Conference Paper

Identify Compliance During Software Development Using System Engineering Principles

A.O. Tretyakov and M.V. Druzhinin

Ural Federal University named after the first Russian President B.N. Yeltsin Russia, 620002, Ekaterinburg, street Mira, 19

Abstract

Errors made during the requirements collection and analysis phase make it very difficult to maintain the software product and cost the company extra costs. The difficulty of directly collecting requirements from stakeholders is due to inconsistencies between the major stakeholder groups, as well as related factors in the collection of requirements itself, as well as the selected methodology for the process of converting stakeholder requirements into development requirements. As a solution, it is necessary to use a high level of prioritization in order to distinguish among many requirements the necessary for successful implementation of the product, as well as to correctly allocate compliance with the requirements in such a way that each group of stakeholders is satisfied, but at the same time setting the goals of the supersystem more priority than the goals of the subsystem. This article discusses the methodology of system engineering to solve issues related to the identification of possible contradictions of requirements.

Keywords: System engineering, requirements engineering, business process, requirements, software product, analysis.

Corresponding Author:
A.O. Tretyakov
Tretiakov.Arkadiy@gmail.com

Received: 5 March 2020
Accepted: 18 March 2020
Published: 8 April 2020

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the SEC 2019 Conference Committee.

1. Introduction

At the moment, a correct, timely and well-organized approach to the collection of requirements is a key factor in the further success of the project. The methodology of requirements collection may differ depending on the level of the project, it can be quite informal in small projects, where in a small group of employees the transfer of requirements and needs of the client comes directly to each employee and employees can act largely at the level of their vision, and in medium and large projects, where the approach to the collection and transfer of requirements is strictly formalized, where interdisciplinary interaction and the use of methods from the field of requirements engineering is required.

How to cite this article: A.O. Tretyakov and M.V. Druzhinin, (2020), “Identify Compliance During Software Development Using System Engineering Principles” in *III Annual International Conference ”System Engineering”*, KnE Engineering, pages 100–104. DOI 10.18502/keg.v5i3.6765

The complexity lies largely in the fact that it is problematic to determine the correctness and correctness of the collected requirements at the level of the business process, and the errors associated with this can be detected only at the level of drawing up the requirements for the system, and only at the stage of direct design of the system. The correction of such errors costs a substantial investment and sometimes for this reason the correction is not recognized as valid, and the error remains in the project. Carl Vigers [1] in his book points out that the errors made at the stage of collecting requirements are 40-60% of all project errors, 30-50% of the errors of the finished part are spent on processing, 70-85% of them are errors in the collection of requirements.

This article aims to show problem areas that arise in the transition of requirements from the level of the business process (business layer) to the level of development (application-layer) that need to be considered and which can become critical in the future, as part of a major project.

2. Main Part

Let’s consider the problem areas that may appear during the fixation and analysis of requirements on the example of a real WFM project from the Russian company ABC Solution.

For a simpler description of the requirements collection process, we will distinguish three different levels of the client company, corresponding to the levels of hierarchy within the company. Let’s call them conditionally: high level, middle level and low level. At each level there are stakeholders corresponding to this level. For further consideration, we will take specific functional roles from some stakeholders (Figure 1).

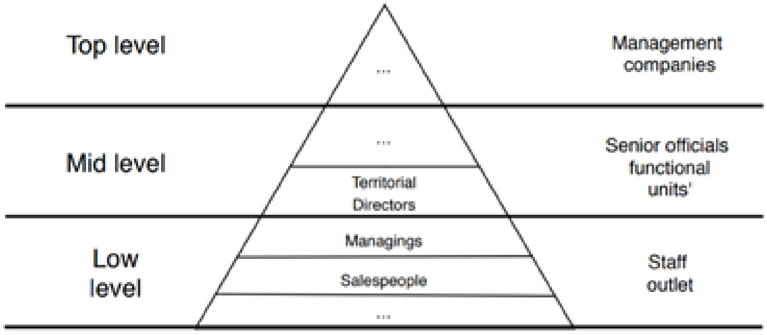

Figure 1: A generalized scheme of levels of requirements collection and target stakeholders in the context of the WFM project.

