Parametric-Activity Based System Development Method Selection

GJCST Computing Classification
D 2.8, D 2.2

Ajay Jangra[1] Deepa Sharma[2] Parbhat[3] Priyanka[4]

***Abstract* -In software development life cycle many models have been developed to evaluate and improve capabilities. This paper proposed two enhance tables which provide necessary guidelines to the developer/ organisation on decision making regarding selecting System Development Methodology (SDM) approach by "Comparing traditional and object oriented SDM". This work is novel in the sense it identify five new parametric activities from SDLC and evaluating characteristic behavior of corresponding to behavior of traditional and object oriented methodology. Furthermore on considering (complexity, testing effort, cost etc.) five parameters are assigned with some weightage distinguish for both system development methodologies. The required result may depend on organization's decision that how well they create software according to how they define and execute their processes.**

Keywords-System development methodology, SDLC, Parametric activity. Object –oriented system design, user, developer.

I. Introduction

Two important phases of system development are: External and Internal. External development deals with the implementation, planning, preparation of manuals, & installing. Internal Development deals with Software development & performance and testing. Selection process consists of several steps i.e. Requirement analysis, System specifications, Request for proposal, Estimation & validation. The main criteria for software selection depends upon benchmarking which is a evaluation technique where the software purchaser compare the software with other to find the best of Speed & cost by pursue the Reliability, Functionality, Capacity, Flexibility, Usability, Security, Performance, Serviceability, Minimal costs which are the quality factors for SDLC.[1,2,3,9]

II. Litrerature Survey

A methodology is a route for solving the problems of the current system or for structure a new one. There are many methodologies for the design and development of systems which include: Systems Development Life Cycle (SDLC), Object-Oriented Analysis and Design and many others (Dennis, Wixom, Teagarden, 2002)[4]. The SDLC is more commonly known as Structured Systems Analysis & Design. Structured methodologies allow the analyst to break down complicated systems into smaller, clearly defined and more manageable parts. The structured systems Development Methodology life cycle moves toward a step by step procedure that goes from one phase to another. The first object-oriented languages came into existence during the 1960's and 1970's with Simula and Smalltalk. However, it was not in anticipation of several years later that the Object-Oriented Analysis and Design (OOAD) methodology came into being (Larman, 2004)[5]. First in 1982 Object-Oriented Design emerged as independent topic (G. Booch, 1982), and later in 1988 Object-Oriented Analysis was introduced by S. Shlaer and S. Mellor (1988) and S. Bailin (1988)[6]. Many different object-oriented analysis and design methods evolved since then such as J. Rumbaugh (1991), P. Coad and E. Yourdon (1991)[7] and many others.The OOAD methodology uses an object-oriented perception rather than a functional perception as in the SSAD methodology. An object is a person, place or thing initially drawn from the problem domain which has three aspects to it: what it knows (its identity and certain attributes), who it knows (relationships to other objects) and what it does (its methods it is responsible for performing on its data) (Norman, 1996)[8].

III. System Development Methodologies

In paper Tabular guidelines for system development methodology [16] two new C-tables (characteristic & cost (efforts)) were proposed which helps developer/client to select a suitable system development methodology. There are different ways to develop an appropriate system. System development life cycle (SDLC) provides an overall framework for managing the process of system development. Traditional approach and object-oriented approach use the SDLC as a project management framework. There are two main approaches to SDLC: *Predictive* and *Adaptive*. (i) *Predictive approach* assumes project can be planned out in advance (ii) *Adaptive approach* is more flexible, assumes project cannot be planned out in advance. SDLC describes as problem solving methodology which describes software in different stages such as: Organization recognizes problem (project planning), Project team investigates, understands problem and solution requirements (analysis), Solution is specified in detail (design), System that solves problem is built and installed (implementation), System used, maintained, and enhanced to continue to provide intended benefits (support). [1, 2, 14, 15,]

