Applicability and issues in traditional model of ERP implementations: an industry perspective

Sunil Kaushik^{1*}, Ashish Bharadwaj¹, Vinay Awasthi¹ and Rajiv Sharma²

University of Petroleum and Energy Studies, Dehradun, India¹ Ministry of Science and Technology, Government of India, New Delhi, India²

Received: 05-January-2017; Revised: 10-March-2017; Accepted: 14-March-2017 ©2017 ACCENTS

Abstract

ERP Implementations follow the traditional model (derived from waterfall model). The traditional model is inflicted with many issues. These issues usually rotate around requirement gathering and defects found in later stage. Most of the literature available is a researcher's experience as an observer or the beliefs. The main objective is to understand the pain of industry practitioners working in ERP Implementations and comparing the same with literature available. The aim of the study is to validate the literature and highlight the issues still not captured. In this paper a study has been presented to understand the actual issues straight from the Industry practitioners' experiences. The literature and findings of the studies were compared to validate the problems in traditional model using empirical research. The interviews and analysis suggest the need of a new framework that includes feedback mechanism, continuous integration with delivery and looking the total value chain instead of small phases.

Keywords

ERP implementation issues, ERP implementation model, Waterfall model issues, Industry view.

1.Introduction

Walter Royce introduced the "waterfall model" in the year 1970 [1]. This model is widely used in the software industry for development [2] and has become the base for the ERP implementation model. The traditional model has the advantages such as the design before code and works well even if the team consists of unseasoned members [3]. ERP implementation researchers also claim that this model would be in use for a while [4]. *Figure 1* defines the ERP Implementation life cycle. Below are the phases of the ERP Implementation Model.

Project Initiation: The project preparation phase focuses on organizing the project. It involves defining and specifying the scope of the project, deciding on implementation strategy and sequence, establishing steering committees, and assigning resources.

Requirements Engineering (RE): Requirements engineering (RE) refers to the process of gathering, analysing, formulating, documenting and maintaining software requirements and to the subfield of software engineering concerned with this process [5].

Realization: In realization phase, ERP system is configured; some functionalities, which are customized or extended using the models and documents produced in RE phase. Realization is a process whereby individual components are assembled and adjusted to construct a working solution.

System Testing: This phase consolidates all the activities of the previous phases. It covers system testing, end-user training, system management, migration of data to the new system, etc. The conversion and interface programs are checked and user acceptance test is carried out.

Go-live: Actual installation of the system takes place and the system comes in use. Immediately after the system goes live, it is reviewed by validating the business processes and technical parameters and interviewing the end-users. The business benefits of the system are measured to estimate return on investment.

Maintenance: Business process keeps on changing. The reasons for these changes could be various such as Govt Rules, Mergers and Acquisitions and new business process after Go live and usage of the system for some time.

^{*}Author for correspondence

Researchers also claim that this model does not fit the environment of ever changing requirements and yields software quality issues [6]. Few researchers also feel the model to be outdated and old fashioned. In order to understand the issue in the waterfall model for ERP implementation following research was done in Inspec & Compendex: ("ERP waterfall model" OR "waterfall model) AND ("empirical" OR "case study" OR "industrial"). Database search resulted in 33 Publications. Most of the research cited was researchers' experience report or belief. None of the study could result the voice of industry practitioners working on large software development projects or ERP implementation projects. The authors conducted a survey to fill this research gap.



Figure 1: ERP implementation traditional model

2.Literature review

Most of the literature talks about the shortcomings of waterfall model for development or the shortcomings of ERP implementation such as rigidity to change [7, 8] or higher effort and cost [8] and lack of opportunity to provide feedback on the system [9]. Traditional models assume that new phases would be started only if the earlier phase is complete. This is tough to attain [10]. Traditional Model that is based on waterfall model has the linear requirement-engineering phase [11]. Few have identified the lack of requirement management [12, 13] and not

addressing customer needs [12]. However, the traditional model is known to be cost saving for offshore projects as fewer trips to customer sites and fewer resources [14]. After understanding literature, the main objective of the research is to identify the issues faced by industry practitioners by using the traditional model for ERP implementations or software development and to find out the correlation, if any, between the literature and issues faced by industry practitioners. Common issues are listed in *Table* 1.

