github.com/governify/zoo/tree/main/bluejay/tpa/seville/ISPP-2024/v1.1

Support: https://app.gitter.im/#/room/#governify_community:gitter.im

Changelog

Version	Date	Description	Authors
v1.0	2024-02-27	Complete description	Javier Fernández Manuel Otero
v1.1	2024-07-25	Minor fixes	Javier Fernández

Index

Introduction and context

This document provides a detailed explanation on how to correctly interpret the TPA in Bluejay.

Some definitions in relation to the Governify ecosystem (more precisely, for Bluejay) that will facilitate the understanding of the TPA are explained in the following points:

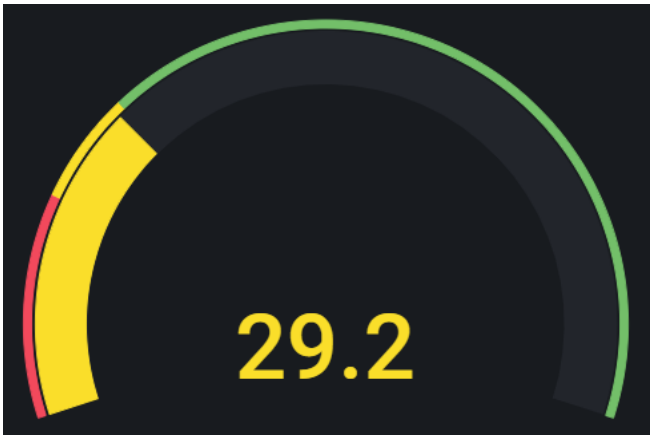
1. A **TPA (Team Practices Agreement)** is made up of **TPs (Team Practices)**.
2. Each **TP** is made up of one **guarantee.**
3. A **guarantee** defines what a work team must comply with to ensure its proper functioning.
4. Each **guarantee** needs an **objective**, which will be using one or more **metrics** along with a threshold value, and a **period** that defines the time boundary for the compliance of the guarantee.
5. Each **metric** represents a single variable that will be measured from a project.

   Example

   If we want the work team to make at least one pull request weekly, that will be our **TP**, which will be included in the **TPA**. We will then need to create a **guarantee,** which will have an **objective** that will need a **metric** to represent the variable "number of pull requests created by the team" (we will give it the ID "NUMBER_PR_CREATED") and a **period**, which will be "weekly" in this case.

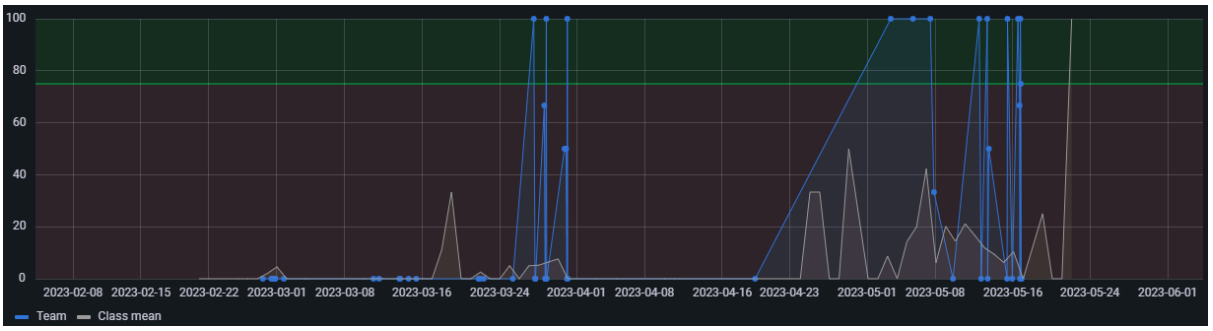
   Given all the previous context, our **guarantee's objective** would be "NUMBER_PR_CREATED >= 1" and the **period** would be "weekly".

6. Each **guarantee** is represented by a **block** in the **dashboard** view of Bluejay. A block is characterized by a title and a series of graphs (one or more).
7. **Blocks** have different display **styles** according to the guarantee. In this TPA we will use the following blocks:
   a. **Gauge**: Percentage of compliance with the guarantee by the team in a certain period.

