Scientific article
UDC 330.47
DOI: https://doi.org/10.57809/2023.2.4.7.3

ARCHITECTURE OF THE SOFTWARE ASSET MANAGEMENT SYSTEM

Sergey Blinov ✉

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ st068499@student.spbstu.ru

Abstract. Software Asset Management (SAM) system is a set of measures aimed at optimizing the acquisition, deployment, use and maintenance of software assets in organizations. Software asset management processes require not only a competent management system, but also support from IT services. This paper presents a way to define the SAM application architecture using a client-oriented approach. Terminological analysis was carried out, software management process was modeled, requirements for the application were compiled and the corresponding functionality of its modules was determined.

Keywords: software asset management, license management, software utilization, SAM architecture

Citation: Blinov S. Architecture of the software asset management system. Technoeconomics. 2023. 2. 4 (7). 25–37. DOI: https://doi.org/10.57809/2023.2.4.7.3

Научная статья
УДК 330.47
DOI: https://doi.org/10.57809/2023.2.4.7.3

АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ПРОГРАММНЫМИ АКТИВАМИ

Сергей Блинов ✉

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия

✉ st068499@student.spbstu.ru

Аннотация. Система управления программными активами (SAM) - это комплекс мер, направленных на оптимизацию приобретения, развертывания, использования и сопровождения программных активов в организациях. Процессы управления программными активами требуют не только грамотной системы управления, но и поддержки со стороны ИТ-служб. В данной работе представлен способ определения и построения архитектуры приложения SAM с использованием клиент-ориентированного подхода. В ходе исследования был проведен терминологический анализ, смоделирован процесс управления программным обеспечением, составлены требования к приложению, а также определена соответствующая функциональность его модулей.

Ключевые слова: управление программными активами, управление лицензиями, использование программного обеспечения, архитектура SAM

Для цитирования: Блинов С. Архитектура системы управления программными активами // Техноэкономика. 2023. Т. 2, № 4 (7). С. 25–37. DOI: https://doi.org/10.57809/2023.2.4.7.3

Introduction

Software asset management activities in the Russian Federation are regulated by the GOST R ISO/IEC 19770-1 – 2014 standard, which is identical to ISO/IEC 19770. This standard defines the basis of a comprehensive set of software Asset Management processes (Software Asset Management, SAM), divided into levels providing for the phased implementation, evaluation and approval of SAM processes.

According to the standard, Software Asset Management is effective management, control and protection of software assets across the organization and effective management, control and protection of information about related assets necessary for the management of software assets. Therefore, within a specific enterprise, the functionality of the SAM solution is determined through a set of processes and services that implement management, control and protection. Their list depends on the scale of the organization, the type of activity, the allocated budget, the maturity of related processes, the range of software assets used, the nature of relationships with software vendors, as well as the choice between in-house development of the SAM system and the acquisition of a ready-made solution (Spewak, 1992).

SAM is a component of IT Asset Management (IT Asset Management, ITAM). ITAM is a coordinated activity of an organization to obtain value from IT assets. The following types of IT assets are distinguished:

– software;
– media (physical and digital);
– IT equipment (physical and virtual);

– licenses (including license confirmation);
– contracts;
– ITAM-IT asset management systems (including ITAM systems and tools, as well as metadata needed to manage all IT assets).

The object of the SAM system is primarily licenses, as well as the corresponding software. When expanding the functionality of the system to other IT assets, it is advisable to talk about creating an ITAM system.

Returning to the definition of SAM, it is necessary to focus on why the task of managing, controlling and protecting software assets should be solved precisely through the creation of a separate IT solution. In order to answer this question, it is necessary to reflect the specifics of software assets as an object of management.

1. Dynamic nature of software and IT assets. A software asset, both separately and as a component of an IT system, may be subject to frequent updates, corrections and user modifications. Tracking these changes is necessary for making centralized decisions on a group of software assets, monitoring usage, protecting against violations of contracts with vendors or security threats.

