Characteristics that Make One Estimation Technique Better than Others

Arunkumar Jayaraman

May 2007
CMU-ISRI-07-111

Advisor
Mel Rosso Llopart

Institute for Software Research
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 - 3890

Abstract

The variety of software projects being carried out today is enormous. Some of them succeed while others fail. One key reason for failure of projects is lack of proper estimation or the use of inappropriate estimation techniques. While there are many estimation techniques developed for projects each of them have their own advantages and disadvantages. No estimation technique can be considered a silver bullet for all the projects.

Keywords: COCMO estimation, Use Case Point estimation, Expert Judgment estimation, Wide Band Delphi estimation

Index

1. Introduction

This report hypothesizes that certain estimation techniques provide better estimation results for certain projects based on their characteristics or factors. To substantiate the hypothesis, the report follows the tasks specified in Figure 1, where it analyzes projects and based on the analysis identifies factors or characteristics of projects. It then goes on to identify input parameters for four estimation techniques namely, COCOMO, Use Case Point, Expert Judgment and Wide Band Delphi. The report then correlates the project factors or characteristics with the input parameters of these estimation techniques to identify the kind of projects for which COCOMO, Use Case Point, Expert Judgment and Wide Band Delphi estimations would provide better estimation results.

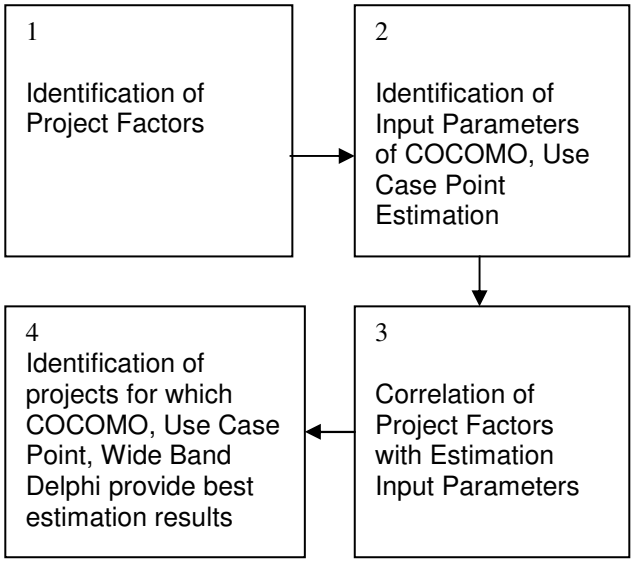

Figure 1: Steps to identify projects for which COCOMO, Use Case Point, and Estimation by Expert Judgment and Wide Band Delphi Estimations provide best estimation results

2. Identification of Project Factors

When software projects are analyzed, it is found that each have their own characteristics or factors. It is very important to identify the project characteristics or factors since the correlation of these project factors or characteristics with input parameters of estimation techniques is necessary to find out which estimation techniques provide better estimation results for which type of projects. Following are some generic characteristics or factors common to most software projects which are classified under project, people, technology and process characteristics of projects.

- **Project Factors**
    - **Project Duration and Complexity** – Project duration describes the time taken to complete the project and project complexity describes how difficult it is to undertake and complete the project from various perspectives like technology, process etc. Some projects span over a few weeks and are simple e.g. a simple web development project which consists of only static HTML pages. There are also projects which are complex and which span over years like those of NASA which involve mission control systems for space craft.

- **People Factors**

  - **Customer Interaction with Project Team** - Customers might interact with the development teams often either for joint development or for quick deliveries. There are also customers who have a vague idea of the system to develop and who give the development teams the leeway to develop the envisioned systems without much interaction.
  - **User Interaction with Developed System** - Systems developed might have a lot of users who directly interact with it, e.g. a banking portal might have customers of the bank as well as the bank's staff interacting with it. However, there are many systems which do not have direct user interaction, e.g. systems which route network connections might have little or no direct user interaction.
  - **Domain Expertise of Project Development Team** - If the project team members have developed similar systems for solving problems in the same domain before, they would have a greater domain expertise. However, if the project is a venture into a new domain, the team members would consider themselves relatively inexperienced in the particular domain, even if they have previously developed several systems in other domains.
  - **Team Size** - Some projects involve only a team of four to eight members, while there are projects which involve large teams. Team size plays an important role in estimation due to factors like delegation of work, communication among team members, and other intra-member activities.
  - **Development Team Distribution** - If a project's team members are dispersed across geography, the communication overheads among them would affect estimation of the project. There might be time zone differences between locations of team members which might affect or which might favor a project.