As a rule, it is with these levels that the main difficulties of collecting requirements arise.

DOI 10.18502/keg.v5i3.6765

First, the requirements collection process itself is not static, but is distributed over time by stages that correlate with the stages of the product life cycle, while the collection of requirements from each level begins at different stages (the dependence of the requirements collection processes from different levels on the stages of the product life cycle is shown in Figure 2).

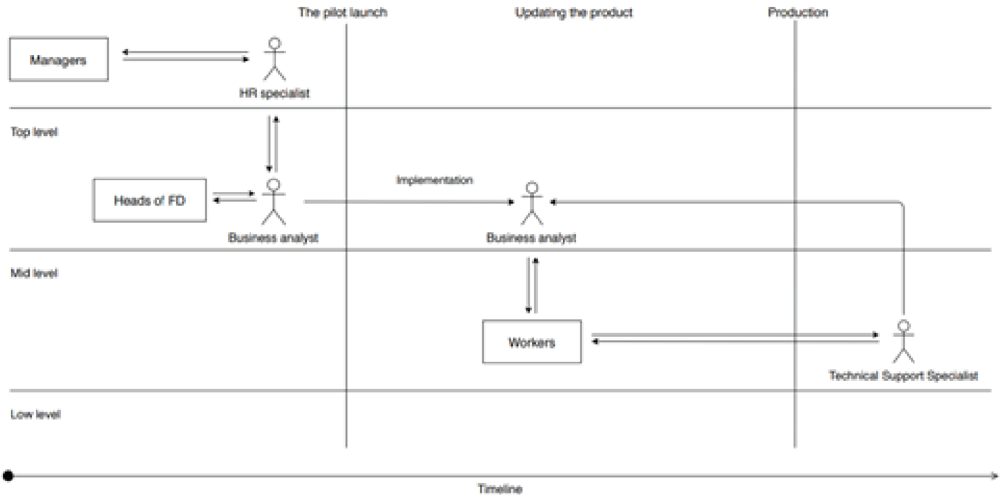

Figure 2: A simplified timing diagram of requirements gathering

In fact, one of the stakeholders of the system comes already in a ready kind. The framework of the project is created at the development stage, and the project itself is a " box " solution and varies slightly from client to client, making changes to it may incur certain costs, so the implementation of some requirements from a new stakeholder may be recognized as inappropriate and omitted. It is also worth considering that the requirements for the system may vary depending on a variety of environmental factors, such as the market situation, the change of the company's goals, the beginning of the existence of the system, etc., and changing the system in accordance with each changed requirement, also requires costs.

Secondly, stakeholders at different levels have very different approaches to the formation of requirements and the vision of their importance. This is due to large information gaps between levels. These gaps can arise for various reasons, whether it is territorial (one division Manager can have many controlled points in different cities with their managers) or simply a highly formalized "boss-subordinate" relationship existing in the company. As a consequence, stakeholders from higher levels can form requirements based on the interests and goals of the company, and at a low level may not know at all about the goals of the company regarding the product, being at their level, they do not see the benefits of the project for the whole company, but only see those moments as the product affects them specifically, and this is mainly the creation of additional load. Based on this, they form requirements aimed at maximum simplification of their work.

As a result, the business analyst receives sets of requirements from three (the value may vary depending on the project, but the principle remains the same) actually non-dependent stakeholder groups formed with different vision and at different stages. At the same time it is impossible to refuse part of requirements as insufficient observance of requirements from stakeholders of high levels as the customer can endanger relevance of the project and threatens with refusal of the customer of a product, but not performance of requirements from stekholders of low level as the main users threatens with loss of efficiency of use of a product, thereby at all innovativeness and manufacturability of the project, makes it uncompetitive.

Further actions aimed at moving from stakeholder requirements to SOFTWARE requirements are that the business analyst collecting requirements from the client after the interview passes them to the project team (or specifically to the project Manager (product owner) if we talk about the agile approach). Then they are postponed by the project team at the level of understanding and on their basis the requirements for the SOFTWARE are formed directly. This process is able to proceed effectively only in small projects where there are not many requirements and the risk of not taking into account something or losing is minimal, but in larger projects there is a risk of losing requirements or approval of mutually contradictory requirements.

