1. Introduction

It is found that more than 90% ERP projects have spent more than allocated budgets and have crossed the time schedule for ERP implementation. Approximately 20% of ERP implementation projects are unsuccessful. The success rate of ERP implementation is below 30%. In 2015 ERP Report as depicted in Figure 1, Panorama Consulting hints that more than 66% of the companies failed to realize the 50% benefit of the ERP, 54% implementations have run over budget and 72% have exceeded the planned and scheduled duration. Traditional models are inspired by waterfall model and are connected to high effort and thus cost.

Despite the higher failure rate of ERP Implementation project, very less studies have been performed in the area of ERP Implementation improvements.

2. Review of Literature

In the literature survey, it is found out that main issue in the current ERP Implementation model is rigidity to change or higher effort and cost and lack of opportunity...
to provide feedback on the system. Authors studied the traditional model of ERP Implementation and surveyed the industry professionals and found out the the validity of issues mentioned earlier in addition to the factors that affect schedule, effort and quality. Looking at this, some researchers have tried to apply Agile Methodologies to ERP Implementations. But these researches assumed the ERP deployment as ERP implementation. Authors point out that customizations are integral part of ERP and ranges from code to creation of a new module. The various examples of customizations are as given in Table 1. In order to understand if Agile Methodologies can be applied to ERP Implementations, Authors studied attributes Agile and various Agile methodologies. On realizing that IT industry is analogous to Manufacturing, Authors came across the Lean Thinking, often called as Lean. A philosophy of maximizing value and minimizing waste, with the purpose of “doing more with less”. Sutton pointed out that Lean Thinking could be used for quality improvement and cost reduction. Brehm et. al. has given the various types of Tailoring tyoes in ERP Implementations. The various tailoring types are provided in Table 1.

### Table 1. Typology of ERP Tailoring Types

| Tailoring Type      | Description                                                                 | Examples                                                                 | Layers Involved                  |
|---------------------|-----------------------------------------------------------------------------|--------------------------------------------------------------------------|----------------------------------|
| Configuration       | Setting of parameters (or tables), in order to choose between different executions of processes and functions in the software package | Define organizational units; create standard reports; formulate available-to promise logic; use of a standard interface to an archive system | All layers                       |
| Bolt-ons            | Implementation of third party package designed to work with ERP system and provide industry-specific functionality | Provide ability to track inventory by product dimensions (e.g. 2500 m. lengths of cable do not equal 11000 m. lengths) | All Layers                       |
| Screen masks        | Creating of new screen masks for input and output (soft copy) of data       | Integrate three screens into one                                         | Communication layers             |
| Extended reporting  | Programming of extended data output and reporting Options                   | Design new report with sales revenues for specific criteria              | Application layer and / or database layer |
| Workflow Programming| Creating non-standard Workflows                                             | Set up automated engineering change order approval process              | Application layer and / or database layer |
| User exits          | Programming of additional software code in an open interface                 | Develop a statistical function for calculating particular metrics        | Application layer and / or database layer |
| ERP Programming     | Programming of additional applications without changing the source code (using the computer language of the vendor) | Create a program that calculates the phases of the moon for use in production scheduling | All Layers                       |
| Interface development| Programming of interface to legacy system or 3rd party products             | Interface with custom-build shop-floor system or with a CRM package      | Application layer and / or database layer |
| Package code modification | Changing the source codes ranging from small changes to change whole modules | Change error message in warning; modify production planning               | Can involve all layers           |

### 2.1 Understanding Agile Methodologies and Lean Thinking

**Agile Methodologies** are based on following 12 Principles

- **AM1** - Satisfy Customer.
- **AM2** - Welcome changing requirements, even in late development.
- **AM3** - Deliver working software frequently.
- **AM4** - Cooperation between business people and developers throughout the project.
- **AM5** - Trust motivated individuals and build projects around them.
- **AM6** - Enable Face-to-face conversation.
- **AM7** - Measure progress by delivered working software.
- **AM8** - Maintain a constant pace of delivery.
- **AM9** - Technical excellence and good design to high quality product for scalability and maintenance.
- **AM10** - Simplicity.
- **AM11** - Self-organizing teams.
- **AM12** - Continuous reflection.

