Validated Prototyping-Based RAD Model Using the Evolutionary Approach

SalehMesbah, Nader Nada, EssamKosba, and NesrineElfeky
Faculty of Computing & IT, AASTMT, Alexandria, Egypt.

***Abstract*--Software development today is a complex area, in which new problems and challenges constantly arise, competition is rather high, and it is difficult to maintain high position on the market. This paper aims to introduce and develop a validated prototype-based RAD model using the evolutionary approach with the purpose of minimizing the risk of changes in software engineering and to have a product that meets its expectations in terms of functionality, cost, quality, and delivery schedule. Rapid application development (RAD) is created to present a higher level of responsibility that includes both a business unit requirement capturing and testing. Contactand interview case study participants approach are taken in deriving and validating the proposed model. Then, survey instruments undergo for testing and evaluation process for measuring the effectiveness and efficiency of the proposed model. The case study results showed reduction in the project risk to be behind of the delivery schedule, cost of software development, improve the quality of the system and deliver software that actually meets the needs of users.**

Keywords- Software Prototyping, Rapid Application Development (RAD), Evolutionary Prototyping

I. INTRODUCTION

Software development today is a complex area, in which new problems and challenges constantly arise, competition is rather high, and it is difficult to maintain high position on the market. The software industry adds more than US$260 billion in value to the US economy in 2007 [1]. In Egypt, IT investments are expected to increase from US$ 1.4 billion in 2010 to US$ 2.6 billion by 2014 [2].The Systems Development Life Cycle (SDLC) is generally interpreted as the set of procedures that transform inputs into outputs and as the method by which information systems are built. Organizations face challenges when attempting to build or improve the SDLC in response to the changing business needs [3].

Rapid Applications Development (RAD) was developed to present a higher level of responsibility that includes both a business unit requirement capturing and testing. RAD was an actual response to meet the constantly changing business needs [4]. RAD uses continuous business user involvement that requires significant principles[5] [6].

Software prototyping is a way of showing the idea behind the designed software in an inexpensive way. A prototype can be changed easily and the design evolved gradually to an optimal solution.Requirements can be tested, and problems are identified early before the system is implemented. The idea behind prototyping is to save on the time and cost to develop something that can be tested with real users [7].

II. OBJECTIVES AND PAPER ORGANIZATION

The objective of this paper is supporting the software development organizations facing the challenges in an effective and efficient way to deploy new quality application software in a timely fashion while maintaining flexibility to adapt to the business environmental changes in a highly competitive marketplace. This paper represents a validated software prototyping based-RAD model using evolutionary approach. Contact and interview the case study participants approach are taken in deriving and validating proposed model, then testing and evaluation process for measuring the effectiveness of this model. Data Management Systems (DMS) organization applied the proposed model for test and validation. It suggests that the proposed model, when employed properly leads to improved software quality and reduce time and cost.

The research has been conducted by developing a model for RAD software prototyping. A software prototyping RAD model using the evolutionary approach has been developed. Cast study and interviews instruments are chosen as the most appropriate method of data collection. The study is based on percentage ratio and the resist the temptation to cover before and after case.

The paper is organized into four main sections. Section (1) presents a general introduction about RAD prototyping. This part also lists the research problems and objectives. Section (2) gives a brief background about SDLC and RAD software prototyping and introduces related works in the area of software prototyping. Section (3) describes the details of the proposed model. Section (4)explains the case study and the results of the evaluated model.

III. SOFTWARE PROCESS MODELS

Every software engineering organization should describe a unique set of frameworks (models) activities for the software process it adopts. Software process model is a simplified description of a software process that presented one view of that process. Software process is the set of activities and associated results that produce a software product. There are four fundamental process activities:

software specification, software development, software validation, and software evolution [8]. Most software process models are:

Waterfall model: It is known as cascade from one phase to another. It is considered as the classic approach to SDLC.The waterfall model describes a development method that is linear and sequential [8] [9]. The advantage of waterfall development is that it allows for departmentalization and managerial control[10]. The disadvantage of waterfall development is that it does not allow for much reflection or revision. The waterfall model should only be used when the requirements are well understood and unlikely to change radically during system development.

Evolution model: Evolution approach is more effective than the water fall in producing system that meet the immediate needs of customer. It is based on the idea of developing an initial implementation, exposing this to user comment and refining it through many versionsuntil the adequate system has been developed. The advantage ofthe evolution approach is that the specifications can be developed incrementally that allow users to develop a better understanding of their problems.The disadvantage ofEvolution approach, managers need regular deliverables to measure progress, lack of process visibility, systems are often poorly structured, and special skills may be required[8].

