Towards Better Help Desk Planning: Predicting Incidents and Required Effort

Anneliese Amschler Andrews^a, Philip Beaver^b, Joseph Lucente^a

^a*Department of Computer Science
University of Denver
Denver, CO 80208*

^b*Daniels College of Business
University of Denver
Denver, CO 80208*

Abstract

In this case study, a cost model for help desk operations is developed. The cost model relates predicted incidents to labor costs. Since incident estimation for hundreds of products is time-consuming, we use cluster analysis to group similarly behaving products in clusters, for which we then estimate incidents based on the representative product in the cluster. Incidents are predicted using software reliability growth models. **The cost to resolve the incidents is predicted using historical labor data for the resolution of incidents.** Cluster analysis is used to group products with similar help desk incident characteristics. We use Principal Components Analysis to determine one product per cluster for the prediction of incidents for all members of the cluster, so as to reduce estimation cost. We were able to predict incidents for a cluster based on this product alone and do so successfully for all clusters with accuracy comparable to making predictions for each product in the portfolio. Linear regression is used with cost data for the resolution of incidents to relate incident predictions to help desk labor costs. The cost model is then validated by successfully demonstrating cost prediction accuracy for one month prediction intervals over a 22 month period.

1. Introduction

Help Desk operations at a large company can be expensive. In 2010, the average cost to resolve a help desk incident was \$48 US dollars at North American companies [18]. In this case study, approximately 123000 incidents were submitted over a four and a half year period to resolve issues with 156 products. Approximately 41000 labor hours were spent to resolve these incidents, therefore costs incurred for incident resolution were significant. **The average hourly rate used in this case study is \$50 US dollars. This results in approximately two million dollars spent to resolve incidents.** Software down time causes revenue loss [23], costs associated with IT operations on the other hand reduce profit. Minimizing software failures in an operational environment is important to customer experience, but more practically to the cost model for offering IT services. Two costs are associated with software failures. First, short-term losses are realized through missed commitments in daily business rhythms when software resources become nonfunctional or inaccessible to employees. Longer term effects are unmet contractual obligations which can lead to financial penalties or contract loss. Secondly, there are costs associated with the effort to resolve problems by help desk technicians and to manage resolution processes by help desk managers.

Workloads at the help desk are dynamic. There is a need for efficient resolution to minimize employee wait time for help desk personnel to begin to work on problems. Employing a sufficiently large team of help desk technicians to handle surges in problems addresses the demand but results in idle staff during periods of nominal help desk operations. Excess staff results in unnecessary costs when there is no backlog of problems to resolve. Conversely, employing too few technicians results in delays with problem resolution due to backlogs. This drives the cost of unproductive employees while problems are being resolved. It would thus be helpful to find a way to predict incidents and the cost to resolve them. Help desk managers need a way to plan staffing levels so that labor costs are minimized while problems are resolved efficiently. An incident prediction method would be useful to help desk managers for planning staffing levels. The potential value of a solution to this problem is important to an IT service provider since software failures are inevitable and their **timing is difficult to predict manually.** In addition to assisting with labor predictions for existing operations, a cost prediction model would be a valuable tool in the

Email addresses: andrews@cs.du.edu (Anneliese Amschler Andrews), Philip.Beaver@du.edu (Philip Beaver), Joe.Lucente@du.edu (Joseph Lucente)

preparation of bids for additional business through demonstration of management efficiency by presenting evidence of historically accurate labor predictions.

This paper presents a cost model that is derived from incident reports and labor data from a large help desk operation in an industrial setting. The approach uses real help desk incident data and actual labor costs collected over a four and a half year period. We present a cost model derived from a database of resolved incidents submitted against the full scope of problem types encountered by the help desk in this case study. Through the development and validation of our technique, we address the following research questions:

- RQ1 Can desktop software product reliability data obtained from help desk incidents be used to predict future incident volume?
- RQ2 Can incident prediction accuracy be obtained through the selection of product clusters vs. analyzing a full product portfolio in a prediction model? Doing so would reduce estimation effort.
- RQ3 Can incident resolution labor data be used to develop a help desk cost prediction model?

The remainder of this paper is organized as follows. In section 2 we present existing work in the context of our research questions. Section 3 explains our approach to predict both incident volume and labor costs. The cost model is validated in section 4. Threats to validity in our case study are presented in section 5. We draw conclusions and lessons learned through a discussion of our results in section 6 and discuss opportunities for future work.

2. Background and Related Work

2.1. Help desk operations

The primary goal of the help desk is to provide cost effective IT services to an organization. Amongst a wide variety of services the help desk resolves problems that occur when using products in the desktop software portfolio. Each help desk request is recorded in an incident record. Help desk labor is managed through resource planning, tracking planned labor costs to actual costs, and adjusting staffing levels to balance service delivery commitments with labor budget targets. The basis of estimate for staffing levels is driven primarily by service delivery metrics rather than through help desk workload predictions. Help desk managers are expected to find ways to improve problem resolution efficiency to reduce costs.

Standard methods for help desk operations facilitate efficiency. ISO 20000 is an international standard for information technology service management[1]. ISO 20000 includes a formalization of IT Service Management (ITSM) through the Information Technology Standard Library (ITIL). Over 850 companies around the world are ISO 20000 certified, as is the organization investigated in this case study. ITSM includes an operational framework for help desks. By complying with ITSM standards, operations described in this case study are similar to many other IT companies.

There are three categories of incidents managed by the help desk of the organization in this case study. The categories are shown in Figure 1. The first category consists of incidents submitted to report software product specific issues. For example, an employee contacts the help desk to report an error message produced by Adobe Reader. Help desk technician labor is spent resolving this category of incidents through a body of knowledge comprised of technician experience and scripted solutions. This case study focuses on incidents in the first category.

A second major category of incidents relates to defective hardware such as broken keyboards, crashed hard drives and problems that are not specific to any particular installed software product. This category also includes incidents associated with general operating system errors and computer performance problems. These types of issues are usually not resolved by addressing the functionality of any product other than the operating system itself or an underlying component such as the .Net framework.

A third major category of incidents is designed to address issues for well-defined situations, through dedicated resolution mechanisms. Incidents in this category are typically routed to dedicated help desk team members through interactive voice prompts that ensure a trained help desk technician is assigned to the incident. Dedicated effort can be directed to products with high call volume through standard scripts or known resolution methods to achieve resolution efficiency.

Table 1 lists seventeen incident attributes that are recorded for each incident with their unique identifier, description, numerical range of values and their role as an input or output during incident processing. All seventeen attributes are mandatory fields in the help desk database. Attributes A1, A2 and A3 are categorical attributes whose range of values consists of four discrete levels for incident urgency, impact and priority. Urgency is assigned

Figure 1: Help desk incident resolution labor categories

by the help desk technician based on standard guidance which describes the amount of time by which resolution is expected based on general problem types. Impact is assigned through discussion with the employee who reports the problem based on the number of employees or devices affected by the unresolved issue. Incident priority is automatically determined through a mapping of the combination of Urgency and Impact to one of fourteen levels as indicated for attribute A1. The lowest priority incidents are assigned a value of zero. Attributes A5 and A6 are binary categorical levels assigned by the incident management system. A4 is assigned a value of 1 if the incident is resolved on the call or chat session initiated to report the problem. A4 is a service metric used as an indicator of the level of service provided by the help desk technician. In general, the help desk service model is designed around a maximization of the level of service and a minimization of cost and effort. A small number of incidents are excluded from First Call Resolution service metrics. These are incidents for which specific business rules apply such as trouble alerts automatically generated through application health monitoring mechanisms. A set of six attributes describe the extent to which new incidents are similar to existing incidents, documented solutions, known problems and those related to specific artifacts. These are attributes A6 through A11. Values for these attributes are the number of existing help desk records with matching resolution methods. An increase in resolution efficiency achieved with effort and cost reduction is targeted through quick reference to other help desk records which assist in problem resolution, although no quantitative studies have been conducted to substantiate the benefits of incident matching. Metrics derived from attributes A6 through A11 may also be used to identify problems which are potentially resolvable through cost-saving self-service methods.

Attributes A12 and A13 measure impacts to the cost of resolving incidents. Some incidents are not resolvable by the technician providing assistance during the initial engagement with the help desk. These incidents must be transferred to subject matter experts. While first call resolution is a cost-based goal, resolution quality is often achieved through reassignment of responsibility to more knowledgeable individuals, at the expense of increasing resolution effort. The number of times an incident is transferred for reassignment is recorded in attribute A12. Managerial assistance is sometimes required to ensure incidents are resolved. This may become necessary for three reasons. 1) Incident volume may exceed the capacity of available technicians, 2) experience and knowledge of the technician may impact the speed and quality with which incidents are resolved, and 3) high priority incidents sometimes need managerial focus to restore productivity to the affected individuals or systems, especially with wide-scope service outages. Similarly, the priority of some incidents may demand the coordinated efforts of managers to bring in subject matter experts. Attribute A13 indicates whether or not an incident was escalated in that way for resolution. A14 and A15 are measures of cost and effort associated with incident resolution. A16 is a per-incident measure of service restoration within time spans defined by incident priority. Not all incidents must meet a time span. A17 identifies whether they do or not.

2.2. Cost Models

In this case study we investigate techniques to predict the cost of help desk operations specific to incident resolution. We apply a cost model to predicted incident volume to estimate the cost of resolving the predicted incidents. We note that costs associated with product failures are mainly from these sources:

- Help desk technician time spent to investigate, diagnose and resolve incidents
- Managerial time spent to coordinate help desk operations
- Employee time spent reporting problems to the help desk and waiting for issues to be resolved

Table 1: Incident Attributes

Attr ID	Description	Range	Role
A1	Urgency	1=low, 2=med, 3=high, 4=crit	input
A2	Impact		
A3	Priority		
A4	Resolved through First Call Resolution (FCR)	0 = False, 1 = True	output
A5	Resolvable through First Call Resolution (FCR)	0 = False, 1 = True	input
A6	Incident Matches a help desk historical record		
A7	Number of Matches to any other incident	(0, 1, 2, ... , n)	input
A8	Number of Matches to any known solution		
A9	Number of Matches to any known error		
A10	Number of Matches to any known problem		
A11	Number of Matches to any configuration item		
A12	Total Transfers of Incident	(0, 1, 2, ... , n)	output
A13	Incident Escalated	0 = False, 1 = True	output
A14	Incident Response Time	(0, 1, 2, ... , n)	output
A15	Incident Resolution Time		
A16	Return to Productivity (RTP) Compliance	0 = False, 1 = True	output
A17	Return to Productivity (RTP) Eligibility	0 = False, 1 = True	input

These are costs directly attributable to software product failures. There are also indirect costs resulting from product failures. Examples of indirect costs are compromised or missed deadlines due to product malfunctions, possibly wrong information rendered from products operating in a broken state, and potentially larger effects such as loss of competitive business due to software product unreliability. In this case study we focus on costs which relate directly to help desk operations. Costs are measured as known labor hours required to accomplish work. Labor rates (dollars per hour) can also be used to account for the resulting dollar cost associated with help desk operations. The application of known or assumed labor rates can be easily applied to our model to determine dollar costs. Labor hours spent on help desk efforts are recorded by employees and managers. Specific cost accounts are set up to collect labor hours spent on incident resolution. Accurate labor reporting and accounting are enforced through process audits to ensure validity.

Our case study focuses on help desk operations for the resolution of incidents from products in their operational phase. By contrast, Do et al. investigate software processes with an economic focus in the management of development and test activity [9]. Cost benefits in a test environment are investigated where test case prioritization techniques for regression testing are empirically evaluated. Similarly, Do and Rothermel investigate test phase cost benefit trade-off in the modeling of component costs associated with regression testing [7, 8]. The economics of regression testing specific to cost-benefit trade-offs are empirically evaluated by Malishevsky [13]. While our research shares similar goals related to software costs, our focus is on products in their operational phase. This is an important distinction in that our cost model applies to products whose operational lifespan is usually unknown at the time of release. Should the lifespan of a product go beyond an incident prediction period, our model can be reapplied using more recent incident data for further cost projections.

