TERMS OF REFERENCE
FOR THE PURCHASE OF
ITSM SYSTEM

BISHKEK 2022

Table of contents

ABBREVIATIONS AND SYMBOLS

Term	Definition
RPO	Recovery point objective - the allowable amount of possible data loss in the event of a failure (incident).
RTO	Recovery time objective - the allowable recovery time of the information system in the event of a failure (incident).
DC	Data Center is a specialized designated room to host server and network equipment that ensures the smooth operation of the company's IT Systems.
DB	Data Backup is a consistent copy of data on removable media (hard disk, floppy disk, etc.) designed to restore data to its original or new location in case of damage or destruction.
ITIL	IT Infrastructure Library is an information technology infrastructure library.
ITSM	IT Service Management is an approach to the management and organization of IT services aimed to meet the business needs.
Incident	Unplanned interruption or loss of quality of IT services.
Service request	Request from the user to provide something.
ID	Identification number
RFC	Requests for Change
KB	Knowledge Base
CU	Configuration unit
CMDB	Configuration Management Data Base is a configuration management database; a repository that contains the necessary information about the IT infrastructure hardware and software components.
Baseline	Configuration selected and fixed at any stage of the development lifecycle as a basis for further work.
CAB	Change Advisory Board that authorizes changes and helps change management evaluate and prioritize changes.
Dashboard	Dashboard that displays real-time data
IVR	Interactive Voice Response

1. General information

1.1. Name

Full name - Terms of Reference for the Purchase of
ITSM system (IT Service Management).

1.2. Supplier and Contractor

Client: Kumtor Gold Company CJSC
Software supplier: organization selected by the Client for the software delivery under this TOR.

2. Basis, purpose, and objectives of the purchase of ITSM system

2.1. Basis

The basis for the purchase of ITSM system is the need to build an effective system to monitor the operation of corporate services - from designing a service approach to automating business processes of various company's subdivisions and automating key functions based on a single platform, thereby reducing the costs of IT infrastructure support, and minimizing the number of IT systems used.

2.1. Purpose of the system

The main purpose of the system is:

a. Creation of a self-service portal that will allow users to independently solve tasks such as: create requests for the services and servicing, track the progress of the execution of requests, find solutions in the Knowledge Base.

b. Automation of the process of assigning requests and incidents to the Contractor, as well as control of execution by notifying responsible persons.

c. Monitoring of the service delivery process and the entire IT infrastructure; dashboards with real-time data on incidents and requests.

d. Effective process management to reduce the cost of physical assets and work resources.

e. Creation of reports based on any of the metrics with submission of information about the progress and performance.

f. Control over IT assets (equipment, software) to improve the efficiency of their use, modernization, and development.

g. Integrated approach to providing services, eliminating incidents, supporting users, and changing the IT infrastructure.

h. Management of financial processes of the IT department: procurement, budget, cost of services.

i. Collecting feedback from users to improve the service quality.

j. Prioritizing requests based on the type of request, specific user, or other parameters.

k. Escalating requests and incidents, notifying relevant administrators.

l. Storing a knowledge base on past requests, which allows specialists to quickly solve problems like those that have already arisen.

m. Reducing risks by implementing process controls that track all related activities.

2.2. Project Objectives

The main objectives of the project are:

a. Centralized management of IT infrastructure, strategically focused on the provision of services and oriented on the user of these services.

b. Reducing the time for processing user requests and incidents.

c. Providing management information and developing suggestions to improve services.

d. Reducing troubleshooting time by covering all independent organizational subdivisions, which allows you to access the relevant information about tickets, incidents, problems, changes, and resources using a single panel.

e. Increasing the user service level by controlling the SLA for the created services.

f. Increasing the transparency of business processes of various subdivisions of the Company.

g. Reducing the cost of processing standard requests and incidents.

h. Planning the workload for employees of various subdivisions of the Company.

i. Measuring satisfaction of the users.

j. Improving the reliability of the most important systems for business by
reducing the time to solve a problem or eliminate an incident.

k. Reducing the duration and number of calls from users.

l. Improving the productivity of the user support team.

3. ITSM System Requirements

3.1. Functional Requirements

The system shall be able to implement the following ITIL 4 processes:

Service Strategy
Strategy Management for IT Services
Business Relationship Management
Portfolio Management
Financial Management
Service Design
Service Catalog Management (SCM)
Service Level Management (SLM)
Capacity Management
Availability Management
Supplier Management
Service Transition
Change Management
Service Asset and Configuration Management (SACM)
Knowledge Management
Service Operation
Incident Management
Problem Management
Event Management
Request Fulfillment
Access Management
Continual Service Improvement
Reporting Management

3.1.1. INCIDENT MANAGEMENT

3.1.1.1. Module Objectives

a. Building a functional structure of the service for further automatic or manual distribution of responsibility of specialists to solve incidents and tasks/work orders.
b. Ensuring control over the workload of employees of various support lines and functional teams.
c. Monitoring the progress of all requests.
d. Tracking the time characteristics of the execution of all requests according to the quality parameters defined at the SLA level.
e. Increasing the transparency of all incident resolution processes through a flexible notification and escalation mechanism (hierarchical and functional).
f. Using the Knowledge Base when resolving user requests.
g. Using the CMDB to track links between CUs.

h. Formation of personalized reporting for the purpose of continuous monitoring of key performance indicators of the support service.
i. Communication with users, including receiving feedback from them in various ways: comments and correspondence on requests and service ratings.

3.1.1.2. Incident lifecycle

The system shall have a tool to implement the entire incident lifecycle (***Appendix 1***).

Detecting incidents

a. Manual and automatic generation of incident records.
b. Records can be created by IT service users using the self-service portal.
c. Records can be created by IT staff on behalf of the user.
d. Sources of outgoing request via Telegram bot, Chat bot
e. Automatic registration of requests received at the support service email address.
f. Registration of incidents by phone (see Integration).
g. Using customizable web forms.
h. Automatic registration of infrastructure incidents and events through the built-in monitoring and inventory module.
i. Registration of requests from corporate portals, websites, and external business systems through integration.

Registration of incidents

Assigning a unique ID to each incident record.

Fixing the time and date of record creation and its subsequent modification.

Saving full contact information in the record (initiator's full name, feedback method).

Recording information about the source of the incident report (person, event, group).

Categorization and prioritization

Separation of incidents and service requests.

Determining the category of incidents based on a customizable directory with a hierarchical structure.

Prioritization of incidents in accordance with the set priorities in manual mode or pre-configured conditions.

Automatic calculation of priority based on information about the SLA, the type of configuration unit, the service that has failed, the degree of its malfunction, etc.

Ability to adjust the priority during the processing of the incident with the recording of changes for subsequent audit and reporting.

Ability to change request type from incident to other type (service request, problem, request for change).

Support service and management tools that help you set the procedure of incident processing in accordance with business objectives.

Primary diagnostics

Recording the signs of failure and the results of diagnosing its causes.

Support for templates used for incident registration, which may include a list of necessary processing steps. They can be defined for services, SLAs, configuration unit types, and incident categories.

Automatic comparison of incident records with data on identified problems and known errors.

Notifying users and support services of the occurrence of already known errors that may be related to the incident being registered or processed, based on information about its category, configuration unit or service.

Escalation

Functional – based on manually defined or pre-configured conditions (service level and operational level target, business priority and support level).

Hierarchical – based on pre-configured (in SLA) and manually set conditions, including notification of the contractor, responsible employee, business client.

Diagnostics and research

Built-in CMDB configuration management system that allows you to identify, analyze and diagnose incidents.

