

# Requirements Prioritization: Survey and Analysis

Sita Devulapalli, Akhil Khare and O.R.S. Rao

**Abstract** Survey on requirements prioritization to identify the practices related to requirements prioritization among the software development organizations and to understand association of requirements prioritization' effects on software deliveries and resources is designed and conducted for projects/products in different domains across organizations. The results are analyzed for identifying areas that needed attention in terms of requirements prioritization. The survey and analysis enable understanding the need for requirements prioritization for stable and smooth release cycles. A multi-level framework utilizing the concepts of ABC analysis is suggested as a method for prioritization, for predictable and stable releases.

**Keywords** Requirements prioritization · Multi-level framework · Stable release cycles

## 1 Introduction

Software solutions being built for various medium to small size organizations to enable them to leverage software solutions for their businesses are often taken up by startup or small companies. While large companies offer generic solutions as products surviving through years and provide customization for specific business needs, there is a good mix of new customized solution offerings developed a new by companies as well as customized solutions on generalized solutions meeting the needs of IT enablement of business. Similar to off-the-shelf products' initial versions, the development of software starts as a solution development and continues to

---

Sita Devulapalli (✉) · Akhil Khare · O.R.S. Rao  
ICFAI University Jharkhand, Ranchi, Jharkhand, India  
e-mail: sitadpalli@yahoo.co.in

Akhil Khare  
e-mail: khareakhil@gmail.com

O.R.S. Rao  
e-mail: orsrao.icfai@gmail.com

undergo enhancements and fixes, thereby evolving into business-specific products. They are certainly not one time buys and live through versions of modifications till scaling of business demands a new solution or simpler and new technology-based solutions are needed. And the cycle of new product solutions begins. Through these cycles, requirements are gathered, analyzed, refined, and prioritized as per client's business needs, technology changes, and resource needs.

In this cycle of product solutions development, often there is less clarity on requirements in the initial stages and requirements change frequently in nature and scope. Changing business needs during the development phase also results in changes in requirements. Chasing the changes in requirements often results in increased development efforts, over worked teams, and extended release dates. In order to understand the requirements handling process during the software development, a survey is designed to gather current methods, difficulties faced, and solutions adopted.

The survey is structured around parameters like products domain, maturity of the products, development process variations, and requirements handling modes. The survey is designed based on the author's industry experience in software products development. The objective of the survey is to identify practices related to requirements' prioritization among software development organizations and to understand association of requirements prioritization' effects on software deliveries and resources. The survey is conducted to understand the effectiveness of the current processes and to identify requirement's prioritization needs for enabling planned deliverables with reduced uncertainties.

The respondents participated in this survey range from organizations that are long term, enterprise products players to relatively new and single product/custom software players. The domains are related to engineering fields to commerce applications to gaming solutions. Some of the products have been under continuous enhancement and maintenance for years.

Different processes—waterfall, iterative, agile—are followed across the organizations. The products developed are typically used by large customer base of the clients for specific applications on different platforms and devices. Products undergo modifications to meet further requirements of the clients, often changing requirements as the development progresses. Providing the customers with ever enhancing products is made possible by successive releases of products at varied intervals, ranging from few weeks to few months to 1 or 2 years.

The fundamental questions that need to be addressed are—What is to be made available in the next release? How to manage the requirements under expanding client needs, cost and time implications? Will prioritization of requirements and planning releases help to streamline the project deliveries to client's satisfaction without overworking the teams or missing time to market deadlines?

This paper describes the survey conducted to bring out information about the domains and applications, process of development, how requirements are handled currently—in Sect. 2. The survey questions are prepared based on the author's experience with product development. The nature of responses and analysis of significant responses is presented in Sects. 3 and 4. Improvements that are feasible and a framework that can help simplify the process of requirements prioritization are discussed further in Sect. 5.

## 2 Survey, Respondents, and Organizations

The organizations of respondents varied from large (>200 employees) to medium (25–200 employees) sized to small (<25 employees), with local and global presence of the products. Some of the organizations have multiple product lines, while some have single product lines. The survey responses are gathered from 53 respondents belonging to 20 organizations. The respondents are involved in business analysis, project management and product development.

The survey questionnaire is divided into three parts. The first part elicited data related to the domains of the project, nature of the project, the role played by the respondent, the stage the product is in, and release cycle durations with 12 questions.

