

ESTIMATION MYOPIA: TINKERING WITH PERCEPTION
IN SOFTWARE ESTIMATION AND PLACEBO
ESTIMATION IN ENTERPRISE
DATA WAREHOUSING

by

HAZEM HASAN YASSIN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE
THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2014

Copyright © by Hazem H. Yassin 2014

All Rights Reserved



Acknowledgements

First and above all, I praise God, the almighty for providing me this opportunity and for granting me the capability and the tenacity to complete this work successfully. Additionally, I would like to thank my supervisor, David Levine for guiding me and supporting me over the years. Without his selflessness, inspiration and continued reinforcement, I would not have been able to finish this work. Mr. Levine sets a high bar with his excellence as a mentor, instructor, and as a role model. In short, his friendship, constant encouragement, and valuable advice gave me the conviction and reassurance necessary to complete this work. Similarly, I would like to thank Christoph Csallner and Bahram Khalili for their enthusiasm in serving on my committee and supporting my research.

I'm also grateful to Southwest Airlines, my leaders, and my colleagues for the support they were able to give me during this project. Special thanks to all the subject matter experts Jefferson, Tamer, Murad, Alia, David, Gary, Murat, Shawn, Pat, Janice, Steve, Osvaldo, and Prashant who were directly involved in the experiments and calibration exercises. Without them, this thesis would not have been possible.

Lastly, my sincere appreciation and gratitude goes out to my dear family members for their unwavering support and constant encouragement throughout my educational journey. A special thanks to my parents, who directly contributed to my academic and professional success through their constant encouragement to aim higher and reach beyond my comfort zone. They helped me through all the hardships and were there for me to lean on. I love you all.

November 14, 2014

Abstract

ESTIMATION MYOPIA: TINKERING WITH PERCEPTION
IN SOFTWARE ESTIMATION AND PLACEBO
ESTIMATION IN ENTERPRISE DATA
WAREHOUSING

Hazem H. Yassin, M.S.

The University of Texas at Arlington, 2014

Supervising Professor: David Levine

The goal of this study is to explore an effective way to provide timely and accurate size estimates for software and for an enterprise data warehouse (EDW). Several research papers attempt to adapt function point (FP) analysis to EDW, but there is not much research in comprehensive techniques to estimate large EDW projects. Despite the generality of FP, it is challenging to employ in an EDW environment. This thesis describes such a technique. Additionally, the thesis provides an overview of general estimating approaches, techniques, models, and tools.

This work presents a software tool that is a custom built estimation utility that takes into account nuances of an EDW. Some of these nuances include type of technology being built; build object complexity, data complexity, and source to target mappings. The utility then uses these components to estimate project effort, and at the same time, provides a common mechanism to communicate such mission-critical estimates to planning teams, delivery teams, managers, and software architects.

To evaluate the effectiveness of this tool, this work then shifts to a quantitative analysis section that compares the estimated numbers from multiple large-scale projects,

with data from actuals. Specifically, the analysis examines estimates produced by expert judgment techniques and then compares these estimates to ones produced by the estimation utility.

Following that, the differences between the two data sets provide a foundation for some statistical analysis and some comparisons of numerous behavioral drivers. Finally, evaluating the three large commercial EDW projects at a national airline, the tool predicted the actual project level of effort within ten to twenty percent accuracy.

Table of Contents

Acknowledgements	iii
Abstract	iv
List of Illustrations	viii
List of Tables	x
Chapter 1 Introduction.....	1
1.1 Software Estimation Analogies.....	1
1.2 Estimate ≠ Commitment ≠ Target.....	7
1.3 Relationship between Estimates, and Plans	9
1.4 Important Estimation Concepts	10
1.5 Estimation Challenges	12
Chapter 2 Related Work.....	18
2.1 Estimation Techniques	18
2.2 Estimating Approaches.....	18
2.3 Estimation Methods	19
2.4 Estimation Methods Comparisons.....	25
Chapter 3 The Enterprise Data Warehouse.....	27
3.1 General Overview	27
3.2 The Enterprise Data Warehouse (EDW)	28
3.3 Data Taxonomies.....	31
3.4 EDW Capabilities.....	32
3.5 Project Classifications	33
3.6 Styles of BI.....	35
Chapter 4 Estimation Utility.....	43
Chapter 5 Experiment	50

5.1 Experiment Motivation	51
5.2 Experiment Questions	52
5.3 Experiments	53
5.4 Experiment II.....	66
5.5 Experiment III.....	78
Chapter 6 Results	86
6.1. Results.....	86
Chapter 7 Conclusion.....	92
Appendix A Chapter 2 Supplemental Material	93
Appendix B Experiments.....	95
Appendix C Threats to Validity.....	97
References.....	107
Biographical Information	112

List of Illustrations

Figure 1-1 Process Maturity Percent of Art Based vs. Science Based (4)	4
Figure 1-2 Classical view of software estimation process (18).....	10
Figure 1-3 Actual cost estimation process (18).....	11
Figure 1-4 Factors Contributing to Project Failure (20)	12
Figure 1-5 An estimate is a distributions (24, 59).	15
Figure 1-6 Issues in estimating effort (13)	17
Figure 2-1 The WMFP algorithm uses a 3 stage process (42).....	24
Figure 2-2 Line counts and file sizes of PNR Graphs.....	25
Figure 2-3 Function Points calculations for the SAAS_pnr_parser ETL graph.....	26
Figure 3-1 Enterprise Insights Simple Information Flow	29
Figure 3-2 Absenteeism example across different organization levels	30
Figure 3-3 EDW Detailed Information Flow	31
Figure 3-4 EDW Project Classifications	33
Figure 3-5 Enterprise Insights Project Classifications	33
Figure 3-6 Styles of BI.....	35
Figure 3-7 Example of an Operational Report	36
Figure 3-8 Business Object Report Weekly Performance Summary.....	37
Figure 3-9 Example of Web Intelligence (WEBI)	38
Figure 3-10 Example of an OLAP Cube (50).....	39
Figure 3-11 Example interactive dashboard	40
Figure 3-12 Example of a scorecard that tracks on time performance.....	41
Figure 3-13 Example of EDW statics graph used for data analysis	42
Figure 3-14 Example of a custom built BI application.....	42
Figure 4-1 Opportunity Assessment Process	43

Figure 4-2 Estimation Utility Features.....	44
Figure 4-3 Estimation Tool Input parameters.....	45
Figure 4-4 Estimation Tool Input parameters.....	46
Figure 4-5 Estimation Tool Complexity Ratings.....	47
Figure 4-6 Estimation Tool Planning Ratings.....	48
Figure 5-1 EDW impact analysis spreadsheet for international.....	50
Figure 5-2 SWA strategic initiatives impacting the EDW	51
Figure 5-3 High-level descriptions of impacted EDW systems	57
Figure 5-4 Experiment I – WP205 & WP 116 EI teams SMEs, Utility, and Actuals.....	60
Figure 5-5 Experiment I – Matrix of scatter plots for SMEs, Utility, and Actuals	63
Figure 5-6 Experiment I Boxplot – SMEs, Utility, and Actuals	64
Figure 5-7 Personnel Projection Matrix – employment projections details	67
Figure 5-8 List of impacted subject area and graphs that need to be migrated	70
Figure 5-9 List of impacted subject area and graphs that need to be migrated	72
Figure 5-10 Project Resources Capacity Step Charts	74
Figure 5-11 Planned resource allocation against actual capacity	74
Figure 5-12 Actual resource allocation and capacity	75
Figure 5-13 Experiment II – Matrix of scatter plots for SMEs, Utility	76
Figure 5-14 Engagement Model – Proposed FTEs	79
Figure 5-15 Essbase Upgrade 2014 - Development Freeze Calendar	80
Figure 5-16 Essbase Upgrade Estimate	82

List of Tables

Table 1-1 Influencing Factors and Characteristics of Art vs. Science Estimating	3
Table 1-2 Similarities between Building a House and Developing Software (5, 6, 7)	6
Table 2-1 Estimation Approach with Examples of Approach (26, 27)	18
Table 2-2 Advantages and Disadvantages of the Consensus Methods (30, 31, 32, 33) .	20
Table 2-3 Metrics for the WMFP Parser (41)	23
Table 2-4 Function Points Variants	24
Table 2-5 Basic COCOMO calculations for the SAAS_pnr_parser ETL graph	25
Table 3-1 Summary of services that an EDW can provide	27
Table 5-1 Deep Dive into experiment 1's estimates from group 1	62
Table 5-2 One-Sample T test for the SMEs, utility, and the actuals	64
Table 5-3 Pearson correlation displayed for SMEs, Utility, and Actuals	65
Table 5-4 Engagement Model – Proposed FTEs (Full-time equivalent)	66
Table 5-5 Results Analysis Postmortem – Actual FTEs (Full-time equivalent)	73
Table 5-6 Pearson correlation displayed for the two estimation rounds	77
Table 5-7 One-Sample T test for the SMEs high round I against utility low	84
Table 5-8 One-Sample T test for the SMEs and the utility (highs from round II)	84
Table 6-1 Size actuals and their respective estimation accuracy values	91
Table A-1 Basic COCOMO Model (56)	94
Table B-1 Personnel Projection Matrix – SOW employment projections details	96

Chapter 1

Introduction

Estimation myopia refers to the short-sightedness and hurried approach of creating placebo estimates to satisfy stakeholders.

You've overestimated this project by a third; go away and cut it by a third.' The division head had 3 or 4 useful ways to cut the project down by a third, but I kept saying I couldn't do it and the project was killed. I walked out of there wondering what had gone wrong. I did what was probably the best estimate that had ever been done in that company, and the project was killed because of it. – Daniel Galorath, A CAI State of the Practice Interview

The goal of this study is to explore effective ways to provide timely and accurate estimates for an enterprise data warehouse (EDW). With the exception of a few papers that attempt to adapt function point analysis to EDW, there is not much research in way of a comprehensive techniques to estimate large EDW projects. The research contained within describes such a technique. Additionally, the thesis provides an overview of general estimating approaches, techniques, models, and tools.

1.1 Software Estimation Analogies

Software estimation is one of the most thought-provoking and essential activities in software engineering. Moreover, proper planning and project control is not possible without verifiable and accurate estimates. Oftentimes, producers and consumers of estimates do not estimate software well nor do they use these estimates properly. According to Frederick Phillips Brooks Jr., software engineer, computer scientist, and author of *The Mythical Man-Month*: "It is very difficult to make a vigorous, plausible, and job-risking defense of an estimate that is derived by no quantitative method, supported by little data and certified chiefly by the hunches of the managers" (1). For that reason, most people refer to software estimation as purely black art (13).

Another common view is that, software estimating is an art that is based on some multiplier of a gut feel coefficient. Others would argue that there a strong connection between software estimation as a science and software estimation as an art. Some say it is like building a house. Others emphatically say the opposite.

1.1.1 The Art and Science Analogies

Many believe that software estimation is both an art and a science. Estimating draws on skills and experience of everyone that is providing the estimates. Because of its virtual nature, software development can be hard to grasp or visualize. Some elaborate estimates take into consideration important influences such as resources, their development history, and their proficiency. Often times, such estimates attempt to make educated guesses as to the duration and work needed for a set of given tasks or development effort. In this sense, it's considered an art to know how long such tasks will take, the resources required, and more importantly what set of skills will be needed to complete such tasks. Furthermore, advocates of the art techniques argue that this guesswork is needed because it is frequently based on vast assumptions and an element of art is always needed to arrive at such size and effort estimates. Similarly, these estimates are regularly exercised as a single point in time calculation, and are commonly associated with Top-Down type estimates (see section 2.1 Estimation Techniques).

On the other hand, the advocates of the scientific techniques would argue that the earlier arguments are merely guesses at best, and that these projections are better served if they are employed only as a last step. To the science advocate, this last step should occur only after a sound mathematical or robust statistical analysis has been performed. The science advocates further articulate that estimating is not a onetime activity that is only done at the beginning of a project. Rather estimates should periodically be refined throughout the project in order to account for unforeseen

occurrences, project adjustments and unknown variables that arise during development (2).

To that end, the science enthusiasts affirm that for projects that are vague, it is even more important for one to start with a relevant scientific based measurement. They argue that this should be the case even if project has no established history, no documented deliverables, and no experienced members. Conversely, for projects with well-established history, well-documented features, and a reliable delivery team, the recommendation is to primarily rely on science. Specifically, the recommendation is to use science as a measurement of size and then artistic techniques can be applied as a refinement.

Another eloquent way to articulate these arguments, is a famous statement made by Lord Kelvin where he summarized the importance of science and being able to measure: "When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind" (3).

Table 1-1 Influencing Factors and Characteristics of Art vs. Science Estimating

Software Estimation Influencing Factors	Software Estimation as an Art	Software Estimation as a Science
Estimates are based on	Art needs a lot of creativity. Estimates are based on the person's creativity	Based on experiments, facts and sample data
Estimates Depend on	Depends on people's ability to estimate	Depends on repeatable and reproducible processes
Estimates Consistency	Not consistent	Consistent
User Requirements Sizing	Requirements are not sized and standards for sizing are not defined at organizational level	Requirements are sized correctly and standards for sizing efforts are defined at the organizational level such as the ISO/IEC 20926:2003 standard for functional

Table 1.1—Continued

		sizing, or the IFPUG Counting Practice Manual (CPM) for counting function points
Historical Data	Unreliable data due to not having a data collection process or a database	Reliable data due to having a thorough data collection process or a historical database
Project Specific Influencing Factors	Influencing factors are not documented and no distinction is made between requirements and influencing factors (type of technology, platform, risk, quality) during estimation.	Influencing factors like project risk, known issues, all have to be documented at the time of estimation or re-estimation These have to be documented along with other project data in a historical database
Maturity of Organizational Processes	Processes at organizational level are either not defined, or defined but not followed, or even considered as liability by the projects	Well established estimation processes exist along with other effective organizational processes that complement the estimation process

The information provided earlier suggests that as estimation processes mature and formalize estimation gravitates to being more of a science in nature than being a form of art. Also, it is not about whether estimation is an art or purely a science, but rather there's an element of less formal artistic techniques in applying the science and vice versa. In short estimation is dependent on both. In this context, the key is in understanding when do to trust experiences and gut when finalizing project estimates (4).

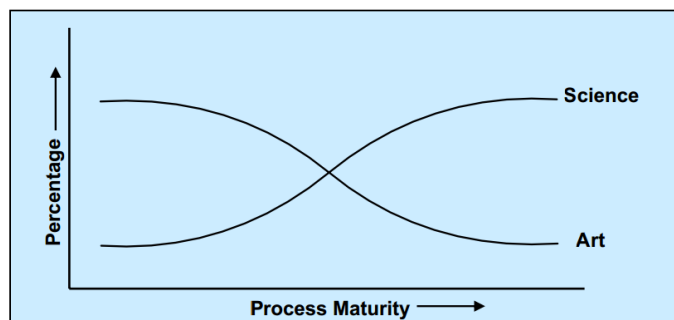


Figure 1-1 Process Maturity Percent of Art Based vs. Science Based (4)

1.1.2 The Building a House Analogy

Due to the abstract and virtual nature of the software development process, it is sometimes hard to estimate or even to visualize the size of the sought-after features of a desired end product. In spite of that, and to help individuals come to grips with the intangible nature of estimating and building software, some SMEs adopted a clever example that relates the software development process to building a house (5).

The other opinion is that writing software is not like building a house. Some view this as an unfortunate overused metaphor that confuses and frustrates consumers that are unfamiliar with the intricacies of custom-built software. The section that follows outlines the similarities and differences between software estimating and building a house.

1.1.3 Software Estimating is Like Building a House

As in any venture, whether it is building a custom home or developing an intricate software solution, the very first step before embarking on such an undertaking is to define or state the goals of the endeavor. That's why architects typically provide detailed plans and blueprints to ensure that what they are planning to build will serve the required purpose of the buyer. Architects and their respective customers use blueprints to understand what they are going to build. Similarly, software developers use functional requirements to understand the scope of the problem they are trying to solve.

After that, when the initial requirements are solidified, a preliminary estimate of cost is provided. Typically functional requirements are reviewed with an architect who will then provide proposals on better ways to meet the requirements. The architect can then provide an estimate of cost. In the building a house example, this is typically done on a square meter basis. In the software application example, this is considerably more complex.

Similarly, subsequent house construction and software planning re-estimates are needed after each major millstone. The table below further outlines chief ways in which software development is comparable to building a house. When coupled with mature processes, the steps below can provide a systematic approach to develop with realistic time estimates:

Table 1-2 Similarities between Building a House and Developing Software (5, 6, 7)

Software Process Steps	House Process Steps	Comments	Estimation Accuracy
Define the high-level requirements	Describe the house specifications	This involves describing the vision of the project. High-level estimates are provided.	Low
Identify Actors and Roles	Identify who will reside in the house?	Different actors generally have different roles and need different levels of access. High-level data dimensionality estimation	Low
Identify business process (use case)	What will the residence expect to do in the house?	Detailed requirements are needed to describe data flow and how actors will interact with the system	Medium
Define the data	What do actors need to use/consume?	Detailed database requirements for all data. Detailed data dimensionality estimation	Medium
WBS detailed tasks and mockups	Specifics on how each room will be utilized/laid out	Detailed requirements are needed along with detailed task level estimates.	High
Integration Planning	How will the house interact or connect with its surroundings?	Contact points need to be identified. Estimates need to reflect that the system is not being built in a vacuum.	High
Test Planning	House Inspection	Clearly defined entry exist criteria and test plans. Testing time needs to be accounted for.	High
Schedule	When can the house be completed?	Detailed estimates	Very High

1.1.4 Software Estimating is NOT Like Building a House

For the experts in this camp, writing software is not like building a house, they are just loosely related concepts. To them this is an unfortunate metaphor that incorrectly

exaggerates the similarities between custom-built software projects and that of the building industry. Even though building a house and building custom software both require some upfront design, consultation with experts, blueprints or specs, specialized tools, and estimators, the analogy ends there.

In custom software projects, priorities can change rapidly. Often times, initial estimates are far from accurate and accurate estimates only emerge after having enough scope and real requirements identified. Unlike building a house it is impossible to document every last detail of a project in a comprehensive spec that outlines the end product accurately before ever starting development.

Furthermore, with new emergent software methodologies, software projects start with only a common vision or an initial plan. Unlike the waterfall approach, projects that are using new methodologies can be harder to estimate initially. This is due to not having elaborate architectural documents, or detailed requirements. At this point, the software builders start developing iteratively, exposing risks, and refining estimates as they learn more and make progress. In this way, estimates are refined based on progress being made, and based on functioning software rather than a onetime activity that is founded on a draft specification (8). See section 2.1.1 for more on estimation methods and methodologies.

1.2 Estimate ≠ Commitment ≠ Target

When an executive team asks for an estimate, it is often the case that they are asking for a commitment or at the very least, a precise plan to meet a specific target. Unfortunately, in most organizations the word “estimate” has become synonymous with the word “commitment.” Hence, understanding the similarities and the dissimilarities between estimates, targets, and commitments is critical to formulating plans that are based on better estimates.

Along these lines, a “target” is considered to be a description of a desirable business objective. In the same way a target defines the objective, it also defines what success will look like (9). An example of a target might be the need to meet a new federal regulation. Generally speaking, business customers have goals and good reasons to establish specific targets but most of the time these targets should be independent from the software estimates. The problem lies in the fact that even if a target is desirable or even in some cases as in the earlier example, mandatory, this does not necessarily mean that it is achievable (10).

Likewise, it is important to distinguish that even though a target is a description of a desirable business objective or specific goal to be pursued a “commitment”, on the other hand, is a promise to deliver on that target. More specifically, a commitment is an assurance that an agreed upon goal or an established bar that is set high to achieve certain performance levels, will ultimately manifest in specified functionality that will be pursued and honored (11).

Conversely, an estimate is a prediction of how long a project will take or how much it will cost to complete it. More precisely, an estimate is a statement of probability that takes the form of a forecast. The estimate promises to deliver a number of desired capabilities or features within a projected time period using available resources (12). In this context, Steve McConnell in his book *Software Estimation: Demystifying the Black Art* describes a good estimate as “an estimate that provides a clear enough view of the project reality to allow the project leadership to make good decisions about how to control the project to hit its targets” (13).

To summarize, it is important not to let a commitment come to pass as an estimate. This is because a commitment can sometimes be more aggressive, more conservative, or in some instances it may even be the same as an estimate. “Simply put,

a commitment represents a promise that will be honored and a target is a goal that will be pursued. An estimate is a prediction based on imperfect information in an uncertain environment” (10).

1.3 Relationship between Estimates, and Plans

It is important to understand the relationship between estimation and planning. Simply put, estimates are responsible for shaping plans. That's why a good estimate is the foundation for achieving a sound plan. However, a plan is not considered to be the same as an estimate. On one hand, as far as estimates are concerned, the goal of an estimate is to accurately and objectively arrive at a precise correct answer. On the other hand, the goal of a plan is to establish the means to seek a particular result. It is very important not to force an estimate to come out to a misleading answer. Consequently “estimation” is regarded as an analytical, unbiased process, whereas “planning” is considered to be a goal shaping and purposefully biased process (14).

Additionally, estimates are vital for generating good plan. Estimates form the foundation to many important components that make up a good plan. For example, accurate estimates are applied when creating a detailed project plan schedule, creating a work breakdown structure (WBS), finding the critical path of a project, cross prioritizing of deliverables, and iteration planning.

The fundamental differences between estimation and planning are important and significant enough to warrant a clear partition between the two concepts. Otherwise, coalescing plans and estimates will lead to weak estimates and very deficient plans. This is most important when estimates are dramatically different from desired project plans or firmly set targets. In this case, the plan is assumed to be incurring a high level of risk that needs to be mitigated by providing remedial steps to bridge the that gap between what the estimate is forecasting and what the plan is targeting. Otherwise, if the estimates are

within range of the proposed target, then the plan is trending towards a sounder more realistic implementation and therefore can usually assume less risk (14).

1.4 Important Estimation Concepts

Software estimation is a process of forecasting a fit-for-purpose use of effort to create new or to sustain existing software. Stereotypically, the forecasting is an optimistic prediction that far too often is made based on imperfect and ambiguous input. Predictions are then used to create effort estimates that are used as input to project plans, budget plans, and even strategic investment plans (14).

As a result, most consumers of these estimates use such predictions to develop a plethora of project plans to forecast the outcome and feasibility of a project. Hence, the purpose of software estimation is not to merely predict a project's outcome, but more importantly estimation is used to assess whether a project is even feasible. By the same token, feasibility implies that for a project to meet its target goals, realistic and reliable estimates are needed to determine if the available project controls are going to be sufficient to meet the agreed upon project goals (15, 16).

1.4.1 Software Estimation Defined

A traditional definition of software cost estimation is that it is “the process of predicting the effort required to develop a software system. The basic input for the software cost estimation is coding size and set of cost drivers, the output is Effort in terms of Person-Months” (17).

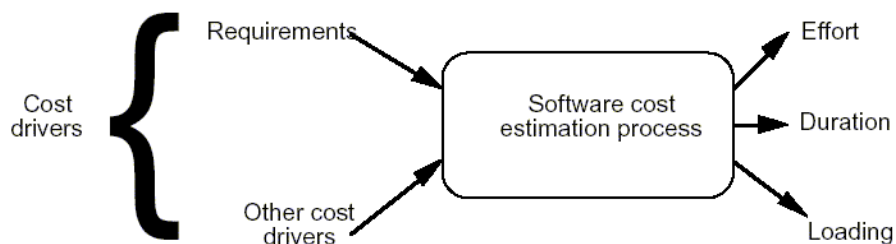


Figure 1-2 Classical view of software estimation process (18)

A more modern and comprehensive definition of software cost estimation states that it is “the process of predicting the most realistic use of effort required to develop or maintain software based on incomplete, uncertain and/or noisy input. Effort estimates may be used as input to project plans, iteration plans, budgets, and investment analyses” (19).

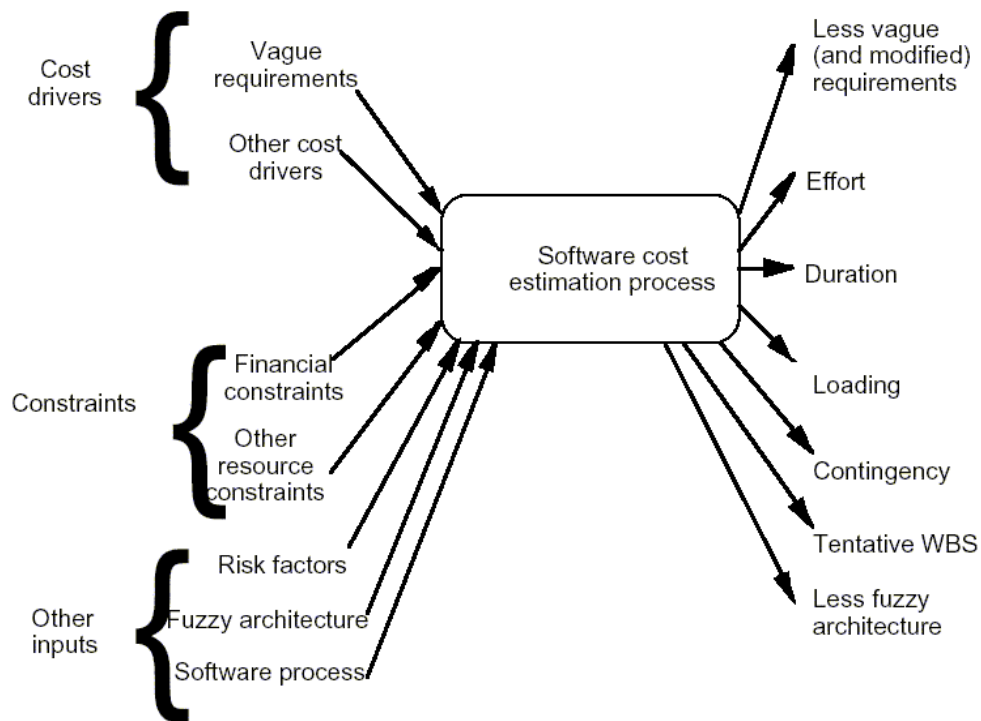


Figure 1-3 Actual cost estimation process (18)

Software Cost estimation is a challenging task in project management. It encompasses a set of techniques and procedures that is used to derive the software cost estimate. The estimation process depends on a set of inputs and then the process will use these inputs to generate the output. The output includes estimating the size, estimating the effort required, developing preliminary project schedules, estimating needed resources, and then ultimately the required schedule for software development (17, 19).

1.4.2 Impacts of Estimates on Projects

When an estimation process fails to reliably produce dependable cost estimates or if the process fails to accurately generate dependable effort duration, the failed process results in poor implementation, cost overruns, and project failure.

Many studies have shown that time and effort estimation is considered to be the main reason for most project failures. The figure below from Price Waterhouse Consulting (PWC) shows the results from a survey for factors contributing to project failure. The number one factor depicted in figure 1.2 shows that 30% of the time projects veer off track due to not having a reliable or repeatable estimation process.