Establishing and maintaining a connection between an incident and a configuration unit record.

Establishing and maintaining the connection of an incident with one or more problem records.

Ability to create a problem based on analysis of previous similar registered/resolved incidents.

Ability to create a service request from an incident record, as well as establish a connection between it and a service request.

Ability to open a request for change (RFC) from an incident record, as well as establish cause-and-effect relationships between events and requests for change.

Incident Resolution and Recovery

Assigning performers responsible for the elimination of the incident (group, department, or employee).

Storing information about the status of the incident.

All authorized users have access to the incident data, while restrictions are imposed on operations with key attributes (priority, status, queue position), which are available only to technical support staff or other responsible persons.

Introduction of a resolution with the possibility of differentiation for viewing for IT specialists/users.

Saving in the incident history complete data on the actions performed in the process of resolving and restoring the service, and the employees who performed them.

Automatic bulk operations with incident records (classification, creation, and consolidation of records).

Automatic control and monitoring of reaction time and problem resolution according to the service level and/or priority.

Closing the incident

Recording data on the resolution/closure of the incident, including the time and date.

Assessment of user satisfaction with the results of work to eliminate the problem.

3.1.1.3. Required features

1. Use of predefined actions of objects that perform various operations according to certain rules.
2. Separate processing flows for common incidents, major incidents, and infrastructure incidents.
3. Building hierarchical incident structures (master incident and dependent incidents).
4. Ability to change request type from Incident to Service Request, Request for Change, Problem.
5. Closing/reopening the incident.
6. Assigning one incident to multiple IT specialists.
7. Ability to create additional sub-requests - work tasks assigned to different employees and ability to assign a deadline for each task.
8. Assignment/reassignment of responsible persons.
9. Working with SLA timers (time counter, stop, start, waiting for delivery, etc.).
10. Setting the incident as a Critical incident (Major Incident).
11. Planning future actions. Entering actions that will be performed at a certain time in the future. Reminders about scheduled tasks.
12. Ability to display a complete list of actions on the incident for printing or sending a report by email.
13. Ability to reserve an incident number before it is saved.
14. Ability to create links between events, group them. Any actions can later be applied to the entire group at once, if necessary.
15. Recording and managing critical incidents.
16. Ability, as part of the unified management of IT services, to create a connection with problems, RFCs, CUs, known errors.
17. Confirmation of the acceptance of the IT event by a team of specialists as an indication in the general list of incidents.
18. Indication of the need for initial communication with the user in the general list of incidents.
19. Statistics on processing incidents in real time.
20. Using the search for "similar incidents" to solve incidents as part of first line support. The search is carried out by CUs, affected user, category, etc.
21. Ability to configure the processes applicable to incidents for their further full automation.
22. You can create entities based on the incident: problem or change (process, tasks, etc.).

23. Function that allows you to paste text, tables, videos, and images from the clipboard as values, including screenshots.
24. Automatic rejection or confirmation of the decision by the system participant using the self-service portal or by e-mail.
25. Communication between employees of various support lines, as well as users.
26. Setting up email and personal account notifications for various parameters: creating applications, changing status, assigning applications, and setting deadlines, etc.
27. Ability to flexibly filter the list of all incidents by various parameters: status, urgency, assigned employee, etc.
28. Color highlighting of requests by various parameters (deadlines, priorities, impact on user groups, etc.).
29. Monitoring the entire incident resolution process as you go through all stages.
30. Creating an incident form with the ability to determine which fields are required.
31. Calculation of labor costs for solving an incident, a separate task, an employee.
32. When assigning incidents by the team leader, it shall be possible to view the current employee load on open requests.
33. Ability to automatically assign an incident to an employee, depending on the type of request.
34. The priority is formed based on the urgency set by the user (can be revised by the support employee) and the influence set by the employee.
35. The system shall monitor incidents that exceed resolution times, service level inconsistencies, according to the specified metrics, and provide the ability to view data.
36. Control of the request processing time.

3.1.1.4. Interaction with other modules

Integration with service request management tools.

Integration with problem management tools.

Integration with knowledge base management tools.

Ability to create a Request for Change (RFC) from the Incident Management Module.

Integration with release and deployment management tools.

Integration with the Configuration Management Data Base (CMDB) and IT Asset Management.

3.1.2. MANAGING SERVICE REQUESTS

3.1.2.1. Module Objectives

a. Create a centralized portal focused on the services and oriented on the consumer of these services.
b. Speed up the execution of service requests.
c. Optimize request costs.
d. Monitor and schedule the workload of employees to fulfill service requests.
e. Provide communication between employees of different departments to reduce the time spent to fulfill the request.

f. Monitor and document the progress of work with requests.
g. Ensure the transparency of the work on the request by monitoring the entire process.

3.1.2.2. Service request lifecycle

The system shall have a tool to implement the entire lifecycle of service requests (***Appendix 2***).

Registering Service Requests

a. Records can be created by IT service users using the self-service portal.
b. Records can be created by IT staff on behalf of the user.
c. Sources of outgoing request via Telegram bot, Chat bot.
d. Registration of requests by phone, IVR (see Integration) – "speech to text".
e. Automatic registration of requests received at the support service email address.
f. Using customizable web forms.
g. Registration of requests from corporate portals, websites, and external business systems through integration.

Categorization and prioritization

Determining the category of requests based on a customizable directory with a hierarchical structure.

Prioritization of requests in accordance with the set priorities in manual mode or pre-configured conditions.

Automatic priority calculation based on information about SLA, type of configuration unit, etc.

Ability to adjust the priority during the processing of the request with the recording of changes for subsequent audit and reporting.

Service Request Approval

Redirection of the request for approval automatically according to pre-configured rules or manually by the responsible employee.

Delegation of authority

Redirection of the request for approval to the deputy of the responsible employee in case responsible employee is absent.

Service Request Completion

Assigning persons responsible for the request (group, department or employee);

Built-in CMDB configuration management database system, which allows linking service requests and CUs.

Ability to create a service request from an incident record, as well as establish a connection between it and a service request.

Storing information about the status.

Introduction of a resolution with the possibility of differentiation for viewing for IT specialists/users.

Saving in the request history complete data about the actions performed during the fulfillment of the request, and the employees who performed them.

Bulk operations with records (classification, creation, and consolidation of records).

Automatic control and monitoring of reaction and fulfillment time according to the service level and/or priority.

Service Request Closing

Recording data on the resolution/closure of the request, including the time and date.

Assessment of user satisfaction with the results of work on the request.

3.1.2.3. Required features

1. Creating service request records and storing detailed information about the request.
2. Ability to change the type of request from "Incident" to "Request for execution”.
3. Ability for users to see a description of the services available to them when creating a service request on the self-service portal.
4. Verification of the user's access rights to the requested service.
5. Recording the date and time when the service request record was created and modified.
6. Categorization of requests.
7. Tool for coordination and approval before executing a request.
8. Recording urgency, influence and priority information in the request record, ability to change these attributes during the request execution.
9. Functional escalation of responsibility for executing or approving a request based on pre-configured or manually selected conditions.
10. Hierarchical escalation of responsibility for executing or approving a request to a manager or a role specified in the SLA based on pre-configured or manually selected conditions.
11. Providing the user with detailed information about the status of the request.
12. Ability to match new service requests with existing ones.
13. Automatic routing of requests to performers: employees, groups, external organizations.
14. Using templates for the most typical service requests.
15. Ability to create rules and processes for specific requests or groups of requests, automating their processing and notifying interested parties.
16. Storing information about the request closing category.
17. Tools for analyzing service requests to identify trends.
18. Tools for conducting user satisfaction surveys.
19. Ability to resume processing of a previously closed request.
20. Automatic rejection or confirmation of the decision by the system participant using the self-service portal or by e-mail.
21. Communication between employees of various support lines, as well as users.