2. Software complexity. The software can be used as a finished product or component. The same software asset can be used at the same enterprise, but with different functionality. Differences arise due to the role model (different access levels and divisions), as well as software versions (see point 1).

3. Licensing and compliance. Software assets have complex licensing agreements that need to be monitored to ensure compliance and prevent legal problems. Their specificity is not only in controlling the number (which can be measured both by the number of devices and users, both simultaneously and during any period, etc.), but also in terms of use (right to change), updates, technical support, security, access to documentation and, of course, available functionality. There are financial and legal consequences for violating the terms of use of the Software (including due to the lack of tools for software management).

The above are only some circumstances that make it difficult to account for software assets by classical asset management tools. Therefore, a separate IT solution is needed that takes into account the specifics of SAM (Kalyatin, 2020).

The purpose is to create the architecture of the SAM application.

To achieve it, it is necessary to solve the following tasks:

1. Outline the list of and define key concepts used in the paper (mainly by analyzing regulatory documentation)

2. Define problems of AS IS situation by modeling and describing software management processes to identify pains of business users

3. Synthesis of functional requirements for the application from the pains of process participants

4. Using methodology of enterprise architecture, create an application model.

Materials and Methods

Before proceeding to the development of solution to the previously identified set of problems, it is necessary to define the key terms that will be needed in framework of the study, including characterization of the proposed SAM tools. The terms and the corresponding definitions are discussed in the section "Literature Review" and allows to form the boundaries of research, separating software assets from IT assets. To ensure the unity of terminology, GOST/ISO standards are mainly used. The functionality of the service system was formed both on the basis of scientific publications and on the basis of existing ready-made solutions in the IT market.

After a single terminological field has been formed, it is still too early to proceed to the description of the solution. The fact is that the upper-level description of the problem does not have sufficient specifics to describe the functionality designed to ensure effective management of software assets. Therefore, the next step is to detail the problems to the level of groups of business users who are stakeholders in the process. The "pains" of the participants are formalized in the business requirements for the IT system (Clements, 2010).

The architecture of the proposed services is described below, taking into account the relationship between them and the problems corresponding to each of the components. After describing the elements of the software asset management system, a comparative analysis of the application implementation options is presented.

A software asset management system is an important tool for businesses and organizations, helping them manage the software used in their infrastructure. Within the framework of this system, the key concepts are software, application program and software architecture, which are highlighted below

Software is the main component of SAM. It includes programs, procedures, rules and any relevant documentation related to the operation of the computer system. It can be either system software necessary for computer operation, or application software designed to solve specific tasks, such as word processing or graphic editors.

An application program is a computer program designed to perform a specific task other than the one related to the operation of the computer itself, usually used by end users.

According to Clements et al., software architecture is a set of structures necessary for reasoning about a system, which includes program elements, the relationships between them and the properties of both. The software architecture of the system is a design solution related to the overall structure and behavior of the system. The architecture helps stakeholders understand and analyze how the system will provide such important qualities as modifiability, availability and security.

In the ISO standards, the author could not find a separate definition for the application architecture. Based on ISO/IEC 42010 IEEE 1471, it can be concluded that application architecture is one of several areas of architecture that form the foundations of enterprise architecture (EA).

Application architecture describes the behavior of applications used in business, focusing on how they interact with each other and with users. It is focused on the data consumed and created by applications, not on their internal structure.

The application architecture is determined based on business requirements and functionality. This includes defining the interaction between application packages, databases, and middleware systems in terms of functional coverage (Maidanova, 2020).

It answers the question: “What computer systems (applications) are required to provide the information necessary for business functions?”

There are several approaches to determining the lifecycle of software assets. So, Fadi Nouh in the article SAM Software Asset Management defines the stages of the life cycle as follows.