Incremental delivery: In this approach,the development and delivery is broken down into increments with each increment delivering part of the required functionality. User requirements are prioritised and the highest priority requirements are included in early increments. Incremental approaches vary in aspects such as the recommended iteration length, the amount of up-front specification work, or emphasis on feedback and adaptation-driven development [11]. Incremental development has the advantagesthatcustomer value can be delivered with each increment so system functionality is available earlier. Early increments act as a prototype to help elicit requirements for later increments. Reduce risk of overall project failure. The highest priority system services tend to receive the most testing[3].

Spiral development: The spiral model is intended for large, expensive, and complicated projects [8]. The advantage ofSpiral model is estimating issues become more realistic as work progresses, because important issues are discovered earlier it is more able to cope with the changes that software development generally entails, and software engineers can get their hands in and start working on a project earlier [9].The disadvantage is that theSpiral model is highly customized limiting re-usability, applied differently for each application, and risk of not meetingbudget or schedule. Spiral model emphases on risk analysis which require highly specific expertise also it don't work well for smaller projects.

Prototyping: Prototyping is the process of developing a trial version of a system or its components or characteristics in order to clarify the requirements of the system or to reveal critical design considerations. Prototyping was seen as the solution to the requirements analysis problem[8] [12].Model prototyping can be an effective aid in determining how effective and ideal the features and the overall design can be for proposed users. Model prototyping and the subsequent testing procedures can provide useful representation information that can reveal possible design flaws, effective features as well as useless functions that can be corrected, enhanced or done away with on the next stages of product development [13].

Process of software prototyping development:

A process model for prototyping development has four stages: establish prototype objectives, define prototype functionality, develop prototype, and finally, evaluate prototype [8][14].Software prototyping has many variants. However, all the methods are in some way based on two major types of prototyping [8].

1. Evolutionary prototyping is an approach to system development where an initial prototype is produced and refined through a number of stages to the final system. The objective of evolutionary prototypingis to deliver a working system to end-users.
2. Throw-away prototyping: The prototype is developed from an initial specification, delivered for experiment then discarded which helps in reduce the requirements risk. The objective of throw-away prototypingis to validate or derive the system requirements.

Rapid Applications Development (RAD):

RAD is a development lifecycle designed to give much faster development and higher-quality results than those achieved with the traditional lifecycle. It is designed to take the maximum advantage of powerful development software that has evolved recently [4][15].RAD takes advantage of automated tools and techniques to restructure the process of building information systems.

Rapid Software Prototyping:

Rapid prototypingis prototyping activity which occurs early in the software development life cycle. Rapid prototyping, when employed properly, leads to improved software quality. The primary improvements are ease of use, better match with user needs, and often better maintainability [15].

Comments:

Through the previous review, it has been realized that:

- Some of the existing management process assumes a waterfall model for development which leads to reducing the chance of identifying the problems in early stage which affect cost, time, productivity, and quality.
- No main framework exists for software prototyping with real testing phase or follows up stage for determining the problem in real testing and finds available solutions.This made a contradiction between developers. So it is better to take advantage of one framework and to identify the problems early before the system is implemented.

This paper aims to develop a validated software prototype-based RAD process model using the evolutionary approach with the purpose of minimizing the risk of changes in software engineering and to have a product that meets its expectations in terms of functionality, cost, quality, and delivery schedule.This paper concentrates onevolutionary prototyping as it is used in the RAD approach.

IV. A PROPOSED VALIDATED PROTOTYPE-BASED RAD MODEL

The proposed prototyping model follows four phases: Analysis, design, validate, and deliver as shown in Figure (1). The planning, designing, and implementation phases are common to most of models although the focus and the approach to each phase of the life cycle may differ. Each one of these phases has multiple steps:

- Analysis phase: requirements analysis and verify the requirements.
- Design phase: prototype interface and task flow, determine the content and characteristics, and determine architecture prototype components.
- Validate phase: user approval and acceptance in real context.
- Deliver the system: deliver the system.

Planning phase: The first important activity is determining the requirements. These requirements are identical to the software requirements and prototyping requirements. Prototyping requirements influence both task flow and prototyping content.