22. Setting up email and personal account notifications for various parameters: creating applications, changing status, assigning applications, and setting deadlines, etc.
23. Ability to flexibly filter the list of all requests by various parameters: status, urgency, assigned employee, etc.
24. Color highlighting of requests by various parameters (deadlines, priorities, impact on user groups, etc.).
25. Monitoring the entire process as you go through all stages.
26. Creating a request form with the ability to determine which fields are required to be filled in.
27. Calculation of labor costs for a request, a separate task, an employee.
28. When assigning the request by the team leader, it shall be possible to view the current employee load on open requests.
29. Ability to automatically assign the request to an employee, depending on the type of request.
30. The system shall monitor requests that exceed resolution times, service level inconsistencies, according to the specified metrics, and provide the ability to view data.
31. Control of the request processing time.

3.1.2.4. Interaction with other modules

Integration with incident management tools.

Ability to create a Request for Change (RFC) from the Service Request Management Module.

Integration with release and deployment management tools.

Integration with the configuration management system.

Integration with the service catalog module to support the creation of a service request by the user directly from the service catalog, as well as to create and maintain links between service request records and the service catalog.

3.1.3. PROBLEM MANAGEMENT

3.1.3.1. Module Objectives

a. Timely and effective identification of the causes of problems, including at the infrastructure level.
b. Preventive identification of potential problems and risks associated with their occurrence.
c. Assess the scale of the problem's impact on the business.
d. Monitor and document the work with problems.
e. Manage changes to solve problems.

3.1.3.2. Required features

1. Creating problems based on incidents.
2. Keeping problem records separate from incident records.
3. Automatic assignment of a unique ID when creating a problem record.
4. Recording the date and time when the problem record was created and updated.

5. Capturing the source of information about the problem (event, person, or group).
6. Creating a problem record in manual mode.
7. Categorization of problems based on data from a structured directory.
8. Prioritization of problems in accordance with pre-configured or manually specified conditions.
9. Automatic calculation of the initial priority of the problem based on the CU type information, the affected business service, the degree of service failure, the security vulnerability, the solution costs.
10. Ability to change the priority of a problem based on information about its impact and/or urgency.
11. Ability to automatically assign a responsible person (employee, group, department) for solving a problem depending on the category.
12. Storing information about the status of the problem (diagnostics, escalated, resolution, closed).
13. Fixing the failure symptoms in the problem record, including the event parameters and/or the user who reported it.
14. Recording information about the work done to diagnose the problem.
15. Fixing information about the problem resolution, including the date and time.
16. Escalation and notification of problems in progress when the threshold value is reached, for example, if the root cause is not established within the designated time period.
17. Ability to create records of known errors that become available to participants in related processes.
18. Access to archived data, information about problems and known errors for use in the diagnosing incidents and problems.
19. Categorization of the reasons for closing the problem.
20. Support for creating and using templates to solve problems.
21. Support for managing and documenting problem assessments.
22. Creating changes based on problems.
23. Distribution of work within the problems, decomposition of the problem into subtasks with the ability to assign different responsible persons and deadlines.

3.1.3.3. Interaction with other modules

Establishing and maintaining links between the problem/known error record and incident records.

Tools to proactively identify problems based on the analysis of incident trends.

Fixing a workaround solution found during the problem resolution process and publishing this information to related processes.

Creating a change request record based on a problem/known error record.

Creating and maintaining links between problem/known error records and change records.

Creating links between problem/known error records and configuration unit records.

Integration with a configuration management system that allows you to identify, investigate, diagnose, and fix problems.

Integration with the knowledge base to support research methods, diagnostics, and root cause analysis, as well as to create/update workaround solutions, temporary patches, and solutions.

3.1.4. KNOWLEDGE MANAGEMENT

3.1.4.1. Module Objectives

a. Organize the creation and maintenance of a knowledge base.
b. Use knowledge base materials in various management processes.
c. Receive feedback from users on the quality of materials presented in the knowledge base.
d. Minimize the involvement of support team and provide proactive mechanisms to help users at various stages of processing requests.

3.1.4.2. Required features

1. Customizable levels of access to knowledge base (KB) materials for users and IT employees.
2. Ability of automatic verification when entering new data for compliance with the established requirements.
3. Generating a unique ID for each entry/item in the knowledge base.
4. Ability to place attached files of arbitrary formats in database records (messages, electronic documents, tables, multimedia content, etc.).
5. Single structured input method using forms for creating new records.
6. Creating and maintaining a link between the database records.
7. Automatic recording of information about the author, data owner, date of creation, etc. in the database record.
8. Tools for identifying redundant or duplicate information in one or more records.
9. Tools for identifying trends in the DB use.
10. Automatic notification of stakeholders about new knowledge base items/solutions suitable for them.
11. Monitoring the frequency of access to the DB item.
12. Search for data in knowledge base records by various conditions (topic, owner, date, keywords, etc.).
13. Search for content that is stored in different formats.
14. Presenting information on the relevance or importance of information when searching.
15. Archiving and deleting obsolete or unwanted information from the DB.
16. Ability to establish links with external data sources, as well as import data from external sources.
17. Tools for creating and maintaining FAQ for customers and users.
18. Ability to record time periods when the importance of the DB record increases for a certain employee or work group.
19. Tools of classifying data in the DB.

20. Coordination of knowledge base materials for their inclusion in general knowledge or knowledge for users.
21. Discussion of materials from the knowledge base, including by users.
22. Ability to submit an item for approval before publication.

3.1.4.3. Interaction with other modules

Ability for personnel responsible for incident management to create DB records.

Ability for the personnel responsible for problem management to create DB records.

Quickly create a knowledge base record from a Change Record (RFC) and establish a link between records.

Integration with the Configuration Database (CMDB) to support links between knowledge base records and configuration unit records.

3.1.5. SERVICE ASSET AND CONFIGURATION MANAGEMENT

3.1.5.1. Module Objectives

a. Keep records of IT assets and configuration units, understand their relationships and impact on each other and on services.
b. Distribute the role responsibility for the CU maintenance, automating part of the procedures of related processes.
c. Monitor the expiration of warranty periods and obligations under license agreements or contracts for the use of hardware, software, and hardware-software.
d. Reconciliation with the data of the financial system of IT assets (by actual and residual values).
e. Audit the existing IT infrastructure (software, servers, network devices).
f. Keep the configuration management database up to date.
g. Document all activities related to configuration units.
h. Support problem and change management processes by analyzing the impacts and trends of IT assets.
i. Strengthening IT security through full control over configuration units.
j. Improving the quality of financial planning by clearly defining all assets and their relationships.
k. Improving the quality of software license management.

3.1.5.2. Required features

1. Storing up-to-date and accurate information about all configuration units of the IT infrastructure.
2. Storage of information about the CU depending on the type (unique ID, type, name, description, version, location, delivery date, licensing details, owner, status).