On the other hand, the Lean Thinking has following 7 principles:
• LT01: Eliminate waste: Anything not adding any value to customer or product should be removed.
• LT02: Amplify Learning: Learning can be termed as getting better understanding of customer needs, solutions and testing strategies. Processes and practice should support and provide environment for learning.
• LT03: Defer commitment: Delay the commitment of irreversible decision till last moment and option for change should remain open till last moment.
• LT04: Deliver as fast as possible: Minimize delivery time turnaround time from requirement gathering to final delivery of software.
• LT05: Respect people: Create a team of technically excellent people, provide them reasonable and realistic goals and trust them on getting the job done.
• LT06: Build quality in: Ensure that defects are not injected in first place and they should be eliminated as soon as they are discovered.
• LT07: Optimize the whole: Optimize the whole value stream i.e. from receipt of a requirement to software deployment and avoid sub-optimization

Authors did a comparative study of Lean Thinking and Agile Methodologies and found out that none of these alone could be solution to the problems of ERP Implementations. Authors did an analysis, as depicted in Figure 2 using a Market winner and qualifier matrix

Authors did a comparative study of Lean Thinking and Agile Methodologies and found out that none of these alone could be solution to the problems of ERP Implementations. Authors did an analysis, as depicted in Figure 2 using a Market winner and qualifier matrix

Figure 2. Analysis Matrix for Lean and Agile

Lean and Agile are two different paradigms, Lean emphasizes on waste reduction and Agile aims at flexibility and adaptability to changing environment. Despite the differences, in principal, Agile Lean support each other. Lean aligns with many Agile principles but considers a more holistic enterprise perspective. Software industry has already been using the Agile Methodologies and, recently, have shown an interest to apply Lean thinking to software development. Agile conceptually relates to as a leaness and it is found that Lean and Agile combination have found to be very much efficient.

Authors felt the need of a new framework using Lean Thinking and Agile Methodologies for efficient ERP Implementations. Authors did a survey of the practitioners to validate the need of a new framework. The survey was participated by industry experts ranging from Developers to Directors and with experience of 7 to 23 years. The results of interaction with experts provide highly significant evidence that experts are looking for a solution (using Lean Thinking and Agile Methodologies) to stop ERP Implementation and Customization failure. Merging Lean and Agile also affects the team’s productivity and over all development process.

3. GenNext Framework for ERP Implementations

Based on the feedback of survey and alarming magnitude of ERP Implementation, Authors have created a new framework - GenNext Framework. The framework is created after keeping the important aspects of practical problems and complementing Agile Methodologies and Lean Thinking for ERP Implementations. Authors have created the framework primarily for ERP Implementations but it can be used other Enterprise implementations such as BI Reporting or any other COTS implementations.

3.1 Values / Pillars of GenNext Framework

GenNext Framework is created on the following
• P1 - Continously improve and optimize the flow and value of the system.
• P2 - Manage change and respond to pulls.
• P3 - Relationship by co-development and delivery of working software.
• P4 - Product success over functional success.
• P5 -Trustworthy empowered and self managing team.

3.2 Principles of GenNext Framework

GN -1 : Customer Satisfaction
• GN-2 : Accept changing requirements from users and provide them option to change requirement even late in cycle.
GN-3: Deliver frequently and possibly at a constant pace.
GN-4: Eliminate non-value adding activities and wastes.
GN-5: Cooperation between users and developers throughout the implementation cycle.
GN-6: Self-organizing team of motivated individuals and respect for every one.
GN-7: Seamless communication.
GN-8: Measure progress by delivery.
GN-9: Technical excellence and quality.
GN-10: Keep every thing simple.
GN-11: Amplified learning should be continuously reflected in the work and delivery.
GN-12: Optimize the whole value chain.
GN-13: Create a Pull Environment.

3.3 Scope of the GenNext ERP Implementation Project
The scope of an ERP Implementation Project is defined by high level requirements, initially, and are blown up further to get the detailed requirements. The requirements thus collected are called backlog. The requirement are divided in to processes and features. Thus, at the end of release of features and processes are delivered to ensure the incremental value addition in stead of piling of components. This ensures a frequent feedback and a glimpse of customer’s vision which helps in maximising the possibility that product/ERP system meets customer requirements. Table 2 represents the terms used in GenNext framework scoping.

| TERMS   | DESCRIPTION                                                                                                                                 |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Backlog | The list of all product, features with appropriate prioritization and complexity or effort rating. This list includes the epics and user stories contained therein where already defined. |
| Theme   | Top level objectives of the ERP system. E.g. Implementing ERP system in APAC region, EMEA Region.                                           |
| Epic    | Group of related processes. Epic is introduced as processes and features in a release.                                                     |
| Process | A specific functionally independent requirement. It includes an acceptance criteria or condition of satisfaction to check the correct functioning of this requirement. A Process is an Independent, Negotiable, Valuable, Estimable, Small, Testable requirement |
| Feature | A group of requirements, from a business perspective, whereby each feature is on a very high level of abstraction and is described concisely in a short paragraph. The description is sufficient for experts to assess the complexity and thus also the scope of work. |

3.4 Phases of GenNext Framework
GenNext Framework divides the whole implementation process in 3 major phases which are as follows -
• Propose
• Construct and Configure
• Delivery and Maintenance

The sequence of these 3 phases is shown in Figure 3. In an ERP implementation, these three phases keep on happening sequentially or in parallel. Usually, requirements are identified, developed and delivered. But, if any new process or requirement identified during the construction phase will add to propose phase and re-prioritization will happen. The terms related to GenNext framework are defined in Table 2.

