Modeling and Analysis of Performance of the Steering Angle Sensor Development Project

by

Miguel A. Hurtado

Submitted to the Department of Aeronautics and Astronautics
and to the Sloan School of Management
in partial fulfillment of the requirements for the degrees of

Master of Science in Aeronautics and Astronautics
and
Master of Science in Management

ABSTRACT

In the highly competitive automotive industry, OEM and tier-I suppliers face the problem of determining costs and creating accurate and rapid schedules for current and future product developments. Successful companies in this industry are those that are able to forecast and meet important deadlines, satisfy performance requirements and reduce costs to keep development within budget. But frequently one or two factors are achieved at the expense of the others. Sometimes, for example, suppliers are able to cut costs, but only at the expense of quality. Or they can increase quality at expense of costs. Both scenarios are of concern, especially when competitors are attempting to capture market share.

Engineers and program managers require powerful techniques to have better estimates of completion time versus expenditures. Unfortunately, though, there are not yet such tools available that are capable of incorporating both dimensions of product quality and cost. Moreover, it would be desirable to incorporate product performance with those two dimensions in order to obtain a broad perspective of the entire design.

The main goal of this thesis is the investigation, evaluation and application of the research reported in the Ph.D. thesis "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development" (Browning 1998) in two product platforms of Valeo, Electronics. The two product platforms selected for this purpose were the steering angle sensor (SAS) and the ultrasonic park assist sensor (UPAS). The research for this project was conducted at Valeo, Electronics, located in Bietigheim-Bissingen, Germany.

First, data were collected concerning development costs, timing and performance of the steering angle sensor. Second, the software was modified and applied to obtain a joint probability distribution of cost and schedule for this platform. Third, the model was tracked and validated. The tracking of the model was performed within the same platform by running the software at various times.

The validation of the model consisted of applying the same methodology for the UPAS and other areas of the SAS. Monte Carlo simulation, optimization, design structure matrices, feedback among activities, and concurrency in product development systems, along with three software tools (Visual Basic, Excel and MATLAB) were used extensively in this work.

Finally, the model and the results were presented to the company, with recommendations for future applications.

Thesis Supervisors: John J. Deyst, Jr., Professor of Aeronautics and Astronautics
Steven D. Eppinger, Associate Professor of Management Science and Engineering Systems
ACKNOWLEDGEMENTS

The culmination of this thesis would not have been possible without the help and direction of many people who were always willing to give me their insight and advice.

First of all, I would like to thank my sponsor company, Valeo, for supporting my graduate studies at MIT. It is truly appreciated. I would like to also thank my company supervisor, Rob Arguelles, R&D Director, Sensors and Electronics North America for his continuous support and vision during my studies at MIT. Without his leadership, many of the required tasks could not have been completed on time. In addition, I want to thank Allen K. Schwartz, chief engineer of switches at Valeo for encouraging me to apply to graduate school; he is a key leader in the group and a major supporter of my studies. I also want to thank Dr. Juergen Pirrung and Dr. Heinrich Gotzig for helping me to accomplish research at Valeo Electronics Europe. They have proven to be key persons at the European division. I would like to thank Peter Kofink and Philipp Haas of the steering angle sensor group at Valeo for sharing their expertise and data for the project. They invested extra hours to explain their project and they were always willing to help.

I would like to especially thank my two thesis advisors, Dr. John J. Deyst Jr., Professor of Aeronautics and Astronautics, and Dr. Steven Eppinger, Associate Professor of Management Sciences and Engineering Science at the MIT Sloan School of Management, for sharing their research and allowing the use of their results in my thesis work. Their expertise and direction were invaluable to me in the completion of this work. I would like to thank Dr. Tyson Rodgers Browning for creating the framework and fundamentals of this thesis. I am fully indebted.

I also would like to thank Dr. Don Rosenfield, director of the Leaders for Manufacturing Program (LFM). He has been an irreplaceable person to me during my two years at MIT. He clearly is the right leader for LFM and I am certain that he has had a profound impact on all of us. I would like to thank my colleagues at MIT in the LFM class of 2000 for their friendship and mutual support during my participation in the program. I would like to thank my friends and colleagues at the School of Engineering and the Sloan School of Management for sharing very good moments during and after class.

I also would like to thank my mother, Josefina Urrusquieta, for her teachings and counsels during my life and to my family and friends who have always believed in me.

Finally, I want to thank my lovely wife Belen for her unconditional help and support and for sharing with me, those long sleepless nights at MIT.
# TABLE OF CONTENTS

ABSTRACT ............................................................................................................. 2

ACKNOWLEDGEMENTS ........................................................................................ 3

LIST OF FIGURES ................................................................................................. 6

LIST OF TABLES ...................................................................................................... 8

1. CHAPTER 1: INTRODUCTION AND OVERVIEW .............................................. 9
   1.1 Background .................................................................................................. 9
   1.2 Objective ................................................................................................... 10
   1.3 Thesis overview ......................................................................................... 11

2. CHAPTER 2: ITERATION AND DESIGN STRUCTURE MATRIX
   METHODOLOGY USED IN THE STEERING ANGLE SENSOR PROJECT (SAS) .... 13
   2.1 Iteration in product development .............................................................. 13
   2.2 Concurrent engineering .......................................................................... 14
   2.3 Design Structure Matrix (DSM) methodology .......................................... 14
   2.4 Steering Angle Sensor, brief description .................................................. 15
   2.5 DSM representation of the SAS project ..................................................... 16
   2.6 Advantages and disadvantages of DSM representations .......................... 16

3. CHAPTER 3: COST AND SCHEDULE PLANNING IN COMPLEX PRODUCT
   DEVELOPMENT ................................................................................................. 17
   3.1 Probability density functions for modeling activity duration and cost ....... 17
      3.1.1 Gamma probability density function .................................................... 17
      3.1.2 Beta probability density function ....................................................... 18
      3.1.3 Triangular probability density function .............................................. 19
   3.2 Best PDF for modeling activity duration and cost ..................................... 19
   3.3 Monte Carlo simulation as an alternative to convolution ......................... 20
   3.4 Modeling Process Cost .............................................................................. 22

4. PERFORMANCE AND RISKS IN COMPLEX PRODUCT DEVELOPMENT ........... 23
   4.1 Performance and dimensions of performance in the SAS ......................... 23
   4.2 Multiattribute Utility theory ....................................................................... 25
   4.3 Risk drivers in complex system product development .............................. 27
      4.3.1 Schedule uncertainty and risk ............................................................. 27
      4.3.2 Development cost uncertainty and risk .............................................. 28
      4.3.3 Performance uncertainty and risk ...................................................... 28
      4.3.4 Technology uncertainty and risk ...................................................... 29
      4.3.5 Market uncertainty and risk ............................................................... 30
      4.3.6 Business uncertainty and risk ............................................................ 30
   4.4 Overall performance and performance risk model ..................................... 31

5. TOOLS AND SOFTWARE USED IN THE MODEL .......................................... 33
   5.1 General model description and Excel platform ......................................... 33
      5.1.1 Control panel and simulation data output ........................................... 34
      5.1.2 Dynamic Gantt chart, risk and PM evolution ...................................... 34
      5.1.3 Design Structure Matrix (DSM) module ............................................. 34
5.1.4 Activity Performance Measure and Effects Table (APMET) module.................. 34
5.1.5 Performance Measure data ........................................................................... 34
5.1.6 Utility module............................................................................................... 35

5.2 Visual Basic platform ...................................................................................... 35
5.2.1 Module 1: Cost-schedule planning module .................................................. 35
5.2.2 Module 2: Cost-schedule risk calculation module ........................................ 35
5.2.3 Module 3: Monte Carlo module ................................................................... 35
5.2.4 Module 4: Performance Risk calculation module ......................................... 35
5.2.5 Module 5: Single run performance risk calculation module ....................... 35

5.3 MATLAB platform .......................................................................................... 36
5.3.1 Module 1: Joint probability density module ................................................. 36
5.3.2 Module 2: Joint cumulative probability module ........................................... 36
5.3.3 Module 3: Impact tri-dimensional function module ...................................... 36
5.3.4 Module 4: Convolution module ................................................................... 36

6. SAS AND UPAS DATA AND RESULTS OF THE MODEL................................. 37
6.1 Software development in the SAS as input data for the simulation model ........... 37
6.2 Gantt chart representations of the development ................................................ 37
6.3 DSM representation of the process ................................................................... 38
6.4 APMET representation of the process .............................................................. 40
6.5 Performance Measure initial targets ................................................................. 41
6.6 Performance Measure evolution along the project .......................................... 41
6.7 Utility and expected utility for the SAS project ................................................. 43
6.8 Tracking and validating the model .................................................................... 44
6.9 First results of the SAS project and MATLAB graphs ....................................... 44
6.10 Second results of the SAS project and MATLAB graphs .................................. 49
6.11 Applying the model to the Ultrasonic Park Assist Sensor ............................... 51
6.12 Results obtained in the UPAS project ............................................................ 52
6.13 Comparison between simulated and actual data ............................................. 56

7. CONCLUSIONS ................................................................................................. 57
7.1 Key lessons ....................................................................................................... 57
7.2 Future applications ........................................................................................... 57
7.3 Final remarks ................................................................................................... 58

APPENDIX A. Valeo, an Overview ........................................................................... 59
APPENDIX B. Steering Angle and Ultrasonic Park Assist Sensors ......................... 62
APPENDIX C. SAS Gantt charts ............................................................................. 65
APPENDIX D. Gamma Probability Density Function ............................................ 72
APPENDIX E. Triangular Probability Density Function ......................................... 73
APPENDIX F. Monte Carlo Simulation Vs. Convolution ......................................... 74
APPENDIX G. Risk, a Mathematical Perspective ................................................... 79

ACRONYMS ............................................................................................................. 82
BIBLIOGRAPHY ..................................................................................................... 83
LIST OF FIGURES

Figure 1.1  Product development model ................................................................. 10
Figure 1.2  Characteristics of the model ................................................................. 11
Figure 2.1  A simple product development research cycle ......................................... 13
Figure 2.2  Example of a DSM representation ......................................................... 14
Figure 2.3  Activity information flows and their DSM representation ......................... 15
Figure 2.4  Example of a DSM representation with probability of rework ................. 15
Figure 2.5  DSM representation of software validation in the SAS project ................ 16
Figure 3.1  Gamma density function for several values of α and λ ............................ 18
Figure 3.2  Beta density function ............................................................................. 18
Figure 3.3  Triangular density function .................................................................... 19
Figure 3.4  Triangular and Gamma density functions, hypothetical case .................... 20
Figure 3.5  Triangular distribution, 10,000 samples ................................................. 22
Figure 3.6  Monte Carlo simulation to obtain total duration of three activities modeled as triangular random variables ................................................................. 22
Figure 4.1  Overall product performance as a function of n activities ....................... 23
Figure 4.2  Effect of development cost and schedule in product performance ............ 24
Figure 4.3  Product development performance, customer perspective ...................... 24
Figure 4.4  Single utility curves for six attributes in the SAS project ....................... 25
Figure 4.5  Sources of Schedule Uncertainty ............................................................ 27
Figure 4.6  Sources of Development Cost Uncertainty .............................................. 28
Figure 4.7  Sources of Performance Uncertainty ....................................................... 29
Figure 4.8  Sources of Technology Uncertainty ........................................................ 29
Figure 4.9  Sources of Market Uncertainty ............................................................... 30
Figure 4.10 Sources of Business Uncertainty ........................................................... 30
Figure 4.11 Overall and technical performance risks model ..................................... 31
Figure 4.12 Working on activities will affect performance measures ....................... 32
Figure 6.1  Example of a dynamic Gantt chart obtained with the model ..................... 38
Figure 6.2  DSM representation with probability of rework and iteration ................... 38
Figure 6.3  DSM representation with impacts of rework, LC effects and activity sequencing .... 39
Figure 6.4  Evolution of reusability ......................................................................... 42
Figure 6.5  Evolution of flexibility ........................................................................... 42
Figure 6.6  Evolution of reliability ............................................................................ 42
Figure 6.7  Evolution of correctness ........................................................................ 42
Figure 6.8  Evolution of development cost ............................................................... 42
Figure 6.9  Evolution of development time .................................................................. 42
Figure 6.10 Evolution of overall performance utility .................................................. 43
Figure 6.11 Evolution of technical performance utility .............................................. 43
Figure 6.12 Tri-dimensional histogram of cost-schedule for the SAS project, first data ........................................................................................................... 45
Figure 6.13 Tri-dimensional joint probability density of cost-schedule for the SAS project, first data........................................................................................................... 45
Figure 6.14 View from the back of the joint probability density ................................... 46
Figure 6.15 Top-view of the joint probability density .................................................. 46
Figure 6.16 Marginal probability density of schedule ................................................ 46
Figure 6.17 Marginal probability density of cost ....................................................... 46
Figure 6.18 Contour plot and regression line .............................................................. 47
Figure 6.19 Risk and impact function ....................................................................... 47
Figure 6.20 Cumulative probability of cost and schedule ......................................... 47
Figure 6.21 Contour plots of cumulative probability ................................................... 47
Figure 6.22 Overall performance risk evolution ......................................................... 48
Figure 6.23 Technical performance risk evolution and cumulative cost ..................... 48
Figure 6.24 Reusability performance risk evolution .................................................... 48
Figure 6.25 Reliability performance risk evolution ..................................................... 48
Figure 6.26 DSM representation, SAS project second data ....................................... 49
<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.1</td>
<td>Overall and technical composite utilities for the SAS project</td>
<td>26</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Inputs and data used in the model. Four modules are presented</td>
<td>34</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>WCV, MLV, and BCV estimates for the cost and schedule triangular density functions</td>
<td>39</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>APMET showing effects of activities on performance measures</td>
<td>40</td>
</tr>
<tr>
<td>Table 6.3</td>
<td>APMST showing strengths of changes</td>
<td>41</td>
</tr>
<tr>
<td>Table 6.4</td>
<td>Multiattribute utility table for performance measures</td>
<td>43</td>
</tr>
<tr>
<td>Table 6.5</td>
<td>Simulation cost and schedule data. The model produced 850 Monte Carlo cost-schedule samples</td>
<td>44</td>
</tr>
<tr>
<td>Table 6.6</td>
<td>Probability of finishing the UPAS project, first application, within a certain schedule and cost ranges</td>
<td>53</td>
</tr>
<tr>
<td>Table 6.7</td>
<td>Probability of finishing the UPAS project, second application, within a certain schedule and cost ranges</td>
<td>55</td>
</tr>
<tr>
<td>Table 6.8</td>
<td>Comparison between simulated and actual data</td>
<td>56</td>
</tr>
<tr>
<td>Table B.1</td>
<td>SAS and its applications in the automotive industry</td>
<td>62</td>
</tr>
<tr>
<td>Table B.2</td>
<td>Technical specifications of the SAS</td>
<td>62</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction and Overview

1.1 Background

In today's world, companies in all of the major industries, but especially in the automotive sector, face a great deal of pressure trying to develop high performance products in short periods of time, while reducing development costs. Often, companies are able to meet very stringent deadlines, but at the expense of high incremental costs. Sometimes they keep budget and timing under control, but at the expense of the quality of the product. Sometimes, they develop outstanding products within the agreed timing but at unsustainable costs. Unfortunately, customers want and require suppliers to balance the three dimensions of product development schedule, cost, and performance.

In general terms, successful product development requires the interaction of several disciplines including engineering, research and development (R&D), marketing, quality control and finance. These disciplines must also be accounted for when creating models to help managers and engineers in the development process. Software tools are available to keep track of scheduling and costs, e.g., Microsoft Project and Excel, respectively, but these tools have limitations. As mentioned earlier, customers demand suppliers take into account all the dimensions of product development, and it would be appropriate for suppliers to have a tool that can incorporate those dimensions. If managers and engineers could have a complete picture of the development at every stage with all the variables involved, they would be able to evaluate the risks and to take proper actions to control risks.

Suppliers would like to have better models for product development, but they also would like those models to be precise and accurate from a statistical point of view. For example, when asked to give a completion date for a certain product, the safest estimate for that number could be given. If this product is composed of several parts, each one of them requiring a number of tasks of specific duration, the total time, to complete the product is in rough terms the sum of the duration for each task. Therefore, the better information available to complete each individual task, the better the estimate that can be given to the customer. Consistently safe estimates will produce a biased sum.

In current software applications, the input for activity duration is a single number indicating duration in days, weeks, or months, for example. This number does not represent reality because when completing tasks, activity duration is a range of possible values. The same situation occurs when considering costs, where these are estimates of a range of possible values. The methodology of this thesis will use as input for schedule and cost, probability density functions rather than fixed values.