3. Identify relationships between change requests, software licenses, software descriptions, locations for authorized system software.
4. Management of existing assets throughout the entire life cycle of each IT asset - from the moment of placing an application to its writing-off.
5. Adding various asset statuses and additional form fields.
6. Ability to automate the linking of assets to software licenses, lease contracts, maintenance, and support.
7. Tracking total cost of ownership, chargebacks, and depreciation.
8. CU support. CU communication with certain support contracts, SLA with external organizations, equipment suppliers.
9. CU movement. Entering and viewing all events related to physical movement, as well as any other changes to the CU configuration during its life cycle.
10. CU Relationship Explorer is a graphical representation of CU relationships. It makes it possible to compare the analyzed information about possible impacts and risks of CU, allows you to view CU, existing dependencies of CU-CU, CU-service.
11. Asset management. It can be used when combining individual CUs into groups, viewing existing relationships of CUs, monitoring the use of unified user licenses, printing reports on the relationships of individual CUs.
12. Configuration audit. Provides the ability to use barcode identifiers to identify the CUs. Uploading barcode files and automatically creating a list of CUs. Identification and elimination of discrepancies.
13. CU systems. CI groups are combined, for example, working physically together or performing the same functions.
14. Fast and flexible search for CUs according to the required parameters.
15. Storage of information about the CU depending on the type (unique ID, type, name, description, version, location, delivery date, licensing details, owner, status).
16. Adding asset statuses and additional form fields.
17. Automatic verification of the entered data for uniqueness and other conditions.
18. Establishing communication with other CUs when adding a record of a new CU.
19. Software control support at all stages of its life cycle: from design to operation.
20. Management and use of baseline (basic infrastructure states or CUs) that can be used to return to them.
21. Keeping the CU history, including installation dates, change and location records.
22. Visualization of links between the CUs in the form of a map.
23. Automatic identification of CUs that may be affected in the event of an incident, problem, known error or change.
24. Automatic update of the CU version number in case of a change in the CU version number that is its component.
25. Ability to link the configuration unit to the records of users using the CU.
26. Control of the amount of CU residues for use.

3.1.5.3. Interaction with other modules

Establishing and maintaining the links between the CU records and records of incidents, service requests, tasks/work orders, problems, and requests for change.

3.1.6. CHANGE MANAGEMENT

3.1.6.1. Objectives

a. Implement standardized change implementation processes that cover the entire lifecycle of a request for change - from planning, implementation to control verification.
b. Assessment of the impact of changes on business processes by analyzing the degree of risk and technical consequences of changes.
c. Increasing the percentage of efficiency by strengthening control over changes.
d. Reducing the time required to implement changes.
e. Clear control and detailed reports on the progress of changes.
f. Monitor all stages of the change and configuration process and thereby reduce the risks associated with the implementation of changes.
g. Optimize the procedure for prioritizing change requests and thereby provide support for the most important business services.
h. Reduce calls to the support service by minimizing failures associated with changes.
i. Plan the upcoming changes in coordination with stakeholders, monitor responsibility for the implementation of changes.
j. Keep records of the costs of preparing and implementing changes.
k. Analyze the results of changes.

3.1.6.2. Required features

1. Assigning a unique ID to each Request for Change (RFC) Record.
2. Ability to create changes based on problems and incidents.
3. Recording the date and time when the request for change record was created and further modified.
4. Monitoring and tracking changes throughout their life cycle (from preliminary assessment to closure).
5. Separation of changes into types and ability to configure the process to process changes of each type;
6. Categorization of changes according to their impact and priority.
7. Documenting the progress and results of the change approval procedure.
8. Ability to reject a request for change by a special role.
9. Ability to configure the authority of the Change Committee (CAB) members depending on their role.
10. Indication of the fact of the change approval.
11. Tools for preliminary assessment of the change.
12. Creating a schedule of changes with appropriate access control, reflecting all approved changes and notifications for users and IT staff.
13. Creating a schedule of expected downtime of services.

14. Monitoring the availability of a tested rollback or recovery plan when approving a change.
15. Reminders about readiness to check completed changes.
16. Recording information about changes that have already been checked.
17. Recording closing date of changes. Ability to specify the closing category.
18. Support for creating and using change templates.
19. Support for the implementation of standard changes.
20. Displaying standard changes in the change schedule.
21. Tools for analyzing, developing, and planning proposals for changes.
22. Maintaining the link between the change and its feasibility study, as well as documentation on risks and requirements.
23. Ability to specify the change impact (via links to CMDB, SLA or other information).
24. Documenting the scale of service change.
25. Manual and automatic allocation of responsibility to an employee or team.
26. Ability to divide the change into stages.
27. Calculation of planned and actual time spent.
28. Approval of changes via e-mail or through the self-service portal.
29. Use of complex approvals for important changes.
30. Ability to link an RFC to one or more incidents and problems related to this change.
31. The RFC can be linked to one or more CUs that will be affected by this change.
32. Managing complex changes. The RFC is linked to the process, which consists of a sequence of levels, and they are divided into tasks. In addition, each level can have subprocesses.
33. Availability of tools to automate the processes of making changes. Different employees or groups of specialists can be automatically assigned to different stages of the change implementation process.
34. Availability of a calendar of changes, which provides a graphic image of all planned and current changes, makes it possible to track possible conflicts as a result of intersections of stages of the change implementation.
35. Possibility to create the so-called "maintenance windows" - a period of time for any selected CU system, during which it is possible to make changes for this CU system. The corresponding period shall be displayed in the calendar of changes. If you try to create a change with this CU system in a time period that does not coincide with the maintenance period, a conflict occurs.
36. Possibility to suspend the progress of changes for selected systems at a pre-set time periods (scheduled audits, etc.) - "Freezing of changes".
37. Notification of changes and their schedule to the support service of users and user groups via email, web portal, etc.

3.1.6.3. Interaction with other modules

Establishing and maintaining links between change records and incident records.

Establishing a link between problem/known error records and change records.

Creating and maintaining links between change records and configuration unit records (CUs).

Tools for preliminary assessment and authorization of changes based on data from the Configuration Management Database System (CMDB).

Ability to coordinate and plan releases and deployments using change management tools.

3.1.7. SERVICE CATALOG MANAGEMENT

3.1.7.1. Required features

1. Centralized control of all services.
2. Placement of service descriptions in electronic form; tool to design and reconfigure services.
3. Managing the structure of hosted data about services (categories, groups, types).
4. Assigning levels of services.
5. Storing information about the current stage of the service lifecycle.
6. Tools for visualizing the catalog of services, taking into account dependencies and relationships between them.
7. Creating customizable service description templates.
8. Separation of service and technical catalogs.
9. Support of the service lifecycle, starting from the creation of the service and up to its cancellation.
10. Assigning tasks in automatic mode to responsible persons to fulfill requests for connecting services and their approval.
11. Organizing a search that makes it easier for users to place an order.
12. Tools for documenting catalog access rules.
13. Service catalog design functionality.
14. Creating a catalog of business services.
15. Ability to publish services on the self-service portal with the ability to restrict access to users, groups, or subdivisions of the Company.

3.1.8. SERVICE LEVEL MANAGEMENT

3.1.8.1. Objectives

a. Create a single source of information about all agreements with service consumers and external suppliers.
b. Manage the level of service provision under agreements.
c. Monitor the quality of service provision.
d. Measure user satisfaction.
e. Track the expiration dates of service contracts with consumers and external contractors.

3.1.8.2. Required features

1. Maintaining a list of active services.
2. Fixing the agreed terms of service provision.
3. Time tracking for any tasks in the system - incidents, incident tasks, change tasks.

