Exhibit B

Analogy based Software Estimation

Murali Chemuturi

Introduction

Analogy Based Software Estimation is based on the principle that ***actual values achieved within the organization in an earlier and similar project are better indicators and predict the future project performance much better than an estimate developed afresh from scratch***. It also facilitates bringing the organizational experience to bear on the new projects.

However, to use this technique, it is necessary for the organization to put in place certain pre-requisites, such as –

1. The organization ought to have executed a number of projects
2. The organization should be keeping meticulous records of past projects
3. The organization must be conducting project post mortem for every project and causes for variances must be identified using meticulous methods and the actual values validated depending on the causes. Care must be taken to prevent erroneous data to influence future projects.
4. The organization should have a well organized and maintained Knowledge Repository from which it is feasible to locate similar past projects and extract the validated project data
5. The estimators should be trained in drawing analogies accurately and in accessing the Knowledge Repository and extracting validated data and extrapolate the same for the current project

Once these pre-requisites are in place, this technique can very profitably be used in the organization.

Selection of similar past projects

This is a crucial step in this methodology and all care must be take in short-listing similar projects. First criteria in short-listing is, of course, the type of project, namely,

1. **Full Life Cycle Software Development Project** – normally referred to as Development Project
2. **Software Implementation Projects** such as ERP, Supply Chain Management, CRM etc.
3. **Conversion Project** – convert an application to make it usable in the new circumstances – such as Y2K projects, Euro Conversion projects etc.
4. **Porting Projects** – Porting from one software platform to another such as porting from one version of Unix to another version of Unix
5. **Migration Projects** – Migrate the application from one hardware platform to another – say from Data General to IBM Mainframe
6. And so on

The most crucial aspect for the success of Analogy Based Estimation is the selection of right set of past projects. The following parameters need to be considered.

1. **Application Domain** - This is perhaps the single most important feature to be considered. It would not make sense to draw analogy between two different domains. For example – would it make sense to select a Marketing Information project to draw analogy for a

ORCLX-PIN-000110.001

Material management Information project? Therefore, draw analogy from similar application domain.

2. **Organization size of the prospective client** – The extent of functionality would differ between different sizes of organizations even if the domain is the same. The functionality of Material management, for example, for medium-sized organizations would significantly differ from a large sized organization. Select a past project that is comparable in size with the current project.
3. **Number of Locations of the prospective Client** – The functionality for a single location would be vastly different for a multi-location organization. Therefore, select a past project that is similar in number of locations of the client organization with the current project.
4. **Nature of modules in the application** – the past project selected needs to include majority of the modules that the current project has. We can adjust and extrapolate for extra modules for one or two modules but not for a majority.

The following parameters from the development platform need to be considered –

1. **Number of application tiers** – A two-tier application would significantly differ from a three-tier project.
2. **Backend** – In present day, almost all applications are built with an RDBMS. As long as the backend is an RDBMS in both the cases it can be considered equivalent. However if one of them is flat files and the other is RDBMS, then they would be different
3. **Web Server** – Different web servers cause different amount of work. We may need to extrapolate based on the web servers used. This would be applicable in web based application development.
4. **Middleware** – Different middleware have different impacts on the amount of effort required.
5. **Rules Engines** – if the proposed project uses a Rules Engine, it would be desirable to select a past project that also used a Rules Engine. Also significant is the fact that different Rules Engines would have different impacts on the amount of effort required for software development.
6. **Programming language** – The amount of work is influenced to a large extent by the language in which programs are developed. If the past project used a different programming language than the present project, we may need to adjust the estimate for difference in programming language.
7. **Development environment** – The type of tools used for editing the programs, debugging, compiling etc have a large impact on the productivity of programmers. Hence it is important to select past projects that have similar software development environments.
8. **Software Development Process used** – It is also important to select projects that are similar in the manner of developing software conforming to the process that is likely to be used in the current project.
9. **Location of Development** – Development locations can be either in-house or at client location. It is better to select a past project that used similar location.

Short-listing of past projects

The over riding criteria for selection of projects is the application domain. It is futile to shortlist projects that are in different application domain. Therefore, it is essential to make this the first criteria for short-listing of projects. Then we can short-list the projects in a table such as the one shown below –

32-78/3 Sainik Nagar Rama Krishna Puram, Secunderabad - 500056
+91-40-2722 0771 - www.chemuturi.com - murali@chemuturi.com

Confidential Information   ORCLX-PIN-000110.002

Criteria	Current Project	Project 1	Project 2	Project 3	Project n
Type of Project	Development	Development	Development	Development	Development
Application Domain	CRM	CRM	CRM	CRM	CRM
Organization Size	Medium	Large	Large	Medium	Small
No of Locatrions	2	6	8	3	1
Modules	Purchase, Warehouse, Inventory Control	Purchase, Warehouse, Inventory control, Shipping, MRP	Purchase, Warehouse, Inventory control, Shipping, MRP	Purchase, Warehouse, Inventory control, MRP	Purchase, Warehouse, MRP
Application Tiers	3	4	4	2	1
Backend	RDBMS	RDBMS	RDBMS	RDBMS	Flat Files
Web Server	IIS	Web Sphere	Web Sphere	NA	NA
Middleware	Nil	Web Logic	Web Logic	NA	NA
Rules Engine	Nil	Ilog	Ilog	Nil	Nil
Programming Language	J2EE	Dot Net 03	Dot Net 05	VB 6	COBOL
Development Environment	J2EE	Dot Net 03	Dot Net 05	VB 6	COBOL
Development Process	CMMI 5	CMMI 3	CMM 5	CMM 3	ISO 9000
Location of Development	In-House	In-House	In-House	Client	Client

The above step of short-listing projects can be perhaps, automated as well as perhaps to narrow down the selection to 2 or 3 projects. The estimator should select the one project that is to be the base-estimate for the present project – it is not preferable to automate this step. Here human judgment is necessary to select the project that is -

1. Closest to the present project in the above parameters
2. Needs the least possible extrapolation for arriving at the new estimate

Once the project is selected, now the following steps may be performed to arrive at the new estimate.