Table 1 List of common issues from literature

ID	Issue
F01	The waterfall model or inspired models increase lead-time due
	to huge volume of documents and software artifacts at end of
	each phase and start of new phase.
F02	Big-bang implementations pushes the test cycle at end that can
	lead to quality problems.
F03	Huge effort or cost needed for approving documents for each
	phase.
F04	Iteration takes considerable effort for rework.
F05	Hard to respond to changes.
F06	Scope is often large and is to be baseline or frozen before start
	of the development.
F07	The system does not reflect the current requirements and
	customer get the system late to provide feedback.
F08	Issues in previous phases are propagated to next phases for a
	solution or are hidden by overworking.

Sunil Kaushik et al.

3.Study design

Data is collected through interview. Carefully, ensuring led us to select the interviews from various parts of the life cycle. Interviews were carried out either in person or telephonically. The interviews lasted for 25 minutes to 42 minutes. The interviewees were of roles solution owners, track owners, functional consultant, project managers, delivery managers, program manager, architects, tech leads, quality assurance managers, QA or test leads.

Interviews were structured in three parts:

- Aim and purpose of study.
- Understanding interviewer's role and experience in the ERP implementations.
- Understanding the challenges, bottlenecks and issues in the current ERP Implementation process.

Data, thus, collected was processed using the following techniques:

- Grouping or clustering: A statement or issues sounding similar or alike or pointing to one issue were clubbed together.
- Derivation of Issues: Raw data or comments with detailed explanations were rephrased into one or two sentences to draw a conclusion and were clustered.

To avoid any bias, interviews were re conducted after a gap of 2 months and most of the interviewees were consistent with their comments.

The data thus clustered were prioritized in 5 categories

- C1-The problem / issue is mentioned by more than 50% of the respondents.
- C2-The problem / issue is mentioned by more than 35% of the respondents.
- C3-The problem / issue is mentioned by more than 25% of the respondents.
- C4-The problem / issue is mentioned by more than 15% of the respondents.
- E-The problem / issue is mentioned by less than 15% of the respondents. So it is not considered.

4.Findings

C1 Issues

As indicated on an ordinal scale that C1 issues are faced by more than 50% of respondents. Most of the respondents agreed that most of the requirements gathered during the requirement gathering or engineering phase are either changed or discarded. The main reason for this is cited as longer lead times, forcing developers to perform CRs. The change in requirements is usually produced by the progressive elaboration, that with further process in phases or when the user sees the working product or system.

The required documents created in requirement engineering phase usually undergo a scrutiny of scope and most of the time requirements dropped earlier owing to scope are made "must" after seeing the working product or system. It has been mentioned in the literature [5, 7, 8]. Literature also talks about the high effort, cost of development and verification [5, 7]. Few of the researchers agree that the feedback of the customer plays and important role and inclusion customer [15, 16, 6]. However, the literature does not, but it does not talk about limits, boundaries and the degree to which customer is included.

Industry practitioners respond that initial requirements gathering usually consumed more effort and time than initially allocated. In order to minimize the schedule deviation, other cycles usually shortened or compromised. The compromise, in the end, yields a faulty product or system that does not match the customer requirement or needs which is in line with literature [1, 15, 16]. However, literature research does not talk about curtailing the requirement gathering to schedule and testing process to ensure that requirements are met.

C2 Issues

The testing cycle is usually done after the complete system is configured and customized have been deployed. Industry practitioners have found that most of the issues found in the testing are not only critical to system functioning also but are also tough to fix. The issues thus found to usually affect non-functional requirements (NFRs) such as performance of the system. Most often the issues found are related to design architecture. These issues are complex and cost intensive. Literature findings agree on high effort and cost [15, 16]. The literature does not talk about taming of NFRs owing to late testing and solution to ensure testing is to check the NFRs also.

Ever-changing customers' requirements of new requirements by customer is found usually near go live / release of the system. Industry practitioners dread such surprise. The main reason behind this fear is the effect of new requirements, which, usually, are not limited to a single system or module, but it tends to touch more than one system. Such dependencies are often looked during the last minute changes. Industry practitioners admit the pressure of curtailing schedule deviation. In order to minimize the schedule deviation, they often start the other phase such as design and development or blueprinting before completion of requirement phase. This often yields to confusion on version on which blueprint has been designed. This leads to effort consumption in correction and adds up the magnitude of faults. This important point was found to be missing to capture the interest of previous researchers.