Help desk process improvements rather than incident volume and labor predictions are the focus of research conducted by Shu and Guang [21] where quality improvement is modeled using data mining techniques. Increases in processing throughput are demonstrated using real-time service metrics such as average speed of answer. Novel approximations methods for call arrival processes in service centers are proposed by Steckley [22] and confirmed through a computational evaluation using call center data. Similarly, Ibrahim [10] exposes the challenges of building realistic statistical tools for analyzing call arrival processes. In our case study we model costs, unlike the approach based on call center arrival used by [10]. Further research in call center operations is presented through the introduction of a novel call type classification technology by Tang [28]. By using actual transcribed conversations from an IT help desk at the University of Colorado in Boulder, CO, Tang applies four text-based call-type classification algorithms to determine the error compared to manually classifying the calls using the call-type taxonomy. An algorithm based on Support Vector Machine (SVM) outperforms other approaches. Tang demonstrates help desk improvements through real-time processes, where our case study 1) relies on historical data, and 2) delivers a managerial solution rather than solving an operational problem. Help desk automation through the development of problem resolution expertise is discussed by McCarthy [14] in the context of high staff turnover and resulting loss of experience at call centers. Case-Based Reasoning (CBR) is proposed as a departure from rule-based methodologies. The application of CBR to historical help desk data is shown to evolve knowledge in problem resolution. An approach to identifying solutions by matching symptoms to an archival database of problems and their solutions is proposed as a method for the automatic diagnosis of software problems reported by users. While McCarthy clearly

focuses on help desk process improvements, specifically with levels of experience, our research addresses efficiency in the managerial aspect of help desk management rather than in live help desk operational scenarios.

An alignment of IT services to a business model is the focus of a study by Bose, et al. [5]. The use of process limits is suggested to understand help desk process behavior such as average service response times. For example, process limits can expose the likelihood of missing a service target. Continuous process improvement based on a set of consistent metrics and an analytical engine for processing anomalous events and process shifts are the key components of a model-based framework for continuous process improvement in the delivery of IT services. The framework is implemented in an industrial setting. While our case study is also in an industrial setting, our approach differs from that of Bose, et al. in the phase in which our techniques are applied. We derive a cost model from help desk data and predict labor. In contrast, Bose, et al. effect process improvement in real-time events.

Table 2 summarizes the related work discussed above.

Table 2: Summary of related work

Citation	Goal	Application area	Notes
[1] ISO 20000	Standardize processes	IT	The help desk in this case study is compliant with ISO 20000 standards
[9] Do et. al.	Prioritize regression testing	Regression testing	Costs associated with software development and test phases are analyzed, as opposed to during the operational phase in our case study
[7, 8] Do and Rothermel	Investigate cost benefit trade-off of cost modeling		
[13] Malishevsky	Investigate cost modeling with test processes	Regression testing	(same as above)
[21] Shu and Guang	Process improvement	Help desk (average speed of answer)	Help desk quality improvement is measured through real-time metrics in contrast to incident and cost prediction in our case study
[22] Steckley	Model call center operations	Help desk (call arrival)	An operational model is investigated, in contrast to the cost model proposed in our case study
[10] Ibrahim	Model call center operations	Help desk (call arrival)	A statistical model is investigated, in contrast to the cost model proposed in our case study
[28] Tang	Classify call type based on transcribed conversation	Help desk (real-time processes)	Real-time processes are investigated as opposed to historical incident data in our case study
[14] McCarthy	Automated problem resolution via case-based reasoning	Help desk	
[5] Bose et. al.	Process improvement in real time	Help desk	

3. Approach

3.1. Analysis Process

By integrating actual cost data with incident history for a selected set of desktop products, we present an approach by which incident resolution cost is predicted. Figure 2 describes the main steps in our approach. The details of the process are included following the approach section. An overview is included in this section to introduce the reader to the overall approach. Following product selection according to the process described earlier, incident data is collected for the incident estimation process steps shown in Figure 2. Cluster analysis is used to group with similar attribute values shown in Table 1.

Principal Components Analysis (PCA) has been adopted as a popular method in software engineering to achieve reduction in large variable sets and to group variables into similarly-behaving subsets [27, 26, 15, 17]. The use of PCA in this research is novel, in that it has not been used previously with the selection of software products for use in incident predictions. PCA is used to determine a product in each cluster to represent all products in that cluster. The incident estimation techniques described in this case study are used to predict incidents for the representatives as well as for all products. In the cost estimation steps in Figure 2, actual cost data is obtained for the resolution of help desk incidents, to determine cost function parameters. The parameters are then used to estimate incident resolution costs. The process concludes with a validation of cost estimates by comparing predicted costs to actual costs.

Our approach uses two major phases:

- incident estimation
- cost estimation

Figure 2: Process Flow: product selection through process validation

Since incident estimation for hundreds of products is expensive, as discussed by Andrews and Lucente [4]¹, we use cluster analysis to group similarly behaving products in clusters, for which we then estimate incidents based on the representative product in the cluster. We determine this representative product via PCA. We use these representative products to estimate the total number of incidents for all selected products using software reliability growth models. In Phase 2 we use incident estimates and cost data to estimate expected labor cost for these expected incidents. Finally, we validate our approach on the data available through our case study. We use several types of help desk data including incident volume and labor data. The characteristics of the data are presented so that the reader understands the role of each data type in the case study.

3.2. Case Study Data

This case study describes a large scale IT operation in a large, multinational company. The IT operation services 800 products installed on over 120000 desktop machines.

Twenty percent of the products in use by the organization in this case study are widely distributed to the majority of its employees. These are mainly office productivity applications (email and word processing), Internet browsers, domain-specific business tools and security-related applications. We based our initial selection on the following product characteristics.

1. Each product is installed on a Windows operating system.
2. There are 50 or more installations of the product.
3. There are at least 50 incidents per product.

¹also see Table 13

Table 3: Installation and incident data for products selected in case study

Prod ID	num installs	num inc	Prod ID	num installs	num inc	Prod ID	num installs	num inc
A5	193	232	A190	756	1367	A382	201	269
A6	1806	122	A200	1874	81	A383	32208	346
A13	15286	184	A204	3238	153	A384	13502	1138
A14	2692	3105	A211	44758	1054	A385	104296	1045
A20	1058	55	A219	4704	111	A387	19278	151
A21	57133	215	A222	2625	153	A391	93	52
A25	12452	75	A228	7570	65	A393	36197	107
A27	37615	115	A229	4483	94	A394	102	79
A28	40267	483	A232	94514	198	A396	9085	1729
A39	6492	1063	A235	238	140	A397	3312	475
A42	38652	3470	A236	17169	705	A403	17557	174
A44	32214	345	A239	578	59	A404	41940	954
A56	6893	305	A244	25440	207	A408	2204	69
A64	249	89	A245	107	67	A409	1691	207
A65	17271	872	A246	1483	52	A413	49067	65
A74	20700	87	A248	442	218	A416	490	322
A75	890	113	A249	121259	199	A421	15398	326
A77	18895	216	A255	5547	351	A427	862	87
A81	4587	213	A260	21434	861	A436	3072	66
A82	176	76	A267	799	71	A439	124882	54
A83	699	201	A271	131874	100	A442	418	197
A86	1037	116	A275	3201	1363	A450	5234	134
A89	1449	107	A277	95	1510	A451	1763	67
A90	5257	3701	A278	70	59	A456	3166	138
A101	255	78	A282	905	84	A464	3356	50
A103	3556	1064	A284	10108	340	A465	399	1587
A104	3231	698	A287	17876	935	A467	64073	1386
A114	17885	1877	A290	393	204	A468	1602	51
A115	20638	3188	A293	79696	777	A474	13567	736
A116	402	3179	A294	3531	1088	A476	3208	91
A117	782	1042	A295	889	178	A487	38512	416
A119	1764	542	A300	4134	66	A488	6712	195
A123	8355	823	A310	5721	61	A495	4732	87
A128	87716	1034	A312	118566	268	A497	126289	2755
A144	480	290	A315	614	81	A499	1194	1477
A145	10611	90	A319	32115	588	A500	63555	392
A146	5094	1559	A320	11861	1224	A501	5609	235
A148	18972	831	A327	345	226	A502	113203	123
A151	3704	52	A329	3133	53	A503	1861	600
A152	1392	162	A333	3246	2837	A508	613	298
A155	522	203	A334	29993	2484	A510	10786	94
A162	82	125	A345	2115	1227	A511	452	481
A165	6394	70	A347	1428	131	A514	4863	60
A166	11016	656	A357	500	85	A516	213	70
A171	10646	161	A359	2074	1778	A517	513	1144
A172	428	175	A361	240	174	A519	1357	269
A177	32022	64	A367	119384	2662	A520	98952	493
A178	21815	3916	A368	763	237	A523	119341	3358
A179	874	395	A372	1754	86	A524	757	413
A182	105885	191	A373	7767	98	A529	760	103
A185	359	120	A375	1304	126	A531	15570	95
A187	12911	1244	A376	4630	990	A532	95	156

4. Product incident records were generated between 4/1/2008 and 10/31/2013.

Searching the organization’s software distribution system resulted in 361 products which were installed on at least 50 computers. Incident counts for each of the initial set of 361 products were obtained from the help desk database. Incidents marked as unresolved or canceled were not included in our analysis. Of the roughly 96,500 incidents considered, fewer than 100 were rejected for their resolution or cancellation status. We removed 205 products for which the 50 incident minimum criterion was not met. The resulting 156 products were used for analysis in the remainder of the case study. Table 3 summarizes these products in three sets of columns. Each of these sets reports a product identifier, the number of installations and number of incidents. Figure 3a shows these 156 products in a scatter plot based on number of installations and number of incidents. Most products are clustered in the range of

(a) Product installations and incident quantities

(b) Number of product installations

(c) Product incident quantities

Figure 3: Product installations and Incident Quantities: 156 Products

less than 20000 installations and less than 500 incidents. Figure 3b shows a box plot of the number of installations of the products. As can be expected from 3a, the box plot shows a skewed distribution with a number of outliers. Specifically, half of the products are installed on fewer than 3544 computers and the remaining are installed on more than 3544 computers. The interquartile range shows us that 50% of the products are installed on between 815 and 17,990 computers. The average number of installations is 17,470. There are approximately 20 outlier products with installations on between 4000 and 120,000 computers. Figure 3c shows a box plot of the number of incidents generated by the products. Half of the products produce fewer than 204 incidents as indicated by the median value. The interquartile range shows us 50% of the products produced between 94 and 812 incidents, with an average of 592 incidents produced. Twelve products shown as outliers produced between 2500 and 4000 incidents. Descriptive statistics are included in Table 4 for a comparison of the selected products and the full product portfolio in terms of installations and incidents. The descriptive statistics in Table 4 indicate one half of the 800 products are installed on fewer than 320 machines. Similarly, one half of the products produce 17 or fewer incidents. This skewness

Table 4: Comparison of selected product descriptive statistics to full help desk product portfolio

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
Incident volume (all products)	1	3	17	321	408	18840
Incident volume (selected products)	50	94	204	592	812	3916
Number of installations (all products)	1	27	320	7450	2589	131900
Number of installations (selected products)	70	815	3544	17470	17990	131874

Figure 4: Monthly incident volume distribution by category

in both product installations and number of incidents generated is evident in Figure 3a. Of the roughly 123000 incidents submitted over the four and a half year period in this case study, 93000 (76%) incidents were produced by the 156 products selected for investigation.

We also obtained actual monthly labor figures for effort in resolving help desk incidents. The labor figures have been validated by help desk managers to ensure their accurate interpretation. Effort is measured as the number of hours recorded by help desk technicians responsible for the resolution of incidents submitted for desktop software products, hardware issues, operating system errors and products outside the scope of this research. We are concerned with the amount of labor associated with the resolution of incidents for the products we use in our cluster analysis. Because monthly labor figures account for resolution effort for all types of incidents, we need a way to extract the portion of labor that represents effort only for the incidents submitted for the 156 clustered products. An attempt to obtain reasonable estimates of the applicable portion through interviewing help desk managers was not successful.

Figure 4 shows incident volume from April 2008 to October 2013. The average monthly incident volume for the software products and hardware/OS categories is roughly 2000 and 1000 respectively, whereas the average volume of incidents for the dedicated effort category is near 10,000. The help desk incident database separates incidents by product name and by time interval, but it does not automatically separate incidents into the three categories. For example, the database can be queried for all incidents reported against Adobe Reader in the May 2013 time frame. Similarly, incidents against the Windows 7 operating system and those against peripheral devices (keyboards, mouse, etc.) are easily obtained, but their separation from named software products is not made.