About[1,2]- CSE deptt. U.I.E.T. Kurukshetra University, India.
(phone +91-9466027922, e-mail er_jangra@yahoo.co.in,)
(phone +91 9729596190, e-mail reach4deepa@gmail.com)
About-[3] CSE deptt. J.M.I.T. Radaur. Haryana India.
(phone +91-9896713172, e-mail parbhatverma@rediffmail.com)
About-[4] ECE deptt. K.I.T.M. Kurukshetra, India.
(phone +91- 9466751345, priyanka.jangra@gmail.com)

IV. Tabular Analysis of Parametric Activities of Traditional and Object Oriented Approach

The objective is to develop an effective system which suggests whether to go for a traditional approach or object-oriented approach to develop software according to requirements. Proposed C-table in the next segment briefly analyze two approaches with their activities and stages respectively.

A. Proposed C-tables

Here, we proposed five parametric activities (SDLC) and discuss/ Enlist corresponding characteristics, behavior and functions of traditional and object oriented approach. The proposed C^1 table analyzes functional behavior of traditional and objects oriented approach which aggregates developer vision about the characteristics and behavior of software.

B. Proposed C^1 table for SDLC

The structured approach is well established. There is a lot of CASE tools exist to support development. Most development projects set their own standards that are adopted for analysis and design. The distinct stages make it easier to schedule, distribute work among a number of people. It is easier to express a system in terms of its functions than its data. The structured methods are based on functional decomposition expressed using DFDs. Class diagrams are more similar to ERDs, which are more difficult to model. Entity Relationship Diagrams (ERDs) contain most of the information of the Class diagrams.

Parametric Activities	Traditional	Object oriented
Planning	Define problem and scope. Produce detailed schedule. Confirm project feasibility. resource management	Define problem and scope Produce detailed schedule Confirm project feasibility Staff the project
Analysis	Gather information to learn problem domain Define requirements Build prototypes Generate alternatives Review recommendations	Defines types of objects users deal with Shows use cases are required to complete tasks
Design	Integrate the network Design the application architecture, user interfaces system interfaces and integrate the database and system controls	Defines object types needed Shows objects interaction. Refines the object for implementation with specific language of environment.
Implementation	Construct Verify, test and Convert data. Train users and document and Install the system.	Writing statements in programming language to define what each type of object does.
Support	Maintain, Enhance system Repairs and updates small upgrades expand system capabilities. Support users	CASE tools are designed to help analysts complete system development tasks

Table C^1: Parametric Activity characteristics of traditional and Object oriented methods

The main difference is that classes also define functionality. The development process is both top-down and bottom-up. The problem is partitioned in terms of objects and classes, which is a top down activity. Re-use is considered at all points, during analysis, design and implementation

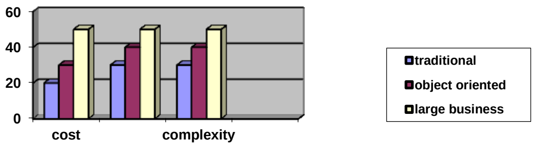

Fig 1

Existing designs, frameworks, patterns, components, classlibraries are considered for re-use. This is a bottom up activity. [10,11,12,13]

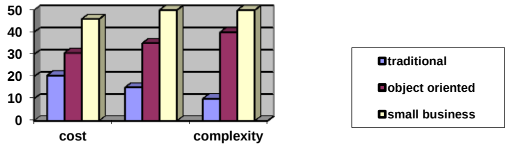

Fig.2

From reference [16] the cost, complexity, testing effort in fig [1,2] required for large and small business software development.The design and implementation will differ only in the level of detail. With corresponding to the above fig. rough estimated weightage are assigned to proposed parametric activities of SDLC and we draw table 2 and fig.3 which shows the cost/effort of both methodologies w.r.t these parametric activities. The effort required with the Object-oriented approach compared to the traditional approach has a difference. Design is much more complex than with traditional development, because of re-use, but coding requires less effort so does testing. Implementation of traditional is more complex than object oriented approach.