On the other hand, during actual product development, the current progress towards meeting partial specifications or expectations is constantly monitored, and if progress is not acceptable previous tasks maybe reworked in order to correct the root cause of the problem. The model that will be used in this thesis incorporates recursivity among tasks accounting for possible reiterations, in the event that specifications require more rework.

Another important topic in this thesis is the inclusion of product performance as the third dimension of product development. Product performance is defined as a set of attributes important for the customer. Those attributes are elicited by utility theory. The model will take this aspect into account by considering several dimensions of performance including reliability, price, and timing.

Finally customers and suppliers want to know the risks involved when developing a product. The model used in this work will incorporate risk by evaluating the probabilities of unsuccessful outcomes of product
development. Figure 1.1 shows that product development is the result of the company's corporate strategy, customer needs, competition, marketing, available technology, and manufacturing. Moreover, product development is directly influenced by the available technology, marketing, and manufacturing.

**Figure 1.1 Product Development model**

### 1.2 Objective

The research for this thesis was conducted at Valeo, working specifically at the R&D and product development facility of the electronics group in Bietigheim-Bissingen, Germany. A brief overview of the company can be found in Appendix A. As an automotive supplier, Valeo experiences enormous pressure to develop inexpensive products of the highest quality within tight schedule constraints. Often, Original Equipment Manufacturers (OEM's) require from the group quotes to develop new products or they just simply demand the status of specific products under development.

The main goal of this thesis is to create and test the best available model or methodology for developing products under highly competitive environments. This thesis explores and applies the results and research reported in the Ph.D. thesis, "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development" (Browning 1998) in two product platforms of Valeo. This methodology addresses the main dimensions of product development. Insight will be provided concerning cost and schedule represented as two-dimensional probability density functions for each product platform and recommendations will be offered for future applications at Valeo.
The two products chosen for examination were the steering angle sensor (SAS) and the ultrasonic park assist sensor (UPAS) because of their availability and strategic impact on Valeo Electronics. A brief description for both products is presented in Appendix B.

Several steps were executed during the research. Data concerning development costs, timing, and performance were collected for both platforms. The application software developed in the Ph.D. thesis (Browning 1998) was modified and applied to obtain joint distributions of cost and schedule. The model was tracked and validated. Tracking of the model was performed within the same platform by running the software at different times during and after the investigation, recording key indicators predicted by the model. The methodology was applied to two different platforms of the UPAS to benchmark the results against the real data. Figure 1.2 shows the main assumptions and input data used in the product development model of this Thesis. The model takes into account iteration, feedback, schedule uncertainty, cost uncertainty, and risk drivers in product development. Six different types of risk drivers were considered in this work cost, schedule, performance, technology, market and business risk drivers. The next chapters describe in more detail the model constituents and the main assumptions.

**Figure 1.2** Characteristics of the model
1.3 Thesis overview

The thesis is divided into seven chapters, addressing the most important results and topics of each corresponding category. This chapter offers a general overview of the thesis and the background supporting the methodology. It also considers the main objective of the research and its implications.

Chapter 2 is concerned with iteration and design structure matrix methodology applied to product development. It explains iteration in product development as well as concurrent engineering as a way to optimize development schedule and cost. The chapter also explains the fundamentals of design structure matrix methodology (DSM) describing the advantages and disadvantages of using it in product development and presents a specific example by representing the SAS project using DSM. A brief description of the SAS product is also given in this chapter.

Chapter 3 covers cost and schedule planning in complex product development. It explores ways to model process schedule and cost using probability density functions to represent activity duration and cost. The chapter considers three density functions, beta, gamma and triangular, to represent activity duration and cost, explaining why triangular is preferred over gamma to model such characteristics. Various solution methods are considered when doing PDFs on Monte Carlo simulation as an alternative to convolution and moment generating functions.

Chapter 4 describes a variety of performance measures and risks involved in complex product development. The specific performance measures in the steering angle sensor model are explained with a description of utility theory and its application in this context. The chapter concludes with a review of various types of risks involved in product development and the implications for overall, technical performance and risk for the SAS project.

Chapter 5 describes the tools and software applications used in the thesis, explaining the general model and its implications. The chapter discusses the Excel platform and Visual Basic code, describing its different sections and modules. It explains the dynamic Gantt chart, risk and performance measures evolution, design structure matrix module, the utility module, etc. The chapter ends with a description of the MATLAB m.files joint probability density file, joint cumulative probability file, impact bi-dimensional function file and convolution files.

Chapter 6 explains the data and results of the model obtained for the SAS and UPAS products. It describes, in particular, the sections of the model corresponding to the SAS, defining different parameters for that product. The chapter makes a distinction between tracking and validation of the model in product development and ways to quantify them. It shows the results of the first simulation by using the first data for the SAS and the results of the second simulation with a posteriori information. The chapter also presents the results of the methodology when applied to the UPAS and makes analogies to the results obtained for the SAS.

Chapter 7 presents key lessons of the methodology and future applications for other product platforms at Valeo. Conclusions and final remarks are offered. Future applications and more user-friendly platforms of the method can be developed and used at Valeo.

Appendix A offers a more detailed description of the company. Appendix B describes the SAS and UPAS products. Appendix C contains Gantt charts pertaining to each application of the model. Appendices D and E provide mathematical formulations and results for the gamma and triangular probability density functions, along with a brief summary of their most important characteristics. Appendix F describes Monte Carlo simulation within the product development context as the way it adds probability density functions, comparing simulations with other methods to add probability density functions. Finally, Appendix G offers a mathematical perspective of risk in product development.
Chapter 2

Iteration and Design Structure Matrix
Methodology Used in the Steering Angle Sensor Project (SAS)

2.1 Iteration in product development

Product development consists of a set of activities whose final objective is the creation of a certain product to satisfy specific consumer needs. Product development can be as simple as designing a pen or as complex as creating an airplane. This thesis will deal with complex product development, using two examples drawn from the automotive industry. The importance and role of iteration in their development will be explored.

Complex product development can be decomposed into tasks whose function is the creation of simple subassemblies or simpler parts of the final product. Product development can be perceived as a simple cycle composed of four activities or steps. Observations of the problem or environment are collected, followed by a model representation of the phenomenon. After understanding the model and its implications, the model can be improved with more assumptions or refinements, leading to observation of the phenomenon once again (Figure 2.1).

Figure 2.1 A simple product development research cycle (Browning 1998)

The basic unit of product development can be a single task or activity, but when those activities produce unacceptable subsystems/subassemblies, there are design iterations or rework in the activity. Often iterations come from customer or suppliers in upstream activities who ask for changes in the design. Sometimes, the changes come from downstream activities feeding back to previous activities or from parallel activities performed at the same time. In any case, iterations will occur to satisfy the respective stakeholders. Much of the cycle time variability in product development is caused by iterations in the process. Therefore, a comprehensive understanding and quantification of variability is highly desirable.

Product development can be also thought of as an algorithm where rapid convergence to the right result is desired. In these terms, iteration can be modeled as a loop in the process and the faster the loop finishes the faster a product result. Iterations can be divided in planned or intentional and unplanned or unintentional.

Planned iterations are desirable because the designer or group sets them up to facilitate the flow of information between activities to achieve the goal faster. On the other hand, unplanned iterations come from new requests, or unanticipated situations creating variation in the process. A significant question is whether faster iterations are preferred to slower ones or whether few iterations are preferred to many. Faster iterations are preferred when quality or expectations are met, but fewer iterations seem to indicate poorer performance attributes (Smith 1997). Perhaps the reason for having poorer performance attributes from the
customer perspective is because there is not sufficient interaction or flow of information between activities, creating final products missing important in-between requirements.

2.2 Concurrent engineering

In the previous section iteration has been discussed and it was noted that iteration comes from activities performed at the same time. Concurrent engineering refers to all the activities performed in the same time frame with interaction or flow of information between activities. This characteristic is one the most powerful and useful features in product development because it allows optimization of development time. Although concurrent engineering demands considerable effort to ensure good coordination, the payoffs are usually very high.

Another important feature of concurrent engineering is the inclusion of different areas of expertise within the team, creating a significant positive impact in the overall quality of the product. In the model used in this thesis various examples and outcomes of concurrent engineering modeled by design structure DSMs will be observed and reflected on the dynamic Gantt charts. Although in the short-term, coordination of activities and team members is expensive, the benefits and increase in overall quality usually far surpasses the costs (Browning 1998).

2.3 Design structure matrix (DSM) methodology

Design structure matrix is a methodology used to model process iteration. There are many types of DSM models, but this thesis will concentrate on one special type of matrix, the activity-based or schedule DSM for modeling process schedule based on information flow. In general terms, a DSM is represented by a square matrix of n activities listed in their chronological order of implementation or execution. In a DSM representation, every time there is an interaction between two activities, the square corresponding to their intersection is indicated with a mark. Feed-forward and feedback sequences of activities are indicated by marks below and above the diagonal, respectively. For each activity of the matrix, rows represent required activities for that specific activity, while columns represent subsequent activities requiring information or input from it.

In real situations not all the activities are serial, parallel or coupled, but are combinations of them. Therefore, DSM representations of complex product developments will have combinations of those representations. Figure 2.2 shows an example of an activity-based DSM. In this figure, the two blocks representing highly coupled activities are potential critical paths in the development. Concurrent engineering was previously defined as the interaction between activities, and because DSM graphically represents those interactions, the methodology can be used to understand the general level of interaction and feedback in the development process.

![Figure 2.2 Example of a DSM representation](image-url)
A feature of activity-based DSM is its ability to represent iteration between activities. For instance if there is a pair of activities, the activities would be classified based of their relationship in dependent, independent or parallel, and interdependent or coupled (Figure 2.3, Browning 1998; Eppinger 1991).

![Figure 2.3 Activity information flows and their DSM representation](image)

If it is assumed that a certain activity requires partial input or information from other activities or if activities are involved only to a certain degree, such variation can be represented with a percentage. Also, if working on a certain activity and if this activity produces feedback, it would be good to know the probability of rework for activities requiring its input. Moreover, it would be good to know how much of a certain activity must be reworked in case the feedback occurs. All those variations could be represented with probabilities or rework and impact, replacing marks between the activities with numbers between 0 and 1. Figure 2.4 shows, for example, a hypothetical situation where the marks are replaced with probabilities of rework to indicate the likelihood that activities would have to be reworked to comply with customer requirements.

![Figure 2.4 Example of a DSM representation with probability of rework](image)

2.4 Steering Angle Sensor, brief description

In this thesis, the DSM methodology was analyzed and applied to the Valeo steering angle sensor for two essential reasons. The first reason involved the availability of information and the second reason was the strategic impact in the company. In general terms, the SAS is a human-machine interface for the electronic stability program (ESP) that can determine the intention of the driver at anytime. Because the ESP uses the
SAS, the sensor must have its own security control with very high reliability, using redundancy where appropriate. For more information concerning the SAS, refer to Appendix B, part 1.

2.5 DSM representation of the SAS project

The DSM methodology was applied to software development in the steering angle sensor development project. The representation consists of a DSM with probabilities of rework for 24 activities (Figure 2.5). Data were collected and provided by two design experts and an engineering manager. For more information concerning the gathering of information, refer to chapter 5 in this thesis.

![Figure 2.5 DSM representation of software validation in the SAS project](image)

2.6 Advantages and disadvantages of DSM representations

Design structure matrix representation is a powerful graphic tool because it allows users to have systematic and organized views of product developments. In general, DSMs provide more realistic representations of project scheduling because of their ability to represent feedback and feed-forward relationships between activities. Also DSMs provide ways to represent probabilities of rework and impact between tasks. Another advantage of DSM representations is their capability to summarize graphically complex interactions among tasks and the way they interact.

On the other hand, DSMs can become burdensome and lose its visual advantages when modeling a large number of activities condensed into a single matrix. It is also difficult for users to find and fill correct information into the matrix because it requires good knowledge and a general perspective of the entire project, requiring help from specialists in charge of specific tasks. However, the advantages offered by DSM representations are significant and promising, and more than offset the disadvantage of the representation.
Chapter 3

Cost and Schedule Planning in Complex Product Development

3.1 Probability density functions for modeling activity duration and cost

Activity duration and cost are best modeled with probability density functions because, in real life, duration and cost for activities are random variables and not fixed values. The essential question is what kind of density function best represents activity duration or cost. Activity duration shows a right skewed behavior with a single mode because people have the tendency to expand available time when completing a task, even if they could finish the job earlier (Browning 1998; Kiefer 1998). Beta distributions have been suggested to model this behavior, but the range of the random variable is limited to 0 and 1. A better alternative is the use of gamma distributions that present the same skewed behavior, but with the advantage of any positive value for the underlying random variable. The following sections discuss a number of possible candidate probability densities for representing activity duration.

3.1.1 Gamma probability density function

The Gamma density function depends on two parameters, $\alpha$ and $\lambda$. The mathematical representation of this density is given by the following equation:

$$f(t) = \begin{cases} \frac{\lambda \alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\lambda t} & t \geq 0 \\ 0 & t \leq 0 \end{cases}$$

where the gamma function $\Gamma(\alpha)$ is defined as

$$\Gamma(\alpha) = \int_0^\infty u^{\alpha-1} e^{-u} du$$

The parameter $\alpha$ is called a shape parameter of the gamma density, and $\lambda$ is called a scale parameter. Varying $\alpha$ changes the shape of the density, whereas varying $\lambda$ corresponds to changing the units of measurement. Appendix D presents a more complete description of the gamma PDF. Figure 3.1 shows several gamma densities for a variety of values of $\alpha$ and $\lambda$. It is interesting to note that the gamma density function has an exponential behavior when the parameter $\alpha$ is equal to 1. From the figure, the two curves of the middle represent gamma densities for two different scale parameters $\lambda = 1.25$ and $\lambda = 1$, but equal shape parameter $\alpha = 7$. On the other hand, the curve corresponding to a shape parameter $\alpha = 12$ and scale parameter $\lambda = 1$ is located to the right of the other curves.
3.1.2 Beta probability density function

The Beta density depends on two parameters, \( p \) and \( q \) and it is defined by the following expression

\[
f(t) = B(p, q) t^{p-1} (1-t)^{q-1} \quad 0 \leq t \leq 1 \quad p > 0 \quad q > 0
\]

where by definition

\[
B(p, q) = \int_0^1 t^{p-1} (1-t)^{q-1} dt = \frac{\Gamma(p) \Gamma(q)}{\Gamma(p+q)} = \text{Beta function}
\]

Figure 3.2 shows graphs of the Beta distribution

\[
f_1(t) = 105 x^2 (1-x)^4 \quad f_2(t) = 105 x^4 (1-x)^2
\]
### 3.1.3 Triangular probability density function

This density is defined for positive values, although mathematically, it could also be defined for negative ones. The values $a$, $b$ and $c$ are the shortest, most likely and longest times for this density. The area under the triangular density has to be normalized to one to represent a valid probability density function. Assuming a normalized area, the triangular PDF is defined by the following expression:

$$f(t) = \begin{cases} 0 & \text{for } t < a \\ \frac{2}{(b-a)(c-a)}(t-a) & \text{for } a \leq t < b \\ \frac{-2}{(c-a)(c-b)}(t-c) & \text{for } b \leq t < c \\ 0 & \text{for } t \geq c \end{cases}$$

Appendix E provides a more complete description of the triangular PDF. A representation of it is depicted in Figure 3.3.

[Figure 3.3: Triangular density function]

### 3.2 Best PDF for modeling activity duration and cost

In the previous section, three potential PDFs modeling activity duration were presented. Although the beta PDF can be used to model activity duration, it has the disadvantage of being defined only for values between zero and one. The gamma PDF has the advantage of being defined for any positive value of the parameter but it has a complicated mathematical expression. A better alternative to the gamma PDF is the triangular PDF, which is completely defined by only three parameters in a simple linear relationship. The previous graphs revealed that the gamma and triangular densities present similar statistical behavior as being unimodal and skewed to the right, although the triangular can be skewed to the left. This thesis uses triangular PDF as approximations of gamma density functions.