Technical members such as architects and technical leads mentioned that ERP implementation projects have a high number of requirements and customizations. Customers or users, using the influence and control, always to try to include a number of requirements during the last phase of requirement gathering or even after that. Estimates usually done based on the initial estimates usually overload technical folks. They find it tough to include these new requirements into existing configurations and customizations. They clearly mentioned the need of layered development with a time given to restructure the design of configurations and customizations without changing functionality. This clearly identifies the need of a single person in control to shield the team with new requirements, an empowered core team that is able to decide the action on its own. It also highlights the need of release based development and need of design or code refactoring. This point could not be found in available literature.

C3 Issues

It is mostly seen that a developed / customization or configured module does well and works as intended. However, when the same module is integrated to create a system, the newly created system does not yield the expected result. It is often found that it takes effort and time to identify the issue in smaller Fixing them, as discussed earlier, components. requires effort and time. Such problem could be reduced to minimum if newly configured or customized components are integrated with the system soon after finishing. Industry practitioners acknowledge that most of times various teams work on different part of system and integration of such components soon after development looks to be tough. It is suggested that in case of absence of newly customized dependent components, a placeholder may be provided which would provide the default response or behaviour and integration could be verified. Gaps were seen on these critical points in earlier available literature.

Industry practitioners also mention that ERP implementations are often beyond boundaries of implementation of ERP Modules. Most of the time, ERP is to be integrated with third party systemswhich could be home grown or bought from other vendors. Fault detection and identifications becomes problematic and impossible. Implementation teams are usually given documentations to understand the system and more often problem identification becomes a herculean task. Reason cited behind is communication or rather formal communication, but no collaboration between teams, customers, product owners. The collaboration in various teams is found to be an essential point missing in available literature for ERP Implementations. The collaboration between various teams and customers should be for end-toend cycle. The collaboration leads to long term relationship and problem solving thus conducive environment for the ERP Implementation satisfying every one's needs and requirements. The long-term relationships and collaboration will do away with multiple quality checks that don't add any value, but as gates to ensure all required documentations. Most of the time, various teams' tries to pass on these gates creating loads of documents. This time spent could have been used for adding value adds into the code or addressing non-functional requirements.

C4 Issues

In case of regular ERP Implementations, team members are found to be narrowly focused and specialist in their area. Consultants admit their specialization and lack of knowledge of other modules or tracks. The reason behind this is inability to look at the whole value chain of the system. This inability often leads to non-value adding activities, thus creating non-optimized value chain. The authors could not find any reference of this in the available literature.

5.Discussion and analysis

The authors identified that most critical issues (Type C1 and C2) in ERP Implementations those cited by industry practitioners are related to requirement engineering and system testing. This is totally in line with the literature. Industry practitioners have cited that most of the requirements are either reworked to take care of customer's change, or earlier requirements are dropped to work on new requirements.

The type C3 issues are pertaining to the continued delivery. The authors have identified that industry practitioners feel that longer lead times owing to

Sunil Kaushik et al.

delay in requirement gathering or changes often lead to scope and in turn lead to schedule and effort changes, It is also seen that despite longer lead time, requirements are not stabilized. The un-stabilized requirements and longer lead time requirements often yield to reduction in test coverage and thus affecting the quality.

The C4 issues are not critical and mostly related to knowledge and inability to view the complete value chain by the consultants. This inability usually leads to system integration failures.

6.Limitations of the study

The study was done using a small data set. The issues highlighted could be verified conclusively if the sample set was big. The study does not talk about the quantitative impact of variables (such as delay in requirements of volatility) on the others such as schedule deviation and effort deviation.

7.Conclusion and future work

Above study tries to find out the issues in the traditional model for ERP Implementations. The study compares literature studied and industry view point. Survey underlines many issues which are different from those of literature. Most of the issues cited are either because of issues in requirement gathering or lack of customer collaboration. In light of the issues, it is seen that traditional model is not suitable for ERP Implementations. To fix the issues of Traditional ERP implementation model, a new model for ERP implementation has to be conceived and proposed.