- **Technology Factors**

  - **Technical Expertise of Team Members** - A project might require usage of cutting edge technologies or might sometimes even require usage of legacy technologies. In either case the project team members should be comfortable with technologies used to successfully complete the projects.
  - **Usage of Sophisticated Tools** - Project teams or their customers might want usage of sophisticated tools. If tools are used, and if the project teams are comfortable with these tools then the resource expenditure for the tasks would be greatly reduced, but if the teams are not acquainted with the tools used, then they would have to learn them before using them and this effects project estimates.
  - **Usage of Reusable Components** - One of the main advantages of current object oriented development methodologies is reusability of components. Component reuse has a major effect on estimation. It can drastically reduce the expenditure of engineer hours for developing the same components again, but one must also take into account factors like time taken to integrate reusable components and time taken to understand reusable components.
  - **Platforms involved in the project** - Systems might be entirely installed and run on a single platform. Or projects might also require the systems to be installed and run across more than one platform. If a system is to be deployed across multiple platforms issues related to deployment may arise. Such a system should also be tested extensively. In such situations, estimation should take into account the resources expended to handle the issues for testing.
  - **Programming Languages Used** - Some projects involve the usage of specialized programming languages like object oriented programming, etc. e.g. projects which involve heavy customer user interactions like a banking portal. On the other hand, there are also projects which still use legacy technologies, e.g. projects which involve batch processing.

- **Process Factors**

  - **Documentation Required** - Some customers might be interested and hence insist more on the built product while others might be interested in the built product as well as accompanying documentation. While estimating projects the project team should take into account the degree of documentation required for the project.

3. Identification of Input Parameters of Estimation Techniques

After identifying project characteristics or factors, the report will discuss input parameters associated with the estimation techniques of COCOMO, Use Case Point, Expert Judgment and Wideband Delphi. Input parameters of estimation techniques are as important as project characteristics or factors since their correlation with project characteristics or factors is necessary in finding out which estimation technique provides better estimation results for a particular type of project.

- **Input Parameters of COCOMO Estimation** - In COCOMO estimation, first the nominal development effort is estimated as a function of product's size in thousands of delivered source instructions and the project's development mode [3].

  Once the nominal development effort is estimated, COCOMO goes on to identify a set of effort multipliers in the form of cost drivers. After weighing these cost drivers, COCOMO multiplies the individual weights to obtain a final product. This product is in turn multiplied with nominal effort to obtain the estimated development effort [3].

  In essence, the input parameters of COCOMO Estimation are:

  - ♦ Software development effort multipliers or cost drivers
  - ♦ The project features which determine if projects are Organic, Semi-detached or Embedded

  - **Nominal Effort Estimation** - Nominal effort is estimated by first classifying projects based on development modes. Projects are classified into three development modes namely [3]:

    - Organic
    - Semidetached
    - Embedded

    This classification is based on various features of project [3]:

    - Organizational understanding of product objectives
    - Experience in working with related software systems
    - Need for software conformance with pre-established requirements
    - Need for software conformance with external interface specifications
    - Concurrent development of associated new hardware and operational procedures
    - Need for innovative data processing architectures, algorithms
    - Premium on early completion
    - Product size range

    Once the projects are classified, nominal development efforts are calculated (Table 1) [3]:

Project Development Mode	Nominal Effort Estimation in Man Months
Organic	3.2(KDSI)1.05
Semidetached	3.0(KDSI)1.12
Embedded	2.8(KDSI)1.20

    KDSI = thousands of delivered source instructions

Table 1: Nominal Effort Estimation

- **Identification and Calibration of Software Development Effort Multipliers** - After estimating the nominal development effort, COCOMO estimation technique identifies software development effort multipliers (cost drivers) under four different categories, namely [3]:

