

# Forbis Banking solutions

## Quality Assurance & Change Management



## Table of Content

1.	Quality Policy .....	3
2.	Forbis Quality Assurance System .....	3
2.1	Goal of the Quality Management System .....	3
2.2	Scope of the Quality Management System .....	3
2.3	Requirements for Products and Services .....	3
3.	Quality Objectives .....	4
4.	Quality Assurance Processes.....	4
4.1	General Development Process .....	5
4.2	Internal Quality Assurance Process .....	12
4.2.1	Testing Levels .....	12
4.2.2	Testing Types.....	12
4.2.3	Internal Quality Assurance (QA) Process .....	15
4.3	Acceptance Testing.....	18
4.3.1	Customer Acceptance of Forbis Software.....	18
4.3.2	Acceptance Testing Process .....	19
4.3.3	Entry Criteria for User Acceptance Testing (UAT).....	19
4.3.4	Exit Criteria for User Acceptance Testing (UAT).....	20
4.4	Fixing Defects Identified in the Customer’s Environment (Support Projects).....	21
5.	Test Environment and Software Deployment .....	24
5.1	General Overview of Test Environments.....	24
5.1.1	Terms and Abbreviations .....	24
5.1.2	General Software Deployment Principles .....	25
5.1.3	Configuration of Environments on the Customer’s Premises.....	26
5.1.4	Environment Setup Process .....	27
5.1.5	Deployment Process for New Functionality and Non-Critical Fixes.....	28
5.1.6	Hotfix Deployment Process.....	36
6.	Change Management.....	39



## 1. Quality Policy

Quality Policy – the top management’s commitment to ensure and continuously improve the quality of information system development, deployment, support, and related activities. Based on this policy, top management establishes the quality objectives.

All functional departments of the company contribute to the improvement of quality by participating in quality management system processes and by striving to satisfy the needs of customers and other stakeholders. Objectives and tasks are established taking into account quality, information security, and service management policies, the results of management reviews, market needs, suggestions and requests from stakeholders, significant environmental aspects, company commitments, statutory and regulatory requirements, financial and technical capabilities, improvement opportunities, and employee suggestions.

Forbis publishes the policy approved by top management on the “Quality Policy” Wiki page.

## 2. Forbis Quality Assurance System

### 2.1 Goal of the Quality Management System

The goal of the quality management system is to ensure that the expectations of stakeholders regarding the company’s IT products and services are met, while quality management processes are continuously analysed and improved. The quality management system aims to ensure the controlled and consistent delivery of high-quality products and services to the company’s customers, as well as to improve and optimize the company’s IT processes, and to increase their efficiency.

To improve the quality of services and products provided, the company implements structured quality management processes based on the ISO 9001:2015 standard.

### 2.2 Scope of the Quality Management System

The scope of Forbis quality management system covers the development, implementation, and support of information systems, as well as the provision of related training and consulting services for financial institutions within the European Union and beyond.

### 2.3 Requirements for Products and Services

- Services and products must be delivered accurately and on schedule (with the ability to demonstrate compliance with requirements, precise accounting, and timely delivery).
- Services and products must meet customer requirements, agreed service levels, and other contractual obligations.
- Services and products must comply with established industry standards for similar offerings and contribute to the stability and security of the financial sector:

- OWASP ASVS (Level 2)
  - International standards ISO/IEC 27001:2022, ISO/IEC 20000:2018, ISO/IEC 9001:2015
- Services and products must comply with local legislation applicable to the provided services, products, and data protection, including (but not limited to):
  - Description of Requirements for Information and Communication Technology and Security Risk Management, approved by the Board of the Bank of Lithuania;
  - Guidelines on Data Protection by Design and by Default in the Information System Life Cycle, issued by the State Data Protection Inspectorate;
  - GDPR.
- Services and products must be continuously improved to enhance customer satisfaction and create long-term value.
- Service and quality management processes must ensure an efficient and productive value chain for all stakeholders.

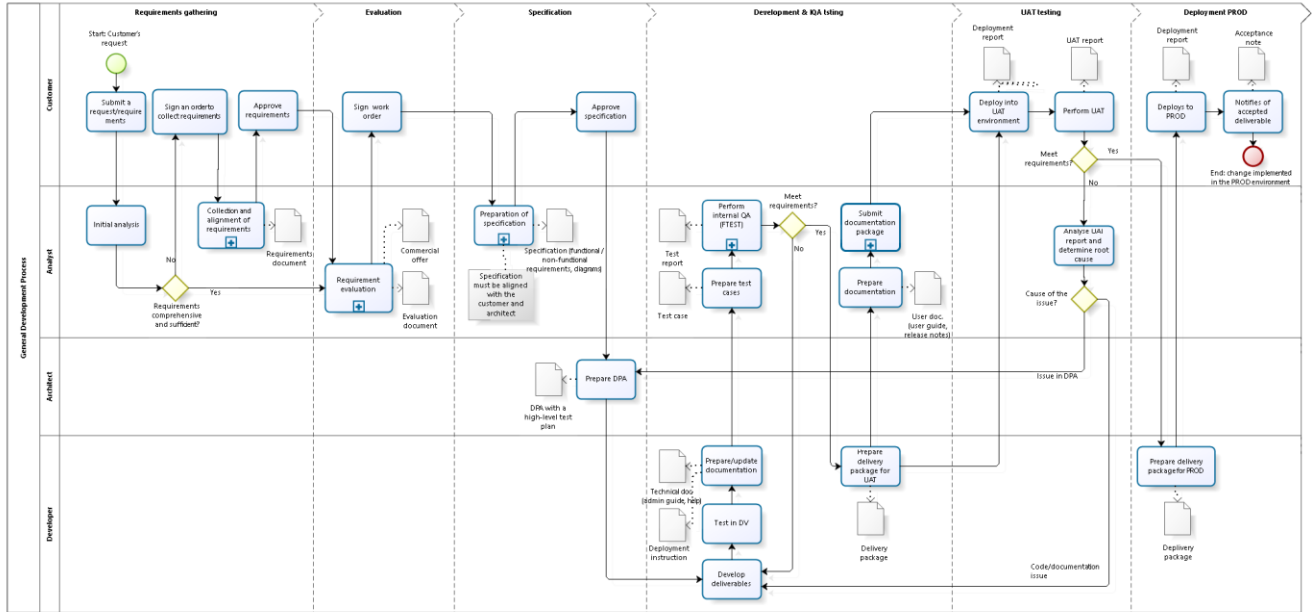
### 3. Quality Objectives

- Strive to understand customer expectations as thoroughly as possible and strictly adhere to their requirements when implementing new technologies.
- Continuously monitor and evaluate the quality planning and implementation processes to improve efficiency and effectiveness, aiming to prevent errors or effectively mitigate their consequences should they occur.
- Select service providers that best meet quality and safety requirements.
- Train competent staff who help achieve the objectives of the quality management system.
- Research global technological trends in the field and integrate the latest solutions into developed products.

### 4. Quality Assurance Processes

- General development process
- Internal quality assurance process (development projects)
- Acceptance testing (development projects)
- Correction of defects identified in the customer environment (support projects)
- Test environment and software deployment
- Process for deploying new functionality and non-critical defect fixes
- Critical fix deployment process

## 4.1 General Development Process



Powered by  
bizagi  
FLOWER

	Responsible person	Step	Input	Description	Output
1	Customer	Start: Customer's request		Process-initiating event – customer submits a request or requirements.	Request for change
2	Analyst	Initial analysis	Request for change	<p>Initial analysis of the submitted request or requirements is performed.</p> <p>If the requirements are comprehensive and clear, the process proceeds to the requirement evaluation step.</p> <p>If additional resources are needed to clarify customer needs and formulate requirements, the customer is offered a separate agreement (order) for collection of requirements.</p>	Proposal for requirements gathering
3	Customer	Sign an order to collect requirements	Proposal for requirements gathering	<p>The customer decides on the proposal for requirements gathering.</p> <p>If the proposal is rejected, the customer must independently formulate comprehensive</p>	Accepted proposal

	Responsible person	Step	Input	Description	Output
				requirements and clarify the request or resubmit it.	
4	Analyst	Collection and alignment of requirements	Accepted proposal	Requirements are collected, aligned, and documented in collaboration with the customer.  Note: upon agreement with the customer, a detailed specification of functional and non-functional requirements may be prepared and approved at this stage.	Requirements Specification
5	Customer	Approve requirements	Requirements document / Specification	The customer aligns and approves the requirements document.	Approved Requirements Specification
6	Analyst (Architect, Developer, Work Order Manager, PM)	Requirements evaluation	Approved requirements document	Evaluation of the costs for a solution meeting the requirements is performed, based on which Forbis prepares a commercial offer for the work order.	Evaluation document Commercial offer
7	Customer	Sign Work Order	Commercial offer	Work order and its execution schedule are approved.	Approved order
8	Analyst (Architect, Customer)	Preparation of specification	Requirements document Approved work order	Forbis specialists prepare the functional specification and align with the customer on the terms for meeting non-functional requirements, acceptance criteria, and/or the testing plan provided by the customer.	Functional and Non-Functional Requirements Specification Acceptance criteria / Test plan
9	Customer	Approve specification	Functional and Non-Functional	Specification (functional and non-functional requirements, acceptance criteria) is approved.	Approved Requirement