V. Proposed C^2 Table

Life cycle stages	Traditional approach	Object oriented approach
Requirement	20	20
Design	10	25
Coding	25	20
Implementation	20	15
Support/Testing	25	20

Table C^2

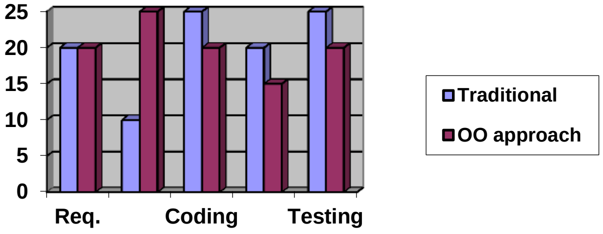

Fig. 3 cost (efforts) v/s Parametric activities for traditional and Object oriented methods

VI. Conclusion

Traditional and OOSD are completely different in many terms. With the help of proposed C tables (C^1,C^2) we found OOSD is complex at design time and structured approach is simple. User/developer agreed on selection of software development methodology on the basis of tables. The proposed work helps in planning, staffing, organizing to developer so that he may easily estimates the development level base requirements(resource, efforts). this work helps to forecast the required development efforts and resources in advance which helps developers to manage the software development process efficiently.

VII. Referance

1) Daisuke Horie, Toshio Kasahara, Yuichi Goto, and Jingde Cheng, A New Model of Software Life Cycle Processes for Consistent Design, Development,Management, and Maintenance of Secure Information Systems, Eighth IEEE/ACIS International Conference on Computer and Information Sciences,2009
2) Mohammad A.Rob, “Issues of Structured VS. Object Oriented Methodology of System Analysis and Design, Issues in Information Systems,2004
3) Kenneth Pefkaros, “Using Object Oriented analysis and design over traditional structured analysis and design”, International Journal of Business Research, March,2008
4) Dennis, A., Wixom, B., and Tegarden, D., Systems Analysis and Design An Object-Oriented Approach with UML, John Wiley & sons, New York, New York, 2002.
5) Larman, C., Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed., Prentice Hall, Upper Saddle River, New Jersey, 2004
6) Bailin, S., Remarks on Object-Oriented Requirements Specification, Computer Technology Associates, Laurel, MD, 1988
7) Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W., Object-Oriented Modeling and Design, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
8) Norman, R., Object-Oriented Systems Analysis and Design, Prentice Hall, Upper Saddle River, New Jersey, 1996.
9) Craig Larman, Victor R. Basili, "Iterative and Incremental Development: A Brief History,"Computer, vol. 36, no. 6, pp. 47-56, June 2003
10) Walt Scacchi, Institute for Software Research, University of California, Irvine “Process Models in Software Engineering”, Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, Inc, New York, December 2001
11) Scacchi, W., Understanding Software Process Redesign using Modeling, Analysis and Simulation. Software Process --Improvement and Practice 5(2/3):183-195, 2000.
12) Booch, G., "Object-Oriented Design", Ada Letters Volume 1 (3), 1982, 64-76.
13) Bailin, S., Remarks on Object-Oriented Requirements Specification, Computer Technology Associates, Laurel, MD, 1988
14) Coad, P., and Yourdon, E., Object-Oriented Analysis, 2nd ed., Yourdon Press, Englewood Cliffs, New Jersey, 1991
15) Mauro Pezz`e, Michal Young, Testing Object Oriented Software, 26th International Conference on Software Engineering (ICSE’04)
16) A.Jangra, Deepa Sharma, J.Singh, Priyanka,“Tabular guidelines for system development methodology”, international Journal of Ultra Scientist of Physical Sciences, ISSN : 0970-9150(2010) Accepted for publication in vol.22 no. 2 aug.2010.