IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Software Process Models and Enterprise Resource Planning System

Basem Zughoul

Department of Software Engineering, Faculty of IT, Aqaba University of Technical, Aqaba, Jordan

Abstract: The field of software engineering is related to the development of building software. The comprehensive systems or Enterprise Resource Planning systems need for systematic development unlike simple programs which can be developed standalone and there may be without any systematic approach. Nowadays, the software industry is an independent market, whereas the world is facing another information technology revolution, which is the biggest and most complex one after the industrial revolution. The increase of the popularity of the internet, WWW and the new developments in mobile computing and artificial intelligence are changing the business requirements and practices. This research deals with a critical and important issue in computer world. It is concerned with the software management processes that examine the area of software development through the software process models, where these models will guide the work and influence the quality of the final software by using several development techniques, which are known as software development life cycle. It represents different development models such as the waterfall, iterative, incremental, spiral and extreme programming. These models have advantages and disadvantages as well. Therefore, the main objective of this research is to represent different models of software development and which of these models can be used to develop ERP systems according to the ERP characteristics to be more adaptive with environment change based on business process concepts.

Keywords: Enterprise Resource Planning, Software Process Model, System Development Life Cycle, Extreme Programming System.

I. INTRODUCTION

Any application on computer runs through software. The computer technologies have changed tremendously lately, in addition to the significant changes in software development. In the early years, the software was developed either by a single programmer or by a small programming team according to the size of project. The development of these programs was dependent on the programmer's skills and no strategic software practices. Late 1980s, the size of software and the application domain of software increased in addition to the increased of complexity. Consequently, the software development required bigger teams were engaged and more organized. Software has been developed from tools used for analyzing information for solving a problem to product software itself. Software consists of documents and programs that consist of a collection that has been established to be a part of software engineering procedures. Moreover, the objective of software engineering is to create a suitable work that constructs high quality programs [1]. For that, there is requirement for standard set of steps, techniques, and tools to solve these problems for building software, called a software process model. It is used to develop and support their information systems and the development of these systems often follows software development life cycle [2], [3], [4].

II. ERP SYSTEMS

ERP system solutions are currently in high demand by both manufacturing and service organizations because they provide a tightly integrated solution to an organization's information system needs [5]. ERP systems have been known as systems that bring integration to several business activities within complex organizations, ERP systems are becoming a standard information system, irrespective of the size and nature of the organization, so the drives for ERP systems are technological and operational factors [6]:

- Technological factor is replacement of disparate systems (hardware/software) because of difficulty of maintenance, integration of processes and systems, and the software does not meet business needs.
- Operational factors are customer satisfaction, improvement of process, simplification of complex process, and standardization of process.

ERP system has the ability to automate and integrate the business processes of organization, to share common data and practices across the organization, and to produce and access information in real time [7].

Nowadays, the world is facing another revolution biggest and most complex after industrial revolution, which is the information technology revolution. The increase of the popularity of internet, WWW, new technological development in mobile computing and artificial intelligence is changing the business requirements and business practices. ERP system are customizable, standard application software that includes integrated business solutions, providing a significant improvements in efficiency, productivity and service quality, and reduce in service costs as well as to more effective decision-making [8].

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

III. SOFTWARE PROCESS MODELS

A software process model is a simplified representation of the process. It presents a description of a process from a particular perspective such as: (1) specification, (2) design, (3) validation, (4) evolution [1]. The software development will guide the work and influence the quality of the final software by using several development techniques. These models need close communication between the client and the software development team to minimize the understanding gap [9].

IV. GENERAL SOFTWARE PROCESS MODELS

Software development methodology or software process model is set of standard steps using by most organizations find it beneficial to develop and support their information systems, such as many processes, so the development of these systems often follow a life cycle called Software Development Life Cycle(SDLC). SDLC is a common methodology for software development in many companies. It uses a set of activities conducted for each project through many phases, which make up the effort of the progress of the system analysis and design [2]. It is one of the key underlying concepts in information system development in many organizations having numerous variations of software process models.
ERP systems life cycle models represent the various stages throughout project development. Any system development life cycle should encompass project definition, system study, design, programming, installation, and post-implementation stages. The approach commonly uses a waterfall method where those stages are executed sequentially and only once for each system; and then a prototyping where the stages are repeated; and finally refining as an initial solution until it can be used as the complete system [10].

