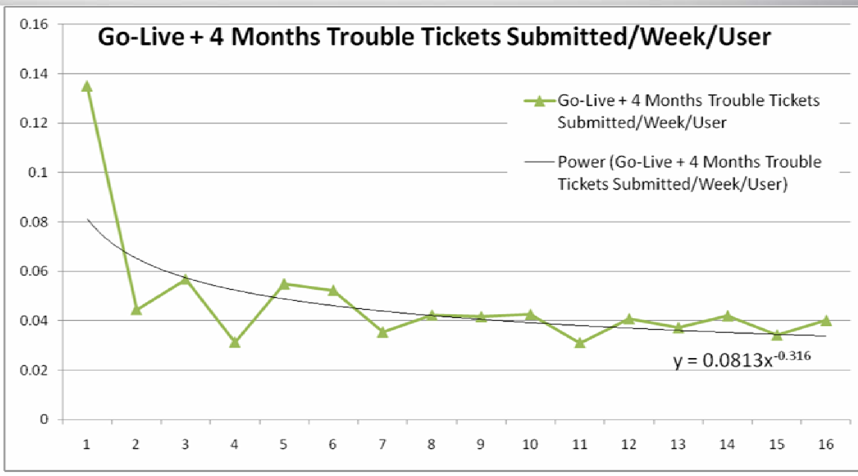

Figure 5: Go-Live Ticket Behavior

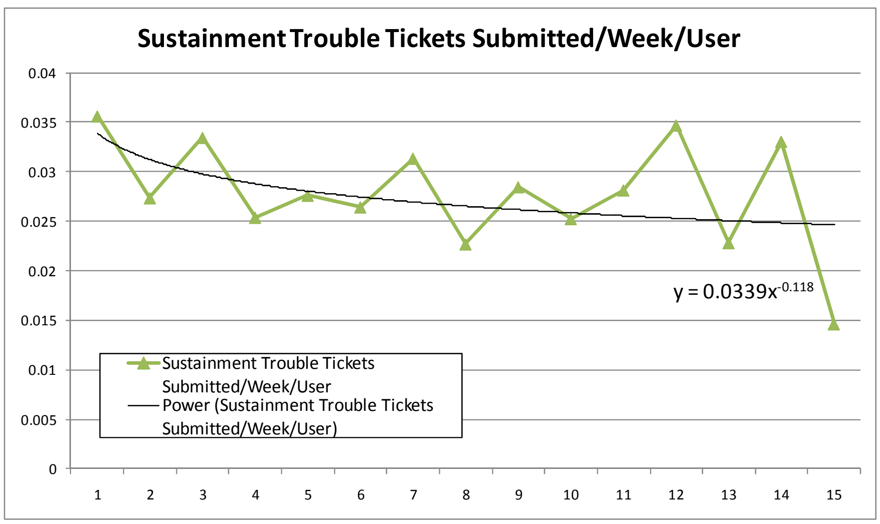

Figure 6: Sustainment Ticket Behavior

After the data was normalized and plotted for each stage of deployment, a regression equation was used to determine the predicted number of help desk tickets that each user would submit per week based on the number of weeks prior to or after the go-live date. The regression equations that were used for each stage of the deployment are shown in Figure 7. In all cases, y represents the projected number of tickets that will be submitted each week per user.

Deployment Stage	Best-Fit	Equation	x	Range of values for x
Pre-Go Live	Linear	y = 0.0007x - 5E-05	The number of weeks after the start of the pre go-live period, where x=1 represents the start of the pre go-live period and x=30 represents the week prior to the go-live date.	1 to 30
Go Live	Power	$y = 0.0813x^{-0.316}$	The number of weeks after go-live where x=1 represents the week of go-live and x=16 represents the final week of the go-live phase.	1 to 16
Sustainment	Power	$y = 0.0339x^{-0.118}$	The number of weeks after the start of the sustainment phase where x=1 represents the 17th week after go-live and x=n represents the n^{th} week of the sustainment phase.	1 to n

Figure 7: Regression Equations for Each Deployment Stage

In order to determine the number of projected tickets in any given week, the model multiplies the total number of users by the corresponding equation in Figure 7 based on the go-live date. As discussed previously, another variable was added to account for the projected reduction in help desk tickets predicted by the technical SMEs based on system maturity.