Both distributions can be superimposed on the same axis calculating the cumulative distribution for a certain range to appreciate the loss by using one distribution over the other. Assume that a certain activity has a best case hypothetical value ($a = \text{bcv}$) of less than one day (i.e., $t = 0$), a most likely value ($b = t_{\text{max}} = \text{mlv}$) of 2 days, and a worst case value ($c = \text{wcv}$) of 6 days. According to the assumptions, the triangular density function corresponding to this case can be constructed and if the area under the density is equal to one, the maximum of the triangular density function is given by

$$f_{\text{Tmax}} = f_{\text{triangular}}(t_{\text{max}}) = \frac{2}{c-a}$$

If it is assumed that the mode of the gamma density is located exactly in this value, the parameters $\alpha$ and $\lambda$ must satisfy the following expressions:

$$\lambda = \frac{\alpha - 1}{t_{\text{max}}}$$

$$\lambda = (\alpha - 1) \alpha^{1 - \alpha} t_{\text{max}}^{\alpha} \Gamma(\alpha) = 0$$
After substituting the values given in the example, and solving numerically the non-linear equation

\[(\alpha - 1) e^{1 - \alpha - \frac{2}{3} \Gamma(\alpha)} = 0\]

The resulted values for \(\alpha\) and \(\lambda\) are 3.953995 and 1.476998, respectively. The result of all of this is shown in Figure 3.4

![Gamma and Triangular distribution](image)

**Figure 3.4** Triangular and Gamma density functions, hypothetical case

If the cumulative density for both functions between 0 and 6 is calculated, the value obtained for the triangular function is "1" due to the normality assumption. But for the gamma function, the value obtained is 0.9777, with a difference between both values of less than 2.4%. In this simple example, the triangular density is an excellent approximation of the gamma density.

Another reason for using the triangular density function is because of the simplicity of its mathematical representation, a characteristic much more appreciated when simulating the density with the Monte Carlo method. In the following section, it will be seen that in order to simulate a random variable with the Monte Carlo method, the cumulative density function of the random variable must be found and then solve the random variable in terms of the cumulative density. Based on the definition of the gamma and triangular densities, trying to perform this operation with the gamma density will require using numerical techniques, whereas doing the same with the triangular distribution will require just some algebraic manipulations.

### 3.3 Monte Carlo simulation as an alternative to convolution

The previous section noted that one of the key problems was adding sampled activity duration and cost coming from different probability density functions to obtain total activity duration and cost of the process. Each activity duration and cost is modeled with a triangular density function. The objective is to obtain the general probability density of duration and cost as a function of individual densities. Generally speaking, the Monte Carlo method consists of taking samples of duration and cost for each activity and adding the values to obtain the total duration and cost for the entire process. This process is repeated many times to arrive at simulated distributions of duration and cost.

If activity duration and cost coming from triangular distributions are to be modeled, random variables that would generate such densities are required. One of the most useful theorems in statistics offers the key to generate random variables with cdf F by applying F⁻¹ to uniform random variables, as long as F⁻¹ can be
easily obtained. In this case, the cumulative density function can be obtained from the triangular density, which by itself will be a uniform random variable. Then, the variable $t$ is solved for, and a triangular distributed random variable $t$ is obtained, generated from the uniform random variable $r$ in the range $[0,1]$. Appendices E and F offer general expressions for the cumulative density function and triangular random variable. The expression for the Cumulative Density Function (CDF) has the following form

$$
\text{CDF}_t = \begin{cases} 
0 & \text{for } t < a \\
\frac{1}{(b-a)(c-a)}(t-a)^2 & \text{for } a \leq t < b \\
1 - \frac{1}{(c-a)(c-b)}(c-t)^2 & \text{for } b \leq t < c \\
1 & \text{for } t \geq c 
\end{cases}
$$

The Cumulative Density Function (CDF) is uniformly distributed between zero and one. The Monte Carlo simulation sample is defined as

$$
t = \begin{cases} 
0 & \text{for } t < a \\
a + \sqrt{r(b-a)(c-a)} & \text{for } a \leq t < b \\
c - \sqrt{(1-r)(c-a)(c-b)} & \text{for } b \leq t < c \\
0 & \text{for } t \geq c 
\end{cases}
$$

Values for $t$ between $a$ and $b$ define the left portion of the triangular PDF. The right portion is defined for values of $t$ between $b$ and $c$. The random variable $r$ is a uniformly distributed random number between zero and one. As an example, suppose a certain activity has a best case value of 2 days, a most likely value of 4 days and worst case value of 8 days. If the Monte Carlo method is used with 10,000 samples, a triangular density function would be obtained that is very close to the theoretical function (Figure 3.5).

![Triangular distribution: Monte Carlo](image)

The Monte Carlo simulation sample equation is

$$
t = \begin{cases} 
0 & \text{for } t < 2 \\
2 + 2\sqrt{3r} & \text{for } 2 \leq t < 4 \\
8 - 2\sqrt{6(1-r)} & \text{for } 4 \leq t < 8 \\
0 & \text{for } t \geq 8 
\end{cases}
$$

where $r$ is uniform in $[0,1]$

Figure 3.5 Triangular distribution, 10,000 samples
Suppose a total duration of three activities is desired, if the simulation is run 10,000 times, the total duration for the final density function will be close to a normal density function because of the central limit theorem. Figure 3.6 shows the Monte Carlo simulation results for the sum of three durations.

Figure 3.6 Monte Carlo simulation to obtain total duration of three activities modeled as triangular random variables

For more information concerning the Monte Carlo method, refer to Appendix F.

3.4 Modeling Process Cost

This thesis considers activity cost to be a function of the time required to accomplish the task, and required resources to execute an activity. It is assumed that required resources are a function of the number of people executing the activity, plus some other factors used by the people or activity. If those assumptions are believed, individual activity cost could be also represented by a triangular PDF similar to the triangular PDF for activity duration, although not necessarily in the same scale. For purposes of proprietary information, data used in the thesis have been modified.
Chapter 4

Performance and Risks in Complex Product Development

4.1 Performance and dimensions of performance in the SAS

Product development creates value by increasing product performance and decreasing product performance risk. Product performance is most easily defined and understood from the customer perspective because it is the customer who decides whether or not the product satisfies the requirements. Another important aspect of product performance is the dimensionality or how many factors have to be considered in its definition and how to measure them.

This thesis uses multiattribute utility theory as a way to quantify the level of satisfaction of the customer. Total product development can be decomposed into a series of activities or tasks that individually, or as a whole, will contribute to the satisfaction of the customer. Every time a certain task is worked on, total performance can be increased or decreased, depending on the way that activity is performed. This incremental behavior corresponds to meeting incremental or partial requirements for the product development. On the other hand, overall product performance could be decomposed in dimensions or attributes important to the customer and important to satisfy overall requirements.

The research reported here concentrated on the development of embedded software for the steering angle sensor project. Performance was evaluated along the following dimensions: reusability, flexibility, reliability, correctness, cost and development time. Reusability is concerned with the number of times the software will be used in future applications; the units for this attribute are the "number of projects". Flexibility measures the number of lines of code that will have to be changed to make the code work in other applications and units for this dimension are "number of code lines changed". Reliability is the number of hours the software application will work without debugging or engineering changes and units of this dimension are "hours". Correctness measures the number of errors per one thousand lines of code and it is expressed in "errors per 1,000 lines of code". Also, customers are always interested in measuring cost of the project and development time. Together these dimensions constitute overall performance for the product development. Individual product development tasks positively or negatively affect the performance (Figure 4.1).

![Figure 4.1 Overall product performance as a function of n activities](image-url)

23
It is important to note that price can be considered a function of development cost, manufacturing cost, and margin, for example, and delivery timing a function of development and manufacturing schedule. In general, gains in product performance increase development schedule and cost but because increasing development cost will increase price and increasing development schedule will increase delivery timing, this will result in a decrease in product performance (Figure 4.2). A very delicate tradeoff exists between how much product performance to increase to maximize the overall performance.

![Diagram 1](image1)

**Figure 4.2** Effect of development cost and schedule in product performance (adapted from Browning 1998)

It is important to pay attention to when to stop increasing technical performance because from the customer perspective, overall performance will start diminishing due to excessive increases in price and delivery timing (Figure 4.3) What matters ultimately is the customer perspective and the market window of opportunity.

![Diagram 2](image2)

**Figure 4.3** Product development performance, customer perspective (adapted from Browning 1998)
4.2 Multiattribute utility theory

Multiattribute utility theory has its foundations in utility theory, which assumes that products and attributes can be measured in terms of the utility or value they provide to customers. Utility can also be defined in terms of the satisfaction a person gets from using a product or by undertaking an activity with that product. Utility theory is widely used in economics, finance, and engineering. In the present context, it was used to elicit overall product performance in product development.

Utility assumes that between two choices or options it can always be decided whether one to the other is preferred or whether there is indifference to both alternatives. This is the completeness axiom. Also, it assumes that if alternative A is preferred to alternative B and if alternative B is preferred to alternative C, then alternative A is preferred to C. This is the transitivity axiom. Building single utility curves requires understanding customer preferences and translating voice-of-the-customer to the utility space. For the steering angle sensor project, several interviews were conducted with program managers and engineers responsible for the product. This yielded utility preferences for six attributes or dimensions: reusability, flexibility, reliability, correctness, cost, and delivery timing. LIB and SIB refer to large-is-better and small-is-better performance measures, respectively. (Figure 4.4)

![Utility Curves for Six Attributes](image)

Figure 4.4 Single utility curves for six attributes in the SAS project
On the other hand, multiattribute utility theory assumes that once utility preferences for single attributes have been constructed, composite utilities can be constructed assuming that single utility preferences are independent of each other, with the composite utility being a function of the six utilities for each attribute. The overall or composite utility has to be normalized between zero and one. Hence the normalized equation for the composite utility is given by the following expression.

\[
U(P) = \frac{\prod_{i=1}^{n} \left( K k_i U(P_i) + 1 \right) - 1}{K}
\]

In this case, \( n \) is equal to the total number of attributes or dimensions being measured. The factor \( k_i \) is the \( i^{th} \) corner point of the hyper-surface in the six-dimensional space corresponding to the value of the overall utility when the \( i^{th} \) attribute is at its maximum, with all of the others at their minimum. In any case, the constant \( K \) is the normalizing factor. From the previous expression, when all of the factors are at their minimum level (i.e. zero), the composite utility is zero. Therefore, \( K \) is calculated for the case in which the factors are at their maximum. Assuming all the utility values for each factor and composite utility are equal to one, an expression is obtained that must be solved in terms of \( K \) to satisfy the normality condition.

\[
K + 1 - \prod_{i=1}^{n} \left( K k_i + 1 \right) = 0
\]

It can be shown that for a six-dimensional space, the equation corresponding to the overall utility reduces to a polynomial of the 5th order, with one real solution and two pairs of complex conjugate solutions. If a similar multiattribute utility function is created for only technical factors, the equation reduces to a polynomial of the 3rd order, with one real solution and one pair of complex conjugate roots. For the SAS project values for different \( k_i \) corresponding to reusability, flexibility, reliability, correctness, price and delivery timing were 0.1, 0.15, 0.2, 0.2, 0.1 and 0.1 respectively calculated. Solving the polynomial equations, the values of 0.467026 and 1.871248 for \( K_{\text{Overall}} \) and \( K_{\text{Technical}} \) respectively, are calculated. Table 4.1 presents the corresponding overall and technical utilities for some combinations of the attributes.

<table>
<thead>
<tr>
<th>( k_i )</th>
<th>Comb 1 Utility</th>
<th>Comb 2 Utility</th>
<th>Comb 3 Utility</th>
<th>Comb 4 Utility</th>
<th>Comb 5 Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusability</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0.3</td>
</tr>
<tr>
<td>Flexibility</td>
<td>0.15</td>
<td>100</td>
<td>0</td>
<td>70</td>
<td>0.45</td>
</tr>
<tr>
<td>Reliability</td>
<td>0.2</td>
<td>15000</td>
<td>0</td>
<td>18000</td>
<td>0.5</td>
</tr>
<tr>
<td>Correctness</td>
<td>0.2</td>
<td>100</td>
<td>0</td>
<td>80</td>
<td>0.2</td>
</tr>
<tr>
<td>Price</td>
<td>0.1</td>
<td>15</td>
<td>0</td>
<td>14</td>
<td>0.1</td>
</tr>
<tr>
<td>Delivery timing</td>
<td>0.1</td>
<td>100</td>
<td>0</td>
<td>80</td>
<td>0.1</td>
</tr>
</tbody>
</table>

| \( U(P_{\text{Technical}}) \) | 0 | 0.276728 | 0.509168 | 0.91427 | 1 |
| \( U(P_{\text{Overall}}) \) | 0 | 0.269194 | 0.550341 | 0.932172 | 1 |

\( K_{\text{Technical}} \) = 1.871248
\( K_{\text{Overall}} \) = 0.467026

Table 4.1 Overall and technical composite utilities for the SAS project

Multiattribute utility theory helps translating the voice-of-the-customer into measurable units for the whole product development team.
4.3 Risks drivers in complex product development

Product development can be viewed as a process of reducing performance risk. As performance risk is reduced other risks, such as cost and schedule risk, are inevitable incurred in the product development process. This section will describe different kinds of risks and uncertainty involved in complex product development (Browning 1999; Brekka 1994).

Product development risk can usually be divided into six categories: schedule risk, development cost risk, performance risk, technology risk, market risk and business risk. There are many more categories, but these categories are considered the most important. Risk is the expected penalty of unacceptable outcomes for the six categories listed earlier.

4.3.1 Schedule uncertainty and risk

Schedule risk can be defined as the uncertainty of achieving specific goals or results on a specified schedule deadline. Causes of schedule uncertainty include intentional and unintentional iterations, activity flexibility, activity set completeness, activity and sub-process length and variance, iteration scope, duration, and available time. Intentional and unintentional iterations between activities create schedule uncertainty, possibly delaying the expected delivery timing. Activity set completeness refers to how good and well specified is the information from the development process, the better and more detailed the specifications, the less the schedule uncertainty in the process. Activity flexibility refers to possible rearrangements of the chronological order of activities to be performed. Activity and sub-process length and variance is concerned with the variation in activity duration or uncertainty in activity duration. Iteration scope and duration refers to the complexity of the project in terms of number of activities and their duration of execution. Available time is concerned with the amount of time available to complete the development. Figure 4.5 shows different causes of schedule uncertainty (adapted from Browning 1998).

![Figure 4.5 Sources of Schedule Uncertainty (Browning 1998)](image-url)
4.3.2 Development cost uncertainty and risk

Development cost risk or non-recurring cost risk is defined as the uncertainty in developing a product within a pre-specified budget. Sources of this risk are cost attentiveness, available budget, quality of budget planning, resource availability, performance uncertainty, schedule rate of change, and schedule uncertainty. Cost attentiveness refers to the ability to monitor costs during the development. Available budget is self-explanatory and the greater the budget the less the cost uncertainty. Quality of budget planning is a key component to keep cost under control. Resource availability is concerned with the tools, the equipment, and human factors necessary to perform the tasks and again the more resources the less the uncertainty. Schedule rate of change refers to the changes of pace while working on specific tasks. If a certain task is being developed at a certain pace, increasing or decreasing that speed will create unnecessary costs for the development. Figure 4.6 presents factors contributing to development cost uncertainty (adapted from Browning 1998).

![Diagram of Development Cost Uncertainty](image)

**Figure 4.6 Sources of Development Cost Uncertainty (Browning 1998)**

4.3.3 Performance uncertainty and risk

Performance risk is defined as the uncertainty of meeting product requirements or specifications. Factors contributing to performance risks include design development and decisions, design evaluation, product complexity, distribution of risk across the system, technology uncertainty, development cost uncertainty and schedule uncertainty. Design development and decisions refers to the amount of interaction among activities and the decisions made to achieve the goal. Design evaluation is concerned with feedback from customers, experts or validations about the current product development. This factor is one of the most important in performance risk reduction. Product complexity is a function of the number of parts or subassemblies of the final product. The more complex the product, the more performance uncertainty there is. Distribution of risk across the system refers to having many low risk subsystems and just a few high risk ones. Of course, the more technology, schedule, and development cost uncertainty, is present, the greater the performance risk. Figure 4.7 summarizes performance uncertainty (adapted from Browning 1998).
4.3.4 Technology uncertainty and risk

Technology risk is defined as the inability of a specific technology to develop a product with the expected customer requirements. Sources of technology risk include availability of substitute technology, technology maturity, technology system coupling and sensitivity, familiarity with technology, reliance on technology supplier and ease of regulatory approval. Availability of substitute technology refers to using alternative technologies to develop a product capable of meeting customer expectations. Technology maturity means that the technology to be used in developing the product has been available for some time, with a good understanding on the part of developers. Technology and system coupling refers to merging a specific technology with a system already in place. This means that the system and the technology have to function properly as a single entity. Familiarity with technology is also important for keeping technology uncertainty under control because the more the designers know about the technology with its intricacies, the less risky the final product will be. Reliance on technology supplier increases the risk because product developers may not be able to control or estimate real capabilities of the offered technology. Finally, increasing regulatory approval releases some pressure when trying to use a technology. Figure 4.8 presents technology uncertainty and its contributing factors (adapted from Browning 1998).
4.3.5 Market uncertainty and risk

Market risk is defined as the possible inability of the product to satisfy a specific market segment. Factors contributing to market risk include, poor market and analysis research, unclear or unstable customer desires, competitor actions, poor specifications and schedule uncertainty. Figure 4.9 presents factors contributing to market uncertainty (adapted from Browning 1998).