Moreover, there are many common software models used to direct the software project based on the life cycle concept. But one software process model is not necessarily suitable for use by all projects [11]. The most common process models are: waterfall, iterative, prototype, incremental, spiral, reuse oriented development, RAD, Rational Unified Process model (Rup), and agile methodology - eXtreme Programming (XP) [2], [12]. However, each model is designed for a specific purpose and most of them has similar aims and thus share many common tasks. The section will describe the structure of each model and present the advantages, disadvantages.

Waterfall Model

Waterfall is the traditional and classical model used for software development, Sometimes called the classic life cycle or the waterfall model, the linear sequential model, it is the first software development process model derived from engineering process (Royce, 1970), this model contains six discrete phases (initial investigation, requirements definition, system and software design, development and testing, implementation, and operation and support) [13], [14] (see Fig. 1).

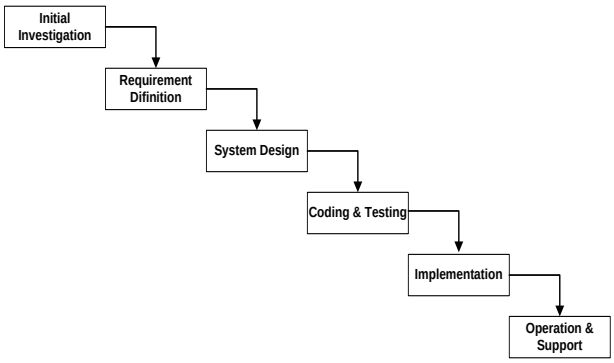

Fig 1. Waterfall model

Advantages: ease of planning and integration, ideal for supporting less experienced project managers through offering easy way to manage and control the work; the progress of system development is measurable, strict on orderly sequence of development steps to help insure the quality, reliability, and the maintenance of the developed software [12].

Disadvantages: it is not possible to collect all current and future requirements from system user at beginning; required a set of plans written and approved by the users; this model is complex, inflexible to modify; a project progresses is time consuming, slow, costly and difficult to respond for later changes in the life cycle; existence of a gap between specification and system deployment; problems are often not discovered until system testing; inherent risks if one step contains an error or is not completed [11], [15].

Iterative model

The difficulty with the waterfall model created order for a new method of solving problems of waterfall model and which could provide faster results, require less in advance information, and offer greater flexibility. This address in the iterative development model, the main idea of this model is based on divided project into small parts like mini-waterfall process. This allows the development team to show results earlier in the process and get valuable feedback from system users [13] (see Fig. 2).

Fig. 2. Iterative process model

DOI 10.17148/IJARCCE.2016.55126

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Advantages: user community is involved actively throughout the project; the project is breaking down into smaller parts for less complexity; and the feedback from system users [11].

Disadvantages: user community actively involved throughout all project phases; this requires more time for the staff, and that means adding more time for a project period; this model needs more communications and coordination skills in project development, and the formal requests for the project improvements after each phase may lead to confusions in controlled mechanism for handling requests to be developed; no proper documentation is made for the projects even in current or future; and fast development often leads to poor quality results [11].

Prototype Model

The framework of this model is iterative, and proposes to integrate with other models like waterfall to make them more effective. Prototyping model is used to display all possible application systems and provide communication between the user and developer. The phases of this model are developers listen to customer's needs, develop some demos, rebuild and these are evaluated by users until they have been accepted the design from users. This model is used when it is difficult to know all requirements at the beginning of the project [13] (see Fig. 3).

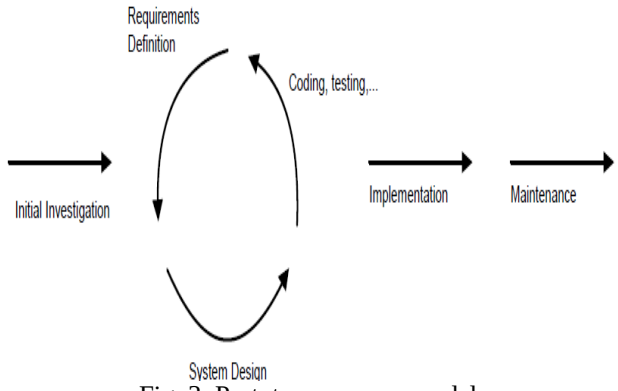

Fig. 3. Prototype process model

Advantages: more effective than the waterfall model in producing a system which meets the needs of customers; provides an early vision of the solution; allows the customer to find more requirements and participate in the process; useful for customization package; rapid feedback across activities; clarification of user for the best practices of a design process [16].