We can represent the equations associated with each deployment phase using the variable P_i where i equals 1 for the Pre-Go Live stage, 2 for the Go-live stage, and 3 for the sustainment stage. Pi is determined by comparing the go-live date to the week being calculated in the model. It is set to equal zero for any week in the model that is more than 30 weeks prior to the go live date. All variables required to project the number of tickets per week are described below:

T_n = Projected number of tickets per week

P_i = Deployment phase equation

d = Deployment 1 to 6 as depicted in Figure 1

U_d = Total number of users per deployment (See Figure 1 for user counts)

S_d = The estimated percentage reduction in the number of tickets from the initial deployment based on system maturity where $0 < S \leq 1$ for deployment, d.

$$T_n = \sum_{d=1}^{6} (U_d * P_i * (1 - S_d))$$

In addition to projecting the number of help desk tickets that will be submitted each week, the team also determined the service rate for closing help desk tickets. Since the desired output of the model is the number of help desk FTEs required to support the users, the team determined the average number of help desk tickets that were closed by each help desk FTE per week. This was calculated by simply dividing the number of help desk tickets closed each week by the number of FTEs that were working on the help desk during that week. We will represent this service rate (expected # of tickets closed per FTE per week) by the variable R.

Based on feedback from the technical team, the team also incorporated an improvement in service rate over time to account for the anticipated learning curve for the help desk resources. So the anticipated service rate is represented by the below equation:

$$\text{Adjusted Service Rate} = R * (1 - L_n)$$

Where L represents the estimated percentage increase in the service rate for week n and $0 < L \leq 1$.

Finally, the projected number of required FTEs for week n is calculated by dividing the projected number of tickets (T_n) by the adjusted service rate

Required Help desk FTEs for week $n = \frac{T_n}{R * (1 - L_n)}$

2.2 Calibrating the Model

In order to ensure the model provides accurate estimates of the resource requirements, the model was calibrated as actual data became available. The most important variable that needs to be calibrated is the system maturity factor since this has a significant impact on the total volume of projected help desk tickets. For the 2nd deployment, the technical SMEs had predicted a 35% reduction in the number of tickets submitted per user; during the first few months of the pre-go live stage, it was determined that the actual reduction was closer to 50%. Based on this early data, the system maturity factor for deployment 2 (S_2) was adjusted from .35 to 0.5. After this early adjustment was made, the actual number of tickets received in the go-live and sustainment phases were within 5% of what the model projected.

The service rate improvement factor should also be continually monitored and updated since this will impact the total number of projected help desk FTEs output from the model. The actual annualized increase in ticket closure rates was ~7% from year 1 to year 2 and 2% from year 2 to year 3 compared to the SME estimate of a 5% improvement each year.

3.0 Discussion of Variables and Cost Drivers

The initial help desk model was developed in late 2008, shortly before the 2nd system deployment. Over the next year, the model was continually adjusted and calibrated to improve the accuracy of the output. The underlying equations and approach described in section 2.1 remained unchanged, but a few of the variables were adjusted over time based on additional data analysis. In this section, we will discuss some of the variables that were adjusted and other observations that need to be considered in the development of comparable models.

User Counts

The initial model projected the number of help desk tickets based on the total number of users with system accounts, regardless of activity level. This was later changed to the number of users that were actively logging in to the system each week and performing transactions. The latter proved to be a more accurate predictor of projected help desk ticket counts.