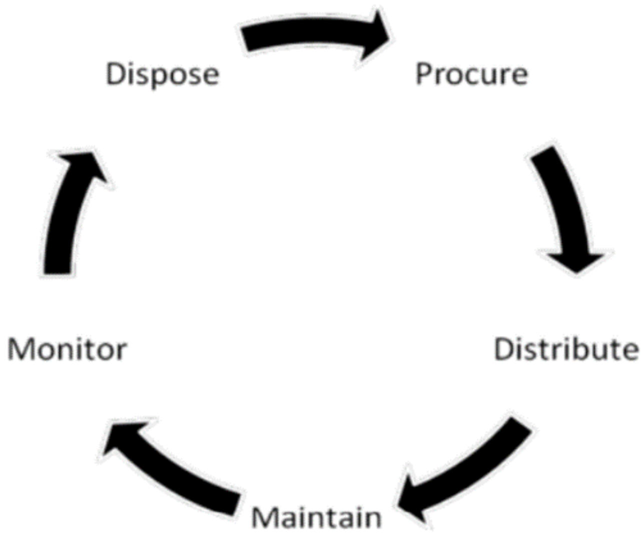

Fig. 1. Stages of the software asset life cycle (Fadi, 2016)

Michael Stone et al. in the work of IT Asset Management, proposed to use, with reservations, a typical asset lifecycle scheme when considering software assets.

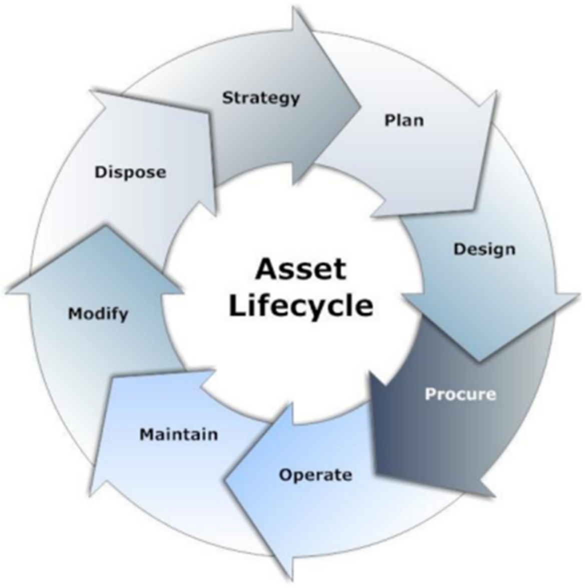

Fig. 2. Typical asset lifecycle (Stone, nd)

The emphasis in these life cycle models is on the sequence of stages and the cyclical nature of the process as a whole; the possibility of returning to previous stages before completing the full cycle is not reflected. In addition to the previous models, the following life cycle model is proposed to focus on generalizing groups of processes.

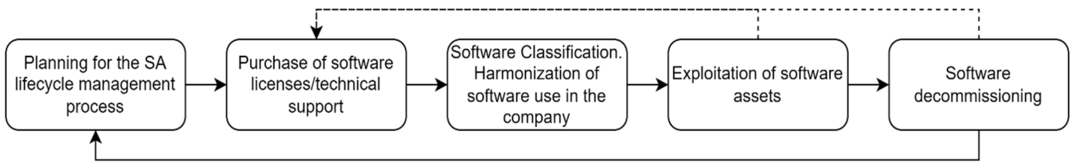
Planning for the SA lifecycle management process
Purchase of software licenses/technical support
Software Classification. Harmonization of software use in the company
Exploitation of software assets
Software decommissioning

Fig. 3. Software asset management lifecycle

Results and Discussion

Challenges that stakeholders face

This work is based on the general assumption that a company with a large portfolio of software products decides to initiate the development of a SAM system in order to further improve the level of software asset management. When identifying internal stakeholders, user groups have been formed that are most affected by the potential introduction of a new informational system. The main challenges of the company's employees are presented below.

