

|  |
| --- |
|  |
| Version 1.1 |

| RELATED ARTIFACTS | |
| --- | --- |
| Ref. | Artifact Name |
| [GLO] | [Common Glossary](https://info.epam.com/acronyms.html) |
| [SC] | [Process Description: Software Construction](https://epam.sharepoint.com/sites/policy/SitePages/Policy/Process%20Descriptions/Process%20Description%20Software%20Construction.aspx) |
| [ST] | [Process Description: Software Testing](https://epam.sharepoint.com/sites/policy/SitePages/Policy/Process%20Descriptions/Process%20Description%20Software%20Testing.aspx?path=Delivery%20Process%20Areas) |
| [DT] | [Process Description: Dynamic Testing](https://epam.sharepoint.com/sites/policy/SitePages/Policy/Process%20Descriptions/Process%20Description%20Dynamic%20Testing.aspx) |

|  |  |
| --- | --- |
| Acronyms & Abbreviations | |
| SEPG | Software Engineering Process Group |
| QMS | Quality Management System |
| POC | Proof of Concept |
| MVP | Minimum Viable Product |
| SOW | Statement of Work |
| DM/PM | Delivery Manager/Project Manager |

Contents

[1 Project specific impact to testing 4](#_Toc99101440)

[2 Scope of TESTING 4](#_Toc99101441)

[2.1 In scope 4](#_Toc99101442)

[2.2 Out of scope 5](#_Toc99101443)

[2.3 Third-Party systems 5](#_Toc99101444)

[2.4 System tested by other organizations 5](#_Toc99101445)

[3 Quality and Acceptance Criteria 5](#_Toc99101447)

[4 Test process description 6](#_Toc99101448)

[4.1 Test planning phase 6](#_Toc99101449)

[4.1.1 Entry/exit criteria for each phase 6](#_Toc99101450)

[4.1.2 Defects severity description 6](#_Toc99101451)

[4.2 Test design phase 6](#_Toc99101452)

[4.2.1 Test cases creation rules 6](#_Toc99101453)

[4.2.2 Managing test cases in the test management tool 7](#_Toc99101454)

[4.3 test execution phase 7](#_Toc99101455)

[4.3.1 Defect reports raising and management rules 7](#_Toc99101456)

[4.3.2 Defect lifecycle 7](#_Toc99101457)

[4.4 Test reporting phase 7](#_Toc99101458)

[4.4.1 Test report containment 7](#_Toc99101459)

[4.4.2 List of the metrics to track on a project 7](#_Toc99101461)

[5 Risk management 8](#_Toc99101463)

[6 test Team 8](#_Toc99101464)

[6.1 Roles and Responsibilities 8](#_Toc99101465)

[6.2 STAKEHOLDERS 8](#_Toc99101466)

[6.3 Communication plan 9](#_Toc99101467)

[6.4 Escalation plan 9](#_Toc99101468)

[7 Test schedule 9](#_Toc99101469)

[8 TEST Deliverables 10](#_Toc99101470)

# Project specific impact to testing

*<https://demoblaze.com/> is Ecommerce platform for selling phone.*

Project phase: *version 1.1*

*Technology stack:*

* *Installed JMeter ,Grafana, InfluxDB, Telegraf*

# Scope of TESTING

## In scope

|  |  |  |
| --- | --- | --- |
| System/component/interface under test | Description | Responsible side |
| *-Create test plan* |  | Performance Team |
| *-Create test scenarios* | *- Anonymous script - User script - Admin script* | Performance Team |
| *-Create Test Cases* | *-  UI Scenarios to select and order Items with logging off*  *-  UI Scenarios to select and order Items with logging on*  *-  UI Scenarios to add and\or remove Items by Admin* | Performance Team |
| *-Simulate Load* | *-Performance Testing Types*   * *Smoke testing* * *Capacity testing* * *Load testing* * *Regression testing* * *Scalability testing* * *Volume testing* * *Stability testing* * *Stress testing* * *Web Service Testing*   *-Performance Test Activates and deliverables*   * *Set-up of test environment* * *Set-up of test data* * *Performance Test Execution* * *Monitoring and collating results* * *Obtaining user acceptance* | Performance Team |
| *Analyze the Results* | * *Analyze the metric data captured during the test against the predefined expectations. If the expectations are not met, necessary modifications are made, and testing is repeated until the desired results are achieved.* | Performance Team |