4. Fixing service level targets.
5. Creating records for each service level requirement.
6. Creating records of Service Level Agreement (SLA) and ability to establish links with service level requirements.
7. Detailing the content of the SLA, including the date of agreement, coverage, contact information and targets.
8. Detailing the Operational Level Agreements (OLA).
9. Detailing agreements with external suppliers.
10. Editing SLA, OLA records and storing the history of all changes.
11. Support for monitoring operational level agreements and performance metrics.
12. Indication of the assessment of services with clients and contractors.
13. Providing a list of concluded and non-concluded SLAs, OLAs and external agreements for any service.
14. Monitoring the success of service level agreements.
15. Monitoring service availability and performance thresholds compared to those defined in service level agreements.
16. Automated collection of data on user satisfaction.
17. Support for creating a quality of service plan.
18. Ability to organize support and responsibility for the service.
19. Notifications about violations of service quality parameters, configurable escalation.
20. Ability to set up an automatic process for a specific type of service.
21. Storing information about external and internal service providers.
22. Storing client's information, including contact information and location.

3.1.8.3. Interaction with other modules

Support for service portfolio management by monitoring and reporting on service attributes and service levels published in the service catalog.

Access to information about service level agreements, technological interruptions, no change periods, and accessibility requirements for participants in the change management process.

Maintain links between a specific service level and personal records or configuration unit records.

Ability to initiate actions related to the support of services when the set thresholds are reached, based on data from the event management process and monitoring tools.

3.1.9. EVENT MANAGEMENT

3.1.9.1. Objectives

a. Be aware of events that may lead to incidents in a timely manner.
b. Promptly initiate activities aimed at preventing incidents.
c. Collect event data in the context of infrastructure elements.
d. Assess the impact of events on the quality of service provision
e. Use flexible mechanisms for categorizing events.

3.1.9.2. Required features

1. Storing detailed information about the event (device ID, type of failure, related components, date and time of the event, etc.).
2. Escalation of alerts.
3. Configurable rules and settings to alert individuals or groups.
4. Filtering event alerts depending on the type of event (notification, warning, or failure).
5. Using rules and processes (workflows) that provide automated actions with events, depending on their type.
6. Indication of completed actions and events that are ready to be closed.
7. Prioritization of events based on the configured criteria and rules for assigning priority.
8. Ability to track trends (increase in the number of events in a time interval, etc.).
9. Possibility of consolidated presentation of all events on the service or system.
10. Ability to compare event information obtained from several monitoring systems.
11. Ability to consolidate events by different types of hardware, platforms, monitoring systems, etc.
12. Automatic identification and consolidation of duplicate events.
13. Proactive alerts about the occurrence of certain events in order to prevent incidents, including those responsible for infrastructure elements.

3.1.9.3. Interaction with other modules

Direct interface for transferring data from alerts and notifications to the incident management process to record incidents.

Automatic actions (sending a message, recording an incident, etc.) as a reaction to certain situations.

Ability to compare related events, which allows you to identify problems in a proactive mode.

Automatic creation of links between events and records about configuration units in the CMDB.

Ability to process events and alerts using data on business processes, service level requirements, the presence of similar and multiple events with a configuration unit or service.

3.1.10. TASK AND WORK ORDER MANAGEMENT

3.1.10.1. Objectives

Management of individual works of employees of various departments for use as part of incident management, change management and a number of other processes.

3.1.10.2. Required features

1. Ability to create tasks and work orders within and outside the processes.
2. Planning personal or team-wide activities for employees, departments, and managers.
3. Control of performance discipline.
4. Distribution of tasks between performers and control of their load level.
5. Accounting for actual labor costs.
6. Presetting task/work order templates.

7. Ability to automatically distribute responsibility for tasks/tasks among employees.
8. Automatic or manual creation of tasks/work orders from incidents, service requests, requests for change, problems with recording information about the initiating object.
9. Automatic change of parameters of related objects (requests, incidents, other tasks/work orders, etc.) when completing/not completing tasks/work orders.

3.1.11. FINANCIAL MANAGEMENT

3.1.11.1. Objectives

a. Accounting and control of the financial component of the provision and support of services (the actual total cost of ownership of IT services and IT assets).
b. Budgeting and estimating the cost of new services being developed (estimated total cost of ownership of IT services and IT assets).
c. Formation of an IT budget based on the detailing of cost items in various areas and the accumulated cost base of the components of these items.
d. Integration with the KGC financial system to reconcile planned, actual, and residual costs of IT services and IT assets.

3.1.11.2. Required features

1. Accounting for the cost of tasks /work orders performed on incidents and service requests.
2. Control of the accounting component of the assets used and its use for calculating the final costs of services.
3. Accounting for the CU cost.
4. Accounting for the costs related to the service.

3.2. Non-functional software requirements

3.2.1. Data location

The solution can be On-Premise or Private Cloud and shall be deployed in Data Centers leased or purchased by KGC.

3.2.2. Performance requirements

The system shall be able to process functions and requests from users connected and working on the system simultaneously (100 users). At the same time, the FI/LO (First In/Last Out) principle shall work.

The response time of the server application to the average request of the client application shall not exceed 500 ms. Transactional processing when saving data shall be independent of the process of locking data tables by priority. Simultaneous editing of the same data by different users shall not be allowed.

The client application shall transmit data to the server for each micro-operation in order to avoid processing big data. The request processing timeout shall not exceed 30 seconds.

3.2.3. Requirements for integration with existing systems

Integration with the Cisco Unified Communications Manager telephone system (System version: 12.5.1.11900-146).

Integration with ERP and electronic document management system (including automatic creation of applications from EDMS to ITSM) will be implemented at the next stage of implementation.

3.2.4. System reporting requirements

Basic reports shall be available that track parameters such as:

a. Average resolution time by employee or department
b. Number of applications exceeding the resolution time
c. Total number of closed applications for a certain period of time
d. Number of applications broken down by day\business services\support levels and
e. Generation of IT asset inventory reports for audit
f. Report showing unauthorized additions of IT assets to the infrastructure
g. Preconfigured reports on key SLA performance indicators
h. Creation of dashboards that show data on service and process metrics in real time

Option to upload reports to Excel, PDF without a limit on the number of rows shall be provided.

Ability to create customized reports on any parameters and attributes.

Ability to send reports by email at a specified time.

3.2.5. Technology stack requirements

3.2.5.1. Using web browsers, "thin" clients, mobile applications on iOS, Android

Preference is given to architectural solutions with a "thin client", web browsers, mobile applications on iOS, Android.

To use an application with a "thin client", only standard software shall be installed on the user's workstation, which does not require subsequent maintenance.

Changes are installed only on the server part of the information system, without requiring multiple changes to workstations and mobile applications.

Application stability to heterogeneity of configurations of workstations, mobile applications.

Minimum requirements for the power of workstations, mobile applications.

3.2.5.2. Application software modularity

The system architecture shall be built from the most independent modules, integrated with each other through universal interfaces (APIs) and services that implement functionality and data acceptance/transmission.

Micro service architecture in combination with SOA (Service Oriented Architecture) provides the most complete looseness of the information environment.

3.2.5.3. Key components of the application technology stack

- Programming language
- Development environment
- Database connection technology
- Application testing methodology and tools
- Database Management System (DBMS)
- Operating systems

- Frameworks
- HTTP server
- Version control system

3.2.5.4. Development environment

The following application development environments are recommended:

- (Java) IntelliJ IDEA
- Vim
- Android Studio
- (Python) PyCharm CE, Jupiter Lab
- VS Code
- Visual Studio
- XCode

To develop applications in Java languages, the IntelliJ IDEA development environment can be used. It's a universal integrated application development environment that provides all the tools you need to create professional desktop, corporate, mobile, and web applications on Java platform. Other tools such as VS or Vim can be used depending on your tasks and language.

3.2.5.5. Database connection technology

To connect to the database system from a Java application, you can use the Hibernate/JPA technology and specification, as one of the most recognized methods of working with a database. This specification works for both versions of Java EE and Java SE. It describes a management system for storing java objects in relational database tables in a convenient way.