Figure 4.9 Sources of Market Uncertainty (Browning 1998)

4.3.6 Business uncertainty and risk

Business risk includes all the other factors not accounted for by the other classes when developing a product. This includes political factors, social factors, economical factors, labor factors, etc. Figure 4.10 shows factors contributing to business uncertainty (adapted from Browning 1998).

Figure 4.10 Sources of Business Uncertainty (Browning 1998)
4.4 Overall performance and performance risk model

In previous sections, different types of risk, in complex product development, have been explained. This section will discuss the model used in this thesis to quantify performance risk. Performance measures are the dimensions in which the overall performance space is divided. They indicate how much customer specifications are being satisfied along the various dimensions of performance space.

In this discussion, it will be very useful to distinguish and classify performance measures according to three types small-is-better (SIB), nominal-is-better (NIB), and large-is-better (LIB). SIB performance means that the smaller the value of the performance measure, the more satisfied and therefore the happier the customer will be with the design. For example, when measuring price, it is obvious that customers will be more satisfied with cheaper products. Also when considering delivery timing usually customers expect suppliers to reduce development time. This is another example of SIB performance measures. NIB performance measures assume that when a specific target exists for a performance measure, the final product is expected to present characteristics very close to that target. For example, when designing a product that has to satisfy a specific tolerance (e.g., internal diameter), the final product is expected to have that final dimension with very little dispersion. LIB performance measures are exactly the opposite of SIB performance measures, i.e., the larger the value of the measure the higher the level of satisfaction of the customer. For example, reliability of a product usually is expected to be high.

Typically customers want to know not only the expected performance measure, but also the uncertainty and ultimately the risk to achieve it. To calculate the risk of a performance measure, a minimum target for that dimension must be provided. Targets of performance measures are chosen and tailored for each application and customer, but they are chosen to differentiate the final product from those of the competitors. Also, targets should be optimal in the sense of obtaining the maximum benefit for the customer at a reasonable expected cost and development time. The model of overall performance risk is summarized in Figure 4.11.

![Overall and technical performance risks model](image)

**Figure 4.11** Overall and technical performance risks model
Another important aspect when calculating performance risk is the assumption that the customer will be satisfied if the design meets specifications within a certain range. This assumption is important because the model assumes that work done on activities will have an impact on the overall performance. Working on specific tasks will have a positive or negative impact on the final behavior of the product, i.e.; some activities will tend to shift the most likely value of performance measures to the left or right of the probability density function. In some instances, working on activities will increase the uncertainty of meeting the specification, having an impact on the dispersion of the density function for the performance measure, usually accounted for by the spread of best and worst case values.

Figure 4.12 shows nine combinations of effects resulting from changes in the mean and dispersion of the densities while working on activities. A nine-digit code (1-9) is used to designate each one of the combinations. Assigning one to the upper left corner, the code increases from left to right and from top to bottom. In the bottom part of each combination, the original PDF of a certain performance measure, before working on an activity, is presented. As a simple example, combination 1 shows that working on the activity will increase the most likely value of the performance measure but with a decrease in the spread of the performance measure values. Similarly, combination 5 shows that working on the activity will have an uncertain impact on both most likely value and dispersion. This coding is used extensively in the APMET (Chapter 5, Section 1.4).

Figure 4.12 Working on activities will affect performance measures
Chapter 5

Tools and Software Used in the Model

5.1 General model description and Excel platform

The main input of the model is the information concerning activity duration and cost, coming from the product development, represented by DSMs and the translation of the voice of the customer into product performance data. One of the main goals of the model is to obtain a joint probability density function of simulated cost and schedule using the Monte Carlo method. The model calculates the risk level of cost, schedule, technical and overall performance with respect to design targets provided by the customer or designer. The model uses utility curves for each performance measure that, combined with the corresponding targets and tolerances, allows the calculation of overall performance of the final product.

The model assumes that activity duration, activity cost, and performance measures can all be modeled with triangular probability density functions as approximations of gamma probability density functions. Definition of probability densities require three estimates, corresponding to the worst case value (WCV), the most likely value (MLV) and the best case value (BCV) of the activity duration, cost, or performance measure. A simulation is performed to obtain combined pairs of cost and schedule for each run of the model. The model makes repeated runs until the mean and variances of cost and schedule stabilize within a predetermined value. Table 5.1 presents a summary of the inputs and units used in the model.

Excel was used as the main software platform for storing input and output data processed by the code written in Visual Basic. MATLAB was used extensively to perform complex mathematical calculations and for tri-dimensional representations of the data. The Visual Basic code has five modules, with several subroutines calculating simulated cost and schedule, performance measures, overall and technical risks, and dynamic Gantt charts. The cost-schedule planning module calculates the simulated cost and schedule outputs, determining which activities have to be worked by individual runs of the algorithm. The same module evaluates performance measure changes, depending on the particular activity in process to be worked. The cost-schedule risk module calculates the cost and schedule risks inherent to the product development process, assuming triangular density functions. The Monte Carlo module performs the Monte Carlo simulation calculation. Again, triangular distributed random variables are sampled and used in the simulation. The performance risk calculation module obtains the performance risks for each dimension or performance measure of the project. The single run performance risk calculation module obtains the performance risks for single runs by using a quadratic impact function and utility curves for each performance measure.

MATLAB code consists of four modules performing complex mathematical computations and tri-dimensional plotting of the data. The joint probability density module obtains the tri-dimensional histogram of cost and schedule using the simulated data of cost and schedule activity. This module performs a bi-cubic interpolation of the data, converting the distribution histogram into a probability density function represented by the tri-dimensional plot of joint probability density of cost and schedule. An optimization algorithm, based on the steepest-ascent-method, is used to calculate the most likely value of the probability density function corresponding to the local-overall-maximum of the graph. Contour plots are also calculated in this module. The joint cumulative probability module calculates the joint cumulative probability density of cost-schedule corresponding to the joint probability of cost-schedule density with appropriate contour plots. The impact tri-dimensional function module obtains a tri-dimensional impact function for the data, with appropriate targets for cost and schedule. The cumulative module performs mathematical convolutions of activity duration and cost modeled with normal or triangular probability densities. This module is used to benchmark the Monte Carlo simulation against the mathematical convolution. This module is not used in the general model of the thesis.
Table 5.1 Inputs and data used in the model. Four modules are presented

5.1.1 Control panel and simulation data output

This section describes the main control parameters necessary to run the model, e.g., the number of activities in the process, the number of performance measures, time step size, performance measure stability and stability batch. This section contains all the output data corresponding to the simulated pairs of cost and schedule and output data corresponding to performance risk.

5.1.2 Dynamic Gantt chart, risk and PM evolution

This section of the model presents dynamic Gantt charts of the process, each corresponding to a different run of the Monte Carlo method. Several charts summarize dynamic performance measures and their evolution along the life of the process data concerning. Overall and technical risk is presented in this section.

5.1.3 Design structure matrix (DSM) module

Design structure matrices representing the process, probability of rework and percentage of impact are shown in this section. Activity duration and cost are also represented in this section, with three estimates corresponding to triangular distributions for each activity. Usually, only two design structure matrices are shown in this section. Learning curve effects are entered into a column for each activity in case of second order rework.

5.1.4 Activity performance measure effects table (APMET) module

This section contains the activity performance measure and effect table (APMET), with the corresponding factors for activities modifying the performance measures. The APMET contains codes indicating the type of change an activity will have on a certain performance measure (Chapter 4 section 4). The codes correspond to the nine possible combinations of most likely value and dispersion for each change, with separate codes indicating targets or initial estimates. The strength factor subsection contains percentages corresponding to three possible impacts when an activity modifies a performance measure. The usual values are 25% for a strong effect, 9% for a medium effect and 4% for a small one, but those values can be modified.

5.1.5 Performance measure data

Performance measures are modeled with triangular density functions. This section contains initial estimates of the worst case value, the most likely value, and the best case value for each dimension or performance measure. This section contains initial cost and schedule estimates of the process.

<table>
<thead>
<tr>
<th>Module</th>
<th>Input Data</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSM</td>
<td>Activities in the process development</td>
<td>Number of activities</td>
</tr>
<tr>
<td></td>
<td>Duration estimates for triangular densities (BCV, MLV, WCV)</td>
<td>Time (i.e. days, weeks,)</td>
</tr>
<tr>
<td></td>
<td>Cost estimates for triangular densities (BCV, MLV, WCV)</td>
<td>Money ($, DM, etc.)</td>
</tr>
<tr>
<td></td>
<td>Learning curve effects for repeating activities</td>
<td>% of original time</td>
</tr>
<tr>
<td></td>
<td>Probability of rework</td>
<td>Number between 0 and 1</td>
</tr>
<tr>
<td></td>
<td>Impact of rework</td>
<td>Percentage</td>
</tr>
<tr>
<td></td>
<td>C&amp;S sampling correlation for each</td>
<td>Correlation coefficient (0,1)</td>
</tr>
<tr>
<td>APMET</td>
<td>Performance measures</td>
<td>Corresponding to individual PM's</td>
</tr>
<tr>
<td></td>
<td>Activity effects on performance measures</td>
<td>Type of effect</td>
</tr>
<tr>
<td></td>
<td>Magnitude of effect on performance</td>
<td>Small, Medium, Large</td>
</tr>
<tr>
<td>Utility</td>
<td>Utility curve and k value for each PM</td>
<td>Utility number between 0 and 1</td>
</tr>
<tr>
<td></td>
<td>Target levels for each performance measure</td>
<td>Depends on individual PM's</td>
</tr>
<tr>
<td></td>
<td>Initial PM estimates (WCV, MLV, BCV)</td>
<td>Depends on individual PM's</td>
</tr>
<tr>
<td>General</td>
<td>Output distribution stability criteria</td>
<td>% of stability</td>
</tr>
<tr>
<td></td>
<td>Run batch size between stability checks</td>
<td>Number of runs</td>
</tr>
<tr>
<td></td>
<td>Simulation Time step size</td>
<td>Time (&lt; smallest activity duration)</td>
</tr>
</tbody>
</table>

34
5.1.6 Utility module

The utility module uses utility curves for each performance measure and input from the performance measure data section to calculate estimates of technical and overall performance. This module uses the calculated values of K and ki as defined in Chapter 4, Section 4. Expected values for overall and technical performance measures are also calculated.

5.2 Visual Basic platform

The code for the model was written in Visual Basic, with some modifications made to adapt data coming from the Steering Angle Sensor and Ultrasonic Park Assist Sensor platforms. The code is divided into five principal modules Cost-schedule planning module (CSP_Model), Cost-schedule risk module (CS_Risk_Cals_Code), Monte Carlo module (Monte_Carlo_Func), Performance risk calculation module (Perf_Risk_Cals_Code), and Single run performance risk calculation module (SR_P_R_Cals_Code).

5.2.1 Module 1: Cost-schedule planning module

This module calculates simulated cost and schedule outputs for the process. Depending on the control panel setting, it can also calculate the technical and overall performance. Initial estimates of the worst case value (WCV), the most likely value (MLV) and the best case value (BCV) are captured for each activity of the process. A similar operation is performed for the performance measures. The model initializes internal variables, assigning values corresponding to probabilities of iteration, rework and the impact of rework, previously stored in the design structure matrices. The APMET, strength values and learning vector are also loaded into the module. The first major step is the resequencing of the activities based on the sequence vector located in the DSM module, this option is very useful when investigating possible effects of different project and activity configurations.

In this module, all of the activities are sampled for schedule and cost by calling the Monte Carlo module. Selection of the activities to be worked during the time step is critical for the model, grouping those activities is determined by the precedence link given by the iteration and 2nd order rework DSM matrix. During the same time step, the model starts building the dynamic Gantt chart and performance measure evolution for the process. This module readjusts the limits and the most likely values of the performance measures, depending on the type of change associated with the activity.

5.2.2 Module 2: Cost-schedule risk module

This module calculates the risks associated with cost and schedule. The risk is calculated multiplying the probability of unacceptable outcomes, determined by the associated targets, by quadratic impact function (Appendix G). Cumulative cost and schedule are calculated after sampling of the probability density functions.

5.2.3 Module 3: Monte Carlo module

This module, mainly called the cost-schedule planning module, provides samples of triangular density functions, determined by the worst case value (WCV), the most likely value (MLV), and the best case value (BCV). Triangular densities are used to model activity cost, schedule and performance measures.

5.2.4 Module 4: Performance risk calculation module

This module calculates the technical and overall performance risk by assuming that each performance measure is modeled by a triangular distributed random variable. Utilities of targets for technical and overall performance are used in the calculation, with quadratic impact functions associated with the corresponding dimensionality constants, converting uncertainty to units of risk. The module calculates cumulative probabilities and ordinates of triangular densities used in the risk formula.

5.2.5 Module 5: Single run performance risk calculation module

This module is very similar to module four, performance risk calculation. The main difference resides in doing the calculation for only single runs of the model.
5.3 MATLAB platform

MATLAB was used extensively in this thesis to perform complex mathematical computations and to produce tri-dimensional graphs of joint probability density functions for cost and schedule. Four main modules were created and used to calculate joint probability densities, cumulative densities, tri-dimensional impact functions and mathematical convolutions. In addition, MATLAB was used to generate tri-dimensional histograms and contour plots of the tri-dimensional graphs. An optimization section based on the steepest ascent method was written in MATLAB to obtain the most likely value of the densities.

5.3.1 Module 1: Joint probability density module

This module uses the simulated cost and schedule outcomes calculated in the Excel platform to generate tri-dimensional joint probability densities for schedule and cost. First, the module obtains the tri-dimensional histogram of schedule and cost. Then, by applying bi-cubic interpolation to this data, the corresponding tri-dimensional density for schedule and cost is obtained. Also, a discrete gradient operator is applied to the density, which, in conjunction with the steepest-ascent method, allows for the calculation of the overall maximum. This optimization method obtains the mode or the most likely value of the tri-dimensional density function.

Average values for cost and schedule are obtained by calculating the center or mass of the distribution. Marginal densities are also obtained for each factor via numerical integration.

5.3.2 Module 2: Joint cumulative probability module

A joint cumulative probability density for cost and schedule is calculated in this module with the corresponding contour plots. The module also provides individual cumulative probabilities for pairs of values for cost and schedule. The module provides a useful view of the probabilities of unacceptable outcomes.

5.3.3 Module 3: Impact tri-dimensional function module

In this section, the tri-dimensional impact function associated with the probability density function for cost and schedule is calculated. The module assumes quadratic impact functions with the corresponding targets for cost and schedule.

5.3.4 Module 4: Convolution module

This module calculates mathematical convolutions for normal and triangular distributed random variables. It is mainly used to benchmark the Monte Carlo method used in the general model. A series of probability density functions are added statistically to the module, providing the final probability densities with the corresponding cumulative plots.
Chapter 6

SAS and UPAS Data and Results of the Model

The Valeo steering angle sensor (SAS) and ultrasonic park assist sensor (UPAS) provided data and research material for the development of this thesis. The thesis had three primarily goals. The first goal was the collection of data concerning costs and schedule for the steering angle sensor and ultrasonic park assist sensor, with special emphasis on the SAS. The second goal was the creation of cost and schedule estimates for the product development process. The model accounts for uncertainty for each activity; and includes feedback, iterations and rework among the activities. The third goal was the validation and tracking of the model, with real data along the product development process. The thesis built a model for performance, including risk and utility functions of six variables cost, schedule and four technical attributes.

The approach consisted of gathering information by meeting with experts and program managers from each platform. A key part of the project was the collection of real data from product managers and engineers responsible of the product and the feedback provided from the team. Finally, the results obtained from the model were compared with the actual product development. The data from the UPAS was used to validate and benchmark the results.

6.1 Software development in the SAS as input data for the simulation model

The main product used in this work was the Valeo steering angle sensor (SAS). This sensor is a human-machine interface for the electronic stability program (ESP) for determining the intention of the driver at any time. For a more complete description of the sensor refer to Appendix B.

The project in software development at Valeo for the SAS consisted of twenty-four activities linked and divided into three phases. Phase zero consisted of all the activities used to check overall design and the creation of the software validation plan. Phase one created the specification for the design. Phase two is the adaptability section for the software development. For proprietary reasons, the details of this process will not be specified. However, the general methodology of the project and the way it was used for the modeling will be described.