For some systems, it might be beneficial to go one step further and base the user counts on "super users" or highly active users that are performing a large number of transactions each week. Although a detailed discussion is beyond the scope of this paper, the team found that a relatively small percentage of the total users submitted a large portion of the help desk tickets.

User Maturity

As predicted by the regression analysis for the sustainment phase, the number of help desk tickets submitted over time continued a gradual but sustained decline after go live. We believe that this is driven by two factors – user maturity and system maturity (the latter will be discussed separately). As users gain more experience with the system, they are less likely to call the help desk for "how-to" questions, access or account issues, or other simple problems. This effect is amplified by users within the organization who become experts in resolving system problems; users will begin turning to these local experts in their organization first before calling the help desk. Unless there is a high level of turnover in the user population, this user maturity should continue to reduce the number of help desk tickets that are received over time.

System Maturity

As predicted by the technical SMEs, there were far fewer tickets submitted by the users in the 2nd deployment than the first deployment. There was also a decrease in the number of tickets that were submitted by the users in the 3rd deployment when compared to the 2nd deployment. Figure 8 shows the number of tickets that were submitted for each deployment after normalizing the data based on active users and aligning the go live peaks for the first 3 deployments. There are several factors that contribute to this reduction in tickets over time: improved user training based on feedback from earlier deployments, resolution of bugs and system defects identified in earlier deployments, and application of lessons learned by the deployment team.

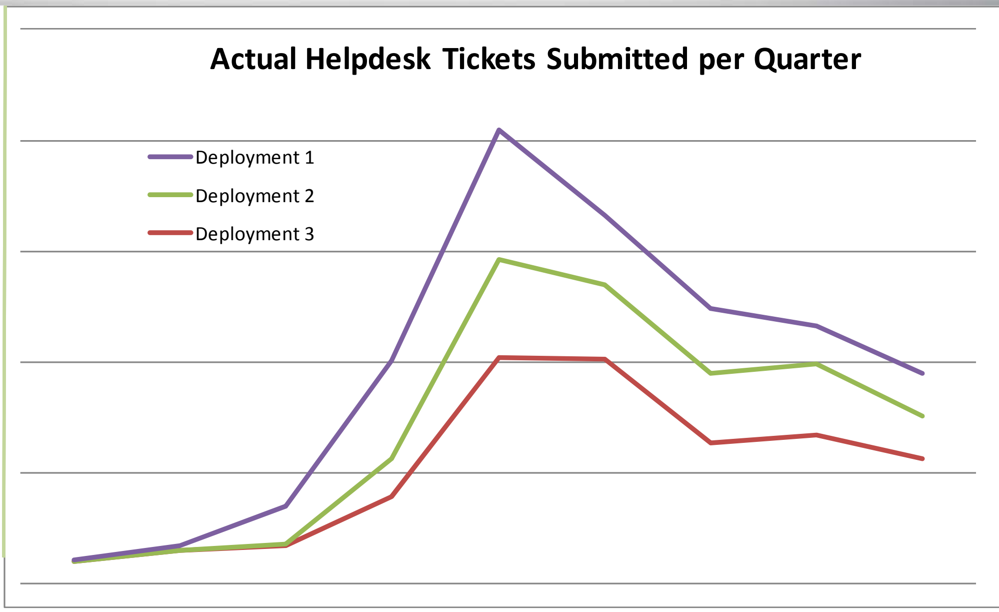

Figure 8: System Maturity

New Functionality and System Upgrades

The reduction in help desk tickets that results from user maturity can be interrupted by the incorporation of new functionality or expansive system upgrades that change the user interface. This phenomenon is demonstrated in Figure 9, where the tickets were gradually decreasing over time until new functionality was implemented in the March 2010 timeframe. As shown in the graph, even though the users were familiar with the system, the incorporation of new functionality caused a spike in the number of help desk tickets.