	Responsible person	Step	Input	Description	Output
			Requirements Specification		Requirements Specification
10	Architect	Prepare DPA	Approved Requirements Specification Acceptance criteria	Based on the approved requirements, a detailed solution design report (DPA) is prepared or updated, and a high-level testing plan (an internal document) is created, if applicable.	Design Report (DPA) Test plan
11	Developer (Architect / Senior Developer (SC))	Develop deliverables	Functional and Non-Functional Requirements Specification Design Report (DPA)	Based on the prepared documents (requirement document, functional specification, non-functional requirements, and design report), development work is performed, and deliverables are developed or identified defects are fixed.	Development deliverables (software components)
12	Developer	Test in DV	Development deliverables (software components)	Initial testing is performed in the production environment.	Corrected development deliverables (software components)
13	Developer	Prepare/update documentation	Development deliverables (software components)	Technical documentation and deployment instructions for the developed solution (development deliverables) are prepared.	Admin guide Help files Deployment instructions

	Responsible person	Step	Input	Description	Output
14	Analyst	Prepare test cases	Functional and Non-Functional Requirements Specification Development deliverables	Test cases for testing the development deliverables are either prepared or selected from the existing ones.	List of test cases / Test plan
15	Analyst (manual testing) (Automation tester, Developer, Architect, SAD)	Perform internal QA	Development deliverables (software components) Acceptance criteria / List of test cases / Test plan	<p>Developed deliverables are subject to an internal quality assurance process, which verifies whether they meet the approved specification and acceptance criteria:</p> <ul style="list-style-type: none"> <li>▪ if defects are detected during verification, the software development deliverables are returned to the development stage for rectification (returning to the step “Develop deliverables”);</li> <li>▪ if the internal quality assurance process for the development deliverables is successful, a delivery package is prepared.</li> </ul>	Decision on compliance with acceptance criteria (successful or unsuccessful UAT report)
16	Developer (Patch manager, PM)	Prepare delivery package for UAT	Successful testing report	If the internal quality assurance process for the development deliverables is successful, a delivery package for UAT is prepared according to the terms and conditions pre-agreed in the work order. A change transfer task is created, and the PM informs the customer about the deliverables being submitted as needed.	Delivery package (patch)
17	Analyst	Prepare documentation	Successful testing report	<p>User manual is prepared.</p> <p>If significant comments regarding the development deliverables are received during the UAT stage that affect the prepared</p>	User guide

	Responsible person	Step	Input	Description	Output
				document, the user manual will be updated accordingly.	
18	Analyst	Submit documentation package	Admin guide Help files Deployment guide User guide	Prepared documentation is submitted to the customer or published.  Note: due to defects identified during UAT, the documentation may be revised, corrected, or updated.	Submitted documentation package
19	Customer	Deploy into UAT environment	Delivery package (patch)	The customer deploys the submitted patch in their UAT environment.	Deployment report for development deliverables (software components)
20	Customer	Perform UAT	Deployed development deliverables (software components) Acceptance criteria  Customer-prepared UAT test cases	The customer performs user acceptance testing (UAT): <ul style="list-style-type: none"> <li>if defects are detected during the quality assurance verification process, the customer must return the task to Forbis immediately, but no later than the end of the UAT period, by logging all defects in the Service Desk system at the location specified by Forbis. A separate entry must be registered for each defect, including a description of the defect; the customer must provide a detailed description of the test case, specifying the acceptance criteria that were not met and describing the defects identified during testing;</li> <li>if it is determined during acceptance testing that the development deliverables meet the approved acceptance criteria, the request shall be considered fulfilled.</li> </ul>	Decision on compliance with acceptance criteria (successful or unsuccessful UAT report)  Failed test case description

	Responsible person	Step	Input	Description	Output
				<p>If the acceptance testing is reported as failed before the end of the defined testing period, the customer has the right to:</p> <ul style="list-style-type: none"> <li>accept or reject all or part of the work performed by Forbis;</li> <li>allow or disallow Forbis to continue the work in full or only in part.</li> </ul>	
21	Analyst (Architect, Developer)	Analyse UAT report and determine root cause	Unsuccessful UAT report Failed test case description	<p>Logged defects are reviewed and their root cause is determined:</p> <ul style="list-style-type: none"> <li>if defects are found in the source code or documentation, the development deliverables are returned to production (step "Develop deliverables");</li> <li>if defects are related to the solution architecture, the deliverables are returned to the solution architects (step "Prepare DPA").</li> </ul>	Root cause is determined
22	Developer (Patch manager, PM)	Prepare delivery package for PROD	Positive UAT report	<p>If user acceptance testing (UAT) is successful and the customer confirms compliance with the acceptance criteria, a delivery package for the production environment is prepared according to the terms and conditions pre-agreed with the customer. A change transfer task is created, and the PM informs the customer about the deliverables being submitted as needed.</p>	Delivery package (patch)
23	Customer	Deploy to prod	Delivery package	<p>The customer deploys the submitted patch in the production environment in accordance with internal policies and procedures.</p>	Deployment report for developed software components

	Responsible person	Step	Input	Description	Output
24	Customer	Notify accepted deliverable	Positive UAT report Deployment report for development deliverables (software components)	<p>If it is determined during acceptance testing that the development deliverables meet the approved acceptance criteria, Forbis is informed that the request has been fulfilled.</p> <p><b>Change delivery and acceptance acts</b></p> <p>Upon completion of the UAT period, Forbis must submit the delivery and acceptance act to the customer for approval. The customer must sign the delivery and acceptance act within 5 (five) working days of its receipt.</p> <p>If the customer refuses to sign the delivery and acceptance act, they must provide a reasoned response within 5 (five) working days of the act's receipt. In such a case, Forbis may suspend work related to the further execution of the Order until the causes of the disagreement are clarified and resolved. If no reasoned response is provided, the work shall be deemed successfully accepted</p> <p>Support services for functionality operating in the customer's production database are provided only after the delivery and acceptance act has been signed. Unless otherwise agreed in a specific order, the delivered software development deliverables are considered acceptable if no more than 3 low-level defects are detected in them.</p>	Signed Acceptance Act
25	PM Project team: Analyst / Developer / Architect	End: Closed request	Signed Acceptance Act	<p>End of the process.</p> <p>Jira project is closed. The work order is moved to the archive. For quality improvement purposes, the customer is requested to fill out a project performance questionnaire. A project retrospective workshop/discussion is organized with the project team and Forbis management as needed.</p>	Project closure activities

## 4.2 Internal Quality Assurance Process

Testing types and levels vary depending on the nature and scope of the system changes.

### 4.2.1 Testing Levels

Level	Description	Test case	Acceptance criteria
Unit testing	Unit testing is performed to ensure that an individual program unit operates according to the specification.	Upon creating a new unit (functionality) or modifying an existing unit (functionality).	The unit operates according to the specification.
Integration testing	Integration testing is performed to evaluate individual software units that are combined and tested as a group.	Upon modification of an approved unit specification.	Interdependent units function together according to the specifications.
System testing	Software system testing is performed to evaluate the compliance of the entire integrated system with the specified requirements.	<ul style="list-style-type: none"> <li>• Upon making changes to core mechanisms, when the necessity of testing is specified during the change design phase or in the testing plan.</li> <li>• Version release.</li> </ul>	The system operates according to the specifications.

### 4.2.2 Testing Types

Type	Description	Test case	Acceptance criteria
Functional Testing	<p>Functional testing is testing conducted based on functional requirements and specifications.</p> <p>Functional testing steps:</p> <ul style="list-style-type: none"> <li>• Identify the functions that the software must perform.</li> <li>• Create input data based on the functional specifications.</li> <li>• Define the expected results based on the functional specifications.</li> <li>• Execute the test case.</li> <li>• Compare the actual results with the expected results.</li> </ul>	<ul style="list-style-type: none"> <li>• Upon completion of software development.</li> <li>• After fixing identified defects.</li> </ul>	The software meets the functional requirements.

Type	Description	Test case	Acceptance criteria
Smoke Testing	Smoke testing is performed to ensure that the core functions are operational.	<ul style="list-style-type: none"> <li>• Upon significant system changes, when the necessity of testing is specified during the change design phase or in the testing plan.</li> <li>• Version release.</li> </ul>	All core functions are working; no major defects have been identified.
Performance Testing	Performance testing is performed to determine the system's responsiveness and stability under a specific workload.	<ul style="list-style-type: none"> <li>• Subject to additional coordination in the Work Order.</li> <li>• Upon changes to core mechanisms, when the necessity of testing is specified during the change design phase or in the testing plan.</li> <li>• When mass process execution is planned, provided the necessity of testing is specified during the change design phase or in the testing plan.</li> <li>• Version release (tests as required).</li> </ul>	The software deliverables comply with non-functional requirements, such as resilience, performance, efficiency, and response time.
Security Testing	Security testing is performed to identify system vulnerabilities and to ensure that its data and resources are protected from potential fraud.	<ul style="list-style-type: none"> <li>• Subject to additional coordination in the Work Order.</li> <li>• Version release (tests as required).</li> </ul>	No vulnerabilities or opportunities for unfair exploitation of the system were detected.
Regression Testing	Regression testing is performed to ensure that software changes have not negatively impacted existing functionality. During regression testing, no new test cases are created; instead, previously developed test cases are performed again.	<ul style="list-style-type: none"> <li>• Upon significant system changes, when the necessity of testing is specified during the change design phase or in the testing plan.</li> <li>• Version release (tests as required).</li> </ul>	The previously developed functionality remains unaffected.