Table 1. Challenges of participants in the process of managing software asset

Business user	Challenges	Context
Employee	Difficulties of finding the right software Long time of software installation on workplace Long time of involvement in the workflow	Lack of a clear description of the software, lengthy approval process, possibility of installing prohibited software with administrator rights, absence of software in the catalog despite available licenses
IT Administrator	Inaccurate or incomplete list of software	Difficulty in locating and tracking all installed software, lack of automated software license inventory management, reliance on manual data entry, problems with maintaining and replicating frequent software updates and changes
Procurement Department	Lack of information on software licenses	Inability to promptly respond to changes in demand for software licenses within the organization, difficulties in justifying the position in negotiations with the supplier, violation of the basic principles of category management when working with counterparties.
IS (information security) officer	Compliance risks and audits	Potential non-compliance with software licensing agreements, difficulties in ensuring that license records are accurate and up-to-date, problems in preparing for software audits and responding to license compliance inquiries
Software expert	Inefficient allocation of resources	Problems in optimizing software license allocation, determining the percentage of license usage, constrains in budget planning and cost optimization
IT support	Significant proportion of excessive requests	Increase in number of unstructured requests regarding installation, licensing, uninstallation, compatibility and software performance in general.
Product teams	Absence of centralized software information	The risks of using specific software and the limits of its use are not clear Lack of transparency in the activities of software product teams Problems with regulatory authorities

Functional requirements

Based on the pains of participants in the process of managing software assets, as well as according to the best practices of implementing the SAM concept, the following characteristics can be distinguished that an effective SAM solution should have:

1. Software inventory management:

a. The tool should provide the ability to automatically detect and inventory software resources throughout the network and infrastructure of the organization.

b. The SAM should support tracking and recording information about software installations, versions, licenses and access rights.

c. The tool should allow manual entry or import of information about software assets for assets that cannot be detected automatically.

2. License Management:

a. The tool should provide technical support for managing and tracking software licenses, including such information as license types, terms, conditions and restrictions.

b. The SAM should provide functionality to link licenses to specific software installations or deployments.

c. The tool should support license usage monitoring and notifications to ensure compliance with license agreements.

3. Compliance and reporting:

a. The tool should generate reports and provide dashboards that allow you to track the status of compliance with software licenses throughout the organization. This should allow you to create customized reports for auditing, license reconciliation, and compliance assessment.

b. The tool should support the ability to export compliance reports in common formats (e.g. PDF, CSV) for easy sharing and documentation.

4. Procurement and contract management:

a. The tool should integrate with procurement processes to track software purchases, contracts, and renewal dates.

b. The SAM should contain alerts and notifications about contract extensions, expiration dates, and important milestones.

c. The tool should support the collection and management of vendor information, including contact details and contract terms.

5. Use and optimization of software:

a. The tool should monitor software utilization rate and provide information about software licensing model metrics, helping to identify underused or unused licenses.

b. The SAM should form optimization recommendations to improve software distribution and reduce costs.

c. The tool should support the identification of software duplication and provide recommendations for consolidation or replacement.

6. Integration with IT infrastructure:

a. The tool should integrate with existing IT infrastructure and systems, such as configuration management databases (CMDBS) and IT asset management tools.

b. SAM should support automatic synchronization of software asset information with other systems to ensure data accuracy and relevance.

c. The tool should provide integration with deployment tools software or IT Service Management Systems (ITSM) to simplify software preparation and management.

7. Security and Risk Management:

a. The SAM tool should help identify security vulnerabilities related to software assets and provide mechanisms for tracking and managing patch and update levels.

b. It should support security risk assessment by providing information about software versions with known vulnerabilities or end-of-life status.

c. The tool should ensure the implementation and enforcement of security policies related to software use and access control.

8. Scalability and performance:

a. The SAM tool must be scalable to manage the inventory of the organization's software assets, regardless of the number of assets or distributed locations.

b. It must have effective performance capabilities that allow for quick search and retrieval of information about software assets.

9. User Access and Permissions:

a. The SAM tool should provide role-based access control to ensure that only authorized personnel can view and modify software data.

b. It should allow different levels of access and permissions depending on user roles and responsibilities.