6.2 Gantt chart representations of the development

Gantt charts were used mainly for benchmarking purposes and as reference for building design structure matrices. Usually, program managers and product engineers use Gantt charts to build their initial plan, view the schedule, and make adjustments to the plan when developing a product. The chart displays task information in columns, and bar graphs are used to illustrate the duration of each task. However, it is not possible to include iteration or feedback among activities. Appendix C shows the Microsoft project Gantt charts of the process.

In this thesis, dynamic Gantt charts of the product development project were obtained by several runs of the modeling software. The main input data came from the DSMs of the product development project and the three estimates of the worst case value (WCV), the most likely value (MLV), and the best case value (BCV) for each activity duration and cost. Usually, dynamic Gantt charts will have longer duration than typical Gantt charts because of the feedback and iteration among activities. Figure 6.1 depicts an example of dynamic Gantt chart obtained from the simulation of the product development process.
Figure 6.1 Example of a dynamic Gantt chart obtained with the model

The vertical axis of the chart represents the twenty-four activities of the process. The horizontal axis represents duration in days. The end portion of the chart shows isolated squares above downstream activities meaning that some activities have to be reworked due to feedback among activities. There is a major probability of iteration and feedback among the final activities. One of the reasons of such feedback and iteration is because designers at Valeo wanted to check if the process is being developed according to specifications. The longer the loop for iterations, the longer the program will converge to a solution.

6.3 DSM representation of the process

The model used two DSM representations of the process. The first DSM contains probabilities of iteration and rework among activities represented by numbers above and below the main diagonal, respectively. Activities in the process were ordered in a rough chronological sequence (Figure 6.2).

![Figure 6.2 DSM representation with probability of rework and iteration](image-url)
In the same module, three duration estimates of the worst case value (WCV), the most likely value (MLV) and the best case value (BCV) were provided for each activity to build the triangular probability density functions for schedule. The three estimates correspond to the minimum, most likely, and maximum activity duration for the product development process. This information was elicited from experts and program managers responsible for the development and design of the product. The three estimates for the cost triangular density functions were obtained by multiplying the respective estimates for the schedule densities by the number of people involved in the respective activities and by a constant factor accounting for cost per person per day. The constant factor turned out to be approximately equal to $540 US dollars. A correlation factor between schedule and cost for those estimates could also have been used but this option will not be investigated here. In either case, samples coming from the triangular density functions for schedule and cost were generated in the model (TriPDFs) to obtain the simulated total duration and cost with the Monte Carlo method (Table 6.1).

<table>
<thead>
<tr>
<th>Durations</th>
<th>Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriPDF</td>
<td>TriPDF</td>
</tr>
<tr>
<td>Min.</td>
<td>Likely</td>
</tr>
<tr>
<td>6 10 12</td>
<td>8.9</td>
</tr>
<tr>
<td>4 5 6 4.8</td>
<td>1</td>
</tr>
<tr>
<td>4 5 6 5.1</td>
<td>1</td>
</tr>
<tr>
<td>2 3 4 3.2</td>
<td>1</td>
</tr>
<tr>
<td>3 5 6 4.3</td>
<td>1</td>
</tr>
<tr>
<td>8 10 12 8.5</td>
<td>1</td>
</tr>
<tr>
<td>1 2 3 2.4</td>
<td>1</td>
</tr>
<tr>
<td>4 5 6 5.4</td>
<td>1</td>
</tr>
<tr>
<td>10 12 14 12.2</td>
<td>1</td>
</tr>
<tr>
<td>4 5 6 5.8</td>
<td>1</td>
</tr>
<tr>
<td>8 10 12 11.2</td>
<td>1</td>
</tr>
<tr>
<td>5 6 7 5.7</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.1 WCV, MLV, and BCV estimates for the cost and schedule triangular density functions.

The second DSM contains impacts of rework for each activity expressed by numbers between zero and one. On the right hand side of the representation, learning curve effects are presented in a column vector as percentages. A second column vector shows activity sequence ordering, which, in this case, respects the initial ordering proposed by the engineers and program managers (Figure 6.3). The chronological order of execution among activities changes when the activity sequence vector is resequenced. This feature is useful when analyzing effects of ordering among activities.

![Figure 6.3 DSM representation with impacts of rework, LC effects and activity sequencing](image-url)
6.4 APMET representation of the process

The activity performance measures and effect table (APMET) contains possible change effects by the activities on the performance measures estimates for the worst case value (WCV), the most likely value (MLV) and the best case value (BCV). Experts and program managers suggested a division of the overall performance measure into a six-dimensional space, four technical and two non-technical dimensions. The four important technical dimensions to measure were reusability, flexibility, reliability and correctness, whereas the two non-technical performance measures were cost and development time. Reusability measures the number of times the software can be used in other applications. Flexibility measures the number of necessary changes to make the software work. Reliability is measured in effective hours the software will work without failures or changes. Correctness measures the number of errors present in the software per one thousand lines. The two non-technical dimensions completing the set are cost and development time. Table 6.2 reveals that some activities will produce changes on the performance measures limit estimates (WCV, MLV and BCV), while others do not produce any change. Activity sixteen, for example, is concerned with "review of specifications and validation document". According to the table, it will have an impact on the cost with a code of three. That code means that the most likely value (MLV) for cost will increase with an increase in the dispersion of the estimate (refer to Chapter 4, Section 4, Figure 4.12).

<table>
<thead>
<tr>
<th>Activities</th>
<th>Reusability</th>
<th>Flexibility</th>
<th>Reliability</th>
<th>Correctness</th>
<th>Cost</th>
<th>Time development</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Integrity Test</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2 Create SW developmental plan</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3 Create SCM software plan</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Set Coding Rules /Select Tool</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Code for 3, Create codes</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 HS-CAN Controller, SAS Function &amp; Safety</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>7 Cruise control device Function &amp; Safety</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 SPI interface Specification</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 LS-CAN Controller, SAS Function &amp; Safety</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>10 SAS and Cruise control device Validation plan</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Switch process / LS-CAN functions</td>
<td>11</td>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 Diagnosis process, and Diag. Specification date</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>13 Diagnosis Validation plan</td>
<td>13</td>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 Switch Validation plan</td>
<td>14</td>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 Specification for &quot;HW in the loop&quot; Tester</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Review of Specifications &amp; Validation document</td>
<td>16</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 Hardware Software Interface</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 MRSM Assembly &quot;HW in the loop&quot; Box + Software</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 MRSM Hardware (hard-faced Board) for Emulator</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 Software Integration Test Plan</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 Software design file (SA, SD Innovator)</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>22 Software Implementation</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>23 Software Integration Test Plan</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 Software Validation</td>
<td>24</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2 APMET showing effects of activities on performance measures

A second table accompanying the APMET indicates the strength or impact of each change on the performance measures. Three impacts were suggested for this work 25%, 9%, and 4% for strong, medium and low effect, respectively. The effects mean that whenever a change is present, it will have an impact on the limits for the performance measures proportional to the value given by the percentage (Browning 1998). Table 6.3 presents the activity performance measures and strength table (APMST).
As an example, working on activity sixteen will have an impact on cost. According to the APMST, the impact of that change will be strong and the limits for that performance measure will change 25%. Transforming equations change limits for the performance, but they will not be discussed in this thesis (refer to Browning 1998).

6.5 Performance Measure initial targets

Initial estimates for the worst case value (WCV), the most likely value (MLV), and the best case value (BCV) for each performance measure in the SAS project are located in the "performance measure data" section on the Excel spreadsheet. Values for those estimates were collected from the SAS design and product development group. The three initial estimates for reusability were 1, 5 and 10 projects, respectively. This performance measure is defined as a large-is-better (LIB) dimension. Estimates for flexibility were 0, 50 and 100 changes, assuming a small-is-better (SIB) dimension. The corresponding values for reliability were 18,000, 20,000 and 22,000 hours for a large-is-better (LIB) performance measure. Finally estimates for correctness were 30, 50 and 100 errors per one thousand lines. This performance measure is defined as a small-is-better dimension (SIM). Performance measures initial estimates are modified, depending on an activity modifying the dimension. The model presents a simulated overall performance for the whole process.

6.6 Performance Measure evolution along the project

The model allows for the calculation of the most likely value (MLV) and the expected values (EV) for each performance measure and their variation along the project life. Working activities modify the limits for performance measures defined by the APMET and APMST tables. Therefore, it would be expected that working on those activities would change the limits of performance. In the model, two values are calculated for each performance measure, the most likely value (MLV) and the expected value (EV). During the first stages of the process, both values would be expected to be equal, but after working on the activities, both values would diverge. Also, both values would be expected to be very close at the end of the project, with very little variability in their estimates. Figures 6.4 to 6-9 portray the evolution of the six performance measures included in the project.
In Figure 6.4, the dark stepwise constant line represents the most likely value (MLV) for reusability, while the light stepwise constant line represents the expected value (EV) for the same performance measure. At the beginning of the project, both values are very close, but as the project evolves, changes start to show up on the graph. Also, the spread of the predictions is very high in early stages, but as the project finishes up, the variation of the estimates decreases. Vertical error bars represent the variation of the predicted estimates for the performance measures. Similarly, the evolution for the overall and technical performance utility along the project life can be obtained (Figures 6.10 and 6.11).
6.7 Utility and expected utility for the SAS project

The utility module provides expected values (EV) and estimates of limits for the six performance measures. A transformation from the performance measure space to the utility space is performed with help of the utility curves for each dimension. The worst case value (WCV), the most likely value (MLV) and the best case value (BCV) are calculated for each performance measure with the respective utility values. Table 6.4 presents the details of calculations of the overall performance measure for the system.

<table>
<thead>
<tr>
<th>PM</th>
<th>E[values]</th>
<th>Targets</th>
<th>k_j</th>
<th>K = 0.467</th>
<th>KTech = 1.871</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusability</td>
<td>5.33</td>
<td>0.81</td>
<td>7</td>
<td>0.90</td>
<td>1.038</td>
</tr>
<tr>
<td>Flexibility</td>
<td>50</td>
<td>0.65</td>
<td>25</td>
<td>0.78</td>
<td>0.15</td>
</tr>
<tr>
<td>Reliability</td>
<td>20000</td>
<td>0.85</td>
<td>21000</td>
<td>0.88</td>
<td>0.2</td>
</tr>
<tr>
<td>Correctness</td>
<td>0.45</td>
<td>35</td>
<td>0.83</td>
<td>0.2</td>
<td>1.042</td>
</tr>
<tr>
<td>Price</td>
<td>12.9</td>
<td>0.21</td>
<td>12.68</td>
<td>0.24</td>
<td>0.1</td>
</tr>
<tr>
<td>Delivery Timing</td>
<td>45.2</td>
<td>0.88</td>
<td>33.00</td>
<td>0.94</td>
<td>0.1</td>
</tr>
</tbody>
</table>

\[ U(P) | 0.61 \quad U(P_{Tech}) | 0.59 \quad 0.79 \]

\[ U(P_{Tech}) \]

\[ E[U(P)] \quad E[U(P_{Tech})] \]

\[ E[U(P)] \quad E[U(P_{Tech})] \]

<table>
<thead>
<tr>
<th>PM</th>
<th>WCVs</th>
<th>MLVs</th>
<th>BCVs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusability</td>
<td>1</td>
<td>0.16</td>
<td>5</td>
</tr>
<tr>
<td>Flexibility</td>
<td>100</td>
<td>0.00</td>
<td>50</td>
</tr>
<tr>
<td>Reliability</td>
<td>18000</td>
<td>0.51</td>
<td>20000</td>
</tr>
<tr>
<td>Correctness</td>
<td>100.00</td>
<td>0.00</td>
<td>50.00</td>
</tr>
<tr>
<td>Price</td>
<td>13.07</td>
<td>0.20</td>
<td>12.9</td>
</tr>
<tr>
<td>Delivery Timing</td>
<td>58.2</td>
<td>0.81</td>
<td>42</td>
</tr>
</tbody>
</table>

\[ U(P) | 0.23 \quad 0.64 \quad 0.83 \quad 0.567 \quad E[U(P)] \]

\[ U(P_{Tech}) | 0.12 \quad 0.64 \quad 0.92 \quad 0.560 \quad E[U(P_{Tech})] \]

<table>
<thead>
<tr>
<th>WCVs</th>
<th>MLVs</th>
<th>BCVs</th>
</tr>
</thead>
</table>

\[ (intermediate values) \]

\[ 1.191 \quad 1.318 \quad 1.341 \]

\[ 1.000 \quad 1.225 \quad 1.337 \]

**Table 6.4** Multiattribute utility table for performance measures.

The last two rows in the table present utilities for overall and technical performance corresponding to the three cases WCV, MLV and BCV. For overall performance, 0.23, 0.64 and 0.83 in utility units are obtained for WCV, MLV and BCV, respectively. Although the value is very high for the best case situation, the value for the worst case is very low, i.e., there is a big spread of utility.
6.8 Tracking and validating the model

Tracking the model in a product development context means comparing periodically the predicted results (i.e., cost and schedule joint probability density function) against the real evolution of the project in terms of cost, schedule or performance measures. Validating the model, on the other hand, means applying the same methodology to other projects and measuring the accuracy of the predictions against real data.

As for tracking in the steering angle sensor project, data were collected at two different dates and the model was run two times, comparing and recording the results for predictions against real data. For validation, the model was employed in two projects of the ultrasonic park assist sensor. The results of the prediction were compared with the real data. The results for tracking and validation are presented in the next sections.

It is expected that the model used in this thesis be capable of tracking projects with accuracy if it is to be used in highly competitive environments, where just a slightly edge could make a significant difference in winning a contract or in keeping a customer satisfied.

6.9 First results of the SAS project and MATLAB graphs

This section will present a more graphical representation of the simulated results obtained from the Monte Carlo method embedded in the model. Tri-dimensional representations of the joint probability density function for cost and schedule will also be displayed.

The basic data to build joint densities are Monte Carlo samples of cost and schedule obtained by different runs of the simulation model. For the steering angle sensor project, the model produced 850 Monte Carlo cost-schedule samples, which were used as input for MATLAB to build the tri-dimensional densities. See Chapter 5, Section 5.3 for a description of the modules. Table 6.5 presents an extract of the data used to build the joint density plots. For each run of the model, the columns C and S represent, respectively, cost in thousands of dollars and schedule in days for the product development.

<table>
<thead>
<tr>
<th>Run #</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>165</td>
<td>128.25</td>
</tr>
<tr>
<td>2</td>
<td>151</td>
<td>116.25</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>124.5</td>
</tr>
<tr>
<td>4</td>
<td>160</td>
<td>136.5</td>
</tr>
<tr>
<td>5</td>
<td>152</td>
<td>130.5</td>
</tr>
<tr>
<td>6</td>
<td>154</td>
<td>123</td>
</tr>
<tr>
<td>7</td>
<td>187</td>
<td>143.25</td>
</tr>
<tr>
<td>8</td>
<td>146</td>
<td>129</td>
</tr>
<tr>
<td>9</td>
<td>153</td>
<td>120</td>
</tr>
<tr>
<td>10</td>
<td>190</td>
<td>153.75</td>
</tr>
</tbody>
</table>

Table 6.5 Simulation cost and schedule data. The model produced 850 Monte Carlo cost-schedule samples.

After using the Monte Carlo cost-schedule samples, the tri-dimensional histogram shown in Figure 6.12, was obtained, clearly showing unimodal behavior, with a dominant local maximum or most likely value (MLV). The data were also interpolated in a 100 by 100 grid array, generating the probability density function of Figure 6.13. The plots are skewed towards high cost and longer schedule, meaning that there are instances where the project is completed late resulting in higher costs for the group, although the probability of those outcomes is very small. From the tri-dimensional plots, the probabilities of cost-schedule outcomes can be calculated by simply integrating the volume under the surface.
Figure 6.12 Tri-dimensional histogram of cost-schedule for the SAS project, first data

Figure 6.13 Tri-dimensional joint probability density of cost-schedule for the SAS project, first data

From the figures, it can be seen that the most likely value (MLV) is located approximately at 120 days and $145K, and the average value is located at 130 days and $160K. The average value differs from the most likely value because it is the center of mass of the distribution, whereas the most likely value (MLV) is the local maximum or mode. Figures 6.14 and 6.15 present two other views of the joint-density.
To observe just the schedule probability density function, it would be necessary to integrate the joint density function over all possible values of cost. This operation is equivalent to integrating the volume over the cost. A similar operation would be performed if interested in costs (Figures 6.16 and 6.17).

A contour plot of the distribution is presented in Figure 6.18, with the regression line of the data. The steepest ascent method was used to obtain the most likely value (MLV) of the joint-density.

The risk is a function of the probability of unacceptable outcomes and the consequences of those outcomes, provided a target is specified. Using definitions of probability of unacceptable outcomes and impact function, risk can be mathematically defined as (see Appendix G)

$$\text{Risk} = k_s \int_{T_s}^{\infty} (x - T_s)^2 f_s(x) \, dx$$
Extending the definition to the joint density function, the risk can be calculated by defining a tri-dimensional impact function resulting in the graph of Figure 6.19.

