ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/104021, December 2016* ISSN (Online) : 0974-5645

A Novel Framework for Requirement Prioritization for ERP Implementation

Sunil Kaushik[1*], Ashish Bharadwaj[1], VinayAwasthi[1] and Rajiv Sharma[2]

[1]University of Petroleum and Energy Studies, Dehradun - 248007, Uttarakhand, India; sunil.kaushik@rediffmail.com, ashish@upes.ac.in, vinay.avasthi@upes.ac.in
[2]Ministry of Science and Technology- Govt of India, New Delhi, India; rajiv.sharma@nic.in

Abstract

Objectives: This paper presents a framework or approach for prioritizing the requirements for an ERP Implementation. **Method:** Proposed a novel framework or algorithm which calculates and ranks for the requirements based on priority rules such as ROI, available alternatives. **Findings:** The algorithm considers requirements as inputs and outputs the priority of processes in ERP Implementation. Improvement: The work can be extended to find the new variables and coefficient of performance or inclusion of constraints and paired requirements.

Keywords: ERP Requirements Priortization

1. Introduction

Software development is more than coding and implementation of COTS in the current scenario. It not only consists of philosophies, principles, frameworks and documentations to ensure the product conforms to desired requirements and quality standards but also includes completing the project on schedule, on budget. Software development process is increasing reliance on stakeholders'**.** Requirements prioritization is one of the principles used in software development. There are several requirement analysis methods or patterns those acts as framework to ensure better communication and flow of domain knowledge[1]. Requirements prioritization is a complex decision making process with multi criteria and multivariable[2]**.** It is necessary to put requirements in priority to meet the cost, schedule and quality[3,4]. Requirements perceived critical and important should be given priority over the other requirements. Software developed using the pre prioritized requirements will prove to be better in design, architecture, coding, release planning and schedule, and also meets the preference of stakeholders, thus increasing the chance of acceptance[5-7]. Requirements can be categorized on the basis of client's will and command if the requirements are frozen in the initial phases of the project. However, due to progressive elaboration and requirements discoveries most of the requirements falling in "must" are only picked up and requirements falling in "should" or "could" are usually avoided.

With authors' personal experience on Traditional and new methodologies such as Lean and Agile, authors have identified that stakeholders don't have any mechanism by which they can categorize or prioritize the requirements. The problem becomes grave when it comes down to ERP Implementation which involves configuration, customization and great deal of change management with customer acceptance. The current article talks about the algorithm to prioritize the requirements for ERP Implementations. The algorithm proposed can be used normal software projects.

2. Algorithm

ERP/COTS or any other application consists of processes, functionalities, features. The functionalities and features

* *Author for correspondence*

are part of a process usually. The purpose of the algorithm is to categorize the requirements into a prioritize sequence based on business value, effort and alternative available. Following terms and considerations should be understood before understanding the algorithm.

- **Business value:** Is usually perceived as user value or ROI (Return Of Investment). In the start of the project the requirements are known at higher level and the business value is always a perceived value and may not be the exact dollar value of the ROI of the process.
- **Effort:** To configure or code the process in the application plays a vital criterion. Processes of high business values and low effort are delightful and should be given higher priority over the others.
- **Reusability:** The features and functionalities or even the process cannot be denied in application. Usually, applications have the processes which are reusable in the other processes too. The Reusability is calculated as R = n+1, where n is the number of times feature is getting reused in the application. For example, if F1 is reusable in 2 process than the R will be 2+1 = 3.
- **Alternatives:** Sometimes processes or functionalities can be achieved through other means without development of current functionality. For example, an order booking button and Order Booking Batch does the same work. Though, Order booking button does the operations real time and Batch does it on frequency of 5 mins but the process does not stop and Order Booking Button has alternative. The Alternative value is calculated as A= n+1, where n is the number of alternatives available.

The prioritization algorithm is based on the user experience and the philosophy that Process with features with high business values and low customization/ configuration should be given the priority over others.

FPN α Business Value/Effort (1)

Based on the authors' practical experience in IT industry, authors feel that if a feature can be reused should be given high priority over others and, if a feature or functionality has alternatives available, and can be achieved without developing the functionality, should be given low priority over others which don't have alternatives available. Thus,

FPN α Reusability/Alternatives (2)

With Equation 1 and Equation 2, it can be easily deduced as –

FPN = k (Business Value * Reusability)/(Effort * Alternatives) (3)

Where k is coefficient of proportion. For the simplicity k is assumed to be 1.