Type	Description	Test case	Acceptance criteria
Usability Testing	Usability testing is a type of testing used to verify whether the system is convenient and easy to use for the end-user.	Upon creating new user interfaces.	The interface is user-friendly for the end user.

Prior to delivery to the customer, Forbis performs internal testing (“alpha” testing). During internal testing, smoke and functional testing is conducted for each development deliverable submitted to the customer. Other types of testing are specified in the testing plan or the detailed design report (optional for testing each individual development deliverable submitted to the customer).

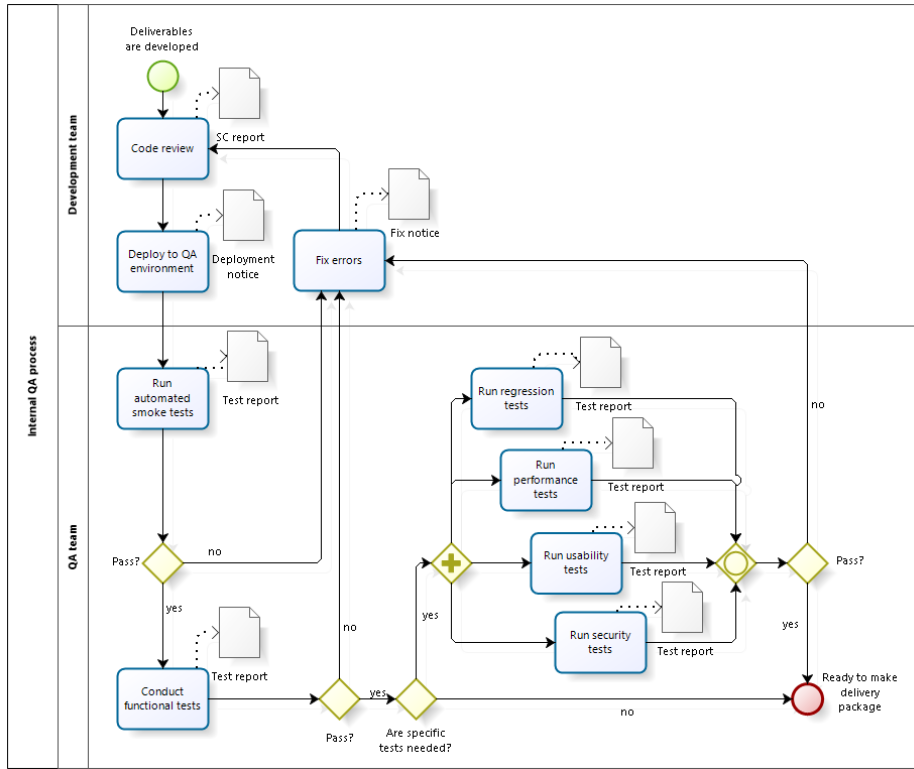
Testing is performed only in Forbis’ internal testing environment using internal testing data, which are not copies of customer data.

Upon delivery to the customer, the customer performs user acceptance testing in their own testing environment as well as in an environment identical to the production environment.

If integration solutions cannot be verified in the internal testing environment, Forbis performs internal testing only to the extent permitted by the technical capabilities provided by third parties. In certain cases, with the consent of both parties, the full functionality may be verified at the customer’s request by deploying it in their testing environment.

Forbis commits to deliver the deliverables ensuring compliance with SLA obligations (at the end of the UAT period, no more than 3 Low priority defects must remain).

### 4.2.3 Internal Quality Assurance (QA) Process



	Step	Input	Description	Output
1	Start: Deliverables developed		Process-initiating event – deliverables have been developed.	Development deliverables (software components)
2	Code review	Development deliverables (software components)	Developed code is reviewed and verified. The task remains with the developer until a positive result is achieved.	Code review report
3	Deployment to QA Environment	Code Review Report Deployment Documentation	Development deliverables are deployed to the testing environment (automated deployment (CI)).	Deployed development deliverables (software components)

	Step	Input	Description	Output
4	Run smoke tests	Deployed development deliverables (software components)	Running smoke tests. The purpose of smoke testing is to verify that the core functions of the system are operating correctly: whether forms open, values can be entered, etc.	Successful or unsuccessful testing report
5	Pass?	Successful or unsuccessful testing report	<p>Verification against the approved acceptance criteria:</p> <ul style="list-style-type: none"> <li>• In case of unsuccessful testing, the development deliverables are returned for correction of the identified defects.</li> <li>• If all approved acceptance criteria are met, the deliverables are forwarded for functional testing.</li> </ul>	Decision to return for rework or proceed to functional testing
6	Conduct functional tests	<p>Deployed development deliverables (software components)</p> <p>Requirements specification</p> <p>Test plan</p>	Upon successful completion of smoke testing, functional testing is conducted.	Successful or unsuccessful testing report
7	Pass?	Successful or unsuccessful testing report	<p>Verification against the approved acceptance criteria:</p> <ul style="list-style-type: none"> <li>• In case of unsuccessful testing, the development deliverables are returned for correction of the identified defects.</li> <li>• If all approved acceptance criteria are met, the deliverables are forwarded for delivery package preparation.</li> </ul>	Decision to return for rework or proceed to the next step
8	Are specific tests needed?	Requirements specification	Based on the nature of the change, it is determined whether specific testing is required.	Decision on the need for additional

	Step	Input	Description	Output
		Design report Test plan		testing or proceeding to package preparation
9	Conduct performance tests	Deployed development deliverables (software components) Test plan	Performance testing is conducted, if applicable.	Successful or unsuccessful testing report
10	Conduct usability tests	Deployed development deliverables (software components) Test plan	Usability testing is conducted, if applicable.	Successful or unsuccessful testing report
11	Conduct security tests	Deployed development deliverables (software components) Test plan	Security testing is conducted, if applicable.	Successful or unsuccessful testing report
12	Conduct regression tests	Deployed development deliverables (software components) Test plan	Regression testing is conducted, if applicable.	Successful or unsuccessful testing report
13	Pass?	Successful or unsuccessful testing report	Verification against the approved acceptance criteria: <ul style="list-style-type: none"> <li>In case of unsuccessful testing, the development deliverables are returned for correction of the identified defects.</li> </ul>	Decision to return for rework or proceed to delivery

	Step	Input	Description	Output
			<ul style="list-style-type: none"> <li>If all approved acceptance criteria are met, the deliverables are forwarded for delivery package preparation.</li> </ul>	package preparation
14	Fix errors	Unsuccessful testing report	If testing is unsuccessful, the development deliverables are returned for rework to resolve the identified defects.	Fixed development deliverable (software components)
15	End: Ready to prepare delivery package	Successful testing report	Process completed – submitted for delivery package preparation.	Notification of readiness to prepare the delivery package

### 4.3 Acceptance Testing

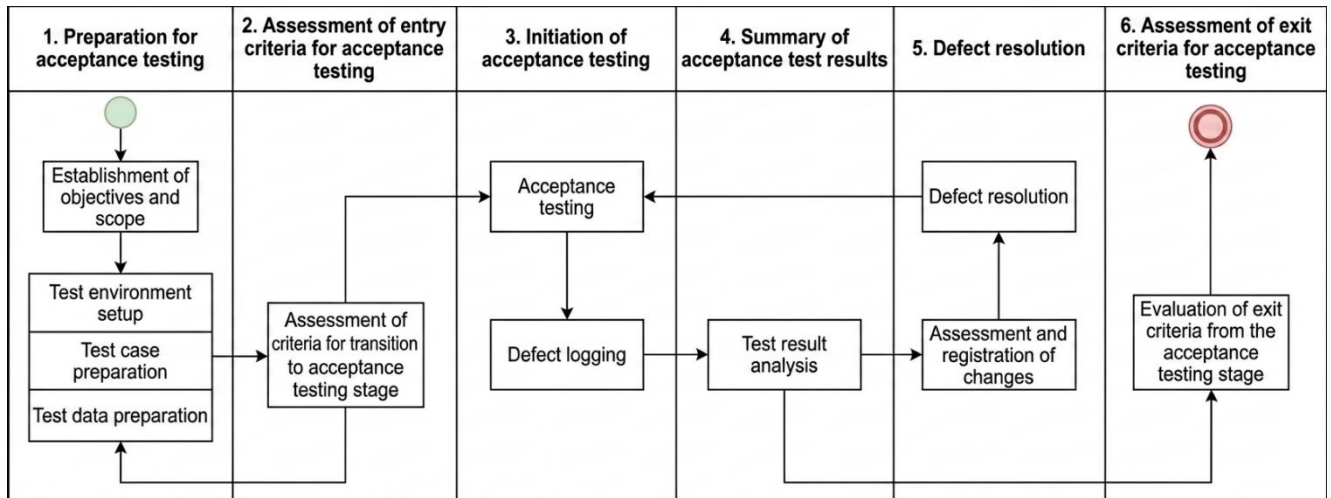
#### 4.3.1 Customer Acceptance of Forbis Software

The following key principles must be followed when accepting software development deliverables:

- The Customer must follow the proper deployment instructions provided by Forbis.
- The software development deliverables must be tested based on the agreed testing plan.
- A User Acceptance Testing (UAT) case for each software deliverable is considered successful if all its steps have been completed successfully and all step results match the expected results.
- After Forbis resolves all defects identified during the initial User Acceptance Testing (UAT), each initially failed test case must be re-executed to confirm that all issues have been successfully resolved.
- All UAT test cases specified in the agreed testing plan must be re-executed until all test cases are deemed successfully completed.
- If any UAT test case must be executed more than two (2) times due to Forbis' fault, it is considered that Forbis has improperly prepared the software deliverable according to the UAT criteria, and the deliverable is considered rejected. In such a case, Forbis undertakes to perform repeated internal testing and to re-prepare the deliverables for UAT. This condition does not apply if the defects were caused by the Forbis environment limitations and such defects could not be detected during testing within the Forbis environment.
- The software development deliverables cannot be submitted as ready for deployment in the customer's PROD environment, unless they have been accepted by the customer.