The data used for this case study fall into two categories defined by Runeson, et. al. [20]. Knowledge of help desk operations in the industrial setting of this case study is obtained by direct methods through semi-structured interviews with two help desk managers. Data obtained by this method compliments the experienced-based knowledge of help desk data analysis by one of the authors of this paper. A second category of data, one in which the majority of the quantitative data in this case study falls, is obtained through the inspection of archival data in the help desk database of incident records. The quantitative data is collected and maintained in the help desk database for the analysis of performance with respect to established service metrics and for trend analyses from which business decisions are made.

Table 5: Monthly labor hours for products selected in case study

Month	Hours	Month	Hours	Month	Hours	Month	Hours	Month	Hours	Month	Hours
-	-	1/2009	822	1/2010	768	1/2011	1008	1/2012	719	1/2013	810
-	-	2/2009	803	2/2010	797	2/2011	796	2/2012	924	2/2013	965
-	-	3/2009	1083	3/2010	820	3/2011	849	3/2012	782	3/2013	1015
4/2008	925	4/2009	901	4/2010	780	4/2011	786	4/2012	742	4/2013	854
5/2008	736	5/2009	821	5/2010	777	5/2011	723	5/2012	686	5/2013	854
6/2008	771	6/2009	903	6/2010	808	6/2011	832	6/2012	615	6/2013	755
7/2008	865	7/2009	854	7/2010	957	7/2011	766	7/2012	724	7/2013	788
8/2008	719	8/2009	933	8/2010	884	8/2011	874	8/2012	690	8/2013	660
9/2008	829	9/2009	782	9/2010	839	9/2011	726	9/2012	562	9/2013	629
10/2008	876	10/2009	868	10/2010	880	10/2011	681	10/2012	646	10/2013	713
11/2008	728	11/2009	859	11/2010	785	11/2011	724	11/2012	595	-	-
12/2008	616	12/2009	669	12/2010	618	12/2011	563	12/2012	455	-	-

For each product in our analysis we obtain the following data.

1. Number of incidents submitted within the time interval for our analysis
2. Date and time at which each incident is submitted
3. Values for seventeen incident attributes as shown in Table 1 for all incidents

We use incident data in two ways. First, we develop an incident volume prediction model using actual incidents resolved by the help desk as described later in our approach. Secondly, we validate the incident prediction process by measuring the accuracy of incident volume prediction compared to the actual number of incidents generated by a set of products over a selected time period.

In this case study we also use help desk cost data in two ways. First, historical cost data is used with incident volume to develop a cost prediction model based on predicted incident volume. Our cost model development will be discussed later in our approach. Secondly, we use actual cost data to validate the accuracy of our cost prediction model. The cost data we collected is in units of hours per calendar month. The data are the collective number of hours for all help desk technicians and managers who spent effort on resolving all incidents for the given month. The actual number of incidents is known as well as the product for which each incident is submitted. Although the numbers represent efforts toward incident resolution, we are careful to note that they include effort for all product-based help desk incidents, inclusive of the 156 products in our case study. The ideal scenario of accounting for labor by product would have greatly assisted our research, but the labor was not accounted for to that level of granularity. The 156 products in this case study is a subset of the overall portfolio of products managed by the help desk. Similarly, the aggregate labor hours each month represents effort spent resolving incidents from products in the overall portfolio. The two entities are related through the identification of a specific product in each incident. This allows us to determine the number of incidents each month that are submitted against the 156 products in our case study and for the balance of the portfolio. Actual labor costs for the effort described above were obtained from help desk managers. We collected monthly labor hours for each month over the time interval of interest in this case study. The labor data is shown in Table 5.

3.3. Cluster Analysis

A goal of this step is to avoid having to estimate incidents for 156 products selected in section 3.2. We use cluster analysis to group products based on incident behavior (attributes in Table 1). This is essentially a data reduction technique using pattern recognition [12][11].

Cluster analysis is a method use to separate a data set into the most similar data in the same cluster and most dissimilar data in different clusters. We apply an agglomerative hierarchical cluster analysis method [30] to group products together.

A vector of seventeen attributes of Table 1 for each of the 156 products is input to the cluster analysis. We used the *R* statistical tool *hclust* package. It includes a utility to plot the hierarchical relationship between clusters as a dendrogram. Clusters with relatively small distances between heights in the dendrogram are more similar to one another than those whose height differences are greater. Since we clustered products on the basis of their centroids, products in clusters with smaller differences in height behave more similarly in terms of their incident attributes. The dendrogram produced from the cluster analysis is shown in Figure 5. Rather than deciding on clusters by specifying a height in Figure 5, we were constrained by the limitations in our linear regression tool to

Figure 5: Dendrogram from cluster analysis showing all products

no more than 16 clusters. We also wanted to have a manageable number of products per cluster. An analysis of the dendrogram with these constraints in mind resulted in 11 clusters. While this may be less than ideal from a theoretical perspective, it nevertheless resulted in high quality incident estimation, as shown later. All 156 product are accounted for in the eleven clusters. Products in the same application domain did not necessarily cluster together. Although they are in the same domain, they can have very different incident attributes, perhaps due to one vendor’s product being more incident-prone than a similar product from another vendor. Table 6 shows the eleven clusters. Column 1 shows the cluster ID, column 2 shows the products in each cluster.²

3.4. Representative Product per Cluster

One of the motivations for clustering was to try to determine a representative product in each cluster, so we need only predict future incidents for one product in each cluster rather than for all. The representative product must behave similarly to the other products in its cluster in terms of incident behavior. The behavior of an incident is observed through its attributes shown in Table 1. We use PCA to determine product similarity. PCA shows us the amount of variance between attributes from a set of incidents for each product in a cluster. We can see how the attributes relate to one another within a product through inspections of the resulting principal components. For example, if attributes $A1$ (incident urgency) and $A2$ (incident impact) show strong covariance in the principal components from three products, we conclude those three products have similar behavior on the basis of the impact and urgency. By discovering the attribute relationships in each of the products within a cluster, we can rank the products according to their similarity based on the number of attribute relationships that are common between the cluster members. Products with higher numbers of attribute relationships that are common with other cluster members get ranked higher. The product with the highest rank is designated the cluster representative.

To achieve this, we assigned a weighted similarity score to each product in a cluster. We measure similarity on the basis of the number of attribute relationships a candidate product shares with the other products in the same cluster. The score is weighted more heavily by attribute relationships which are shared by larger numbers of products in a cluster and weighted less by attribute relationships that are shared with fewer products. The similarity score of the candidate product is decreased by the number of attribute relationships which are unique to the product. These are attribute relationships that are not shared with any products in the cluster.

For example, consider cluster c containing four products p_1, p_2, p_3, p_4 . The goal is to determine the product that best represents the other products in cluster c . We perform a PCA for each product and show the PCA loadings in Tables 7a through 7d. In these tables, the row IDs correspond to the attributes shown earlier in Table 1. The column IDs are the names of the principal components. The numbers in each column are the loadings for the principal component. The PCA loadings shown in boldface form the principal component relationships for each product. These are identified using techniques in [6, 31, 16, 29, 3]. For example, in Table 7a the PCA loadings for PC1 result in a relationship between $A3$ and $A5$. This relationship is interpreted as “attribute $A3$ (incident priority) varies with $A5$ (incident is resolvable through First Call Resolution) for product p_1 ”.

²In Table 6 the numbers in parenthesis may be ignored at this time as they will be discussed later.

Table 6: Cluster product scores based on number of shared and unique principal components within cluster

Cluster ID	Cluster members (raw similarity score)	Selected product
BV	A421 (19), A284 (26), A287 (23), A155 (13), A116 (8), A383 (28), A387 (23), A5 (12), A81 (23), A347 (16), A44 (29), A235 (20)	A44
EW	A501 (29), A531 (25), A394 (22), A427 (25), A145 (47), A372 (66), A146 (24), A334 (61), A6 (41), A13 (32), A497 (53), A171 (64), A249 (71), A21 (63), A393 (23), A75 (48), A359 (28)	A249
EG	A179 (27), A467 (34), A90 (11), A499 (8), A172 (16), A39 (18), A333 (47), A487 (40), A14 (32), A255 (28), A232 (13), A236 (46), A511 (52), A517 (45)	A511
ER	A248 (84), A375 (30), A101 (109), A295 (53), A327 (53), A532 (53), A514 (27), A391 (64), A468 (123), A151 (45), A439 (107), A246 (85), A300 (90), A329 (108), A413 (115), A200 (26), A239 (55), A510 (119), A450 (89), A502 (111), A245 (77), A267 (107), A282 (118), A436 (46), A20 (75), A310 (96)	A468
EI	A185 (98), A228 (106), A82 (36), A464 (40), A25 (62), A165 (53), A77 (94), A368 (68), A277 (34), A190 (44), A524 (39), A27 (46), A229 (39), A28 (40), A278 (47)	A228
EH	A244 (36), A260 (25), A117 (36), A367 (31), A523 (51), A320 (16), A404 (48), A42 (32), A211 (41), A451 (18), A56 (28), A373 (17)	A523
EK	A64 (27), A315 (47), A65 (24), A119 (26), A456 (24), A89 (26), A219 (54), A476 (42), A271 (47), A408 (35), A495 (48), A357 (31), A516 (27), A123 (38), A416 (33), A178 (13), A104 (24), A376 (37)	A219
EF	A115 (56), A204 (76), A385 (72), A500 (77), A396 (69), A474 (71), A275 (49), A294 (86), A128 (79), A520 (71), A319 (62), A397 (46), A74 (34), A312 (43), A403 (45)	A294
DA	A529 (39), A114 (21), A290 (35), A488 (47), A345 (44), A503 (31), A152 (50), A162 (25), A222 (14), A508 (37), A86 (30), A83 (22), A361 (55)	A361
BU	A382 (5), A442 (6), A519 (9), A409 (1), A384 (7), A148 (7), A103 (-2), A187 (3)	A519
DV	A166 (11), A293 (7), A177 (2), A182 (-1), A144 (3), A465 (-7)	A166

Table 8 shows all of the principal component relationships found in products p_1 , p_2 , p_3 and p_4 in cluster c . A bullet in Table 8 indicates the principal component relationship in the left column is found in the product identified in the column header. Some attributes vary independently, such as A9 in Table 8 for PC1. In other words, attribute A9 does not vary with any other attribute. For each principal component relationship, a relationship score is calculated and shown in the right-most column in Table 8, according to the following assignments:

- A relationship found in exactly four products is assigned a score of 3
- A relationship found in exactly three products is assigned a score of 2
- A relationship found in exactly two products is assigned a score of 1
- A relationship found in exactly one product is assigned a score of -1

Table 8 shows the relationship score summation for each of the four products and the resulting similarity score. The product with the highest similarity score is designated as the cluster representative. In this example, product p_1 has a similarity score of 7, which is higher than the scores for the other three products as shown in the bottom row of Table 8. Product p_1 is therefore determined to be the representative of cluster c .

In the similarity scoring technique for n products in c , the attribute relationships for the product to be scored are compared against the $n - 1$ other products. The similarity score is influenced positively by higher numbers of shared relationships and incorporates a penalty for the non-shared (unique) relationships. Table 6 shows the eleven clusters and the products which comprise each cluster. The similarity score assigned to each product using the scoring technique described above is shown in parenthesis next to each product ID. The product selected to represent the cluster members is shown in the column labeled “Selected Product”. As a means of comparing the cluster representatives, Table 9 shows the number of installations and incidents for each cluster representative, and the time period over which the incidents were generated.

