

# Life Cycle Support Processes and Sustainment: Goals and Differences

Boris Pozin<sup>1, 2</sup> [0000-0002-0012-2230]

<sup>1</sup> National Research University Higher School of Economics, Moscow

<sup>2</sup> EC-leasing, Moscow

bpozin@ec-leasing.ru

**Abstract.** The approaches to optimization of total cost of ownership in the life cycle of software of information systems for various purposes: critical systems, primarily for public use, as well as the so-called Software – Intensive Systems, primarily weapons systems (aviation, space, guided weapons). The first group of systems in the Russian literature is called the processes and Life Cycle Support System of Mission-Critical System, the second in American sources is called Sustainability - processes. The goals and differences of both approaches are analyzed. Recommendations on the use and development of the achieved results are formulated.

**Keywords:** Life Cycle, Life Cycle Support Processes, Life Cycle Support System, Sustainment.

## 1 Management and Economic Aspects of Software Life Cycle for Systems of Different Classes

### 1.1 Software for Systems of Different Classes

Systems for various purposes differ in complexity and cost depending on the required functionality, quality requirements, the degree of severity of real-time regulations, the degree of change in the initial requirements during the development of the system (software), the intensity of changes in requirements and software during system maintenance. A significant role is played by the number of copies of the system, that is, the number of copies simultaneously in operation, and requiring changes to each instance. On the other hand, the role of centralized information processing systems, including data processing using cloud technologies, is increasing. This leads to the need to improve the processes and technologies of operation and maintenance of both unique software and COTS – applications included in the systems.

Since 2003-2009, a common approach to solving these problems is the approach to reducing (optimizing) the total cost of ownership throughout the life cycle of systems – from the formation and engineering requirements to the creation, implementation and utilization of automated technologies of changes at all stages of the software and system life cycle. Moreover, we are talking not only about the developers of systems

and their software, but also stakeholders: customers, specialists from organizations - owners of systems and organizations – system operators and organizations - maintainers.

This direction simultaneously developed in various organizations of our country, engaged in the development of systems of centralized processing of information mainly for public purposes. A special feature of the developing technologies and tools was the need to organize the development, maintenance and development of systems and their software in parallel with their operation.

In parallel, the SEI Institute in the USA developed an initiative in the same direction in relation to software of another type of systems – Software – Intensive Systems. This type includes control systems for aircraft (helicopters, airplanes, etc.) and other defense systems. Systems of this type are characterized by the complexity of the software and a fairly high cost, complexity and duration of changes. In addition, these are replicated objects. As a result – there is a high frequency of changes in requirements, the emergence of a large number of versions of configuration management objects (CMOS) and the need to make every change to all the same type instances of systems (copies of software), including taking into account the peculiarities of the production of CMOS. Accordingly, it is necessary to test versions of the CMOS, including in interaction with other CMOS that are part of the system.

## **1.2 Ways to Reduce Total Cost of Ownership**

The life cycle (LC) of system of interest and its software is the period from the emergence of the concept of creating a system to its removal from service. For the considered types of systems, the duration of the LC is about 15-20 years or more. The cost of the actual development of application software and its integration into the system in the LC is estimated at about 20% of the total cost. The remaining part is the cost of operation, maintenance and development of the software. Such costs cannot be uncontrolled and unregulated. In addition, relations between investors, owners, operators, specialists in support and development should be built on the scale of the operated system, regulations should be established, and the infrastructure for making, testing changes, technical maintenance of the system and its elements should be built. The need to create such a system for life cycle support is prescribed by the international standard ISO/IEC/IEEE 15288 [1] - Enabling System: a system that complements the system of interest during the stages of its life cycle, but does not necessarily directly contribute to its operation.

Systematic work is needed to maintain the consistency of the system's functional characteristics and purpose indicators in a state of usability, and thus to maintain the system as an asset.

Of course, for systems of sufficiently high complexity and multiplicity, the development and commissioning of processes and procedures of such a level, the creation of such an infrastructure to maintain a certain level of quality of functioning, the speed of changes and guarantee their quality is not a cheap project. However, the creation of an enabling system essentially has no alternative, because it is an organizational and technical mechanism to combat the main potential losses that need to be

reduced or minimized: the risks that are implemented at the stages of operation and maintenance of systems. The cost of implementing the risk at these stages can be comparable to the cost of creating the asset, that is, to the cost of developing the system. Or surpass her. Preservation of the asset requires additional organizational and technical resources: personnel, equipment, tools, regulatory and methodological support of the system in the LC.