- Partial delivery of software development deliverables is possible if it can be accepted according to the acceptance criteria. Subsequent delivery procedures will apply only to new subsets. In the case of partial delivery, Forbis must guarantee that the previously delivered part of the deliverables will not be altered or damaged and will meet all requirements.
- The Forbis ServiceDesk system is used to register defects and other issues.

#### 4.3.2 Acceptance Testing Process



#### 4.3.3 Entry Criteria for User Acceptance Testing (UAT)

Before the software development deliverables are accepted by the customer for UAT testing, Forbis must perform internal testing procedures to ensure that the software development deliverables are fully prepared for acceptance. Forbis must ensure that the internal testing of the deliverables covers all the functionality specified in the requirements and meets the requirements applicable to the software development deliverables.

#### Software development deliverables will be considered ready for UAT when:

- High and medium-priority defects have been resolved. A valid and customer-approved plan exists for resolving low-priority defects identified during the internal testing of the software development deliverables.
- All test cases for the software development deliverables specified in the testing plan have been successfully executed.
- High and medium-level defects have been resolved. There is a valid and customer-approved plan for resolving low-level defects identified during the internal testing of the software development deliverables.
- The customer has prepared the testing environment in accordance with Forbis recommendations and rules.
- The customer has prepared and approved the User Acceptance Testing (UAT) plan and test cases.
- The customer has prepared and approved the data required for UAT execution.

- The customer has selected and appropriately prepared the members of the User Acceptance Testing (UAT) group to perform the testing.
- Forbis Alpha Testing Summary Report will be provided upon the customer's request if a defect is identified. The minimum scope of the Alpha Testing Summary Report includes:
  - Test identifier (ID).
  - Test type (specified if non-functional testing of the agreed scope is performed).
  - Test description (list of test cases).
  - Final status of the test result.

#### 4.3.4 Exit Criteria for User Acceptance Testing (UAT)

The acceptance of software development deliverables shall be conducted based on the agreed testing plan and the UAT test cases prepared and approved by the customer.

All identified defects must be registered in the ServiceDesk system in accordance with the established procedure and within the timeframe allocated for User Acceptance Testing (UAT).

The severity level of defects in the software development deliverables is determined based on the potential impact when using the deliverables:

- **Level 1 (High/Critical)** – significant / critical business impact:
  - The software development deliverables cannot be used (e.g., the system is non-functional, core business logic is incorrectly implemented, data loss, improper implementation, severe performance issues, etc.). There is no workaround acceptable to the customer.
  - AND (OR)
  - The functionality/tools of the software development deliverables are non-functional or function incorrectly, and there is a severe impact on efficiency. There is no workaround acceptable to the customer.
- **Level 2 (Medium/Normal)** – moderate business impact:
  - Certain functionality/tools of the software development deliverables function incorrectly; there is a moderate impact on efficiency. The software development deliverables can only be used if a workaround acceptable to the customer is available.
- **Level 3 (Low)** – low business impact:
  - Minor defects that do not impact the use of the software development deliverables (e.g., grammatical errors, documentation errors, etc.).
  - The customer requests information, improvement, or clarification of documentation regarding the software development deliverables; however, the operation of the deliverables remains unaffected and/or a very simple and obvious workaround exists that is acceptable to the customer.

The decision regarding the assigned severity level is made by the customer and specified in the Service Desk system when registering tasks for the rectification of software development deliverables' defects.

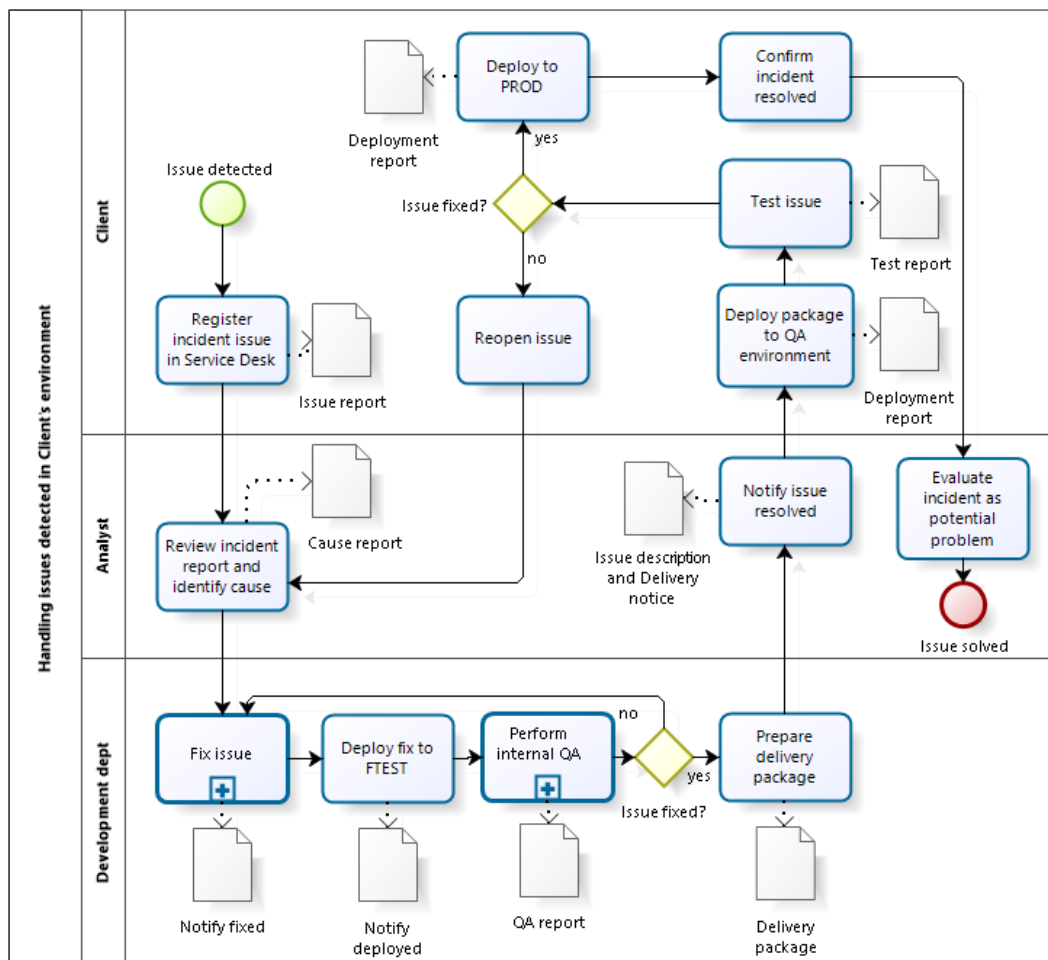
Forbis reserves the right to change the severity level, providing a reasoned explanation, if it does not meet the defined criteria.

**Software development deliverables shall be considered accepted if:**

- All UAT test cases specified in the testing plan have been successfully executed.
- The software development deliverables meet all delivery criteria. The thresholds and rules for defect identification do not apply to defects caused by the limitations of the customer’s environment.
- A valid and customer-approved plan exists for the rectification of defects of three severity levels identified during the UAT of the software development deliverables.
- No defects were registered in the Forbis Service Desk system during the time allocated for User Acceptance Testing (UAT).

**4.4 Fixing Defects Identified in the Customer’s Environment (Support Projects)**

All defects and errors identified by the customer must be registered in the Forbis ServiceDesk system.