Disadvantages: poor structure; the process is unclear while the system is developed quickly and not cost effective to produce documents; requires special tools and techniques; poorly designed system can lead to a quick and poor quality system; not full integration of the system.

Incremental Model

The framework type is a combination of linear and iterative. It combines the elements of the waterfall model with the iterative of prototyping. This process supports a software engineer to develop more comprehensive software [13]. The objective of this model is used to reduce the inherent project risk by breaking the project into smaller components or increments. The phases in this model are: communication, planning, analysis, design, implementation, integration, testing, and customers' evaluation. Each iteration process and feedback loops produces a new increment until the product is complete [9], [16].

Advantages: customers can get important functionality early, initial products are delivered fast, more flexible than waterfall and less cost to change scope and requirement, help to manage the risks of the project during the iterations, gradual implementation, production more quickly and less number of developers, reducing rework in the process [11], [13].

Disadvantages: requires good planning and design, required well-defined interfaces, difficult problems tend to be pushed as a management problem. Furthermore, the validation for each phase and the overlapping stages is not considered [11], [16].

Spiral Model

The framework type is a combination of linear and iterative. It includes the best features from the waterfall and prototype models, and introduces a new component [11]. This model repeated the process bypasses through iterations in this model the angular component represents progress and the radius of the spiral represents a cost. In addition, this model is used for more negotiation to resolve more conflicted requirements and achieves better resolution.

According to [14], [15] presented that this model is proposed by Barry Boehm in 1988. It was designed to combine the best features from its iterative nature of prototyping and the controlled and systematic aspects of the waterfall model. It introduces a new component risk-assessment as a step to evaluate each version of the system and determines whether or not continues development. Also, this model is presented as a spiral; each loop in spiral represents into four sectors (see Fig. 4).

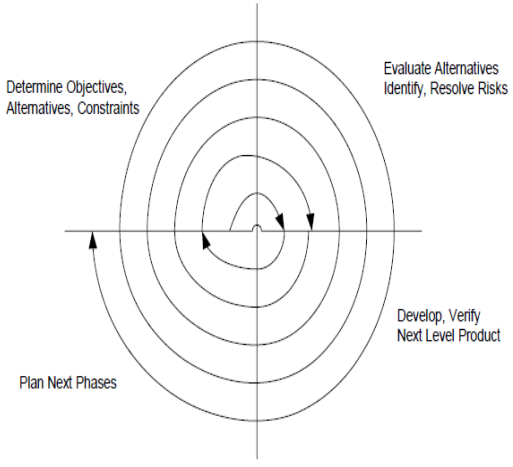

Fig.4. Spiral model

Advantages: high analysis to mitigate project risk, provides early indication for quality assurance through prototyping at each stage in system, combination of waterfall, prototyping and incremental model as special cases in the framework, the combination gives early feedback from users and produced the software early [1], [11].

Disadvantages: high cost for more customization and quite complex, needs specific expertise in risks analysis, not suitable for small projects, no controls for moving from one cycle to another, so each phase may generate more works for the next cycle; they have no specific deadlines, also there is a problem when requirements of users are unclear [1], [16].

Rapid Application Development Model (RAD)

The Rapid Application Development (RAD) is iterative model based on short incremental cycles of business priorities and customer requirements. The phases of RAD are: requirements and planning, user design, construction and cutover phase. RAD relies on joint activities such as Joint Requirement Planning (JRP), Joint Application Development (JAD) and Database Management Systems (DBMS) to complete the software. It focuses on development instead of analysis, design and feedback from users and covers all phases of the waterfall at a fast rate [9], [13], [16].

Advantages: to complete user's requirements with fast development, costs saving and less human efforts, encourages customer's feedback, customer participates throughout the complete cycle, allows customization to package, uses modeling concept to achieve a high degree of flexibility and maintainability, provides the integration at the beginning, manages the changes in system design as demanded by users [11], [16].

Disadvantages: fast development and low cost may lead to poor quality; requires highly skilled developers and designers; hard to use with legacy systems; risks for never close the project problems; a lot of needs to customer participation; difficult with module reuse for future systems; not suitable for cheaper projects and for safety and critical projects.