Figure 3. Phases in GenNext Framework.

3.4.1 Propose
Proposing or Planning is an essential activity usually done in start of any project. Planning shows the intention that how a project would be handled and steered to ensure the deliveries. Planning sets the priority and pitch of the project. This stage can be defined as collating the ideas from conception to materialization. Planning activity can be divided in following sub phases -
• Prepare - Objective of the sub phase is to identify the scope of the project and listing all the processes
including the As-Is/To Be Analysis. All the requirements, at high level, are stored in a backlog.

- **Estimate** - Objective of the subphase is to get the approximate effort for carrying out the activities. Different organizations and researchers have devised many number of estimation techniques. Estimation technique, as agreed with customer, should be used and should provide the results in Person Hours.

- **Prioritize** - Objective of the subphase is to rank the process or requirements in an order based on the Business value, effort, alternatives and reusability. Authors have proposed to divide the process in to sub process, features and get the Process Priority Number or Feature Priority Number by applying the framework devised by the Authors.

### 3.4.2 Construction and Configuration

This phase can be summarized as continuous construction or configuration of the defined requirements. This phase encompasses analyze, design, development, integration and testing and execute these as a continuous activity, not as a sequential process as done in traditional models.

Purpose of this stage is not only deploying the release but to ascertain that release is deployable. This stage is all about preparing for release, including final documentation, pre-release staged testing and releasing the product to end users.

The implement phase consists of three sub phases viz.

- **Release Planning and Development** - The main purpose of this sub phase is to plan the release, design the architecture or choose a way by minimally disturbing the architecture and developing the application. The phase also serve as input to the requirements in case a new requirement is discovered during the progressive elaboration phase. This subphase helps us not only includes development but also a feedback mechanism.

- **Monitoring and Control** - The Monitoring and control subphase is to capture and control the vital statistics such as cost, quality and effort. In the GenNext Framework monitoring is done from the customer/end user perspective and effective measure are taken to control deviations. This sub phase includes the usage of visual controlboards and daily clarification meeting with time boxing of the development or configuration. At the end of each iteration a review and demo is conducted. The main purpose of the review is to retrospect and identify the points which could have been better and reflect the knowledge thus gained in next iterations. The acceptance criteria of the

- **Quality Assurance** - Quality Control and assurance is the part of any project to check the conformance to the requirements. In GenNext framework, this role is not limited to testing and ensuring the processes that defects are unearthed or curtailed. In GenNext framework QA is adaptive and ensures that processes are tested as soon as they are developed and configured. In GenNext framework, process test cases are written first with the category of simple, medium and complex. The processes are built to ensure that they initially meet the simple test cases then medium and complex. It ensures that no gold plating or extra/ non value adding work is done.

### 3.4.3 Delivery and Maintenance

This phase encompasses the activities related to incremental and final delivery. The deployable releases are integrated incrementally with a view of final delivery and product success instead of functional successes of each function. This can be divided in two sub phases viz.

- **Deployment and Stabilization** - Each release is continuously delivered and integrated with the final product. A new release may cause few issues in the product and needs to be closely monitored and product needs to be supported at the highest priority.

- **Documentation and Training** - Each release has few activities such as documentation and training is required. Documentation and training is required for both the end users and team members who are going to support the application.

### 4. Results and Discussions

GenNext framework was applied in a training program where participants were trainees and professionals with 4 years of experience. Results are given in Table 3. The framework was seen to be achieving its objective. GenNext framework was able to save effort in a range of 15-20% from the traditional models. In addition, most of the effort was saved in design and testing activities. This was as expected because the GenNext framework emphasis on test driven development and elimination of waste i.e. gold plating of features.
5. Conclusions, Limitation and Future Work

The current paper presents a framework with an intention to reduce the time elapsed thus maintaining the schedule, cost and defects or cost of quality. The current implementation saw very small effort and schedule deviation with high customer satisfaction whereas traditional model was seen capturing a significant effort and schedule deviation and thus low customer satisfaction. Results of this study are under review. However, GenNext framework needs to be applied to a real time ERP Implementation and effect needs to be studied.