	Step	Input	Description	Output
1	Start: Issue detected		Process-initiating event – issue detected.	
2	Register incident in Service Desk system	Issues with system performance identified	Task is registered in the Forbis Service Desk system. When registering the task, it is necessary to specify the environment where the issue was detected, provide a description of the issue, list the steps to reproduce the issue, and indicate the severity level of the defect.	Description of the registered error
3	Review incident report and identify cause	Description of the registered issue	Forbis responsible persons review the issue description and determine the root cause of the incident.	Identified cause of the defect
4	Fix issue	The root cause of the defect has been identified and resolved.	Forbis resolves the issue.	Software components
5	Deploy fix to FTEST	Software components	The fix is deployed to the testing environment.	Fix deployment notification
6	Perform internal QA	Software components	Internal testing of the fixed functionality is performed.	Test report
7	Issue fixed?	Test report	Verification if the issue has been fixed: <ul style="list-style-type: none"> <li>• if the issue is not fixed, it is returned to the developers;</li> <li>• if the issue is fixed, the delivery package is submitted for preparation to the customer.</li> </ul>	Decision on compliance with criteria: positive / negative
8	Prepare delivery package	Positive decision on compliance	If the acceptance criteria are met, the delivery package is prepared and uploaded to the SFTP.	Delivery package

	Step	Input	Description	Output
		with acceptance criteria		
9	Notify delivery	Delivery package	The delivery package is submitted to the customer by marking the delivery in the Service Desk system (status "Resolved").	Notification of delivered fix Change description
10	Deploy package in QA environment	Notification of delivered fix Uploaded delivery package	The customer deploys the package to the test environment.	Deployed software components
11	Test issue	Deployed software components Test plan Acceptance criteria	The customer performs testing of the changed functionality.	Test report
12	Issue fixed?	Test report	Verification if the issue has been fixed: <ul style="list-style-type: none"> <li>if the issue is not fixed, Forbis is informed by returning the task;</li> <li>if the issue is fixed, it is deployed to the PROD environment.</li> </ul>	Decision on compliance with criteria
13	Reopen issue	Decision on non-compliance with criteria	If the issue remains unresolved, Forbis is notified by reopening the task.	Information on non-compliance decision
14	Deploy to PROD	Decision on compliance with criteria	If the issue is fixed, it is closed in the Service Desk system, and the solution is deployed to the PROD environment (status "Done").	Notification of compliance decision

	Step	Input	Description	Output
15	Evaluate issue as potential problem	Notification of compliance decision Notification of a suitable solution	The analyst must evaluate if the incident shows signs of a potential problem and close the task.	Issue assessment report
16	End: issue solved		End of process: issue resolved.	

## 5. Test Environment and Software Deployment

### 5.1 General Overview of Test Environments

This section describes the architectural concept of test environments, including recommendations for organizing test environments on the Customer's premises and within Forbis.

#### 5.1.1 Terms and Abbreviations

Abbreviation	Name	Description
<b>PROD</b>	Production environment	Environment where banking systems operate. Always hosted within the customer's infrastructure.
<b>PRELIVE</b>	Hot fix test environment	Environment where both critical defects and fixes are tested prior to deployment to the production environment. It is a copy of the PROD (production environment), updated as frequently as possible after all changes are deployed to PROD. Always hosted within the customer's infrastructure.
<b>TEST</b>	Test environment	Environment where the functionality of the production version is tested. Always hosted within the customer's infrastructure.
<b>FPRELIVE</b>	Supplier hot fix test environment	Environment where Forbis specialists test critical issue fixes and patches before delivering them to the customer. This is a PRELIVE environment with anonymized or removed data. This database is hosted within the customer's infrastructure; the customer also plans to test critical issues in this environment; however, Forbis is responsible for deploying components to this DB.
<b>FTEST</b>	Supplier test environment	Environment where Forbis experts test new production version functionality before delivering it to the customer. This is the latest

Abbreviation	Name	Description
		available TEST copy. This database is hosted within the customer's infrastructure; however, Forbis is responsible for deploying components.
<b>HOTFIX</b>	Hot fix	Unscheduled software fix intended for the urgent resolution of critical issues in the PROD (production) environment.
<b>Release</b>	Release	Software changes related to the deployment of a new release.
<b>Patch</b>	Patch	Package of software changes intended to implement additional functionality or to fix an issue based on the current release.
<b>Application software</b>	Application software	Forbis system and its components. From a technological perspective, the system consists of database components, Oracle Forms/Reports components, and JAVA components deployed on the application server.
<b>Repository</b>	Program objects repository, PO repository	Database where information about the installed software components is stored.
<b>Acceptance testing</b>	Acceptance testing	Final software testing before starting the deployment to the production environment.

5.1.2 General Software Deployment Principles

Under normal operating conditions, the software runs in a single (Production) environment; therefore, the testing of prepared changes (releases and fixes) must also be performed in a single environment corresponding to a single version. Testing in multiple environments (i.e., those dedicated to different changes) is highly discouraged, as inconsistencies between changes may lead to errors. If the use of separate environments is necessary, it must be coordinated in advance.

Software changes are classified as critical (hotfixes) and non-critical. Hotfixes are implemented outside the regular schedule, while non-critical changes are implemented according to the project plan.

Fixes prepared by Forbis experts are deployed in the FTEST and FPRELIVE environments, where they are tested by Forbis specialists. After successful testing, the components are transferred to the customer's specialists for testing in the TEST and PRELIVE environments, respectively.

All changes delivered to the customer are deployed in the order specified by Forbis experts. This sequence must not be altered.

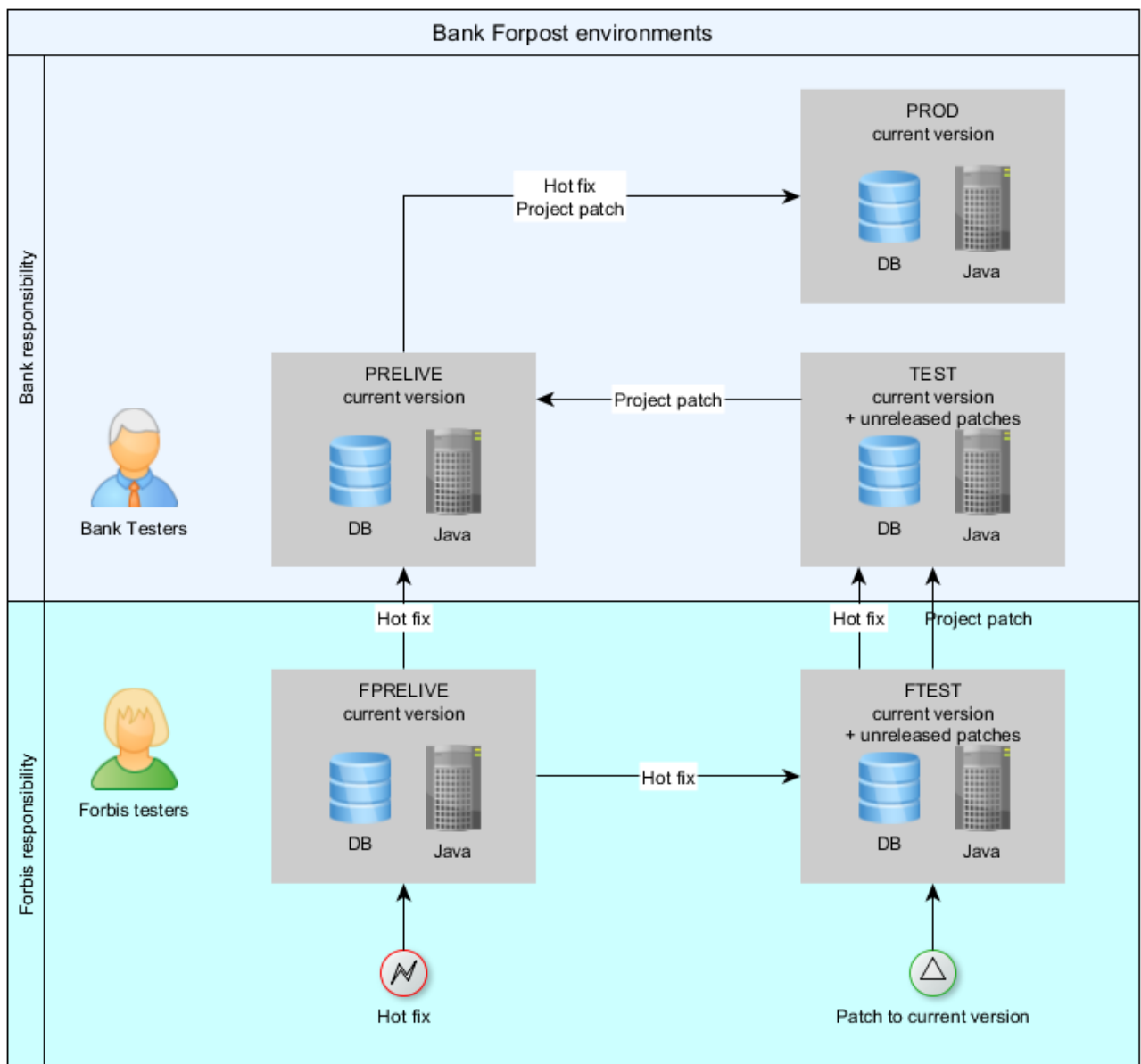
A change tested in the TEST environment must be re-tested in the PRELIVE environment before deployment to the PROD environment. If there are discrepancies between the TEST and PRELIVE environments, the customer must request Forbis to prepare a fix for the PRELIVE environment prior to deployment.

The customer orders the delivery of changes to PRELIVE/PROD via the Service Desk system, Support project.

After deploying the changes to the PROD environment, the customer must inform Forbis of the deployed fixes.

### 5.1.3 Configuration of Environments on the Customer's Premises

The diagram below presents the recommended configuration of environments. The purpose of this configuration is to ensure quality control of the implemented changes.



#### 5.1.4 Environment Setup Process

This section describes the environment setup process, definition of tasks and allocation of responsibilities, as well as the frequency of execution.