3.5. Incident Estimates for Representative Product per Cluster

We adopt the approach from Stringfellow et al., Andrews et al. and Andersson [24, 25, 2]. Figure 6 shows

	PC1	PC2	PC3	PC4	PC5
A1	0.2124	0.3130	-0.3699	0.1651	-0.2458
A2	-0.0200	-0.2648	-0.4168	0.1978	0.1871
A3	0.8038	0.0296	0.2538	-0.1603	-0.2805
A4	0.0920	0.5006	-0.2795	0.1706	-0.1452
A5	0.7889	0.1478	-0.2066	-0.1390	0.0682
A6	0.0680	-0.6004	-0.2265	0.3063	0.2982
A7	0.1459	0.2198	-0.0025	0.5123	-0.1574
A8	-0.0170	0.3011	0.1069	-0.8662	0.0874
A9	-0.2071	0.2267	0.0837	0.0779	0.4223
A10	-0.2216	-0.2184	0.2822	0.1746	0.3298

(a) Product p_1

	PC1	PC2	PC3	PC4	PC5
A1	-0.3955	0.3166	0.1674	0.0457	0.3023
A2	-0.4587	0.1164	-0.0872	0.1867	-0.1206
A3	-0.0014	0.4892	0.0337	0.1202	0.1584
A4	-0.0347	-0.1873	0.7821	-0.0747	0.2256
A5	0.1937	0.6891	0.0499	0.0998	-0.0116
A6	0.2892	-0.1487	-0.4520	-0.0989	-0.3184
A7	-0.0612	0.0632	0.2316	-0.3972	-0.2350
A8	-0.2587	-0.2237	0.0771	-0.2857	-0.6123
A9	-0.3155	-0.2147	0.1263	0.2232	0.5563
A10	-0.0530	0.0961	0.1915	-0.4235	-0.1711

(b) Product p_2

	PC1	PC2	PC3	PC4	PC5
A1	0.2513	-0.3109	0.0627	-0.0179	0.2369
A2	-0.0269	0.2793	0.1674	0.1557	-0.0635
A3	-0.2033	-0.2574	0.7756	0.2081	-0.2958
A4	-0.1123	0.4561	0.0039	0.1345	-0.2210
A5	0.2524	0.0489	0.3287	-0.0200	0.2015
A6	0.2381	-0.6154	0.0947	0.0364	-0.2633
A7	-0.2174	0.1160	-0.1342	0.6122	0.0968
A8	0.0526	-0.0044	0.2559	-0.7708	0.1678
A9	-0.8223	-0.2031	0.0585	-0.1161	-0.0078
A10	-0.3055	-0.0926	0.2675	0.2706	-0.4800

(c) Product p_3

	PC1	PC2	PC3	PC4	PC5
A1	-0.0378	0.1164	-0.2083	0.2569	0.0725
A2	0.0959	0.2390	0.5514	0.1444	-0.2855
A3	0.5621	-0.3020	-0.0036	0.1263	-0.1564
A4	0.2028	0.3298	0.2717	-0.1862	0.0111
A5	0.6632	0.2143	0.0933	0.0893	0.1494
A6	0.2890	-0.4877	0.0527	-0.1172	-0.0276
A7	0.1640	0.2642	0.1142	0.1550	0.3550
A8	0.0361	0.0793	0.1827	0.2627	-0.0749
A9	-0.3185	0.2864	0.0351	-0.2780	-0.2404
A10	-0.0971	-0.1790	0.0054	-0.3870	0.1424

(d) Product p_4

Table 7: PCA loadings for scoring example

Figure 6: Incident prediction process flowchart

Table 8: Product similarity scoring example for cluster c

Principal component relationship	Products in cluster c				Relationship score
	p_1	p_2	p_3	p_4	
A1, A2	•	•			1
A3, A5	•	•	•	•	3
A4, A6	•	•	•	•	3
A7, A8	•		•		1
A9, A10	•				-1
A7, A10		•			-1
A8, A9		•			-1
A9			•		-1
A10			•	•	1
A2				•	-1
A7				•	-1
Relationship score sum	1 + 3 + 3 + 1 -1	1 + 3 + 3 - 1 -1	3 + 3 + 1 -1 -1	3 + 3 -1 -1 -1	
Similarity score	7	5	5	3	

Table 9: Cluster representatives - product characteristics

Cluster ID	Cluster rep ID	Num installations	Num incidents	Dates of incidents
BV	A44	32214	345	4/11/2008 - 10/17/2013
EW	A249	121,259	199	6/24/2008 - 10/23/2013
EG	A511	452	481	4/1/2008 - 10/22/2013
ER	A468	1602	51	8/15/2008 - 1/7/2013
EI	A228	7570	65	4/11/2008 - 8/31/2012
EH	A523	119,341	3358	5/22/2008 - 9/13/2013
EK	A219	4704	111	4/9/2008 - 10/30/2013
EF	A294	3531	1088	4/2/2008 - 10/24/2013
DA	A361	240	174	4/1/2008 - 10/25/2013
BU	A519	1357	269	4/10/2008 - 10/10/2013
DV	A166	11,016	656	4/1/2008 - 5/23/2013

our estimation process. The process begins with obtaining cumulative incident data for the eleven representative products. The incident data is applied to all four SRGMs by running a curve fit to estimate model parameters for each SRGM. Models which do not converge are rejected. In the next step we determine the R^2 GOF value for each model that converges. The model with the best R^2 value is selected for prediction of incidents. The model is used to predict incidents for the next period (for example, one month), and to predict the remaining number of incidents. In these last two steps the model may be used to predict incidents for more than one period (one month and two month predictions, for example). After the predictions are made, the first process is repeated, starting with the collection of incidents for the next period.

Tables 10 and 11 show the results of applying this approach to the representative products in all eleven clusters for an R^2 value of 0.95 and 0.99. Incident data from April 2008 through August 2012 was used to obtain the results. To the right of the cluster ID in each table is the ID of the representative product. The SRGM column lists the model that resulted in the best R^2 value. The number of incidents predicted for one month through five months out is shown along with the remaining number of incidents for the one month through five month prediction periods. The remaining number of incidents is obtained by subtracting the predicted number of incidents submitted from the total number of incidents predicted by the SRGM, as shown in Figure 6. For $R^2 \geq 0.95$ (Table 10), all curve fits resulted in at least one model that converged with R^2 greater than 0.95. However for $R^2 \geq 0.99$ (Table 11), cluster EI resulted in R^2 less than 0.99 for all models. The best R^2 value for EI was 0.9541 for the GO Musa model. We note in Table 9 that product A228, the representative for cluster EI, produced the second smallest number of incidents over the time period in this case study. In fact, this product has only 65. An inspection of the cumulative incident data shows the number of incidents for this product reaches 100% of its total incidents after 48 months. The remaining 4 months have no more incidents. In contrast, the representative for cluster ER (product A468) which has even fewer total incidents than A228 (51), achieved an R^2 value of 0.9951. An inspection of the incident data for A468 indicates the number of incidents for this product reaches 100% of its total during the last month of analysis. None of the other representative products have stopped producing incidents at the end of the data collection period. These results suggest products at or beyond their incident-producing periods may not be suitable

candidates for representing other products in a cluster. It is interesting to see that this method was able to come

Table 10: Incident estimations for cluster representatives, $R^2 \geq 0.95$

Cluster ID	Cluster rep	SRGM	Incidents for next period					Remaining incidents				
			1 mo	2 mo	3 mo	4 mo	5 mo	1 mo	2 mo	3 mo	4 mo	5 mo
BU	A519	Mod Gompertz	246	249	252	255	259	242	239	236	233	229
BV	A44	Mod Gompertz	315	322	328	335	342	704	697	691	684	677
DA	A361	Mod Gompertz	167	169	172	174	177	117	115	112	110	107
DV	A166	Mod Gompertz	626	629	631	633	636	43	40	38	36	33
EF	A294	Mod Gompertz	1064	1069	1073	1077	1081	121	116	112	108	104
EG	A511	Mod Gompertz	452	455	458	461	464	2792	2789	2786	2783	2780
EH	A523	Gompertz	3170	3197	3222	3246	3269	546	519	494	470	447
EI	A228	GO Musa	59	59	59	59	59	0	0	0	0	0
EK	A219	Gompertz	99	100	100	101	102	10	9	9	8	7
ER	A468	Gompertz	51	51	52	52	52	2	2	1	1	1
EW	A249	Mod Gompertz	130	132	134	137	139	292	290	288	285	283

Table 11: Incident estimations for cluster representatives, $R^2 \geq 0.99$

Cluster ID	Cluster rep	SRGM	Incidents for next period					Remaining incidents				
			1 mo	2 mo	3 mo	4 mo	5 mo	1 mo	2 mo	3 mo	4 mo	5 mo
BU	A519	Mod Gompertz	246	249	252	255	259	242	239	236	233	229
BV	A44	Mod Gompertz	315	322	328	335	342	704	697	691	684	677
DA	A361	Mod Gompertz	167	169	172	174	177	117	115	112	110	107
DV	A166	Mod Gompertz	626	629	631	633	636	43	40	38	36	33
EF	A294	Mod Gompertz	1064	1069	1073	1077	1081	121	116	112	108	104
EG	A511	Mod Gompertz	452	455	458	461	464	2792	2789	2786	2783	2780
EH	A523	Gompertz	3170	3197	3222	3246	3269	546	519	494	470	447
EI	A228	none	-	-	-	-	-	-	-	-	-	-
EK	A219	Gompertz	99	100	100	101	102	10	9	9	8	7
ER	A468	Gompertz	51	51	52	52	52	2	2	1	1	1
EW	A249	Mod Gompertz	130	132	134	137	139	292	290	288	285	283

Table 12: Actual incidents and relative error for predictions by cluster representatives, $R^2 \geq 0.95$

Cluster ID	Actual incidents for next period					Relative error				
	1 mo	2 mo	3 mo	4 mo	5 mo	1 mo	2 mo	3 mo	4 mo	5 mo
BU	239	243	251	258	260	0.0293	0.0247	0.0040	-0.0116	-0.0038
BV	321	329	331	334	338	-0.0187	-0.0213	-0.0091	0.0030	0.0118
DA	165	168	170	171	173	0.0121	0.0060	0.0118	0.0175	0.0231
DV	627	633	639	641	641	-0.0016	-0.0063	-0.0125	-0.0125	-0.0078
EF	1057	1061	1064	1069	1074	0.0066	0.0075	0.0085	0.0075	0.0065
EG	462	463	470	470	470	-0.0216	-0.0173	-0.0255	-0.0191	-0.0128
EH	3227	3251	3265	3274	3276	-0.0177	-0.0166	-0.0132	-0.0086	-0.0021
EI	65	65	65	65	65	-0.0923	-0.0923	-0.0923	-0.0923	-0.0923
EK	101	101	103	105	107	-0.0198	-0.0099	-0.0291	-0.0381	-0.0467
ER	52	52	52	52	52	-0.0192	-0.0192	0.0000	0.0000	0.0000
EW	108	119	125	141	146	0.2037	0.1092	0.0720	-0.0284	-0.0479

up with an individual estimate in all but the EI cluster representative at $R^2 \geq 0.99$ (at least one of the models converged).

Having estimated future incidents for the representative product in each cluster, we turn to estimating incidents for all products in a cluster. Note that this method allows for flexible prediction time frames. We have shown 1-5 month predictions as an example.

3.6. Incident Estimation Methods Using Cluster Representatives

We present two approaches for estimating the number of incidents that will be generated by each cluster.

- Scalar approach: The estimate of incidents produced by all cluster members in a given cluster is determined by multiplying the number of incidents predicted by the cluster representative by the number of products in the cluster. This is represented in Equation 1.

$$Incidents_S = Incidents_{rep} * |cluster| \quad (1)$$

- Linear function approach: A function derived through linear regression based on historical incident data is used to estimate incidents produced by all cluster members for a given cluster. The linear function relates actual incidents numbers from the cluster representative to the number of incidents produced by all cluster members. With the linear function approach, the estimated number of incidents for a cluster is a linear transformation of the number of incidents predicted by the cluster representative, as shown in Equation 2.

$$Incidents_L = a * Incidents_{rep} + b \quad (2)$$

For both methods, the estimated number of incidents for all 156 products is obtained by summing the per-cluster estimates for all eleven clusters.

3.6.1. Scalar Method

Let N_{cr_i} be the number of incidents predicted for the representative product of cluster c_i ($i = 1 \dots n$, where n is the number of clusters).

Let m_i be the number of products in cluster i .