Reuse-Oriented Model

Ruse Oriented Model is system that assembles from the existing components. It is also been viewed as a concept for rapid systems development. This kind of models called Components Based System (CBS) that build or develop software applications from existing software components or parts. These components are glued together to compose an application. The basic idea behind the reuse model is software developed in parts, this software parts are called components. In addition, each component represents a part of the system to complete functions and it is designed to serve a particular purpose [13]. The phases of this model are: requirements, analysis and design, development and testing, and release and deployment phases [17].

Advantages: faster delivery of software to the organization: reduces the cost and risks: reduces the amount of software to be developed: reduces the need for maintenance; it provides efficiency and reliability.

Disadvantages: it does not meet the real needs of users because of reducing the requirements are a must: lost some control over the system evolution: limited for use in object-oriented development environment and the low performance is a major problem with this kind of applications.

Extreme Programming (XP)

Extreme Programming (XP) is an agile software development methodology that handles rapid changes in requirements and environments in short increments and release cycles. Also XP can be described as a development method for projects where not all requirements and features of a system can be defined and planned before the development start [18]. The phases of XP model are listed below as [2]:

- Exploration phase - customers accumulate the requirements for the first release, then the project leader and platform owner estimate how many persons per day.
- Planning phase - platform owner and customers to discuss the requirements.
- Designing phase - change requires parts to the system, so this part of the daily business of all programmers in XP.
- Coding phase - the code is used to learn about the system itself, communicate solutions, algorithms description, and express tests.
- Testing phase - all tests must be executed to cover all functions and procedures

Advantages: fast development and cost saving, produce better team cohesion and cooperative, emphasis on the final product, integrate with traditional development methods, iterative process, means the short gap between system releases, simplicity and rapid feedback, the testing based on requirements and quality assurance [19].

Disadvantages: fast development leads to poor quality software, needs experience and skills, costly by using some tools as programming pairs (two programmers), difficult to prioritize changes, spend time between customers and development team will lead to success, and test case construction needs specialize skills.

V. ERP SYSTEM CHARACTERISTICS

ERP system has a set of specific characteristics, so these characteristics is derived and based on the literature search on ERP system characteristics in general and the comparisons with other software solutions. Therefore, in this work the unique features of ERP project were identified and used as a help to understand what they are, what they can do, how they differ from other IT packages, and what characteristics of an ERP development methodology [15].

There are several specific characteristics affect the ERP life cycle: complexities, flexibility, software packages, integration, best practice, completeness, openness, and change process [15], [20], [21], [22], [23], [24].

VI. EVALUATION SOFTWARE PROCESS MODELS FOR ERP SYSTEMS

The evaluation of software process models to develop ERP system according to system characteristics, where the scales are High, Medium and Low. High means that this process model covers the characteristic as shown below (see Table I).

TABLE I EVALUATION OF SOFTWARE PROCESS MODELS AGAINST ERP CHARACTERSTICS

ERP Characteristics	Software Process Models							
	Waterfall	Iterative	Prototype	Spiral	Increment	RAD	OO	XP
Integration	High	Medium	Low	Low	Medium	Low	Medium	Medium
Complexity	High	Medium	Medium	Medium	Medium	Medium	Medium	Low
Completeness	High	Low	Low	Low	Low	Low	Medium	Low
Flexibility	Low	High	High	High	High	High	Low	High
Best Practice	Medium	High	High	High	High	Medium	Medium	High
Software Packages	Low	Medium	Low	Low	Medium	Low	Low	High
Openness	Medium	Medium	Low	Low	Low	Low	Medium	Low

VII. CONCLUSION

The following conclusions can be drawn from the present study, that ERP is known as an integrative information system that supports the work processes and resource management of an organization. These software systems are still highly risk and complex because it contained several applications, large size, high cost, large development teams, and usually required limit time schedule. The implementation process of these systems associated massive change in organization structure, business process, and people work. The development of these systems is usually requiring substantial investments of time, cost, and efforts together with a redesign of work process. Therefore, ERP systems are more than just new software solution, which is an application software package developed by independent software venders.

In general, most software process models working as a standalone model did not achieve all or most of the main characteristics of ERP. Therefore, a need to establish a model to achieve most of main characteristics of ERP that depends on the integration of current software process models to meet the general requirements and needs to organization

REFERENCES

[1] Akhilesh. "Assessment of Software Process Models." International Journal of Engineering and managment Sciences, vol. 3, no. 4, 2012: 500-506.

[2] Hoffer, A. Jeffery. Modern System Analysis and Design (Sixth Edition ed). Upper Saddle River, New Jersey 07458: Pearson Education, Inc, 2011.