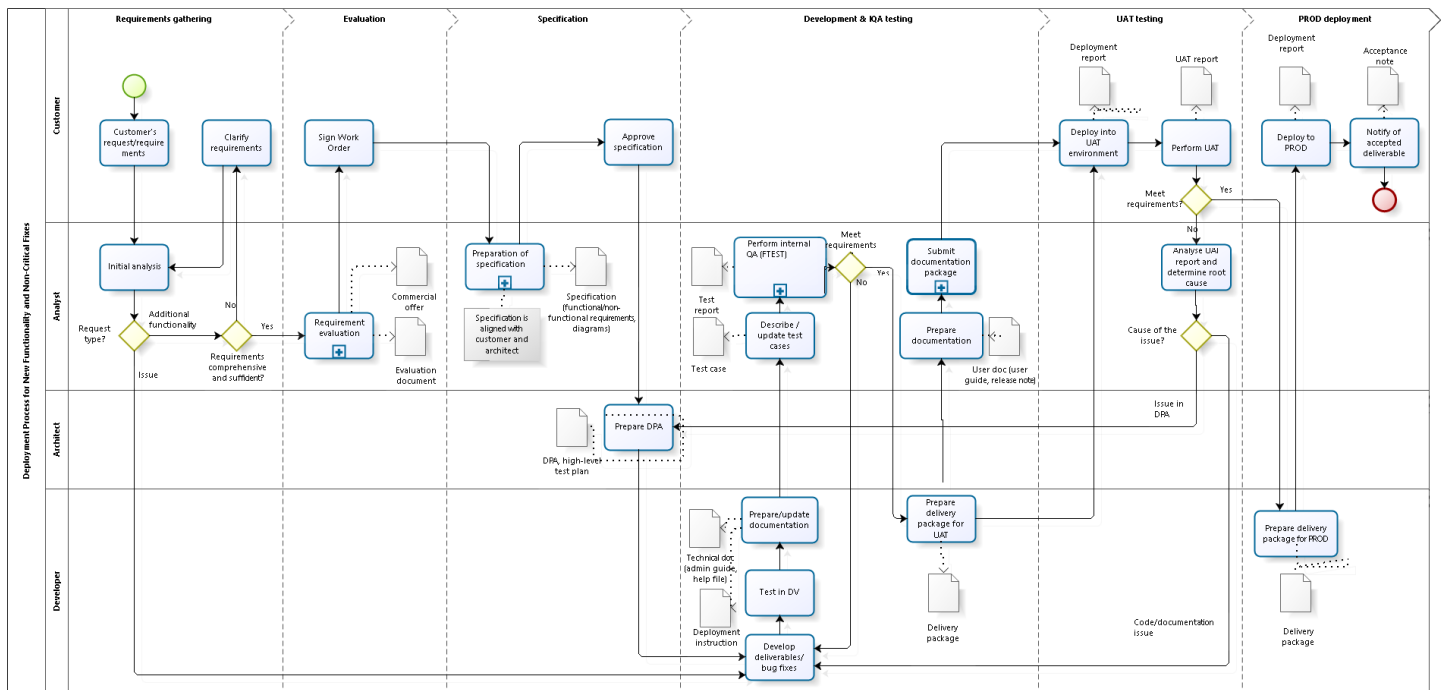
Indicative list of environments and their update frequency:

Environment	Creation	Responsibility	Update frequency	Environment
<b>PROD</b>	New version is released and tested.	Customer	Production environment is configured during the initial deployment of the FORPOST system. Once the fixes prepared by Forbis specialists have been deployed and tested in the TEST and PRELIVE environments, the Customer's employees update the Production environment. The frequency of these updates is not regulated.	Customer
<b>PRELIVE</b>	PROD environment is ready.  FPRELIVE and PRELIVE are updated synchronously at a frequency agreed with the customer.	Customer	Customer updates this environment as frequently as possible after deploying changes to PROD. Daily updates are recommended; in any case, the environment must be updated before testing the changes intended for the PROD environment.	Customer
<b>FPRELIVE</b>	FPRELIVE and PRELIVE are updated synchronously at a frequency agreed with the customer.	Customer	Customer periodically prepares an anonymized copy of the Production environment for Forbis; it is updated immediately after the customer deploys fixes to the PROD environment, but no less than once a week. This ensures that Forbis specialists have access to an environment where the software is identical to the customer's PROD environment, while the data remains different.	Forbis
<b>TEST</b>	PROD environment is ready.  According to the customer's needs, in coordination with Forbis. TEST and	Customer	Customer prepares the testing environment after the deployment of new release. The customer's employees update the testing environment by deploying fixes prepared by Forbis (upon the release of new functionality). The frequency is not regulated (it depends on the customer's needs).	Customer

Environment	Creation	Responsibility	Update frequency	Environment
	FTEST are updated synchronously.			
<b>FTEST</b>	TEST environment is ready.	Customer	Customer prepares the testing environment for Forbis along with the TEST environment. Forbis employees update the environment during the deployment of changes (upon the release of new functionality). The frequency is not regulated (it depends on the customer's needs regarding the support of the current version.).	Forbis

### 5.1.5 Deployment Process for New Functionality and Non-Critical Fixes

This section describes the deployment process for a new functionality / non-critical error fix patch, where fixes are applied to the current release.



	Responsible person	Step	Input	Description	Output
1	Customer	Start: Customer's request		Process-initiating event – customer has submitted a request or requirements.	Request for change
2	Analyst	Initial analysis	Request for change	<p>Initial analysis of the submitted request or requirements is performed, and the reason for request submission is determined.</p> <p>If the bug has been registered, the next step is “Develop deliverables / bug fixes”.</p> <p>If new or additional requirements (features) have been submitted and they are comprehensive and clear, the next step is requirements evaluation.</p> <p>In case of any questions regarding the submitted requirements, the customer is requested to provide answers and clarify the requirements.</p>	<p>Issue description in JIRA INT</p> <p>Correspondence in JIRA EXT</p>
3	Customer	Clarify requirements	Request for change	Customer answers the questions, formulates detailed requirements, and clarifies or resubmits the request.	Aligned requirements
4	Analyst (Architect, Developer, Work Order Manager, PM)	Requirement evaluation	Aligned requirements	Estimation of production costs for a solution meeting the requirements is performed, based on which Forbis prepares a commercial offer for the work order.	Evaluation document Commercial offer
5	Customer	Sign Work Order	Commercial offer	Order and its execution schedule are approved.	Approved order

	Responsible person	Step	Input	Description	Output
6	Analyst (Architect, Customer)	Preparation of specification	Requirement document Approved work order	Forbis specialists prepare/supplement/update the functional specification, coordinate the terms of non-functional requirements fulfilment with the customer, as well as acceptance criteria and/or the testing plan provided by the customer.	Functional and Non-Functional Requirements Specification Acceptance criteria / Test plan
7	Customer	Approve specification	Functional and Non-Functional Requirements Specification	Specification is approved (functional and non-functional requirements, acceptance criteria).	Approved requirements specification
8	Architect	Prepare DP	Approved requirements specification Acceptance criteria	Based on the approved requirements, the Detailed Design Report (DPA) is reviewed. If the additionally ordered functionality changes the solution described in the DPA, DPA is updated and a high-level test plan (internal document) is prepared, if provided.	Design Report (DPA) Test plan
9	Developer (Architect / Senior Developer (SC))	Develop deliverables / bug fixes	Functional and Non-Functional Requirements Specification Design Report (DPA)	Based on the prepared documentation (JIRA INT description, functional specification, non-functional requirements, and design report), development is performed to produce deliverables or resolve identified defects.	Development deliverables (software components)

	Responsible person	Step	Input	Description	Output
10	Developer	Test in DV	Development deliverables (software components)	Initial testing is performed in the production environment.	Fixed development deliverables (software components)
11	Developer	Prepare/update documentation	Development deliverables (software components)	Technical documentation and deployment instructions for the solution (development deliverables) are prepared.	Admin guide Help files Deployment guide
12	Analyst	Prepare test cases	Functional and Non-Functional Requirements Specification Development deliverables	Test cases, to be used for testing development deliverables, are prepared or selected from existing ones.	List of test cases / Test plan
13	Analyst (manual testing) (Automation tester, Developer, Architect, SA D)	Perform internal QA	Development deliverables (software components) Acceptance criteria/ List of test cases / Test plan	Internal quality assurance process is applied to the developed deliverables – it is verified whether the developed deliverables meet the approved specification and acceptance criteria. <ul style="list-style-type: none"><li>If defects are detected during verification, the software development deliverables are returned to the development stage for defect rectification (to the step “Develop deliverables / bug fixes”).</li></ul>	Decision on compliance with acceptance criteria (successful or unsuccessful testing report)

	Responsible person	Step	Input	Description	Output
				<ul style="list-style-type: none"> <li>If the internal quality assurance process for development deliverables was successful, a delivery package is prepared.</li> </ul>	
14	Developer (Patch manager, PM)	Prepare delivery package for UAT	Successful testing report	If the internal quality assurance process for development deliverables was successful, a delivery package for UAT is prepared in accordance with the terms and conditions pre-agreed in the work order. A change submission task is created, and the PM informs the customer about the deliverables being submitted, as needed.	Delivery package (patch)
15	Analyst	Prepare documentation	Successful testing report	<p>User guide is updated.</p> <p>If substantial comments regarding development deliverables are received during the UAT stage that affects the prepared document, the user guide will be adjusted accordingly.</p>	User guide
16	Analyst	Submits the documentation package	Admin guide Help files Deployment guide User guide	<p>The prepared documentation is submitted/published to the customer.</p> <p>Note: due to defects identified during UAT, the documentation may be revised, corrected, and updated.</p>	Delivered document package
17	Customer	Deploy into UAT environment	Delivery package (patch)	Customer deploys the provided patch in their UAT environment.	Report on implemented development deliverables (software components)