Then N_{c_i} , the number of incidents predicted for cluster c_i is given by

$$N_{c_i} = m_i * N_{cr_i} \quad (i = 1, \dots, n) \quad (3)$$

The total number N_S of incidents is then given by

$$N_S = \sum_{i=1}^n N_{c_i} = \sum_{i=1}^n m_i * N_{cr_i} \quad (4)$$

Table 13: Scaled incident prediction relative error for one month prediction using Scalar Method (September 2012)

Cluster ID	Actual num inc cluster rep	Predicted num inc cluster rep N_{cr_i}	Relative error per cluster rep	Actual num inc per cluster	Estimated num inc per cluster N_{c_i}	Relative error per cluster
BU	239	246	0.0293	4773	1968	-0.5877
BV	321	315	-0.0187	6089	3780	-0.3792
DA	165	167	0.0121	4340	2171	-0.4998
DV	627	626	-0.0016	3409	3756	0.1018
EF	1057	1064	0.0066	11525	15960	0.3848
EG	462	452	-0.0216	12692	6328	-0.5014
EH	3227	3170	-0.0177	12028	38040	2.1626
EI	65	59	-0.0923	1210	885	-0.2686
EK	101	99	-0.0198	7911	1782	-0.7747
ER	52	51	-0.0192	2032	1326	-0.3474
EW	108	130	0.2037	6118	2210	-0.6388
Total	6424	6379	-0.0070 (overall rel. error)	72127	78206 N_S	0.0843 (overall rel. error)

Table 13 shows the results of applying the scalar incident prediction method to estimate the cumulative number of incidents that will be generated by all 156 products in one month. Incident data from April 2008 through August 2012 was used in the predictions, so the one month predictions in Table 13 are for September 2012. Starting in the left two columns of Table 13, each cluster ID is shown with the actual number of incidents produced by its cluster representative. In the third column, the predicted number of incidents for each cluster representative is shown. These are the values of N_{cr_i} in Equation 3. The relative prediction error per cluster representative is included in the table to show the accuracy with which each cluster representative product predicts its own incidents. The overall relative error based on the cluster representatives is shown at the bottom of the column titled “Relative error per cluster rep”. This number is calculated from the total of the predicted number of incidents for each cluster

representative, and the total of the actual number of incidents for each cluster representative, using the formula $(\text{predicted} - \text{actual})/\text{actual}$. The column titled “Actual num incidents per cluster” contains the known number of incidents produced by all members of each cluster through the month of September 2012. As with the numbers in the column “Actual num incidents cluster rep”, the actual number of incidents produced by all cluster members is used to calculate relative prediction error by cluster. The numbers in the column titled “Estimated num incidents per cluster” are the scaled values for N_{c_i} obtained through Equation 3. Their sum is used along with the sum of the actual number of incidents per cluster to determine the relative incident prediction error for all 156 products combined.

Using the scalar incident prediction method, the relative error per cluster representative shown in the fourth column from the left in Table 13 varies from a minimum (worst under-prediction) of -0.0923 (cluster representative for EI) to a maximum (worst over-prediction) of 0.2037 (cluster representative for EW). The two most accurate cluster representative incident predictions (relative error closest to zero) are -0.0016 and 0.0066. These are from the clusters DV and EF respectively. The mean and median incident prediction relative error for all cluster representatives are 0.0055 and -0.0177 respectively.

The overall one month incident prediction error of the cluster representatives is -0.0070. Although this is a reasonably small overall prediction error, we wanted to investigate why some incident predictions from the representatives were more accurate than others. The predicted and actual cumulative incidents as a function of time are

Figure 7: Cluster representative curve fits: two best (DV, EF) and two worst (EI, EW) based on one month prediction accuracy for September 2012 predictions

shown in Figure 7 for the two most accurate one month predictions, Figures 7a and 7b, and the two least accurate one month predictions, Figures 7c and 7d. We can see how data points for the actual incidents fit the curve for the predicted incidents somewhat better for DV and EF in Figures 7a and 7b, compared to EI and EW in Figures 7c and 7d. The difference is more evident in months 45 through 53 for all four cluster representatives compared to earlier months. Since one month predictions beyond month 53 are measured against actual incidents (see Table 13, column “Relative error per cluster rep”), the error at month 53 detected visually helps explain why predictions one month out have the same approximate magnitude of error as one month earlier. For clusters DV and EF, whose representatives resulted in the two most accurate predictions, the curve fit error shown graphically in Figures 7a and 7b at month 53 is very small. Another distinction between the four results in Figure 7 is that number of incidents at month 53 for the two least accurate predictors is only about 10% of the number of incidents for the two most

accurate predictors. The appearance of the predicted curve for EI in Figure 7c for the later months is close to horizontal. This denotes the product selected to represent EI is approaching the end of its incident producing period. Each SRGM in this case study has a parameter which holds the value for the cumulative number of incidents that would eventually be reached if sufficient time were permitted. By appearance of Figure 7c for cluster representative EI, that number is slightly under 60 incidents. An inspection of the detailed results for the curve fit for EI confirms the number is 59. This validates our suspicion of the life cycle phase of the desktop product selected to represent cluster EI. In contrast, the appearance of Figure 7d for cluster representative EW indicates incidents are still being produced at month 53 since there is no evidence of the curve flattening out. Given the resulting prediction error, this may suggest the cluster representative for EW is not sufficiently mature to accurately represent the other cluster members, although for the other seven cluster representatives whose curve fits graphs are not shown for brevity, three indicate a similar level of maturity as seen with EF. These findings present an opportunity for an extension of this research with a focus on product life cycle, product incident volume, and candidacy for cluster representation.

The two right-most columns in Table 13 show the results of scaling the number of incidents estimated for each cluster representative, using Equation 3, and the per-cluster relative error. Table 13 also includes the actual number of incidents per cluster. This number is the sum of the actual incidents produced by each member of the cluster identified in the “Cluster ID” column. The per-cluster actual number of incidents is used with the per-cluster estimated number of incidents to calculate the relative error per cluster shown in Table 13. Additionally, totals for the actual and estimated number of incidents and overall relative error are shown at the bottom of the table. These values are associated with all 156 products.

In addition to incident predictions based on cluster representatives, Table 7 shows the per-cluster incident prediction and relative prediction error. Similar to our analysis of relative error for each cluster representative, we are interested in per-cluster relative error and why some cluster predictions are more accurate than others. In the “Relative error per cluster” column in Table 13 we see a range of -0.7747 for cluster EK (worst underprediction) to 2.1626 for cluster EH (worst overprediction). The mean and median relative error are -0.1226 and -0.3792 respectively. The overall relative error is 0.0843. Our first observation is that overall **relative error of 0.0843** (bottom of right-most column of Table 13) is worse compared to the overall relative error of the cluster representatives (-0.0070). Secondly, the magnitude of the per-cluster relative error is much greater than that of each cluster representative for all eleven clusters. An inspection of the clusters which resulted in the two most accurate predictions (DV and EI) and two least accurate predictions (EH and EK) indicates weak correlation with the two best and two worst predictions for the cluster representatives. Cluster DV resulted in the most accurate incident prediction for both the cluster representative and for all cluster members. Interestingly, EI is the next most accurate cluster predictor while it was one of the worst as a cluster representative. This tells us that incident prediction accuracy by the cluster representatives (using the scalar incident prediction method) is not a strong indicator of cluster-wide accuracy based on the one month prediction results. This is not surprising since the scalar method uses only one variable, m_i , to transform the prediction of the cluster representative to a prediction of incidents for all members of the cluster it represents. More than one variable in the transformation, as will be discussed with the linear function method, is likely to achieve better prediction accuracy through an improved function fit. As will be seen, however, the trade-off for improved prediction accuracy is the additional effort required to perform the linear regression associated with the linear function method.

3.6.2. Linear Function Method

In the linear function approach to incident prediction, we develop a set of linear functions to relate the number of incidents predicted by the cluster representative to the number of incidents for the aggregate of all members of the cluster.

Let $A_{cr_i}(t)$ be the actual incident counts for time frame t for the cluster representative cr_i for cluster c_i ($i = 1, \dots, n$), where n is the number of clusters.

Let $A_i(t)$ be the actual incident counts for time frame t for cluster c_i ($i = 1, \dots, n$). Assume that $1 \leq t \leq T$, where T is the time period for which measurements are available. We perform linear regressions of the form

$$A_i = x_i A_{cr_i} + y_i \quad (i = 1, \dots, n) \quad (5)$$

to determine constants x_i and y_i for each cluster C_i .

Then N_{c_i} , the number of incidents predicted for cluster c_i is given by

$$N_{c_i} = x_i N_{cr_i} + y_i \quad (6)$$

We then can use these linear functions to predict total number of incidents N_L based on the predicted incidents for each cluster representative:

$$N_L = \sum_{i=1}^n N_{c_i} \quad (7)$$

These functions allow us to predict the number of incidents for all members of a cluster based on the cluster representative alone. Table 14 shows for each cluster actual cumulative incidents vs. estimated incidents and

Table 14: Incident prediction relative error for one month prediction using Linear Function Method (September 2012)

Cluster ID	Actual num incidents per cluster	Estimated num incidents per cluster $x_i N_{c_i} + y_i$	Relative error per cluster
BU	4773	5373	0.1257
BV	6089	8603	0.4129
DA	4340	3236	-0.2544
DV	3409	3849	0.1291
EF	11525	10726	-0.0693
EG	12692	13844	0.0908
EH	12028	12181	0.0127
EI	1210	417	-0.6554
EK	7911	6496	-0.1789
ER	2032	1617	-0.2042
EW	6118	5970	-0.0242
Total	72127	72312 (N_L)	0.0026 (overall rel. error)

relative error. As with the scalar method described earlier, incident data from April 2008 through August 2012 was used in the predictions, so the one month predictions in Table 14 are for September 2012. The overall relative error is shown at the bottom of the column titled “Relative error per cluster”. This number is calculated from the total of the estimated number of incidents for each cluster, and the total of the actual number of incidents for each cluster representative, using the formula (predicted - actual)/actual.

Similar to our analysis of relative error for each cluster representative using the scalar incident prediction method, we are interested in per-cluster relative error with the linear function method as well as overall relative prediction error. Specifically, we investigate why some cluster predictions are more accurate than others. The relative error per cluster shown in Table 14 varies from a minimum (worst under-prediction) of -0.6554 (cluster representative for EI) to a maximum (worst over-prediction) of 0.4129 (cluster representative for BV). The two most accurate incident predictions (relative error closest to zero) are 0.0127 and 0.0242. These are from the cluster representatives for EH and EW respectively. The mean and median relative error calculated from the set of eleven estimations are -0.0559 and -0.0242 respectively. The overall relative error is 0.0026.

Our first observation with the results of the linear function method is that the overall incident prediction is rather accurate for this particular case of one month incident predictions. The absolute error of 185 incidents is quite small compared to the total incident volume of approximately 72,000 incidents. The other significant finding with the linear function method is that in spite of the relatively inaccurate per-cluster predictions (for example EI and BV in Table 14), the overall relative error is small.

3.6.3. Estimate of Incidents for All Products

This requires using the method in step C for all 156 products. It is by far the most expensive. This method estimates 77,491 incidents compared to 72,127 actual incidents with a **relative error of 0.0585**. So not only is it very expensive, it is also rather inaccurate. A comparison of the estimation methods in terms of effort is presented in the next section.

3.6.4. Comparison of approaches to estimate future incidents

We compare all three approaches to incident prediction for a one month interval. Table 15 shows the comparison in the order in which the estimation methods is presented. The results are obtained using incident data from April

Table 15: Incident prediction relative error for one month prediction: comparison of three methods (September 2012)

Cluster ID	Scalar			Linear function		All products		
	Actual incidents per cluster	Estimated incidents per cluster	Relative error per cluster	Estimated incidents per cluster	Relative error per cluster	Actual number incidents	Estimated incidents all products	Relative error
BU	4773	1968	-0.5877	5373	0.1257	72127	77491	0.0585
BV	6089	3780	-0.3792	8603	0.4129			
DA	4340	2171	-0.4998	3236	-0.2544			
DV	3409	3756	0.1018	3849	0.1291			
EF	11525	15960	0.3848	10726	-0.0693			
EG	12692	6328	-0.5014	13844	0.0908			
EH	12028	38040	2.1626	12181	0.0127			
EI	1210	885	-0.2686	417	-0.6554			
EK	7911	1782	-0.7747	6496	-0.1789			
ER	2032	1326	-0.3474	1617	-0.2042			
EW	6118	2210	-0.6388	5970	-0.0242			
Total	72127	78206	0.0843 (overall RE)	72312	0.0026 (overall RE)	72127	77491	0.0585 (overall RE)

2008 through August 2012, to predict incidents for September 2012. Starting with the left-most column in Table 15 we show the cluster IDs. The actual number of incidents produced by each cluster representative is shown to the right of the column of cluster IDs. Column two shows the actual incidents per cluster. Column three shows the number of incidents estimated for each cluster using the Scalar method. In column four we show the incident prediction relative error for the cluster representatives. Columns five and six show the estimated incidents for each cluster and the relative prediction error respectively, for the Linear function method. Columns seven, eight and nine show the results of the All products method.