10. User-friendly interface:

a. The SAM tool should have an intuitive and user-friendly interface that simplifies navigation, data entry and reporting functions.

b. It should offer search and filtering options to quickly search for specific software resources or create specific reports.

Based on the functional requirements, as well as the existing ready-made SAM products on the market, an overlapping list of application modules necessary for effective management of software assets in the organization was formed.

Software Registry module

The basic module of the SAM system is the main component that provides the necessary functionality. It includes the following functions:

1. Automatic standardization and ordering of software data;
2. The software catalog, which can be easily changed or expanded with new additions;
3. Classification of software into 16 groups, fixed in the standard GOST R ISO/IEC TO 12182-2002;
5. Automated software commissioning, software status display;
6. Linking software to software assets in management accounting;
7. Information about risks and limitations for OSS licenses;
8. Managing libraries and software components;
9. Mapping and automatic comparison of the business user's software list and the software catalog;
10. Categorization of access control software;
11. Definition of the software owner: definition of a financially responsible person or organization owning software assets;
12. Platform and Installation Type identification: Identification of available platforms and installation types supported by the software;
13. Reference documents related to the software.

Application Showcase module

The main functions of the module:

1. Ordering/installing software;
2. Software search;
3. Role Matrix;
4. Identification of available devices for installation;
5. Configuring Software Availability;
6. Delivery of applications in the App-V format;
7. Providing the ability to deliver and install applications;
8. The ability for managers to delete software;
9. Search for alternatives based on functions;
10. Role-playing system;
11. Publishing any software;