Thus –

FPN = k (Business Value * Reusability)/(Effort * Alternatives) (Eq 4)

The proposed prioritization rule is "To prioritize the processes with the highest ratio of importance to actual effort will be prioritized first and skipping user stories that are "too big" for current release".

The various steps in the calculations of Priority are –

- List all the Processes to be implemented. Suppose P is the process i.e. there $P_{1,}$,$P_{2,}$,P_3 ……..$P_{n,}$
- List all the Sub Processes in a Process. Suppose Process P1 has n number of sub process S P_1, SP_2, SP_3….. SP_n.
- List all the Features embedded in Functionalities and Independent features in a Sub Process. Suppose Sub Process SP1 has embedded features EF_1, EF_2, EF_3….. EF_n and independent features IF_1, IF_2, IF_3….. IF_n.
- Identify the Business Value in agreed term of each feature (both embedded and independent). Let V_i be the business value of a feature F_i.
- Identify the Effort in agreed terms to build each feature (both embedded and independent). Let E_i be the effort of a feature F_i.
- Identify the Reusability of each feature (both embedded and independent). Let R_i be the Reusability of a feature F_i in other features. For example if F1 is reusable in 2 processes than the R will be 2+1 = 3.
- Identify the if the feature has an alternative feature. Let A_i be the alternatives available for a feature F_i.
- Calculate FPN with formula - FPN = (R*V)/ (E*A).
-
- Repeat the above step for all Embedded and Independent Functions
-
- Calculate the Sub Process Priority Number as SPN = $\sum^{n}_{i=1}$ EFPN + $\sum^{k}_{j=1}$ IFPN
- Calculate the PPN as PPN = $\sum^{n}_{i=1}$ SPN.
- Order the data as $PPN_i > PPN_{i+1.}$

Two processes P1 and P2 are ranked using the

Table 1. Comparison of two processes using the proposed algorithm

PID	SPID	FID	BV	RU	E (PD)	A	FPN	SPN	PPN
P1									1041.389
	SP1							697.2222	
		IF1	1000	0	12	0	83.33333		
		IF2	1100	0	12	0	91.66667		
		EF1	1200	1	6	0	400		
		EF2	1100	0	9	0	122.2222		
	SP2							344.1667	
		IF1	1400	1	12	1	116.6667		
		IF2	1100	0	8	0	137.5		
		EF1	900	0	10	0	90		
		EF2	250	1	4	0	125		
P2									884.2314
	SP1							516.3743	
		IF1	1000	1	19	0	105.2632		
		IF2	1000	1	18	0	111.1111		
		EF1	1800	0	6	0	300		
		EF2	1900	0	1	1	950		
	SP2							367.8571	
		IF1	1300	1	14	1	92.85714		
		IF2	1400	1	16	1	87.5		
		EF1	1500	0	8	0	187.5		
		EF2	1300	0	8	1	81.25		

algorithm in the Table 1. Process P1 has higher PPN then Process P2 and hence P1 is ranked higher than P2.

3. Conclusion and Future Work

The purpose of the research is to present the solution to the problem based by most of the customers implementing ERP. Above algorithm has yielded good prioritized requirements in the testing environment. The above algorithm uses few of the variables such as Business Value (calculated in dollar terms), reusability and alternatives (which can be calculated after proper analysis) which are person dependent. Further research can be done to find the new variables and coefficient of performance or inclusion of constraints and paired requirements.

4. References

1. Kumar K, Saravanaguru R. Requirement pattern extraction using cluster based framework - RAPID. Indian Journal of Science and Technology. 2016 Feb; 9(5):1-5.
2. Leftngwell D, Widrig D. Managing Software Requirements: A United Approach. Addison-Wesley Longman Inc.; 2000.
3. Ruhe G, Eberlein A, Pfahl D. Trade-off analysis for requirements selection. International Journal of Software. Engineering. Knowledge Engineering. 2003; 13 (4):345–66.
4. Finkelstein, Harman M, Mansouri S, Ren J, Zhang Y. A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. Requirements Engineering. 2009; 14:231–45.
5. Barney S, Aurum A, Wohlin C. A product management challenge: Creating software product value through requirements selection. The Journal of *Systems Architecture.* 2008; 54:576–93.
6. Ahl V. An experimental comparison of tive prioritization methods – investigating ease of use, accuracy and scalability [Master's thesis]. Sweden: School of Engineering, Blekinge Institute of Technology; 2005.
7. Berander P, Khan K, Lehtola L. Towards a research framework on requirements prioritization. Proceedings of 6th Conference on Software Engineering Research and Practice in Sweden (SERPS'06); 2006.