    - Product Attributes (Table 2)
    - Computer Attributes (Table 3)
    - Personnel Attributes (Table 4)
    - Project Attributes (Table 5)

Product Attributes
Required software reliability
Data base size
Product Complexity

Table 2: Product Attributes for COCOMO Estimation

Computer Attributes
Execution time constraint
Main storage constraint
Virtual machine volatility
Computer turnaround time

Table 3: Computer Attributes for COCOMO Estimation

Personnel Attributes
Analyst capability
Application experience
Programmer capability
Virtual machine experience
Programming language experience

Table 4: Personnel Attributes for COCOMO Estimation

Project Attributes
Use of modern programming practices
Use of software tools
Required development schedule

Table 5: Project Attributes for COCMO Estimation

Each of the software development effort multiplier is weighted based on its importance relative to the projects. The weights assigned to individual software development effort multipliers are multiplied and this product would be used in calculation of development effort estimation [3]:

- **Estimation of Development Effort** - Development effort is calculated by multiplying nominal effort estimate with product of weights of individual software development effort multipliers [3].

- **Input Parameters of Use Case Point Estimation** - In Use Case Point Estimation, first each use case in a project is assigned a numerical factor based on its complexity (in terms of number of transactions). The weighted sum of factors of individual use cases gives unadjusted use case points for the project [2].

After the calculation of unadjusted use case points, a set of technical factors are listed and are assigned weights relative to the particular project. The sum of these weights is computed and is termed as TFactor. This TFactor is used to calculate technical complexity factor of the project [2].

Similar to technical factors a set of environmental factors are listed and are assigned weights relative to the particular project. The weights are summed up and the sum is termed as EFactor. Using EFactor, the weighted environmental factor for the project is computed [2].

After obtaining unadjusted use case points, technical complexity factor and environmental factor, using them the final use case points for the project is computed [2].

Once use case points are computed, use case point estimation goes on to compute the estimated effort in person-hours. This is done based on number of simple, medium and complex use cases in the project [2].

Hence to sum it up, the main input parameters of use case point estimation are [2]:

- Complexity of use cases
- Technical Factors in the project
- Environmental Factors in the project

- **Calculation of Unadjusted Use Case Points** - Each use case in the project is assigned a numerical factor based on its complexity (in terms of number of transactions) (Table 6).

Use Case Type	Description	Factor
Simple	3 or fewer transactions	5
Medium	4-7 transactions	10
Complex	>7 transactions	15

Table 6: Weighing use cases based on their complexity

The weighted sum of the factors assigned to individual use cases is obtained and this represents unadjusted use case points for the project [2].

- **Calculation of Technical Complexity Factor** - The generic set of technical factors for the project is listed (Table 7) and each of these factors is weighed relative to the project.

Technical Factor	Weight
Distributed system	2
Response or throughput performance objectives	1
End-user efficiency (online)	1
Complex internal processing	1
Code must be reusable	1
Easy to install	0.5
Easy to use	0.5
Portable	2
Easy to change	1
Concurrent	1
Includes special security features	1
Provides direct access for third parties	1
Special user training facilities required	1

Table 7: Technical Factors and Weights

The sum of the individual weights is computed and termed as TFactor. Using this TFactor, technical complexity factor for the project is calculated using the formula:

Technical Complexity Factor (TCF) = 0.6+ (0.01*TFactor) [2].

- **Calculation of Environmental Factors** - After computing technical complexity factor, Use Case Point estimation lists a set of Environmental Factors for the team (Table 8) and weights each of them relative to the team.

Environmental Factor	Weight
Familiar with internet process	1.5
Application experience	0.5
Object-oriented experience	1
Lead analyst capability	0.5
Motivation	1
Stable requirements	2
Part-time workers	-1
Difficult programming language	-1

Table 8: Environmental Factors for Team and Weights

The weighted sum of the individual environmental factors is computed and is termed as EFactor. Using EFactor, Use Case Point estimation computes Environmental Factor with the following formula:

Environmental Factor = 1.4+ (-0.03*EFactor) [2].

