Logo

Description automatically generated

**Product Requirements Document**

**on**

**DORA Metrics Tool**

**Document Management & Version Control**

|  |  |
| --- | --- |
| **Document Title** | DORA Metrics Tool |
| **Issued By** | Rithvick Abraham Rajesh |
| **Issue Date** | 30/09/2022 |
| **Approved By** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Remarks** |
| 30/09/2022 | 1.0 | Rithvick A R | First draft |

Contents

[**1.** **Overview** 4](#_Toc115442071)

[**2.** **Business Process Affected** 4](#_Toc115442072)

[**3.** **High Level Scope** 4](#_Toc115442073)

[**4.** **High Level Architecture** 5](#_Toc115442074)

[**5.** **Workflow** 5](#_Toc115442075)

[**5.1. Backend** 5](#_Toc115442076)

[**5.2. Frontend** 5](#_Toc115442077)

[**6.** **Functional Requirement** 6](#_Toc115442078)

[**6.1. Use Case FPR-001** 6](#_Toc115442083)

[**6.2. Use Case FPR-002** 6](#_Toc115442084)

[**6.3. Use Case FPR-003** 7](#_Toc115442085)

[**6.4. Use Case FPR-004** 7](#_Toc115442086)

[**7.** **Technology Stack** 7](#_Toc115442087)

[**8.** **DORA metrics** 7](#_Toc115442088)

[**8.1 Deployment Frequency** 7](#_Toc115442089)

[**8.2 Lead Time for changes** 8](#_Toc115442090)

[**8.3 Change Failure Rate** 8](#_Toc115442091)

[**8.4 Time to Restore Services** 8](#_Toc115442092)

[**9.** **Assumptions/Constraints** 8](#_Toc115442093)

[**10.** **Questions** 8](#_Toc115442094)

# **Overview**

Requirement is to create a DORA metrics assessment and continuous monitoring tool to view the metrics from different DevOps providers on custom built dashboards.

# **Business Process Affected**

Currently, the flexibility provided by existing solutions are very limited and cannot be integrated easily with different DevOps providers and code repositories. This makes it difficult to track the DORA metrics since the deployments, changes like pull requests and commits, and incident data are usually in separate or unconnected systems. To have better flexibility and accessibility in getting the metrics data, a custom tool is to be designed that can capture data from these different sources and display them in fully configurable dashboards. This makes it easy to combine the data from disparate systems and to create and display the DORA metrics in customizable and intuitive dashboards that provides a bird’s-eye view of the KPIs.

# **High Level Scope**

|  |  |
| --- | --- |
| **No.** | **Scope statement** |
| 1 | Create a fully configurable tool that can gather the required data from different sources to create customizable dashboards for viewing the DORA metrics. |

# **High Level Architecture**

# **Workflow**

# **5.1. Backend**

* Event data from different systems and sources are collected.
* Event data is parsed into deployments, changes and incidents.
* Parsed data is stored in data warehouse
* Tables are created for transforming and storing KPI specific data for calculating the different metrics.
* Backend API calculates the metrics details.
* Monitoring and dashboard tools and integrated with the derived tables.

# **5.2. Frontend**

* User logs in
* User configures the data endpoints
* User selects the DevOps provider
* User selects the organization and project for authorization and access
* User selects the source code repository and incident management service for authorization and access.
* User selects the pipeline and branches to call the required APIs
* API endpoints are hit
* User selects and configures the metrics
* Response is parsed and displayed in the dashboard

# **Functional Requirement**

|  |  |
| --- | --- |
| **No** | **Use case** |
| FPR-001 | Frontend application to configure the endpoints and dashboard. |
| FPR-002 | Backend pipeline for the collection and parsing of data. |
| FPR-003 | Backend application for calculating metrics. |
| FPR-004 | Dashboard must be pluggable to other services. |



# **6.1. Use Case FPR-001**

* Homepage should display the currently configured dashboards.
* Users should be able to choose the different projects.
* Users should be able to create new projects or configure existing projects (select the repo, branch, pipeline, release etc.)
* Users should be able to create new dashboards or configure existing dashboards (type of charts, customizations etc.).
* Users must be able to manually refresh the dashboard data.
* Users have the option to export dashboards.

# **6.2. Use Case FPR-002**

* Basic data type is an event.
* Events are internally created from inputs to the service. For example, when you push a commit, an Event is added to the database.
* Event contains information like the event ID, commit SHA, time of the commit, etc.
* Individual domain-specific records respectively for changes, deployments or incidents based on the incoming data must be added.
* Data from changes like commits, pull requests etc. from source code repositories like GitHub should be set up using webhooks if possible.
* Incident handling webhook must be setup.
* The issue tracker must send events by webhook based on actual user interactions with issues or tickets.
* The status of an incident created must be updated once the issue is resolved and the time of creation and resolving must be stored.

# **6.3. Use Case FPR-003**

* APIs to bit hit for each of the DevOps providers must be categorized.
* APIs to be hit for the source repositories and incident handlers needs to be categorized.
* APIs are hit to collect data from the different services and providers based on the user configuration.
* Metrics calculations are performed.

# **6.4. Use Case FPR-004**

# **Technology Stack**

|  |  |  |
| --- | --- | --- |
| Service | Tool | Comments |
| Event Handler |  |  |
| PUB/SUB |  |  |
| Data Parsing |  |  |
| Data warehouse |  |  |
| Database |  |  |
| Frontend |  |  |
| Backend Functions |  |  |
| Monitoring/Dashboard |  |  |

# **DORA metrics**

# **8.1 Deployment Frequency**

How often an organization successfully releases to production. We calculate this by looking at the number of successful pipeline runs to the environment.

Data collection

We send a deployment event after pushing code to the production environment by a direct call to the API endpoint.

Calculation:

Weekly: {number of deployments} / 7

Monthly: {number of deployments} / 30)

|  |  |
| --- | --- |
| **DF (based on monthly values)** | **Ranking** |
| > 1 | Elite |
| < 1 & > 0.14 | High |
| < 0.14 & >0.03 | Medium |
| < 0.03 | Low |

# **8.2 Lead Time for changes**

The amount of time it takes a commit to get into production.

# **8.3 Change Failure Rate**

# **8.4 Time to Restore Services**

# **Assumptions/Constraints**

# **Questions**