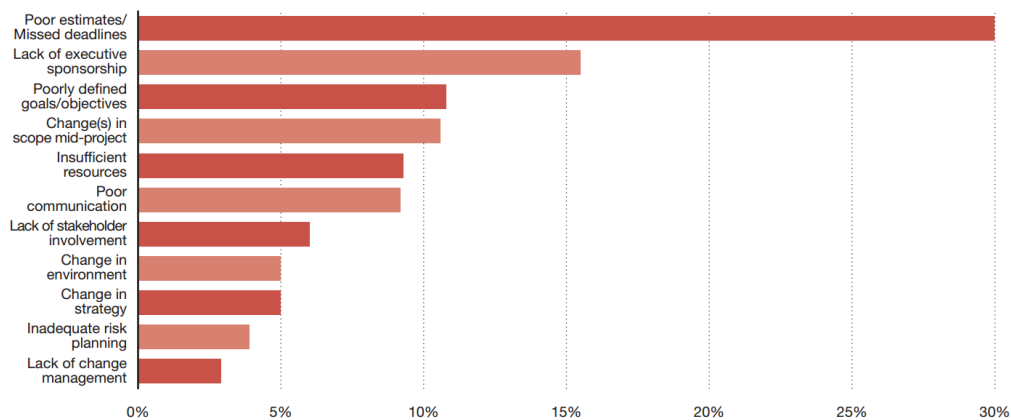


Figure 1-4 Factors Contributing to Project Failure (20)

1.5 Estimation Challenges

Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns -- the ones we don't know we don't know. And if one looks throughout the history of our country and other free countries, it is the latter category that tend to be the difficult ones (55).

Donald Rumsfeld coined the phrase 'known unknowns and unknown unknowns'. One of the biggest challenges in estimating software is the inability to possibly account for every contingency. In complex systems it is hard enough for estimations to accurately predict the known unknowns. As a result, the unknown unknowns are even harder to predict (20).

Figure 1.2 expresses the picture-perfect world of software estimation. It depicts a world where the input of the software estimation process and the output are proportionate. It depicts a perfect world where tasks fit meticulously on spreadsheets, and the amount of effort spent on a task is proportionate to the measure of one's success. Project planners would love to live in such a world.

Unfortunately, as was described in Table 1.1, the aspects of science based estimating are more comparable to the science of climatology. Figure 1.3 represents such a model where small adjustments lead to disproportionately massive impacts.

A common example of such adjustments includes estimates that are gathered at the beginning of a project. Due to fungible requirements and due to teams' limited knowledge at the inception of a project, early estimates are rarely accurate. Consequently, small adjustments to scope can lead to massive impacts.

Alternatively, estimates may be changed by stakeholders due to lack of confidence in the estimate. This might be due extraneous drivers such as politics, funding, and needing to meet predefined targets. Other factors include lack of confidence in the validity of the estimate. This in turn might be due to lack of confidence in the estimation process itself, and the inputs of the estimation process. Another challenge is sometimes stakeholders disagree about the assumptions which the estimate was founded upon.

Another case in point has to do with translation error when estimating size and effort. For example, using lines of code (LOC) to estimate software size is relatively easy. However, translating into effort or to man month can be error prone. Similarly, Function-Point (FP) estimates may show evidence of low accuracy due to subjective judgment when using organizational benchmarks to compute FPs per month (13). See section 2.1.1 for more on Estimation Methods.

1.5.1 Data Warehousing Estimation Challenges

When determining how to appropriately estimate Data Warehouse (DW) projects, the first consideration must be the unique characteristics of the DW projects being estimated. Specifically, this research is interested in the estimation challenges of DW and business intelligence (BI) application developments. The research is not focused on the hardware and infrastructure aspects of maintaining a DW.

Another challenge pertaining to estimating DW projects relate to wrong perceptions and lack of understanding of DW and BI applications. These applications have requirements that are subjective in nature and have often unique characteristics.

Also, unlike traditional applications, DW size and effort estimation complexity are directly related to three factors. The first item is linked to the number of data elements being estimated. The second major component is strongly correlated with the number of source files being estimated. Third component is connected to the quality and complexity of the data. These factors are extremely important because it is estimated that “60% - 80% of BI/DW project hours deal specifically with DATA” (22).

1.5.2 Issues in Estimating Size

Software size is one of the most influential factors of determining software cost and duration. “The road to project hell is paved with single number estimates” (23).

This quote describes the single most important issue that can make or break a project. Since tasks can vary in size and complexity, no one knows with certainty how long a task will take until development is completed.

Such variation is attributed to the probability distribution that explains estimation size. In short, size estimation is a distribution (24). Ideally, size is assigned a probability to each measurable subset of possible outcomes. The size could follow a normal distribution, a narrow distribution, or some other variant distribution.

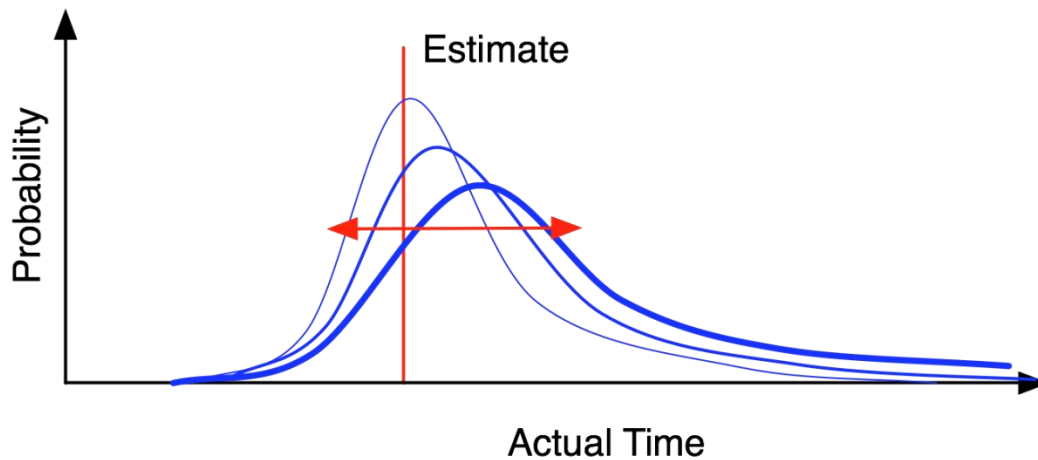


Figure 1-5 An estimate is a distributions (24, 59).

Regrettably, most estimators ignore the fact that an estimate is a distribution and instead provide a single number estimate. Alternatively, some realize the pitfalls of providing a single number estimate and instead attempt to narrow down the distribution (24). This technique is just as bad as a single point estimate because it is impossible to control all possible factors that affect any an estimate's distribution. The best way to mitigate this risk is to provide 3 estimates that take into account the distribution of best possible outcome, most likely, and worst case scenario.

Other major issues in estimating size include poor requirements and scope creep. It is paramount to have clearly defined requirements, and it is equally important to

follow a scope change process. It is important to define initial scope and track any scope changes. When scope changes occur it is extremely important to re-estimate. Failure to do so will leave the project vulnerable to cost overruns and introduce unnecessary risk.

Another obvious factor but worth noting is the lack of estimation experience in the subject area being estimated. Moreover, not having a reliable estimation process is arguably just as harmful. Failing to follow a process or not having a methodical estimation tool will lead to subjective estimates. It is extremely important to be able to defend estimates and have a demonstrable way of backing them up. Otherwise, optimistic assumptions will overrule reasonable estimates and lead to costly over commitments.

1.5.3 Issues in Estimating Effort

The number one factor that influences a project's effort is the size of the estimate. The second factor is linked to productivity (13). The third relates to how effort is computed.

For example, a common practice is that as soon as size estimates are solidified, some project managers (PMs) attempt in earnest to compute effort. The PMs often choose to produce effort estimates by comparing past projects using company historical data or even industry data. Using this method effort is computed by dividing the low end range for the size estimate by the highest productivity rate. This produces a low estimate for effort in man months. Similarly, dividing the high size with the lowest productivity gives high man month range for effort.

Sadly, most of these estimates are usually defined during the initial concept definition of a project. It is very rare for these estimates to be accurate. Nevertheless, PMs use this information to compute level of effort (LOE) and commit resources accordingly.

Unfortunately, another effort computational issue has to do with the nature of the historical data that is used to generate the effort estimates. For instance, if the date that is used to compute the effort only accounts for development time then that is going to be the only influence that is factored into the estimates (13).

Other ways to compute effort include using estimation software, industry average effort graphs, and the International Software Benchmarking Standards Group (ISBSG) function points (FP). Figure 1.5 illustrates the main issue as it relates to how effort is computed. The figure depicts ranges of values of effort estimates for the same project using different styles of computing effort. It is quite remarkable to notice how wide the ranges vary from one computational method to the next. The dot size and line thickness represents the weight of convergence or spread among estimates. In this case, more weight is given to techniques that are believed to more accurate (13).

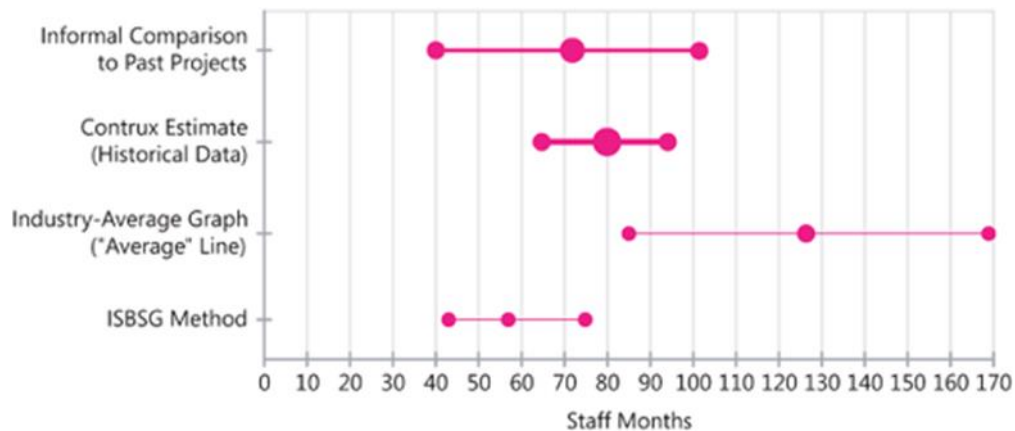


Figure 1-6 Issues in estimating effort (13)

Chapter 2

Related Work

2.1 Estimation Techniques

Today there are many estimation approaches available including algorithmic, estimating by analogy, using expert judgment, price to win, top-down, bottom-up, COCOMO, Function Points, Use Case Points, and Object Points to name a few. A common approach that is utilized by some of these methods is to measure software size using Lines of Code (LOC). This approach is the most widespread measurement of software size that is utilized by some of the methods to assess procedural languages. An alternative popular approach to LOC for sizing software is the use of Function Point Analysis (FPA). Unlike LOC, FPA measures the software functionality from an application's end user or software functionality point of view (24).

2.2 Estimating Approaches

No one method is essentially superior or more suited than another method (27). In fact, these techniques are often complimentary to one another. What determines apparent strength or perceived weaknesses of a particular method depends more on how they are being used and for what purpose (25). Below are some classifications and categories of the commonly used estimation techniques with some examples:

Table 2-1 Estimation Approach with Examples of Approach (26, 27)

Estimation Approach	Examples
Analogy-based	ANGEL, Weighted Micro Function Points
WBS-based/bottom up	Project management software, company specific activity templates
Parametric models	COCOMO, SLIM, SEER-SEM
Size-based models	Function Point Analysis, Use Case Analysis, SSU (Software Size Unit), Story points-based estimation in Agile software development
Group estimation	Planning poker, Wideband Delphi
Mechanical combination	Average of an analogy-based and a Work breakdown structure-based effort estimate

Table 2.1—Continued

Judgmental combination	Expert judgment based on estimates from a parametric model and group estimation
------------------------	---

2.3 Estimation Methods

Below are examples of some popular estimation techniques:

2.3.1 Consensus Methods

Expert judgment technique is an empirical estimation technique. This technique involves asking an expert (or a group of experts) for software cost estimation. Subject matter experts (SMEs) discuss and then use their knowledge of the proposed project or system to reach a high-level estimate (28).

A = The most pessimistic estimate.

B = The most likely estimate.

C = The most optimistic estimate.

$\hat{E} = (A + 4B + C) / 6$ (Weighted average; where \hat{E} = estimate).

Another well-known group consensus technique is the Delphi technique. Using the Delphi software estimation approach, project specifications are given to a few experts to give their opinions. The number of experts depends on their availability. It is recommended to have a minimum of three experts to have a greater range of values. Process consists of expert selection, briefing the experts, estimates collation from the experts (anonymous estimation), and finally the finalization and the merging of the estimates (29).

For larger systems that require painting an estimate canvas from a broad swath of experts, a wideband Delphi technique is preferred over the standard Delphi. In this approach to facilitate the exchange of a large volume of information, and to calibrate estimates of all estimation participants, the following steps are utilized (30):

1. Coordinator presents each expert with a specification and an estimation form.
2. Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.
3. Experts fill out forms anonymously.

4. Coordinator prepares and distributes a summary of the estimation on an iteration form.
5. Coordinator calls a group meeting, specially focusing on having the experts discuss points where their estimates varied widely.
6. Experts fill out forms, again anonymously, and steps 4 and 6 are iterated for as many rounds as appropriate. and the project was killed because of it (30).

Table 2-2 Advantages and Disadvantages of the Consensus Methods (30, 31, 32, 33)

Advantages	Disadvantages
Iterative, team-based, and collaborative (less biased than individual estimation).	Harder to quantify and may develop a false sense of confidence
Does not require historical data, but experts can take into consideration deltas between past projects and new requirements.	Can reach consensus on an incorrect estimate (not skeptical enough, biased, optimistic, or pessimistic).
Can be used to factor in impacts due to new technologies, and even personnel issues and interactions.	Might be difficult to repeat due to relying on certain experts. Also hard to get similar results with a different group of experts
Can be used at both high-level and detailed level estimation	At times, it may be hard to find more than one expert.

2.3.2 Estimating by Analogy

Estimating by analogy requires comparing a proposed project to a previously completed, but similar project. This is done by means of using information from a completed project to then extrapolate a new estimate for a planned project. It is worth noting that this method is useful for a small component, that may need to be modified or can even be extended or for larger system-level type changes.

Estimating by analogy needs a proper specification for the proposed project. Additionally, this method of estimating requires being able to select from a set of similar completed projects that have similar attributes (ideally stored in a historical database) to drive the new estimates for the proposed project. The choice of attributes includes the number of inputs, references or interfaces, number of UIs, etc.

After determining the attributes of the project, it is important to decide if there is adequate resemblance to the chosen software estimation analogies. It is important that

there be enough similarity to lead to a higher level confidence estimate. The right balance of analogies is important because few analogies lead to unconventional projects being used, and too many, may lead to the declining of the weight given to the closest analogies (35).

A common method for finding analogies is by means of “measuring Euclidean distance in n-dimensional space where each dimension corresponds to a variable. Values are standardized so that each dimension contributes equal weight to the process of finding analogies. Generally speaking, two analogies are the most effective.

Finally, we have to derive an estimate for the new project by using known effort values from the analogous projects. Possibilities include means and weighted means which will give more influence to the closer analogies” (34).

2.3.3 Putnam Model

Another popular empirical software cost model is the Putnam model (36). The original paper by Lawrence H. Putnam published in 1978 is seen as pioneering work in the field of software process modelling (37). As was discussed earlier, empirical models collect important software project data such as effort and size and then attempts to fit the data into a curve. Subsequently, the size and effort are computed by calculating, with some marginal error, the related effort using the equation that was made to fit the original data (38, 39).

2.3.4 COCOMO Models

The Constructive Cost Model (COCOMO) model is an algorithmic (parametric) software cost estimation approach. It was developed by Barry W. Boehm to become one of the most common and most transparent cost estimation techniques (38).

In the basic COCOMO model, man months are computed using parameters that depend on the type of application being developed, the development environment, and

on a size measurement based on thousands of lines of code of the target application. The mathematical model based on the data from 63 historical software projects. In Boehm's 1981 book 'Software Engineering Economics' he documented COCOMO and explained that it relates to the software development effort for a program, in man-years, to source lines of code (KLOC). The basic formula for COCOMO takes the form:

Effort Applied (E) = $a \cdot (\text{KLOC})^b$ [man-month]
Development Time (D) = $c \cdot (\text{Effort Applied})^d$ [months]
People required (P) = Effort Applied / Development Time [count]
where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a, b, c and d are given in figure A.2 (56).

Estimates from the basic COCOMO model can be made more accurately by taking into account other factors concerning the "required" characteristics of the software to be developed, the qualification and experience of the development team, and the software development environment. Other factors include the complexity of the software, desired reliability, size, efficiency (memory and execution time), team capability, and experience of team (application area and programming language).

It is worth noting that this formula is best suited for projects with proven software development teams or teams that have completed multiple projects together. After that, the formula is then used to compute a man-month or man-years estimates. It is also worth noting that since man-years are not interchangeable with years, adding programmers to a late project will only makes it later.

Additionally, the COCOMO estimation technique assumes that the requirements have already been solidified, and that these requirements generally speaking are stable. However, stable is a subjective term (36, 37).

2.3.5 Function Point Analysis

Function points (FP) can be estimated from requirements or design specs. FP allows for providing estimates in the early phases of the development life cycle. There are

several variations of function points. The most commonly used is maintained by the International Function Point Users Group (IFPUG). Below we examine the Weighted Micro Function Points (38). Month = FP divided by no. of FP's per month (Using your organizations or industry benchmark) (40).

2.3.6 Weighted Micro Function Points

Weighted Micro Function Points (WMFP) is a method developed by Logical Solutions to help estimators properly size software with very little knowledge of the code. WMFP performs automatic parsing and then computes detailed measurements of existing source code. The parser breaks down the code “into micro functions and derive several code complexity and volume metrics, which are then dynamically interpolated into a final effort score.” The measurement produced by the parser then computes a total effort score using the metrics in the table below.

Table 2-3 Metrics for the WMFP Parser (41)

WMFP Measured Elements	Description
Flow Complexity (FC)	Similar to Cyclomatic Complexity, FC measures the control path flow complexity of a program but with higher accuracy by using weights and relations calculation.
Object Vocabulary (OV)	Measures the quantity of unique information contained by the programs' source code, similar to the traditional Halstead Vocabulary with dynamic language compensation.
Object Conjunction (OC)	Measures the quantity of usage done by information contained by the programs' source code.
Arithmetic Intricacy (AI)	Measures the complexity of arithmetic calculations across the program
Data Transfer (DT)	Measures the manipulation of data structures inside the program
Code Structure (CS)	Measures the amount of effort spent on the program structure such as separating code into classes and functions
Inline Data (ID)	Measures the amount of effort spent on the embedding hard coded data
Comments (CM)	Measures the amount of effort spent on writing program comments

The parser computes the final results in three stages. In the first stage, a function analysis performs a deep dive of the source code to produce the WMFP Measured Elements computing items such as Cyclomatic Complexity, arithmetic manipulation, and manipulation of data measurements (41). In the next stage, an Average Programmer Profile Weights (APPW) is used to produce a Statistical Cost Mode to transform the measurements into an intermediate that subsequently gets translated into time. In the final step, an algorithm is used to balance and add the all the measurements and scores to produce a total effort score based on programmers work hours (42).

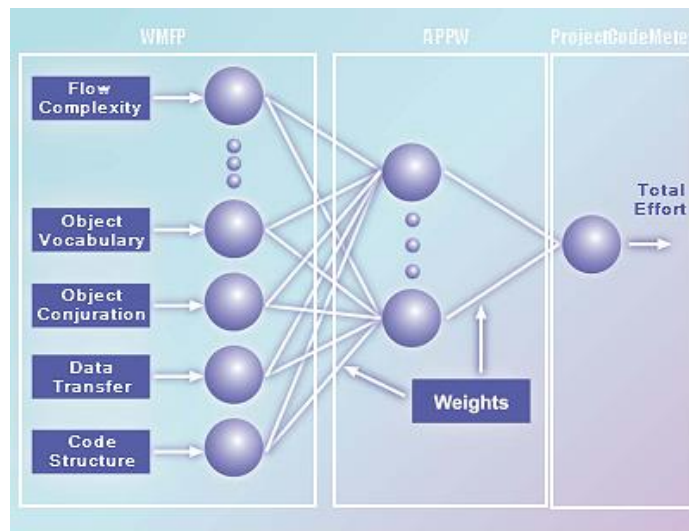


Figure 2-1 The WMFP algorithm uses a 3 stage process (42)

Other less well-known functional points variants are use case points, class points, application points, and web related measures. These variants are summarized in the table below:

Table 2-4 Function Points Variants

FP Variant	Description
Use Case Points	Used to estimate application size based on use cases (UC). This approach is very useful for organizations that use UCs as part of their development methodology (43)
Class Points	Class points were invented by PRICE Systems as a functional measurement for OO properties (44).

Table 2.4—Continued

Application Points	Application Points use a simple approach that works well with high-level software specifications. They are good alternative that help for defining things like reports or user interface (UI) screens (45).
Web Points	Web-points were developed web applications. They were invented by David Cleary to help measure function and content of static websites (46).

2.4 Estimation Methods Comparisons

This section provides a comparison of some of the estimation techniques discussed earlier and uses an EDW ETL graph as a biases for comparison. Particularly, Function Points and COCOMO are used to compare estimation results of a well-known ETL graph. The ETL graph is well-known due to its familiar and frequently modified central component.

The graphs depicted in figure 2.2 are the passenger name record (PNR) graphs. Specifically, the SAAS_pnr_parser ETL graph manipulates and loads these passenger records from the airline’s computer reservation system (CRS). In turn, the source CRS contains all the itinerary information for passengers, or groups travelling together.

```

DEV (e98085 on XLDETL01)
xldet101:/opt/etl_developers/e98085/secure/spnr/mp:
> wc -l *GO[34]00*.mp
 93628 PDPNRG0300_SAAS_pnr_parser.mp
101275 PDPNRG0400_pnr_load.mp
194903 total
xldet101:/opt/etl_developers/e98085/secure/spnr/mp:
> ls -ltr *GO[34]00*.mp
-r--r--r-- 1 e98085 swauser 7826226 Jan 16 2014 PDPNRG0300_SAAS_pnr_parser.mp
-r--r--r-- 1 e98085 swauser 8951938 Sep 23 14:11 PDPNRG0400_pnr_load.mp
xldet101:/opt/etl_developers/e98085/secure/spnr/mp:
>
  
```

Figure 2-2 Line counts and file sizes of PNR Graphs

This graph contains 93628 lines of code (LOC). Using this LOC size as an input into the basic COCOMO calculations reveals the following (57):

Table 2-5 Basic COCOMO calculations for the SAAS_pnr_parser ETL graph

Mode	a	b	c	d	KLOC	Effort	Duration	Staffing
organic	2.40	1.05	2.50	0.38	93.628	281.96	21.33	13.22
semi-detached	3.00	1.12	2.50	0.35	93.628	484.28	21.76	22.25
Embedded	3.60	1.20	2.50	0.32	93.628	835.58	21.53	38.82

The average time to modify this graph is about 3.5 weeks. However, for this project an expert SME was able to complete modifications within 2.5 weeks. In contrast, the COCOMO estimation results demonstrated how poorly basic COCOMO performs in an EDW setting. Similarly, function points were used to evaluate the same graph from figure 2.2. Using an online worksheet (58) to compute the function points yields the results displayed in figure 2.3.

Function Point Worksheet		Weighting Factor						
Measurement parameter	Count		simple	average	complex	Choice		
# of user inputs	1	X	3	4	6	6	= 6	
# of user outputs	20	X	4	5	7	4	= 80	
# of user inquiries	0	X	3	4	6	0	= 0	
# of files	1	X	7	10	15	15	= 15	
# of external interfaces	0	X	5	7	10	0	= 0	
						Count-total =	101	
Rate each factor on a scale of 0 to 5:		0 - No Influence	1 - Incidental	2 - Moderate				
		3 - Average	4 - Significant	5 - Essential				
1. Does the system require reliable backup and recovery?						4		
2. Are data communications required?						1		
3. Are there distributed processing functions?						1		
4. Is performance critical?						3		
5. Will the system run in an existing, heavily utilized operational environment?						4		
6. Does the system require on-line data entry?						0		
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?						0		
8. Are the master files updated on-line?						0		
9. Are the inputs, outputs, files, or inquiries complex?						0		
10. Is the internal processing complex?						5		
11. Is the code designed to be reusable?						0		
12. Are conversion and installation included in the design?						0		
13. Is the system designed for multiple installations in different organizations?						0		
14. Is the application designed to facilitate change and ease of use by the user?						0		
						sum of Fi =	18	
Function Point Metric =		count-total * [.65+.01*sum Fi]						
		= 84						

Figure 2-3 Function Points calculations for the SAAS_pnr_parser ETL graph

The Function points estimate for the ETL graph is 84 function points for the SAAS_pnr_parser ETL graph. If we use about 20 function points to 1 person month of effort, the LOE using function points yields about 4.2 month. Alternatively, using the same input parameters to the FP estimate from figure 2.3, the estimation tool produced a LOE estimate in the range of 3 – 4 weeks (assuming averaged skilled resources).

Chapter 3

The Enterprise Data Warehouse

3.1 General Overview

The Enterprise Data Warehouse's (EDW) mission is to provide timely, reliable, and actionable information to facilitate the best strategic and operational business decisions to support Southwest Airlines (SWA) company objectives. The EDW teams seek to deliver advanced analytic capabilities through partnership with business customers to provide relevant and insightful information to businesses.

The team facilitates access to the EDW which is used to consolidate data from many data sources both within SWA and outside SWA. The EDW has several purposes:

- Consolidate and integrate enterprise data by subject area in a relational store at the lowest level of detail necessary for reporting and populating different subject area data marts.
- Improve enterprise data quality by enforcing a usable, consistent metadata strategy.
- Identify and manage a consistent master set of data entities for the enterprise.
- Narrow the technology footprint by provide a single point of source data to be used as "system of record".

Below is table of all the functions many services that the EI department provides to its customers:

Table 3-1 Summary of services that an EDW can provide

Area	What can an EDW provide?	Resources
Customer Visualization Tier (Details on next slide)	Selecting the correct information delivery medium for your users Strategic / Operational	Wire framing, Compositions and / or POCs for creating effective UI delivery Report Definition and Design

Table 3.1—Continued

	Reports OLAP Analysis Dashboards / Scorecards	OLAP , Dashboard and Score carding design expertise
Data Assessment	Assessing source data Determining data needs	Source System Data Assessment Source System Data Analysis
Data Design	EDW Content Understanding	EDW Design and implementation in partnership with Database Administrators and Data Architects
Data Extraction	Data Extraction, Transformation and Load design and implementation	ETL expertise ETL Standards & Guidelines ETL Design Checklists and Best Practices Tivoli expertise Job Flow documentation & design expertise
Performance Metrics	Identifying financial and operational metrics for project	Metric Definition Business Metadata Creation
Ongoing Data Analysis	EDW / Corporate Reporting / Data Mart query assistance	Expertise with Teradata Queryman Expertise with other SQL tools Expertise with reporting tools for data analysis
Future data needs	Enhancements / Changes to EDW data or BI deliverables	RFS work for future enhancements / changes

Below is a brief overview of the SWA EDW and some of the customer reporting and visualization examples. Also, the overview provides a description of the foundational components and building blocks for estimates that are formulated in a data warehouse. Lastly, the overview includes examples of EDW and business intelligence artifacts that use the proposed estimation utility.

3.2 The Enterprise Data Warehouse (EDW)

"A data warehouse is a subject oriented, integrated, time variant, non-volatile collection of data in support of management's decision making process" (47).

There are three types of data warehouses (48):

- I. Enterprise Data Warehouse - An enterprise data warehouse provides a central database for decision support throughout the enterprise.

- II. ODS (Operational Data Store) - This has a broad enterprise wide scope, but unlike the real enterprise data warehouse, data is refreshed in near real time and used for routine business activity.
- III. DataMart - DataMart is a subset of data warehouse and it supports a particular region, business unit or business function.