- **Calculation of Use Case Points** - Once unadjusted use case points (UUCP), technical complexity factor (TCF) and environmental factor (EF) are calculated, Use Case Point estimation goes on to calculate Use Case Points using following formula:

Use Case Points = UUCP*TCF*EF [2].

- **Input Parameters of Estimation by Expert Judgment** - All of the project factors enlisted in this report could be considered as input parameters of Estimation by Expert Judgment. But one key input parameter which uniquely distinguishes Expert Judgment estimation technique is the involvement of human factor in the form of experts for project estimation [1].

- **Input Parameters of Wide Band Delphi Estimation** - Wide Band Delphi estimation is more or less similar to Estimation by Expert Judgment. Its key input parameter which makes it unique is experience of estimation team members. The main value which a project team gets in Wide Band Delphi Estimation over Expert Judgment is the process which encourages anonymous individual estimation of estimation team members followed by the team consensus on estimation [1][4][5].

4. Correlation of Project Characteristics or Factors with Input Parameters of Estimation Techniques

After identifying generic project characteristics or factors and input parameters of the four estimation techniques, the report now correlates the project characteristics or factors with the input parameters to see how each of the four estimation techniques provides better estimation results for projects of different characteristics or factors.

- **Correlation of Project Characteristics or Factors and Input Parameters of COCOMO Estimation** - The report correlates various project characteristics or factors with input parameters of COCOMO

estimation (Table 9). Based on the correlation the report identifies the type of projects for which COCOMO estimation would provide a better estimation result.

Project Characteristics or Factors	Input Parameter of COCOMO Estimation Technique
Project Duration and Complexity	Product Complexity
Customer Interaction with Project Team	Required development schedule
Domain Expertise of Project Development Team	Experience in working with related software systems, Application experience
Technical Expertise of Team Members	Analyst capability, Programmer capability, Virtual machine experience, Programming language experience
Usage of Sophisticated Tools	Use of software tools
Platforms involved in the project	Execution time constraint, Main storage constraint, Virtual machine volatility, Computer turnaround time
Programming Languages Used	Programming Language Experience

Table 9: Correlation of Project Characteristics or Factors with Input Parameters in COCOMO Estimation

From above correlation (Table 9), it is inferred that, COCOMO estimation technique would be able to produce better estimation results for projects:

- With varying complexities (although preferably moderate to complex projects because of the estimation overheads involved in COCOMO estimation technique)
- With varying development schedules (typically long, again due to the overheads involved in this estimation technique)
- In which domain expertise of team members plays a role in estimation
- In which technical expertise of team members plays a role in estimation
- In which tools and programming languages used play a role in estimation

Taking into account the above conclusions, COCOMO would be typically used in moderate to complex projects like those of Department of Defense, NASA where above stated project factors or characteristics play an important role in project estimation.

- **Correlation of Project Characteristics or Factors and Input Parameters of Use Case Point Estimation** - The report correlates the generic set of project characteristics or factors with input parameters of Use Case Point estimation (Table 10) and from this correlation, identifies the type of projects for which Use Case Point estimation would provide better estimation results.

Project Characteristics or Factors	Input Parameter of Use Case Point Estimation
Project Size and Complexity	Weighted sum of factors assigned to individual use cases based on complexity of transactions involved in them, Complex internal processing
Customer Interaction with Project Team	Stable Requirements
User Interaction with Developed System	End-user efficiency (online), Ease of installation, Ease of use, Easy to change, Special user training facilities

	required
Domain Expertise of Project Development Team	Application experience, Lead analyst capability
Technical Expertise of Team Members	Object-oriented experience
Usage of Sophisticated Tools	Ease of installation, Ease of use, Special user training facilities required
Usage of Reusable Components	Reusability of code
Platforms involved in the project	Distributed system, Portable, Reusability of code Ease of installation, Ease of use, Portable, Easy to change, Concurrent Includes special security features, Special user training facilities required
Documentation Required	Special user training facilities required
Programming Languages Used	Difficult programming language

Table 10: Correlation of Project Characteristics or Factors with Input Parameters in Use Case Point Estimation

From the above correlation (Table 4) it is clear that Use Case Point estimation technique would provide better estimation results for projects whose deliverables are subject to direct user interaction. E.g. A mail server project where web portals host email accounts of users and users in turn often interact with the system.