This specification does not limit the methods of connecting to a database from Java, there are other technologies/drivers for various databases. Therefore, depending on the tasks and the database, other methods can be used.

3.2.5.6. Testing applications

Each application shall be thoroughly tested after development by the SDLC process. There are a lot of software testing methods and tools. The most accepted ones for web applications are the following:

- Selenium,
- JSFUnit,
- JUnit,
- TestNG

3.2.5.7. Requirements for data exchange protocols when integrating information systems

The platform shall support data exchange via APIs (for example, the use of microservices in Java SpringBoot).

The integration technology shall support one of the following data transfer protocols: JSON (RPC), gRPC, XML (RPC), HTTPS, batch import, SQL.

The technology and standards for data exchange shall support RESTful API and SOAP technologies.

3.2.5.8. Frameworks

Depending on the tasks, the following frameworks can be used for development. In addition to those listed, other frameworks can be used, depending on the task, which are designed for web and mobile solutions.

- Spring Framework
- PrimeFaces
- Blade
- Dropwizard
- Google Web Toolkit (GWT)
- JavaServer Faces (JSF)
- JHipster
- Spark Framework
- MyBatis
- Play Framework
- (JavaScript) React.js, Vue.js, Node.js

3.2.5.9. Version control system

Each software developed shall have a repository and version control. Each application shall be blocked from modification before starting development and shall be unlocked after the application is introduced into the production environment according to the SDLC process.

Recommended version control systems:

- GITHUB
- GitLab

3.2.5.10. ETL processes

ETL (Extract, Transfer and Load) — refers to the process of extracting, transferring, and loading. This is a kind of data integration stage, when data coming from different sources is extracted and sent to data warehouses. Data extracted from various resources is first converted to a specific format according to business requirements.

To build an ETL process, various methods and tools can be used, depending on the tasks for reports and data. To integrate and build ETL at the microservices level, the following technologies and standards shall be applied:

- Apache Kafka message broker

- Containerization: Containerized technology from Kubernetes solution, open source technology - Red Hat OpenShift

- DataOps and DevOps support, which provides interaction within teams and departments to deploy changes

- Technology and transfer protocols — HTML, XML, JSON, SAOP, REST, API

The ELT approach can also be applied — first extract data, then load, and then transform. This approach is better suited for processing machine data.

3.2.5.11. SDLC process

All software developments and changes, regardless of whether the change relates to the client or server side, shall go through the entire development lifecycle — SDLC (System Development Life Cycle).

Description of the life cycle stages of the standard SDLC process:

1. Requirement and architecture. Before designing, it is necessary to agree and approve the requirements (functional and non-functional). Based on a non-functional requirement, it is necessary to develop a software architecture (client platform, server platform, data transfer protocol, etc.).

2. During the design phase, it is necessary to design and document all planned components of the application: user interface (UI), software interface (requirements for frontend and backend development), data structure and model (DBMS)

3. Development is the direct development of the application. Before starting development, you need to lock the version of the application in the version control system (VC). During the development process, there may be clarifications and changes to the documents of the previous phases. It is necessary to achieve a minimum of changes and return to previous phases.

4. During the unit testing phase, the developer shall test all affected parts of the software separately. Automated tools can be used.

5. During the autotesting phase, autotesters shall test the entire affected module for functionality and performance.

6. During the acceptance testing phase, it is required to deploy the software in a pre-production environment and conduct user testing according to the requirements for the task.

7. During the phase of deployment to the prod version, all tested development versions shall move to the prod environment. At the same time, it is important to ensure that the acceptance testing, and production deployment versions match 100%. For integration developments, deployments can be automated through DevOps and CI/CD pipeline technology. After deployment to the production environment, all changes shall be saved in the version control system and unblocked for future changes.

3.2.5.12. User Interface

The system shall have a thin client/web browser user interface, an adaptive web interface design for smartphones and tablets. The requirement to have a mobile version of the application on iOS, Android will be determined for each task individually.

The client application shall be accessible via web and mobile applications according to the HTTPS protocol.

3.2.5.13. Interface and data language

The application shall have an option to select the language in which the interface to the end user shall be displayed. English and Russian languages shall be available for the selection. In addition, the system shall support regional settings of Russia and Kyrgyzstan to process dates and numbers.

The user interface shall display the data in Unicode, i.e., regardless of the data language (encoding – ANSI, Latin).

3.2.5.14. Caching and saving data

According to the technology of data storage and processing, the client application shall be able to cache and process data locally. If a server-side failure occurs or the server is unavailable, the client application shall be able to locally complete data processing of the current function and store data until the system is fully operational again.

3.2.5.15. Friendliness and convenience of the user interface

The user interface shall be friendly and convenient for the users. The program shall have hints and pointers to the functional components of the application. Applications designed to process big data shall have automatic data processing functions, such as copying, duplicating, importing/exporting from an Excel file, entering, and approving identical data on multiple lines, etc. If an error or failure occurs, the software shall produce an appropriate information message/notification that is understandable to the end user.

Links/commands to the latest or most recently active applications shall be available on the main page of the client software. A list of documents or transactions in the form of a hyperlink shall be displayed that require the user's attention, when clicked, the user shall get to the necessary application for processing.

The auto-save function shall be available when entering several stages and big data during the operation of an incomplete function. The user shall be able to continue the process from where it was last saved.

3.2.5.16. Directories module

When developing an application, it is necessary to provide a separate module or modes for editing the system directories. The directory editing module/mode shall be able to add new directories, if necessary, without interfering or involving additional development at the code level.

3.2.6. System class by recovery time and availability per year

Fault Tolerance Template Code	Fault Tolerance Template Name	Template description	System class code using this template	Regulated percentage of system availability per year (SLA)	Maximum System Downtime Per Year	*Recovery in case of a local system failure		*Recovery in case of a fault of the main data center		Number of equipment sets required to ensure the declared fault tolerance
						RTO	RPO	RTO	RPO	
RC4	Medium Speed	Systems, unavailability of which affects the inability to generate income in the long term, or significantly affects the efficiency of a large number of the company employees	BO	99.5%	up to 1d 19h 50m	1-12h	1-12h	up to 5 days	up to 24 h	Two sets of servers and one data storage system in the main data center

3.2.7. System Class by Recovery Priority

System Class Code	System Class Name	Classifier's description

BO	Business Operational	Systems, unavailability of which affects the inability to generate income in the long term, or significantly affects the efficiency of a large number of KGC employees. That is, these are systems that support various operations of KGC. The unavailability of these Systems for 1 day does not lead to significant financial losses.

3.2.8. Typical architectural template for Medium Speed (RC4) systems

Fault Tolerance Template Code	Fault Tolerance Template Name	Template description	Recovery in case of a local failure of the System		Recovery in case of a fault of the main Data Center	
			RTO	RPO	RTO	RPO
RC4	Medium Speed	Systems, unavailability of which affects the inability to generate income in the long term, or significantly affects the efficiency of a large number of the company employees	1-12h	1-12h	Up to 5 days	Up to 24h

RC4 systems are **BO (Business Operational)** Systems in terms of recovery priority, and **HA (High Availability)** in terms of the failure processing type.