Figure 6.18 Contour plot and regression line  Figure 6.19 Risk and Impact function

An interesting application of the tri-dimensional probability density function would be the calculation of the probability to complete a project within a certain budget and within a certain allotted time. The tri-dimensional cumulative probability (Figure 6.20) corresponding to the tri-dimensional probability density could be calculated. This cumulative plot can be used to calculate the probabilities of combination cost-schedule. Perhaps, managers and engineers would want to know the probability of finishing the project in the range of 100-140 days and $120K-$160K. This probability is easily calculated from the cumulative plot, specifically from the contour plot (Figure 6.21). The value of the probability under these conditions is approximately equal to 0.16. Calculation of the cumulative probability is obtained by integrating the tri-dimensional probability density function from minus infinity to the different values of schedule and cost. This operation is similar to the bi-dimensional case.

Figure 6.20 Cumulative probability of cost and schedule  Figure 6.21 Contour plots of cumulative probability

One of the many advantages of having probability density functions is the facility to obtain probabilities of success in terms of meeting an agreed schedule or budget. The next section will discuss the results obtained for the steering angle sensor, second data.
Another interesting chart that explains the behavior of the process is obtained by plotting the risk along the project's life. In the model, the definition of risk as explained in Appendix G is used. The risks for the six performance measures are plotted, along with the overall and technical risk of the project. Figures 6.22 and 6.23 show risk evolution for overall and technical performance, respectively, along the project's life.

Figure 6.22 Overall performance risk evolution

Figure 6.23 Technical performance risk evolution and cumulative cost

Figure 6.22 reveals that overall performance risk remains virtually the same through the project's life, but with a significant decrease in risk during the final stages. Technical risk reveals a similar behavior. But it is interesting to note that cumulative cost increases and it will intersect technical risk. Both vertical scales are different but the technical risk can be continually decreased at the expense of higher cumulative costs. On the other hand, the risk behavior for the six performance measures can also be shown. As an example, the behavior of reusability and reliability performance risks can be investigated (Figures 6.24 and 6.25).

Reusability performance risk decreases over time, which is expected because reusability was defined as the number of potential projects using the same code. At the end of the project, it will be known for certain what projects would be able to use the written code because the code's capabilities will be known. With respect to reliability, the risk goes up but at the end it goes down. If the APMET data are examined, the risk goes up because some activities have a negative impact on reliability (e.g., activities 6, 9, and 12) But because of the final validation in the project, risk performance goes down at the end of the project. Similar behaviors are observed in the remaining performance measures but the general trend is a decrease of performance risk at the end of the product development cycle, unless something unusual in the product development occurs. However, this would contradict the assumption that product development reduces overall and technical performance risk.
6.10 Second results of the SAS project and MATLAB graphs

After running the simulation with the first data and comparing the results with the real ongoing project, designers and program managers provided data for the second simulation to track the project's behavior, predicted by the model. The data collection for the second time was done approximately after three months of running the first simulation. Less variation was anticipated in the second simulation because the designers understood many activities. The new DSM representation consisted of only 19 activities versus 24 activities for the first data. Cost per person per day was assumed to be the same, and the probabilities of rework, feedback and second order iterations were validated with Valeo designers (Figure 6.26).

![Figure 6.26 DSM representation, SAS project second data]

Less feedback among activities was present due mainly to the reduction and rearrangement of activities decided by the product development team. At this stage, they thought the process could be accomplished with fewer activities and less uncertainty. When a project is reaching the final stages, most of the uncertainty disappears because the process involves past and already accomplished tasks. Moreover, the team has a comprehensive understanding of some of the open issues that occurred at the early stages of the project. Concerning the dynamic Gantt chart, less iteration is expected, resulting in a smaller total duration of the project (Figure 6.27).

![Figure 6.27 Dynamic Gantt chart, SAS project second data]
Evolutions of the overall and technical performance utilities are depicted in Figures 6.28 and 6.29, respectively. There is a slight increase in both utilities along the project, and the gap between the most likely and the expected utilities is smaller than that of the previous simulation with the first data (Figures 6.10 and 6.11). The light stepwise constant line represents the utility expected value, while the dark stepwise constant line represents the utility most likely value (MLV).

![Overall performance utility evolution](image)

*Figure 6.28 Overall performance utility evolution*

![Technical performance utility evolution](image)

*Figure 6.29 Technical performance utility evolution*

Similar results for overall and technical performance risks are exhibited in Figures 6.30 and 6.31. If the graphs in those figures are compared to those of Figures 6.22 and 6.23, it can be seen that the cumulative cost goes down. If the work continued, the risks involved for both performance measures would also decrease.

![Overall performance risk evolution](image)

*Figure 6.30 Overall performance risk evolution*

![Technical performance risk evolution](image)

*Figure 6.31 Technical performance risk evolution*

The performance measure risk evolution for the six performance measures defined in the SAS project can also be demonstrated. As an example, the evolution of risk for reusability and reliability is presented in Figures 6.32 and 6.33. A reduction in reusability and reliability risks with the second data is observed.

![Reusability performance risk evolution](image)

*Figure 6.32 Reusability performance risk evolution*

![Reliability performance risk evolution](image)

*Figure 6.33 Reliability performance risk evolution*
Finally, the tri-dimensional histogram and joint probability density corresponding to the SAS project, second data are presented in Figures 6.34 and 6.35. The histogram consisted of a 10-by-10 array generated with 350 Monte Carlo cost-schedule samples from the simulation. Because of less iteration in the second data, the number of samples generated decreased from 800 samples for the first data. In this case, the tri-dimensional probability density was generated by bi-cubic interpolation of the data in a 100-by-100 grid.

![Figure 6.34 3D histogram, SAS project 2nd data](image1)

![Figure 6.35 3D PDF, SAS project 2nd data](image2)

Symmetry of the tri-dimensional probability density function and the histogram around the most likely value (MLV) suggests less iteration in the data. Also, finishing the project late with high expenditures is very unlikely according to the data. The probability of such an outcome would be very small.

Simulation results for both data in the SAS project suggest that the methodology employed in this thesis could be used as a mean to track product development for the entire project. The feedback received from project managers and engineers about the simulation was positive. They were satisfied with the overall results and tri-dimensional representation of their work. The next section will present the results obtained when applying the methodology to the UPAS project.

### 6.11 Applying the model to the ultrasonic park assist sensor

The simulation model was applied twice in the UPAS platform. The first application was for predicting schedule and cost of increasing current production volume capacity per year. The second application was for predicting schedule and cost of producing the fourth generation of ultrasonic sensors. The methodology was applied to the UPAS platform because of interest in validating the model, and most importantly, to assess the level of prediction in terms of cost and schedule in product development.

DSM methodology was used as input data for the model. Information regarding feedback, rework, and impact of rework for both applications was collected. Costs per person per year were assumed to be equal to those of the SAS project. The simulation model was benchmarked against more traditional approaches such as Microsoft Gantt charts. The time step used for simulating the first application was 5 days or 22% of total duration for the shortest activity, while the time step for the second one was 1 day or 8% of total duration for the shortest activity.

Data required to predict product performance measures and risks in product development were not used in this case. The next section presents a summary of the results obtained in the UPAS. Appendix C presents the corresponding Microsoft Gantt charts for both projects.
6.12 Results obtained in the UPAS project

The first charts to be presented are the DSM representations of the UPAS project (Figure 6.36). In this case, there are ten activities to model, with probabilities of feedback and rework represented in the first DSM. The second DSM representation contains the impacts of rework and potential learning curve effects. Three estimates of duration for each of the ten activities were collected from designers minimum, most likely and maximum duration. These were used to build triangular density functions for the random samples used as input to the Monte Carlo simulation. Information for cost was obtained by multiplying the respective durations by the resources needed for each activity and by a constant factor. This constant factor represents the cost per person per day, which was assumed to be the same as in the SAS project.

When testing alternative configurations, just paste in new DSM planes and change activity sequence below. Do not resequence APRT, durations, costs, LC, or work vectors.

<table>
<thead>
<tr>
<th>Dimension k = 1 (Rework Probabilities)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
</tr>
<tr>
<td>-------------------------------------</td>
</tr>
<tr>
<td>1 Cleaning unit (for glue process)</td>
</tr>
<tr>
<td>2 Glue machine 2</td>
</tr>
<tr>
<td>3 Curing oven for splice</td>
</tr>
<tr>
<td>4 Assembly 2. Pre-assembly line (manual)</td>
</tr>
<tr>
<td>5 Cleaning unit (for foaming process)</td>
</tr>
<tr>
<td>6 2K-foam tool for Backing Material</td>
</tr>
<tr>
<td>7 Curing oven for Backing Material</td>
</tr>
<tr>
<td>8 Assembly unit for padding and cover</td>
</tr>
<tr>
<td>9 Pallets system</td>
</tr>
<tr>
<td>10 Burn-in</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimension k = 2 (Rework Impacts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>1 Cleaning unit (for glue process)</td>
</tr>
<tr>
<td>2 Glue machine 2</td>
</tr>
<tr>
<td>3 Curing oven for splice</td>
</tr>
<tr>
<td>4 Assembly 2. Pre-assembly line (manual)</td>
</tr>
<tr>
<td>5 Cleaning unit (for foaming process)</td>
</tr>
<tr>
<td>6 2K-foam tool for Backing Material</td>
</tr>
<tr>
<td>7 Curing oven for Backing Material</td>
</tr>
<tr>
<td>8 Assembly unit for padding and cover</td>
</tr>
<tr>
<td>9 Pallets system</td>
</tr>
<tr>
<td>10 Burn-in</td>
</tr>
</tbody>
</table>

An example of the interactive Gantt chart produced by the simulation model is presented in Figure 6.37. There are similarities between this representation and the typical Microsoft Gantt chart (Appendix C) but the two approaches have some fundamental differences. The main difference is that rework is accounted for in the Monte Carlo simulation. While working on activities five, six and seven, there is a return to rework on tasks three and four, stopping momentarily those activities in order to complete activity three. At the end of the project, some rework activity shows up on tasks eight, nine and ten represented by horizontal bars.

![Figure 6.36 DSM representation of the UPAS project, first application](image)

![Figure 6.37 Dynamic Gantt chart representation of the UPAS project, first application](image)
The tri-dimensional histogram and tri-dimensional probability density obtained from 325 Monte Carlo cost-schedule samples generated by the model are presented in Figures 6.38 and 6.39.

Based on the tri-dimensional representations and the MATLAB calculations, the most likely and average values for schedule-cost were calculated to be 395 days with a cost of $698, and 428 days with a cost of $730, respectively. The average value (AV) is different from the most likely (MLV) because the former represents the center of gravity of the density, while the latter represents the local maximum of the distribution. Similarly, the probability of finishing the project can be calculated within a certain schedule and cost ranges. Table 6.6 presents a summary of the results.

<table>
<thead>
<tr>
<th>Min schedule days</th>
<th>Max schedule days</th>
<th>Min cost $thousands</th>
<th>Max cost $thousands</th>
<th>Probability</th>
<th>Cumulative Probability*</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>420</td>
<td>690</td>
<td>710</td>
<td>0.04</td>
<td>0.27</td>
</tr>
<tr>
<td>375</td>
<td>445</td>
<td>665</td>
<td>735</td>
<td>0.31</td>
<td>0.48</td>
</tr>
<tr>
<td>350</td>
<td>470</td>
<td>640</td>
<td>760</td>
<td>0.56</td>
<td>0.60</td>
</tr>
<tr>
<td>325</td>
<td>495</td>
<td>615</td>
<td>785</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>300</td>
<td>520</td>
<td>590</td>
<td>810</td>
<td>0.80</td>
<td>0.81</td>
</tr>
<tr>
<td>275</td>
<td>545</td>
<td>565</td>
<td>835</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td>250</td>
<td>570</td>
<td>540</td>
<td>860</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>225</td>
<td>595</td>
<td>515</td>
<td>885</td>
<td>0.96</td>
<td>0.96</td>
</tr>
</tbody>
</table>

* Cumulative probability uses maximum values for schedule and cost as upper limits

Table 6.6 Probability of finishing the UPAS project, first application, within a certain schedule and cost ranges

Because of the significant variation in the data, completing the project in 400 to 420 days and with a cost of $690-$710 is very unlikely. It will happen only 4% of the time. Moreover, the probability of completing the project in less than 420 days with or out of less than $720 is 0.27. However, finishing the project in 350 to 470 days and with a cost of $640-$760 will happen approximately 56% of the time (Table 6.6). The corresponding probability of completing the project in less than 470 days with a cost of less than $760 is 0.6. These results provide some insight into how to keep schedules and budgets within planned ranges by are simply reducing schedule variation for individual tasks to a minimum.

Results for the second simulation in the UPAS are similar to the ones just presented.
DSM representations of the second UPAS project consist of seventeen activities modeled with probabilities of feedback, rework, impacts of rework and potential learning curve effects. Three estimates of duration for each one of the seventeen activities were collected the minimum, the most likely, and the maximum duration. These were used to build triangular density functions with the Monte Carlo method. Information for cost was obtained by multiplying the respective duration by the resources needed for each activity, given by the number of people involved and by the cost per person per day (Figure 6.40).

One of the many interactive Gantt charts produced by the simulation model is presented in Figure 6.41. This representation and the typical Microsoft Gantt chart (Appendix C) have some similarities. The main difference is the level of feedback and iteration accounted for in the dynamic model. Small horizontal bars interrupting the flow of activities represent feedback and rework among activities. According to this, feedback activity is present at both the beginning and end of the project.
The tri-dimensional histogram and tri-dimensional probability density for this project were obtained from 400 Monte Carlo cost-schedule samples generated by the model (Figures 6.42 and 6.43).

From the tri-dimensional representations and MATLAB calculations the most likely and the average values (AV) for schedule-cost were calculated to be 463 days with a cost of $557, and 467 days with a cost of $566, respectively. The probability of finishing the project can be calculated within a certain schedule and cost ranges. Table 6.7 summarizes the results.

<table>
<thead>
<tr>
<th>Min schedule days</th>
<th>Max schedule days</th>
<th>Min cost $thousands</th>
<th>Max cost $thousands</th>
<th>Probability</th>
<th>Cumulative Probability*</th>
</tr>
</thead>
<tbody>
<tr>
<td>460</td>
<td>465</td>
<td>555</td>
<td>560</td>
<td>0.03</td>
<td>0.29</td>
</tr>
<tr>
<td>455</td>
<td>470</td>
<td>550</td>
<td>565</td>
<td>0.20</td>
<td>0.43</td>
</tr>
<tr>
<td>450</td>
<td>475</td>
<td>545</td>
<td>570</td>
<td>0.39</td>
<td>0.55</td>
</tr>
<tr>
<td>445</td>
<td>480</td>
<td>540</td>
<td>575</td>
<td>0.57</td>
<td>0.65</td>
</tr>
<tr>
<td>440</td>
<td>485</td>
<td>535</td>
<td>580</td>
<td>0.72</td>
<td>0.75</td>
</tr>
<tr>
<td>435</td>
<td>490</td>
<td>530</td>
<td>585</td>
<td>0.83</td>
<td>0.84</td>
</tr>
<tr>
<td>430</td>
<td>495</td>
<td>525</td>
<td>590</td>
<td>0.89</td>
<td>0.89</td>
</tr>
<tr>
<td>425</td>
<td>500</td>
<td>520</td>
<td>595</td>
<td>0.93</td>
<td>0.94</td>
</tr>
</tbody>
</table>

* Cumulative probability uses maximum values for schedule and cost as upper limits

Table 6.7 Probability of finishing the UPAS project, second application, within a certain schedule and cost ranges

Finishing the project in 460 to 465 days and with a cost of $555-$560 is very unlikely. It happens only 3% of the time, whereas finishing the project in 445 to 480 days and with a cost of $540-$575 happens approximately 57% of the time. The corresponding probability of finishing the project in less than 480 days with a cost of less than $575 is 0.65. The table reveals that almost the entire population of outcomes falls between 425 to 500 days with a cost of $520-$595, with a probability of 93%. To reduce such dispersion in the data, it will be necessary to reduce individual task variation to a minimum.