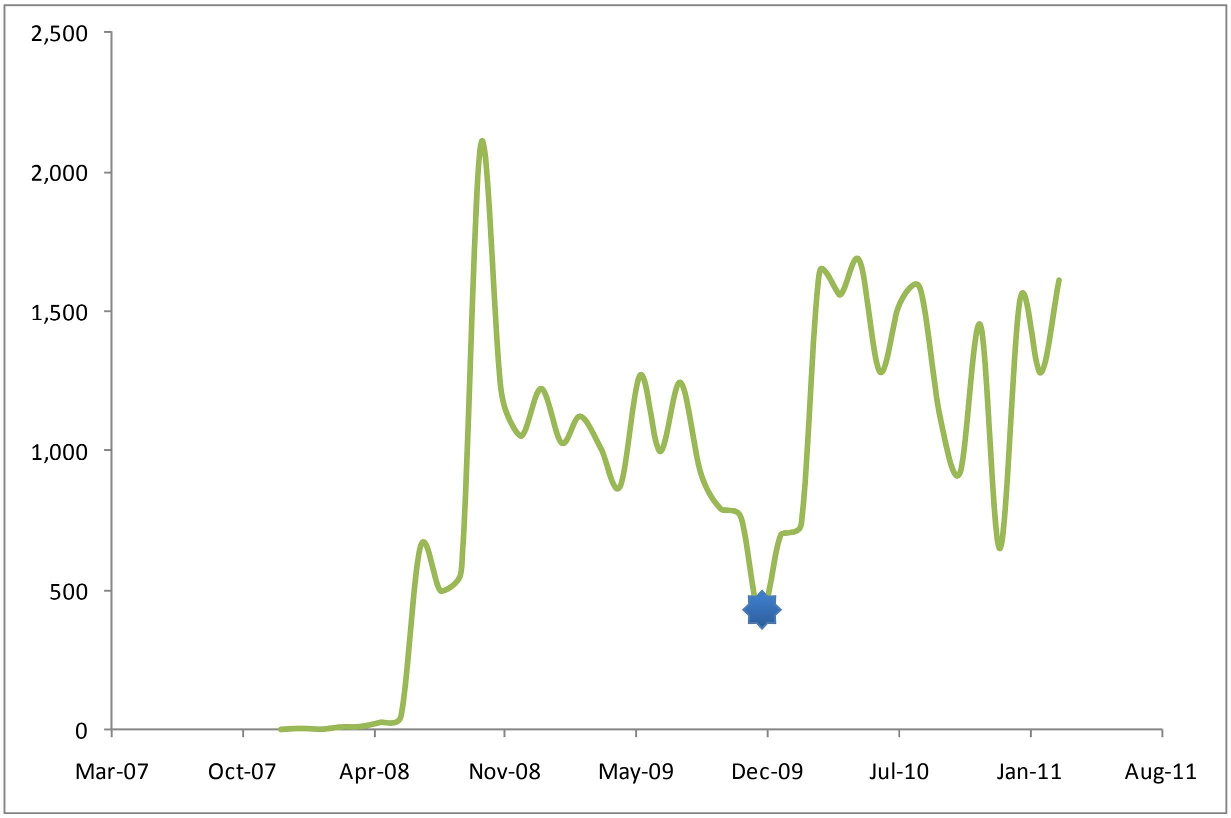

Figure 9: New Functionality

System Complexity

For the model discussed in this paper, the same system was deployed in multiple waves. Since the system complexity did not vary from one deployment to the next, the data from the initial deployment proved to be a valid predictor for future deployments after adjusting for user counts and system maturity. However, if this same model were to be used for a system that was more or less complex, the output would likely be a poor predictor of the resource requirements. So any model that is being developed to represent multiple system types should have an added variable to account for system complexity.

System complexity could be determined by looking at the following characteristics of the system:

- How much training is required for a user to be proficient in the system? This could be measured in terms of the number of hours required or the total number of courses on the system.
- How unique is the system relative to industry / commercial standards? Is the user interface simple and intuitive or unique and difficult to navigate?
- What is the complexity of transactions being performed in the system? For instance, is it a simple e-mail system or a complex procurement system?