The part II focused on the process followed for development and gathered information on what process is followed for development, how the requirements are collected and analyzed, problem areas like over work or over-runs on time with 10 questions. Part III focused on how the requirements are handled across the projects and has 20 questions, covering collection of requirements, prioritization methods used, areas of problems, and current solutions adopted. Responses to part II and part III are presented in the following section.

## 3 Nature of Responses on Processes

The survey has 3 parts and 42 questions, overall, and notable points are discussed here. The domains of applications developed varied from engineering to consumer applications, across manufacturing, telecom, and finance to e-commerce. The product's life cycle stage varied from less than 2 years to greater than 10 years. The applications are typically enterprise applications, web applications, and mobile applications working across devices and platforms, used in multiple countries and are mostly three-tier applications. The processes followed are waterfall, iterative, agile, and agile being the predominant process. Classification of respondents' data is presented in Table 1.

**Table 1** Development process, complexity

| Size of org | Respondents following     |                 |       | Respondents working with products complexity |        |             |
|-------------|---------------------------|-----------------|-------|--|--------|-------------|
|             | Waterfall/iterative/agile | Iterative/agile | Agile | 3 tier/n tier                                | 2 tier | Single tier |
| Large       | 7                         | 2               | 8     | 15   | 1      | 1           |
| Medium      | 2                         | 3               | 21    | 16   | 8      | 2           |
| Small       |                           | 3               | 7     | 7  | 1      | 2           |
| Total       | 9                         | 8               | 36    | 38   | 10     | 5           |

No specific method is used for requirement’s prioritization. The focus has been mostly on the customer demand for specific features. Relevant parameters considered for requirements prioritization are—business value (BV), availability of resources (AR), difficulty of implementation (DI), and impact on existing customers (IC) without weight to the parameters, mostly. The responses collated with regard to current requirements selection criteria, problems faced, and solutions adopted are presented in Tables 2, 3 and 4.

**Table 2** Current processes

| Features/requirements to be implemented for next release |    | Problem areas with current process of feature selection for upcoming release |    | Circumventing the problems with current process of feature selection |    |
|--|----|--|----|--|----|
| Based on customer needs                                  | 36 | Estimation-time resources  | 14 | Client management/meetings   | 10 |
| Time to market   | 5  | Lack of prioritization wrt complexity, time                                  | 12 | Discussions with stakeholders  | 10 |
| No preference  | 6  | Requirement clarity/changes  | 12 | Do nothing   | 5  |
| Enhancements   | 1  | Dependencies—other modules, new tech   | 6  | Extra time and hard work   | 6  |
| Impact analysis  | 5  |  |    | Estimate/extend/analyze  | 9  |
|  |    | No response  | 9  | No response  | 13 |

**Table 3** Additional time needs and replanning

| Response   | Teams working for release under pressure and for long hours in a day | Some of the team members over worked during releases | Abandoning features being implemented for a release and restart on new features |
|------------|--|--|---|
| Often      | 19   | 18   | 2   |
| Rarely     | 7  | 3  | 24  |
| Sometimes  | 21   | 23   | 26  |
| Very often | 5  | 8  |   |

**Table 4** Resource availability, impacts, rework of resources’ bandwidth

| Response   | Right resources availability is an issue for meeting release schedules | Abandoning features during release due to realized impacts on existing customers | Analyzing the impacts on core structure/architecture/data model, of features to be implemented a priori | Reworking of resource (time, personnel, S/W, H/W) estimates for the features during the development cycle for a release |
|------------|--|--|---|---|
| Often      | 20   | 5  | 21  | 23  |
| Rarely     | 11   | 26   | 1   | 6   |
| Sometimes  | 16   | 17   | 11  | 22  |
| Very often | 5  | 2  | 19  | 1   |

A clear and systematic approach was not apparent from the responses for the question—How does the respondent choose features/requirements to be implemented for next release. It is largely based on customer needs alone. Changes in the requirements often resulted in extending the dates for releases.

Analysis of the responses to the question—“What are the problem areas you see with your current process of feature selection for upcoming release?” narrows down the problem areas with the current process followed to analysis, estimation, and planning.

The question—How do you circumvent the problems with your current process of feature selection?—indicated to typical solutions being followed like over work, extended releases, and attempts to convince clients. Often the teams worked under pressure and for long hours in order to meet requirements for release.

Response to the question—How often do you have teams working for release under pressure and for long hours in a day?—is given in Table 3.

Lack of clear-cut requirements analysis prioritization resulted in teams working for additional time often and also in replanning the releases by abandoning some features and adding new features into the release.