Designing phase:

- A task flow depicts the steps (scenarios) which users perform to complete a task or several tasks.Task flow also shows the order and the dependence of tasks. Mapping the requirements and interface them allocate tasks in a sequence screens. The task flow should be clearly defined by determining each task goal, use case, or passing or data objects.
- Determining the content and characteristics based on how much time and recourses needed to be spending on the prototype. Choose the correct content avoid misunderstanding of what the prototyping is trying to show. Define characteristics is important because they inform the developers how to prototype

After the prototyping is created, software team makes an internal review to collect feedback regarding the prototype. To achieves the best results from reviewing the design by identifying the goals of presented prototyping, the audience and what they expect, presentation method, and the level of the finished design.

Validate:

This phase includes a test plan of what and how it will be tested, with individual test cases, along with successful completion, which satisfy the plan, and map directly back to the requirements. In many service industry shows high dependencies on the context factors such as location, situation, weather, temperature, network connection, and user's profiles etc., which has to be considered and tested in real life. This means that real acceptance tests and satisfactory leads to implementing the software by getting feedback from users, if the developed prototype does not match the user's expectations, the process goes back again where all changes are taking place or in case of real test failure the process of determining the problems and the available solutions takes place. The solutions take place if there is no need to be approved by the stakeholders. Ifthe proposed solutions need to be checkedforstakeholders' acceptance and find alternative solutions for the problem, this may lead to return to the design or planning phase. If the stakeholders refused all the suggest solutions the project will be rejected.

As soon as system testing is completed, software acceptance testing is performed. Software acceptance testing should be performed by the system's end users or a group of test specialists with assistance of the development group. The test cases and data should be derived from the system external specifications and the acceptance criteria. Once the software system has been accepted by the user, formal certification of the system should be issued.

System Delivery:

The last step is transforming the prototype into areal delivered software product. The following are the steps that should be taken to make a successful delivery of the system:

- Implementation strategy
- Install , configure new software
- Enter data to the new system
- Test processes in conjunction with system.
- Schedule training
- Plan how the system will go live (all at once or by module)

V. CASE STUDY IMPLEMENTATION

This research seeks to answer the question of how to remain competitive and ensure software meets the expectations of customers; partners, and shareholders, businesses need by improve quality, productivity, reduce cost and time, and increase the developer satisfactory. The research methodology is based on technology acceptance model theory and IS success model theory according to certain dimensions; information, system and service quality, use, user satisfaction, and benefits [16]. An interview form is chosen as the most appropriate method of data collection in this study, and is based on percentage ratios and the resist the temptation to cover the case before and after.

Interviews are performed in a semi structured fashion. Participants are presented with an outline that listed the topics to be covered. Participants were asked a number of standard questions. The interview began with the participant describing history with the company. Each person was then asked to give a more detailed description of his/her personal experience with the initiative. At this point only clarifying questions are asked. As the interview

progressed, questions are asked that required progressively more speculation. Each interview ended with the participant assessing the key successes and failures of the initiative. Quantitative data are obtained from the company. To protect confidentiality some data have been either normalized or disguised.

At the beginning stage of the implementation phase, the case study participants based on four organizations are chosen as lead companies in the market. After contacting the case study participants, two of the recommended companies haven't any suitable projects to test the proposed model. The third companywas facing unstable user requirements due to contract problems. As a result for this management gap, it was difficult to implement the proposed model before solving those problems. The last company (Data Management Systems DMS organization) is found to be appropriate to implement and test the proposed model. DMS is one of the lead software developing companies in Egypt, as it has a good management and varieties of different project types

Data Management Systems (DMS) is a powerful medium size company thatis doing very well in the market. DMS proudly describes itself as the only Egyptian company that has mass marketed its 100 percent Egyptian software internationally. In addition, DMS is the only company in the Middle East to own fully integrated management systems for all the major fields (http://www.dmsegypt.com) [17].DMS is the first Egyptian company to be appraised for CMMI level 4 by Software Engineering Institute (SEI).

In this case study, DMS demonstrates the RAD prototyping model when developing software prototypes on a setup information synchronization system. The project output is report viewing utility. From the interviews, the main problem in the project, the user can't understand his needs and visualize the end product. The size of the project consider as medium-small project (less than or equal 500.000 lines of code), and the tools used visual studio 2008, .Net Framework 3.5, and Database independent. The development environment is MS windows XP with full test in windows 7 environment. According to the company historical data the expected project life cycle time is 35 ± 5 working days by each team member.