3.1 What do we mean by Dynamic?

A help desk for an enterprise IT system is part of a highly dynamic system that is affected by multiple variables, many of which were discussed in this paper. Any help desk model that assumes a steady-state number of support resources over time should be questioned, considering that user maturity, system maturity and service rate improvements should all combine to reduce the required support resources over time. Conversely, the incorporation of new functionality or system upgrades may increase the number of required support resources.

The model discussed in this paper allows the following input variables to be easily changed and instantaneously updates the projected number of help desk resources required.

- User Counts
- Deployment Dates
- System Maturity Factor
- Baseline Service Rate
- Service Rate Improvement Factor
- Implementation of new functionality / upgrades (treated as mini-deployments)

Based on the changes in the above variables and the dynamism of the regression equations, the number of required help desk resources constantly changes over time as shown in Figure 10.

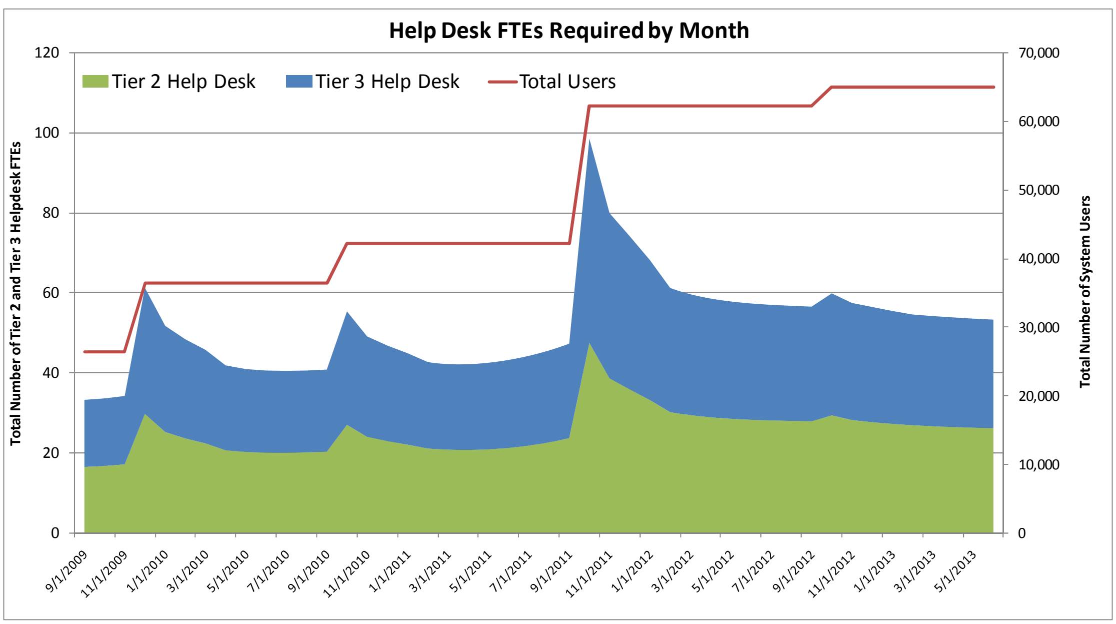

Figure 10: Help Desk FTEs Required by Month

One thing that is immediately obvious from the model output is the large spike in resource requirements during the week of go-live. Although some of this can be mitigated through advanced planning, resource surges, and overtime, it is not realistic to fully staff the help desk to the meet the requirements around go-live. In reality, during the go-live periods, high priority tickets are addressed first and lower priority tickets are placed in the queue to be worked as the number of new help desk tickets being submitted decreases over time.