This chapter describes the most important results of the simulation and explains various ways to interpret the data. In summary, the following information was presented for the SAS and UPAS projects DSM representations, dynamic Gantt charts, performance measures, utility of performances, performance risks and tri-dimensional representations of the Monte Carlo schedule-cost simulation sample with MATLAB graphs. Some possible applications for program managers and engineers were also presented. But most importantly, a new methodology to perform product development was used.
6.13 Comparison between simulated and actual data

In the previous sections, the results of the simulation for each project and platform were presented. The next question to ponder is whether this method provides useful outputs. According to the joint distributions and dynamic Gantt charts, the method produces higher cost and schedule estimates (Table 6.8). For proprietary reasons real cost and schedule information will not be displayed in this thesis, however, the information presented is a good indicator of real data. This behavior is the result of additional iterations and feedback loops in the data resulting in longer completion times.

<table>
<thead>
<tr>
<th></th>
<th>Actual ***</th>
<th>Simulated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>schedule*</td>
<td>cost**</td>
</tr>
<tr>
<td>SAS project, first data</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>SAS project, second data</td>
<td>94</td>
<td>100</td>
</tr>
<tr>
<td>UPAS project, first application</td>
<td>365</td>
<td>600</td>
</tr>
<tr>
<td>UPAS project, second application</td>
<td>355</td>
<td>575</td>
</tr>
</tbody>
</table>

* schedule is given in days
** cost is given in thousands of dollars
*** actual schedule and cost are estimates of Valeo allotments for the project

<table>
<thead>
<tr>
<th>Increase from actual to simulated data</th>
</tr>
</thead>
<tbody>
<tr>
<td>33% 53% 44% 68%</td>
</tr>
<tr>
<td>22% 5% 23% 5%</td>
</tr>
<tr>
<td>8% 16% 17% 22%</td>
</tr>
<tr>
<td>30% -3% 32% -2%</td>
</tr>
</tbody>
</table>

Table 6.8 Comparison between simulated and actual data

The method provides interesting lessons for managers who sometimes underestimate cost and schedule for complex product development. The model might be useful for managers when asked to submit quotes to OEMs. Managers could submit an upper estimate of schedule-cost to suppliers, as predicted from the model, or simply managers could assess the level of risk when developing a complex product, for OEMs. Since this was the first experience with estimated ranges of schedule times it is likely that the model would be conservative in estimated extended schedules and increased costs. It might be anticipated that more experience with the method would produce more accurate results.
Chapter 7

Conclusion

7.1 Key lessons

This thesis used a probabilistic methodology to analyze complex systems product developments. Many areas of knowledge were used in this work, especially in the area of engineering, statistics, management, economics and marketing.

From the engineering standpoint, the two products developed at Valeo were the steering angle sensor and the ultrasonic park assist sensor. Product design and management personnel for both platforms, interacted in this thesis. Both products provided the opportunity to test the simulation model under real situations involving schedule, cost, and performance as key indicators of successful product development.

Several multivariate techniques were used to calculate probability density functions, cumulative probabilities, histograms, and first and second moments to obtain the mean and variance of various joint probability densities. The powerful and versatile Monte Carlo method was used to model variations in duration and cost. Marginal distributions of the joint densities were obtained by projecting the tri-dimensional probability densities into one of the two planes for schedule and cost, as a way to understand their behavior in terms of a single variable. However, the basis of the simulation was the modeling of activity duration and cost with triangular density functions as approximations of gamma density functions.

Graphical methods were applied to model product development, (e.g., dependency structure matrices), as a way to represent feedback and interaction among activities. Gantt charts were created as another way to model rework in product development. Tables were used to represent the effects of activities on the performance measures. Different types of risks involved in complex product development were examined including those related to schedule, cost, performance, technology, market and business.

Concepts from microeconomics, including utility and multiattribute utility theory were employed. Those concepts try to explain and quantify consumer behavior, arguing that if the consumer is defined as rational in the sense that he or she prefers more of a good item or less of a bad one, this behavior can be described by a utility function. In this thesis, the consumer was the final user or customer of the product, and with the help of engineering and program managers, multiattribute utility functions were created corresponding to the multidimensional performance measure space. Also, economics concepts were used to maximize the multidimensional utility for the final customer and to address the implications of using learning curve effects in the development.

The voice-of-the-customer was employed to elicit the principal product performance measures important for the customer in evaluating the final product. The model translates the voice-of-the-customer into quantitative measures of product performance with aid of utility theory.

7.2 Future applications

The model quantifies the level of risk for cost, schedule and performance in product development. It can be used to investigate the effect on those dimensions by changing the process configuration, i.e., by rearranging individual tasks. Increasing the amount of rework and iteration will usually increase, schedule and cost, but with a potential reduction in technical performance risk. The model could be used then to test several configurations and then select the best one in terms of overall performance and overall performance risk reduction. Also, the possibility of moving manufacturing activity, usually done at the final stages of the product development, to the beginning of the process, could be evaluated.
A second application would be to evaluate intentional iteration in a process, as decided by management, and to predict the impact in overall performance. Usually the team benefits from sharing information among members and, in this case, the increase in overall performance would be quantified by doing more iteration.

A third application of the model could be the evaluation of adding and/or removing activities in the process. Perhaps some of the activities have a lesser impact on overall performance than other activities. The task could be to reduce or remove less effective activities and increase more effective activities. The activity-performance-measure-and-effect-table (APMET) could be used to identify these factors.

A fourth application, but not yet implemented in the model, could be the use of uncertain targets for each one of the performance measures. The targets were used to obtain different levels of risk for each performance measure. But sometimes, there is not a clear idea of what the final value for each target will be. Uncertain targets could be represented by random variables.

The list provided here is by no means exhaustive, but it provides some insight into the potential applications and limitations of the model in a complex product development.

### 7.3 Final remarks

The model applied in this thesis can be thought of as a tool for analyzing complex product developments. Although it does not exactly replicate a real process it offers a more realistic view of the real environment by accounting for uncertainty, feedback and iteration. The Valeo product development team provided positive feedback on the model. This model proved to be an excellent candidate to simulate complex product developments like the Valeo SAS and UPAS used in the automotive industry.

The next step for the model is to make it user-friendly and create, a stand-alone version with some of the MATLAB and Excel features used in the simulation. In this thesis, both packages were used to produce the results and graphs. But it should be possible to create a newer version incorporating all the necessary tools to perform the simulation and results presented in this thesis.
Appendix A

Valeo, an Overview

Valeo is an international automotive supplier dedicated to the design, manufacturing and sale of components, systems and modules for cars and trucks. It operates 151 facilities, including 29 R&D centers, 118 production plants and 10 distribution centers in 20 countries worldwide (Figure A.1), with sales of approximately $7 billion.

Valeo is a partner with all the major original equipment manufacturers with strong positions in Europe and North America (Figure A.2), helping them with their designs and developments. The company has 9 industrial operating units, one per product and system line, and a distribution unit (Figure A.3), with electronics accounting for approximately 6.8% of total sales (Figure A.4)

Every year, the group invests approximately 6.2% of total sales in R&D, with special emphasis on its electronics division, investing 10% of its sales, making the division responsive to its customers. Some of the products developed and manufactured by the electronics unit are electronic clutches and HID lamps, electronically controlled motors, security electronics, steering angle sensors, magnetic field sensors, ultrasonic park assist sensors, short distance Radar and body controllers (Figure A.5).

![VALEO SITES WORLDWIDE](image)

Figure A.1 Valeo presence around the world
1987-1998 WORLDWIDE SALES

1987
Non European 14% France
35% 56%
Rest of Europe

1998
North America 29% France
8% 41%
Other Markets
Rest of Europe

INTERNATIONAL = 78% OF TOTAL SALES

Figure A.2 1987-1998, Worldwide sales

Figure A.3 Valeo Industrial branches, Electronics $620 million of total sales
Figure A.4 Contribution of Electronics branch to total sales

Figure A.5 Valeo Electronics product portfolio
Appendix B

Steering Angle and Ultrasonic Park Assist Sensors

B1. Steering Angle Sensor

Probably one of the simplest definitions of the steering angle sensor (SAS) describes the sensor as a human-machine interface for the electronic stability program (ESP), determining the intention of the driver at any time (Figure B.1).

Because of the use in the ESP, the SAS must have its own security control with a very high reliability, using redundancy where appropriate. The ideal location of the SAS is between the steering wheel and the column switch, allowing for a space-efficient integration of clock-spring or module solution (Figure B.2).

Some of the applications of the SAS are summarized in Table B.1 and its characteristics are presented in Table B.2.

<table>
<thead>
<tr>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Electronic stability programs</td>
</tr>
<tr>
<td>2 Navigation systems</td>
</tr>
<tr>
<td>3 Damping control systems</td>
</tr>
<tr>
<td>4 Electronic power steering</td>
</tr>
<tr>
<td>5 Steering by wire</td>
</tr>
</tbody>
</table>

Table B.1 Steering Angle Sensor and its applications in the automotive industry

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>1.5 degrees</td>
</tr>
<tr>
<td>Measuring range</td>
<td>+/- 1040 degrees</td>
</tr>
<tr>
<td>Angle of initialization</td>
<td>4.5 degrees</td>
</tr>
<tr>
<td>Maximum angular velocity</td>
<td>2000 degrees/s</td>
</tr>
</tbody>
</table>

Table B.2 Technical specifications of the steering angle sensor
Figure B.1 Location of SAS in the car

Figure B.2 Ideal location of SAS in the steering wheel
B2. Ultrasonic Park Assist Sensor

This sensor uses the sonar ranging method to detect the distance between one's own vehicle and any other object. The system detects and announces what the driver cannot see, for example fences, walls and concrete posts.

A typical system consists of 8 ultrasonic sensors, with 4 located on the front bumper and one on the back. The system consist also of an electronic control unit (ECU) typically located inside the vehicle, front and rear speakers, activation push button and activation by reverse gear (Figure B.3). Measuring distances of the ultrasonic transducers are 25 to 60 cm laterally, 25 to 70 cm front middle, and 25 to 150 cm rear middle. The system is also activated manually using a pushbutton with an indicator lamp.

Figure B.3 Location of UPAS and schematic of main functions

Activated by the central ECU, the electronic transducers successively emit ultrasonic pulses receiving the echo signals from the objects and then transmitting them to the ECU for evaluation. The ECU calculates the distance to the obstacle from the time difference between transmitting and receiving. The measuring areas of the sensors have different adjustments and complement each other to form a total measuring area that is optimized (Figure B.4).

Figure B.4 Lateral view and approximate lateral detection range

If an object is recognized behind the vehicle, the ECU generates an intermittent tone through a loudspeaker in the rear, whose duty cycle changes proportional by the decreasing distance, until it turns into a continuous tone at a distance of approximately 25 cm. The front speaker is adjusted to emit a higher frequency sound to differentiate obstacles located in the back. If several sensors indicate an obstacle, the ECU verifies the signal and selects the least measured distance for display. The control unit continuously checks all equipment components for perfect function, and in the event of an error, the system warns the driver with a prolonged tone when the equipment is turned on. The system stores a code for future technical servicing. When active, the system disengages automatically when the vehicle has traveled more than 50 km or if the speed is higher than 30 km/h.
Appendix C

SAS Gantt charts
<table>
<thead>
<tr>
<th>ID</th>
<th>Event</th>
<th>Duration</th>
<th>June</th>
<th>July</th>
<th>August</th>
<th>September</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phase 0: Integrity Analysis</td>
<td>20.5 dys</td>
<td>21 22 23</td>
<td>24 25 26</td>
<td>27 28 29</td>
<td>30 31 32 33 34 35 36 37 38</td>
</tr>
<tr>
<td>2</td>
<td>DC Integrity Test</td>
<td>10 dys</td>
<td>Ruff, Wigger, Haas, Weiß, Tornar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Create SW developmental plan</td>
<td>5 dys</td>
<td>Haas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Create SCM Software plan</td>
<td>5.5 dys</td>
<td>Wigger</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Set Coding Rules / Select Tool</td>
<td>3 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Code for 3. Create Code</td>
<td>5 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Phase 1: Specification</td>
<td>34 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>HS-CAN Controller: SAS Function &amp; Safety</td>
<td>10 dys</td>
<td>Ruff, Wigger, Haas, Weiß, Tornar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Cruise control device Function &amp; Safety</td>
<td>2 dys</td>
<td>Haas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SPI Interface Specification</td>
<td>5 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>LS-CAN Controller: SAS Function &amp; Safety</td>
<td>12 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SAS and Cruise control device Validation plan</td>
<td>5 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Switch process / LS-CAN Controller function</td>
<td>10 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Diagnosis process and Diag. Specification date</td>
<td>7 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Diagnosis Validation plan</td>
<td>10 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Switch Validation plan</td>
<td>15 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Specification for &quot;HW in the loop&quot; test</td>
<td>11 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Review of Specification &amp; Validation document</td>
<td>3 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Hardware Software Interface</td>
<td>10 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>End of Specification's phase</td>
<td>0 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>MRSM Assembly &quot;HW in the loop&quot; Box + Software</td>
<td>24 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>MRSM Hardware (hard-faced Board) for Emulator</td>
<td>15 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Phase 2 Adaptation</td>
<td>35 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Software Integration test plan</td>
<td>2 dys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SOFTWARE MRSM: First data

<table>
<thead>
<tr>
<th>ID</th>
<th>Event</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Software design file (SA, SD innovator)</td>
<td>13 dys</td>
</tr>
<tr>
<td>26</td>
<td>Software Implementation</td>
<td>12 dys</td>
</tr>
<tr>
<td>27</td>
<td>Software Integration test</td>
<td>5 dys</td>
</tr>
<tr>
<td>28</td>
<td>Software Validation</td>
<td>5 dys</td>
</tr>
<tr>
<td>29</td>
<td>Software Sample-A finished</td>
<td>0 dys</td>
</tr>
</tbody>
</table>

### Diagram

- Vorgang: Project tasks
- Rollup-Vorgang: Project overview
- Rollup-Meilenstein: Milestone overview
- Rollup-Fortschritt: Progress overview
- Projekt-Sammelvorgang: Project summary
- Unterbrechung: Break
- Rollup-Unterbrechung: Break overview

**Projekt: MRSM_SW.mpp**
**Datum: Thu 05/25/00**

*Seite 2*
SOFTWARE MRSM: Second data

<table>
<thead>
<tr>
<th>ID</th>
<th>Event</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Phase 0: Integrity Analysis</td>
<td>20.5 dys</td>
</tr>
<tr>
<td>8</td>
<td>DC Integrity Test</td>
<td>10 dys</td>
</tr>
<tr>
<td>9</td>
<td>Create SW developmental plan</td>
<td>5 dys</td>
</tr>
<tr>
<td>10</td>
<td>Create SCM Software plan</td>
<td>5.5 dys</td>
</tr>
<tr>
<td>11</td>
<td>Set Coding Rules and Select Tool</td>
<td>3 dys</td>
</tr>
<tr>
<td>12</td>
<td>Code for J. Create Code</td>
<td>5 dys</td>
</tr>
<tr>
<td>13</td>
<td>Phase 1: Specification</td>
<td>34 dys</td>
</tr>
<tr>
<td>14</td>
<td>HS-CAN Controller, SAS Function &amp; Safety</td>
<td>10 dys</td>
</tr>
<tr>
<td>15</td>
<td>Cruise control device Function &amp; Safety</td>
<td>2 dys</td>
</tr>
<tr>
<td>16</td>
<td>SPI Interface Specification</td>
<td>5 dys</td>
</tr>
<tr>
<td>17</td>
<td>LS-CAN Controller, SAS Function &amp; Safety</td>
<td>12 dys</td>
</tr>
<tr>
<td>18</td>
<td>SAS and Cruise control device Validation plan</td>
<td>5 dys</td>
</tr>
<tr>
<td>19</td>
<td>Switch process / LS-CAN Controller function</td>
<td>10 dys</td>
</tr>
<tr>
<td>20</td>
<td>Diagnosis process and Diag. Specification date</td>
<td>7 dys</td>
</tr>
<tr>
<td>21</td>
<td>Diagnosis Validation plan</td>
<td>10 dys</td>
</tr>
<tr>
<td>22</td>
<td>Switch Validation plan</td>
<td>15 dys</td>
</tr>
<tr>
<td>23</td>
<td>Specification for &quot;HW in the loop&quot; test</td>
<td>9 dys</td>
</tr>
<tr>
<td>24</td>
<td>Review of Specification &amp; Validation document</td>
<td>3 dys</td>
</tr>
<tr>
<td>25</td>
<td>Hardware Software Interface</td>
<td>10 dys</td>
</tr>
<tr>
<td>26</td>
<td>End of Specification's phase</td>
<td>0 dys</td>
</tr>
<tr>
<td>27</td>
<td>Phase 2: Adaptation Sample A</td>
<td>39 dys</td>
</tr>
<tr>
<td>28</td>
<td>Software design file (SA, SD Innovator)</td>
<td>13 dys</td>
</tr>
<tr>
<td>29</td>
<td>MRSM Assembly &quot;HW in the loop&quot; Box + Softv</td>
<td>35 dys</td>
</tr>
<tr>
<td>30</td>
<td>Software Sample A Finished</td>
<td>0 dys</td>
</tr>
</tbody>
</table>

Projekt: MRSM SW mpp
Datum: Thu 05/25/00
<table>
<thead>
<tr>
<th>ID</th>
<th>Event</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>Phase 2: Adaption Sample B</td>
<td>55 dys</td>
</tr>
<tr>
<td>36</td>
<td>LSS Software HIT Tester</td>
<td>48 dys</td>
</tr>
<tr>
<td>37</td>
<td>LWS Software HIL Tester</td>
<td>16 dys</td>
</tr>
<tr>
<td>38</td>
<td>Implementation Update of sample A</td>
<td>17 dys</td>
</tr>
<tr>
<td>39</td>
<td>Validation Update of Sample A</td>
<td>4 dys</td>
</tr>
<tr>
<td>40</td>
<td>Update of Sample A Finished</td>
<td>0 dys</td>
</tr>
<tr>
<td>41</td>
<td>Switch Validation plan &amp; Diagnosis</td>
<td>14 dys</td>
</tr>
<tr>
<td>42</td>
<td>Implementation Sample B</td>
<td>18 dys</td>
</tr>
<tr>
<td>43</td>
<td>New sample's lab Finished</td>
<td>0 dys</td>
</tr>
<tr>
<td>44</td>
<td>New sample's lab integration test</td>
<td>4 dys</td>
</tr>
<tr>
<td>45</td>
<td>New HIL test with LR-Switch Finished</td>
<td>0 dys</td>
</tr>
<tr>
<td>46</td>
<td>Validation with LR-Switch</td>
<td>7 dys</td>
</tr>
<tr>
<td>47</td>
<td>Software Sample B Finished</td>
<td>0 dys</td>
</tr>
</tbody>
</table>

**Diagram:**

[Diagram of project milestones and timelines related to the software MRSM project.]