Analyzing and taking into account resources availability impacts on existing customers are two areas that seem to be only partially considered for defining requirements for upcoming releases. Impacts of new requirements on existing product structures is another area that seem to be not taken into account by everyone. Table 4 gives classification of data on these three aspects and resulting rework of resource bandwidth for releases.

## 4 Responses to the Survey on Requirements Prioritization

Requirement collection across the organizations appears to be through all channels available—marketing, existing clients, executive direction, and development team. Development teams and planning teams are providing the assessment/analysis mostly. Simple classification of requirements into three groups of—must have, good to have and need not have—appears to be the familiar method followed for requirements prioritization for product releases. Tables 5, 6, and 7 present the data for requirements collection, and classification.

**Table 5** Requirements classification

| Weights are associated with parameters considered for prioritization |    | A multi stage prioritization scheme is useful for requirements prioritization |    | Working out prioritization exactly for each requirement for product releases |    |
|--|----|---|----|--|----|
| Most often   | 5  | Most often  | 27 | Most useful  | 21 |
| No weights   | 10 | Not used/never  | 2  | Not useful   | 1  |
| Not often  | 11 | Not often   | 13 | Not useful often   | 9  |
| Often  | 23 | Always  | 8  | Often useful   | 20 |

**Table 6** Requirements prioritization

| Requirement prioritization/response   | Most often | Always | Not always |
|---|------------|--------|------------|
| It is essential to know how much important each requirement is when compared to other for prioritization              | 25         | 21     | 4          |
| It is sufficient to know relative importance of requirements for prioritization rather than “how much more important” | 28         | 13     | 9          |
| Cost—value ratio for requirements is the best indicator of priority   | 21         | 4      | 25         |
| Classifying into—1. must have 2. GOOD to have 3. can live without—groups  | 29         | 13     | 8          |

**Table 7** Requirements prioritization

| Response   | Ranking of requirements (in sequence of priority) based on a parameter is sufficient for prioritization | Numerical assignment of priority (grouping by assigning priority 1, 2,3, ...) to requirements is sufficient | Requirements prioritization provides traceability along the product life cycle for improved quality of the product |
|------------|---|---|--|
| Most often | 27  | 26  | 22   |
| Always     | 7   | 6   | 16   |
| Not always | 16  | 19  | 13   |

Association of weight to prioritization parameters for requirements is used in some organizations. Similarly, exact prioritization for requirements finds favor to a good extent. A multi-stage prioritization method is expected to be useful to a large extent as is evident from data in Table 5.

Relative importance and quantification of relative importance appear to be important for prioritization. Perception about cost–value ratio for prioritization has a mixed response. Table 6 indicates the respondents’ preferences.

Ranking of requirements based on a preferred parameter, numerical assignment of priority for prioritization of requirements do find a favor by many, though not by all participants. Prioritization of requirements added improved quality as seen in the Table 7.

## 5 Analysis and a Framework for Prioritization of Requirements

Responses to the survey indicate a need for focus on requirements prioritization for planning releases systematically, with controlled changes during the course of release cycle. The methods being used appear to be relative ranking and grouping

into—must have, good to have, and need not have. Utilization of weighted parameters for requirements prioritization, adopting multi-level prioritization finds a place in practice, though not by all. Lack of appropriate requirement prioritization methods, process often appears to have resulted in teams working under pressure, extended release dates, and dropped features.

The survey covered large companies with mature products releasing successive versions of products with longer release cycle, as well as midsize to small companies working on specific project based product versions with less maturity and shorter duration release cycles. Across this range of organizations, requirements analysis and prioritization for products/projects first versions, as well as successive versions is an area that needed attention and systematic methods to be adopted for stable, successful, and smooth deliverables in a predictable manner. Taking the nature of products/projects and the process prior to development as constraints, requirements prioritization for the purpose of predictable releases of products is analyzed. The following baseline is suggested for the requirements prioritization.

The purpose of getting a set of requirements implemented for the next release (time bound) is to maximize the business value of the release for the most valued customers. A strict ordering of requirements may not be the need. Need is more for a near-optimal sets of requirements. Since a release is always timed to meet customers expected needs, the following additional constraints are considered for prioritization of requirements

1. Time/duration—minimum time required for development.
2. Nature of development needed for the requirements.
3. Resources—knowledgeable in domain/technology/skill.
4. Uncertainties—changes due to expanded/extended scope.
5. Impacts on existing customers and existing product modules.