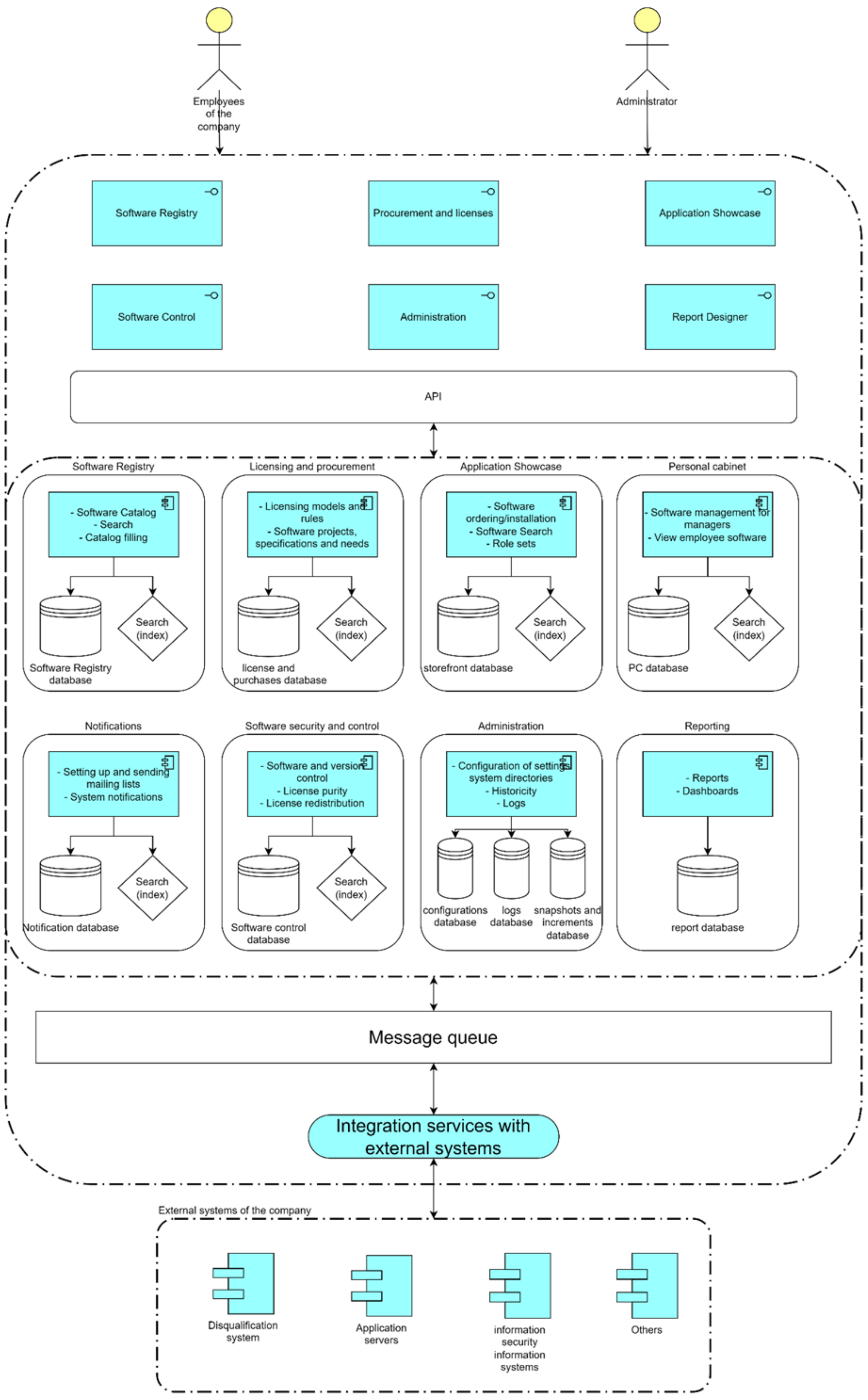

Fig. 4. SAM system architecture

12. Installing software for another employee.

Licensing and Procurement Module

Licensing models and rules: Catalog of current licensing models of software products used in the company, including:

– License types
– Allowed number of installations
– Binding to the core/device/employee/location

Software Procurement Projects and Specifications: A catalog of approved/protected software procurement projects integrated with the company's management and accounting systems, including:

– Project details and codes
– Software nomenclature
– The cost of the software at the time of purchase
– License validity period and technical support
– Contracts and Keys

Catalog of prices and suppliers: Catalog of current prices for items and software kits with information about suppliers, including:

– Current software costs
– Supplier Information
– Comparison of nomenclature names with catalog items

Justification of needs: A module for collecting information to justify needs, such as the actual use of the product for license renewal or rejected software installation requests due to lack of license.

License control and management: License expiration control, automatic distribution of license keys, flexible redistribution of paid software in case of shortage of licenses, management of license agreements, sublicensing and alienation of software.

Integration and procurement: Integration with external contract management systems to search for information, determine the cost of software for project protection, prepare specifications for investment planning, form a purchase queue and automatically determine prices based on previous purchases with correction factors.

Software security and control module

Control of portable/prohibited software is a tool to ensure operational control of portable and prohibited software in the company.

Basic definitions

– Tracking and management of prohibited and sanctioned software

Dynamic, changeable list of prohibited software

– Integration with application uninstall/blocking tools
– Notification of managers about the installation/launch of prohibited software

Version control is a tool for monitoring the versioning of the positions of the existing fleet of the company, as well as tracking functional and legal changes with the release of new versions.

Basic definitions:

– Use of current software versions
– Compliance with new copyright holders' requirements

Licensed cleanliness is a tool for ensuring licensed cleanliness. Comparison of data based on two variables: availability of a license for an installed copy of the software, availability of an agreed application (if required).

License redistribution is a tool that allows you to release licenses and redistribute them among company employees. For example, the release of a license due to dismissal or non-use for more than 90 days.

Discovery and inventory: automatic acquisition of data about workstations, laptops, servers, etc. and software installed on them

Directory of software recognition rules.

Reporting module

Graphical visualization of data. The module allows you to build dynamic reports and graphs for analyzing events related to software management.

Main types of reports:

Cost of ownership of software – a report that allows you to determine the cost of ownership of software by the consolidated parameter: manufacturer, product, functional unit, etc.

Unification of software versions is an analytical report that allows you to bring a large variety of versions of one software name to a single standard

The asset lifecycle is a tool that allows you to determine the entire chronology of the movement of each license from purchase to write–off.