## Out of scope

|  |  |  |
| --- | --- | --- |
| System/component/interface under test | Description | Reference |
| *Simulate Load on Prod Inv* | *Customer does not order prod environment* | *Will not tested on PROD environment* |

# Quality and Acceptance Criteria

*• The product should work according to the requirements and functional specification listed at*

*sections Scope of work.*

*• The product should not have bugs with severity Critical and Major to be released for production.*

# Test process description

|  |  |
| --- | --- |
| *Testing Type* | *Purpose of the testing* |
| *Load Testing* | *By doing the load testing we need to verify demoblaze.com UI, server behavior under normal and peak load conditions*  *Objectives:-  To Check the  response times, throughput rates, resource utilization levels* |
| *Stress Testing* | *By doing the stress testing  we need to evaluate our  demoblaze.com UI, server behavior when it is pushed beyond the normal or peak load conditions.*  *Objectives:- synchronization issues and memory leaks. Stress testing enables you to identify your application's weak points, and how it behaves under extreme load conditions* |
| *Capacity Testing* | *By doing the capacity, we can determine the server's ultimate failure point. We can use capacity planning to plan for future growth of the application, such as an increased user base or increased volume of data. For example, to accommodate future loads we need to know how many additional resources (such as CPU, RAM, disk space, or network bandwidth) are necessary to support future usage levels. Capacity testing helps you identify a scaling strategy to determine whether you should scale up or scale out.* |
| *Volume Test* | *By doing volume testing is to evaluate the performance of a system while increasing data flow. During testing, an increase in the intensity of system operations with a simultaneous increase in database volumes is emulated.* |
| *Scalability testing* | *By doing scalability testing analyzes how well our system scales in response to different load levels* |
| *Web Service Testing* | *Need to test the each individual web-service and need to find out the system behavior , when we apply the load*  *Objective:- To check the response time and the server utilization* |

## Test planning phase

### Entry/exit criteria for each phase

*New Feature testing Entry Criteria*

* *User stories are completed and groomed*
* *Feature development is complete*
* *All planned test cases are created and reviewed*
* *There are no questions regarding requirements on test cases and user stories*
* *New build is successfully deployed to test environment*
* *Test Environment is available and stable*

*Success Criteria*

* *There is no Blocker, Critical or Major defects in 'Open' state*
* *All known issues are fixed, accepted, or moved to the next Sprint*
* *All planned tests are run*

### Defects severity description

|  |  |  |
| --- | --- | --- |
| Severity | Meaning | Examples |
| Blocker | Behavior causes the business/consumer to stop working. Application under test crashes or inoperable | * the business users are unable to continue a successful business operation in the system unless the issue is resolved; * loss or partial loss of key functionality; * system crash; * massive performance degradation. |
| Critical | Behavior causes the business/consumer to operate in a limited way. No workaround available. | * loss or partial loss of key functionality; * operational error; * data integrity; * some performance degradation. |
| Major | Behavior allows business/consumer to continue working in a limited way with work around. | * partial loss of functionality of the software, but allows the user to continue proceeding normal business operations; * usability/UI issues causing confusion due to inconsistency or ambiguity. |
| Minor | Behavior has no functional impact on business/consumer and can also be attributed to UI aesthetics. | * the business user decides that the issue around this functionality is not vital to his use of the system; * a minor feature that is not functional in one module but the same task is easily doable from another module. |
| Trivial | The defect does not affect functionality or data, it is merely an inconvenience. | * cosmetics (font face/font size/text alignment; * misspelling. |

## Test design phase

### Test cases creation rules

Test cases must be created according to Performance Profile. Performance Profile is a set of operations with given intensities, obtained on the basis of collecting statistical data or determined by analyzing the requirements for the system under test.