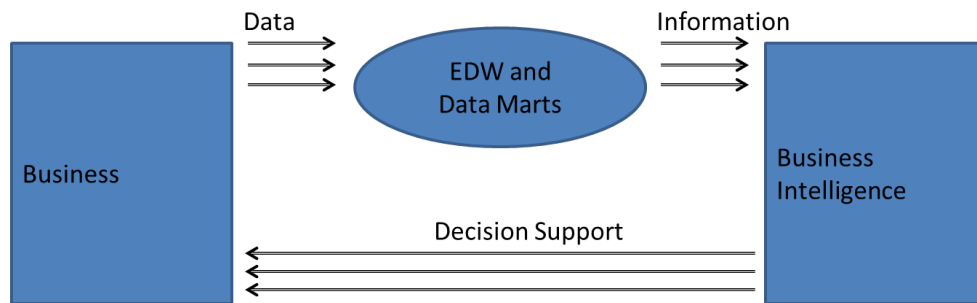


Figure 3-1 Enterprise Insights Simple Information Flow

The EDW technology department is the foundational layer that epitomizes the earlier definition. Furthermore, the time variant nature of the data in the EDW is stored at the lowest level of detail. It is important to note that the EDW only contains data that is enterprise in nature. Data is required to enable analytics and non-application reporting. The EDW is used to provide historical perspective on related items at different levels of detail, across different enterprise topics. One example is offering a perspective on boarding pass activity, by device, to enable customer traffic analytics.

Likewise, the EI department also provides the means for delivering executive and departmental scorecards for key corporate business indicators such as 'Cost Per Available Seat Mile' and other 'Operational Performance Metrics' such as 'On Time Performance' with drilldown capability to a departmental and/or Team level.

Equally important are the team scorecards and reports that provide a view of key operational performance indicators at a team level (ex. schedule adherence for a team).

Also Individual Scorecards and Reports are available to provide a view of key operational performance indicators at an individual level (ex. talk time for a reservations agent). The following illustration of absenteeism demonstrates the different levels of possible reporting scenario to support the needs that were described earlier:

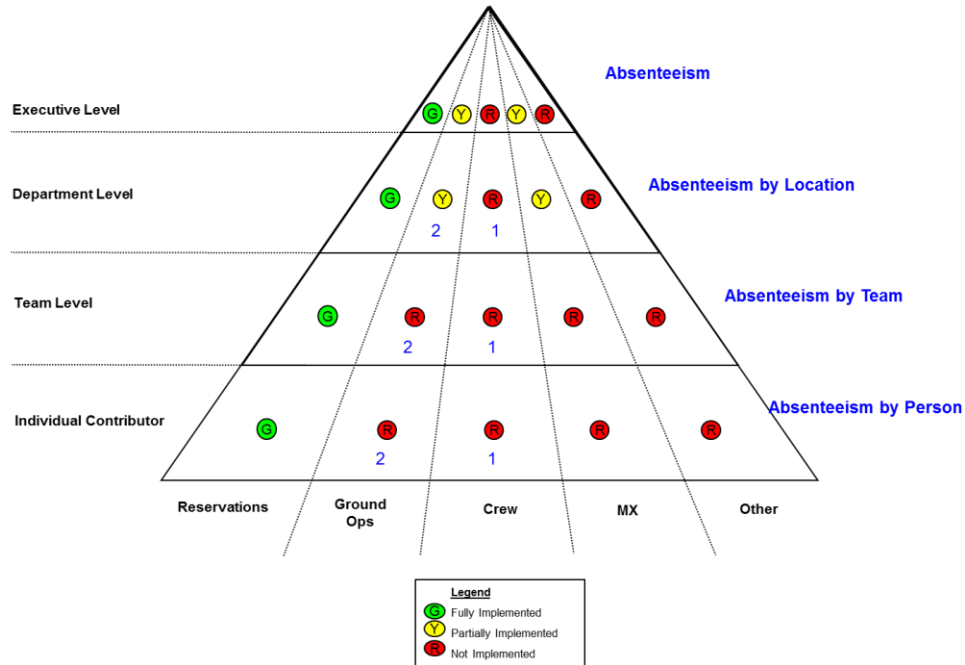


Figure 3-2 Absenteeism example across different organization levels

Such examples provide EDW internal and external customers with valuable access to large amounts of data that can be used for historical analysis and trending. The EDW helps users by making pertinent data available to the larger community of users by expanding their decision-making ability at various levels of the organization. Users can look into the EDW to examine trends across different functional units. The EDW is therefore the go to source for broader, clean, consolidated, and deeper sets of data using historical analysis and over time trends.

The “one version of the truth” concept described earlier, improves information accuracy by enabling quality reporting and profound analysis using historical evaluation

and data retention. Additionally, this reliable quality minded design provides for cost savings for the development and consolidation of data related estimates used for reporting and analysis applications by reducing the chances of data variability across multiple domains.

More importantly the EDW fits into the overall Business Intelligence (BI) organization by compiling and cleaning up data from several sources before feeding it to BI data marts. The data marts feed operational reporting tools described below, which in turn provide data to the end users.

3.3 Data Taxonomies

The enterprise data warehouse is a collection of data primarily housed in a Teradata database (the EDW), but also spanning numerous data marts (Oracle, SQL Server, Sybase). The purpose of the EDW is to store data collected from across the enterprise and make it available for analysis and reporting to support business intelligence.

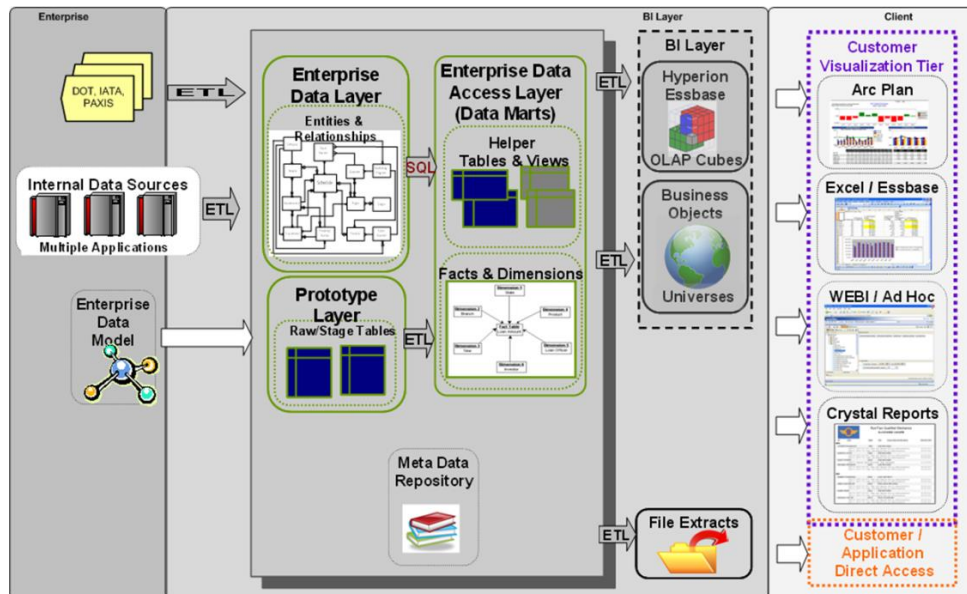


Figure 3-3 EDW Detailed Information Flow

3.4 EDW Capabilities

To sum up the capabilities of an EDW, table 3.2 describes the two main competences of a data warehouse (DW). The estimation tool is designed to work with these two capabilities:

Table 3.2 Summary of EDW / BI Capabilities

Capability	Best Used When	Example
EDW	<ul style="list-style-type: none"> • Data is required to enable analytics and non-application reporting. • Historical perspective is required. • Data is related, at a detailed level, across enterprise topics 	Boarding pass activity by device to enable Customer traffic analytics
BI	<ul style="list-style-type: none"> • Consistent application of measures and metrics is needed • Analysis is dimensional and historical in nature 	Boarding pass activity analysis by time by device by location by Customer type (or any combination of those)

As far as estimation, projects that are EDW centric are waterfall process centric in nature. An example would be ingesting flight tracking mainframe data into the EDW. These projects are, estimated by reviewing the requirements and examining the need source-to-targets mappings. Even though there is some level of iteration, the requirements and source-to-targets mappings are typically more straight forward and easier to estimate.

On the other hand, the BI projects are highly iterative, with very few hard requirements and more concept-based (e.g. the Customer has a general idea of what they want, and the prototyping phase...define/design/build...are iterated through until the Customer has what they want). These projects are harder to estimate as they do not fit well into a standard engagement model.

It is important not to confuse Agile Development with BI Development. While they do have similar characteristics (shorter durations during “sprints”), Agile

development still has much more solid requirements going into a Sprint whereas BI apps do not.

3.5 Project Classifications

The EDW capabilities described earlier detail the two main classifications of work for most development requests that come into the ED. Some of these requests are siloed in nature affecting one group or functional area, while others cut across different operating units.

The project classification below explicitly demonstrates that EDW projects are different and do not generally fit into a single set of rules to apply. For example, the interrelationships between the data and the users of that data are sometimes far too complicated and it makes it difficult to select a one size fits all profile for all projects. The figure below describes different projects that were adapted based on an article by Suzanne Robertson published in the Cutter Edge – a newsletter from the Cutter Group. The classifications below help establish some guidelines for high-level estimates for EDW projects.

Would you rather be chased by one elephant-sized mouse or a hundred mouse-sized elephants?

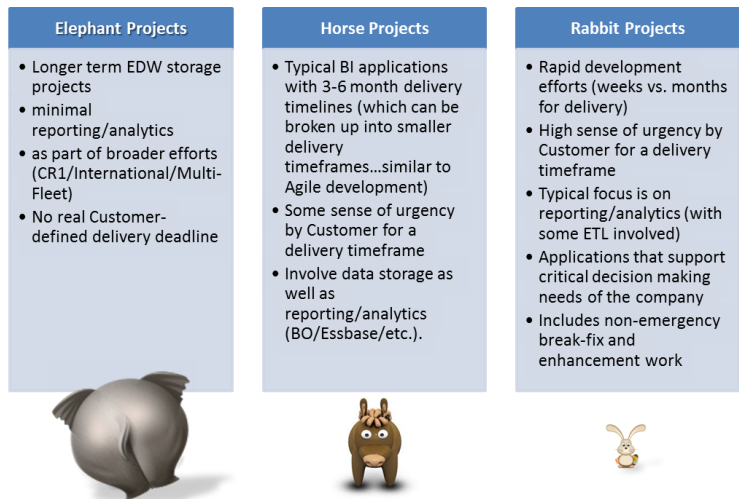


Figure 3-4 EDW Project Classifications

3.5.1 Elephant Projects

Elephant Projects are longer-term EDW projects. An example would be the large storage upgrade, platform upgrades, and large company strategic initiatives. These projects typically involve several stakeholders with several developers immersed in carrying out the work. Elephant project tend to be more formal in nature. Project plans for these types of projects include sufficient time to account for scope changes. Those requirement specifications are normally in the form of work package (WP) specifications. Example elephant projects are provided in figure 3.4. Also, high level WP examples are provided in section 5.3.

3.5.2 Horse Projects

Horse Projects are mid-term smaller EDW and BI projects. An example would be BI applications with a 3-6 month delivery timelines. These projects typically exhibit some sense of urgency from a delivery timeframe. Horse project can involve data storage as well as reporting/analytics. Horse project tend to be more iterative in nature. Project plans for these types of projects are time boxed and are constrained by the size of the project. There are only a few stakeholders for these types of projects. Requirement specifications are normally in the form of informal paperwork that facilitates the constant communication between SMEs and what the customers are seeking. Due to the nature of the incongruent source of data that makes up a BI project, SMEs often complain about the difficulty in pinning down requirements. Some SMEs describe this process as the “go fetch me a rock game.” It is very iterative in nature where you show the customer the “rock” and then the customer says “no I don’t like it, go fetch me another rock”. To use a horse riding analogy: “To keep a horse galloping, you need to keep questioning whether everything in the saddlebags is still necessary” (49).

3.5.3 Rabbit Projects

Rabbit projects are the rapid development efforts. Their timeframe is typically a few weeks. These small projects usually exhibit a very high sense of urgency by customer for a delivery timeframe. Most of these efforts typically focus on frontend reporting/analytics with some support from backend extract transform and load (ETL) developers. Despite their smaller scope, these applications typically support critical decision making needs for the company. They also include non-emergency break-fix enhancement.

3.6 Styles of BI

Business intelligence (BI) is a broad set of applications, technologies and knowledge for gathering and analyzing data for the purpose of helping users make better business decisions. The main challenge of Business Intelligence is to gather and serve organized information regarding all relevant factors that drive the business and enable end-users to access that knowledge easily and efficiently and in effect maximize the success of an organization.

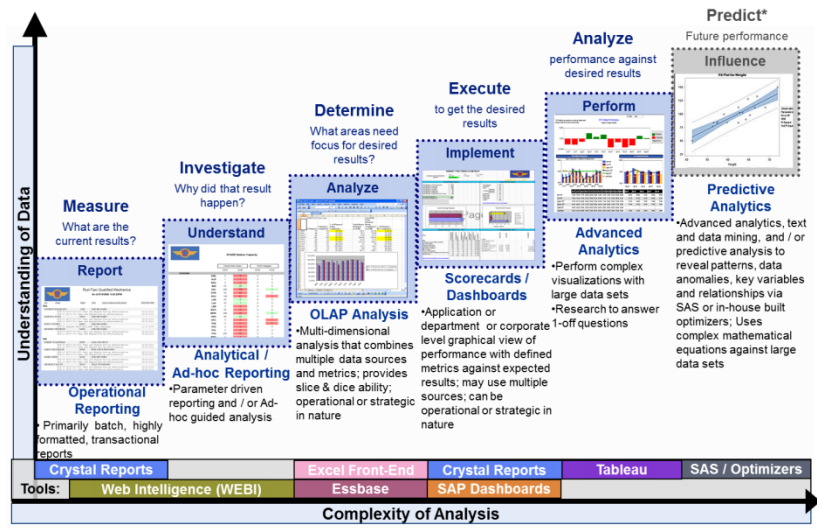


Figure 3-6 Styles of BI

3.6.1 Operational Reporting

Operational reports provide the data necessary for making timely operational decisions by delivering detailed analysis and insight into a specific application transaction database or transaction reporting database. EI has a reporting “center of excellence” to deliver diverse reporting across multiple departments. EI developers possess a deep knowledge base of multiple business areas and interdependencies.

Work spans across Business Areas, Application Teams, different project tiers, RFS and baseline efforts. The developers are considered subject matter experts (SMEs) in their respective areas and can provide guidance to help define future metrics, multiple options for look/feel, and other run on-demand or scheduled reporting activates.

Account	Orig. Bal.	1st Payment	1st Payment	1st Payment	Total Outstanding
001 - 001	100,000.00	100,000.00	100,000.00	100,000.00	100,000.00
002 - 002	200,000.00	200,000.00	200,000.00	200,000.00	200,000.00
003 - 003	300,000.00	300,000.00	300,000.00	300,000.00	300,000.00
000 TOTAL	600,000.00	600,000.00	600,000.00	600,000.00	600,000.00
004 - 004	400,000.00	400,000.00	400,000.00	400,000.00	400,000.00
005 - 005	500,000.00	500,000.00	500,000.00	500,000.00	500,000.00
006 - 006	600,000.00	600,000.00	600,000.00	600,000.00	600,000.00
007 - 007	700,000.00	700,000.00	700,000.00	700,000.00	700,000.00
008 - 008	800,000.00	800,000.00	800,000.00	800,000.00	800,000.00
009 - 009	900,000.00	900,000.00	900,000.00	900,000.00	900,000.00
010 - 010	1,000,000.00	1,000,000.00	1,000,000.00	1,000,000.00	1,000,000.00
000 TOTAL	5,000,000.00	5,000,000.00	5,000,000.00	5,000,000.00	5,000,000.00

Figure 3-7 Example of an Operational Report

3.6.2 Analytical Reporting

Analytical Reporting focuses on trending and analysis over time comparisons. Reports are typically run against integrated data from the EDW or smaller data marts rather than transactional application databases. These types of reports allow for detailed

analysis and insight into a specific area in the EDW DataMarts (DM) that has been through some transformation and validation. See section 3.1.2 The Enterprise Data Warehouse for DM definition.

The services also allow for reporting via multiple functional areas. The reports may include color-highlighted metrics, view time or run time filtering, multiple options for look/feel, and the run on-demand or scheduled reporting activities.



Figure 3-8 Business Object Report Weekly Performance Summary

3.6.3 Ad-Hoc Reporting

Web Intelligence (Webi) is SAP/Business Objects' (BO) strategic web-based tool for ad hoc reporting and analysis. It provides access to BO Universes created to meet the needs of business customers that access specific data in the EDW. The tool provides widespread query and reporting capabilities in a web friendly environment.

The EI teams help define database relationships that need to be created and optimized. They also help the end users of WEBI reports by translating complex fields

and table names into user-friendly business terms. The tool provides reusable formulas and objects built for easier query and report design. Finally query optimizers are built in so users can easily get the best performance without database query expertise.

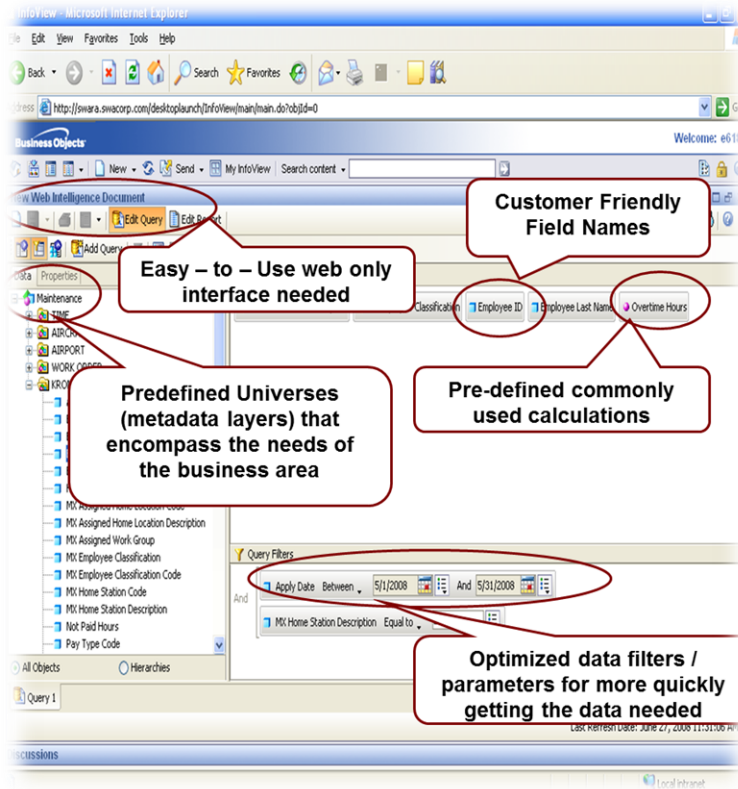


Figure 3-9 Example of Web Intelligence (WEBI) is an ad-hoc analysis tool from Business Objects (BOBJ) that provides ad-hoc query ability and “on-the-fly” report generation via an easy-to-navigate web interface

3.6.4 OLAP Analysis

Online Analytical Processing (OLAP) Analysis provides summary level views of sets of data across multiple measures to quickly pinpoint the outliers that require more in-depth analysis. This technology allows the business customers to roll up by date, station, workgroup, or other defined groupings. The OLAP team provides slice and dice views to allow users to see a dimension such as maintenance bases over multiple measures such

as number of planned checks, and total hours. The OLAP tools integrate with Excel and provide an interface into the data so it is easy to navigate and absorb. The main two technologies are Essbase and MS SQL analysis server.

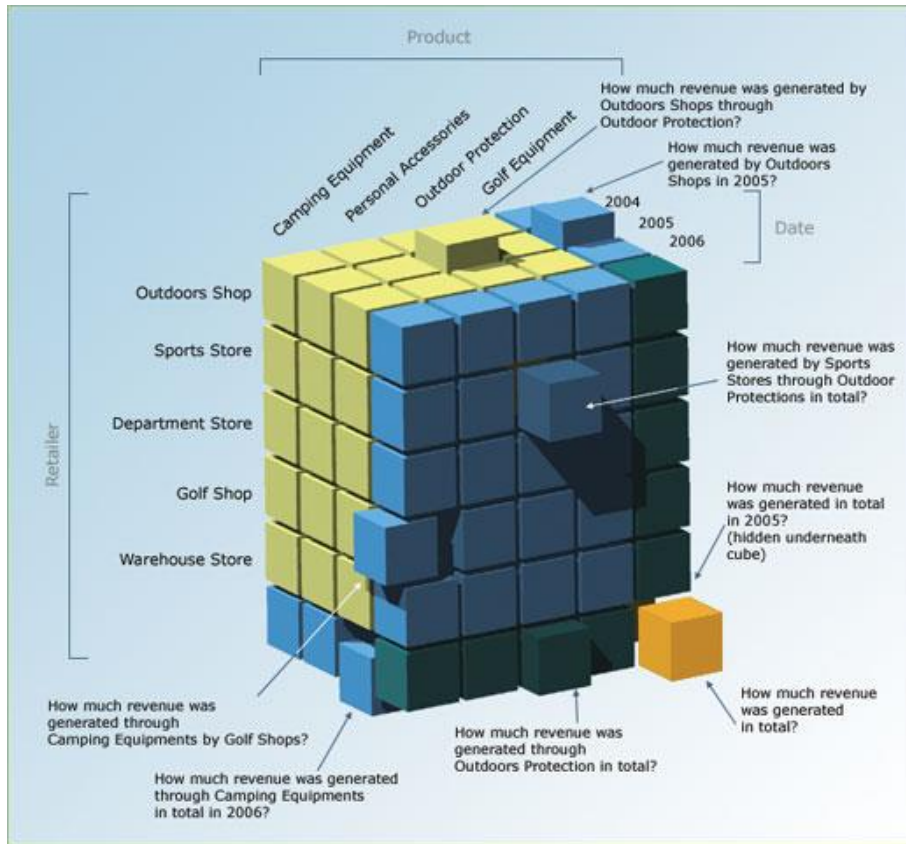


Figure 3-10 Example of an OLAP Cube (50)

3.6.5 Dashboards

A dashboard is typically a one page visualization of the most important data a user needs to see. Static dashboards focus on a given area and related key operationally focused metrics. They are run as reports and are not interactive.

Interactive Dashboards focus on multiple areas and related key operationally focused metrics. They are interactive in nature and delivered via the web. It is a common requirement that a dashboard can drill down from a chart to view individual detail items.

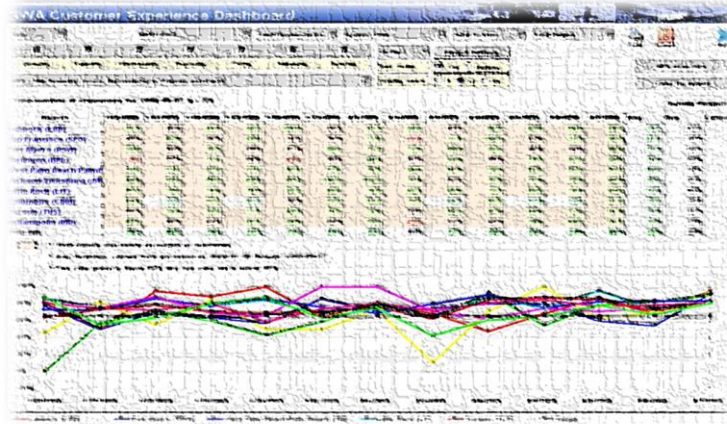


Figure 3-11 Example interactive dashboard

3.6.6 Scorecards

Scorecards deliver key metrics related to balanced SWA corporate strategies that typically incorporate multiple business areas. SWA uses a “Cascading Scorecard” approach which provides the ability to logically drill from the corporate level down to the department level scorecards for additional detail for performance and strategy management. Scorecards are requested to aid business intelligence (BI) functions of performance management by monitoring key performance indicators (KPIs) and helps SWA business customers stay on track.

Metrics are defined at each scorecard level to tell the appropriate story related to performance for the defined audience of that scorecard. Scorecards are developed for many different levels of the organization. Those scorecards are tailored towards specific audiences, but the overall look and feel/content is consistent across SWA.



Figure 3-12 Example of a scorecard that tracks on time performance

3.6.7 Statistics/Optimizers

Statistics/Optimizers provide advanced high-end statistical analysis against large data sets powered by complex mathematical engines to reveal patterns, data anomalies, key variables and relationships. Some of these advanced analytics include:

- Statistics
- Data and Text mining
- Data visualization
- Forecasting & Econometrics
- Optimization
- Model Management and Deployment
- Quality Improvement
- Predictive analysis.

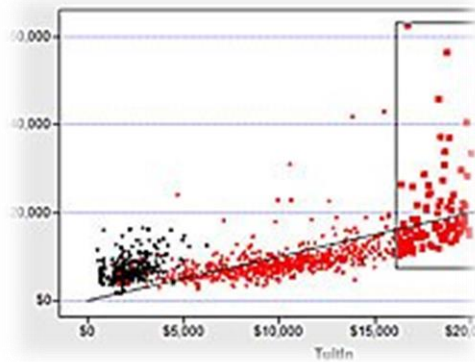


Figure 3-13 Example of EDW statics graph used for data analysis

3.6.8 Analysis Applications

Analysis Applications are custom designed and built Business Intelligence applications to meet specific customer data input and display requirements that cannot be satisfied with existing BI toolset. These custom BI applications can include data input, custom graphics, in depth reporting, exporting, and printing capabilities. Most of these are built in ArcPlan or custom designed web interfaces and inputs.

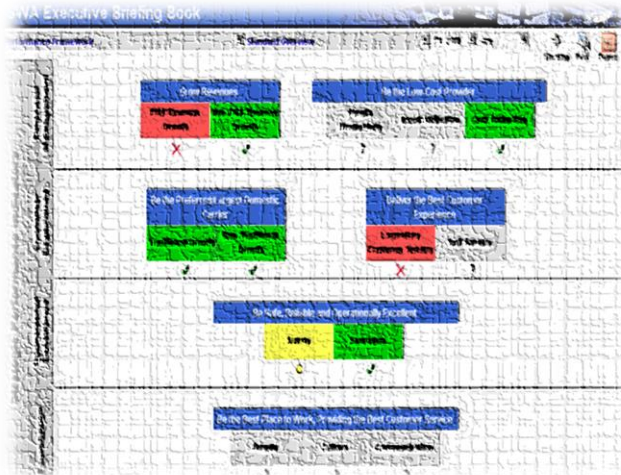


Figure 3-14 Example of a custom built BI application

Chapter 4

Estimation Utility

Most projects are strapped for time and resources leading to insufficient time to use sophisticated estimation techniques or even to hold lengthy group sessions for estimation. Despite that, the data warehousing and business intelligence team at Southwest Airlines is the conduit to the Enterprise Data Warehousing (EDW). In essence, the EDW transforms raw data into meaningful and useful information for business customers.

The custom built estimation tool for the EDW is meant to be used primarily to estimate project efforts and to provide a common mechanism to communicate those estimates to planning teams that request these high level estimates (see figure 4-1). The output of tool is meant to provide a high level estimate and to be used as a starting point for discussions with the Planners, Delivery Teams, Managers, and Architects.