In our opinion, the main directions of reducing the total cost of ownership (TCO) are:

- improvement of the processes of financing projects for the creation of automated systems, taking into account the issues of ensuring their life cycle
- inclusion in the structure of the created systems of the necessary solutions to ensure the life cycle as mandatory subsystems, technological equipment, providing increased productivity and quality of operation, maintenance and development
- creation, implementation and maintenance of appropriate complexes of tools that guarantee the ability to control the quality of changes and indicators of the purpose of the operated, accompanied, developed system while maintaining high productivity of personnel at all stages of the LC of this system.

### 1.3 Support for System Life Cycle and Sustainment

The literature distinguishes between the life cycle processes of systems and Software Sustainment (or Sustaining). The first group consists of the processes that describe the enabling system [1], that is, they allow to organize activities for the creation, maintenance, operation, maintenance and development of all components of the target system as a whole, including software throughout the life cycle. The second group consists of processes regulated [2,3]. It is clear that the goals and objectives of both groups of processes are close, require considerable attention and significant means of implementation. Thus, according to [6] financing of software maintenance and support processes is currently estimated at approximately \$ 5.6B per year; at least 15,000 civil servants and several thousand in commercial companies are employed in this field.

The main processes of LC systems of the first of the above types are described sufficiently fully in [4]. To them, perhaps, it is possible to add only the processes of requirements engineering regulated in [5] and described in detail in [6].

The main problem in the creation of both types of systems are the processes of supporting the software LC as the most flexible part of the systems. The apparent simplicity of making changes to the software of target systems is actually quite a complex set of activities on the implementation and commissioning of complex decision-making systems. This is true both for fairly universal systems for the field of application (but related to decision support, for example, in the financial field or in the field of management of material flows, objects, industrial facilities), and in the field of control systems of material objects (vessels of all types, processing centers, etc.)

Changes to operating software always involve a number of risks due to:

- lack of certainty as to the purpose of the changes,

- insufficiently clearly defined quality criteria for the evaluation of the work performed,
- a large number of links affected by change,
- it is not always clear to the customer and the developer of the changes about the degree of impact of these changes on the already achieved by the system purpose indicators and achievable as a result of changes in the value of these indicators (for example, productivity),
- and a number of other factors.

For systems of the second type, which are called Software – Intensive Systems, the SEI since 2005 forms a common approach to the restructuring of the industry to new tasks of building industrial technology for such systems [7]. The authors therefore rightly separate the notion of Software Maintenance from the software Sustainment. The first is defined as " the process of modifying a software system or component after delivery to correct errors, improve performance or other characteristics, or adapt to a changed environment." In contrast, Software Sustainment is: "the processes, procedures, people, materials, and information required to support, maintain, and operate the software aspects of the system."

## **2 Technological Solutions: What are the Differences**

### **2.1 Systems Software Life Cycle and its Coverage**

The need to reduce the operational risks of systems, including Software – Intensive Systems, in the context of the development of requirements for systems in the course of their LC led to a change in the point of view of the processes of ensuring sustainable changes in the operating system together with teams of customers, operators, system support, as well as specialists engaged in the development of the functionality of systems according to the requirements of customers (stakeholders). Software maintenance includes correcting the faults; improving performance or other attributes; adapting to a changed environment. However, for the software included in the target system, this is not enough to support the processes of ensuring the functioning of the software in the system and for the processes of functional and non-functional development of the software in the system [4]. The above-mentioned approach, called Sustainability, significantly expands the set of processes and operations that should complement the maintenance of systems in an efficient condition. Such aspects of Software Sustainability, which are used in different systems, although not necessarily all: operations, documentation, deployment, security, configuration management (CM), training, help desk, COTS management, technology refresh. Thus, for the software, which is a part of the systems, it is very important to coordinate the processes of support and development of this software synchronously other activities to support the system as a whole. This requires serious work on planning and coordination of work (governance) of all participants of the system and its part of the software. Coverage of all stages of the target system's LC and its software with processes according to the agreed goals and results, reliable material support of these processes can significantly reduce the total cost of ownership in the LC system and software.