|  |  |  |
| --- | --- | --- |
| User | Action | Frequency |
| Admin | Login | 1/4h |
| Admin | Ban user | 10/1h |
| Admin | Delete message | 5/1m |
| User | Login | 20/1m |
| User | Open feed | 34/1m |
| User | Post comment | 2/1s |
| Guest | Open feed | 0.5/1s |

## test execution phase

### Defect reports raising and management rules

*The required fields in a bug report are:*

* + *Bug Summary*
  + *Bug ID*
  + *Severity*
  + *Status*
  + *Priority*
  + *Steps to reproduce*
  + *Actual Result*
  + *Expected Result*

### Defect lifecycle

*The bug life cycle consists of a definite number of steps such as New, Assigned, Opened, Duplicate, Differed, Not a Bug, Rejected, Reopened, Fixed, Retest, Verified, and Closed.*

## Test reporting phase

### Test report containment

* *List of the tasks with priority and status.*
* *Test Purpose*
* *Test Summary*
* *Results of test with metrics.*
* *Graphics with metrics.*

### List of the metrics to track on a project

|  |  |  |  |
| --- | --- | --- | --- |
| *utThroughp* | *It is the sum of the data rates that are delivered to all terminals in a network* | *Server metrics* | *We need to monitor each individual system resources like Memory, CPU, Disk I/O, network bandwidth* |
| *Response Time* | *We need to check the web services and UI response time* | *Client side metrics* | *FCP (First Contentful Paint)*  *LCP (Largest Contentful Paint)*  *SI (Speed Index)*  *TTI (Time to Interactive)*  *TBT (Total Blocking Time)*  *CLS (Cumulative Layout Shift)* |
| *Maximum concurrent and simultaneous users* | *We have to check how many concurrent user can sustain in the environment* |

# Risk management

|  |  |  |  |
| --- | --- | --- | --- |
| Risk name | Severity | Symptom | Mitigation plan |
| Ills or vacations of Team members | High | Planned Functionality won’t be delivered properly due to lack of resources. | In the case of a vacation, take this into account when planning a sprint.  Take on higher priority tasks.  Help from experienced colleagues |
| Permanent changes in back-end /API. | High | Be notified by developers if any changes | Ask Dev teams |
|  |  |  |  |
|  |  |  |  |

# test Team

## Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Project Role | Name | Location | Responsibilities |
| 1 | QA | Nehrii Stas | Kyiv | Performance Team |

## STAKEHOLDERS

|  |  |  |
| --- | --- | --- |
| # | Project Role | Name, e-mail, location |
| Internal (inside EPAM) | | |
|  | <Project Manager> | Name PM |
|  | <Test Leader> | Nehrii Stas |
| External (from customer side) | | |
|  | <Product Owner> | Name PO |
|  | <Business Analyst> | Name BA |

# Test schedule

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Sprint day 1 | Sprint day 3 | Sprint day 4 | Sprint day 5 | Sprint day 7 | Sprint day 8 | Sprint day 9 |
| *-Create test plan* |  |  |  |  |  |  |  |
| *-Create test scenarios* |  |  |  |  |  |  |  |
| *-Create Test Cases* |  |  |  |  |  |  |  |
| *-Simulate Load* |  |  |  |  |  |  |  |
| *Analyze the Results* |  |  |  |  |  |  |  |

# TEST Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Artifact to be provided | Author/ Responsible Person(s) | Frequency (delivery time) | Method of delivery |
|  | Test Plans | Performance Team | Once before the testing start | Document in the KB |
|  | Test Cases | Performance Team | Before the testing start | Document in the KB |
|  | Bug reports | Performance Team | Upon finding a bug | Tickets in Bug Tracking system  (Jira) |
|  | Test Result Reports | Performance Team | Upon releasing a build | Document in the KB |

| REVISION HISTORY | | | | | |
| --- | --- | --- | --- | --- | --- |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Date |
| 1.1 | Created | Nehrii Stas | 2.06.2023 |  |  |
|  |  |  |  |  |  |