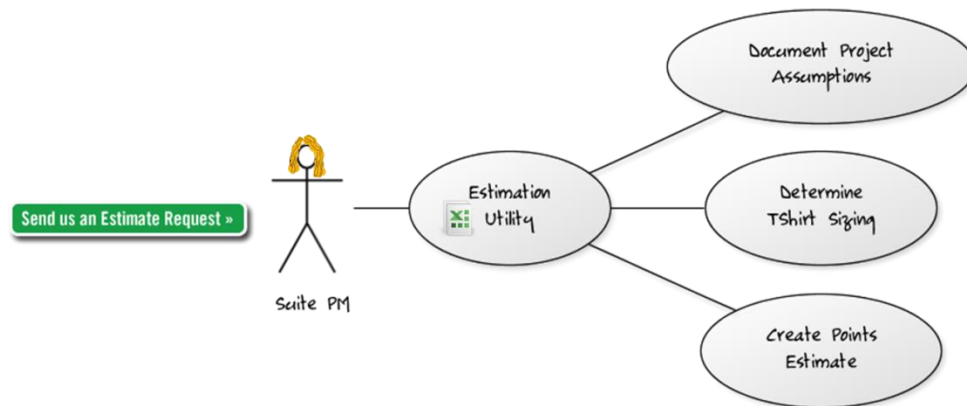


Figure 4-1 Opportunity Assessment Process

Also, instead of setting up multiple checklists for each new project or major initiative, it is necessary to provide a way to make estimates more reliable and repeatable by having a standard estimation utility that works with all types of EDW and BI projects.

Likewise, it is important that the results of the utility be used as a way to facilitate group discussions and to improve expert estimation.

Additionally, the utility packages up estimates with scope items, assumption, known constraints, and potential risks. Therefore, the estimation utility can be used as checklist that assists in preventing important aspects from being forgotten when eliciting underlying assumptions and document any possible threats to the estimates.

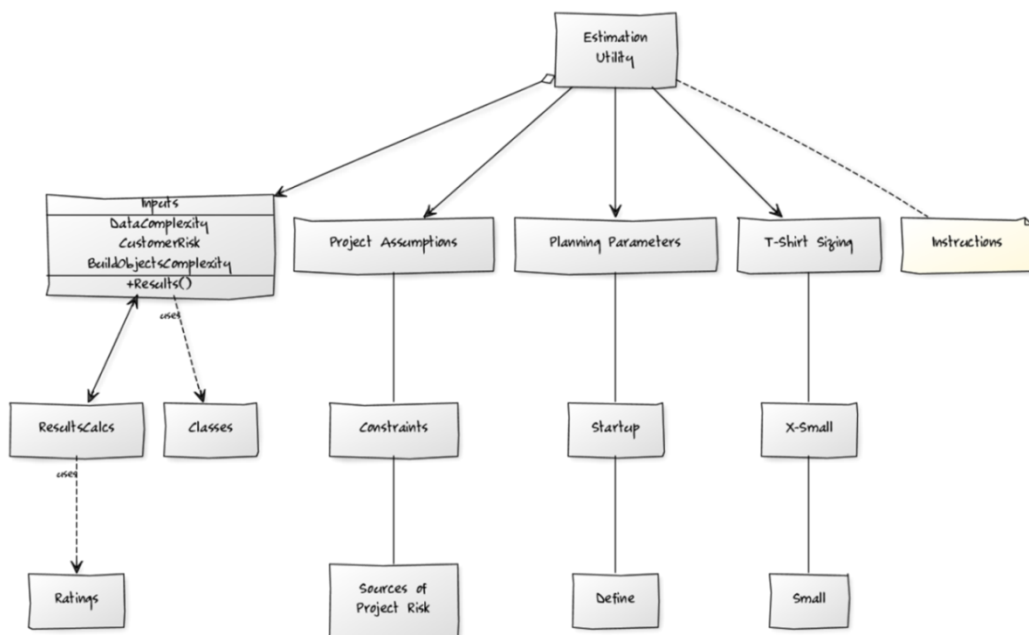


Figure 4-2 Estimation Utility Features

As far as desired features, the utility helps software engineers estimate the size of the work effort involved for data warehousing (DW) and business intelligence (BI) projects. The goal of tool is to estimate the breakdown of the total work into smaller clearly-defined build objects. After refining the project deliverables into smaller work items, the tool is used to re-estimate the total work based on the desired objects that will

need to be created. This should happen continuously as estimates are refined to deliver the most business value.

Proposed ways to use the tool:

- Estimate EDW and BI projects (see figures 3-4 and 3-5).
- Communicate Estimate Assumptions.
- Simulate project outcomes.
- Account for economies of scale.
- Account for creeping requirements and What-if analysis.
- Referee for unrealistic project expectations.
- Act as an objective authority when revising estimation assumptions..
- Provide a common estimating methodology across all EDW and BI projects.

PMs and Engineers are able to enter EDW centric project parameters. These include known data complexity, the Customer relationship risk, and the estimated number of Build Objects. Other parameters include # of Source Application Subject Areas, # of Source Application Front Ends Needing Analysis, and the estimated number of Build Objects per platform.

Please Fill out the Grid Below to get Point Estimates:	
Complexity of Data	Low
Customer Relationship Risk	Low
# of Source Application Subject Areas	0
# of Source Application Front Ends Needing Analysis	0
# of Extreme Complexity Crystal Reports	0
# of High Complexity Crystal Reports	0
# of Medium Complexity Crystal Reports	0
# of Low Complexity Crystal Reports	0
# of Extreme Complexity WEBI Templates	0
# of High Complexity WEBI Templates	0
# of Medium Complexity WEBI Templates	0
# of Low Complexity WEBI Templates	0
# of Extreme Complexity BO Universes	0
# of High Complexity BO Universes	0
# of Medium Complexity BO Universes	0
# of Low Complexity BO Universes	0
# of Extreme Complexity Essbase Cubes	0
# of High Complexity Essbase Cubes	0
# of Medium Complexity Essbase Cubes	0
# of Low Complexity Essbase Cubes	0
# of Extreme Complexity arcPlan Templates	0
# of High Complexity arcPlan Templates	0
# of Medium Complexity arcPlan Templates	0
# of Low Complexity arcPlan Templates	0
# of Extreme Complexity ETL Graphs	0
# of High Complexity ETL Graphs	0
# of Medium Complexity ETL Graphs	0
# of Low Complexity ETL Graphs	0

Figure 4-4 Estimation Tool Input parameters

Other metadata parameters should be used primarily for intermediate calculations, and are used for tweaking the weighted ratings of the build objects. Moreover, these parameters should include contingency and discount factors for predetermined economies of scale that will adjust based on the number of units entered. This should allow for uncertainties and also should allow for assuming some benefit from doing multiple items.

Duration Times Per Unit (in Points including Design and Build Phases)				
	Low Complexity	Medium Complexity	High Complexity	Extreme Complexity
Crystal Reports	1	2	3	4
WEBI Templates	0.5	1	2	3
BO Universes	1.5	2.5	4.6	6.6
Essbase Cubes	2	3.5	6	8
ETL Graphs	0.6	1.5	3	5.5
arcPlan Templates	0.75	2.5	3.8	5.5
	Low Complexity	Medium Complexity	High Complexity	Extreme Complexity
Data Complexity Multi	80%	100%	120%	150%
Customer Risk Multipl	90%	100%	110%	120%
Duration Times Per Unit (in Points For Test Phase)				
	Low Complexity	Medium Complexity	High Complexity	Extreme Complexity
Crystal Reports	0.2	0.6	1	1
WEBI Templates	0.2	0.4	1	2
BO Universes	1	2	3	4
Essbase Cubes	1	2	3	4
ETL Graphs	0.2	0.6	1	2
arcPlan Templates	0.2	0.6	1	1
	Economies of Scale Discount	*Note All Estimates Assume a 32 Hour work-week and dedicated avg. skilled resource. *Will lower overall totals assuming some benefit from doing multiple items.		
Crystal/WEBI	80%			
BO Universes	80%			
Essbase Cubes	80%			
ETL Graphs	80%			
arcPlan Templates	80%			

Figure 4-5 Estimation Tool Complexity Ratings

The results are calculated based on the numerical weights of the complexity factors derived from the metadata and based on the build object ratings. These weights are then multiplied by the number of build objects to determine the overall number of points required for each component or task ($Weights \times Inputs = Points$).

Additional data warehousing project nuances are also accounted for and used to adjust the overall level of effort (LOE) estimate. These nuances are in the form of EDW project planning ratings and weights that account for time SMEs spend analyzing source systems and requirements (see figure 4-5).

Fixed Project Variables	Rating	Variable Project Deliverables	Rating
Source Readiness Analysis	2.5	Source Analysis	0.5
Project Plan & Deliverable Checklist	0.75	Source Data Analysis	2.5
Business Cases	1.25	Requirements Package	1
Communication Plan	0.2	Current Systems Analysis (Per FE)	1.25
Stakeholders Matrix	0.2	Gap Analysis	1.25
Risk & Mitigation Plan	0.2	Conceptual Model	1.25
Software Architecture Diagrams	0.75	Extract Requirements	1.25
Infrastructure Plan	0.75	Logical Storage Data Model	0.75
Training Plan	0.75	Logical Access Data Model	0.75
Implementation Plan	0.75	Physical Storage Data Model	2.5
Master Test Plan	0.75	Physical Access Data Model	2.5
Business Change Management Plan	0.75	Functional Test	5.5
Turnover Documentation	1.25	System Test	2.75
		UAT	3.75
		Recon. And Reasonableness Test	3.75
		Performance Test	0.75

Rating per Application Topic

Figure 4-6 Estimation Tool Planning Ratings

The fixed project variables are items that account for SME fixed project time. Fixed project time describes items that have to be completed for any EDW or BI project. These items are related to documenting data warehouse artifacts and catalogs. Conversely, variable project deliverables are ratings per subject area. They account for multiple artifacts such as *Data Modeler/Architect Time* where SMEs spend a big portion of their time analyzing source systems and requirements.

Additionally, assumptions need to be captured along with high level estimates. “Assumptions are factors that, for planning purposes, are considered to be true, real or certain without proof or demonstration” (51). Estimates should be based on assumptions that the team supposes at the time of making the estimate. Assumptions are then documented as part of the estimate. “Keeping track of these assumptions formally allows the team to review them and do a simple sensitivity analysis of the estimate with respect to the assumptions” (52). It is also worth noting that to raise awareness of high probability risk items that tool instructs the SMEs to document such risks as assumptions.

As far as a sizing scale, the tool uses a common scale for teams to estimate their EI work. Instead of estimating in hours the tool uses points to quantify size. All sizing of build objects is based on an average developer's perceived high, medium, low (HML) complexity for the desired build object. Additionally, all estimates should assume a 32 hour work-week and dedicated average skilled resource. For project planning purposes, PMs can assume that 1 week equates to 1 point.

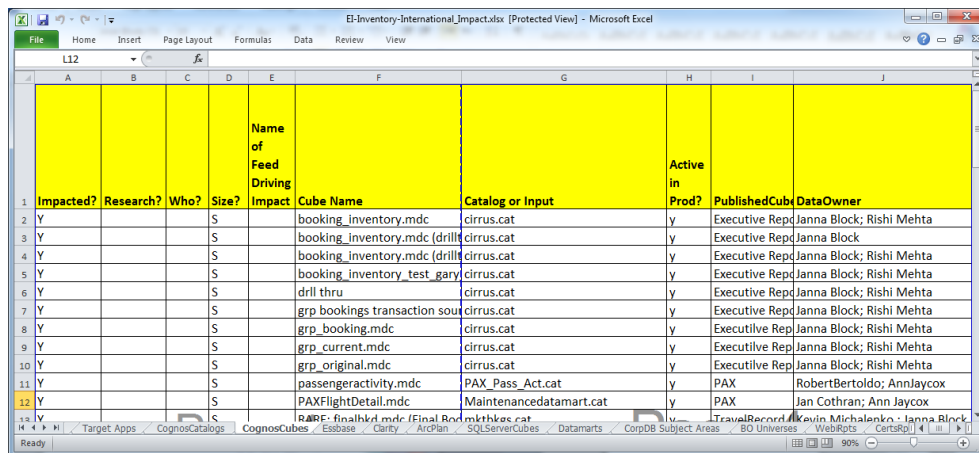
As far as risks, the SMEs are asked not only to document risks, but also document assumptions, and constraints. In this regard, the tool helps lockdown any potential risks that can jeopardize the estimate. Also, the tool instructs SMEs to document the likelihood (probability), and impact of all known risks items.

As a final point, it is worth noting that checklists help convey and remind users of some very important aspects that have to be considered in a particular project. Yet, despite their best effort, some SMEs can easily forget certain features and may either over or underestimate the total effort needed for a particular endeavor. To that end, one of the biggest advantages of the estimation utility is that it is a tool-oriented estimation utility, i.e. the project is decomposed into the actual build artifact to be delivered (Cubes, user interfaces, documentation, etc.). It is also project and process-oriented, and lists the activities necessary to build the product. This in turn improves consistency by having a standard checklist that reduces the chance of having incomplete estimates for a project. Finally, for all the reasons stated earlier, the estimation utility is particularly useful when the estimators are new or inexperienced.

Chapter 5

Experiment

Since 2011 the Enterprise Insights Team had to partake in providing estimates for major strategic initiatives. The SMEs that provided these detailed estimates would often perform a thorough impact analysis and then aggregate the estimates in a spreadsheet that contains an inventory listing of all major EDW systems. Also, the spreadsheet dives into all the artifacts that make up a particular system. Equally, the spreadsheet is organized into worksheets for each of the EI Capabilities and tools that were described in chapter 4. For example, the worksheets contain impacted ETL Graphs, SAP Data Services, Essbase Cubes, Cognos Cubes, SQLServer Cubes, DataMarts, ArcPlan Dashboards, BO Universes, Crystal Reports, and Webi Reports (see figure 5.1).



	A	B	C	D	E	F	G	H	I	J
1	Impacted?	Research?	Who?	Size?	Name of Feed Driving Impact	Cube Name	Catalog or Input	Active in Prod?	PublishedCube	DataOwner
2	Y			S		booking_inventory.mdc	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
3	Y			S		booking_inventory.mdc (drill	cirrus.cat	y	Executive Rep	Janna Block
4	Y			S		booking_inventory.mdc (drill	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
5	Y			S		booking_inventory_test_gary	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
6	Y			S		drill thru	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
7	Y			S		grp bookings transaction sou	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
8	Y			S		grp_booking.mdc	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
9	Y			S		grp_current.mdc	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
10	Y			S		grp_original.mdc	cirrus.cat	y	Executive Rep	Janna Block; Rishi Mehta
11	Y			S		passengeractivity.mdc	PAX_Pass_Act.cat	y	PAX	Robert Bertoldo; Ann Jaycox
12	Y			S		PAXFlightDetail.mdc	Maintenancedatamart.cat	y	PAX	Jan Cothran; Ann Jaycox
13	Y			S		RAIRP- finalshl mdc (Final Rod	mthbae.cat	y	TravelRepor	Kevin Mirshankin; Janna Block

Figure 5-1 EDW impact analysis spreadsheet for international

For instance, based on high level requirements, the SMEs would go through the worksheets and markup impacted systems. Using the column for impact size, the SMEs would assign a value ranging from extra small all the way to extra-large (see appendix A, for t-shirt size guidelines).

The SMEs used the approach described in figure 5.1 to communicate impact and size that would later be used to deduce LOE. The results from such estimates are

sometimes used as the biases for another strategic effort's base estimate. The figure below is a graphical representation of major strategic initiatives (elephant projects) that impacted the EDW.

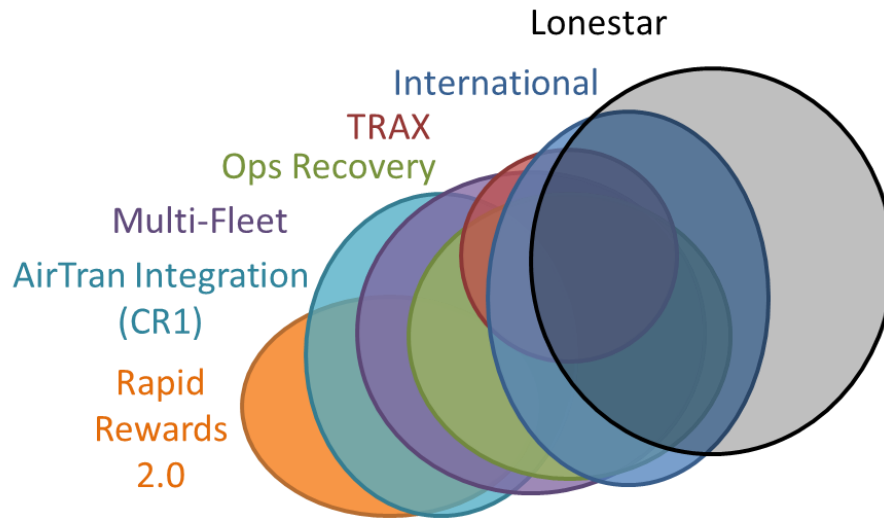


Figure 5-2 SWA strategic initiatives impacting the EDW

As an example, due to the overlap in terms of what components need to be modified from one strategic program to the next, SMEs would often use one base estimate as some multiple to drive out another estimate. For instance, the international effort that was recently completed by SWA was used as a base estimate for the reservation system replacement project (Lonestar). The SMEs estimated that Lonestar was going to be twice as big as international.

5.1 Experiment Motivation

Despite the many projects completed by the EDW Team and the great deal of effort that was put into providing estimates for each major initiative, expert judgment was the dominant approach used to provide these estimates. Consequently, an estimation utility was built to provide effective support for the skillful estimator. This section that follows demonstrates using the estimation tool by examining three estimation

experiments. These experiments explore difference between estimates conducted by SMEs and the tool using some of SWA strategic initiatives described in figure 5.2 as a backdrop. The data offers examples of varying sizes and effort estimates that were traced over numerous important development ventures. The primary goal of these experiments is to demonstrate how enterprise data warehousing (EDW) and business intelligence (BI) estimates can be greatly enhanced with the use of a supportive estimation utility. Fittingly, the goal of this section is to prove the effectiveness of using the utility with its built in checklists to ultimately achieve more accurate and transparent results.

As was described in the beginning of this chapter, the experiments explore several data points from various projects and teams. The experiment takes these reported estimates and compares them to the initial expert judgment or SME estimates, the utility estimates, and when possible, the actual effort that was registered for the particular piece under evaluation. By comparing these numbers to outputs from the estimation utility, we seek to answer the questions detailed in the section below (See Experiment Questions).

The objective is then to show that the utility can drastically reduce the subjectivity and variance of SME estimates as was described in the 2.1.1.1 Consensus Methods.

5.2 Experiment Questions

“When you lie about the future, that’s called optimism and it is considered a virtue. Technically speaking you can’t ‘lie’ about the future because no one knows what will happen. When you apply this unique brand of optimism (not lying!) at work, that’s called forecasting.”

- Scott Adams (2002).

To recap, the experiment will investigate how using the estimation utility enhances expert judgment and helps facilitate group discussions. To measure success the experiment will attempt to show that using the estimation utility will also help improve estimation accuracy in quantitative terms as well as qualitative aspects such as transparency, consistency, and estimation confidence. The specific questions that the experiment will attempt to answer:

- EI teams frequently underestimate major efforts. Will the estimation utility help decrease the optimism biases for major efforts and increase size and LOE of estimates? Will the estimation utility help improve estimates to be within 20% target range?
- Will the use of the estimation utility help facilitate group discussions by taking into account EDW nuances that often increase the size of the estimates?
- Will the estimation utility improve the accuracy of the estimates?
- Will the estimation utility improve the consistency and transparency of estimates?

The software engineers participating in the following experiments often rely on the expert judgment of knowledgeable business analysts who perform a preliminary impact assessment. As requirements get solidified overtime, both the engineers and the analysts team up for requirements elicitation, source to target mapping, and in depth impact analysis. The estimation utility participants were asked to estimate the size of all scope items in points. The results were then compiled to derive the effort and cost needed to develop certain features.

5.3 Experiments

In the section below we will examine in detail estimation aspects of the International project and the resulting estimates of impacted EDW systems and teams. Experiment I, examines the estimates that were produced by the SMEs and then

compare them to the estimation utility results (experiment I). In experiments II and III, we conduct other EI projects to see how the utility fairs when it comes to internal standalone EDW and BI projects (experiments II & III).

Moreover, the experiments below focus on 3 technologies used extensively in the EDW. Experiment I provides details of how the EDW team used the known requirements to derive estimates for the International effort. Specifically, a thorough analysis of the quantitative and qualitative results is conducted to evaluate how the estimation utility faired in evaluating the reporting components of the EDW. Similarly, in experiment II we examine the accuracy of the estimation utility with ETL graphs. Finally, we examine a large scale Essbase project to evaluate how the utility fares against Essbase cubes. All experiments compare the estimates of the SMEs, against the utility, and then a postmortem is conducted to review the results.

5.3.1 Experiment I Context

Traditionally Southwest Airlines (SWA) pursued domestic travel within the US. In 2011 SWA decided to pursue flying international as a way to serve more customers and generate more revenue. This was a commitment and a promise to the SWA stakeholder to honor enabling international travel by 2014. The target was to be able to sell international itineraries by 1/27/2014 and to be able to operate by 7/1/2014. SWA Technology was asked to provide estimates for modifying all affected systems to support International capabilities.

At the time, estimates were needed to make predictions based on imperfect information. Specifically, for the EDW department, estimates were needed to provide details around the EDW assets potentially impacted by enabling the international capability. Estimates needed to consider all of the work required to analyze, secure, configure, customize, test, deploy, and support the EDW applications.

The section below describes the high-level information that was shared with the EDW teams as a basis for sought after high-level estimates. Below is an infographic that details all the business processes that were impacted by the International effort. Additionally, actual meeting minutes that documented the information shared with the different EDW teams are provided below.

5.3.2 Early International Meeting Minutes

In the context of integration there's an aircraft conversion schedule which dictates that over the next two years SWA will be converting AirTran into the SWA operational spec (repainting them, configuring the cabin, cockpits, etc.). That is true for both their domestic and international aircrafts. Once an aircraft is converted, it then jumps over under SWA control, and then operates and sells out of SWA systems. The 1st international aircraft is planned to be converted early 2013. As soon as that first aircraft comes off the conversion line, we have to be able to support selling and operating international service in SWA systems. What that means is that we need to be able to sell and operate international by end of 2012.

The biggest obstacle at the moment is the Res system. Some consideration was given to lump the international work with the new Res system replacement. Another was to try and keep all international in Navitaire. However, it was then concluded that the easiest and best path right now is to go ahead and do the international work in SAAS. This implies that we'll need to add all the capabilities of adding the taxes, fees, and passports into the appropriate feeds and systems that consume them. This also implies modifying all the sales and service frontends (QUICK, .COM, etc), and all the Aircraft Ops systems (SWIFT, OTIS).

The business has given us the high level direction of "internationalize SAAS," and now we need to figure out the effort (how much is it going to cost, and how much

time is it going to take). Our teams need to try, in a very short time frame, to get some high level swag at those two things. We do know that it can't take longer than 2012. The consequence of not being able to make 2012 is huge. First, we risk taking down and delaying our international service. Second, we risk losing route authority with those governments. Finally, we risk not being able to get them back, because other airlines can sweep in and take them.

5.3.3 Scoping Boundaries to Figure Out the Estimates by Next Week

The premise when we go to estimate is what we would have to estimate changes to keep the lights on, not seeking any new requirements. We need to sustain existing functionality but we're not soliciting any new requirements from the business. This is a slippery slope and it would be naïve to presume that no new international reporting requirements from the business would be needed. However, we need to be mindful of the business' desire to pile on things that they wanted for years. Estimates should take that into account, but it should be reasonably aggressive and a minimal version of that.

The biggest area of impact in general is going to be around variable taxes and fees. Specifically, the number of taxes, what their rates are, and how they are calculated is a concern. These all vary by country. With this effort, the goal is not to achieve fly to any country around the globe within a few days' notice, but we are trying to achieve flexibility beyond AirTran's current set of destinations. AirTran currently flies to Puerto Rico (taxed like international), Mexico, and the Caribbean. A good rule of thumb is to assume Mexico, Central America, and the Caribbean. Flying to Canada is still TBD due to language and currency requirements. Also, we're assuming USD currency and English only. Please note that all Caribbean countries generally tax the same (according to a Saber studies).

Taxes and fees impact feeds that EI ingests (see graph below). The other area of impact is network related. Overnight flights, overnight connects, double connects. Another is the Pax activity feed. Another area is the 'Lufthansa related' areas. Last area is the third party and ground handling aspects of the business.

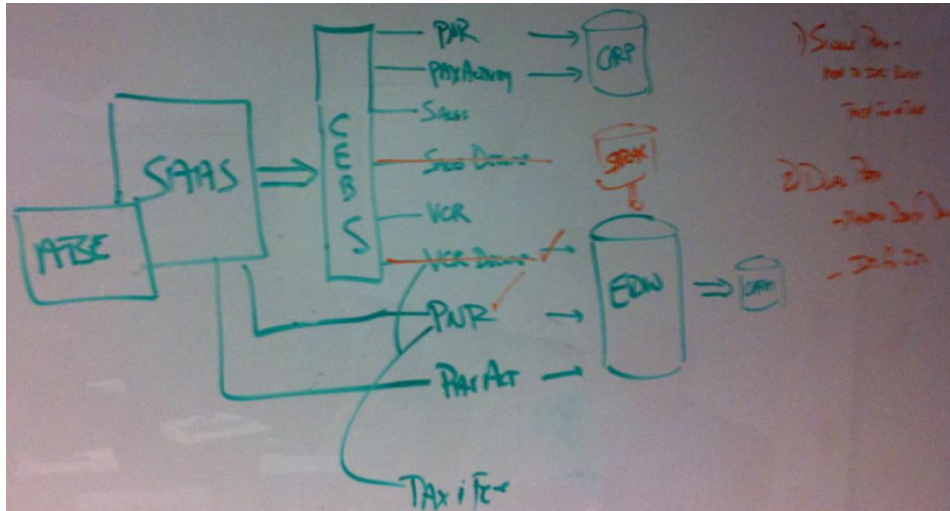


Figure 5-3 High-level descriptions of impacted EDW systems that need to be made to the reservation system to support international

5.3.4 High-level Approach Assumption:

The Software Architect assigned to this area discussed the two possibilities below:

- I. Single Path: Move to an international rule set (instead of one way for domestic and another for international). Rollup the taxes and fees to the ticket level (instead of segment level). Frontends that go through CEBS will have to change. Purpose of VCR Decomp feed is to break down the taxes and fees. Same goes for Sales Decomp. Also note that sales feed goes to SIRAX and we get our sales from SIRAX. The fare still needs to be at maintained at the OND level. Business (Marketing, Ground ops)

concerns about front end displays. Concern is selling this to the Business (Marketing, Ground ops), and concerns about front end displays.

- II. Dual Path: Maintain Domestic for Domestic and International for International.

All agreed that Single Path would be preferred method, but not feasible at the moment due to the following reasons:

Purpose of VCR Decomp feed depicted is to break down the taxes and fees. Same goes for Sales Decomp. Also, note that sales feed goes to SIRAX and we get our sales from SIRAX. The fare still needs to be maintained at the O&D level. EDW gets a direct feed from SAAS for PNR and PAX Activity. The Near Real time PNR and PAX Activity from CEBS is not enough. CEBS doesn't parse out everything we need. Group remarks, check in remarks, and boarding activity are examples.