Examples of indicators: software usage (% utilization) taking into account licensing metrics, license purity, financial indicators, validity periods of licenses and contracts.

Administration Module

SAM operation management: configuration settings, alerts, mailing lists, system directories. The administrator role is provided to perform the functions. Ability to configure and create snapshots/increments. The configuration should be made at least in terms of the schedule, data set, storage location, etc.

Personal account module

A tool for:

– Monitoring of the installed software of subordinate employees
– Calculating the cost of software ownership within its management level
– Configurations of non-standard role sets for their employees

Software lifecycle checklist linked to the calendar for those responsible for the software

Notification Module

Mechanisms for sending system notifications, as well as mailings to users according to specified rules.

Conclusion

Effective management of software assets implies the presence of an information system supporting the relevant business processes, which complements the landscape of the enterprise architecture in that part of the functionality that is not provided by standard asset management systems, such as, for example, a procurement management system, inventory management system, security systems, etc. At the same time, the system should ensure integration with related components of the enterprise architecture to avoid duplication of tools with different software solutions.

The architecture of a software asset management system is a structural design and organization of the system itself. It defines the composition, structure and relationship of functional components, modules and data for managing software assets in an organization.

It helps to identify approaches to digitalization of tools to meet the needs of business users.

The implementation of the main functionality of the SAM application is possible within the framework of 8 interrelated modules: Software Registry, Licensing and Procurement, Application Showcase, Personal Account, Notifications, Software Security and Control, Administration, Reporting. The software management tool is aimed at increasing license utilization, minimizing legal, financial and operational risks when using software, increasing awareness of key indicators of software asset management, improving the efficiency of software lifecycle management business processes and providing complete and structured information about the company's software assets.

REFERENCES

Clements, Wesley A. 2010. Documenting Software Architecture: Views and Beyond (2nd Edition) URL: https://www.iso.org/standard/63598.html (accessed: 05.11.2023).

Fadi N. SAM Software Asset Management. ResearchGate URL: https://www.researchgate.net/publication/291818013_SAM_Software_Asset_Management (accessed: 06.11.2023).

Kalyatin V. 2022. Establishing of Subject of Rights to Intellectual Property Created with Use of Artificial Intelligence Law. Journal of the Higher School of Economics, 4, pp. 24–50.

Maidanova S.A., Ilin I.V. 2022. Development of digital transformation strategy in the context of enterprise architecture. Technoeconomics, 2, 1 (4), 64-75.

Spewak S., Hill S. 1992. Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology.

Stone M. nd. IT Asset Management. doi: 0.6028/NIST.SP.1800-5

Applications architecture. URL: https://en.wikipedia.org/wiki/Applications_architecture (accessed: 05.11.2023).

GOST 33707-2016 (ISO/IEC 2382:2015) INTERSTATE STANDARD OF Information technologies. Vocabulary. URL: https://docs.cntd.ru/document/1200139532 (accessed: 06.11.2023).

GOST R 57100-2016/ISO/IEC/IEEE 42010:2011 NATIONAL STANDARD OF THE RUSSIAN FEDERATION Systems and software engineering. Architecture description. URL: https://docs.cntd.ru/document/1200139542 (accessed: 05.11.2023)

GOST R ISO/IEC 12207-2010 IS THE NATIONAL STANDARD OF THE RUSSIAN FEDERATION Information technology. System and software engineering. Software life cycle processes. URL: https://docs.cntd.ru/document/1200082859?marker=7EI0KJ&section=text (accessed: 05.11.2023).

GOST R ISO/IEC 19770-1-2014 IS THE NATIONAL STANDARD OF THE RUSSIAN FEDERATION Information technologies. Software asset management. URL: https://docs.cntd.ru/document/1200116600 (accessed: 06.11.2023).

GOST R ISO/IEC TO 12182-2002 is the state standard of the Russian Federation Information technology. Categorization of software. URL: https://docs.cntd.ru/document/1200030161 (accessed: 05.11.2023).

IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://cabibbo.dia.uniroma3.it/ids/altrui/ieee1471 (accessed: 05.11.2023)

Naumen Software Asset Management. URL: https://www.naumen.ru/products/sam / (accessed: 05.11.2023).

Oxford Dictionary. URL: https://www.oxfordlearnersdictionaries.com/definition/english/application?q=application (accessed: 05.11.2023).

Software Architecture. Carnegie Mellon University. URL: https://www.sei.cmu.edu/our-work/software-architecture / (accessed: 05.11.2023).

СПИСОК ИСТОЧНИКОВ

Clements, Wesley A. 2010. Documenting Software Architecture: Views and Beyond (2nd Edition) URL: https://www.iso.org/standard/63598.html (accessed: 05.11.2023).

Fadi N. SAM Software Asset Management. ResearchGate URL: https://www.researchgate.net/publication/291818013_SAM_Software_Asset_Management (accessed: 06.11.2023).

Kalyatin V. 2022. Establishing of Subject of Rights to Intellectual Property Created with Use of Artificial Intelligence Law. Journal of the Higher School of Economics, 4, pp. 24–50.

Maidanova S.A., Ilin I.V. 2022. Development of digital transformation strategy in the context of enterprise architecture. Technoeconomics, 2, 1 (4), 64-75.

Spewak S., Hill S. 1992. Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology.

Stone M. nd. IT Asset Management. doi: 0.6028/NIST.SP.1800-5

Applications architecture. URL: https://en.wikipedia.org/wiki/Applications_architecture (accessed: 05.11.2023).

ГОСТ 33707-2016 (ISO/IEC 2382:2015) Межгосударственный стандарт. Информационные технологии. Словарь. URL: https://docs.cntd.ru/document/1200139532 (дата посещения: 06.11.2023).

ГОСТ R 57100-2016/ISO/IEC/IEEE 42010:2011 Национальный стандарт Российской Федерации. Системная и программная инженерия. Описание архитектуры. URL: https://docs.cntd.ru/document/1200139542 (дата посещения: 05.11.2023)

ГОСТ R ISO/IEC 12207-2010 Национальный стандарт Российской Федерации. Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств. URL: https://docs.cntd.ru/document/1200082859?marker=7EI0KJ&section=text (дата посещения: 05.11.2023).

ГОСТ R ISO/IEC 19770-1-2014 Национальный стандарт Российской Федерации. Информационные технологии (ИТ). Менеджмент программных активов. URL: https://docs.cntd.ru/document/1200116600 (дата посещения: 06.11.2023).

ГОСТ R ISO/IEC ТО 12182-2002 Государственный стандарт Российской Федерации. Информационная технология (ИТ). Классификация программных средств. URL: https://docs.cntd.ru/document/1200030161 (дата посещения: 05.11.2023).

IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://cabibbo.dia.uniroma3.it/ids/altrui/ieee1471 (accessed: 05.11.2023)

Naumen Software Asset Management. URL: https://www.naumen.ru/products/sam / (accessed: 05.11.2023).

Oxford Dictionary. URL: https://www.oxfordlearnersdictionaries.com/definition/english/application?q=application (accessed: 05.11.2023).

Software Architecture. Carnegie Mellon University. URL: https://www.sei.cmu.edu/our-work/software-architecture / (accessed: 05.11.2023).

INFORMATION ABOUT AUTHOR / ИНФОРМАЦИЯ ОБ АВТОРЕ

BLINOV Sergey V. – student.
E-mail: st068499@student.spbstu.ru
БЛИНОВ Сергей Валерьевич – студент.
E-mail: st068499@student.spbstu.ru

Статья поступила в редакцию 29.11.2023; одобрена после рецензирования 15.12.2023; принята к публикации 18.12.2023.

The article was submitted 29.11.2023; approved after reviewing 15.12.2023; accepted for publication 18.12.2023.