6. References

1. Appleton E. How to survive ERP Datamation. 1997; 43(3):50-3.
2. Davenport TH. Putting the enterprise into the enterprise system. Harvard Business Review. 1998; 76(4):121-33. PMid:10181586.
3. Martin MH. Smart Managing Fortune Magazine. 1998.
4. Ho C, Wu W, Tai Y. Strategies for the adaptation of ERP systems. Industrial Management and Data Systems. 2004; 104(3):234-51. Crossref
5. ERP Report 2015 Panorama Consulting. 2015. Available from: Crossref
6. McBreen P. Addison-Wesley Boston: Software craftsmanship the new imperative. 2002. PMid:12005556.
7. Pfleeger S, Atlee J. Prentice Hall: Software engineering theory and practice. 2006. PMid:16916045.
8. Jones C. International Thomson Computer Press: Patterns of Software Systems Failure and Success. 1995; 28(3):86-7. Crossref
9. Tipaldi M, Gotz C, Ferraguto M, Troiano L, Bruenjes B. The robust software feedback models an effective waterfall model tailoring for space sw. Proceedings of DASIA. 2013; 720.
10. Botta-Genoulaz V, Millet PA, Grabot B. A survey on the recent research literature on ERP systems. Computers in Industry. 2005; 56(6):510-22. Crossref
11. Esteves J, Bohorquez V. An updated ERP systems annotated bibliography 2001-2005. Communications of AIS. 2007; 19:1-63. Crossref
12. Julia H. Bringing Lean Principles to Service Industries. HBS Working Knowledge. 2007.
13. Agarwal A, Shankar R, Tiwari MK. Modeling the metrics of lean agile and agile supply chain An ANP-based approach. European Journal of Operational Research. 2006; 173(1):211-25. Crossref
14. Sutton JM. Lean Software for the Lean Aircraft in 15th Digital Avionics Systems Conference. 1996; p. 49-54.
15. Brehm L, Heinzl A, Markus M. Tailoring ERP systems A spectrum of choices and their implications. Proceeding of 34th Hawaii IEEE International conference on System sciences HICSS. 2000; p. 9.
16. Kaushik S, Avasthi V, Bharadwaj A. Scrutinizing Lean Thinking and Agile Methodologies from Practitioners Point of View. Indian Journal of Science and Technology. 2016; 9(20):1-5. Crossref
17. Mason-Jones R, Naylor J, Towill D. Engineering the Leagile Supply Chain. International Journal of Agile Management Systems. 2000; 2(1):54-61. Crossref
18. Krishnamurthy R, Yauch C. Leagile manufacturing A proposed corporate infrastructure. International Journal of Operations and Production Management. 2007; 27(6):588-604. Crossref
19. Naylor J, Naim M, Berry D. Leagility integrating the lean and agile manufacturing paradigms in the total supply chain. International Journal of production economics. 1999; 62(1-2):107-18. Crossref
20. Laanti M. Agile methods in large-scale software development organisations. University of Oulu: Doctoral thesis: Applicability and model for adoption. 2012; p. 1-192.
21. Vilkkki K. When agile is not enough. Springer: Lean Enterprise Software and Systems. 2010; p. 44-7. Crossref
22. Aravindraj V. Evaluation of leagility in supply chains using fuzzy logic approach. Taylor Francis Online: International Journal of Production Research. 2013; 51(4):1186-95.
23. Dingsoyr T, Nerur S, Moe N. A decade of agile methodologies towards explaining agile software development. Jour-

### Table 3. Comparison of efforts in Traditional Model and GenNext framework

| Activity                              | Traditional Model (Person Days) | Gen Next (Person Days) | % Saving using GenNext Framework |
|---------------------------------------|--------------------------------|------------------------|----------------------------------|
| Requirement Specification including rework | 62                             | 51                     | 17.74                            |
| Design including rework               | 68                             | 54                     | 20.59                            |
| Coding including rework               | 149                            | 127                    | 14.77                            |
| Integration Testing and Bug Fix       | 56                             | 48                     | 14.29                            |
| UAT and BugFix                        | 45                             | 34                     | 24.44                            |
| **Total**                             | **380**                        | **314**                | **17.37**                        |
nal of Systems and Software. 2012; 85(6):1213-21. Crossref

24. Kaushik S, Bharadwaj A, Avasthi V. Need for blending Agile Methodologies and Lean Thinking for ERP Implementation An industry point of view. 2015 1st International Conference on Next Generation Computing Technologies (NGCT). 2015; p. 751-5.

25. Fagerholm F, Ikonen M, Kettunen P, Munch J, Roto V, Abrahamsson P. Performance Alignment Work How software developers experience the continuous adaptation of team performance. Information Software Technology. 2015; 64:132-47. Crossref

26. Kaushik S, Bharadwaj A, Avasthi V, Sharma R. A Novel Framework for Requirement Prioritization for ERP Implementation. Indian Journal of Science and Technology. 2016; 9(48):1-3.