## 2.2 Infrastructure and Regulation of Support Processes

In the collective work to support software of target system are changing in the direction of system integration and testing on target platform. That means that central issues became ensuring the activities of specialists who carry out the assembly (integration) of the system, comprehensive testing and testing of purpose indicators of the system on the target computing platform and in the processing of information close to the real. It is the solution of these tasks should be oriented and infrastructure to support processes and procedures of staff interaction.

Infrastructure for Software Support (Sustainment) has not to be separate. It partly must be part of the overall ITSM infrastructure (in the CM, help desk, documentation part) of the target system. At the same time, additional technological stands should be created for testing various levels of readiness of the developed (target) system. Such stands should have different levels of information security to perform complex works on automated complex functional testing of the system as a whole, as well as for load testing of software on the target architecture in order to assess the achievable purpose indicators of the system equipped with modified software (release or version of the software).

Quite large teams of specialists of various qualifications are involved in the maintenance and development of the system. The coordinated work of these specialists requires regulation of their activities.

The basis of the regulation is the careful setting of goals in various areas of activity in accordance with the established technological process of work and the established expected results of work. These objectives should be set out in the Support Plan document.

The Support Plan should be based on the selection of a reasonably valid software scheme of software releases over the medium term, for example, four software releases during the year on a quarterly basis. At the same time, each release should be defined by functionality, that is, by the composition of the requirements implemented in it. Planning a series of releases allows all participants to pre-distribute efforts at clear planning intervals and form a single series – Stable Software Production Baseline for the year. Under this medium-term plan, it is easier to plan the finalization of documentation and training of all user groups (Sustainability staff training plan). Under the same schedule it is easy to plan a plan of testing and evaluation of system characteristics and the use of stands. The operations service can also better organize its work, understanding the General scheme of work. The described organization of work allows you to build a well-planned process with estimated and predictable characteristics.

## 2.3 Level of Automation

The formation of a stable technological production process allows us to formulate requirements for its automation. With a well-organized technological process, it is possible to reduce the variety of automation tools used and focus on the most time-consuming operations and chains of operations that use the same data with a single

level of information security. The center of gravity in the automation of course should be on the operations of integration of the system and complex testing. Automation of the most labor-intensive operations and processes will lead to a significant reduction in the complexity of work and increase in collective productivity of specialists-participants of the software life cycle.

### 3 Conclusion

The main technological direction of works on ensuring the life cycle of mission-critical systems for state and defense purposes is to reduce the total cost of ownership of the system and its software by streamlining, regulating and automating the life cycle processes. To date, in the domestic and foreign literature formed similar content approaches to solving this problem. The analysis of the available works shows the need to create targeted systems for the development of well-equipped centers of their technical support, which will significantly reduce the total cost of ownership of systems and reduce the risks of loss of quality of systems in their operation, maintenance and development.

### References

1. ISO/IEC/IEEE 15288: 2015 Systems and software engineering. Systems Life Cycle Processes.
2. ISO/IEC 14764:2006 Software Engineering -- Software Life Cycle Processes -- Maintenance
3. ISO/IEC 12207:2008 Systems and software engineering -- Software life cycle processes.
4. Pozin B.A. The Principles of Life Cycle Supporting System for Mission-Critical Systems. Trudy ISP RAN/Proc. ISP RAS, vol.30, issue 1, 2018, pp. 103-114. DOI:10.15514/ISPRAS-2018-30(1)-7
5. ISO/IEC/IEEE FDIS 29148:2017 Systems and software engineering. Life cycle processes. Requirements engineering.
6. Batovrin V. K., Pozin B. A. Requirements Engineering at a Modern Industrial Enterprise, Programmaya Ingeria, 2019, vol. 10, no. 3, pp. 114—124. DOI: 10.17587/prin.10.114-124
7. Lapham M.A. Sustaining Software-Intensive Systems – A Conundrum. (2005) In: NDIA Systems Engineering Conference 2005, p.p. 1-15.
8. Proceedings of Software Solutions Symposium, Carnegie Mellon University SEI (2017)