- For the Scalar method, the predicted number of incidents per cluster representative, scaled by the number of products in each cluster, is shown along with the relative prediction error per cluster. The relative prediction error of 0.0843 is determined by the sum of the estimated number of incidents per cluster (78206) and the actual number of incidents per cluster (72127).
- For the Linear function method, the predicted number of incidents per cluster representative, transformed by the linear function described earlier, is shown along with the relative prediction error per cluster. The relative prediction error of 0.0026 is based on estimated total incidents (72312) versus the actual number of incidents per cluster (72127).
- For the All products method, the estimated number of incidents (77491) and the sum of the actual number of incidents (72127) are used to calculate a relative error of 0.0585.
- Overall in Table 15, we observe the Linear function estimation method results in the most accurate prediction of incidents. The All products method is the second best, and the Scalar method is the least accurate of the three estimation methods. We will investigate the quality of prediction beyond one month in the validation section.³

We conclude our analysis with a comparison of the incident estimation methods, and discuss the trade-off of effort that is necessary with the more accurate incident prediction method. In terms of overall prediction accuracy, the Linear function method is more accurate than both the Scalar method and the All products method when predicting incidents one month out for the month of September 2012. Additionally, per-cluster incident prediction accuracy with the Linear function method is generally better than with the Scalar method. A more extensive investigation of the All products incident estimation method is presented in our previous work [4], where help desk incidents are estimated for 156 products using 22 cumulative incident data sets. For each data set, incidents are predicted at five prediction intervals (one month, two months, through five months). The median of relative prediction error found at the one month predictions was 0.0582 using the All products method. In this case study the All products method resulted in a relative prediction error of 0.0585 which is very close to the median error from our previous work. As discussed earlier, the Scalar method uses only one variable to estimate the number of

³This is not surprising because the linear method uses a function that has two variables, as opposed to the scalar method that uses one variable

incidents for all members of a cluster based only on incidents predicted by the cluster representative, so we expect less accurate predictions than what is produced by the Linear function method which uses two variables.

3.7. Cost Data Analysis

The cost model validated in this case study is developed using monthly incident data from product clusters and help desk technician labor costs. Using linear regression, a function is produced which relates per-cluster incident predictions to the cost to resolve the incidents. The role of the cost prediction model is shown in Figure 8. Per-cluster incident predictions from either the scalar model or the linear function model are inputs to the cost prediction model. Cost is predicted using linear regression.

The actual monthly labor costs obtained from the help desk managers in this case study is a set of monthly labor figures to resolve all incidents from all products used in the organization. As explained earlier, we applied a set of criteria to select the 156 products. Our cost model is based on estimated incidents from the 156 products, and predicts costs to resolve incidents from only the 156 products. In order to validate the cost model we need a way to identify the amount of monthly labor attributable to them. We investigate a proportion method in which we assume the amount of labor to resolve incidents from the 156 products is based on the ratio of the monthly number of incidents produced by the 156 products, to the total number of incidents produced by all products in the month.

Let I_c be the number of incidents generated by the set of clustered products at time t .

Let I be the number of incidents generated by all products at time t .

Then k_c , the proportion of incidents from the clustered products at t is given as

$$k_c = I_c/I \quad (8)$$

Let R be the cost to resolve I at time t .

Then R_c , the cost to resolve I_c at t is given as

$$R_c = k_c R \quad (9)$$

For the cost model, we develop a set of linear functions to relate the number of incidents predicted by all members of a cluster to the cost associated with resolving incidents produced by the members of the cluster. The incident predictions can be determined through either the Scalar or Linear Function methods, as shown in Figure 8.

Let $A_{c_i}(t)$ be the actual incident counts for timeframe t for the cluster c_i for cluster c_i ($i = 1, \dots, n$), where n is the number of clusters.

Then $\bar{A}_c$, the vector of actual incident counts for timeframe t is given as

$$\bar{A}_c = \begin{bmatrix} A_{c_1} \\ A_{c_2} \\ \vdots \\ A_{c_n} \end{bmatrix} \quad (10)$$

Let $R_c(t)$ be the actual cost of resolving incidents in $\bar{A}_c$ for time frame t . We perform linear regressions of the form

$$R_c = k_c R = \bar{A}_c \bar{P} + q \quad (11)$$

to determine coefficients $p_i \in \bar{P}$ ($i = 1, \dots, n$), where n is the number of clusters, and q is the intercept.

Let R_{c_i} be the cost to resolve the estimated number of incidents in cluster c_i .

$$R_{c_i} = p_i N_{c_i} \quad (12)$$

Figure 8: Help desk cost prediction

Then R , the cost to resolve incidents in all clusters is given as

$$R = q + \sum_{i=1}^n R_{c_i} \quad (13)$$

We can use Equation 13 to predict the total cost of resolving incidents based on the predicted incidents for each cluster. Table 16 shows one month cost estimations for September 2012. Results from incident predictions using the Scalar and Linear function methods are shown side by side for comparison. Table 16 shows the cluster IDs on the left. In column 2 we show the coefficients and intercept obtained from the linear regression described above. The coefficients are the components of $\bar{P}$ in Equation 11. The intercept is q . Column 3 shows the actual total cost of resolving incidents produced by the 156 incidents. The total cost is a cumulative number (total cost through August 2012 to resolve the cumulative incidents produced by the 156 products through August 2012). Columns 4, 5 and 6 show the cost predictions when incidents are estimated using the Scalar method. Columns 7, 8 and 9 show the cost predictions when incidents are estimated using the Linear function method. For both methods, the estimated incidents per cluster are the same as in Table 15 since we are predicting incidents and cost for the same month (September 2012). For both incident prediction methods in Table 16 we show the estimated cost (labor hours) for each cluster. These numbers are calculated using Equation 12. The total estimated cost is calculated using Equation 13. We now proceed to a discussion of the results of our cost predictions.

Table 16 shows for each cluster actual labor cost, estimated labor cost, and relative error for both the scalar and linear method of incident estimation. The sum of the per-cluster cost components and the intercept produce the overall cost estimate to resolve the predicted number of incidents. The cost prediction relative error of 0.0165 using the Linear incident estimation method is much better than the cost prediction error of -0.2703 obtained using the Scalar incident prediction method. These results are not surprising since cost predictions are based on the estimates from the incident estimation method (Scalar versus Linear function). Since the Linear function method produced more accurate incident estimations, we expect more accurate cost predictions compared to the scalar method.

4. Validation

The method presented in section 3 uses one month predictions to estimate the number of incidents at the help desk and the labor effort required to deal with them. The results were shown to be generally quite good, but only cover a single month. We now turn to evaluate their performance when used over a longer period of time, i.e.

- RQ How stable are one month incident predictions when used over a long period of time?

Table 16: Cost prediction relative error for one month prediction (September 2012)

Cluster ID	Actual cost (labor hours)	Scalar method		Linear function method	
		Estimated cost (labor hours)	Relative error	Estimated cost (labor hours)	Relative error
BU	40822	237	-0.2703	646	0.0165
BV		3769		8577	
DA		2136		3138	
DV		5420		5554	
EF		4627		3109	
EG		3721		8140	
EH		5331		1707	
EI		1031		486	
EK		1091		3977	
ER		79		96	
EW		2158		5828	
intercept		192		192	
Total		29792		41496	

4.1. Incident Estimation

To answer this research question, we produced one month incident predictions over the 22 month time period of January 2012 through October 2013. To address prediction stability over subsequent months, we apply the three estimation methods to one month incident predictions for all 22 months and observe the range of relative error for each estimation method. Additionally, we compare the difference in prediction stability when the predictions are limited to R^2 GOF thresholds of 0.95 and 0.99. For each approach, we look for median relative error closest to zero as an indicator of prediction accuracy. We also look for the approach with the smallest interquartile range (IQR) of relative error as an indicator of prediction stability. Finally, we look for the approach with the smallest range of outliers as an additional indicator of prediction stability.

Figure 9 shows relative incident estimation error for the three estimation approaches, at GOF thresholds of 0.95 and 0.99 for the SRGM that results in the best GOF at each of the 22 one month periods. A numerical summary of the box plots in Figure 9 is shown in Table 17. Figure 9a shows the range of relative incident prediction error from the three estimation methods when the predictions are limited to a GOF greater than or equal to 0.95. The All products approach results in a median relative prediction error of -0.0025 with first and third quartiles at -0.0292 and 0.0217 respectively. Although this method shows stability with median relative error centered close to zero, it results in outliers as low as -0.3636 and as high as 0.6829. This wide range of error suggests instability in prediction accuracy over the 22 months of predictions. The Scalar approach results in a median prediction error of -0.5114, with first and third quartiles at -0.6172 and 0.0846 respectively, and outliers at -0.8184 and 2.0770. The Linear function approach shows a median prediction error at -0.0648, first and third quartiles at -0.1645 and 0.0421 respectively, and outliers at -0.7642 and 0.2071. Of the three incident prediction approaches at $R^2 \geq 0.95$ in Figure 9a and summarized in Table 17, the All products approach has the best median prediction accuracy (median relative error closest to zero) followed by the linear function approach (method 3) as second-best and the scalar approach with the least prediction accuracy. The All products approach also results in the smallest interquartile range, again followed by Linear function then scalar. In terms of the overall range of outliers at $R^2 \geq 0.95$, Linear function results in the smallest range followed by All products then Scalar.

Turning our attention to $R^2 \geq 0.99$ in Figure 9b, we investigate incident prediction stability for the more restrictive case in which predictions with GOF no less than 0.99 are included. As with $R^2 \geq 0.95$, the more accurate prediction comes from the all products approach, followed by Linear function method, then the Scalar method. An investigation of interquartile range at $R^2 \geq 0.99$ shows us the same order as with $R^2 \geq 0.95$. The All products method results in the smallest interquartile range, followed by the Linear function method and then the Scalar method. In terms of the overall range of outliers at $R^2 \geq 0.99$, the Linear function method results in the smallest range followed by the All products method and then the Scalar method. Although this same rank order of median accuracy, IQR and outlier range was observed with $R^2 \geq 0.95$, the range of outliers is smaller with all three approaches for $R^2 \geq 0.99$ as compared with $R^2 \geq 0.95$. From this we observe the more restrictive GOF threshold also excludes a greater number of the less accurate predictions. For $R^2 \geq 0.99$, an interpretation of prediction error tells us the Linear function approach results in the most stable predictions over the 22 month period of one month

predictions since the spread of prediction error is the smallest of all three approaches. Additionally, at $R^2 \geq 0.99$, incident prediction stability over the 22 month period is greater than at $R^2 \geq 0.95$ through the exclusion of a higher number of less accurate predictions.

Figure 9: One month incident estimation relative error: comparison between three estimation methods

Table 17: Summary of one month incident estimation relative error

Estimation method	$R^2 \geq 0.95$					$R^2 \geq 0.99$				
	Min.	1st Qu.	Median	3rd Qu.	Max	Min	1st Qu.	Median	3rd Qu.	Max
1 All products	-0.3636	-0.0292	-0.0025	0.0217	0.6829	-0.1479	-0.0228	-0.0018	0.0223	0.6829
2 Scalar	-0.8184	-0.6172	-0.5114	0.0846	2.0770	-0.7889	-0.6128	-0.5117	0.0935	2.0770
3 Linear function	-0.7642	-0.1645	-0.0648	0.0421	0.2071	-0.1986	-0.1506	-0.0511	0.0459	0.2070

Table 18 ranks the three estimation methods in terms of median prediction accuracy, IQR, and outlier range, using the data in Table 17. The rankings are the same for both GOF thresholds. In Table 18, we observe the all products estimation method predicts incidents more accurately than the Linear function approach. The Scalar method is the least accurate. We see the same rank order of estimation methods for IQR, where smaller IQR indicates less spread and therefore more stability. The outlier range is smallest for the Linear function method and largest for the Scalar method. The Scalar method results in the least prediction stability, as was observed with prediction accuracy and IQR. In the introduction we listed a research question (RQ2) about achieving prediction

Table 18: Incident estimation method comparison by one month median prediction accuracy, IQR and outlier range

Median prediction accuracy			IQR			Outlier range		
best		worst	smallest		largest	smallest		largest
All products	Linear function	Scalar	All products	Linear function	Scalar	Linear function	All products	Scalar

accuracy through the selection of product clusters rather than analyzing all 156 products. Essentially we are asking if we can eliminate the All products incident estimation method and still predict incidents accurately. To help answer this question, we add up the number of operations required to produce an estimate for each of the estimation methods, and compare the results. If a method requires less work to produce an accurate estimate, that method would be preferred by help desk practitioners. In Table 19 we present the number of operations required for each method to achieve the goal of incident estimation for all 156 products. We analyze the methods in terms of the number of predictions and number of processing steps that are unique to the method. The All products incident estimation method requires the most number of operations because it requires running predictions for all products. The Scalar method requires the least number of operations and results in the greatest number of operations saved compared to the All products method. The Linear function method requires about twice the number of operations as the Scalar method but still saves 71% compared to the All products method. In terms of processing effort alone, we can reject the All products method and concentrate on the other two methods.