Based on the above considerations, the following framework [1] is suggested for simple and effective prioritization at multiple levels enabling implicit weight application for relevant parameters for the requirements, which enables flexible planning through the development cycle. The framework provides visualization for the changes in requirements during the release cycle and acts as a easy communicator to the involved stakeholders including testing team members.

The framework is defined as five sets based on most used parameters in the sequence of priority determination. Each set is defined by three classes/bins defined by % value of the respective set parameters. Requirements are grouped into the classes in the sets in the process of prioritization. The percentage bands may vary from industry to industry and organization to organization to some extent.

Prioritization sets—S1–S5 and classes/bins—A, B, C within are described in Table 8.

The framework is applied in a layered approach through the sets. The order of preference emerges for the requirements set through the filtering process. Not all

**Table 8** Framework—sets, classes

| Sets   | Classes/bins—A, B, C         |
|--|------------------------------|
| S1. Business value (BV) in conjunction with customer base (CB)   | A: 20 % of CB with 70 % BV   |
|  | B: 30 % of CB with 25 % BV   |
|  | C: 50 % of CB with 5 % BV    |
| S2. Requirements applicability with respect to product, where UW: user interface, BI: business logic, CP: core | A: 70 % UW, 30 % BI, 0 % CP  |
|  | B: 50 % UW, 40 % BI, 10 % CP |
|  | C: 30 % UW, 50 % BI, 20 % CP |
| S3. Implementation cost, where MI: marginal implementation, NI: new implementation, IR: impact recovery        | A: 70 % MI, 25 % NI, 5 % IR  |
|  | B: 50 % MI, 40 % NI, 10 % IR |
|  | C: 30 % MI, 50 % NI, 20 % IR |
| S4. Time requirement, where L: 8–16 person weeks, M: 4–8 person weeks, S: 2–4 person weeks                     | A: 10 % L, 20 % M, 70 % S    |
|  | B: 15 % L, 25 %M, 60 % S     |
|  | C: 20 % L, 30 % M, 50 % S    |
| S5. Resource requirement, where RC: core aware, RI: industry aware, RT: technology aware                       | A: 10 % RC, 20 % RI, 70 % RT |
|  | B: 15 % RC, 25 % RI, 60 % RT |
|  | C:20 % RC, 30 % RI, 50 % RT  |

sets may be required to be used. When all sets are used for classification, we will arrive at 243 bins of requirements. Based on the constraints and release theme, the bins can be selected in the order of preference for the releases.

## 6 Conclusion

The survey conducted across organizations developing software products—first version to multiple versions, brings out the lack of systematic methods usage for requirements prioritization. It also brought out the associated problem areas and difficulties in achieving successful software product deliveries. Prioritization of requirements based on parameters relevant to the product development a priori and during the development cycle facilitates stable and predictable deliveries with less resource allocation uncertainties. The framework suggested enables simple and effective methodology for requirements prioritization for successive releases under dynamic changes and leads to better understanding and planning of releases. It helps build traceability and visualize effects of plan changes and helps in informed quality planning.

## Reference

1. Sita Devulapalli, Akhil Khare: “A Framework for Requirement Prioritization for Software Products” IUJ Journal of Management, Vol 2, No. 1, May 2014.



## Bibliography

2. Frank Moisiadis.: THE FUNDAMENTALS OF PRIORITISING REQUIREMENTS. Systems Engineering, Test & Evaluation Conference, Sydney, Australia, October 2002.
3. Karlsson J, Ryan K.: A cost-value approach for prioritizing requirements. IEEE Software 1997;14(5):67–74.
4. Wiegers KE.: Software Requirements, Microsoft Press (1999): Redmont, DC.
5. Karlsson J, Wohlin C, Regnell B: An evaluation of methods for prioritizing software requirements, Inform. Software Technol. 1998, 39(14–15): 939-947.
6. Alan M. Davis, Ed Yourdon, Ann S. Zweig: Requirements Management Made Easy. 39–947. [www.omni-vista.com](http://www.omni-vista.com).
7. Regnell, B., H'ost, M., Natt och Dag, J., Beremark, P., Hjelm, T.: An Industrial Case Study on Distributed Prioritization in Market-Driven Requirements Engineering for Packaged Software. Requirements Engineering 2001, vol 6, no 1, pp 51–62.
8. Laura Lehtola, and Marjo Kauppinen: Suitability of Requirements Prioritization Methods for Market-driven Software Product Development. Software Process Improvement Practice 2006; 11: 7–19.