The actual work time in number of days is 23 working days by each team member The development team size assigned for the project three members (one senior and two juniors)

Simple statistics formulas are used to show the differences of cases between before and after using the proposed model. According to the company historical data the expected project life cycle time μ is 35 days by each member team with standard deviation σ could be ± 5. After implementing the proposed model the actual work time in number of days x is 23 working days by each team member. The difference Δ show a reduction by 12 days for each team member which is greater than 2σ.

VI. RESULTS AND DISCUSSION

DMS applied the proposed model for projects test and validation. The most benefit gained was reducing time, which is reflected in cost. The proposed model succeeded to improve quality and to reduce cost and time by 20%, enhanced the developer's satisfaction and improved the quality of the system by 25%. DMS reported that the highly software development satisfaction after implementing the proposed model has been influenced by various factors as follow, in order from most to least:

- New experience for the developers
- Nature of the project
- Nature of the customer

The case study was chosen by DMS and it does not face any problem in the real testing. The case study didnot use any COTS or reusable components.

In this research, we did not find support for implementing the proposed model in large scale due to lack of time and project types that need to use software prototyping, and management problems. Extend the implementation in large scale will reflect the advantage of using the proposed model. The proposed model needs also to be tested on large scale projects.

ACKNOWLEDGEMENT

The Department of Technology and Research at Data management Systems (DMS) involved in this study. Eng. Ahmed Moharam (Manger of Technology & Research Dept.) was very interested in the research topic and his team provided the information about the processes inside DMS. After a number of meetings we validate our model and collect the results.

REFERENCES

[1] OECD-2008, "Software Industry Facts and Figures", http://www.bsa.org/country/Public%20Policy/~/media/Files/Policy/Security/General/sw_factsfigures.ashx, Last accessed June 1, 2011.

[2] GFS Finance, 2010, http://www.finance.greenfuelspot.com, Last accessed May 1, 2011.

[3] C. A. Crabtree, "Presenting a conceptual model for the systems development life cycle", M. Sc. Thesis, University Of Maryland, Baltimore County, 81 pages, 2007.

[4] P. Beynon-Davies, C. Carne, H. Mackay and D. Tudhope,. "Rapid application development (RAD): an empirical review", European Journal of Information Systems, Volume 8, Number 3, 1999 pp. 211-223.

[5] C. Andrea and H. Elias, "Development of E-government services for cultural heritage: Examining the key dimensions", International Journal of Technology and Human Interaction, Volume 3, Number 2, pp. 45-70, 2007.

[6] H. Cuifeng and T. Yanyan, "Risks and risk management in the development of

information system", Second International Conference on Communication Systems, Networks and Applications, June 29 2010-July 1 2010, Hong Kong, pp. 345-349, 2010.

[7] J. Z. Guan and Luqi, "A software prototyping framework and methods for supporting human's software development activities", ICSE third International Conference on Software Engineering, Portland, pp. 114-121, 2003.

[8] Sommerville, "Software Engineering", 9th Edition, Addison Wesley, 2010.

[9] W. Scacchi, "Process models in software engineering", in J.J. Marciniak (Ed.), "Encyclopedia of Software Engineering", 2nd Edition, John Wiley and Sons, Inc, New York, 2001.

[10] D. ChenalandP. Schwartz, (2007), "Improve Your Software Development Lifecycle Process: Practical Tips and Guidelines", Published by StraightForward Tools and QAVantage, http://qavantage.toolsforproductmanagement.com/sdlcpaper.pdf , January 1, 2007.

[11] P. Mohagheghi, "The Impact of Software Reuse and Incremental Development on the Quality of Large Systems", Ph.D.Thesis, Norwegian University of Science and Technology, Norway, July 2004.

[12] G. Wang, "Definition and review of virtual prototyping", Journal of Computing and Information Science in Engineering, Volume 8, Number 3, pp. 220-232, 2002.

[13] L. Bernstein, "Importance of Software Prototyping", *Journal of Systems Integration –special issue on Computer Aided Prototyping*,V 6. No. 1, 1996, pp. 9-14.

[14] J. Arnowitz, M. Arent and N. Berger, "Effective prototyping for software makers", Morgan Kaufmann, 2007.

[15] K. Zhang, G-L Song and J. Kong, "*Rapid Software Prototyping Using Visual Language Techniques"*, IEEE International Workshop on Rapid System Prototyping, 2004, pp. 119-126.