	Responsible person	Step	Input	Description	Output
18	Customer	Perform UAT	<p>Deployed development deliverables (software components)</p> <p>Acceptance criteria</p> <p>UAT test cases prepared by the customer</p>	<p>Customer performs User Acceptance Testing (UAT).</p> <ul style="list-style-type: none"> <li>If defects are identified during the quality assurance check, the customer must immediately, but no later than within the time allocated for User Acceptance Testing (UAT), return the task to Forbis by registering all defects in the designated area of the Service Desk system. A separate entry must be registered for each defect, providing a description of the issue. The customer must provide a detailed description of the test case, specifying the acceptance criteria that are not met and describing the defects identified during testing.</li> <li>If it is determined during User Acceptance Testing that the development deliverables meet the approved acceptance criteria, the request shall be considered fulfilled.</li> </ul>	<p>Decision on compliance with acceptance criteria (successful or unsuccessful UAT report)</p> <p>Failed test case description</p>
19	Analyst (Architect, Developer)	Analyse UAT report and determine root cause	<p>Unsuccessful UAT report</p> <p>Failed test case description</p>	<p>Registered defects are reviewed and the root cause of the defect is determined.</p> <ul style="list-style-type: none"> <li>If defects are found in the source code or documentation, the development deliverables are returned to production</li> </ul>	Identified root cause of the defect

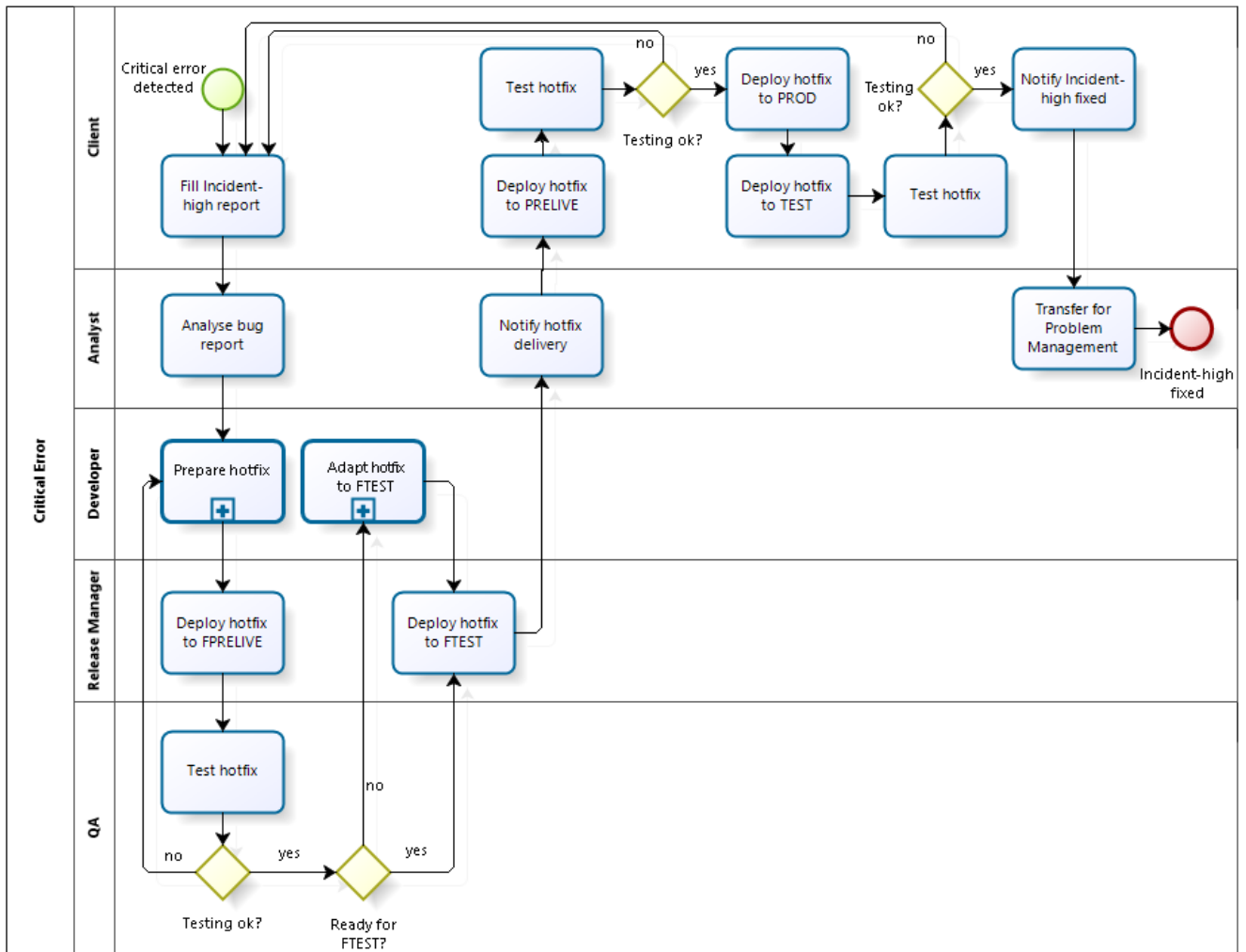
	Responsible person	Step	Input	Description	Output
				<p>(step “Develop deliverables / bug fixes”).</p> <ul style="list-style-type: none"> <li>If defects are related to the solution architecture, the deliverables are returned to the architects (step „Prepare DPA”.</li> </ul>	
20	Developer (Patch manager, PM)	Prepare delivery package for PROD	Positive UAT report	If User Acceptance Testing (UAT) is successful and the customer confirms compliance with the acceptance criteria, a delivery package for the production environment is prepared according to the terms and conditions pre-agreed with the customer. A change submission task is created, and the PM informs the customer about the changes being submitted, as needed.	Delivery package (patch)
21	Customer	Deploy to PROD	Delivery package	Customer deploys the provided patch in the production environment in accordance with internal rules and procedures.	Deployment report for development deliverables (software components)
22	Customer	Notify accepted deliverable	Positive UAT report Deployment report for development deliverables (software	<p>If it is determined during User Acceptance Testing that the development deliverables meet the approved acceptance criteria, Forbis is notified that the request is considered fulfilled.</p> <p>Change Acceptance Acts</p>	Signed Acceptance Act

	Responsible person	Step	Input	Description	Output
			components )	<p>Upon the expiration of the time allocated for User Acceptance Testing, Forbis must submit the Acceptance Act to the customer for approval. The customer must sign the Acceptance Act within 5 (five) working days from the moment of receipt.</p> <p>If the Customer refuses to sign the Acceptance Act, they must provide a reasoned response within 5 (five) working days from the date of receipt. In this case, Forbis may suspend work related to the further execution of the Order until the causes of the disagreement are clarified and resolved. If no reasoned response is provided, the works shall be deemed successfully accepted.</p> <p>Support services for the functionality running in the customer's production database shall only commence upon the signing of the Acceptance Act. Unless otherwise agreed in a specific Order, the software development deliverables shall be deemed acceptable if no more than 3 low-level defects are identified.</p>	
23	PM Project team: Analyst / Developer / Architect	End: Closed	Signed Acceptance Act	<p>End of process.</p> <p>Jira project/task is closed. Work Order is moved to the archive.</p>	Project/task closure actions

### 5.1.6 Hotfix Deployment Process

This section describes the hotfix deployment process.

Hotfixes shall be deployed outside the regular deployment schedule, irrespective of the standard change deployment timeline. After resolving a critical error, Forbis experts deploy the changes to the FPRELIVE environment and conduct testing there. Upon completion of testing, the hotfixes are transferred to the customer’s specialists for acceptance testing. If the testing is successful, the changes are deployed to the production environment (PROD).



