KnE Engineering
SEC 2019
III Annual International Conference "System Engineering"
Volume 2020

Conference Paper

Design Quality Assurance System «Product Studio»

D.A. Zakharov[1], V.V. Mizgulin[2], and V.S. Malitsky[1]

[1]LLC «ABC Solutions», Russia, Moscow
[2]INCOSE

Abstract

The article describes the systems engineering processes, the current problems encountered in the work of a system engineer, the reasons for their occurrence and the goals, functionality of the new system tool "Product Studio". «Product Studio» is Design Product Quality Assurance System and it aims to simplify the activities of the system engineer, providing the necessary functionality to perform the relevant processes from requirements definition and management, to the formation of architectural and technical solutions.

Keywords: system engineering, system modeling tools, system life cycle process, software system, system engineering activities, architectural design, system modeling languages, current problems in the SE work

Corresponding Author:
D.A. Zakharov
d.zakharov@goodt.me

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

The unlimited growth of needs each year leads to the emergence of complex technical and software systems that require the involvement of a large number of versatile professionals, as well as the use of different methodologies, practices and technologies.

Activities on realization of complex systems always represent collective work of experts on the full life cycle of a system. Consequently, there are difficulties both in dividing the areas of responsibility between specialists and in general in uniting the organizational activities of companies to implement complex systems. Undoubtedly, complexity leads to an increase in the number of risks of unsuccessful systems. Success of systems in this case means not only that the implemented system works, but first of all, that it satisfies the needs (allows to achieve the desired result) of all stakeholders (interested persons), subject to the planned terms and budget.

The majority of modern companies cannot find a single format of work for themselves - a set of related practices that allows to systematize the result in the form of successful systems. Each new project has its own unique development line. There are no formalized, established principles and ways to achieve the goals of the project.

The question arises as to whether there is a methodology in place to ensure the systematic establishment of successful systems with existing resource constraints and at a stage of complete uncertainty.

System engineering comes to the rescue. System engineering is a universal methodological basis necessary for successful implementation of coordinated efforts of different specialists in formation and implementation of well-structured activity on creation of systems throughout the whole life cycle [1].

2. System Engineer Work

To a greater extent, the obvious need for the work of a system engineer is noticeable at the initial stage of system development (at the stage of conceptual modeling). This fact is confirmed by the following arguments.

First of all, we must not forget that an important factor influencing the system definition processes is the competitive environment within which the company has to be located. Often companies have to participate in various tenders held by the customer as part of the competition. Based on the results of the tender, the winner (enterprise) is determined, who receives the legal right to develop, implement and deliver the system. During the tender, which is mostly at the initial stages of system development, representatives of participating companies strive to generate system concepts that meet the needs of all stakeholders and surpass the competitors' offers. It is important to maintain a balance between the performance indicators, the cost of the system, the timing of its implementation, as well as the resources necessary to maintain its functioning throughout the entire LC. During the tender it is important for the customer to demonstrate a cost-effective project program with a realistic forecast of production/delivery times and the funds needed to finance the development of the system.

Secondly, as a result of its activities, a systems engineer should receive a working product, on the basis of which the company wants not only to receive investment funds to launch the project, but also to begin a more specialized development of the system. Such a working product is either a project proposal for the development of the system, or, as is customary in Russia, the Terms of Reference.

Accordingly, in a short period of time, the team (led by a system engineer) should form a complete, coherent and consistent ToR. Requirements from the ToR will be used to perform the system design.

If the terms of reference contain an incomplete set of conflicting, unstructured requirements, this may lead to serious problems at further stages of work with the system.

This hypothesis has been substantiated by the results of the above studies. Several companies have conducted studies to assess the cost of errors that occur at different stages of the life cycle of systems [2].