Appendix

The interviewers were categorized in four sets based on the role performed in the projects.

- S1-Solution owners, track owners, functional consultant.
- S2-Project managers, delivery managers, program manager.
- S3-Architects, tech leads.
- S4-Quality assurance managers, QA or test leads.

The distribution of the count in the sets is as follows (*Table 2*).

Table 2	Distribution of	f respondents	(role wise)	
S1	S2	S 3	S4	
6	8	7	11	

Conflicts of interest

The authors have no conflicts of interest to declare.

Acknowledgment

None.

References

- Royce WW. Managing the development of large software systems. In proceedings of IEEE WESCON 1970 (pp. 1-9).
- [2] Thomas M. It projects sink or swim. British computer society review. 2001.
- [3] Singh D, Thakur A, Chaudhary A. A comparative study between waterfall and incremental software development life cycle model. International Journal of Emerging Trends in Science and Technology. 2015; 2(4):2204-8.
- [4] Raccoon LB. Fifty years of progress in software engineering. ACM SIGSOFT Software Engineering Notes. 1997; 22(1):88-104.
- [5] Sommerville I. Software Engineering. Pearson Education Ltd; 2011.
- [6] Cohen D, Larson G, Ware B. Improving software investments through requirements validation. In annual NASA Goddard software engineering workshop 2001 (pp. 106-14).
- [7] McBreen P. Software craftsmanship: the new imperative. Addison-Wesley Professional; 2002.
- [8] Pfleeger SL, Atlee JM. Software engineering: theory and practice. Pearson Education India; 1998.
- [9] Jones C. Patterns of software system failure and success. ITP New Media; 1996.
- [10] Tipaldi M, Götz C, Ferraguto M, Troiano L, Bruenjes B. The robust software feedback model: an effective waterfall model tailoring for space SW. In DASIAdata systems in aerospace conference 2013.
- [11] Madgunda S, Suman U, Praneeth GS, Kasera R. Steps in requirement stage of waterfall model. International Journal of Computer & Mathematical Sciences. 2015:86-7.
- [12] Yin RK. Case study research: design and methods. Sage Publications; 2013.
- [13] Johnson J. Keynote speech: build only the features you need. In proceedings of the international conference on extreme programming and agile processes in software engineering 2002 (pp. 73-82).
- [14] Chandra V. Comparison between various software development methodologies. International Journal of Computer Applications. 2015; 131(9):7-10.
- [15] Anderson DJ. Agile management for software engineering: Applying the theory of constraints for business results. Prentice Hall Professional; 2003.
- [16] Boehm B. Get ready for agile methods, with care. Computer. 2002; 35(1):64-9.

International Journal of Advanced Computer Research, Vol 7(30)



Sunil Kaushik is a PhD research scholar from UPES; MBA B.E (Hons) is an ERP implementation program /technology manager having worked with MNCs for over 15 Years in India and abroad. He is a certified in various module of Oracle Apps ERP.

Email: sunil.kaushik@rediffmail.com



Dr. Ashish Bharadwaj has received PhD in Computer Science from University of Petroleum & Energy Studies. He has published many Papers at both National and International levels. He has served as CIO of many Indian MNCs and Member of the SAP Executive Advisory Council on Higher

Education and Research and the SAP Global Solutions Advisory Group.



Dr. Vinay Avasthi has received PhD in Computer Science from University of Petroleum & Energy Studies. He has published many Papers at both National and International levels. He has rich experience in Academic & Research includes Data mining, Networking algorithm, distributed computing, cloud

computing and IOT. He is the member of ACM, CSI and ISTE. He is also the recipient of CSI Significant Contribution Award 2012-13.



Dr. Rajiv Sharma has received PhD from IIT Kanpur. He is currently serving as Scientist 'G', Ministry of Science and Technology where he Heads Technology Mission handling divisions and managing several mission critical projects of Nano Science & Technology, Solar Energy, Water and

Supercomputing. He has served as Executive Director of IUSSTF, USA. He has rich experience in Industry and Academic & Research.