In Figure 10, incident estimation error from one month predictions is compared for the scalar and linear function incident estimation methods. We observe the scalar estimation method results in a median error less than zero.

Table 19: Comparison of estimation efficiency

	Estimation method 1	Estimation method 2	Estimation method 3
	All products	Scalar	Linear function
Operations	156 predictions	11 predictions	11 predictions
	1 summation	11 scalar multiplications	11 incident data lookups
		1 summation	11 linear regressions
			11 linear functions
			1 summation
total # operations	157	23	45
Relative Estimation Efficiency *	100%	85%	71%

* compared to All products method

Figure 10: Help desk one month incident estimation relative error by incident estimation method

Overall, the linear function method shown in Figure 10 indicates the improved prediction accuracy compared to the scalar method, as previously observed. As with the scalar method, there is little difference in terms of prediction accuracy across the models and between the two GOF thresholds.

4.2. Effort estimation

In the final part of our validation we analyze the stability of one month cost predictions over a 22 month period. In the approach section we made cost predictions for a single month. Because help desk managers routinely reestimate costs on a monthly basis, we need to ensure our technique produces stable results when applied over time. As with our analysis of incident prediction stability we look for cost prediction median relative error closest to zero as an indicator of prediction accuracy. We also look for the approach with the smallest interquartile range of relative error as an indicator of cost prediction stability. Finally, we investigate outlier range as an additional indicator of prediction stability. The set of incident estimates from one month predictions over a span of 22

Table 20: Summary of one month labor cost estimation relative error by incident prediction method

Pred method	$R^2 \geq 0.95$						$R^2 \geq 0.99$					
	Min	1st Q	Med	Avg	3rd Q	Max	Min	1st Q	Med	Avg	3rd Q	Max
Scalar	-0.8160	-0.6105	-0.4912	-0.1831	0.0735	1.9990	-0.7889	-0.6075	-0.5105	-0.1595	0.0758	1.9990
Lin fun	-0.7773	-0.1668	-0.0668	-0.0972	0.0421	0.2071	-0.1930	-0.1250	-0.0110	0.0224	0.1745	0.2861

months results in a range of relative error, as shown in the box plots in Figure 9. Since incident estimations are the basis of cost predictions, we can expect to find cost prediction error. Similar to the validation of one month incident prediction accuracy and stability, we investigate cost prediction accuracy and stability over a longer period of time.

Figure 11: Help desk one month labor cost estimation relative error by incident estimation method

The box plots in Figure 11 show the range of cost prediction error when the Scalar and Linear function incident estimation methods are used to predict incidents. The results of each method are shown when SRGM curve fit results are limited to the two R^2 thresholds of 0.95 and 0.99. Table 20 shows the descriptive statistics for cost prediction results.

As with incident predictions over 22 one month periods, the Scalar method results in a wider range of outliers compared to the Linear function method. For the Scalar method, the range is 2.8150 for $R^2 \geq 0.95$ and 2.7879 for $R^2 \geq 0.99$. The Linear function method results in a range of 0.9844 for $R^2 \geq 0.95$ and 0.4971 for $R^2 \geq 0.99$. The differences in the outlier ranges as well as the interquartile ranges across the two incident prediction methods can be seen in Figure 11. These results indicate cost predictions over the 22 month period are more stable with the Linear function method. The results also indicate little difference in cost prediction stability between the two R^2 thresholds with the Scalar method, but greater stability with the Linear function method at $R^2 \geq 0.99$ versus 0.95. In terms of the median cost prediction relative error, there is not much difference between the two R^2 values using the Scalar method (-0.4912 with $R^2 \geq 0.95$ and -0.5105 with $R^2 \geq 0.99$). There is a more noticeable difference in median cost prediction error with the linear function method, compared to the Scalar method. The linear function method results in a median cost prediction error of -0.0668 for $R^2 \geq 0.95$ and -0.0110 for $R^2 \geq 0.99$. These relative error values indicate greater cost prediction accuracy compared to using the Scalar method.

We turn to an analysis of the cost estimation results from the perspective of help desk managers. Tables 21 and 22 show cost predictions for each month using the Scalar and Linear function incident estimation approaches. For brevity we limit the results to cost predictions obtained at the $R^2 \geq 0.99$ threshold. Starting with the left column in Tables 21 and 22 we show the calendar month for which one month cost predictions are made. We include the estimated and actual number of incidents, as well as incident prediction relative error, in columns 2, 3 and 4 for each month. The numbers for estimated and actual incidents are cumulative. We include values for incident estimations since they are the basis of cost estimations. The values for effort estimation and actual monthly help desk effort are shown in columns 5 and 6. The relative error of the effort predictions is shown in column 7. The estimated costs in column 8 are calculated from an assumed hourly rate of \$100. The values in column 5 are multiplied by this rate to arrive at the estimated cost numbers in column 8. Similarly, the values in column 6 are multiplied by the same hourly rate to arrive at the actual cost numbers in column 9. Column 10 shows the percent variance between the estimated cost in column 8 and the actual cost in column 9. Cost variance is used routinely by the help desk managers in this case study to assess how far off they are each month in their projection of costs.

Table 21 shows cost estimations and percent variance when the Scalar method is used for one month cost predictions. The range of cost variance is -81.42% to 199.90%. The mean and median cost variance is 5.47% and -51.90%, respectively. From the perspective of the help desk manager, these results mean costs are under-predicted by as much as 81% and over-predicted by nearly 200%. On average, the manager can rely on cost predictions to vary by a little over 5%. While the average monthly cost prediction variance may seem reasonable, predictions that vary at the extreme ends of the range are not useful for predicting help desk costs. We observe the months for which

Table 21: One month cost estimations: Scalar method $R^2 \geq 0.99$

Month	Est. inc	Actual inc	Inc pred RE	Est effort (hours)	Actual effort (hours)	Effort pred RE	Est cost (\$100/hr)	Actual cost	% variance
01/2012	39447	64135	-0.6128	89364	34072	1.6149	\$8,936,380	\$3,417,537	162.28%
02/2012	39478	65397	-0.5187	92257	35293	1.6141	\$9,225,711	\$3,529,251	162.41%
03/2012	39507	66661	1.8966	17174	36411	-0.5283	\$1,717,446	\$3,641,073	-52.83%
04/2012	39538	67804	-0.7482	97416	37255	1.6149	\$9,741,567	\$3,725,465	161.49%
05/2012	39568	68888	-0.635	17983	38197	-0.5292	\$1,798,293	\$3,819,743	-52.92%
06/2012	39599	70042	-0.7834	102100	39146	1.6149	\$10,210,020	\$3,904,615	160.82%
07/2012	39629	71192	0.0935	19599	40039	-0.5105	\$1,959,895	\$4,003,871	-51.05%
08/2012	39660	72127	0.0843	19291	40822	-0.5274	\$1,929,077	\$4,082,170	-52.74%
09/2012	39691	73207	-0.5286	19600	41662	-0.5295	\$1,960,005	\$4,166,216	-52.95%
10/2012	39721	74149	-0.785	16682	42503	-0.6075	\$1,668,235	\$4,250,280	-60.75%
11/2012	39752	74880	1.9985	128612	42885	1.9990	\$12,861,238	\$4,288,509	199.90%
12/2012	39782	76249	-0.0938	41983	43774	-0.0409	\$4,198,251	\$4,377,366	-4.09%
01/2013	39813	77995	-0.7889	21208	44879	-0.5274	\$2,120,788	\$4,487,854	-52.74%
02/2013	39844	79650	-0.5925	16037	45677	-0.6489	\$1,603,679	\$4,567,662	-64.89%
03/2013	39872	81304	2.007	44539	46459	-0.0413	\$4,453,937	\$4,645,893	-4.13%
04/2013	39903	82956	-0.7152	16615	47134	-0.6475	\$1,661,542	\$4,713,353	-64.75%
05/2013	39933	84210	-0.0375	16821	47828	-0.6483	\$1,682,132	\$4,782,804	-64.83%
06/2013	39964	85579	-0.2149	46638	48618	-0.0407	\$4,663,780	\$4,861,769	-4.07%
07/2013	39994	86685	-0.5117	53191	49443	0.0758	\$5,319,114	\$4,944,334	7.58%
08/2013	40025	87668	-0.5454	17612	50101	-0.6485	\$1,761,178	\$5,010,077	-64.85%
09/2013	40056	88781	0.8582	9454	50883	-0.8142	\$945,412	\$5,088,333	-81.42%
10/2013	40086	89922	0.4752	49376	51492	-0.0409	\$4,937,640	\$5,148,011	-5.11%

some of the higher cost variances result are the end of the year and the beginning of next year. The organization in this case study has a planned shutdown in the last week of December. Additionally, many employees take vacation late in the year. This results in a reduced number of incidents submitted to the help desk. These conditions may explain the resulting overestimation of costs. We recommend taking unusual months into account when estimates are made and to adjust effort estimates accordingly.

Table 22 shows cost estimations and percent variance when the Linear function method is used for one month cost predictions. The range of cost variance is -19.30% to 28.61%. The mean and median cost variance is 3.82% and 0.63%, respectively. Compared to the Scalar method, the Linear function method results in more stable one month cost predictions. The mean and median predictions are also an improvement over the Scalar method. As with the Scalar method, we see the high and low ends of the cost prediction range tend to occur around the months at the end of the year and the beginning of the next year.

Consultation with the help desk managers in this case study indicates an average cost prediction variance of less than 5% would be a improvement over what actually occurs in their environment. While historical data for monthly cost estimation was not available for comparison with our results, we confirmed help desk monthly cost prediction variance is usually greater than 10%. Based on this information, cost predictions using either the Scalar or Linear function methods would be an improvement over current cost prediction variance, but the extreme under-estimations and over-estimations would likely be cause for rejection of the Scalar method in favor of the Linear function method.

5. Threats to validity

We analyze the validity of our case study according to the approach taken by Runeson et. al. in their presentation of case study research [19].

Construct validity addresses the extent to which operational measures of a case study are accurately represented in the research questions. To ensure our research questions captured the operational aspect of cost estimation, we engaged the help desk managers through discussions focused on daily operations specific to the incident resolution.

Internal validity concerns an awareness of all factors which influence causal relationships. In this cases study, incidents drive help desk costs. In our discussion of the three categories of help desk incidents, we were clear about our knowledge of the separation of labor between the categories. However, our assumption about relationship between labor associated with the set of software products we investigated and labor for incidents from full product portfolio is not actually used in the industrial setting of this case study since labor is not accounted for by product.