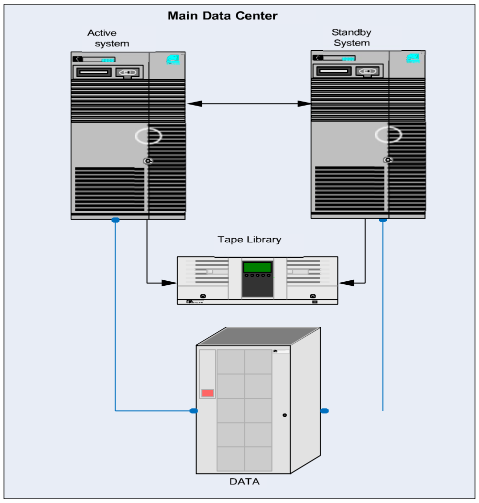
Technological solution for RC4 IT Systems:

To protect data from loss and logical distortion, a DB strategy and recovery from magnetic tapes and/or disks will be applied. At the same time, it will be necessary to perform a full DB on a magnetic tape every week and carry out an incremental DB every day (and not just archive backup copies of the log) on a magnetic tape.

The following data backup technologies can be used:

1) DB over LAN or SAN network;
2) DB to disk memory;
3) DB to magnetic tapes.

Recovery Scheme for RC4 (Medium Speed) Systems



For RC4 class Systems, the following requirements shall be taken into account:

1) a full DB shall be performed at least once a week, and an incremental DB shall be performed at least once a day.

2) DB shall be performed to a local media server on the local network.

3) The backup image shall exist in at least two copies.

4) it is necessary to test the System data recovery according to the approved and signed DB testing plan.

5) allocation of one network connection used for DB purposes for application servers >2 Tb (for physical servers).

6) clustering technology can be used.

7) level 2-3 disk drives connected via SAN can be used (internal hard drives are more likely).

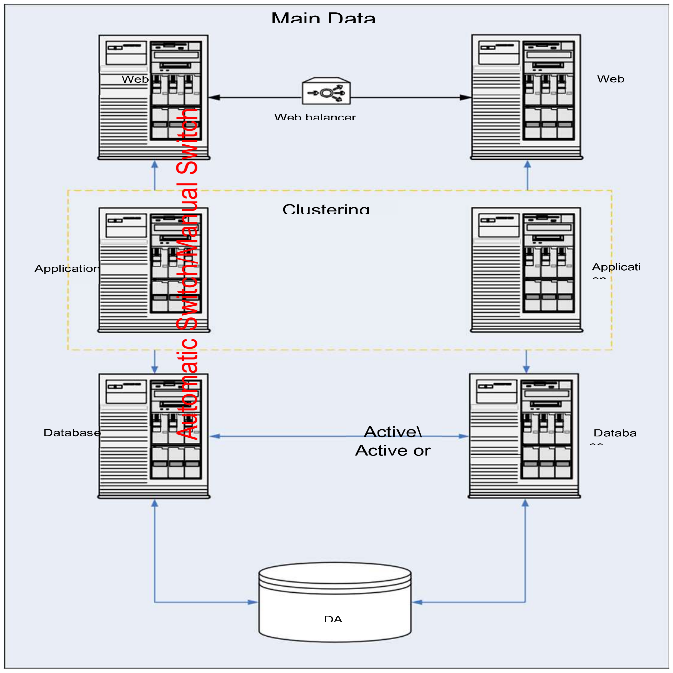
8) server and network equipment are located within the same data center.

9) mandatory use of the Web balancer in the HA scheme.

10) a valid contract (technical support) is required for software and hardware maintenance by the software/equipment Vendor (response time is 24 hours or less).

Redundancy within the main data center:

High Availability



3.2.9. System class by support mode

Code	Name	Descriptio n
S11x7	S11x7	IT system supported by IT in the mode of 11 hours a day and 7 days a week.

3.2.10. System documentation requirements

Based on the results of the project implementation, the Contractor shall develop, agree, and submit the following documents to the Client:

- Technical specifications (functional and non-functional requirements with a description of services for integration with other systems).
- Specification (Program composition and description). Information on the logical structure and functioning of the program. Application description: Information about the purpose of the program, scope of application, methods used, class of tasks to be solved, restrictions for application, Solution architecture (logical structure of the application, broken down into modules; functional architecture; database structure and scheme; application integration scenarios; system deployment scheme in a fault–tolerant architecture, broken down by environments - development, test, preprod, prod).
- Sizing on the system hardware, broken down by environments (development, test, preprod, prod).
- Test program and methodology (test object; test purpose; program requirements; requirements for program documentation; composition and procedure of tests with indication of technical and software tools used during tests, as well as the test procedure; test methods with indication of test results (lists of test examples)).
- Testing protocols (unit, integration, performance, stress tests, vulnerabilities).
- Developer's Guide (Information for checking, maintaining, and configuring the program, API library of classes and functions with descriptions of signatures, semantics of functions).
- System Administration Requirements (Installation, Version Upgrades, etc.)
- Application administrator's guide.
- User manual.

3.2.11. Requirements for the role model of the system

During the implementation of the project, the CRUD matrix (Create, Read, Update, Delete) shall be implemented in the system. When implementing the system, roles, actions, and access will be reviewed.

Action Role	Creating directories, configuring business processes	Approval	Implementation	Control	Reporting
Supervisor		RU	R	R	R
Contractor			CRU		CRU
Administrator	CRU				
Information security officer		CRU			

3.2.12. Information security requirements

3.2.12.1. Identification and authentication

Identify and authentication of access subjects and access objects through integration with Active Directory.

Managing identifiers, including the creation, assignment, and destruction of identifiers.

Managing authentication tools, including storing, issuing, initializing, blocking authentication tools, and taking measures in the event of loss and/or compromise of authentication tools.

Protection of feedback when entering authentication information (the process of exchanging and confirming authentication codes, confirmation of authorization by the reverse side) for external counterparties.

Identification and authentication of file system objects, launchable and executable modules, objects of database management systems, objects created by application and special software, other access objects.

3.2.12.2. Access control of access subjects to access objects

Management (establishment, activation, blocking and destruction) of user accounts, including external users.

Implementation of necessary methods (discretionary, mandatory, role-based, or other method), types (read, write, execute or other type) and access control rules.

Management (filtering (data set restriction), routing, connection control, unidirectional transmission, and other management methods) of information flows between devices, segments of an information system, as well as between information systems.

Separation of powers (roles) of users, administrators and persons ensuring the functioning of the information system.

Assigning the minimum necessary rights and privileges to users, administrators and persons ensuring the functioning of the information system.

Restriction of unsuccessful attempts to log in to the information system (access to the information system).

Warning the user when he/she enters the information system that information protection measures have been implemented in the information system, and about the need for him/her to comply with the information processing rules established by the operator.

Limiting the number of concurrent access sessions for each information system user account.

Blocking the session of access to the information system after the set time of inactivity of the user or at his/her request.

Support and preservation of security attributes (security tags) associated with information during its storage and processing.

Implementing secure remote access of access subjects to access objects through external information and telecommunication networks.

Regulation and control of the use of wireless access technologies in the information system.

Managing interaction with information systems of third-party organizations (external information systems).

Ensuring the trusted loading of computer equipment.

3.2.12.3. Restricting the software environment

Managing the launch (accesses) of software components, including the definition of the components to be launched, setting the parameters for launching components, monitoring the launch of software components.

Managing the installation of software components, including determining the components to be installed, configuring the installation parameters of components, monitoring the installation of software components.

Installation of only authorized software and (or) its components.

Managing temporary files, including banning, allowing, redirecting records, deleting temporary files.

3.2.12.4. Protecting machine media

Accounting for machine media.

Managing access to machine media.

Control of the movement of machine media outside the controlled area (if necessary).

Exclusion of the possibility of unauthorized familiarization with the content of information stored on machine media, and (or) the use of media in other information systems.