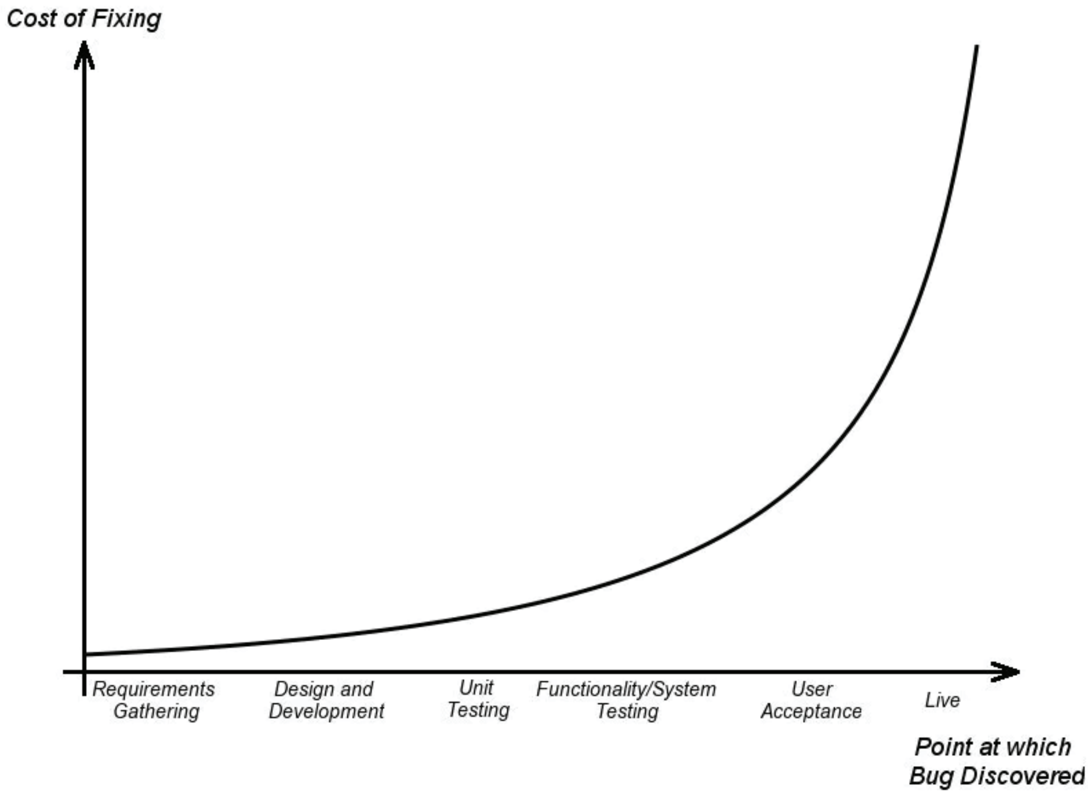

Figure 1: graph of the cost of error correction on the stage of LC, where they were found

Thus, it can be concluded that the system engineer should focus his or her efforts and those of the entire team - primarily on the conceptual modeling stage, at the time of the system concept formation.

2.1. Existing problems in the subject area

We have already noted that system engineering includes a set of different practices for defining and implementing the system. But the introduction of systems engineering practices in an enterprise alone cannot guarantee that you will not have any problems. Even with the use of practices, the following problems often arise:

- The product created does not fully satisfy stakeholders (or, in principle, does not solve the pain of stakeholders);
- Contribution to the investment product is not justified (the investor does not receive a refund);

- There is no understanding of what resources are needed to implement the product (what kind of specialists and competences are needed, what kind of capacities and equipment are needed);
- It takes a great deal of time to develop a system concept that, even after prolonged suffering, does not give a complete and holistic understanding of the product. And a lot of time is spent harmonizing the decisions regarding the vision of the product;
- A huge number of system errors occur at the stage of conceptual modeling and are detected (at best) only at later stages;
- Important project data is often lost when communicating between different specialists within a team;
- It takes a long time to monitor changes in requirements and assess their possible impact on architectural solutions.

Accordingly, for the full application of the practices, an accompanying technology is required. In this case, the technology is meant as a toolkit that can simplify the system-engineering activity (in particular, the formation of the product concept).

The existing solutions simply cannot provide a solution to the above problems due to the many inconveniences that will be discussed in section 5.

That's why the idea of creating a special tool of system modeling of the product appeared. This tool is our target system and it is its conceptual description that will be presented in this article. But before you begin to describe the target system directly, you need to understand the operation context in more detail and understand where our system will be used.

2.2. Description of the subject area

All of these problems exist within a specific using system. Problems are the starting point for assessing the current state of the environment and act as an incentive for stakeholders to move to the desired state (the "as is" state).

Modeling systems makes it necessary to abstract from the existing reality and pay attention to the most important details of the context. In the same way, we will act when modeling a system that uses it. In this article we will consider not a particular enterprise from real life but a certain class of enterprises responsible for product development in which system engineering is implemented.

System engineering is a set of practices and currently there is an ISO/IEC 15288:2015 standard that describes the main blocks of processes (practices) of the system life cycle [3]. In sum, all the blocks form a map of interrelated activities of both system engineers and enterprise management, defining a unified approach to describing the life cycle.