Table 22: One month cost estimations: Linear function method $R^2 \geq 0.99$

Month	Est. inc	Actual inc	Inc pred RE	Est effort (hours)	Actual effort (hours)	Effort pred RE	Est cost (\$100/hr)	Actual cost	% variance
01/2012	69343	64135	0.0812	40926	34072	0.1975	\$4,092,638	\$3,417,537	20.12%
02/2012	52833	65397	-0.1921	44608	35293	0.2640	\$4,460,832	\$3,529,251	26.40%
03/2012	56203	66661	-0.1569	42622	36411	0.1706	\$4,262,198	\$3,641,073	17.06%
04/2012	59862	67804	-0.1171	30065	37255	-0.1930	\$3,006,450	\$3,725,465	-19.30%
05/2012	76648	68888	0.1126	32903	38197	-0.1386	\$3,290,327	\$3,819,743	-13.86%
06/2012	82338	70042	0.1755	39921	39146	0.0224	\$3,992,079	\$3,904,615	1.98%
07/2012	65149	71192	-0.0849	38209	40039	-0.0457	\$3,820,941	\$4,003,871	-4.57%
08/2012	72315	72127	0.0026	41495	40822	0.0165	\$4,149,526	\$4,082,170	1.65%
09/2012	62182	73207	-0.1506	34822	41662	-0.1642	\$3,482,200	\$4,166,216	-16.42%
10/2012	59423	74149	-0.1986	49596	42503	0.1669	\$4,959,589	\$4,250,280	16.69%
11/2012	69766	74880	-0.0683	50407	42885	0.1754	\$5,040,713	\$4,288,509	17.54%
12/2012	79749	76249	0.0459	53390	43774	0.2197	\$5,338,986	\$4,377,366	21.97%
01/2013	89234	77995	0.1441	52156	44879	0.1622	\$5,215,572	\$4,487,854	16.22%
02/2013	78226	79650	-0.0179	45070	45677	-0.0133	\$4,507,028	\$4,567,662	-1.33%
03/2013	98134	81304	0.2070	38560	46459	-0.1700	\$3,856,013	\$4,645,893	-17.00%
04/2013	78666	82956	-0.0517	46948	47134	-0.0039	\$4,694,830	\$4,713,353	-0.39%
05/2013	71960	84210	-0.1455	59394	47828	0.2418	\$5,939,381	\$4,782,804	24.18%
06/2013	81206	85579	-0.0511	46330	48618	-0.0471	\$4,632,966	\$4,861,769	-4.71%
07/2013	89605	86685	0.0337	43263	49443	-0.1250	\$4,326,292	\$4,944,334	-12.50%
08/2013	77905	87668	-0.1114	64435	50101	0.2861	\$6,443,460	\$5,010,077	28.61%
09/2013	74001	88781	-0.1665	43113	50883	-0.1527	\$4,311,302	\$5,088,333	-15.27%
10/2013	92972	89922	0.0339	49914	51492	-0.0304	\$4,991,382	\$5,148,011	-3.07%

External validity reflects the generalizability of our findings, and the extent to which our conclusions are useful to others. We limited our study to products which run on the Windows operating system since the vast majority of products managed at the help desk run on Windows. Conclusions drawn from the SRGM we used to make incident predictions may not hold with non-Windows products. A replication of this case study using non-Windows products would be of value to further address this aspect of validity. This case study was conducted in a large multinational company which uses hundreds of different software products. Our findings may not be directly applicable to small companies with fewer software products.

Reliability addresses how repeatable the results of a case study might be if conducted independently. We were careful to describe the incident data and scope of product distribution to the extent the confidentiality of this case study allowed, but the specific product types were not disclosed. Since product type, functionality, and the environment in which they are used influences operational behavior, a repetition of this case study may produce results that are different from ours. However, we could show the level of reliability for a rather large number of months.

6. Conclusions and Future Work

This paper describes an approach to estimate incident volume and related labor cost for a product portfolio of an IT help desk. A case study in a major international corporation shows that the approach can predict costs with relative error as little as 0.0165, in spite of a number of assumptions that had to be made to deal with the limitations of available data. We discuss lessons learned at each process step then conclude with opportunities for future work.

Criteria established for minimum incident quantity and number of installations per product established for this step of the process resulted in 20% of the products generated 80% of the incidents. Our process accommodates adjustment of the product selection criteria in the event the product portfolio size grows or shrinks. We recommend careful attention to the minimum number of incidents per product, as this influences curve fit results. The assumption of cost apportionment to a subset of products could have been avoided had the help desk tracked labor by product, although the cost estimation techniques worked out well in spite of the assumption. Recurring use of the estimation techniques requires monthly updates to actual cost data. Arrangements for standard reporting of actual cost data would have avoided the manual aggregation of cost information obtained through contact with help desk managers. As discussed earlier, we were constrained by the limitations of the linear regression tool to no more than 16 clusters motivated a non-standard approach to dendrogram height selection to define clusters. Selection of

a different regression tool that has less stringent limitations would facilitate a more standard approach in cluster selection. In spite of this constraint, we were successful in demonstrating incident prediction accuracy through the selection of product clusters vs. analyzing a full product portfolio in a prediction model.

We demonstrated how incident prediction accuracy can be obtained through the selection of cluster representatives (vs. analyzing all products) to answer RQ2 “Can incident prediction accuracy be obtained through the selection of product clusters vs. analyzing a full product portfolio in a prediction model?”. We used PCA to determine the product which best represented the other products in a cluster on the basis of the number of shared principal components. To answer RQ1 regarding the use of desktop software product reliability data from the help desk to predict future incident volume, we successfully demonstrated three incident prediction methods. The All products, Scalar and Linear function prediction methods were compared on the basis of prediction accuracy and effort to arrive a predictions. The Linear function method was shown to predict incidents more accurately than the Scalar method at one month prediction intervals. Although the All products method shows promise with incident prediction accuracy, it was rejected due to requiring an analysis of all 156 products. The Linear function and Scalar methods required less effort to predict incidents. We successfully demonstrated that help desk labor data can be used to estimate costs of future incidents in response to RQ3. Linear regression was used to predict help desk labor from incident volume. The regression coefficients were used as parameters for the cost estimation model.

This case study presents opportunities for extended research in the following areas. As stated in the approach section for the scalar method, this research presents an opportunity to apply incident and cost estimation techniques with a focus on product life cycle. The reality of today’s IT environment demands changes in the product portfolio through the introduction of new or upgraded products and the removal of outdated ones. A cost estimation model that accommodates product changes would be of benefit to help desk managers.

In this case study we selected the best SRGM based on GOF. The SRGM which produced the highest R^2 value was selected as the model to predict incidents. Alternatively, we could have selected the SRGM which results in the most accurate incident prediction rather than selecting on the basis of GOF. An investigation into the correlation of GOF with prediction may confirm an alternate model selection criteria achieves improved prediction accuracy overall. As discussed in threats to validity, the generalizability of this case study is limited to the selection of Windows products. An expansion of scope to include Linux, iOS, MacOS, and other popular desktop and mobile device operating systems might better represent the IT environment of other organizations.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation under grant #0934413 to the University of Denver.

- [1] International Standards Organization (ISO) standard 20000. <http://www.iso.org>.
- [2] Carina Andersson. A replicated empirical study of a selection method for software reliability growth models. *Empirical Softw. Engr.*, 12(2):161–182, 2007.
- [3] Anneliese Andrews and Joseph Lucente. From incident reports to improvement recommendations: Analyzing it help desk data. In *Australasian Software Engineering Conference*, April 2014.
- [4] Anneliese Andrews and Joseph Lucente. On the viability of using SRGMs for IT help desk incident predictions. *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015.
- [5] A. Bose, A. Heching, and S. Sahu. A framework for model-based continuous improvement of global it service delivery operations. In *Services Computing, 2008. SCC '08. IEEE International Conference on*, volume 2, pages 197–204, July 2008. doi: 10.1109/SCC.2008.160.
- [6] C. Wohlin, and A. Amschler Andrews. Prioritizing and assessing software project success factors and project characteristics using subjective data. *Empirical Software Engineering: An International Journal*, 8(3):285–308, 2003.
- [7] H. Do and G. Rothermel. An empirical study of regression testing techniques incorporating context and lifetime factors and improved cost-benefit models. In *Proceedings of the ACM SIGSOFT Symposium on Foundations of Software Engineering*, pages 141–151. ACM, November 2006.
- [8] H. Do and G. Rothermel. Using sensitivity analysis to create simplified economic models for regression testing. In *Proceedings of the International Symposium on Software Testing and Analysis*, pages 51–61, July 2006.

- [9] H. Do, S. Mirarab, L. Tahvildari, and G. Rothermel. An empirical study of the effect of time constraints on the cost-benefits of regression testing. In *Proceedings of the ACM SIGSOFT Symposium on Foundations of Software Engineering*, pages 71–82. ACM, November 2008.
- [10] R. Ibrahim, P. L’Ecuyer, N. Regnard, and Haipeng Shen. On the modeling and forecasting of call center arrivals. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, pages 1–12, 2012.
- [11] A.K. Jain, R.P.W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, Jan 2000. ISSN 0162-8828. doi: 10.1109/34.824819.
- [12] Leonard Kaufman and Peter J. Rousseeuw. *Frontmatter*, pages i–xiv. John Wiley and Sons, Inc., 2008. ISBN 9780470316801. doi: 10.1002/9780470316801.fmatter. URL <http://dx.doi.org/10.1002/9780470316801.fmatter>.
- [13] A.G. Malishevsky, G. Rothermel, and S. Elbaum. Modeling the cost-benefits tradeoffs for regression testing techniques. In *Software Maintenance, 2002. Proceedings. International Conference on*, pages 204–213, 2002. doi: 10.1109/ICSM.2002.1167767.
- [14] D. McCarthy. Automation of help desks using case-based reasoning. In *Case Based Reasoning: Prospects for Applications, IEE Colloquium on*, pages 9/1–9/3, 1994.
- [15] Magnus C. Ohlsson, Anneliese Von Mayrhauser, Brian McGuire, and Claes Wohlin. Code decay analysis of legacy software through successive releases. In *Proceedings of the IEEE Aerospace Conference*, pages 69–81, 1999.
- [16] Magnus C. Ohlsson, Anneliese Amschler Andrews, and Claes Wohlin. Modelling fault-proneness statistically over a sequence of releases: a case study. *Journal of Software Maintenance and Evolution: Research and Practice*, 13(3):167–199, 2001. ISSN 1532-0618. doi: 10.1002/smr.229. URL <http://dx.doi.org/10.1002/smr.229>.
- [17] M.C. Ohlsson and C. Wohlin. Identification of green, yellow, and red legacy components. In *Procs. International Conference on Software Maintenance*, pages 6–15, 1998.
- [18] Jeff Rumburg. Metric of the month - cost per ticket. <https://www.thinkhdi.com/~media/HDICorp/Files/Library-Archive/Insider%20Articles/Cost%20per%20Ticket.pdf>.
- [19] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Engg.*, 14(2):131–164, apr 2009. ISSN 1382-3256. doi: 10.1007/s10664-008-9102-8. URL <http://0-dx.doi.org.bianca.penlib.du.edu/10.1007/s10664-008-9102-8>.
- [20] Per Runeson, Martin Høst, Austen Ranier, and Bjørn Regnell. *Case Study Research in Software Engineering*. John Wiley & Sons, inc., 1st edition, 2012. ISBN 978-1-118-10435-4.
- [21] He Shu-guang, Li Li, and Qi Er-shi. Study on the continuous quality improvement of telecommunication call centers based on data mining. In *Service Systems and Service Management, 2007 International Conference on*, pages 1–5, 2007.
- [22] S.G. Steckley, S.G. Henderson, and V. Mehrotra. Performance measures for service systems with a random arrival rate. In *Simulation Conference, 2005 Proceedings of the Winter*, pages 566–575, 2005.
- [23] E. Stoker and J. Bechta Dugan. When does it pay to make software more reliable? In *Software Reliability Engineering, 2003. ISSRE 2003. 14th International Symposium on*, pages 321–331, Nov 2003. doi: 10.1109/ISSRE.2003.1251053.
- [24] C. Stringfellow and A. Andrews. Deriving a fault architecture to guide testing. *Software Quality Journal*, 10: 299–330, 2002.
- [25] C. Stringfellow and A. Andrews. An empirical method for selecting software reliability growth models. *Empirical Software Engineering*, 7:319–413, 2002.
- [26] T. M. Khoshgoftaar and D. L. Lanning . Are the principal components of software complexity stable across software products? In *Procs. International Symposium on Software Metrics*, pages 61–72, October 1994.

- [27] T. M. Khoshgoftaar and E.B. Allen and N. Goel and A. Nandi and J. McMullan . Detection of software modules with high code debug churn in a very large legacy system. In *Procs. International Symposium on Software Reliability Engineering*, pages 364–371, October 1996.
- [28] Min Tang, Bryan Pellom, and K. Hacioglu. Call-type classification and unsupervised training for the call center domain. In *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on*, pages 204–208, 2003.
- [29] Julia Varnell-Sarjeant, Anneliese Amschler Andrews, Joe Lucente, and Andreas Stefik. Comparing development approaches and reuse strategies: An empirical evaluation of developer views from the aerospace industry. *Journal of Information and Software Technology*, 61(0):71 – 92, 2015.
- [30] U. Wattanachon and C. Lursinsap. Agglomerative hierarchical clustering for nonlinear data analysis. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 2, pages 1420–1425 vol.2, Oct 2004.
- [31] Claes Wohlin and Anneliese Amschler Andrews. Analysing primary and lower order project success drivers. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering, SEKE '02*, pages 393–400, New York, NY, USA, 2002. ACM. ISBN 1-58113-556-4. doi: 10.1145/568760.568829. URL <http://0-doi.acm.org.bianca.penlib.du.edu/10.1145/568760.568829>.