	Step	Input	Description	Output
1	Critical error detected		Event initiating the preparation of changes – a critical error was detected.	Critical error identified


	Step	Input	Description	Output
2	Fill incident-high	Critical error identified	Customer describes the incident and, if known, specifies the required changes.	Error description
3	Analyse bug report	Error description	Analyst analyses the error description, identifies the required changes, or proposes a workaround.	Description of required changes
4	Prepare hotfix	Description of error and required changes	Forbis experts prepare the changes following the internal procedures.	Patch
5	Deploy hotfix to FPRELIVE	Patch	Forbis experts deploy the prepared patches in the FPRELIVE environment.	Deployed software components
6	Test hotfix	Deploy software components	Forbis experts carry out testing in the FPRELIVE environment.	Test report
7	Testing OK?	Test report	Based on the testing results: <ul style="list-style-type: none"> <li>in case of unsuccessful testing, the changes are returned for correction;</li> <li>in case of successful testing, the changes are transferred to the customer.</li> </ul>	Notification of successful testing / Returned for correction
8	Ready for FTEST?	Notification of successful testing	Forbis experts review the components included in the hotfix patch and determine if their versions differ from those deployed in the TEST environment (e.g., in cases where they were modified in other projects that have not yet been deployed to the PROD environment and are currently under testing).	Component version comparison report
9	Adapt hotfix for FTEST	Component version	If a hotfix is required, it must be adapted to the FTEST environment to reflect the actual components deployed there (e.g., the FTEST environment contains a newer version of a component which also needs to be fixed).	Patch with merged

	Step	Input	Description	Output
		comparison report		software components
10	Deploy hot fix to FTEST	Patch with merged software components	Changes are deployed in the Forbis testing environment – this is necessary to prevent the error from recurring in future releases. If a hotfix is required, it should be adapted to the FTEST environment to reflect the actual components deployed there (e.g., the FTEST environment contains a newer version of a component which also needs to be fixed).	Deployed software components
11	Notify hotfix delivery	Notification of successful testing	Upon successful testing, changes are transferred to the customer's specialists.	Patch package
12	Deploy hot fix into PRELIVE	Patch package	Customer specialists deploy changes in the PRELIVE environment.	Deployed software components
13	Test hotfix	Deployed software components	Customer specialists test the changes in the PRELIVE environment.	Test report
14	Testing OK?	Test report	Based on the testing results: <ul style="list-style-type: none"> <li>in case of unsuccessful testing, the changes are returned to Forbis for correction;</li> <li>in case of successful testing, the changes are deployed in the PROD environment.</li> </ul>	Successful UAT notification / Return for correction
15	Deploy hot fix into PROD	Successful UAT notification Patch package	Tested changes are deployed in the PROD environment. Customer specialists send the deployment log to Forbis. This is performed by sending it from the FORPOST software repository; example: <code>fru.fru_ver_adm.send_installation_info( par_recipients =&gt;'patchreg@forbis.lt', par_sender=&gt;'no-replay@sb.lt', --nurodyti par_smtp_host=&gt;'smtp.forbis.lt', --nurodyti</code>	Information about the deployed hotfix

	Step	Input	Description	Output
			par_days_back=>1);--nurodyti	
16	Deploy hotfix to TEST	Information about the deployed hotfix	Customer specialists deploy changes in the TEST environment.	Deployed software components
17	Test hotfix	Deployed software components	Customer specialists test the changes in the TEST environment.	Test report
18	Testing OK?	Test report	Based on the testing results: <ul style="list-style-type: none"> <li>in case of unsuccessful testing, the changes are returned to Forbis for correction;</li> <li>in case of successful testing, Forbis is notified of the resolved critical error.</li> </ul>	Successful UAT notification / Return for correction
19	Notify critical error fixed	Successful UAT notification	Customer notifies that the critical error has been resolved, and the task in ServiceDesk system is closed.	Critical error fix notification
20	Transfer for Problem management	Critical error fix notification	Task in the internal JIRA system is closed, and the information regarding the critical error is transferred to the problem management process.	Completed incident assessment report, closed task
21	End: Incident-high fixed			

## 6. Change Management

For system development and continual service improvement, Forbis has implemented the quality management system to identify and manage the overall quality assurance, which is based on principles of standard programme/project management methodologies, e.g. IT enabled major programme



transformation method and Project Management Institute framework. Worldwide-known standards and best practice methodologies, such as ISO 20000, apply where appropriate.

Core banking system is divided into the following environments:

- Production is the main operational environment in primary data centre.
- Backup is a copy of the main operational environment in reserve data centre, which in case of failure of the main environment becomes operational.
- First test environment is the main test environment, which as a rule, is created from the copy of the main environment with data depersonalization.
- Second test environment is a test environment with irrelevant data.

### **Release management**

Forbis has established a process of change management and quality assurance that covers verification, checkpoint reviews and testing to assure the performance quality.


For new releases and system updates, Forbis issues a new version of the core system once a year. Besides, ongoing changes of the system are also possible. A patch for current release is formed for this purpose. The installation may consist of:

- Release only. In this case any current version of core system could be upgraded by the new one presented.
- Release + Patch. Upgrade either the version or the patch, or upgrade the version and then set the patch on it.
- Patch only. Installation of the patch on current release.

New full versions are issued once per year. New versions include changes aimed for the improvement and extension of the functional scope of the system. Updates and new versions of the system are distributed in accordance with the Support agreement. Description of all updates is provided to the financial institution; therefore banks are acquainted with the content of changes before new version appears.

Development of new products is based on the awareness of the standard software used by the banks and other financial institutions. All activities directed towards the extension of functional possibilities of the system (new developments, modernization, optimization, etc.) are thoroughly tested by Forbis. Testing is considered successfully completed when the satisfaction of the financial institution's requirements is recognized.

New developments and modifications are accompanied with comprehensive instructions and documentation (reasons for changes, testing data, conditions and expected results).



The main advantage of using new products developed by Forbis is that they provide ideological and architectural integration of the entire financial system. In addition, there are a great number of other advantages the customer gains:

- New products and functions provided by Forbis are organically integrated into the CBS and use standard methods and algorithms. Thus, it is easy for the customer staff to master the new products more quickly and efficiently.
- The experience of system operation in different banks ensures versatility and flexibility of solutions.
- Forbis developers and analysts use international financial methods and principles and aim at the maximum automation and rationalization of the customer staff activities.

For smaller system updates, ongoing changes of the system are also possible. A patch for current release is formed for this purpose. The installation may consist of:

- Release only. In this case any current version of the Forbis CBS system could be upgraded by the new one presented.
- Release + Patch. Upgrade either the version or the patch or upgrade the version and then set the patch on it.
- Patch only. Installation of the patch on current release.

Updating the version of the running system can be performed in two scenarios: simple update and complex.

- Simple update. Updating the version is not related to changing the structure of database links or adding new subsystems. Can be executed without stopping financial institution's work according to the scenario.
- Complex update. The update of the version is associated with a major change in the structure of the database. It can only be performed after the financial institution's work has been stopped and requires additional testing in accordance with the attached scenario. It usually takes 2-4 hours to install a new version of the system.

## **Development and customization**

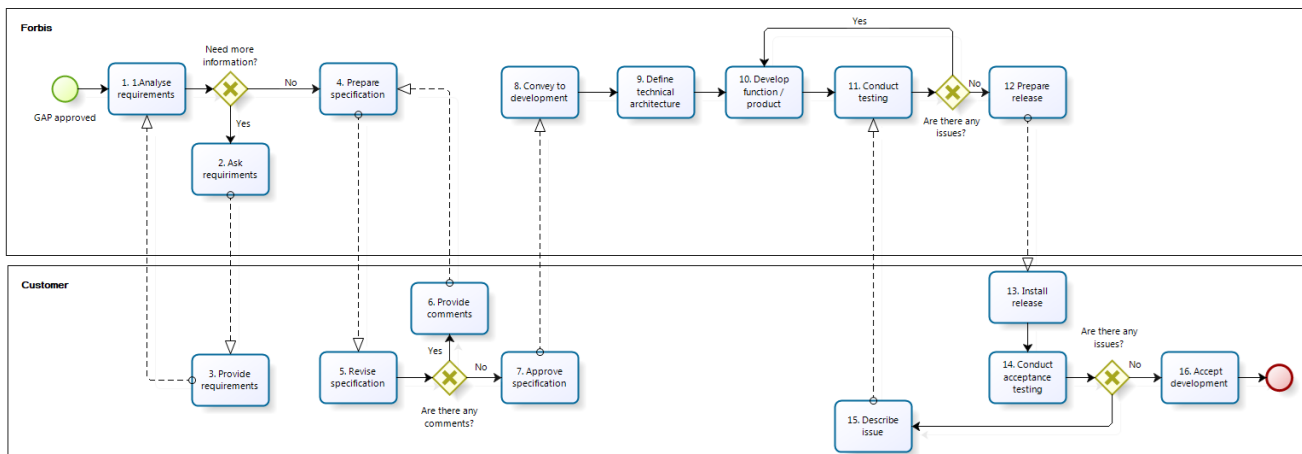
Ordered revisions are performed at a request of the customer as an additional service, in the case if the requirements of the customer exceed the terms of support. At this, the customer in writing provides formulated technical requirements for the revision, on the basis of which the analysis, development and coordination of the functional specification are made. On the basis of the agreed upon functional specification, the efforts required for implementation of the customer requirements

are estimated. On the basis of the performed estimation the cost of work and terms of implementation are agreed upon.

All activities directed towards the extension of functional possibilities of the system (new developments, modernization, optimization, etc.) are thoroughly tested by Forbis. Testing is considered successfully completed when the satisfaction of the financial institution's requirements is recognized.

New developments and modifications are accompanied with comprehensive instructions and documentation (reasons for changes, testing data, conditions and expected results).

The scheme of functionality development process:



A brief description of the process:

- Event, initiation of preparation of changes – a non-critical error has been detected or there has appeared a need in new functions. Filling out of the request for changes in Help Desk
- Patch preparation – Forbis experts prepare the changes, following the internal procedures.
- Patch installation in the test environment, installation log is recorded.
- Patch testing in the test environment.
- Is testing OK? Depending on the results of testing, the changes are returned for final revision and then transferred to the financial institution.
- Patch testing by the customer's specialists.
- Is testing OK? Depending on the results of testing, the changes are returned to CBS or are additionally tested in the main test environment, installation log is recorded.
- If the testing has been successful, the patch is installed in the production environment, if not, CBS is informed on the detected errors, installation log is recorded.
- Patch installation in the production environment, installation log is recorded.
- Completion of the process – the changes are successfully installed in the production environment.