The problem with using a static modeling technique like a step function to model help desk requirements is clearly evident in Figure 11 and Figure 12. Notice that early on, the step function understates the number of resources that are required but as the users and system mature over time, the step function dramatically overestimates the number of resources that are required. Figure 12 also shows that the number of users that can be supported by each FTE will increase over time rather than remaining static.

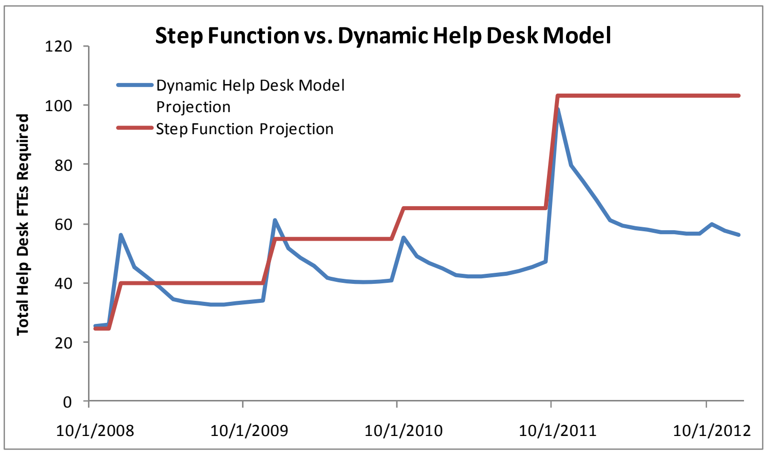

Figure 11: Step Function versus Dynamic Help Desk Model

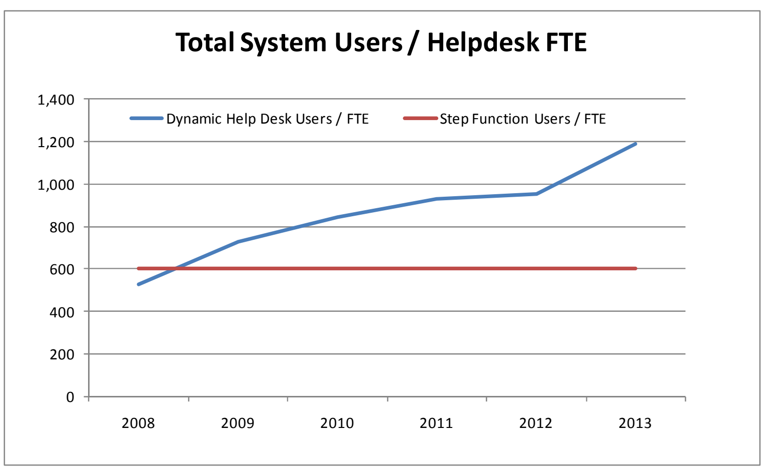

Figure 12: Users to Help Desk FTEs

4.0 Results and Benefits of the Modeling Approach

The initial funding for the help desk was based on SME input and assumed that the help desk staff could be level loaded over time, even as additional users were added to the system. The assumption was that the increased tickets from new users would be completely offset by the decrease in tickets from existing users. As a result of budget pressures, program leadership made additional reductions to the SME input. Without a tool to understand the implications of these budget reductions, the program leadership had unwittingly created a significant funding shortfall that would have resulted in major user support issues, including long wait times to resolve tickets.

After the help desk model was developed, program leadership was able to quickly understand the additional resources that would be required in the long-term to support the user community. They used the output from the model to convince decision makers to provide additional funding while clearly articulating the impacts if no additional funding was provided.

The model was also used to recoup funding when customers added additional users to the system. The model provided the program with a reliable and defensible method for identifying the incremental cost for each additional user over the life of the program. When customers requested additional user accounts, the program was able to bill the customer for the long-term help desk costs associated with the additional users. Prior to implementation of the model, the program only billed customers for additional costs associated with required user licenses.