The complex PNR parser rules would need to move into CEBS. This is huge. The estimate for the PAX alone (which was much simpler) was a significant effort. CR1 numbers for PAX was 1 POD working for 2 month and 1500 hours for EI.

It was therefore concluded, that we stay on the PNR and PAX, ACT feeds from SAAS, and modify them to deal with international. This implies modifying the PNR parsers to handle international taxes and fees, and rolling up taxes at the ticket level. It was concluded that the shorter path and that we should estimate what we're on despite the fact that when we go to a Res system, all of the above will need to change.

5.3.5 Summary of Problem Statement

In the first experiment, a deep dive was conducted on the International project. The International initiative outlined scope as 2 sets of work packages, WP101-119 (Sell) and WP201-214 (Operate). While the general solution strategizing is conducted using different stakeholders and strategic boards, architectural due diligence for complex

solutions is conducted under the auspices of architects and they engage the affected Product teams.

Below is an example of the International requirements refinement that outlined in terms of work package specifications for the international Release 2 (Operate) portion of the project:

- WP201 – Checkin, Bags & Denied Boarding.
- WP205a – Data Feeds and Reporting – Operate (Book & Ticket).
- WP205b – Data Feeds and Reporting – Operate (Required).
- WP205c - Data Feeds and Reporting – Operate (Optional).
- WP206 – Fleet Management.
- WP211 – Crew Scheduling.
- WP214a – AO Safety Systems (pre-operate).
- WP214b – AO Safety Systems (post-operate).
- WP215 – International Security – Crew.
- WP216 – Refunds – Operate.

Below is a summary example for epics and features for WP215:

- WP215.Epic-1: Create and maintain a master crew list, and send it to the government when necessary.
 - Feature-1: Create an initial master crew list with eligible employees and transmit to DHS.
 - [Stories ...]
 - Feature-2: Add new eligible employees and transmit to DHS and Crew Scheduling.
 - [Stories ...]

Note that factoring out common sub functionality is “design” and is done at the next, feature-story level of decomposition.

The above classifications and work package definitions are used to derive the initial high-level estimates across multiple portfolios. Teams attend work package definition workshops and are then able to clarify assumptions and agree on features to be delivered. After the feature definitions get flushed out, affected teams are then asked to provide their detailed estimates. Specifically, the teams are then expected to come back to the program team leading the International effort with size estimates. The program team then uses the size estimates to derive cost and LOE estimates. The scope defined by the work packages above affected all areas of the EDW.

5.3.4 Experiment I Setup

In the first experiment, the EDW team was asked to provide detailed estimates to the program team. In particular, the work packages that the program team needed the EDW team to estimate was “WP116 – Data Feeds and Reporting – Sell” and “WP205 – Data Feeds and Reporting – Operate.” The bulk of this work was going to be completed by the four Enterprise Insights team described below. The data below is summary of what each team estimated for the work that their respective teams needed to complete.

Estimation Input Parameters			Results	
Team	Work Package #	Work Package	SME LOE	Utility LOE
Team 1	205	Data Feeds and Reporting - Operate	3660	11346
Team 2	116	Data Feeds and Reporting - Sell (Book & Ticket)	8640	24711
Team 3	205	Data Feeds and Reporting - Operate	5745	21257
Team 4	205	Data Feeds and Reporting - Operate	1670	3895
Team 5	116	Data Feeds and Reporting - Sell (Book & Ticket)	1590	5952
Total			21305	67161
Adjustments (Confidence Level)			27910	68000
Actual LOE			74169	

Figure 5-4 Experiment I – WP205 & WP 116 EI teams SMEs, Utility, and Actuals

The Enterprise Insights – Enterprise Management (EI – EM) team, for WP205 estimated that it would take the team about 3660 hours to complete the work. On the other hand, the estimation utility (for the same work) quote was 11349 hours (almost 3 times more than the SME estimate).

The Enterprise Insights – Commercial Experience (EI – CE) team, for WP116 estimated that it would take the team 8640 hours to complete the work. The utility estimate was 24711 hours (almost 2.86 times more than the SME estimate).

Enterprise Insights – Aircraft Operations (EI – AO) team, for WP205 estimated that it would take them 5745 hours to complete the work. The utility estimate was 21257 hours (almost 3.7 times more than the SME estimate).

Enterprise Insights – STARS team’s subject matter experts (SMEs) were asked to estimate the level of effort (LOE) of enabling all reports used by SWA internal customers to support the international sell operation. The requirements were defined in the “Data Feeds and Reporting – Sell” work package (WP116) “Data Feeds and Reporting – Operate” work package (WP205). The EI – STARS team estimated that it would take the team 1670 hours to complete the work for WP205. Similarly for WP116 the estimate was 1590 hours. On the other hand, the utility estimate was 3895 hours for WP205 and 5952 hours for WP116 (almost 2.33 to 3.74 times more than the SME estimates). In the next section we take a closer look at the EI – STARS team estimation for WP116.

5.3.5 Results

Based on WP116’s scope, 344 reports were identified as needing to be modified to support the international operation. This impact spanned across many business areas and affected multiple internal SWA application teams. Additionally, the types of reports

that were impacted included operational reporting, analytical reporting, and ad-hoc reporting as discussed earlier in Chapter 3 (see EI Styles of BI).

The table below is subset of the estimates that were provided from group 1. In particular, this table compares the estimates that were produced by the SMEs, the utility, and then the actuals (please see Appendix B, Figure B.1 for the full data set).

Table 5-1 Deep Dive into experiment 1's estimates from group 1

Applications Areas	# Reports	Experts	Utility	Actuals
SPN	27	117	416	425
CS2	31	134	576	592
QIK	2	10	32	34
RefundPro	22	96	384	399
CFM	44	190	1152	1249
Station Reporting	34	147	480	413
BRUTIS	71	306	960	884
OQS	20	86	288	294
SPT	9	41	128	130
LMS	15	66	224	233
SOP1	47	203	704	726
MOM	13	58	192	202
FOAR	4	18	64	67
Taxes	5	23	80	88
Airport App. Suite	14	61	224	241
Flight Ops	3	15	32	33
EDW	?	19	16	54
Total	344	1590	5952	6064

Based on the SMEs' analysis that was performed in Figure A.1 (Appendix A), the SMEs concluded that it would take 1590 hours to complete the work defined in WP116. Similarly, the utility was then used to compute the LOE for the different types of reports described above. Utilizing the same analysis that was performed by the SMEs in Figure A.1, and plugging the information into the utility, the utility's LOE estimate was 5952 hours.

The matrix plot below is used to visualize the relationship between each pair of estimates. Comparing each pair of scatter plots reveals that the relationship between

utility and actuals exhibits a tight fit. On the other hand, the relationship between SMEs estimate and actual results appear to have higher variance.

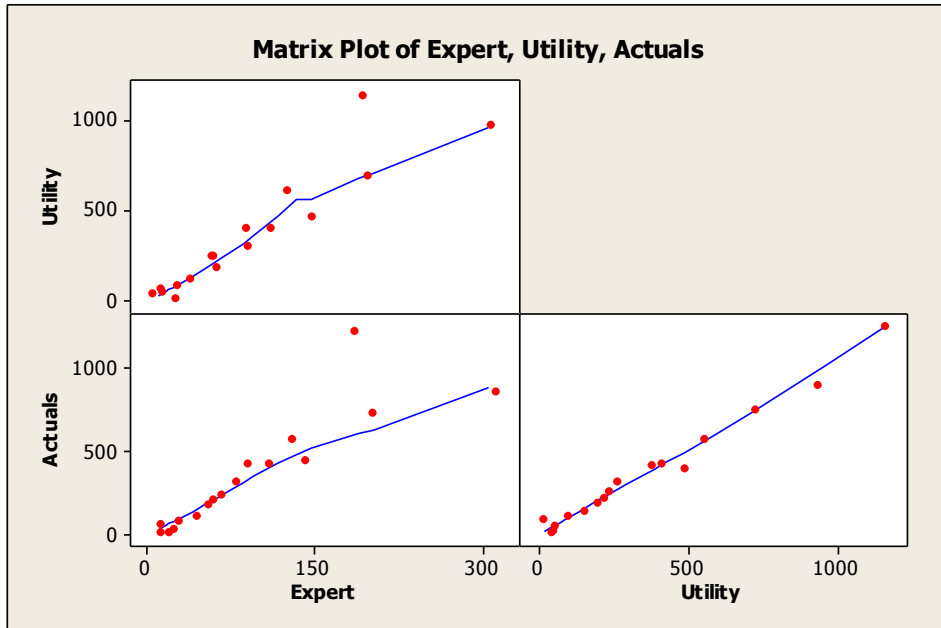


Figure 5-5 Experiment I – Matrix of scatter plots for SMEs, Utility, and Actuals

From examining the data and the totals, it is clear that the SMEs underestimated the LOE required to complete WP116. The actual LOE was larger than the estimated value by about 381%. The utility was closer to the actual and was off by 1.88%. In the section below we use 4 statistical techniques to derive an overall measure of reliability.

In the first technique, we summarize data from the multiple estimates and then display the results in boxplots (figure 5.5). From examining the summary data and descriptive statistics information in Appendix A, the mean values for the estimates above are as follows. The mean for the SMEs is 93.53, the Utility is 350.12, and the actuals are 356.71. In addition, the median values are 66.0 for the SMEs, 224.0 for the utility, and 241.0 for the actuals. These values are depicted on the boxplots below with connect

lines. From examining the data and the connect lines, the values show a closer correlation between the actuals and the utility than that of the SMEs.

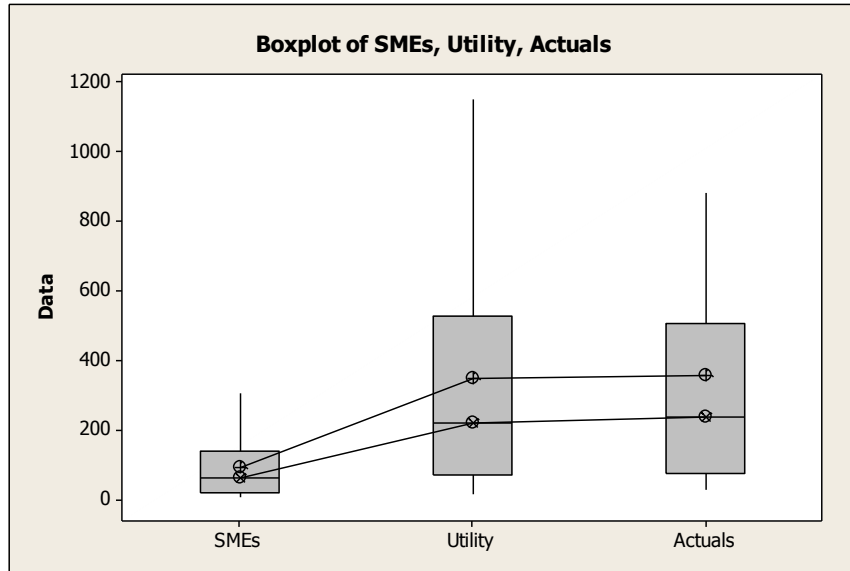


Figure 5-6 Experiment I Boxplot – SMEs, Utility, and Actuals

In the first test, a One-Sample T test was chosen to compare values for the SMEs, utility, and the actuals. The T and P values were then computed using the actuals mean with a test of hypothesized mean (μ) equal to 356.7. Please note that all the results in this research paper were computed using Minitab® 16.2.0. Results displayed in table 5.2.

Table 5-2 One-Sample T test for the SMEs, utility, and the actuals using a test of mu value equal to 356.7 vs. not equal to 356.7

Variable	N	Mean	StDev	SE Mean	95% CI	T	P
SMEs	17	93.5	81.7	19.8	(51.5, 135.5)	-13.28	0.000
Utility	17	350.1	334.7	81.2	(178.0, 522.2)	-0.08	0.936
Actuals	17	356.7	339.1	82.3	(182.3, 531.1)	-69.39	1.000

The results from table 5.2 clearly show that the T values between the utility and the actuals is 1, whereas the SMEs vs. the actual was much larger (231.97). Additionally, the two-tailed P value for the SMEs vs. the actuals was less than 0.0001. By conventional

criteria, this difference is considered to be extremely statistically significant. This implies that the results of the SMEs are vastly different from the actuals.

Conversely, the P value and the statistical significance test for the utility vs. the actuals, reveals that the two-tailed P value is equal to 0.9362. In this case, the difference is not considered to be statistically significant. This in turn implies that the results of the utility are more closely aligned to the results of the actuals than those of the experts.

In the next test we compute the Intra-Class Correlation Coefficient (ICC) to examine the reliability and validity of the data. The resulting Pearson correlation values are displayed below.

Table 5-3 Pearson correlation displayed for SMEs, Utility, and Actuals

Correlation Matrix	SMEs	Utility
Utility	0.928	
Actuals	0.888	0.994

All of the Pearson correlation values are high, which indicates that values are closely correlated. The lowest Pearson correlation value is between the actuals and SMEs (0.888). The highest is between the actuals and the utility (0.994). In this context, the ICC of 0.994 (Actual, Utility) indicates that the estimates from the utility and those of the actuals have a very high level of agreement.

The Minitab® 16.2.0 tool also produced a Cronbach's Alpha value coefficient. This coefficient measures how closely related a set of items are as a group and is considered to be a measure of scale reliability (53). The calculated alpha coefficient for all comparison items is 0.8732. This number suggests that the estimates in the table above have relatively high internal consistency.

5.4 Experiment II

Southwest Airlines undertook an initiative to upgrade their Ab Initio environment from its current version to version 3.x along with upgrading the operation platform from Solaris to a Linux VM environment.

An external vendor was engaged to deliver the upgrade to the Ab Initio environment from its current version to version 3.x. The primary scope of the engagement included the following tasks:

- a. Identifying active ETL (Extract, Transform, and Load) Objects.
- b. Migrating Sandboxes for each application.
- c. Upgrading identified Ab Initio Objects to v3.x.
- d. Identifying Recovery Plan.
- e. Standardization of Code.
- f. Identifying Optimization opportunities and documenting.

5.4.1 Engagement Model

The vendor decided to deploy the following team for duration of 10 months to deliver the service required for this engagement.

Table 5-4 Engagement Model – Proposed FTEs (Full-time equivalent)

Resource Type	Number of Resources	Work Location
Project Manager	1	On-site
Team Lead	1	On-site
Senior Developer	2	On-site/Off-site
Developer	1	Off-site
Test Analyst	1	Off-site
Technical Writer	1	Off-site
Total FTE	7	

This engagement was a “Fixed Bid” contractual agreement. The agreement specified that the vendor was allowed to work this project on-site and offshore to complete the work. The major duties, responsibilities, and experience levels for each of

the roles defined in the engagement model above were defined in the service agreement. The figure below is derived from the initial personnel projection matrix (see appendix A Table A.1 Personnel Projection Matrix) as it was communicated in the statement of work (SOW). The SOW provided a Personnel Projection Matrix for a total duration of 10 months starting from Jun 2013 all the way into March 2014. The histogram graph below demonstrates the initial planned utilization of vendor and SWA resources described in Table 5.4.

Moreover, the graph shows the proposed plan of how these resources will be scheduled to work over a predetermined time period described in the SOW. Specifically, the proposal was to have 5 FTEs work the 1st month and allowed for 7 FTEs for the remaining 6 month. As a contingency, four month slack was built into the schedule. For more details, please see Appendix A Table A.1. The Personnel Projection Matrix table in Appendix A was furnished from the vendor’s SOW, and it outlines the complete details of the proposed engagement staff projections.

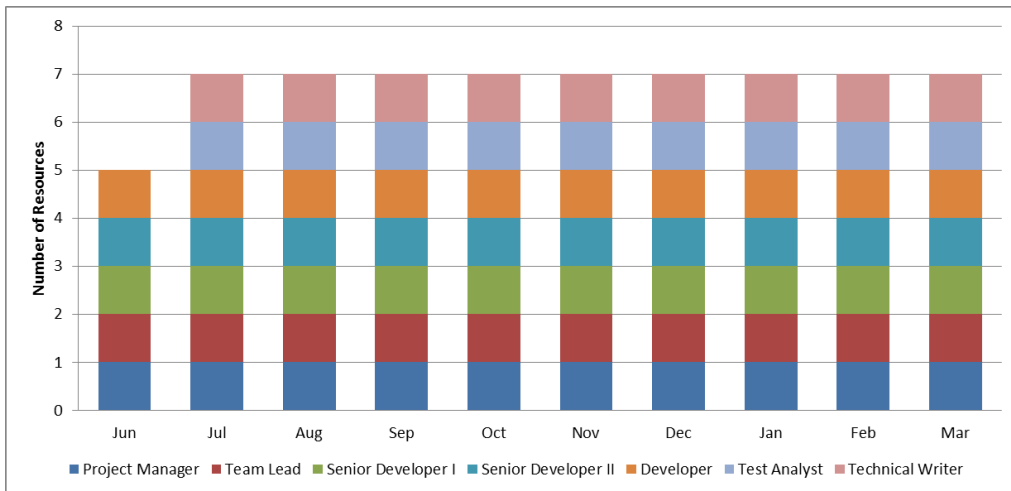


Figure 5-7 Personnel Projection Matrix – employment projections details

5.4.2 Solution Approach

This experiment focuses on the ETL conversion. Specifically, there are about 105 applications that consist of 1300 ETL graphs, 150 scripts, and 260 Tivoli job schedules that need to be converted from the existing Solaris environment to a new Linux environment, and to be upgraded to run on Ab Initio 3.1. The intention was not to just simply convert, test, and migrate to new production environment but to also stabilize the new environment by upgrading all the conversion artifacts newly defined coding standards (naming convention, etc.), and updating support documentation.

Consequently, the estimation for this initiative needed to capture the following in scope requirements for this conversion project:

- Convert each graph / script in the application so it:
 - Works in Linux / AbInitio 3.1.
 - Is renamed to its name based on its target subject area and application.
 - Uses the new run control system based on .cfg files.
 - Runs with the new etl-wrapper.
- Convert each related Tivoli Job / Tivoli Schedule as follows:
 - Rename the Tivoli schedule names to match the Tivoli Naming Convention.
 - Rename the Tivoli job names to match the Tivoli Naming Convention.
 - Identify and update all references to the new schedules.
 - Generate New Tivoli Migration Artifacts.
 - Identify the migration requirements:
 - Tivoli requirements / dependencies.
 - Run control requirements / dependencies.
 - File security permission requirements / dependencies.

- Apply select ETL coding standards.
- Update Documents.
- Master Job Flows: As each application is migrated, the job flows document will need to be also updated because the jobs are being renamed. Dependencies should not change.
- Tivoli Job Docs: The job docs used to document Tivoli jobs will need to be updated or risk obsolescence if we change the way we deploy and document jobs.

5.4.3 Experiment II Setup

The early initial assumptions for the estimate projected that about 5000+ ETL graphs needed to be converted. The original expert judgment estimate, assumed that the work could be completed based on a fixed rate of 4 graphs per day per developer (2 hours per graph), for a total duration of 6 month. The proposal was to have 2 developers complete the work within 6 month.

The final expert judgment estimate predicted about 1500+ graphs with the same rate of 4 graphs per day per developer, but additionally added 1 tester, 1 technical writer, for a total duration of 6 month as well. The breakdown of the impacted EI subject area and graphs is listed below (see figure 5.1).

Subject Area	Number of Apps	Number of Graphs
Airport	9	262
Bookings	15	231
Crew	6	209
Customer	8	127
Executive	5	76
Financial	20	121
Industry	3	79
Maintenance	8	83
Marketing	2	11
People	5	97
Schedules	4	70
Secure	4	0
Pax_DSS	5	0
Utility	5	14
Loyalty	6	23
Total	105	1403

Figure 5-8 List of impacted subject area and graphs that need to be migrated

A vendor that provided a competitive fixed-bid based on the estimate above was selected to do the work. Again, based on the expert's estimate, a project plan was created that stipulated that the firm will complete the work in 6 month with 2 developers being responsible for the bulk of the conversion.

Afterwards, and against the backdrop of the information provided by the two expert estimates above, the estimation utility was used to contrast the size and the duration estimates that were provided by the experts.

Finally, when the project completed, all the estimates that were produced by both the utility and the experts were all measured up and compared to the actual work and cost of the project. For example, the actual number of graphs that needed to be converted ended up being 1728 graphs. The vendor that completed the conversion work maintained that the graphs were converted at a rate of 4 graphs per day per developer.

However, the same firm acknowledged that they had to add two additional developers, 4 testers, 1 tech writer, for a total of nine people. The work was completed in

a little over 10 months. Additionally SWA SMEs had some work that was left over that the SMEs had to complete to close out the project.

5.4.4 Results

The figure below shows the results of the two rounds. Each round shows the number of graphs estimated, the initial group/expert estimates, and finally (while using the same input parameters), the results from the utility are shown as well.

In the first round the number of graphs that were estimated was 1403. Applying the micro estimate of 2 hours per graph, the group derived a macro estimate for the remaining graphs, and concluded that it would take about 2760 hours to complete the work (see figure 5.2 below for more details).

The estimation utility was then used to produce a size estimate. The utility produced a range of sizes estimates that was between 170 to 211 points. Due to the uncertainty regarding the actual number of graphs, and due to the skill level of the resources, the upper bound of the range of the size estimate was selected. Based on the utility assumptions described in chapter 4 (1point \approx 1week), the selected size estimate is equivalent to about 211 people weeks. To use a common scale to compare to what was produced by the experts; the equivalent hours estimate was calculated to be about 6752 hours.

Source Data		Round I					Round II				
Subject Area	Number of Apps	Number of Graphs	Complexity	Group: Initial Estimate (hours)	Utility: Initial Estimate (points)	Utility: Initial Estimate (hours)	Number of Graphs	Complexity	Group: Final Estimate (hours)	Utility: Final Estimate (points)	Utility: Final Estimate (hours)
Airport	9	262	Extrem	524	46	1472	197	Medium	394	34	1088
Bookings	15	231	Extrem	462	41	1312	274	Extrem	548	31	992
Crew	6	209	Extrem	418	16	512	193	Medium	386	13	416
Customer	8	127	High	254	14	448	218	Extrem	436	14	448
Executive	5	76	High	152	9	288	70	Medium	140	7	224
Financial	20	121	High	242	36	1152	212	Extrem	424	29	928
Industry	3	79	High	158	5	160	92	Medium	184	6	192
Maintenance	8	83	High	166	14	448	113	Medium	226	11	352
Marketing	2	11	Low	22	1	32	31	Low	62	3	96
People	5	97	High	194	9	288	129	High	258	9	288
Schedules	4	70	Medium	140	7	224	77	High	154	6	192
Secure	4	4	Low	8	3	96	58	Low	116	5	160
Pax_DSS	5	5	Low	10	3	96	19	Low	38	2	64
Utility	5	14	Low	28	3	96	34	Low	68	4	128
Loyalty	6	23	Low	46	4	128	11	Low	22	3	96
Total	105	1403	=High	2824	211	6752	1728	=Medium	3456	177	5664

Figure 5-9 List of impacted subject area and graphs that need to be migrated

In round II, the actual number of graphs that needed to be converted was found; 1728 ETL graphs. Similar to the strategy above, the experts applied the same estimate of 2 hours per graph. The group then derived an estimate for the remaining graphs and concluded that it would take about 3456 hours to complete the work (see figure 5.2 above for more details).

On the other hand, the estimation utility produced a size estimate range of 113 to 177 points. It is important to note that even though more graphs were added, the size estimate that was produced by the utility for this round was less (177 points vs. 211 points). This is due to the complex graphs were being dropped and replaced with less complex graphs. This time, due to the large discrepancy between the number that was produced by the experts and the number that was produced by the utility, the upper bound of the range of the size estimate was selected. This size estimate is equivalent to about 177 people weeks. To use a common scale to compare to what was produced by the experts; the equivalent hours estimate was calculated to be about 5664 hours.

5.4.5 Results Analysis Postmortem

The total effort for this project was 53 people month or approximately 220 people weeks. The work started with 2 average skilled developers working with some SWA

experts to convert a few graphs together. As work progressed into the second month, the vendor realized that not all graphs were created equal. For some, the conversion process was straightforward, while other graphs were significantly harder to convert (see appendix C).

This worked against the original assumption that it would take 2 hours per graph. As a result, and well into the third month of the project, the vendor decided to add testers and developers to help make up for lost time. The number of developers increased from 1 to 6, and the sum of testers rose from 1 to 4. The addition of testers and developers caused the FTE counts to increase from 7 to 15. See Table below for details.

Table 5-5 Results Analysis Postmortem – Actual FTEs (Full-time equivalent)

Resource Type	Initial Estimated Resources	Actual Number of Resources	Location
Project Manager	1	1	On-site
Team Lead	1	1	On-site
Senior Developer	2	2	On-site/Off-site
Developer	1	6	Off-site
Test Analyst	1	4	Off-site
Technical Writer	1	1	Off-site
Total FTE	7	15	

Another resource factor contributing to the overall timeline was the availability of the resources. In the beginning of the project, based on the Personnel Projection Matrix (please refer to Figure 5.1), 8320 hours of capacity was assumed to be available to complete the work in the statement work. The visual representation below shows the planned work (bar chart) and the assumed available capacity (the step chart above the bar chart). It is worth noting that due to the uncertainty of the group estimates; slack was built into the schedule. Of the 8320 hours, 3456 hours were slated for the developers to complete the bulk of the graph conversion work. The figure below describes what was known at the time of the SOW for the planned capacity vs. the assumed available time.

The Project Manager (PM), and Team Lead were assumed to have a fixed contribution throughout the project. The Business Analysts (BAs) included the testing resource, and the technical writer. The developers were also assumed to have a fixed contribution thought the project.