1. Make a comparative statement of all parameters between the past project and present project
2. Make adjustments for differences as necessary. There are two possibilities –
    a. Detailed estimates are available for the past project – in this case, we can take the old estimate and tailor it to fit the new project
    b. Only gross estimates are available – in this case, we need to make adjustments to the gross estimate
3. Finalize new estimate and get it reviewed by a peer and implement feedback, if any
4. Arrange for managerial review and implement feedback, if any
5. Present the estimate to originator of Estimation Request using the organizational format

When detailed estimate is available for the selected past project, to arrive at the new estimate, we need to follow the below mentioned steps –

1. Save the old as estimate as the new estimate
2. Examine each of the details of the old estimate and –
    a. If the detail is identical for the current project retain it
    b. If the detail is completely out of scope for the current project delete it
    c. If the detail is partially applicable for the current project, adjust the detail to reflect the reality fo the current project

32-78/3 Sainik Nagar Rama Krishna Puram, Secunderabad - 500056
+91-40-2722 0771 - www.chemuturi.com - murali@chemuturi.com

Confidential Information   ORCLX-PIN-000110.003

       d. Carry out the above three steps for every detail of the old estimate and come out with the new estimate

3. Examine if any of the details that are pertinent to the current project but are totally missing from the old estimate and include them in the estimate to make it comprehensive in terms of details of estimate
4. Now examine the environmental aspects such as the development platform, programming language, number of application tiers, development location, middleware etc and make appropriate adjustments to the estimate

Now the estimate is ready for review. Arrange for peer review and implement feedback, if any. Arrange for managerial review and obtain approval. Now submit it to the originator of estimation request using the standard formats from the organizational estimation process.

Now only if gross estimates are available, you may –

1. Examine each parameter of the new project against the old project
2. Assign weights for each parameter –
       a. A weight of 0 (zero) indicates that this parameter is not applicable at all for the current project
       b. A weight of 1 (zero) indicates that this parameter is identical for the new project
       c. A weight of more than zero but less than 1 indicates that this parameter is applicable but the applicability is less than 100%
       d. A weight of above 1 indicates that this parameter is applicable and the applicability is more than 100%
3. Sum up all the weights and derive arithmetical average to obtain the Composite Weight Factor (CWF)
4. Multiply the gross estimates of the past project with the CWF weights to obtain the new estimate.

Now the estimate is ready for review. Arrange for peer review and implement feedback, if any. Arrange for managerial review and obtain approval. Now submit it to the originator of estimation request using the standard formats from the organizational estimation process. The below table illustrate the procedure.

Let us assume that the selected past project has –

1. A size of 500 Function Points
2. A development Effort of 5,000 Person Hours

32-78/3 Sainik Nagar Rama Krishna Puram, Secunderabad - 500056
+91-40-2722 0771 - www.chemuturi.com - murali@chemuturi.com

Confidential Information   **ORCLX-PIN-000110.004**

Criteria	Current Project	Selected Project	Weight
Type of Project	Development	Development	1
Application Domain	CRM	CRM	1
Organization Size	Medium	Large	0.4
No of Locatrions	2	6	0.4
Modules	Purchase, Warehouse, Inventory Control	Purchase, Warehouse, Inventory control, Shipping, MRP	0.4
Application Tiers	3	4	7
Backend	RDBMS	RDBMS	1
Web Server	IIS	Web Sphere	0.8
Middleware	Nil	Web Logic	0
Rules Engine	Nil	Ilog	0
Programming Language	J2EE	Dot Net 03	1
Development Environment	J2EE	Dot Net 03	1
Development Process	CMMI 5	CMMI 3	1.2
Location of Development	In-House	In-House	1
Composite Weight Factor (CWF)			1.16

Now from the above table the CWF is 1.16. Now the past project values are multiplied by this CWF to arrive at the new estimate values –

1. Size = 500 X 1.16 that is 580 Function Points
2. Development Effort = 5000 X 1.16 that is 5,800 Person Hours

Merits of Analogy based Estimation

1. It is based on actual values achieved within the organization in an earlier project and hence are more reliable than other methods of estimation
2. Easy to learn and very quick to come out with a good estimate
3. For new organizations, they can purchase estimation data from organizations such as ISBSG (International Software Benchmarking Standards Group) and make use of this technique. While these estimates are not from within the organization, they provide variety of estimates to choose from and provide a starting point.
4. This technique facilitates use of organizational expertise and experience to be brought forth for the current project like no other technique of software estimation.

Demerits of Analogy based Estimation

1. The short-listing of past projects and selection of the final project do require meticulous record keeping and software tool support. Any laxity in this step would have serious consequences for the estimate
2. Organization needs to maintain well designed Knowledge Repository and maintain it conforming to meticulous process
3. The current project may not have any relevant past projects at all in the organization
4. This can not be implemented in a new organization

>>
Your Feedback is gratefully accepted – murali@chemuturi.com
>>

32-78/3 Sainik Nagar Rama Krishna Puram, Secunderabad - 500056
+91-40-2722 0771 - www.chemuturi.com - murali@chemuturi.com

Confidential Information   ORCLX-PIN-000110.005