**Software MRSM: Second data**

**Projekt: MRSM_SW.mpp**
**Datum: Thu 05/25/00**

**Vorgang** | **Rollup-Vorgang** | **Projekt-Sammlvorgang**
---|---|---
**Fortschritt** | **Rollup-Meilenstein** | **Unterbrechung**
**Meilenstein** | **Rollup-Fortschritt** | **Rollup-Unterbrechung**
**Sammelvorgang** | **Externe Vorgänge** | ****
UPAS Volume capacity extension

<table>
<thead>
<tr>
<th>ID</th>
<th>Event</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pre-assembly</td>
<td>Wed 05/26/99</td>
<td>Mon 05/29/00</td>
</tr>
<tr>
<td>2</td>
<td>Cleaning Unit (for glue process)</td>
<td>Wed 05/26/99</td>
<td>Fri 06/25/99</td>
</tr>
<tr>
<td>3</td>
<td>Glue machine 2</td>
<td>Fri 06/25/99</td>
<td>Tue 07/27/99</td>
</tr>
<tr>
<td>4</td>
<td>Curing Oven for splice</td>
<td>Tue 07/27/99</td>
<td>Fri 08/20/99</td>
</tr>
<tr>
<td>5</td>
<td>Assembly 2. Pre-assembly line</td>
<td>Mon 08/30/99</td>
<td>Mon 11/15/99</td>
</tr>
<tr>
<td>6</td>
<td>Cleaning Unit (for foaming process)</td>
<td>Mon 08/30/99</td>
<td>Mon 01/03/00</td>
</tr>
<tr>
<td>7</td>
<td>2K-foam tool for Backing Material</td>
<td>Mon 01/03/00</td>
<td>Fri 04/14/00</td>
</tr>
<tr>
<td>8</td>
<td>Curing Oven for Backing Material</td>
<td>Fri 06/25/99</td>
<td>Thu 01/06/00</td>
</tr>
<tr>
<td>9</td>
<td>Assembly Unit for Padding and Cover</td>
<td>Mon 01/03/00</td>
<td>Mon 03/27/00</td>
</tr>
<tr>
<td>10</td>
<td>Pallets system</td>
<td>Mon 03/27/00</td>
<td>Thu 05/04/00</td>
</tr>
<tr>
<td>11</td>
<td>Burn-in</td>
<td>Mon 05/01/00</td>
<td>Mon 05/29/00</td>
</tr>
</tbody>
</table>
Appendix D

Gamma Probability Density Function

The gamma density function depends on two parameters, \( \alpha \) and \( \lambda \). The mathematical representation of this density is given by the following equation

\[
f(t) = \begin{cases} 
\frac{\lambda^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\lambda t} & t \geq 0 \\
0 & t \leq 0 
\end{cases}
\]

The gamma function \( \Gamma(\alpha) \) is defined as

\[
\Gamma(\alpha) = \int_0^\infty u^{\alpha-1} e^{-u} du
\]

If \( \alpha = 1 \), the gamma density converts to the exponential density. The parameter \( \alpha \) is called a shape parameter for the gamma density, and \( \lambda \) is called a scale parameter. Varying \( \alpha \) changes the shape of the density, whereas varying \( \lambda \) corresponds to changing the units of measurement. The gamma density is useful for modeling non-negative random variables. Figure D.1 presents several gamma densities for various values of \( \alpha \) and \( \lambda \).

The expected value, variance, moment generating function and local maximum are given by the following equations

\[
E(t) = \frac{\alpha}{\lambda}, \quad \text{Var}(t) = \frac{\alpha}{\lambda^2}, \quad \text{M}(t) = \left(\frac{\lambda}{\lambda - t}\right)^\alpha, \quad t < \lambda \\
t_{\text{max}} = \frac{\alpha - 1}{\lambda}, \quad f_{\text{max}} = \frac{\lambda(\alpha - 1) e^{\lambda - 1 - \alpha}}{\Gamma(\alpha)}
\]

Figure D.1 Gamma probability densities for several parameters
Appendix E

Triangular Probability Density Function

The triangular density is used as an approximation of the gamma distribution. It is useful for modeling activity duration. This density is defined for positive values, though mathematically it could also be defined for negative values. It is expressed according to the following equation:

$$f_A(t) = \begin{cases} 
0 & \text{for } t < a \\
\frac{2}{(b-a)(c-a)}(t-a) & \text{for } a \leq t < b \\
\frac{2}{(c-a)(c-b)}(t-c) & \text{for } b \leq t < c \\
0 & \text{for } t \geq c 
\end{cases}$$

![Triangular Probability Density Function](image)

**Figure E.1** Triangular probability density defined by three values

This density has some interesting properties, which are summarized in the following equations:

$$E[t] = \frac{a + b + c}{3}$$

$$\text{Var}[t] = \frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}$$

Cumulative Probability

$$CP_r = \begin{cases} 
0 & \text{for } t < a \\
\frac{1}{(b-a)(c-a)}(t-a)^2 & \text{for } a \leq t < b \\
1 - \frac{1}{(c-a)(c-b)}(c-t)^2 & \text{for } b \leq t < c \\
1 & \text{for } t \geq c 
\end{cases}$$

Monte Carlo simulation sample

$$t = \begin{cases} 
0 & \text{for } t < a \\
a + \sqrt{r(b-a)(c-a)} & \text{for } a \leq t < b \\
c - \sqrt{(1-r)(c-a)(c-b)} & \text{for } b \leq t < c \\
0 & \text{for } t \geq c 
\end{cases}$$

$r$ is uniformly distributed between 0 and 1.
Monte Carlo Simulation Vs. Convolution

One of the main problems is to add sampled activity duration and cost coming from different probability density functions to obtain total activity duration and cost for the process. Each activity duration and cost was modeled with a triangular density. The objective was to obtain the general probability density of duration and cost as a function of individual probability density functions.

Several techniques and methods can be employed to add probability density function, e.g., convolution, method of moments and simulation. However the main focus was on convolution and simulation, especially the Monte Carlo simulation.

**F1 Mathematical Convolution**

Suppose X and Y are independent continuous random variables, with probability density functions $f_X(x)$ and $f_Y(y)$, respectively. The probability density of function of $Z = X + Y$, denoted by $F_Z(z)$ is given by

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x)f_Y(z-x)dx$$

This integral is called the convolution of the functions $f_X(x)$ and $f_Y(y)$. As an example, consider the convolution of two normally distributed random variables, $f_X(x)$ and $f_Y(y)$

$$f_X(x) = \frac{1}{\sigma_X \sqrt{2\pi}} e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}}$$
$$f_Y(y) = \frac{1}{\sigma_Y \sqrt{2\pi}} e^{-\frac{(y-\mu_Y)^2}{2\sigma_Y^2}}$$

After some calculations, the convolution of those two random variables is given by

$$f_Z(z) = \frac{1}{\sqrt{2\pi \left(\sigma_X^2 + \sigma_Y^2\right)}} e^{-\frac{(z-(\mu_X + \mu_Y))^2}{2\left(\sigma_X^2 + \sigma_Y^2\right)}}$$

Convolution of two normally distributed random variables generates another normally distributed random variable whose mean is the sum of the means and whose variance is the sum of the variances. These results can be generalized to demonstrate that the convolution of $n$ normally distributed random variables with means $\mu_i$ and standard deviations $\sigma_i$ is given by

$$f_Z(z) = \frac{1}{\sigma_Z \sqrt{2\pi}} e^{-\frac{(z-\mu_Z)^2}{2\sigma_Z^2}}$$
$$\mu_Z = \sum_{i=1}^{n} \mu_i$$
$$\sigma_Z = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$$
F2 Monte Carlo simulation

Adding probability density functions is performed with mathematical convolutions. Convolution of normally distributed random variables is straightforward. However, this operation with a large and sometimes variable number of densities, not necessarily normally distributed, is extremely complicated and burdensome. An alternative approach to address this problem is to use the Monte Carlo simulation method, taking samples of duration and cost for each activity and adding the values to obtain the total duration and cost for the whole project. This process is repeated many times to arrive at simulated distributions of duration and cost.

As an example, assume that the activity duration is modeled with normally distributed random variables though the model is better represented with gamma densities or triangular densities, used as approximations of the gamma densities. Suppose there are three activities whose durations are normally distributed, with means and standard deviations given by $N(\mu, \sigma)$

Activity 1: $t_1 \sim N(15, 4)$
Activity 2: $t_2 \sim N(30, 6)$
Activity 3: $t_3 \sim N(45, 8)$

If the convolution is applied to these three distributions, the total duration will behave as a normal distributed random variable, with mean $\mu = 90$ and standard deviation $\sigma = 10.7703$. If the Monte Carlo simulation is applied taking duration samples from each one of the three activities, adding the values, and repeating the process, 5000 times, the new density function would have an almost normal behavior, with mean $\mu = 90.1251$ and standard deviation $\sigma = 10.7453$ (Figure F.1).

![Convolution of Normal Distribution Functions](convolution.png)

**Figure F.1** Monte Carlo simulation to obtain total duration of three activities modeled as normal random variables
Similarly, the cumulative density for the convolution and simulation results could be obtained (Figure F.2). Both graphs show that simulation could be used instead of convolution without significant differences.

![Graph showing comparison between fitted and observed total duration of three normally distributed random variables](image)

**Figure F.2** Comparison between fitted and observed total duration of three normally distributed random variables

Triangular distributed random variables are required to model activity duration and cost. However, such random variables are not usually available in currently available statistical software packages. There is now the problem of generating triangular distributed random variables for the simulation of the process.

One of the most useful results of mathematical statistics can be used to generate triangular distributed random variables.

**Proposition:** Let $U$ be a uniformly distributed random variable in $[0,1]$, and let $X = F^{-1}(U)$. Then the cumulative and probability density function of $X$ are $F(x)$ and $f(x) = \frac{dF}{dx}$, respectively.

**Proof**

\[ P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x), \text{ because } U \text{ is uniform in } [0,1] \]

And

\[ f(x) = \frac{dF(x)}{dx} = \frac{dF}{dx} \]

This proposition demonstrates that random variables can be generated with cdf $F$ by applying $F^{-1}$ to uniform random variables, as long as $F^{-1}$ can be easily obtained. In the present case, the cumulative density function can be obtained from the triangular density, which by itself will be a uniform random variable. Then, the variable is solved for $t$ and a triangular distributed random variable $t$ is obtained, generated from the uniform random variable $r$ in $[0,1]$ (Appendix E). As an example, triangular densities are generated with Monte Carlo simulation using 500; 5,000 and 10,000 points, respectively (Figures F.3, F.4, and F.5).
The Monte Carlo simulation sample

\[
\begin{cases} 
0 & \text{for } t < 2 \\
2 + 2\sqrt{3r} & \text{for } 2 \leq t < 4 \\
8 - 2\sqrt{6(1-r)} & \text{for } 4 \leq t < 8 \\
0 & \text{for } t \geq 8 
\end{cases}
\]

where \( r \) is uniform in [0,1].

Figure F.5 Triangular distribution, 10,000 sample

Suppose again, the total duration of three activities is desired but they are modeled this time as triangular distributed random variables. If the simulation is run 10,000 times, the results are those shown in Figures F.6 and F.7.

The Monte Carlo simulation will generate results that are very close to those given by mathematical convolution. For the present purposes, simulation will suffice for the product development model used in this thesis.
Convolution of Triangular Distribution Functions

**Figure F.6** Monte Carlo simulation to obtain total duration of three activities modeled as triangular random variables

**Figure F.7** Comparison between fitted and observed total duration of three triangular distributed random variables
Appendix G

Risk, a Mathematical Perspective

Risk is a function of the probability of unacceptable outcomes and the consequences of those outcomes, provided a target is specified. Some of the outcomes may have worse consequences therefore the need to define a weighting function to quantify for this possibility.

The probability of unacceptable outcomes is defined by:

\[ P_s(\text{unacceptable}) = \int_{T_s}^{\infty} f_s(x) \, dx \]

The previous expression is defined for "small is better" (SIB) processes, such as schedule and cost, but for "large is better" (LIB) processes, such as reliability, the limits of the integral will be minus infinity to \( T_s \). A weight function, known as impact function, can be defined penalizing more drastically outcomes located farther away from the target \( T_s \). One way to define the impact function is by using a quadratic function defined for values greater than the target. Although different functions of different orders and forms can be chosen for that purpose, such a choice must make sense in the current context. Assuming a quadratic impact function, the impact is defined as:

\[ I(x) = \begin{cases} 0 & \text{for } x \leq T_s \\ k_s(x - T_s)^2 & \text{for } x > T_s \end{cases} \]

The constant \( k_s \) scales the risk resulting in consistent units of risk for the customer or the firm, e.g., lost in revenue and cost. If both definitions of probability of unacceptable outcomes and impact function are adopted, risk can be define mathematically as:

\[ \text{Risk} = k_s \int_{T_s}^{\infty} (x - T_s)^2 f_s(x) \, dx \]

In Figure G.1 we show graphically the concepts previously discussed.

Figure G.1 Normal probability density, quadratic impact density, and target.
Figure G.3 Tri-dimensional probability density with directional impact density

Figure G.4 Tri-dimensional risk density
Figure G.3 Tri-dimensional probability density with directional impact density

Figure G.4 Tri-dimensional risk density
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APMET</td>
<td>Activity Performance Measures and Effects Table</td>
</tr>
<tr>
<td>APMST</td>
<td>Activity Performance Measures and Strengths Table</td>
</tr>
<tr>
<td>BCV</td>
<td>Best Likely Value</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Density Function</td>
</tr>
<tr>
<td>DSM</td>
<td>Design Structure Matrix</td>
</tr>
<tr>
<td>LIB</td>
<td>Large-is-better</td>
</tr>
<tr>
<td>MLV</td>
<td>Most Likely Value</td>
</tr>
<tr>
<td>NDRV</td>
<td>Normally Distributed Random Variable</td>
</tr>
<tr>
<td>NIB</td>
<td>Normal-is-better</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PMF</td>
<td>Probability Mass Function</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Research and Development</td>
</tr>
<tr>
<td>RV</td>
<td>Random Variable</td>
</tr>
<tr>
<td>SAS</td>
<td>Steering Angle Sensor</td>
</tr>
<tr>
<td>SIB</td>
<td>Small-is-better</td>
</tr>
<tr>
<td>UPAS</td>
<td>Ultrasonic Park Assist Sensor</td>
</tr>
<tr>
<td>WCV</td>
<td>Worst Likely Value</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


- Pearson, Karl, "Tables of the incomplete Beta Function", Originally prepared by under the direction of Karl Pearson, with a new introduction by E. S. Person and N. L. Johnson, Published by the Biometrica Trustees by Cambridge V.P., London, 1968


- Smith, Robert, Eppinger, Steven, "Identifying Controlling Features of Engineering Design Iteration", Management Science, 1997