[16] W. H. DeLone and E. R. McLean, (2004), "Measuring E-Commerce Success: Applying the DeLone& McLean Information Systems Success Model", International Journal of Electronic Commerce, Volume: 9, Issue: 1, 2004, pp. 31-47.

[17] Data Management systems (DMS), http://www.dmsegypt.com/ , Last accessed July 1, 2011.

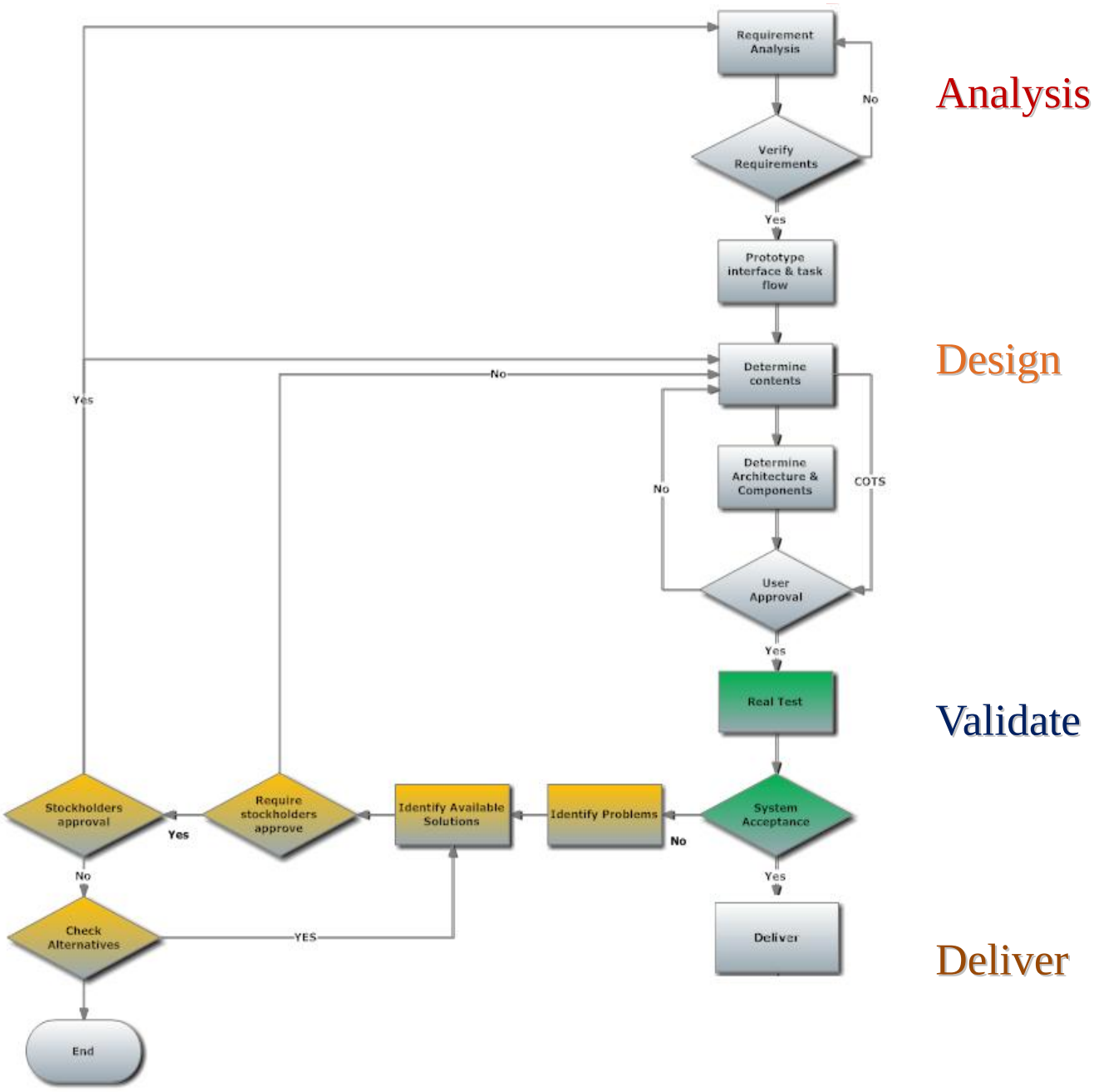

Figure (1): Prototyping based RAD model