Control of the use of interfaces for the input (output) of information on machine media. Control of input (output) of information to machine media.

Control of connection of machine media.

Destruction (erasure) of information on machine media when they are transferred between users, to third-party organizations for repair or disposal, as well as control of destruction (erasure).

3.2.12.5. Recording security events

Defining security events to be recorded and their storage periods.

Determining the composition and content of information about security events to be recorded.

Collecting, recording, and storing information about security events for a set storage time.

Responding to failures when recording security events, including hardware and software errors, failures in information collection mechanisms and reaching the limit or overflow of the memory volume (capacity).

Monitoring (viewing, analyzing) the results of recording security events and responding to them.

Generation of timestamps and (or) synchronization of system time in the information system.

Protection of information about security events.

Providing the ability to view and analyze information about the actions of individual users in the information system.

3.2.12.6. Antivirus protection

Implementation of anti-virus protection or integration with the existing protection systems.

3.2.12.7. Intrusion detection

When intrusions are detected (Dos attacks), signaling and blocking actions.

Updating the base of decisive rules (if there is an upper level of detection, optional).

3.2.12.8. Control (analysis) of information security

Identification, analysis of information system vulnerabilities and prompt elimination of newly identified vulnerabilities.

Monitoring the installation of software updates, including software updates of information security tools.

Monitoring the operability, configuration parameters and correct functioning of software and information security tools.

Control of the composition of technical means, software, and information security tools.

Control of the rules for generating and changing user passwords, creating, and deleting user accounts, implementing access control rules, the rules for separating user authorities in the information system.

3.2.12.9. Ensuring the integrity of information system and information

Software integrity control, including information security software.

Control of the integrity of the information contained in the databases of the information system.

Ensuring the ability to restore software, including software for information security tools, in case of emergency situations.

Detection and response to unsolicited electronic messages (letters, documents) and other information not related to the functioning of the information system (spam protection) entering the information system.

Control of the content of information transmitted from the information system (container, based on the properties of the access object, and content, based on the search for prohibited information using signatures, masks and other methods), and the exclusion of illegal transmission of information from the information system.

Restricting the rights of users to enter information into the information system.

Control of accuracy, completeness and correctness of data entered into the information system.

Control of erroneous user actions when entering and/or transmitting information and warning users of erroneous actions.

3.2.12.10. Ensuring information availability

Use of fault-tolerant technical tools.

Reservation of technical tools, software, information transmission channels, tools ensuring the functioning of the information system.

Monitoring of the trouble-free functioning of technical tools, detection, and localization of functional failures, taking measures to restore failed tools and test them.

Periodic backup of information to backup machine media.

Ensuring the possibility to restore information from backup machine data carriers (backups) within a specified time interval.

Clustering the information system and (or) its segments.

Monitoring the status and quality of the provision of computing resources (capacities) by an authorized person, including the transmission of information.

3.2.12.11. Protecting virtualization environment

Identification and authentication of access subjects and access objects in the virtual infrastructure, including administrators of virtualization management tools.

Access control of access subjects to access objects in the virtual infrastructure, including inside virtual machines.

Recording security events in the virtual infrastructure.

Managing (filtering, routing, connection control, unidirectional transmission) the information flows between components of the virtual infrastructure, as well as along the perimeter of the virtual infrastructure.

Trusted loading of virtualization servers, virtual machines (containers), virtualization management servers.

Managing the movement of virtual machines (containers) and data processed on them.

Monitoring the integrity of the virtual infrastructure and its configurations.

Data backup, backup of hardware, software of the virtual infrastructure, as well as communication channels within the virtual infrastructure.

Implementation and management of antivirus protection in the virtual infrastructure.

Dividing the virtual infrastructure into segments (segmentation of the virtual infrastructure) for processing information by an individual user and/or group of users.

3.2.12.12. Protection of technical equipment

Protecting information processed by technical means from leakage through technical channels.

Organization of a controlled zone, within which stationary technical means that process information, and information protection means, as well as means of ensuring functioning, are permanently located.

Control and management of physical access to technical means, information security means, means of ensuring the functioning, as well as to the premises and structures in which they are installed, excluding unauthorized physical access to information processing means, information security means and means of ensuring the functioning of the information system and premises and structures, in which they are installed.

Placement of information output (display) devices, excluding its unauthorized viewing.

Protection against external influences (environmental influences, instability of power supply, air conditioning and other external factors).

3.2.12.13. Protection of the information system, its means, communication, and data transmission systems

Separation in the information system of functions for managing (administrating) the information system, managing (administrating) the information security system, information processing functions and other information system functions.

Preventing high-priority processes from being delayed or interrupted by low-priority processes.

Ensuring the protection of information from disclosure, modification, and imposition (entering false information) during its transmission (preparation for transmission) via communication channels that go beyond the controlled area, including wireless communication channels.

Providing a trusted channel, route between the administrator, the user and information security means (security functions of information security tools).

Prohibition of unauthorized remote activation of video cameras, microphones and other peripheral devices that can be activated remotely and notifying users about the activation of such devices.

Transfer and control of the integrity of security attributes (security tags) associated with information when exchanging information with other information systems.

Control of authorized use and prevention of unauthorized use of mobile code technologies, including recording events related to the use of mobile code technologies, their analysis and responding to violations related to the use of mobile code technologies.

Control of authorized use and prevention of unauthorized use of speech transmission technologies, including recording events related to the use of speech transmission technologies, their analysis and responding to violations related to the use of speech transmission technologies.

Control of authorized transmission and prevention of unauthorized transmission of video information, including recording events related to the transmission of video information, their analysis and responding to violations, related to the transmission of video information.

Confirmation of the origin of the information obtained in the process of determining network addresses by network names or determining network names by network addresses.

Ensuring the authenticity of network connections (interaction sessions), including to protect against spoofing of network devices and services.

Excluding the possibility of a user denying the fact of sending information to another user.

Excluding the possibility of a user denying the fact of receiving information from another user.

Use of terminal access devices for processing information.

Protection of archived files, settings of information security tools and software, and other data that cannot be changed during information processing.

Identification, analysis and blocking of hidden information transmission channels in the information system, bypassing implemented information protection measures or inside permitted network protocols.

Dividing the information system into segments (segmentation of the information system) and ensuring the protection of the perimeters of the segments of the information system.

Ensuring the loading and execution of software from machine-based read-only media and monitoring the integrity of this software.

Isolation of processes (program execution) in the allocated memory area.

Protection of wireless connections used in the information system.

Exclusion of user access to the information resulting from the actions of the previous user through registries, RAM, external storage devices and other information system resources shared by users.

Protection of the information system from information security threats aimed at denying service to the information system.

Protection of the perimeter (physical and (or) logical boundaries) of an information system in its interaction with other information systems and information and telecommunication networks.

Termination of network connections upon their completion or upon expiration of the time interval of inactivity of the network connection specified by the operator.

Use of various types of system-wide, applied, and special software in the information system or its segments (creating a heterogeneous environment).

Use of applied and special software that can function in environments of various operating systems.

Creation (emulation) of false information systems or their components designed to detect, record, and analyze the actions of violators in the process of implementing information security threats.

Reproduction of false and (or) concealment of true individual information technologies and (or) structural and functional characteristics of the information system or its segments, ensuring the imposition of a false idea of true information technologies and (or) structural and functional characteristics of the information system on the violator.

Transfer of the information system or its devices (components) to a predetermined configuration that provides information protection in the event of failures (faults) in the information protection system of the information system.

Protection of mobile technical means used in the information system.