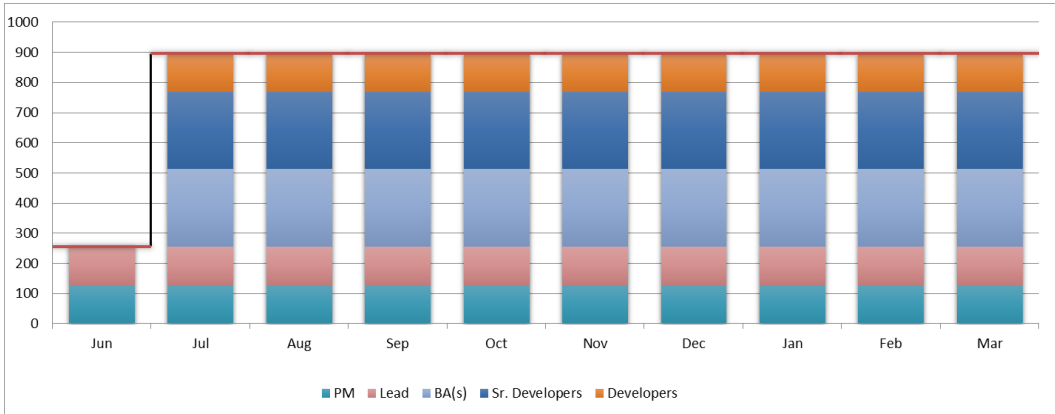


Figure 5-10 Project Resources Capacity Step Charts *Planned* resource allocation and capacity

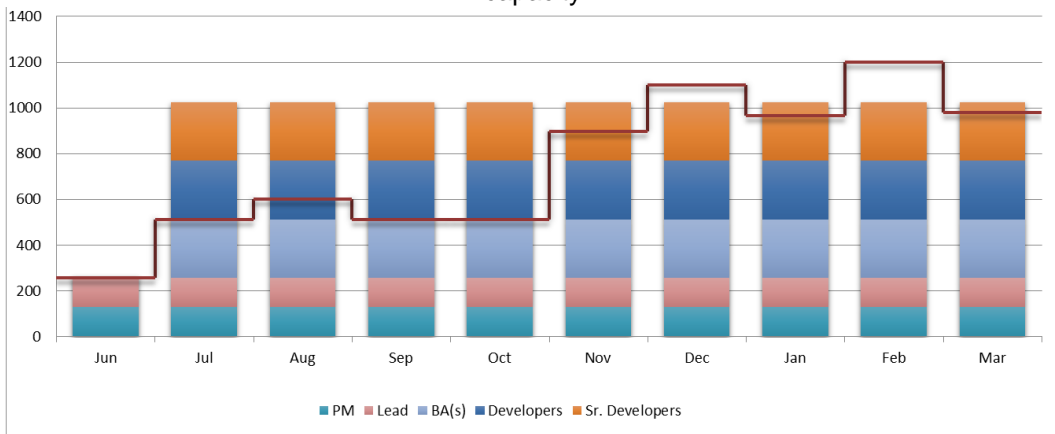


Figure 5-11 Planned resource allocation against actual capacity

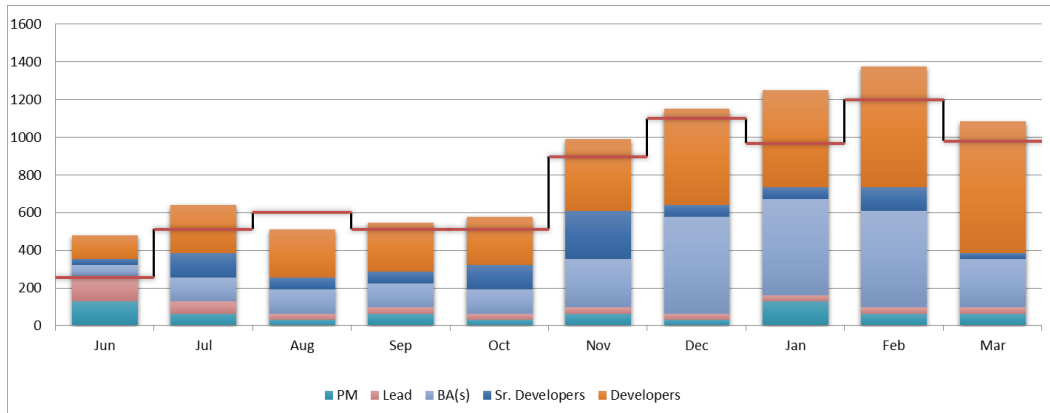


Figure 5-12 Actual resource allocation and capacity

5.4.6 Data Analysis

In figure 5.2 a list of graphs that needed to be converted was provided in the form of pairs. These pairs of estimates consisted of values that were assessed by experts and their equivalent values that were estimated by the estimation utility. It is interesting to note that each expert estimate value has a natural partner element in the form of an estimated utility value. Due to the pairings and nature of this information, the Paired t-test was selected to examine the difference between the pair of values in the two estimation rounds.

Specifically, in this context, the goal of the test is to examine the variation of values within each round, and then produce a single t-value number for each round. These computed t-values are derived from comparing the values “before” using the estimation utility and the values “after” using the estimation utility. In the end, the t-values will in turn determine whether a significant change or noticeable improvement occurred from using the utility (54).

In the next section, a statistical analysis is conducted for both rounds. The analysis compares the paired samples of ‘before’ and ‘after’ using the utility. The emphasis is the data set from two rounds that were described in figure 5.2.

5.4.7 Statistical Analysis

For each pair of estimates representing the subject areas above, a paired T-test analysis was performed. The estimate values “before using the utility” and “after using the utility” are reported in pairs. The paired T-test was used to demonstrate any potential estimation improvement due to using the utility.

For the first estimation round, the two-tailed P value equals 0.0111 with T-Value = -2.99. By conventional criteria, this difference is considered to be statistically significant. Similarly, for the second round the P value equals 0.0190 and T-Value = -2.65 which is below the .05 standard, so the result is also statistically significant. Since we have 15 pairs of estimates, this experiment has 14 degrees of freedom. Given that the test is trying to determine if the utility only improves (not reduces) the estimates, we use the one-sided alternative hypothesis t-values for the p-value in the area to the right of T-Values above.

Next, a matrix of scatter plot is used to visualize the relationship between like estimates. Inspecting the graphs below reveals that the utility estimates exhibit a tighter fit with much higher internal consistency.

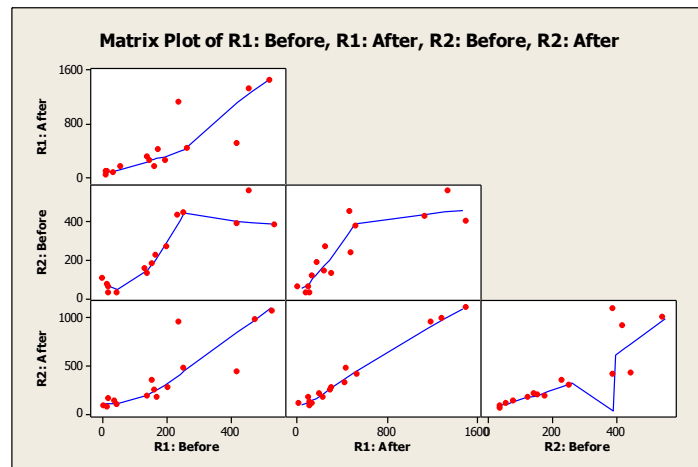


Figure 5-13 Experiment II – Matrix of scatter plots for SMEs, Utility

Following that, we compute the Intra-Class Correlation Coefficient (ICC) to examine the reliability and validity of the data. The resulting Pearson correlation values are displayed below. Please see Appendix A, Experiment II for intermediate statics calculations and corresponding r-squared values.

Table 5-6 Pearson correlation displayed for the two estimation rounds

Correlation Matrix	R1: Before	R1: After	R2: Before
R1: After	0.864		
R2: Before	0.886	0.836	
R2: After	0.860	0.994	0.868

All of the Pearson correlation values are relatively high. In this experiment we use the ICC test to compare likes with likes (before from round I against the before values from round II and similarly for the after values). The test reveals that the highest correlation is found in the after using the utility values (0.994). This indicates that the estimates produced by the SMEs and the utility all have very high level of agreement, but the highest and most consistent are the ones produced by the utility.

Taking these Pearson correlation values into account and matching them with their corresponding scatter plots above, reveals that the higher values also have the smoother and tighter fits. This smoothness and tightness is represented by the data on the scatter plot and is illustrated by a line of best fit (or "trend" line) that is almost a straight line.

The tests above provided evidence that the utility caused the estimates to be more accurate than merely relying on the experts. From examining the data, the amount of actual increase was also significant.

5.5 Experiment III

Extended Spread Sheet dataBASE (Essbase) is a “multidimensional database management system (MDBMS) that provides a multidimensional database platform upon which to build analytic applications.” Essbase is the chief OLAP technology that is used by SWA.

As a reminder, internal SWA Business customers use Essbase cubes’ capabilities to be able to slice and dice data to grasp dimensionality of data for certain business domains. For example, SWA ground operations could use the data to provide calculations and metrics related to SWA maintenance bases over multiple measures. A specific type of measurement might be the number of planned aircraft checks, and total hours spent on each aircraft (see section 3.1.1.5.4 OLAP Analysis for more details). The type of analysis described is one of key competencies for SWA BI capabilities.

SWA objective was to upgrade their current Essbase environment from the antiquated v7 to the latest v11. This Essbase modernization project was a full platform upgrade and the work was to be completed by Professional Services and internal SWA resources. A Professional Services vendor that was selected to install Hyperion Essbase V11.1.2.2 and all its required components to operate on the new Linux 64 bit system. Additionally, the vendor was asked to migrate all of the v7.1.5 Essbase Cubes into the new environment. The Cube migration had a hard completion deadline of Oct 2014 for two main reasons.

First motive was due to contractual agreements, resulting in version 7.1.5 not being supported by Oracle. Second incentive, was due to critical financial information contained within certain Cubes. These cubes needed to complete by the deadline above so that they can be used to report to Wall Street and SWA Shareholders the company’s

financial performance. To summarize, the primary scope of the engagement included the following phases:

- Phase I: Assessment of Essbase environment, baseline cubes, and creating high level roadmaps for migration and implementing Essbase V11.1.2.2.
- Phase II: Migration of objects, testing, knowledge transfer, and support for new Essbase environment.

5.5.1 Engagement Model

During the early phases of the project, the vendor proposed that 9 resources will be required to complete the project successfully.

Name/Role	Dates	Total Hrs
Hyperion Project Manager	Jan 6 – Jun 6	981
Hyperion Essbase Architect	Jan 6 – Jun 20	1071
Hyperion Lead / PM	Jan 6 – Jun 24	1089
Linux Developer	Apr 1 – May 30	387
Essbase Developer	Jan 27 – Jun 24	954
Essbase Developer	Jan 27 – Jun 24	954
Essbase Developer	Feb 3 – Jun 17	864
Essbase Tester	Feb 3 – Jun 17	864
Essbase Tester	Feb 3 – May 14	657

Figure 5-14 Engagement Model – Proposed FTEs

5.5.2 Solution Approach

Similar to the previous experiment, this engagement was a “Fixed Bid” contractual agreement. The proposed duration was 6 months to deliver the project outlined in the two phases above. At the time of the SOW, it was estimated that the vendor would need to convert 59 cubes. Taking the deadline date of October 31st into consideration, the SOW stated that work should start in January 2014. The early start date allowed for a four month contingency plan. In turn, the plan allowed for the infrastructure piece to be put in place first. Consecutively, this permitted the developers to use a phased approach for the cube conversions. This phased approach was necessary

to work around customer schedules and not to impact on going business processes. The figure below describes the development freeze calendar developed to take into account the business constraints imposed on this project.

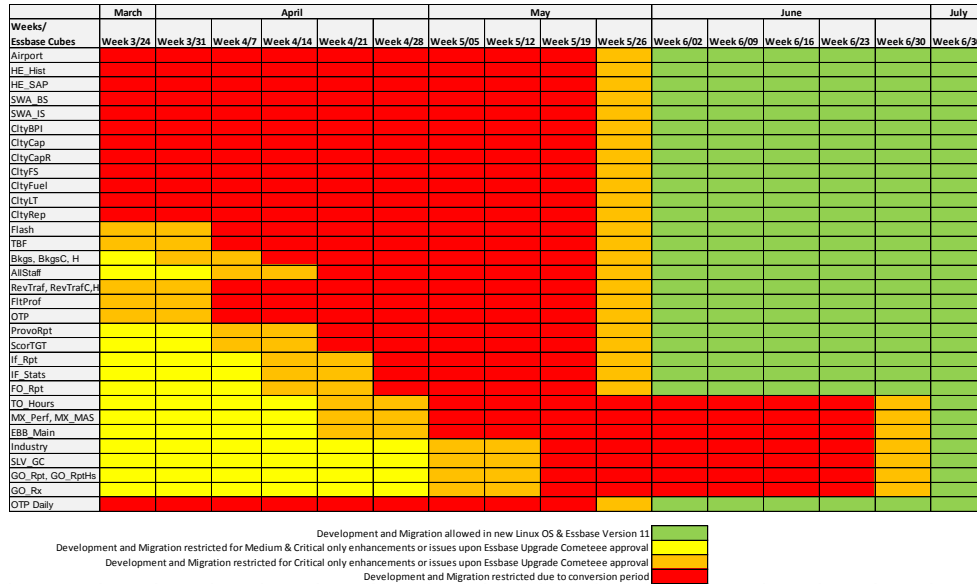


Figure 5-15 Essbase Upgrade 2014 - Development Freeze Calendar

5.5.3 Results Analysis Postmortem

Similar to previous experiments, the estimates for the Essbase cube conversions had two rounds. However, in this experiment, a low and high range was established for each of the estimation rounds. The idea was to use a range of estimates for each major task to more accurately and responsibly predict the overall estimate for this project. On the surface this approach of having high and low values seemed to help improve accuracy. See section 6.1.3. Threats to Validity for more details.

Initially, when looking at the estimated cost and LOE for this project, this work seemed to be tracking perfectly to proposed time and cost commitments. After starting the project, it was discovered that the actual number of cubes that needed to be converted was 32.

Further into the project, it was discovered that not all cubes were created equal. One unexpected discovery included the LOE that needed to convert the control scripts and were used to load the cube data. These load scripts and other scripts were used to execute various calculation scripts before and after the data load scripts executed.

Additionally some cubes required complex transformation that simply weren't possible with simple load rules. Other features included extensive logging for troubleshooting purposes needed to support the cube data load process, and for auditing that is required to meet certain service level agreements (SLAs).

In order to complete the work on time, , 3 additional vendor resources and 2 internal SWA resources were needed.. To offset the budget cost overruns, some vendor resources that were already at SWA were used to complete this work. Additionally, SWA was able to provide the vendor with flight benefits to offset the Travel and Incidental (T&I) costs by 70%.

On the estimation front, the table below (Figure 5.9 Essbase Upgrade Estimate) provides a summary of the estimation rounds. In the first round the SMEs estimated 3 days per cube for 59 cubes ($59 \times 3 = 177$ days). The actuals for this project were around 212 total days. The problem with this estimate is that it didn't take into account cube complexity factors and it overestimated the number of cubes. The utility's high round estimate even though it was only conducted for the actual 32 cubes that needed converting was 164 days due to taking into account some of the complexity factors.

In the final rounds of estimates, the SMEs took into account some complexity factors and came up with 159 days. The utility on the other hand, using the same input parameters (numbers from figure 5.9) concluded that it would take 194 days to complete the conversion and upgrade.

Essbase Cubes	Experts Low (R1)	Experts High (R1)	Experts Low (R2)	Experts High (R2)	Cube Complexity	Data Complexity	Customer Sensitivity	Control Scripts Complexity	Front-End Complexity	Utility Low (R1)	Utility High (R1)	Utility Low (R2)	Utility High (R2)
Airport	2	3	3	3	L					2	3	2	3
HE_Hist	2	3	3	5	M					4	5	4	5
HE_SAP	2	3	3	5	M					4	5	4	5
SWA_BS	2	3	3	3	L					2	3	2	3
SWA_IS	2	3	3	7	H	H				6	8	8	9
CltyBPI	2	3	3	5	M		H	H		4	5	5	7
CltyCap	2	3	3	5	M		H	H		4	5	5	7
CltyCapR	2	3	3	5	M		H	H		4	5	5	7
CltyFS	2	3	3	5	M		H	H		4	5	5	7
CltyFuel	2	3	3	5	M		H	H		4	5	5	7
CltyLT	2	3	3	5	M		H	H		4	5	5	7
CltyRep	2	3	3	7	H		H	H		6	8	9	11
Flash	2	3	3	5	M		H	H		4	5	5	7
TBF	2	3	3	10	E	H		H	H	8	10	10	13
Bkgs, BkgsC, H	2	3	3	5	M					4	5	4	5
AllStaff	2	3	3	7	H	H				6	8	8	9
RevTraf, RevTrafC,H	2	3	3	7	H	H				6	8	8	9
FltProf	2	3	3	5	M	H				4	5	5	6
OTP	2	3	3	5	M		H			4	5	5	6
ProvoRpt	2	3	3	3	L					2	3	2	3
ScorTGT	2	3	3	3	L					2	3	2	3
If_Rpt	2	3	3	5	M					4	5	4	5
IF_Stats	2	3	3	5	M					4	5	4	5
FO_Rpt	2	3	3	3	L					2	3	2	3
TO_Hours	2	3	3	3	L					2	3	2	3
MX_Perf, MX_MAS	2	3	3	3	L					2	3	2	3
EBB_Main	2	3	3	5	M					4	5	4	5
Industry	2	3	3	5	M					4	5	4	5
SLV_GC	2	3	3	5	M					4	5	4	5
GO_Rpt, GO_RptHs	2	3	3	7	H	H		H		6	8	9	11
GO_Rx	2	3	3	3	L			H		2	3	3	4
OTP Daily	2	3	3	5	M		H			4	5	5	6
Total (days)	64	96	96	159						126	164	151	194

Figure 5-16 Essbase Upgrade Estimate

5.5.4 Data Analysis

As was mentioned earlier, the list of cubes that needed to be converted (Figure 5.9 Essbase Upgrade Estimate) was provided in the form of a low and a high range. For each of the estimation rounds, pairs of estimates were provided that consisted of low and high values that were calculated by SMEs and their equivalent low and high values that were measured by the estimation utility.

Once again, due to the characteristics of the data (a set of paired observations from a normal population), the Paired t-test was selected to examine the difference between the pair of values from the two estimation rounds. Additionally, ICC values were computed to examine likes with likes (lows from both rounds and then highs from the same rounds). Here, the purpose of the tests is to determine whether or not there is a significant statistical difference between values being measured. Additionally, examining the Pearson values should reveal reliable proportions of smoothness and tightness and quantifiable measurements for the data above.

5.5.4.1 One-Sample T-Test

In this context, we used the t-test to compare one set of estimated measurements with a second set from the same sample data above. We compare the before values (generated by the SMEs) against the after values (generated by the utility) to determine whether significant change has occurred.

From the estimation values above, one of the interesting test cases was to test the highs for SMEs against the lows of the utility. This test is more interesting because it is closer to reality in terms of what was chosen by the SMEs. In this test we wanted to see if the utility increases estimates even when we use the lower bounds of the utility for initial high level estimates. Additionally since all values produced by the SMEs were multiples

of 3, the mean value to test against was chosen to be 3. Using this value as the hypothesized mean we get the following results:

Table 5-7 One-Sample T test for the SMEs high round I against utility low from round I using a test of mu value equal to 3 vs. not equal to 3

Variable	N	Mean	StDev	SE Mean	95% CI	T	P
Utility Low (R1)	32	3.938	1.480	0.262	(3.404, 4.471)	3.58	0.001

The mean value for the SMEs estimates was 3 for the high. On the other hand, the mean for the utility was 3.938 which is even higher. The test above proves that it is statistically different using an 85% confidence interval (CI).

What is more, this problem has 31 degrees of freedom. The test above is one-tailed because we're only trying to determine whether the utility increases estimates and not reduces them. The critical value from the t-table for $t_{.05,32}$ is 1.6939. Because the computed t-value of 3.58 is larger than 1.6939, the null hypothesis can be rejected. The test above has provided sufficient evidence that the utility caused the estimate to be more than if it had not been used by the SMEs. From examining the data above it is obvious that the amount of actual increase was large (30 days more than what the SMEs estimated), and it is worth noting that it was statistically significant.

Similarly, testing the final estimates (SMEs high against utility high) with mean value for the SMEs estimates 4.969 reveals the results below:

Table 5-8 One-Sample T test for the SMEs and the utility (highs from round II) using a test of mu value equal to 4.969 vs. not equal to 4.969

Variable	N	Mean	StDev	SE Mean	95% CI	T	P
Utility Low (R1)	32	6.063	2.590	0.458	(5.129, 6.99)	2.39	0.023

The two-tailed P value equals for this data set is 0.0231. By conventional criteria, this difference is also considered to be statistically significant (see Appendix A, Experiment III for remaining t-test results).

5.5.4.2 Intra-class Correlation Coefficient (ICC) Test

Next, a matrix of scatter plot is used to visualize the relationship between like estimates. Since there was no variability in the estimates produced by the SMEs that were multiples of 2s and 3s, these estimates were automatically eliminated by the tool. Inspecting the graphs below reveals that the utility values exhibit tighter fits with much higher internal consistency when compared against itself.

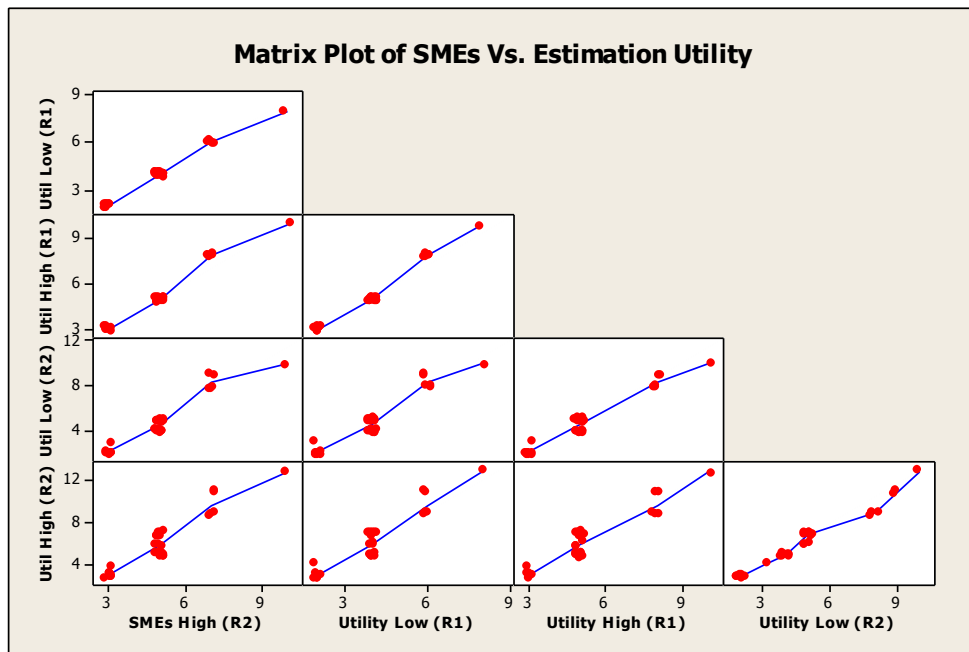


Figure 5-17 Experiment III – Matrix of scatter plots for SMEs, Utility

Following that, we compute the Intra-Class Correlation Coefficient (ICC) to examine the reliability and validity of the data. The resulting Pearson correlation values are displayed in Appendix A, Experiment III. All of the Pearson correlation values were extremely high and once again the test reveals that the highest correlation is found after using the utility values (0.995). This indicates that the estimates produced by the SMEs and the utility all have very high level of agreement, but the highest and most consistent are the ones produced by the utility.

Chapter 6

Results

6.1. Results

The section below reviews quantitative and qualitative aspects of the estimates produced by the utility. This is done by examining and answering the research questions that were proposed in section 5.2.

6.1.1. Quantitative results

“In God we trust; all others must bring data.”
—Attributed to W. Edwards Deming

After finalizing the results that were discussed in the data analysis sections of chapter 5, this section will focus on proving or disproving the proposed hypothesis. The experiments conducted earlier in section 5 included a statistical analysis portion. This analysis was useful for testing the results of the experiments. The goal was to reach a comprehensive answer that filters out external factors that may be potentially seen as real and unbiased.

Distinctively, the result from the analysis in the previous section shows that the actual level of effort for all experiments was found to be almost 3 to 4 times larger than those that were estimated by the SMEs. For large initiatives, the major contributing factors to the increase the estimates include not accurately capturing the number of data elements, not correctly capturing number of source files, and most of all, not factoring in issues pertaining to data quality and complexity.

Moreover, when comparing the results of the actuals with the estimates that were produced by the estimation utility, the utility was well within the plus minus 20 percent margin of error. In fact, in experiment 3, the utility was within 5% of the actual estimate.

Additionally, in all of the experiments, when examining the pairs of data for each line item, the difference between the utility and the SME estimates was quite substantial.

The main difference was that the utility took into account the particular complexities of the artifacts being estimated themselves. This included not accurately capturing the number of elements that needed to be evaluated, the complexity of the proposed items being built, and the complexity of the data.

As an example, in the experiment III (from Figure 5.9 Essbase Upgrade Estimate), a blanket value of 3 days was given to all cubes. In turn, that caused the technology budget forecasting (TBF) cube, one of the most complex cubes, to receive the same estimate as some of the least complex cubes. On the other hand, the low range value for the utility estimate for the TBF cube was 10 days and the high value was 13 days.

6.1.2. Qualitative results

In this section, we focus on the hypothesis questions that were proposed in section 5.2. In the first question the query dealt with if the utility would help decrease the optimism biases. In other words, the goal was to see if the utility can prevent the prevalent underestimating that was taking place throughout most of the major initiatives.

From examining the quantitative data (chapter 5) it was clear that LOE and size estimates that the utility produced were even better than the plus minus 20 percent margin of error. The utility definitely helps EI teams who frequently underestimate major efforts improve their estimates to be within the 20% target range.

As far as improving the accuracy of the estimates, the results above show that with quantifiable certainty that the utility is able to produce more accurate estimates. This is due to the fact that the utility factors in all the details of proposed objects to be built and doesn't make sweeping generalizations.

Another accuracy consideration is that the utility provides a low and high value estimate. This is better than a signal number estimate because it takes into consideration

changeability or likely variability of a particular task. Low estimates tend to be interpreted as the absolute smallest amount of time to complete a particular task (lowest probability of coming true). On the other hand, the high estimate is usually interpreted as the most likely LOE for a task to get done. However, due to the fact that developers tend to be optimistic (see section 1.2 Important Estimation Concepts), these low and high estimates are not any better than a single value estimate. When this optimism is amplified over multiple tasks, this leads to the eventual underestimation of the entire project. Good estimates are ones that take into account the long right tail of the estimate's probability distribution function. The utility addresses this by factoring this into account in the low and high range values. Explicitly, the utility makes it easy for the SMEs to account for the details that are normally left out of most estimates. These details are often the very reason for tasks being pushed out or most likely are often unaccounted for.

The second question dealt with whether the use of the estimation utility could help facilitate group discussions. When reviewing the results from the experiments above, it is worth noting that in subsequent rounds the utility played an important role in reminding the SMEs of certain characteristics that they had forgotten to consider in previous rounds. These are often small but vital details for a proposed build objects. In most cases, if the SMEs consider such quintessential details such. Customer relationship risk, number of source data feeds, estimates are buffered to some extent. With these items factored into the estimation utility, seasoned SMEs can remember to take them into consideration and helps SMEs be consistent when considering such factors. Likewise, the details in the utility would help average estimators remember important aspects that may not have come to their minds, yet such detail is imperative and can greatly influence estimated results.

Similarly, it was evident from the quantitative details that the utility helps offset cognitive bias by reducing initial risky or optimistic estimates. Again, this is attributed to the utility in essence acting like a checklist that bounds the discussion in terms of facilitating coming to a decision regarding important influence factors such how complex are the items themselves that are being built, the data they use, and even important subtle weights such a customer influence factor.

Additionally, using the utility can help the SMEs avoid making blanket assumptions about the components being estimated. Instead of assuming that components being built are some multiple of a certain duration (two hours per graph, or 3 days per cube), the utility uses a more nuanced approach for evaluating these elements. This is accomplished by having groups of similar components only be examined against complexity factors that were calibrated and agreed upon for such components.

As far as improving the consistency and transparency of estimates, the case studies above demonstrated that using the estimation utility will not only improve the estimation accuracy and transparency in quantitative terms, but also in qualitative terms. In situations where SMEs agreed that TBF is considered complex, SMEs had different interpretation of what complex means. The utility provides such consistency. For instance, the utility is meant to be used as a tool that helps guide discussions as to why a particular estimate is of certain accuracy. Here the utility acts like guide rails that can be used to help SMEs be confident in justifying his or her estimates. Ultimately, this will help increase the confidence the SMEs had in their estimates

Another added benefit, is that due to the fact the utility documents assumption and constraints, this prevents managers from a depleting condition known as premature commitment. Again, this increases consistency and transparency by allowing the teams

to have an intelligent conversation in regards to what they have estimated and committed.