Let's move on to a more detailed definition of the operating context, using the processes of the standard. The set of processes of interest to us is presented in Figure 2.

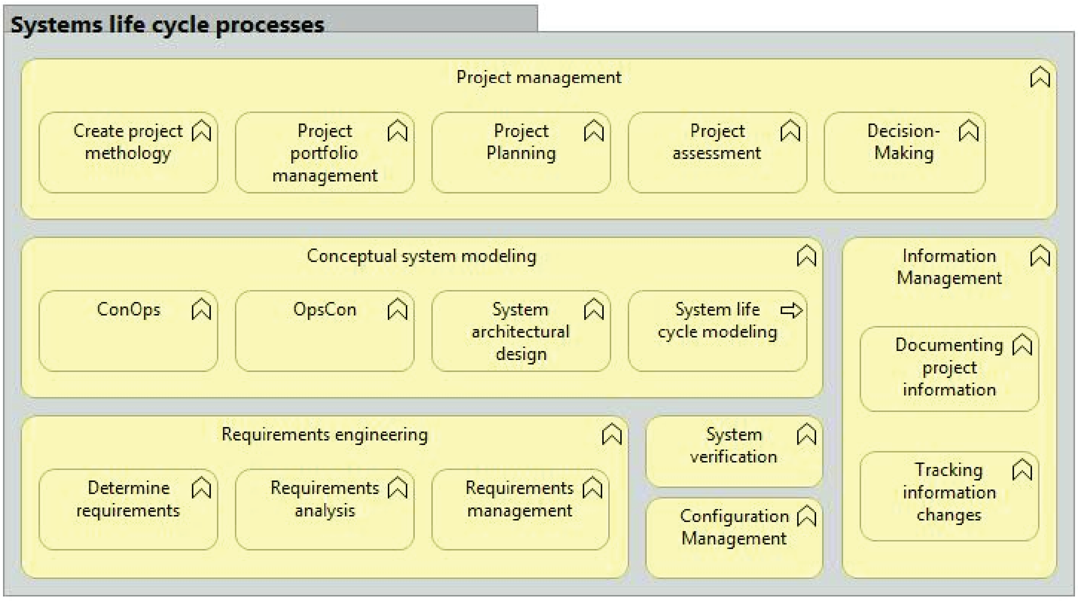

Figure 2: System life cycle processes

It is very important to note that due to the development of information technology, it is difficult to imagine a company without the presence of IT systems that automate the business processes of the enterprise.

At present the basic tool which the system engineer uses is the graphic designer (archi, capella, modelio). Without any doubt, these software packages cause the most inconvenience in the work. That's why we will focus on the program's data when analyzing existing solutions and try to assess their disadvantages, which significantly slow down the system engineer's work.

2.3. Description of the subject area

Existing system modeling languages (ArchiMate, SysML, Capella, UML) and corresponding modellers (Archi, Modelio, Cradle, Capella Arcadia) do not have features like:

- synchronous simulation of target, using and supporting systems;

- the ability to perform analytical queries to the created models (checking the correctness, completeness and integrity of the model);
- collaborative work mode with the ability to track data changes;
- auto-generation of customized work products based on project data;
- creation of uniform patterns for modeling systems (a set of scenarios that allow you to consistently receive the necessary set of documents).

In 2018, the Russian branch of the system engineering community INCOSE initiated the SysArchi project. The project showed that system modeling of target and supporting systems within ArchiMate 3.0 is extremely inconvenient. This success finally contributed to the development of the project to create a system modeling platform for target and supporting systems.

2.4. Definition of the target system

Having identified the shortcomings of existing solutions and the needs in the subject area, we turn to the description of the target system.

Despite the existing backlog in understanding the functions of the system and its structure, first of all, it is important to consider the target system as a black box, determining its main function, inputs and outputs. The black box model is shown in Figure 3.

The main function of the target system is the “Design quality assurance system”.

