The overall DevOps Maturity Metric is part of the CB franchise scorecard. It is used to measure the efficiency and capabilities of the development and release processes within the franchise.

It is a combination of DORA metrics and the DevOps Assessment scores.

This and the DORA metrics are captured in the DevOps Assessment tool: [Tech Tree Assessment](https://devops-techtree-assessment.prd.spa.paas.pnf.banksvcs.net/)

DORA (DevOps Research & Assessments) metrics were first defined in the Accelerate book by Nicole Forsgren et al. and have been widely adopted across the world. The DORA organisation is now managed by Google: <https://cloud.google.com/devops/state-of-devops/>

The idea behind these metrics is that they very simply measure how a team is performing by balancing two throughput measures and two stability measures.

The below definitions and levels are directly from DORA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Aspect of software delivery performance** | **Elite (4)** | **High (3)** | **Medium (2)** | **Low (1)** |
| **Deployment frequency**  For the primary application or service you work on, how often does your organization deploy code to production or release it to end users? | On-demand (multiple deploys per day) | Between one per day and one per week | Between once per week and one per month | Between once per month and once every six months |
| **Lead time for changes**  For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)? | Less than a day | Between one day and one week | Between one week and one month | Between one and six months |
| **Time to restore service**  For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)? | Less than one hour | Less than one day | Less than one week | Between one week and one month |
| **Change failure rate**  For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)? | 0 - <= 2% | 2 - <= 5% | 5 - <= 10% | > 10% |

A score of between 1 and 4 is given for each DORA metric, based on the DORA Low (1), Medium (2), High (3), Elite (4) levels.

The slight anomaly of the duplicated range in values for Change Failure Rate of 0-15% in Elite, High and Medium is a reflection of lack of confidence in statistical relevance of the more fine grained data shown in the 2019 report.

**Example**

|  |  |  |  |
| --- | --- | --- | --- |
| **DORA Metric** | **Value** | **Rank** | **Score** |
| Deployment Frequency | 2 days | High | 3 |
| Lead Time for Changes | 20 hours | Elite | 4 |
| Time to restore service | 1 week | Low | 1 |
| Change Failure Rate | 47% | Low | 1 |
| **Average** |  | **Medium** | **2.25 or 56.25%** |

**Deployment Frequency**

**DORA Metrics definition here are based on book "Accelerate"**

**Definition:**

How often code is deployed in production for the primary service or application

**Source of Data:**ServiceNow Change Records

**Formula:**365/ (# of deployments\* in last one year for the CI)

\*Rolling one year data for the specific month

**Data Capture:**

* Calculated as average time interval at which a Change is deployed for a CI (rolling yearly average)
* For Specific cases where number of changes per CI is less than 1, it is taken as 1 and accordingly maximum number of days for Devlopyment Frequency is taken as 365.
* CIs performing deployment in last one year are considered 'Active', else treated 'Passive'.

**Change Records Selection criteria as below:**

1. Change Records included based on Type: **MCR / ECR / RCR**
2. Change Records included based on Change State: **Closed Complete**
3. Exclusion by change closure code: **Cancelled, Aborted, Backed out**

**Detail of calculation steps:**

* Extract all Change Records for last one year prior to the reporting month as per the selection criteria

**Example:**  
If a Platform, which has 50  CIs,  does 100  deployments in a year. Yearly deployments per CI within the Platform = 100 / 50  = 2 Deployments   
**Deployment Frequency is calculated as:**

                         365  /  2        =      182.5 Days.

* Grouping is done at respective levels  (Platform,  Domain, Franchise  and Group ) .
* CI is the most granular entity for reporting DF

**Lead Time For Changes**

**Definition:**

Average Time taken to go from "In Progress” to "Last Completed Status" (Can be "Closed" or "Done or  "Resolved")

**Source of Data:**JIRA.

**Formula:**Sum of ( "Last Completed Status" time  - "In Progress" time )  for all JIRA for the specific month / Total # of JIRA tickets for that specific  month.

**JIRA Issue types:** Sub Task, Epic, Investment Theme, Outcome, Opportunity, Initiative, Portfolio Epic and Feature will be excluded while Calculating Lead Time.