6.1.3. Threats to Validity

Threats to the validity are mainly on the small population size. The quantitative and qualitative arguments in this paper are based on 3 case studies and their sample observations. Even though the sample data from the experiments was large, a few concerns emerge. The first trepidation is related to the fact that most of these observations are stored in disorganized files and not necessarily from an estimation data base (DB). Ideally, such data should come from years of data stored in a DB where teams store cycle times for all work items.

Related to the concern above, is that if the same data analysis was carried out with a much larger population, it could potentially lead to different results. It is hard to ensure that the results are not based on some claim flip model, where by chance, 50% of the time the estimation utility is a hit. If such a DB existed, ideally we would examine the whole DB or some portion of it over time to see if the number of hits standouts from a number that you'd expect by chance.

Another threat to validity is that not all the participants are taught with sound principals of software engineering or have much experience with points estimating. It is possible that if the experiments are run again with a large population of SMEs it might lead to different results.

Another concern is that in some cases the financial results don't seem to back up the actuals. This might be due to budgetary concerns and charging time to buckets that have excess dollars or to the wrong buckets. The experiments largely focused on the size and effort estimation and only took into consideration cost and planning aspects of projects when they were available (experiment II). Without having the costs, the

estimates, and the project plans all sing from the same song sheet, it is hard to conduct a thorough postpartum to verify that we delivered on what was estimated.

Finally, the last consideration in most of these projects is that the time to execute them was predefined to meet certain business objectives. This introduces some bias that could influence the overall estimates.

6.1.4. Accuracy of Estimates

Calculating the accuracy of estimates can help reveal if the SMEs are reporting more effort that is needed (fluffing) or if they are underestimating. Clearly from reviewing the data analysis, it is evident that the SMEs were underestimating. One way to measure estimation accuracy is it to calculate a value for estimation error. This is done by comparing the estimate to an exact value (the actual). The formula Estimating Error = Estimate – Actual is used to compute error as a percent of the exact value. Negative errors indicate underestimating whereas, positive errors indicate overestimating.

As stated earlier it is clear that the SMEs are on average underestimating. The formula above is used to summarize the effort actuals and estimation accuracy. None of the experiments show any overestimating by the utility or the SMEs. However, due to the limited data set, we can't conclude that the utility doesn't exacerbate overestimation if it were to occur in certain projects.

Table 6-1 Size actuals and their respective estimation accuracy values

Experiment	Estimating Error
International	-7008 (hours)
International – Reporting	-112 (hours)
ETL Graphs	-24 (points)
Essbase Cube	-30 (points)

It is worth noting that the values for the actuals included many project change requests (PCRs) that were not accounted for in the original estimate. When PCR occurred, they were absorbed and no new estimates were produced.

Chapter 7

Conclusion

The goal of this research was to explore an effective way to provide estimates for an enterprise data warehouse (EDW). For large EDW projects, SMEs would often use improper linear extrapolation from previously completed projects to extrapolate estimates for newly proposed projects. This led to drastic underestimation when projects ended up being dramatically different in scope, complexity, and size.

The thesis described a comprehensive technique to help SMEs estimate large EDW projects. This technique employed a custom built tool or estimation utility. The tool takes into account various nuances of an EDW. Some of these nuances include type of technology being built; the number of data sources, build object complexity, and data complexity. The tool uses these components to substantially improve project effort estimation. The tool is then used to communicate estimates to planning teams, delivery teams, managers, and software architects.

The effectiveness of the tool was demonstrated by examining estimated numbers from three large commercial EDW projects at a national airline. Comparing the estimates with the actuals revealed that the tool predicted the projects' level of effort within ten to twenty percent accuracy.

Appendix A

Chapter 2 Supplemental Material

Table A-1 Basic COCOMO Model (56)

Software Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

		X-Small	Small	Medium	Large	X-Large
Financial	Total cost	\$50,000 - \$250,000	\$250,000 - \$500,000	\$500,000 - \$1,000,000	\$1,000,000 - \$10,000,000	>\$10,000,000
Schedule	Elapsed Time	[2 - 4 mos]	[4 - 6 mos]	[6 - 9 mos]	[9 - 12 mos]	[Over 12 mos]
Team Complexity	# of Teams	1-2 Application Suites	1-2 Application Suites	2-4 Application Suites	4 - 6 Application Suites	> 6 Application Suites
Dependency	# of other Projects dependent on this Project	1-3	1-3	3-6	6-9	>9
Technology	Architecture Compliance	No New Tech	No New Tech	Minor Differences / Changes	Major Differences / Changes	Significant Differences / Changes
	Architecture Complexity	Normal	Normal	Complex	Major	Significant
Business Alignment / Scope	Scope Complexity	Minor	Average	Major	Complex Req Set	Complex with many dependencies
	Business Impact	Not Significant Business Change	Impactful Business Change	Major Business Change	Significant Business Change	Competitive Advantage

Figure A-1 T-Shirt Sizing Standards

Appendix B
Experiments

The Personnel Projection Matrix table below was furnished from the vendor's SOW. The table outlines the complete details of the proposed employment projections.

Table B-1 Personnel Projection Matrix – SOW employment projections details

Resource Type	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
Project Manager	1	1	1	1	1	1	1	1	1	1
Team Lead	1	1	1	1	1	1	1	1	1	1
Senior Developer I	1	1	1	1	1	1	1	1	1	1
Senior Developer II	1	1	1	1	1	1	1	1	1	1
Developer	1	1	1	1	1	1	1	1	1	1
Test Analyst	0	1	1	1	1	1	1	1	1	1
Technical Writer	0	1	1	1	1	1	1	1	1	1
Total	5	7	7	7	7	7	7	7	7	7

Experiment I (EI-STARS team)

The figure below is the full data set that was used to generate the estimate that was requested in support of the reporting capabilities for the international sell operation. This estimate for the "Data Feeds and Reporting" Work Package (WP116).

Applications and/or Areas	Confidence Factor Percentage	Work Type Category (see categories below)	# Reports Impacted By Work Packages	Total Hours					Actuals	High			Medium			Low			
				Design	Code	Test	Expert	Utility		Design	Code	Test	Design	Code	Test	Design	Code	Test	
SPN	60%	B	27	WP101	25	46	46	117	416	425	5.4	10.8	10.8	13.5	27	27	5.4	8.1	8.1
CS2	60%	B; ND	31	WP106, 107, 108, 109, 110, 111, 115, 201, 202, 203, 204	28	53	53	134	576	592	6.2	12.4	12.4	15.5	31	31	6.2	9.3	9.3
QIK	60%	B; ND	2	WP106, 107, 108	2	4	4	10	32	34	0.4	0.8	0.8	1	2	0.4	0.6	0.6	
RefundPro	60%	B; ND	22	WP111, 215	20	38	38	96	384	399	4.4	8.8	8.8	11	22	4.4	6.6	6.6	
CFM	60%	B; ND	44	WP106, 109, 110, 111, 216	40	75	75	190	1152	1249	8.8	17.6	17.6	22	44	8.8	13.2	13.2	
Station Reporting	60%	B; ND	34	WP106, 107, 108, 109, 110, 111, 115, 116, 201, 202, 203, 205	31	58	58	147	480	413	6.8	13.6	13.6	17	34	6.8	10.2	10.2	
BRUTS	60%	B	71	WP201, 205	64	121	121	306	960	884	14.2	28.4	28.4	35.5	71	14.2	21.3	21.3	
ODS	60%	B	20	WP212	18	34	34	86	288	294	4	8	8	10	20	4	6	6	
SPT	60%	B	9	WP212	9	16	16	41	128	130	1.8	3.6	3.6	4.5	9	1.8	2.7	2.7	
LMS	60%	B	15	WP212	14	26	26	66	224	233	3	6	6	7.5	15	3	4.5	4.5	
SOPI	60%	ND	47	WP204, 2148	43	80	80	203	704	726	9.4	18.8	18.8	23.5	47	9.4	14.1	14.1	
MDM	60%	ND	13	WP204, 2148	12	23	23	58	192	202	2.6	5.2	5.2	6.5	13	2.6	3.9	3.9	
FOAR	60%	B	4	WP201, 205, 216	4	7	7	18	64	67	0.8	1.6	1.6	2	4	0.8	1.2	1.2	
Taxes	60%	B	5	WP101	5	9	9	23	80	88	1	2	2	2.5	5	1	1.5	1.5	
Airport Application Suite	90%	B; NR	14	WP215	13	24	24	61	224	241	2.8	5.6	5.6	7	14	2.8	4.2	4.2	
Flight Ops	70%	NR	3	WP215	3	6	6	15	32	33	0.6	1.2	1.2	1.5	3	0.6	0.9	0.9	
EDW	0%	O	2	WP116, 205	11	4	4	19	16	54	0	0	0	0	0	0	0	0	
Total Reports			344		342	624	624	1590	5952	6064									

Work Type Category	High	Medium	Low	High	10%
R = Required to Fly	2	1	0.5	Medium	50%
B = Breakfix	4	2	0.75	Low	40%
NR = New work - RTF	4	2	0.75		
ND = New work - desired		5			
NR = New Regulatory	10		2		
O = Other					

Figure B-1 Full data set used to generate Table 5.1 for experiment 1 from group 1

Appendix C

Threats to Validity

MCP_ITIN_FLT_LEG graph embedded SQL query:

```
INSERT INTO ${EDW_ST_SCHEMA}. INSERT INTO ${EDW_ST_SCHEMA}.MCP_ITIN_FLT_LEG
(OPNG_ITIN_FLT_PATH_ID,
ITIN_DEP_DT,
LEG_DEP_DAY_OFST_ARRAY_TXT,
ITIN_LEG_SEQ_NUM,
OPNG_CARR_CDE,
OPNG_FLT_NUM,
FLT_DEP_DT,
ORIG_STN_CDE,
DEST_STN_CDE,
ITIN_ORIG_STN_CDE,
ITIN_DEST_STN_CDE,
MAX_ITIN_LEG_SEQ_NUM,
ITIN_LEG_DEP_DAY_OFST_NUM,
ITIN_LEG_TYPE_CDE,
CONN_FLAG,
ACTL_CONN_MINI,
SCHD_CONN_MINI,
ACTL_CONN_VRNC_MINI,
ROW_SRCE_CDE,
ORIGINAL_JOB_ID,
LATEST_JOB_ID)
/* comment */
SELECT
COALESCE (actual.OPNG_ITIN_FLT_PATH_ID, booked.OPNG_ITIN_FLT_PATH_ID) AS
OPNG_ITIN_FLT_PATH_ID /* operating path ID */
,COALESCE (actual.ITIN_DEP_DT, booked.ITIN_DEP_DT) AS ITIN_DEP_DT /* Itinerary
departure data */
,COALESCE (actual.LEG_DEP_DAY_OFST_ARRAY_TXT,
booked.LEG_DEP_DAY_OFST_ARRAY_TXT, 0) AS LEG_DEP_DAY_OFST_ARRAY_TXT /* leg
departure offset number from the itinerary departure */
,COALESCE (actual.ITIN_LEG_SEQ_NUM, booked.ITIN_LEG_SEQ_NUM) AS
ITIN_LEG_SEQ_NUM /* leg sequence number specific to the itinerary path ID */
,COALESCE (actual.OPNG_CARR_CDE, booked.OPNG_CARR_CDE) AS OPNG_CARR_CDE /*
operating carrier code */
,COALESCE (actual.OPNG_FLT_NUM, booked.OPNG_FLT_NUM) AS OPNG_FLT_NUM /* flight
number */
,COALESCE (actual.FLT_DEP_DT, booked.FLT_DEP_DT) AS FLT_DEP_DT /* flight leg
departure date */
,COALESCE (actual.ORIG_STN_CDE, booked.ORIG_STN_CDE) AS ORIG_STN_CDE /*
originating station code */
,COALESCE (actual.DEST_STN_CDE, booked.DEST_STN_CDE) AS DEST_STN_CDE /*
destination station code */
,COALESCE (actual.ITIN_ORIG_STN_CDE, booked.ITIN_ORIG_STN_CDE, '') AS
ITIN_ORIG_STN_CDE /* itinerary originating station */
,COALESCE (actual.ITIN_DEST_STN_CDE, booked.ITIN_DEST_STN_CDE, '') AS
ITIN_DEST_STN_CDE /* itinerary destination station */
,COALESCE (actual.MAX_ITIN_LEG_SEQ_NUM, booked.MAX_ITIN_LEG_SEQ_NUM, 0) AS
MAX_ITIN_LEG_SEQ_NUM /* number of leg sequences associated with one itinerary
path */
,COALESCE (actual.ITIN_LEG_DEP_DAY_OFST_NUM, booked.ITIN_LEG_DEP_DAY_OFST_NUM,
0) AS ITIN_LEG_DEP_DAY_OFST_NUM /* leg departure offset number from the
itinerary departure */
,COALESCE (actual.ITIN_LEG_TYPE_CDE, booked.ITIN_LEG_TYPE_CDE, 0) AS
ITIN_LEG_TYPE_CDE /* Itinerary leg type code from ITIN_FLT_PATH_LEG */
,COALESCE (actual.CONN_FLAG, booked.CONN_FLAG, '') AS CONN_FLAG /* connection
flag */
```

```

,COALESCE (actual.ACTL_CONN_MINI, booked.ACTL_CONN_MINI, 0) AS ACTL_CONN_MINI
/* actual connection minutes */
,COALESCE (booked.SCHD_CONN_MINI, 0) AS SCHD_CONN_MINI /* scheduled connection
minutes */
,COALESCE (actual.ACTL_CONN_VRNC_MINI, 0) AS ACTL_CONN_VRNC_MINI /* connection
minutes variance */
,CASE WHEN actual.ROW_SRCE_CDE_FOS = 1 AND booked.ROW_SRCE_CDE_BPPSL = 1 THEN
'BOTH' /* itinerary source info. The row either came from BPPSL or FLT OCCR
SEAT aka FOS */
WHEN booked.ROW_SRCE_CDE_BPPSL = 1 THEN 'BPPSL' /* the itinerary was sourced
from BPPSL */
WHEN actual.ROW_SRCE_CDE_FOS = 1 THEN 'FOS' /* the itinerary was sourced from
FLT OCCR SEAT */
END AS ROW_SOURCE_CDE
,${JOB_ID} AS ORIGINAL_JOB_ID /* JOB ID */
,${JOB_ID} AS LATEST_JOB_ID /* JOB ID */
/* SQL PULL to get actual itinerary information using FLT_OCCR_SEAT as the
driving data source */
FROM (
SELECT fos.FLWN_OPNG_ITIN_FLT_PATH_ID AS OPNG_ITIN_FLT_PATH_ID,
fos.FLWN_ITIN_DEP_DT AS ITIN_DEP_DT,
fos.FLWN_ITIN_LEG_SEQ_NUM AS ITIN_LEG_SEQ_NUM,
fos.FLWN_OPNG_CARR_CDE AS OPNG_CARR_CDE,
fos.FLWN_FLT_NUM AS OPNG_FLT_NUM,
fos.FLWN_LEG_DEP_DT AS FLT_DEP_DT,
fos.FLWN_LEG_ORIG_CDE AS ORIG_STN_CDE,
fos.FLWN_LEG_DEST_CDE AS DEST_STN_CDE,
itin.ITIN_ORIG_ARPT_CDE AS ITIN_ORIG_STN_CDE,
itin.ITIN_DEST_ARPT_CDE AS ITIN_DEST_STN_CDE,
itin.MAX_ITIN_LEG_SEQ_NUM AS MAX_ITIN_LEG_SEQ_NUM,
itin.ITIN_LEG_TYPE_CDE AS ITIN_LEG_TYPE_CDE,
fos.FLWN_ITIN_DEP_DT - fos.FLWN_LEG_DEP_DT AS ITIN_LEG_DEP_DAY_OFST_NUM,
array.LEG_DEP_DAY_OFST_ARRAY_TXT AS LEG_DEP_DAY_OFST_ARRAY_TXT,
CASE WHEN itin.CONN_IND_CDE = 'X' AND ITIN_LEG_SEQ_NUM > 0
THEN 1
ELSE 0 END AS CONN_FLAG,
fo.ACTL_TM_OUT_LCL/100 *60 + fo.ACTL_TM_OUT_LCL MOD 100 AS TIME_OUT, /* These
columns are mostly used to help debug */
fo.ACTL_TM_IN_LCL/100 *60 + fo.ACTL_TM_IN_LCL MOD 100 AS TIME_IN, /*
calculate the time in */
CASE WHEN itin.CONN_IND_CDE = '-'
THEN TIME_IN
ELSE TIME_OUT END AS TEMP_TIME, /* establish temp_time column use in the
MDIFF function */
CASE WHEN itin.CONN_IND_CDE = 'X' AND ITIN_LEG_SEQ_NUM > 0
THEN MDIFF(TEMP_TIME,1,OPNG_ITIN_FLT_PATH_ID, OPNG_CARR_CDE, ITIN_DEP_DT,
ITIN_LEG_SEQ_NUM) /* Use an OLAP function to calculate connection mintues. It
subtracts the current value of TEMP TIME column by the previous */
ELSE 0 END AS ACTL_CONN_MINI,
CAST(schd.CONN_MIN_ITRVL AS INTEGER) AS SCHDMINS,
ACTL_CONN_MINI - SCHDMINS AS ACTL_CONN_VRNC_MINI,
1 AS ROW_SRCE_CDE_FOS,
0 AS ROW_SRCE_CDE_BPPSL
FROM
(SELECT FLWN_OPNG_ITIN_FLT_PATH_ID,
FLWN_ITIN_DEP_DT,
FLWN_ITIN_LEG_SEQ_NUM,
FLWN_OPNG_CARR_CDE,
FLWN_FLT_NUM,

```

```

FLWN_LEG_DEP_DT,
FLWN_LEG_ORIG_CDE,
FLWN_LEG_DEST_CDE
FROM ${ETL_VW_SCHEMA}.FLT_OCCR_SEAT
WHERE FLWN_LEG_DEP_DT BETWEEN '${LOAD_FROM_DT}' and '${LOAD_TO_DT}'
AND NON_REV_STBY_IND = 'N'
GROUP BY 1,2,3,4,5,6,7,8) fos
INNER JOIN ${ETL_VW_SCHEMA}.ITIN_FLT_PATH_LEG itin
ON
fos.flwn_opng_carr_cde = itin.CARR_CDE AND
fos.FLWN_OPNG_ITIN_FLT_PATH_ID = itin.ITIN_FLT_PATH_ID AND
fos.FLWN_ITIN_LEG_SEQ_NUM = itin.ITIN_LEG_SEQ_NUM AND
fos.FLWN_LEG_ORIG_CDE = itin.LEG_ORIG_ARPT_CDE AND
fos.FLWN_LEG_DEST_CDE = itin.LEG_DEST_ARPT_CDE
LEFT OUTER JOIN
(SELECT
FLT_NUM,
LEG_ORIG_CDE,
LEG_DEP_DT,
LEG_DEST_CDE,
ACTL_TM_IN_LCL,
ACTL_TM_OUT_LCL
FROM ${ETL_VW_SCHEMA}.FLT_OCCR) fo
ON
fos.FLWN_FLT_NUM = fo.FLT_NUM AND
fos.FLWN_LEG_ORIG_CDE = fo.LEG_ORIG_CDE AND
fos.FLWN_LEG_DEP_DT = fo.LEG_DEP_DT AND
fos.FLWN_LEG_DEST_CDE = fo.LEG_DEST_CDE
LEFT OUTER JOIN
(SELECT ITIN_FLT_PATH_ID,
FLT_NUM,
FLT_DEP_DT,
ORIG_ARPT_CDE,
DEST_ARPT_CDE,
CONN_MIN_ITRVL
FROM ${ETL_VW_SCHEMA}.SCHD_ITIN_FLT_LEG
WHERE EFF_TO_DT = '2099-12-31' AND
ITIN_EFF_TO_DT = '2099-12-31' ) schd
ON
fos.FLWN_OPNG_ITIN_FLT_PATH_ID = schd.ITIN_FLT_PATH_ID AND
fos.FLWN_FLT_NUM = schd.FLT_NUM AND
fos.FLWN_LEG_ORIG_CDE = schd.orig_arpt_cde AND
fos.FLWN_LEG_DEP_DT = schd.FLT_DEP_DT AND
fos.FLWN_LEG_DEST_CDE = schd.dest_arpt_cde
LEFT OUTER JOIN (
SELECT FLWN_OPNG_ITIN_FLT_PATH_ID,
MAX( CASE WHEN row_nbr=1 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE ''
END) ||
CASE WHEN MAX( CASE WHEN row_nbr=2 THEN CAST
(ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE
WHEN row_nbr=2 THEN ';' || CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE ''
END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=3 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=3 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=4 THEN CAST
(ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE
WHEN row_nbr=4 THEN ';' || CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE ''
END) END ||

```

```

CASE WHEN MAX( CASE WHEN row_nbr=5 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=5 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=6 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=6 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=7 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=7 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END
AS LEG_DEP_DAY_OFST_ARRAY_TXT
FROM
(SELECT FLWN_OPNG_ITIN_FLT_PATH_ID, ITIN_LEG_DEP_DAY_OFST_NUM, SUM(1) OVER
(PARTITION BY FLWN_OPNG_ITIN_FLT_PATH_ID ORDER BY FLWN_OPNG_ITIN_FLT_PATH_ID,
ITIN_LEG_DEP_DAY_OFST_NUM ROWS UNBOUNDED PRECEDING) AS row_nbr
FROM (SELECT FLWN_OPNG_ITIN_FLT_PATH_ID, FLWN_ITIN_LEG_SEQ_NUM,
FLWN_ITIN_DEP_DT - FLWN_LEG_DEP_DT AS ITIN_LEG_DEP_DAY_OFST_NUM FROM
${ETL_VW_SCHEMA}.FLT_OCCR_SEAT WHERE FLWN_LEG_DEP_DT BETWEEN '${LOAD_FROM_DT}'
and '${LOAD_TO_DT}' GROUP BY 1,2,3) y ) x
GROUP BY 1
) array
ON
fos.FLWN_OPNG_ITIN_FLT_PATH_ID = array.FLWN_OPNG_ITIN_FLT_PATH_ID
) actual
FULL OUTER JOIN
(SELECT bpspl.MKT_OPNG_ITIN_FLT_PATH_ID AS OPNG_ITIN_FLT_PATH_ID,
bpspl.MKT_DEP_DT AS ITIN_DEP_DT,
bpspl.MKT_LEG_DEP_DAY_OFST_ARRAY_TXT AS LEG_DEP_DAY_OFST_ARRAY_TXT,
bpspl.MKT_OPNG_ITIN_LEG_SEQ_NUM AS ITIN_LEG_SEQ_NUM,
bpspl.OPNG_CARR_CDE AS OPNG_CARR_CDE,
bpspl.OPNG_FLT_NUM AS OPNG_FLT_NUM,
bpspl.FLT_DEP_DT AS FLT_DEP_DT,
bpspl.ORIG_ARPT_CDE AS ORIG_STN_CDE,
bpspl.DEST_ARPT_CDE AS DEST_STN_CDE,
itin.ITIN_ORIG_ARPT_CDE AS ITIN_ORIG_STN_CDE,
itin.ITIN_DEST_ARPT_CDE AS ITIN_DEST_STN_CDE,
itin.MAX_ITIN_SEG_SEQ_NUM AS MAX_ITIN_LEG_SEQ_NUM,
itin.ITIN_LEG_TYPE_CDE AS ITIN_LEG_TYPE_CDE,
bpspl.MKT_LEG_DEP_DAY_OFST_NUM AS ITIN_LEG_DEP_DAY_OFST_NUM,
CASE WHEN itin.CONN_IND_CDE = 'X'
THEN 1
ELSE 0 END AS CONN_FLAG,
NULL AS ACTL_CONN_MINI,
NULL AS ACTL_CONN_VRNC_MINI,
CAST (sched.CONN_MIN_ITRVL AS INTEGER) AS SCHD_CONN_MINI,
0 AS ROW_SRCE_CDE_FOS,
1 AS ROW_SRCE_CDE_BPPSL
FROM (
SELECT
MKT_OPNG_ITIN_FLT_PATH_ID,
MKT_DEP_DT,
MKT_LEG_DEP_DAY_OFST_ARRAY_TXT,
MKT_OPNG_ITIN_LEG_SEQ_NUM,
OPNG_CARR_CDE,
OPNG_FLT_NUM,
FLT_DEP_DT,
ORIG_ARPT_CDE,
DEST_ARPT_CDE,
MKT_LEG_DEP_DAY_OFST_NUM
FROM ${ETL_VW_SCHEMA}.BKNG_PNR_PAX_SEG_LEG

```

```

WHERE EFF_FM_DBD >= 1
  AND EFF_TO_DBD < 1
and FLT_DEP_DT BETWEEN '${LOAD_FROM_DT}' and '${LOAD_TO_DT}'
GROUP BY 1,2,3,4,5,6,7,8,9,10) bpsl
LEFT OUTER JOIN ${ETL_VW_SCHEMA}.ITIN_FLT_PATH_LEG itin
ON bpsl.MKT_OPNG_ITIN_FLT_PATH_ID = itin.ITIN_FLT_PATH_ID AND
bpsl.MKT_OPNG_ITIN_LEG_SEQ_NUM = itin.ITIN_LEG_SEQ_NUM
LEFT OUTER JOIN
(SELECT ITIN_FLT_PATH_ID,
FLT_NUM,
FLT_DEP_DT,
ORIG_ARPT_CDE,
DEST_ARPT_CDE,
CONN_MIN_ITRVL
FROM ${ETL_VW_SCHEMA}.SCHD_ITIN_FLT_LEG
WHERE EFF_TO_DT = '2099-12-31' AND
ITIN_EFF_TO_DT = '2099-12-31' ) schd
ON
bpsl.MKT_OPNG_ITIN_FLT_PATH_ID = schd.ITIN_FLT_PATH_ID AND
bpsl.OPNG_FLT_NUM = schd.FLT_NUM AND
bpsl.ORIG_ARPT_CDE = schd.orig_arpt_cde AND
bpsl.MKT_DEP_DT = schd.FLT_DEP_DT AND
bpsl.DEST_ARPT_CDE = schd.dest_arpt_cde
) booked
ON
actual.OPNG_ITIN_FLT_PATH_ID = booked.OPNG_ITIN_FLT_PATH_ID
AND
actual.FLT_DEP_DT = booked.FLT_DEP_DT
AND
actual.ITIN_LEG_SEQ_NUM = booked.ITIN_LEG_SEQ_NUM;
(OPNG_ITIN_FLT_PATH_ID,
ITIN_DEP_DT,
LEG_DEP_DAY_OFST_ARRAY_TXT,
ITIN_LEG_SEQ_NUM,
OPNG_CARR_CDE,
OPNG_FLT_NUM,
FLT_DEP_DT,
ORIG_STN_CDE,
DEST_STN_CDE,
ITIN_ORIG_STN_CDE,
ITIN_DEST_STN_CDE,
MAX_ITIN_LEG_SEQ_NUM,
ITIN_LEG_DEP_DAY_OFST_NUM,
ITIN_LEG_TYPE_CDE,
CONN_FLAG,
ACTL_CONN_MINI,
SCHD_CONN_MINI,
ACTL_CONN_VRNC_MINI,
ROW_SRCE_CDE,
ORIGINAL_JOB_ID,
LATEST_JOB_ID)
/* comment */
SELECT

COALESCE (actual.OPNG_ITIN_FLT_PATH_ID, booked.OPNG_ITIN_FLT_PATH_ID) AS
OPNG_ITIN_FLT_PATH_ID /* operating path ID */
,COALESCE (actual.ITIN_DEP_DT, booked.ITIN_DEP_DT) AS ITIN_DEP_DT /* Itinerary
departure data */