Let's consider the solution of this problem from the point of view of system engineering. Aksenov A. A. [2] in his proposed concept of SOFTWARE requirements development says that the main goal of a business analyst after receiving all the information about the future system is to divide the requirements into parts and carefully structure them. After that, you need to create a use case diagram (UML) or several diagrams that will give us complete information about all possible uses of the system. Next, you need to make a prototype that will give an interactive view of all the functions used, as well as give an idea of the correctness and completeness of the requirements.

Further, according to the principle of" global goal "and the principle of" reasonable sufficiency " (Adaptive Satisfying) D. Hitchins [3], we define the main super-system and all its requirements (criteria for requirements, for example, can be found in ISO 29148 [4]), then using various methods prioritization (weights) prioritize the implementation of the requirements in such a way that would try to lead to success all groups of stakeholders, but putting the goals of the main super-system is always above the goals of any sub-system.

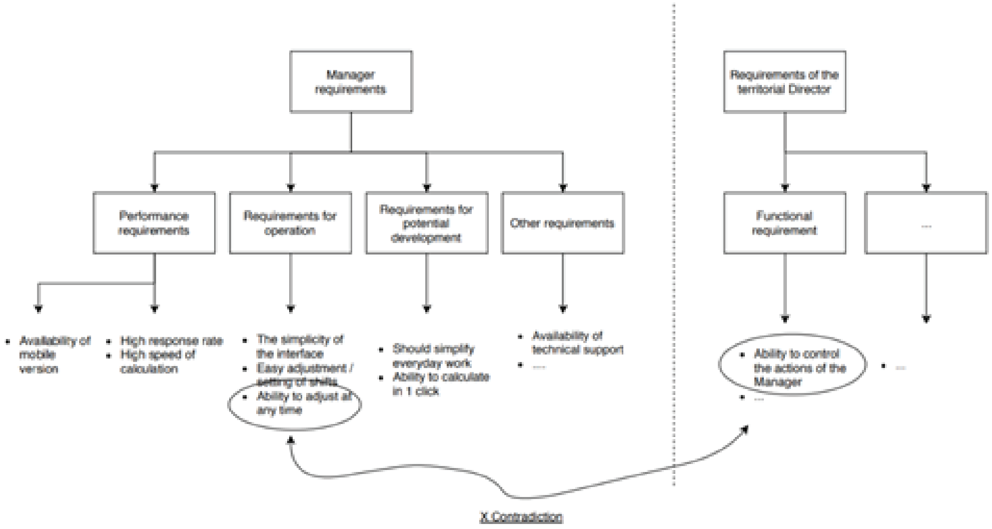
Manager requirements
Performance requirements
Requirements for operation
Requirements for potential development
Other requirements
Availability of mobile version
High response rate
High speed of calculation
The simplicity of the interface
Easy adjustment / ~~setting of shifts~~
Ability to adjust at any time
Should simplify everyday work
Ability to calculate in 1 click
Availability of technical support
...
Requirements of the territorial Director
Functional requirement
...
Ability to control the actions of the Manager
...
...
X Contradiction

Figure 3: Example of finding conflicting requirements using the requirements tree

3. Summary

The process of collecting requirements is fundamental to any modern system, it is the basis that will largely determine the future success of the entire project. This article discusses the problem areas in the formation of requirements that can cause inconsistency of requirements, and as a consequence the costs of their elimination, as well as shows the methodology of their solution by methods of system engineering.

In the future, it is planned to consider in detail the requirements for tools and changes in development processes.

References

[1] Karl I. Vigers, software requirements Development, Moscow: Russian edition, 2004, ISBN 5-7502-0240-2.

[2] A. A. Sergeevich, ""Information support of process of development of requirements to a hi-tech product", " Economics, pp. 59-62, 2017.

[3] Hitchins D. What are the General Principles Applicable to Systems? — INCOSE INSIGHT. — V. 12, Issue 4. — December 2009. — pp. 59–64

[4] ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes — Requirements engineering