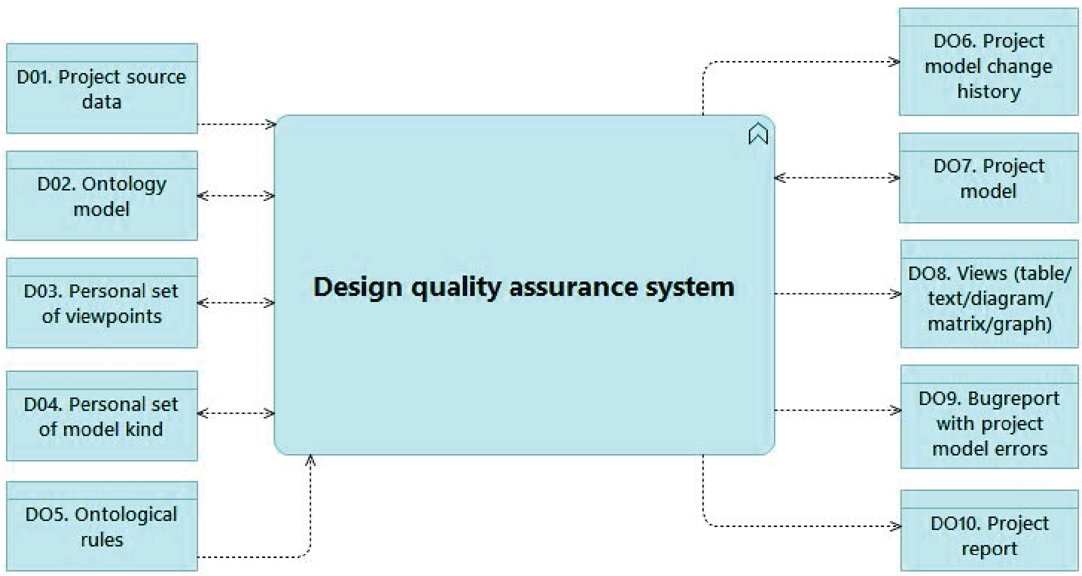

Figure 3: Context model system-of-interest

2.5. System-of-interest definition

Having dealt with the main function and the input / output streams, we will return to the processes of the life cycle. A simplified operating context model (OpsCon) allows you to focus specifically on the target system and the supported life cycle processes, respectively.

Through the appropriate software service, the target system ensures the implementation of the processes shown in Figure 4.

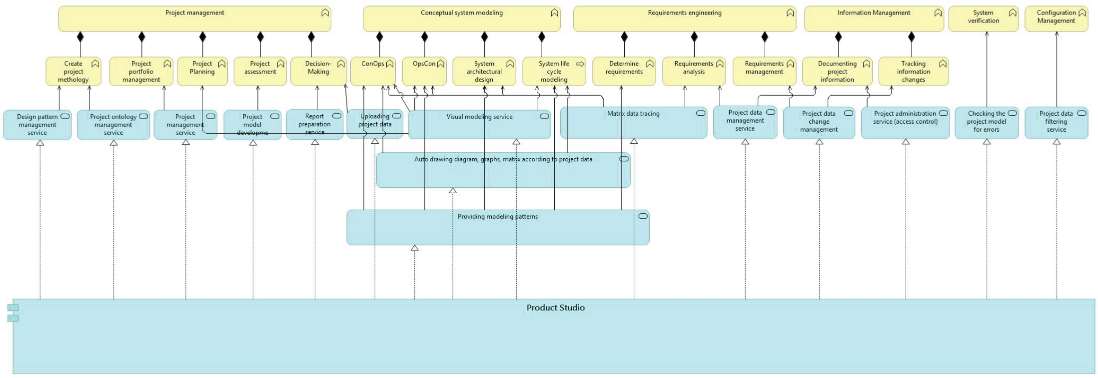

Figure 4: Application service system-of-interest

In general, the Product Studio development team hopes that the system will solve the problems described and simplify the development of complex systems by:

- providing tools for the formation and editing of a unified ontology and its own patterns of modeling systems;
- maintaining a single database of all project data with the ability to track changes (filtering project data depending on the context and concern of the stakeholders);
- access control to project data;
- auto-generation of the necessary representations (matrices / columns / diagrams) based on the database;
- checking the project model for system errors;
- auto-generation of work products for coordination and decision-making regarding the system being developed.

References

[1] Cecilia Haskins System engineering handbook, A guide for system life cycle processes and activities, INCOSE-TP-2003-002-03.1, version 3.1, August 2007.

[2] Mnemonic for remembering the properties of the Cost of Change curve [Electronic resource] / Milos Bejda, Access Mode: https://www.mbejda.com/mneo/.

[3] 15288-2015 - ISO/IEC/IEEE International Standard - Systems and software engineering – System life cycle processes [Electronic resource] https://ieeexplore.ieee.org/document/7106435.