However, from the correlation, it is clear that Use Case Point estimation would not be helpful to projects whose end deliverables don't have user interaction. E.g. Use Case Point estimation would not be helpful for a project which aims at development of batch processing system or software since there would be no use cases to go about with the estimation.

- **Correlation of Project Characteristics or Factors and Input Parameters of Expert Judgment Estimation** - All the project characteristics or factors can be considered as input parameters for Expert Judgment estimation. However there is one notable correlation which makes Expert Judgment estimation unique (Table 11).

Project Characteristics or Factors	Input Parameters of Estimation by Expert Judgment
Domain expertise of project development team, technical expertise of team members	Involvement of experts for project estimation

Table 11: Correlation of Project Characteristics or Factors with Input Parameters in Estimation by Expert Judgment

It is better to use Estimation by Expert Judgment along with other estimation techniques rather than only by itself. One main disadvantage of this estimation technique is that it cannot be used in projects involving a new domain where there is no existing expert knowledge. But this estimation technique provides great value to a project in the presence of experts who have prior experience in similar kinds of projects.

- **Correlation of Project Characteristics or Factors and Input Parameters of Wide Band Delphi Estimation Technique** - Similar to Expert Judgment, Wide Band Delphi estimation considers all the project characteristics or factors as input parameters for estimation. However there is one important correlation which makes Wide Band Delphi Estimation unique (Table 12).

Project Characteristics or Factors	Input Parameters of Wide Band Delphi Estimation
Domain expertise of project development team, technical expertise of team members	Experience of estimation team members

Table 12: Correlation of Project Characteristics or Factors with Input Parameters in Wide Band Delphi Estimation Technique

Wide Band Delphi even though similar to Expert Judgment estimation provides more value to the project team due the presence of a process which has following advantages:

- **Normalization of Individual Members' Views** - In Wide Band Delphi, the project teams go for normalization of the views of their individual team members through common consensus. This is better than a technique where the weights of estimation are given on a subjective basis by a single individual.
- **More Accurate Estimation due to inputs from various angles** - Each member of the estimation group provides valuable inputs from various angles. The consolidated estimation is hence more accurate and is bound to less deviation from actual value.
- **Anonymous Individual Estimation** - Individuals in the estimation team provide their estimations anonymously. Hence they are not bound to any influence from a powerful team member. This prevents the overall estimation from skewing towards interests of a single person.

However similar to Expert Judgment Wide Band Delphi estimation technique has one major disadvantage, dependency on the domain and technical experience of the estimation group. If the members of estimation group are experienced and have worked with similar projects in the past, the estimation results tend to be less deviant from the actual values else the results tend to deviate more from the actual values.

Wide Band Delphi estimation technique provides better estimation results in product line projects where the team members have an understanding of the domain and technology since they execute a series of similar projects. It can also be carried out in service or solutions industry where solutions to similar kinds of problems are provided as services. E.g. Projects for banking domain are more or less the same with some differences.

However, Wide Band Delphi would not be suitable for use in new domains or in research projects where the team members have no previous relevant experience.

5. Conclusion

The correlations attempt to show that although there are several estimation techniques available, certain estimation techniques provide better estimation values for projects based on their characteristics or factors. Further studies could verify the correlation with data from projects. This report primarily shows that a correlation can be developed and could be further investigated.

6. References

[1].Mel Rosso-Llopart. Lecture Critical Concepts about Estimation, 2006.
[2].Pankaj Jalote. Software Project Management in Practice, Addison Wesley, 2002, Section 4.2.3.
[3].Barry W.Boehm. Software Engineering Economics, IEEE Transactions on Software Engineering, 1984, SE-10, 1, 4-21.

[4].Karl E. Wiegers. Stop Promising Miracles/Process Impact, http://www.processimpact.com/articles/delphi.html/reprinted (with modifications) with permission from *Software Development* magazine, 2000.
[5]. Andrew Stellman & Jennifer Greene. Applied Software Project Management, http://www.stellman-greene.com/aspm/images/ch03.pdf, O'Reilly.