[3] Pressman, R." Software Engineering A Practitioner's Aproach, New York: McGraw-Hill" 2010.

[4] Dahalin, Z., Abd Razak, R., Ibrahim, H., Yusop, N. and Kasiran, M.K. "An Enterprise Architecture Methodology for Business-IT Alignment: Adopter and Developer Perspectives." Communications of the IBIMA, vol.2010, (2011): 1-14.

[5] Shehab, E. M., M. W. Sharp, and T. A. Spedding. "Enterprise resource planning An integrative review." Business Process Management Journal 10, no. 4 (2004): pp. 359-386.

[6] Okunoye, Adekunle, Frolick Mark, and Crable Elaine. "ERP Implementation in Higher Education: An Account of Pre-Implementation and Implementation Phases." Cases on information technology 8, no. 3 (2006): 110-133.

[7] Nah, Fiona Fui-hoon, Janet Lee-shang Lau, and Jinghua Kuang. "Critical factors for successful implementation of enterprise systems." Business Process Management Journal 7, no. 3 (2001): 285-296.

[8] Ioannoua, G., and C. Papadoyiannisab. "Theory of Constraint-Based Methodology for Effective ERP Implementation." International Journal of Production Research 42, no. 23 (2004): 4927-4954.

[9] Khan, U.A. "Improved Iterative Software Development Method for Game Design. University of Tampere,Department of Computer Sciences." 2008.

[10] Souza, C. and Zwicker, R. "ERP System' Life Cycle: An Extended Version." Technology-Idea group Inc., 2009: 1426-1431.

[11] Sasankar, A. and Chavan, V. "SWOT Analysis of Software Development Process Models." International Journal of Computer Science Issues, vol. 8, no. 5, (2011): 390-399.

[12] Ratbe, Dina, William R King, and Young-gul Kim. "the fit between project characteristics and application development methodologies." The Journal of Computer Information Systems, 2000.

[13] Qureshi, M, and S Hussain. "A reusable software component-based development process model." Advances in Engineering Software 39, no. 2 (2008): 88-94.

[14] Sommerville, I. Software Engineering (8th Edition ed.). Addison Wesely , 2006.

[15] Brehm, L. "Tailoring ERP Systems : A Spectrum of Choices and their Implications." Proceedings of the 34th Hawai'i International Conference on System Sciences. 2001.

[16] Services, O.I. "Selecting a development approach." Office of Information Services, centers for medicare & medicaid, (2008): 1-10.

[17] He, Y. "A Comparative Study of Critical Success Factors for ERP System Implementation in China and Finland", (2007).

[18] Cordeiro, Lucas, and Carlos Mar. "An Agile Development Methodology Applied to Embedded Control Software under Stringent Hardware Constraints." Edited by Lucas Cordeiro, Carlos Mar, E. Valentin and F. Cruz. ACM SIGSOFT Software Engineering Notes 33, no. 1 (2008): 1.

[19] Zeng, M., Wang, C. and Long, Q.-y. "Practices of Extreme Programming for ERP Based On Two-dimensional Dynamic Time Scheduling Interface Method." WRI World Congress on Software Engineering, (2009): 236-240.

[20] Alsultanny, Y. A., & Wohaishi, A. M. "Essential Characteristics of Software Model that Provide the Software Quality Assurance. " International Review on Computers and Software(IRECOS). (2009). 4. n. 5, 577-585.
[21] Negi, T., and Veena Bansal. "A Metric for ERP Complexity." Business Information Systems, 2008.
[22] Ahituv, Niv, Seev Neumann, and Moshe Sviran. "A System Development Methodology for ERP Systems." Computer Information Systems, 2002: 56-67.
[23] Appuswamy, R. "Implementation iss in ERP." International Conference on System Thinking in Managment. 2000. 1-8.
[24] Lind, Mikael, and Jonas Hedman. Is there only one systems development life cycle? Springer-Verlag US, 2009.

BIOGRAPHY

Basem Zughoul Ph.D. in Software and Information Systems Engineering from University Technical Malaysia Melaka in 2014. He received his M.Sc. in Computer Information Systems from University of Banking and Financial Sciences in 2006. He received his B.Sc. in Computer Science from Yarmouk University in 1985. He is currently Assistant Professor in software engineering department in IT faculty in Aqaba University of Technology in Jordan. His main research interests in n software process models, and ERP systems.

DOI 10.17148/IJARCCE.2016.55126