APRIOIRIT

CONFIGURATION MANAGEMENT PLAN

Demo Marketplace

1. Introduction

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. The purpose of this document is to list the Configuration Management procedures, resources, tools.

2. Definitions and Abbreviations

Term/Abbreviation	Definition
PM	Project Manager
QA	Quality Assurance

3. Roles and Responsibilities

Role	Responsible
PM	
Team Leader	
Developer	
QA	

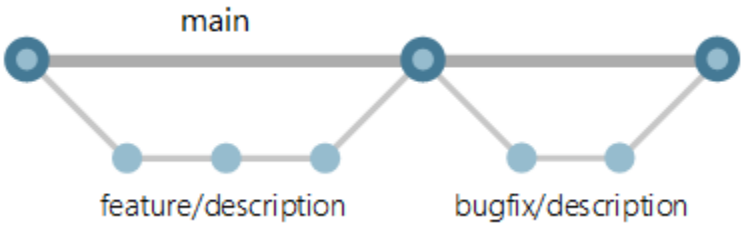
4. Configuration Items

4.1 Controlled Items

CI	Location	Responsible
Sources	https://github.com/apriorit/Demo1_Marketplace	Team Leader/Developers
Documents	Project_source_folder/docs	Team Leader/PM
Diagrams (HLD, DD)	Project_source_folder/docs	Team Leader/Architector
Reports	-	PM
Minutes Of meeting	-	Developers
Review Notes	-	Team Leader
Sonar Report	Web Demo (dev.local)	Team Leader
WIKI	https://github.com/apriorit/Demo1_Marketplace/tree/main/Docs/WIKI	Team Leader/Developers/ QA

4.2 Branching Policy

4.2.1 Branche structure



4.2.2 Feature merging – every merge include:

- **Pull request**
- **Code review**

5. Repository Structure

- **DB**
  - **Marketplace.Data** - DB context/seeding
  - **Marketplace.Data.Infrastructure** - Migrations
  - **Marketplace.Data.Models** - DB models
- **Doc** – Documentation(HLD, DD)
- **Integrations**
  - **Elastic**
    - **Marketplace.Integrations.ElasticSearch** - Elastic Search integration
  - **Payments**
    - **Marketplace.Integrations.Payments.LiqPay** - LiqPay integration
    - **Marketplace.Integrations.Payments.Stripe** - Stripe integration
- **Test**
  - **Tests** – unit tests
- **Marketplace.Api** - End points
- **Marketplace.Auth** - Identity server integration
- **Marketplace.Bootstrapper** - Initialization
- **Marketplace.Contracts** - Interfaces for Services
- **Marketplace.Infrastructure** - Tools
- **Marketplace.Models** - Models
- **Marketplace.Services** - Business Logic

6. Tools and Instructions/Techniques

6.1 Tools

Purpose	Tools
Version Control	
Docs Version Control	Git

Code Version Control	Git
Project Management	
Project Planning	Microsoft Project
Project Monitoring	Microsoft Project
Requirements Management	Microsoft Office 365
Development	
Compiler/Build system	VS 2022
Testing	
Test Run	

6.2 Techniques/Frameworks

Purpose	Techniques/Frameworks
Source code	.NET 6.0
DB	Microsoft SQL Server
Auth Server	Identity Server
Payment systems	Stripe
Log system	Elastic Search
Testing	XUnit

7. Quality Assurance Policy

The responsibilities within different test levels are listed below:

- **Unit testing -** carried out by developers
- **Integration testing -** passed in particular cases when integration with other products was affected or added. Mostly carried out by QA
- **System testing -** performed by QA. System testing is performed in the Swagger UI and Microsoft SQL Server.

We submit bug reports for each discovered bug in the system. Bugs are reported in Jira

Confirmation and regression testing are performed for each implemented requirement or other change in a product.

- **Confirmation testing:** verification of a requirement implementation**.**
- **Regression testing:** verification of impacted areas of a product after new requirements implementation.

8. History

Version	Date	Status	Change description	Author/Editor
0.1	25.09.2022	Draft	Initial version	O.Biriukov
	12.12.2022	Draft	Links	O.Onyshchenko