```



```

,COALESCE(actual.LEG_DEP_DAY_OFST_ARRAY_TXT,
booked.LEG_DEP_DAY_OFST_ARRAY_TXT, 0) AS LEG_DEP_DAY_OFST_ARRAY_TXT /* leg
departure offset number from the itinerary departure */
,COALESCE(actual.ITIN_LEG_SEQ_NUM, booked.ITIN_LEG_SEQ_NUM) AS
ITIN_LEG_SEQ_NUM /* leg sequence number specific to the itinerary path ID */
,COALESCE(actual.OPNG_CARR_CDE, booked.OPNG_CARR_CDE) AS OPNG_CARR_CDE /*
operating carrier code */
,COALESCE(actual.OPNG_FLT_NUM, booked.OPNG_FLT_NUM) AS OPNG_FLT_NUM /* flight
number */
,COALESCE(actual.FLT_DEP_DT, booked.FLT_DEP_DT) AS FLT_DEP_DT /* flight leg
departure date */
,COALESCE(actual.ORIG_STN_CDE, booked.ORIG_STN_CDE) AS ORIG_STN_CDE /*
originating station code */
,COALESCE(actual.DEST_STN_CDE, booked.DEST_STN_CDE) AS DEST_STN_CDE /*
destination station code */
,COALESCE(actual.ITIN_ORIG_STN_CDE, booked.ITIN_ORIG_STN_CDE, '') AS
ITIN_ORIG_STN_CDE /* itinerary originating station */
,COALESCE(actual.ITIN_DEST_STN_CDE, booked.ITIN_DEST_STN_CDE, '') AS
ITIN_DEST_STN_CDE /* itinerary destination station */
,COALESCE(actual.MAX_ITIN_LEG_SEQ_NUM, booked.MAX_ITIN_LEG_SEQ_NUM, 0) AS
MAX_ITIN_LEG_SEQ_NUM /* number of leg sequences associated with one itinerary
path */
,COALESCE(actual.ITIN_LEG_DEP_DAY_OFST_NUM, booked.ITIN_LEG_DEP_DAY_OFST_NUM,
0) AS ITIN_LEG_DEP_DAY_OFST_NUM /* leg departure offset number from the
itinerary departure */
,COALESCE(actual.ITIN_LEG_TYPE_CDE, booked.ITIN_LEG_TYPE_CDE, 0) AS
ITIN_LEG_TYPE_CDE /* Itinerary leg type code from ITIN_FLT_PATH_LEG */
,COALESCE(actual.CONN_FLAG, booked.CONN_FLAG, '') AS CONN_FLAG /* connection
flag */
,COALESCE(actual.ACTL_CONN_MINI, booked.ACTL_CONN_MINI, 0) AS ACTL_CONN_MINI
/* actual connection minutes */
,COALESCE(booked.SCHD_CONN_MINI, 0) AS SCHD_CONN_MINI /* scheduled connection
minutes */
,COALESCE(actual.ACTL_CONN_VRNC_MINI, 0) AS ACTL_CONN_VRNC_MINI /* connection
minutes variance */
,CASE WHEN actual.ROW_SRCE_CDE_FOS = 1 AND booked.ROW_SRCE_CDE_BPPSL = 1 THEN
'BOTH' /* itinerary source info. The row either came from BPPSL or FLT OCCR
SEAT aka FOS */
WHEN booked.ROW_SRCE_CDE_BPPSL = 1 THEN 'BPPSL' /* the itinerary was sourced
from BPPSL */
WHEN actual.ROW_SRCE_CDE_FOS = 1 THEN 'FOS' /* the itinerary was sourced from
FLT OCCR SEAT */
END AS ROW_SOURCE_CDE
,${JOB_ID} AS ORIGINAL_JOB_ID /* JOB ID */
,${JOB_ID} AS LATEST_JOB_ID /* JOB ID */
/* SQL PULL to get actual itinerary information using FLT_OCCR_SEAT as the
driving data source */
FROM (
SELECT fos.FLWN_OPNG_ITIN_FLT_PATH_ID AS OPNG_ITIN_FLT_PATH_ID,
fos.FLWN_ITIN_DEP_DT AS ITIN_DEP_DT,
fos.FLWN_ITIN_LEG_SEQ_NUM AS ITIN_LEG_SEQ_NUM,
fos.FLWN_OPNG_CARR_CDE AS OPNG_CARR_CDE,
fos.FLWN_FLT_NUM AS OPNG_FLT_NUM,
fos.FLWN_LEG_DEP_DT AS FLT_DEP_DT,
fos.FLWN_LEG_ORIG_CDE AS ORIG_STN_CDE,
fos.FLWN_LEG_DEST_CDE AS DEST_STN_CDE,
itin.ITIN_ORIG_ARPT_CDE AS ITIN_ORIG_STN_CDE,
itin.ITIN_DEST_ARPT_CDE AS ITIN_DEST_STN_CDE,
itin.MAX_ITIN_LEG_SEQ_NUM AS MAX_ITIN_LEG_SEQ_NUM,

```

```

itin.ITIN_LEG_TYPE_CDE AS ITIN_LEG_TYPE_CDE,
fos.FLWN_ITIN_DEP_DT - fos.FLWN_LEG_DEP_DT AS ITIN_LEG_DEP_DAY_OFST_NUM,
array.LEG_DEP_DAY_OFST_ARRAY_TXT AS LEG_DEP_DAY_OFST_ARRAY_TXT,
CASE WHEN itin.CONN_IND_CDE = 'X' AND ITIN_LEG_SEQ_NUM > 0
THEN 1
ELSE 0 END AS CONN_FLAG,
fo.ACTL_TM_OUT_LCL/100 *60 + fo.ACTL_TM_OUT_LCL MOD 100 AS TIME_OUT, /* These
columns are mostly used to help debug */
fo.ACTL_TM_IN_LCL/100 *60 + fo.ACTL_TM_IN_LCL MOD 100 AS TIME_IN, /*
calculate the time in */
CASE WHEN itin.CONN_IND_CDE = '-'
THEN TIME_IN
ELSE TIME_OUT END AS TEMP_TIME, /* establish temp_time column use in the
MDIFF function */
CASE WHEN itin.CONN_IND_CDE = 'X' AND ITIN_LEG_SEQ_NUM > 0
THEN MDIFF(TEMP_TIME,1,OPNG_ITIN_FLT_PATH_ID, OPNG_CARR_CDE, ITIN_DEP_DT,
ITIN_LEG_SEQ_NUM) /* Use an OLAP function to calculate connection minutes. It
subtracts the current value of TEMP TIME column by the previous */
ELSE 0 END AS ACTL_CONN_MINI,
CAST(schd.CONN_MIN_ITRVL AS INTEGER) AS SCHDMINS,
ACTL_CONN_MINI - SCHDMINS AS ACTL_CONN_VRNC_MINI,
1 AS ROW_SRCE_CDE_FOS,
0 AS ROW_SRCE_CDE_BPPSL
FROM
(SELECT FLWN_OPNG_ITIN_FLT_PATH_ID,
FLWN_ITIN_DEP_DT,
FLWN_ITIN_LEG_SEQ_NUM,
FLWN_OPNG_CARR_CDE,
FLWN_FLT_NUM,
FLWN_LEG_DEP_DT,
FLWN_LEG_ORIG_CDE,
FLWN_LEG_DEST_CDE
FROM ${ETL_VW_SCHEMA}.FLT_OCCR_SEAT
WHERE FLWN_LEG_DEP_DT BETWEEN '${LOAD_FROM_DT}' and '${LOAD_TO_DT}'
AND NON_REV_STBY_IND = 'N'
GROUP BY 1,2,3,4,5,6,7,8) fos
INNER JOIN ${ETL_VW_SCHEMA}.ITIN_FLT_PATH_LEG itin
ON
fos.flwn_opng_carr_cde = itin.CARR_CDE AND
fos.FLWN_OPNG_ITIN_FLT_PATH_ID = itin.ITIN_FLT_PATH_ID AND
fos.FLWN_ITIN_LEG_SEQ_NUM = itin.ITIN_LEG_SEQ_NUM AND
fos.FLWN_LEG_ORIG_CDE = itin.LEG_ORIG_ARPT_CDE AND
fos.FLWN_LEG_DEST_CDE = itin.LEG_DEST_ARPT_CDE
LEFT OUTER JOIN
(SELECT
FLT_NUM,
LEG_ORIG_CDE,
LEG_DEP_DT,
LEG_DEST_CDE,
ACTL_TM_IN_LCL,
ACTL_TM_OUT_LCL
FROM ${ETL_VW_SCHEMA}.FLT_OCCR) fo
ON
fos.FLWN_FLT_NUM = fo.FLT_NUM AND
fos.FLWN_LEG_ORIG_CDE = fo.LEG_ORIG_CDE AND
fos.FLWN_LEG_DEP_DT = fo.LEG_DEP_DT AND
fos.FLWN_LEG_DEST_CDE = fo.LEG_DEST_CDE
LEFT OUTER JOIN
(SELECT ITIN_FLT_PATH_ID,

```

```

FLT_NUM,
FLT_DEP_DT,
ORIG_ARPT_CDE,
DEST_ARPT_CDE,
CONN_MIN_ITRVL
FROM ${ETL_VW_SCHEMA}.SCHD_ITIN_FLT_LEG
WHERE EFF_TO_DT = '2099-12-31' AND
ITIN_EFF_TO_DT = '2099-12-31' ) schd
ON
fos.FLWN_OPNG_ITIN_FLT_PATH_ID = schd.ITIN_FLT_PATH_ID AND
fos.FLWN_FLT_NUM = schd.FLT_NUM AND
fos.FLWN_LEG_ORIG_CDE = schd.orig_arpt_cde AND
fos.FLWN_LEG_DEP_DT = schd.FLT_DEP_DT AND
fos.FLWN_LEG_DEST_CDE = schd.dest_arpt_cde
LEFT OUTER JOIN (
SELECT FLWN_OPNG_ITIN_FLT_PATH_ID,
MAX( CASE WHEN row_nbr=1 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE ''
END) ||
CASE WHEN MAX( CASE WHEN row_nbr=2 THEN CAST
(ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE
WHEN row_nbr=2 THEN ';' || CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE ''
END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=3 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=3 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=4 THEN CAST
(ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE
WHEN row_nbr=4 THEN ';' || CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE ''
END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=5 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=5 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=6 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=6 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END ||
CASE WHEN MAX( CASE WHEN row_nbr=7 THEN CAST (ITIN_LEG_DEP_DAY_OFST_NUM
AS CHAR) ELSE '' END)='' THEN '' ELSE MAX( CASE WHEN row_nbr=7 THEN ';' ||
CAST (ITIN_LEG_DEP_DAY_OFST_NUM AS CHAR) ELSE '' END) END
AS LEG_DEP_DAY_OFST_ARRAY_TXT
FROM
(SELECT FLWN_OPNG_ITIN_FLT_PATH_ID, ITIN_LEG_DEP_DAY_OFST_NUM, SUM(1) OVER
(PARTITION BY FLWN_OPNG_ITIN_FLT_PATH_ID ORDER BY FLWN_OPNG_ITIN_FLT_PATH_ID,
ITIN_LEG_DEP_DAY_OFST_NUM ROWS UNBOUNDED PRECEDING) AS row_nbr
FROM (SELECT FLWN_OPNG_ITIN_FLT_PATH_ID, FLWN_ITIN_LEG_SEQ_NUM,
FLWN_ITIN_DEP_DT - FLWN_LEG_DEP_DT AS ITIN_LEG_DEP_DAY_OFST_NUM FROM
${ETL_VW_SCHEMA}.FLT_OCCR_SEAT WHERE FLWN_LEG_DEP_DT BETWEEN '${LOAD_FROM_DT}'
and '${LOAD_TO_DT}' GROUP BY 1,2,3) y ) x
GROUP BY 1
) array
ON
fos.FLWN_OPNG_ITIN_FLT_PATH_ID = array.FLWN_OPNG_ITIN_FLT_PATH_ID
) actual
FULL OUTER JOIN
(SELECT bpps1.MKT_OPNG_ITIN_FLT_PATH_ID AS OPNG_ITIN_FLT_PATH_ID,
bpps1.MKT_DEP_DT AS ITIN_DEP_DT,
bpps1.MKT_LEG_DEP_DAY_OFST_ARRAY_TXT AS LEG_DEP_DAY_OFST_ARRAY_TXT,
bpps1.MKT_OPNG_ITIN_LEG_SEQ_NUM AS ITIN_LEG_SEQ_NUM,
bpps1.OPNG_CARR_CDE AS OPNG_CARR_CDE,
bpps1.OPNG_FLT_NUM AS OPNG_FLT_NUM,

```

```

bppsl.FLT_DEP_DT AS FLT_DEP_DT,
bppsl.ORIG_ARPT_CDE AS ORIG_STN_CDE,
bppsl.DEST_ARPT_CDE AS DEST_STN_CDE,
itin.ITIN_ORIG_ARPT_CDE AS ITIN_ORIG_STN_CDE,
itin.ITIN_DEST_ARPT_CDE AS ITIN_DEST_STN_CDE,
itin.MAX_ITIN_SEG_SEQ_NUM AS MAX_ITIN_LEG_SEQ_NUM,
itin.ITIN_LEG_TYPE_CDE AS ITIN_LEG_TYPE_CDE,
bppsl.MKT_LEG_DEP_DAY_OFST_NUM AS ITIN_LEG_DEP_DAY_OFST_NUM,
CASE WHEN itin.CONN_IND_CDE = 'X'
THEN 1
ELSE 0 END AS CONN_FLAG,
NULL AS ACTL_CONN_MINI,
NULL AS ACTL_CONN_VRNC_MINI,
CAST (sched.CONN_MIN_ITRVL AS INTEGER) AS SCHD_CONN_MINI,
0 AS ROW_SRCE_CDE_FOS,
1 AS ROW_SRCE_CDE_BPPSL
FROM (
SELECT
MKT_OPNG_ITIN_FLT_PATH_ID,
MKT_DEP_DT,
MKT_LEG_DEP_DAY_OFST_ARRAY_TXT,
MKT_OPNG_ITIN_LEG_SEQ_NUM,
OPNG_CARR_CDE,
OPNG_FLT_NUM,
FLT_DEP_DT,
ORIG_ARPT_CDE,
DEST_ARPT_CDE,
MKT_LEG_DEP_DAY_OFST_NUM
FROM ${ETL_VW_SCHEMA}.BKNG_PNR_PAX_SEG_LEG
WHERE EFF_FM_DBD >= 1
AND EFF_TO_DBD < 1
and FLT_DEP_DT BETWEEN '${LOAD_FROM_DT}' and '${LOAD_TO_DT}'
GROUP BY 1,2,3,4,5,6,7,8,9,10) bppsl
LEFT OUTER JOIN ${ETL_VW_SCHEMA}.ITIN_FLT_PATH_LEG itin
ON bppsl.MKT_OPNG_ITIN_FLT_PATH_ID = itin.ITIN_FLT_PATH_ID AND
bppsl.MKT_OPNG_ITIN_LEG_SEQ_NUM = itin.ITIN_LEG_SEQ_NUM
LEFT OUTER JOIN
(SELECT ITIN_FLT_PATH_ID,
FLT_NUM,
FLT_DEP_DT,
ORIG_ARPT_CDE,
DEST_ARPT_CDE,
CONN_MIN_ITRVL
FROM ${ETL_VW_SCHEMA}.SCHD_ITIN_FLT_LEG
WHERE EFF_TO_DT = '2099-12-31' AND
ITIN_EFF_TO_DT = '2099-12-31' ) sched
ON
bppsl.MKT_OPNG_ITIN_FLT_PATH_ID = sched.ITIN_FLT_PATH_ID AND
bppsl.OPNG_FLT_NUM = sched.FLT_NUM AND
bppsl.ORIG_ARPT_CDE = sched.orig_arpt_cde AND
bppsl.MKT_DEP_DT = sched.FLT_DEP_DT AND
bppsl.DEST_ARPT_CDE = sched.dest_arpt_cde
) booked
ON
actual.OPNG_ITIN_FLT_PATH_ID = booked.OPNG_ITIN_FLT_PATH_ID
AND
actual.FLT_DEP_DT = booked.FLT_DEP_DT
AND
actual.ITIN_LEG_SEQ_NUM = booked.ITIN_LEG_SEQ_NUM;

```

References

- [1] Frederick P. Brooks Jr. (1975). *The Mythical Man-Month*: Addison-Wesley. ISBN 0-201-00650-2.
- [2] Estimation, an Art or a Science? Accessed November 11, 2014.
<https://www.cprime.com/2012/12/estimation-an-art-or-a-science/>
- [3] The Tough Stuff. Accessed November 11, 2014.
<http://www.qualitydigest.com/sept97/html/qmanage.html>
- [4] Software estimation: Art or Science? Accessed November 11, 2014.
<http://www.ifpug.org/Conference%20Proceedings/IFPUG-2004/IFPUG2004-23-Dasari-software-estimation-art-or-science.pdf>
- [5] How Software Development is Like Building A House. Accessed November 11, 2014.
<http://visual.ly/how-software-development-building-house>
- [6] Building software is like building a house. Accessed November 11, 2014.
<http://www.expert.co.nz/site/blog/build-software.aspx>
- [7] Why We Should Build Software Like We Build Houses. Accessed November 11, 2014.
<http://blog-admin.wired.com/opinion/2013/01/code-bugs-programming-why-we-need-specs/>
- [8] Software Is NOT Like Building a House. Accessed November 11, 2014.
<http://www.tablexi.com/blog/2013/01/why-building-software-is-not-like-building-a-house/developers/>
- [9] Software Cost Estimation. Accessed November 11, 2014.
https://courses.cs.ut.ee/MTAT.03.244/2013_fall/uploads/Main/workshop3.pdf
- [10] Software Project Estimation: Not a Commitment, Not a Target. Accessed November 11, 2014. <http://tcagley.wordpress.com/2014/01/25/software-project-estimation-not-a-commitment-not-a-target/>
- [11] Separate Estimating from Committing. Accessed November 11, 2014.
<http://www.mountaingoatsoftware.com/blog/separate-estimating-from-committing>
- [12] Software project estimates – and targets and commitments. Accessed November 11, 2014. <http://www.ibm.adison.com/Blogger/Open-Mic/November-2011/Software-project-estimates-ndash-and-targets-and-commitments-submitted-by-Robert-T-Merrill/index.php>
- [13] Steve McConnell (2006). *Software Estimation: Demystifying the Black Art (Developer Best Practices)*.
- [14] Effective Estimation of Software. Accessed November 11, 2014.
Delivery http://secure.com.sg/courses/ICT353/Session_Collateral/TOP_07_TUT_06_SLIDES_SW_Estimating.pdf

- [15] Webinar: Estimation, Planning & Control Can Make the Difference Between Project Success and Failure. Accessed November 11, 2014.
<http://www.galorath.com/blogfiles/itmpi%20ROI%20on%20software%20process%202012.pdf>
- [16] Software development effort estimation. Accessed November 11, 2014.
http://en.wikipedia.org/wiki/Software_development_effort_estimation
- [17] Comparison and Analysis of Different Software Cost Estimation Methods. Accessed November 11, 2014. http://thesai.org/Downloads/Volume4No1/Paper_24-Comparison_and_Analysis_of_Different_Software_Cost_Estimation_Methods.pdf
- [18] Vigder, M. R. and Kark, A. W. (1994). Software Cost Estimation and Control. Software Engineering Institute for Information Technology. . Accessed November 11, 2014. <http://wwwsel.iit.nrc.ca/seldocs/cpdocs/NRC37116.pdf>.
- [19] Software Estimating Taxonomy. Accessed November 11, 2014.
http://herdingcats.typepad.com/my_weblog/2013/07/software-estimating-taxonomy.html
- [20] Insights and Trends: Current Portfolio, Program, and Project Management Practices (The third global survey on the current state of project management), PwC, 2012.
- [21] Estimation is at the root of most software project failures. Accessed November 11, 2014. <http://blog.robbowley.net/2011/09/21/estimation-is-at-the-root-of-most-software-project-failures/>
- [22] Tools and Techniques for Accurately Estimating BI/DW Projects. Accessed November 11, 2014.
<http://www.damaindiana.org/Presentations/BIProjectEstimating.pdf>
- [23] 7 Signs You Have a Bad Project Estimate. Accessed November 11, 2014.
http://www.officeworksoftware.com/presentations/7_Signs_You_Have_a_Bad_Project_Estimate_1-20-10-PMI.pdf
- [24] Estimates in Software Development. New Frontiers. Accessed November 11, 2014.
<http://www.targetprocess.com/articles/estimates-software-development.html>
- [25] Understanding of Software effort Estimation at the early Software Development of the life cycle - A literature View. Accessed November 11, 2014.
http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Project_Management/Software_Estimation
- [26] Introduction to Software Engineering/Project Management/Software Estimation. Accessed November 11, 2014.
http://www.ijera.com/papers/Vol2_issue1/EI21848852.pdf
- [27] Parametric Models for Effort Estimation for Global Software Development. Accessed November 11, 2014. <http://www.lnse.org/papers/40-IE1011.pdf>

- [28] Cost Estimation. Accessed November 11, 2014.
<http://groups.engin.umd.umich.edu/CIS/course.des/cis375/ppt/lec4.ppt>
- [29] Personal Software Process Software Estimation. Accessed November 11, 2014.
<http://groups.engin.umd.umich.edu/CIS/course.des/cis376/ppt/lec8b.ppt>
- [30] The Comparison of the Software Cost Estimating Methods. Accessed November 11, 2014. <http://www.computing.dcu.ie/~renaat/ca421/LWu1.html>
- [31] Literature Survey On Algorithmic And Non- Algorithmic Models For Software Development Effort Estimation. Accessed November 11, 2014.
<http://ijecs.in/ijecsissue/wp-content/uploads/2013/03/623-632ijecs.pdf>
- [32] Cost Estimation Methods. Accessed November 11, 2014.
<http://www.bignerds.com/papers/78474/Cost-Estimation-Methods/>
- [33] Software Size Estimation Final. Accessed November 11, 2014.
<http://www.informatics.buu.ac.th/~athitha/bangmod/project-estimate/Software%20Size%20Estimation%20%20Final.ppt>
- [34] A Comparison of Software Cost Estimation Methods. Accessed November 11, 2014.
https://www.academia.edu/4447532/A_comparison_of_software_cost_estimation_methods_A_
- [35] The Comparison of the Software Cost Estimating Methods. Accessed November 11, 2014. <http://www.computing.dcu.ie/~renaat/ca421/estimsummary.ppt>
- [36] Putnam, Lawrence H.; Ware Myers (2003). Five core metrics : the intelligence behind successful software management. Dorset House Publishing. ISBN 0-932633-55-2.]
- [37] Putnam, Lawrence H. (1978). "A General Empirical Solution to the Macro Software Sizing and Estimating Problem". IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-4, NO. 4, pp 345-361.
- [38] The Comparison of the Software Cost Estimating Methods. Accessed November 11, 2014. <http://blog.naver.com/PostView.nhn?blogId=kimegoo&logNo=120125588172>
- [39] Software Estimation, Enterprise-Wide. Accessed November 11, 2014.
<http://www.ibm.com/developerworks/rational/library/jun07/temnenco/>
- [40] Calculating-Function-Points. Accessed November 11, 2014.
<http://www.codeproject.com/Articles/18024/Calculating-Function-Points>
- [41] Weighted Micro Function Points (WMFP). Accessed November 11, 2014.
http://www.projectcodemeter.com/cost_estimation/help/GL_wmfp.htm
- [42] Project Code Meter Pro Manual. Accessed November 11, 2014.
http://www.projectcodemeter.com/cost_estimation/help/PCMProManual.pdf

- [43] Empirical Studies of Construction and Application of Use Case Models. Accessed November 11, 2014.
https://www.simula.no/research/se/publications/SE.3.Anda.2003/simula_pdf_file
- [44] Measuring object oriented software with predictive object points. Accessed November 11, 2014.
http://www.researchgate.net/publication/229002349_Measuring_object_oriented_software_with_predictive_object_points
- [45] Function and application points. Accessed November 11, 2014. <http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/Planning/FPs.html>
- [46] Web Points. Accessed November 11, 2014.
<https://www.swalife.com/eipNoAuth/swaLogoutPage/logout.html>
- [47] Data Warehouse Definition. Accessed November 11, 2014.
<http://www.1keydata.com/datawarehousing/data-warehouse-definition.html>
- [48] Data Warehouse & Data Mart. Accessed November 11, 2014.
http://www.learn-datamodeling.com/dw_all.php
- [49] Potential for Agility. Accessed November 11, 2014.
<http://www.cutter.com/research/2007/edge071009.html>
- [50] Multidimensionale. Accessed November 11, 2014.
http://www.elml.uzh.ch/preview/fois/DSSII/de/html/le2_learningObject3.html
- [51] PMT Project Charter. Accessed November 11, 2014.
<http://www.virginia.edu/cio/itpm/documents/PMT-Project%20Charter.docx>
- [52] Making Better Estimates, Part 9: Assumptions & Risk. Accessed November 11, 2014. <http://spin.atomicobject.com/2009/05/07/making-better-estimates-assumptions-risk/>
- [53] SPSS FAQ. What does Cronbach's alpha mean? Accessed November 11, 2014.
<http://www.ats.ucla.edu/stat/spss/faq/alpha.html>
- [54] Paired Difference t-test. Accessed November 11, 2014.
<http://www.cliffsnotes.com/math/statistics/univariate-inferential-tests/paired-difference-t-test>
- [55] Defense.gov News Transcript: DoD News Briefing – Secretary Rumsfeld and Gen. Myers, United States Department of Defense. Accessed November 11, 2014.
<http://www.defense.gov/transcripts/transcript.aspx?transcriptid=2636>
- [56] COCOMO. Accessed November 11, 2014. <http://en.wikipedia.org/wiki/COCOMO>
- [57] COCOMO Online Calculator. Accessed November 12, 2014.
<http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/kutcher/kutcher.html>

- [58] Function Points Worksheet. Accessed November 22, 2014.
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CB4QFjAA&url=http%3A%2F%2Fwww.cs.bsu.edu%2Fhomepages%2Fwmz%2Ffuncpt.xls&ei=ylpxVPHrI2wsATGgYGoDA&usg=AFQjCNFXZVpLTdprTRsLmV7q_5ha543paw&sig2=Os-m5JNPBLuWllpHfvrePw&bvm=bv.80185997,d.eXY
- [59] Software Effort Mis-Under-Estimations (part 1). Accessed November 25, 2014.
<http://psygrammer.com/2011/03/12/misunderestimations-part-1/>

Biographical Information

Hazem Yassin is a software engineer working for a national airline and has eighteen years of wide-ranging experience in architecture, design, project management, portfolio management, demand management, business analysis, and in agile development. Hazem is an expert in developing applications through the entire software lifecycle. His technical expertise includes application development in .Net, Visual Basic, C, C#, C++, VC++, MFC, JAVA, Hibernate, Struts, Oracle 11g, Service Oriented Architecture (SOA), and Web Services. Hazem is also has an ITIL certification and is a Sun Certified Java Programmer for the Java 2 Platform. Additionally, Hazem has earned an MBA and a MS in Computer Science from UTA. On a personal note, Hazem is married to Asma and they are both blessed with 5 amazing children. Hazem is a huge fan of the Dallas Cowboys (but thinks Romo should be benched). Hazem also has many interesting hobbies including racket-ball, gardening, scuba diving, and mountain biking. He can be reached at hazem.yassin@gmail.com.