**Acko.com (Waterfall model)**

**Test Strategy**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1.Scope

2.Test Approach

3.Test Environment

4.Testing Tools and Release Control

5.Risk Management

6.Review and Approvals

**1. Scope**

This Test Strategy Document applies to the [www.acko.com](http://www.acko.com) insurance web application and outlines the comprehensive approach to testing under the Waterfall software development model. The document defines objectives, testing phases, roles, processes, environments, tools, and acceptable deliverables to ensure product quality and compliance.

The scope includes:

* Responsibilities of QA, developers, and stakeholders.
* Structured test case design, review, execution, and defect lifecycle.
* Manual and automated testing aligned with Waterfall phase sign-offs.
* Environment setup requirements and test data management.
* Test reporting and evaluation criteria.

**2. Test Approach**

The testing follows a phased Waterfall lifecycle:

* Requirements Phase: Review and clarification, preparing Requirement Traceability Matrix (RTM).
* Design Phase: Test planning, test case design and review.
* Implementation Phase: Verification of environment readiness.
* Testing Phase: System, integration, regression testing with defect logging and tracking.
* Deployment Phase: Final verification and validation.
* Maintenance Phase: Testing of fixes and enhancements post-release.

Test levels and types include functional, integration, system, regression, and acceptance testing conducted in a sequential manner according to Waterfall milestones.

**3. Test Environment**

|  |  |  |
| --- | --- | --- |
| **Environment** | **Purpose** | **Setup Requirements** |
| Development | Developer testing | Local machines, Git repositories, basic DB and web servers |
| QA/Test | Full functional and regression testing | Stable builds, test data, automation setup |
| Staging | UAT and final validation | Mirror production stack and data |

Component distribution:

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Development** | **QA/Test** | **Staging** |
| Web Server | 1 | 1 | 1 |
| Application Server | Local | 1 | 1 |
| Database Instance | Local/Shared | 1 | 1 |
| Supported Browsers | Chrome, Firefox | Chrome, Edge, Firefox | Chrome, Edge, Firefox |
| Test Users (Dummy Data) | Optional | 20 | 30 |

**4. Testing Tools**

|  |  |  |
| --- | --- | --- |
| **Category** | **Tool** | **Purpose** |
| Test Management | Jira, TestRail | Test case management and tracking |
| Defect Tracking | Jira | Defect logging and workflows |
| Automation | Selenium (Java) | Automated UI testing |
| Performance | JMeter | Load and performance testing |
| CI/CD | Jenkins | Build and test automation |
| Documentation | Confluence, Google Docs | Test documentation and collaboration |
| Version Control | Git | Source and test artifact control |

**Release Control**

Release and Regression Control:

* Release versioning follows Major.Minor.Patch format (e.g., v1.0.0).
* Code freeze enforced prior to testing phases.
* Regression scope updated based on change impact analysis.
* Build tagging aligned with Jenkins CI pipelines.
* Formal test reports and sign-offs before production deployment.

**5. Risk Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Impact** | **Mitigation** | **Contingency** |
| R1 | Changing or unclear requirements | High | Requirements freeze after sign-off | Change control board |
| R2 | Delay or unstable builds | High | Enforced code deadlines | Partial tests, escalate |
| R3 | Outdated/insufficient test data | Medium | Data refresh schedules | Backup data use |
| R4 | Environment downtime | Medium | Schedule management | Alternate environments |
| R5 | High defect density post-handover | High | Early unit/integration testing | Test extensions |
| R6 | Key personnel unavailability | Medium | Cross-training | Ownership redistribution |
| R7 | Automation script failures | Medium | Script review, stability checks | Manual fallback |
| R8 | Performance/security gaps post-release | Medium | Non-functional test cycles | Hotfixes/patch releases |
| R9 | Last-minute changes or bug fixes | High | Scope lock and change control | Formal changes and testing windows |

Weekly risk tracking and quarterly reviews ensure proactive mitigation.

**6. Review and Approvals**

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| QA Lead | Ensures compliance with testing strategy and scope |
| Project Manager | Oversees timeline, resources, and budget adherence |
| Product Owner | Approves requirements alignment and user acceptance |
| Development Lead | Confirms technical feasibility and integration |
| Business Team | Provides final business review and approval |