As a result of the improved projections from the model, the program successfully transitioned from being significantly underfunded to having sufficient funding and resources to support the user community and meet all Service Level Targets. The model proved to be a key enabler in ensuring the overall success of the system implementation.

5.0 Additional Considerations and Future Work

Although not discussed in detail in the body of the paper, the team also incorporated risk into the model utilizing Crystal Ball to run Monte Carlo simulations and develop an expected range of FTE requirements rather than a discrete estimate. Although the model proved accurate once it was calibrated with data from the pre-go live stage, the initial input variables were adjusted to some extent for all of the deployments once actual data was available. Given the uncertainty around the input variables in the model (active users, system maturity factor, service rate improvements, etc.), the risk adjusted model output is preferred over the discrete estimate for budget forecasts.

Future considerations for improving this model include a more detailed exploration of time series data modeling using techniques such as Autoregressive Integrated Moving Average (ARIMA), exponential smoothing, or moving averages. Early explorations of these concepts have yielded promising results, though there is more to be done to achieve a full solution with this approach. Figure 13 shows projections that were created using Crystal Ball software that closely aligned with the regression equations that were described in Section 2.1.

Developing the projections using this ARIMA technique took significantly less time than developing the regression equations and subsequent model described in this paper. However, more research is required to determine if this technique would improve the accuracy of the model's projections.

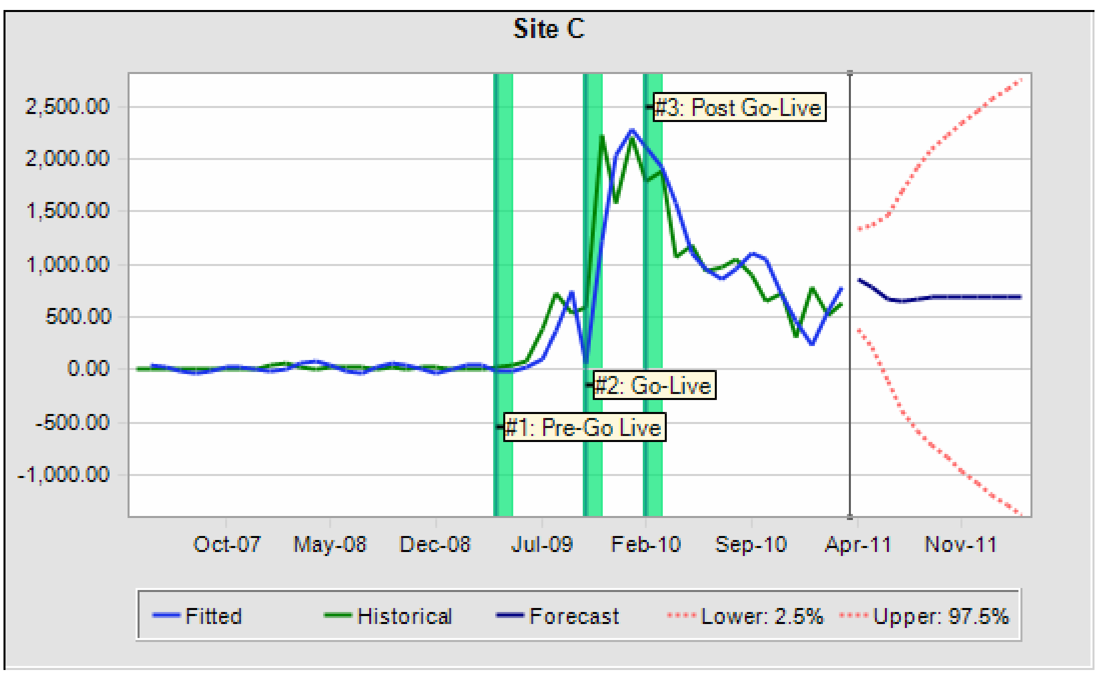

Figure 13: Early implementation of ARIMA Analysis For Help Desk Ticket Modeling