**Data Capture & Calculation:**

* All the Jira tickets are pulled from Sandbox instance (with prod data up to a specific refresh date) which has "In Progress" status and last status as "Resolved" or  "Done" or "Closed"
* Sum of time difference between "In Progress" to "Last Completed Status" is taken for all JIRA issues for the specific month.
* This sum is divided by total # of JIRA issues in that month.
* Grouping is done for the JIRA Project Key.
* Based on the JIRA Project Key to CI mapping (as per inputs provided by teams )  this is aggregated at Platform/ Domain / Franchise level

**Time To Restore Service**

**Definition:**

Average time (minutes) to restore service(s) (following an outage) in a calendar month.

**Source of Data:** ServiceNow Incidents Records

**Formula:**Sum of impact duration / total # of impacting incidents in the month

**Data Capture:**

For calculating below gudelines used for fetching records for the spacified period:

* All incident records with Impact 1,2 and 3 excluding child incidents.

**Detail of calculation steps:**

* Extract all the Incidents as per the selection criteria
* Only incidents where "parent incident" column is blank are considered to remove impact counted twice ( for parent & child)
* Impact Duration is calculated as difference of Work Start and Work End ( and not based on open time and  resolved time )
* A simple average of impact duration gives us **MTTR or Time to Restore Service in minutes**
* For average, grouping is done at the respective levels:**CI, Platform,  Domain, Franchise and Group**

## Change Failure Rate

### Definition:

%of Changes for the primary system which resulted in degraded service or required remediation.

**Formula:**For a specific month

( # of Changes causing incidents / Total Completed Changes )  \* 100

### Data Capture:

**Change Records Selection criteria:**

* Change Records included based on Type: **MCR / ECR /RCR**
* Exclusion by change closure code: **Cancelled**
* Change Records included based on Change State: **Closed Complete**

### Detail of Calculation Steps:

* Extract All Change Records for the respective month  based on the selection criteria
* Change Records which have a value "True" for the column "Caused Incident" are considered as  Failed Change.
* A simple % is taken to calculate the CFR.

**Example:** If there are total 100 change records for a platform in that specific month and 2 changes caused incident then the CFR for that platform in that month is calculated as 2%

* Calculate CFR with grouping at respective level: CI , **Platform,  Domain, Franchise and Group**

**Source of Data:** Service now Incident records

**Time On Inner Loop Activities & Toil (NWG)**

**Toil**% is calculated based on the Assessment taken by teams. There is a Toil question:

"Thinking about your software delivery lifecycle what % of your team's work can be categorized as toil?"

**Possible Answers are:**

  - "Less than 10%"

  - "Between 10% to 25%"

  - "Between 25% to 50%"

  - "Above 50%"

  - "N/A“

**We take the punitive value for each answer to calculate the overall Toil % for NWG**

**Below is one sample calculation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **​% of Teams work categoried as Toil** | **​# of Teams** | **​Toil (%)** | **Weighted Toil (# of tems \* Toil%)​** |
| ​Above 50% | ​29 | ​70 | ​2030 |
| ​Between 10% to 25% | ​91 | ​25 | ​2275 |
| ​Between 25% to 50% | ​31 | ​50 | ​1550 |
| ​Less than 10% | ​65 | ​10 | ​650 |
| ​**Total** | **​216** | **​** | **​6505** |

**Effective Toil% for NWG = 6505/216 = 30.12%**

**Measurement Formula:**

**# of staff involved in inner loop activities\*(1-Toil%)/Total Tech Population**

Data from TechTree is being used to estimate the level of Toil (self-assessed).

Time on inner loop and toil data calculations are done based on NWG level.

\*Inner Loop Job Families are: Software Engineers, Infrastructure Engineers, Data Analysts and Testers

The DevOps Assessment is made up of 100+ questions across 11 categories for Generic persona and 13 catagories for Mainframe persona. Each category has a score of between 0 and 100. These scores are added up and averages to get an overall DevOps Assessment score.

**Example of 11 categories for Generic persona:**

|  |  |
| --- | --- |
| **Assessment Category** | **Score** |
| Dormancy Strategy | 74% |
| Configuration Management | 60% |
| Code Quality | 60% |
| Build Practices | 40% |
| Deployment Practices | 40% |
| Release Practices | 60% |
| Infrastructure as a Service | 80% |
| Automated Testing | 80% |
| Application Architecture | 70% |
| Monitoring and Reporting | 60% |
| Security | 50% |
| **Average** | **61%** |

**Final Combed DevOps Metric**

The final score is weighted equally between the DORA score and the assessment score. So for our examples:

56.25% (DORA) + 61% (Assessment) = 117.25 / 2 = 58.6%