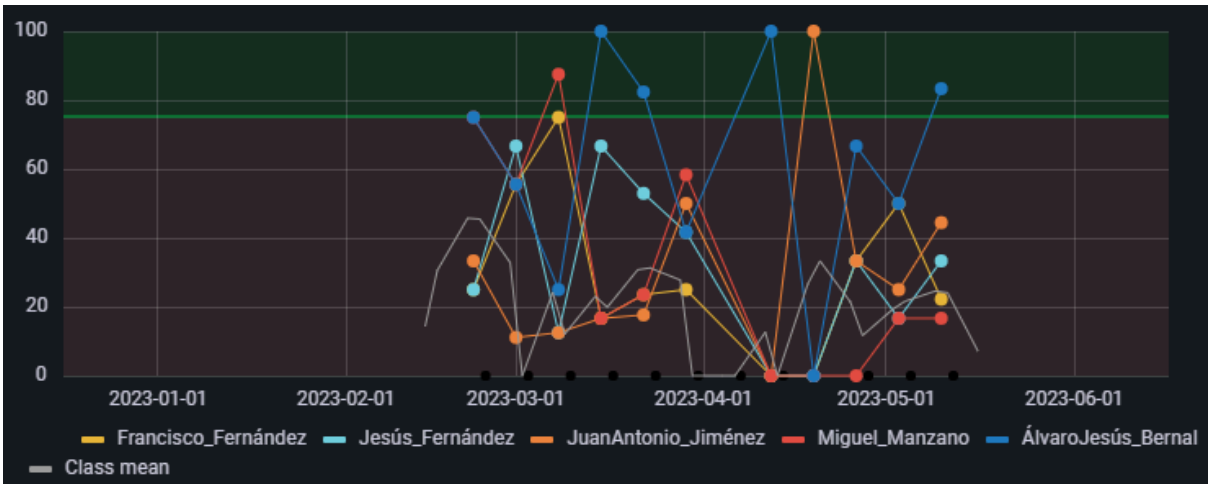


b. **Timegraph by team/member**: Evolution of the value that a guarantee takes over time. This block requires 1 guarantee and can be used for displaying data by team (blue line) or by each member (colored line), both along with the class mean (gray line).

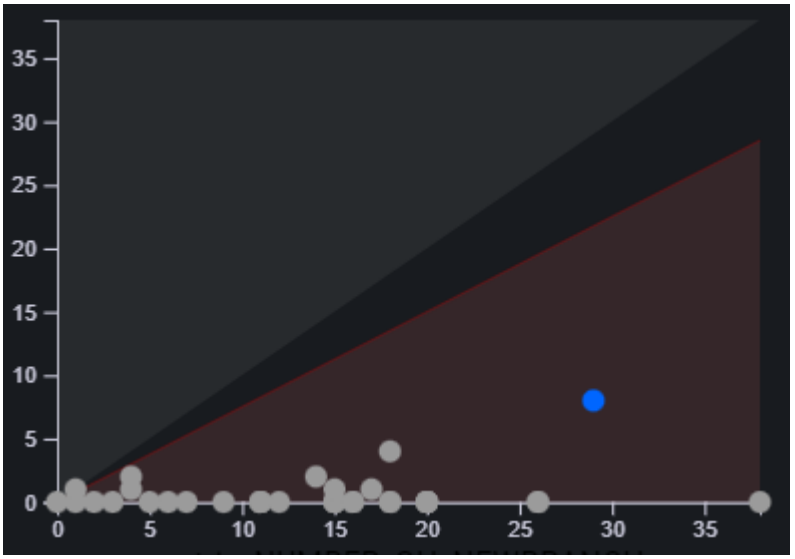
Team



Member



c. **Correlation**: Percentage of compliance with the guarantee by the team in a certain period in relation to the class mean. This block requires 2 metrics for the X and Y axes.



TP1: Percentage of new branches related to “In Progress” issues (by Team, Hourly)

When an issue is started, it should usually get its own branch.

This TP provides information about the team in relation to creating branches right after moving an issue to "In Progress". The correct application of this practice requires that the following actions are done simultaneously (checked every hour):

1. An **issue** is moved from the "Todo" column to the "In Progress" column.
2. A **branch** that contains the number of the **issue** moved to "In Progress" in the title, in the format **/number**, is **created**.

> **Example**
>
> 1. The issue called “Creation of pets for vets **#35**” is moved from “Todo” to “In Progress”.
> 2. Then, a new branch is created with the number of the issue included in its name as follows: “PetsCreation**/35**”, “feature**/35**-Pets-creation”, etc.

Calculation Pre-Requirements (if not met, a point won’t be created in the corresponding period):

- A new branch was created during the last hour.

Metrics:

- **COUNT_INPROGRESSISSUES_NEWBRANCH**: number of branches created including in its name the number of an issue in the column of “In Progress” by the team in the last hour.
- **NUMBER_GH_NEWBRANCH**: number of branches created by the team in the last hour.

Guarantee:

$$\frac{COUNT\_INPROGRESSISSUES\_NEWBRANCH}{NUMBER\_GH\_NEWBRANCH} \times 100 \geq 75$$

HOURLY by **TEAM**

Possible False-Negative:

- Sample sequence:

  1) The issue is moved to the “In Progress” column.

  2) The branch is created with the right issue number within the same hour than step 1.

  3) The issue is moved to “In Review” or “Done” within the same hour as step 1.

Dashboard:

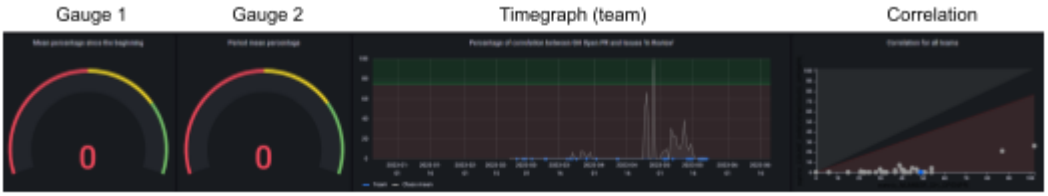
Derived metric

CORRELATION_INPROGRESSISSUES_NEWBRANCH: Ratio between **COUNT_INPROGRESSISSUES_NEWBRANCH** and **NUMBER_GH_NEWBRANCH**

CORRELATION_INPROGRESSISSUES_NEWBRANCH =

$$\frac{COUNT\_INPROGRESSISSUES\_NEWBRANCH}{NUMBER\_GH\_NEWBRANCH} \times 100$$

Blocks



- **Gauge 1:** Mean percentage of **CORRELATION_INPROGRESSISSUES_NEWBRANCH** since the beginning of the project.
    - **Low range**: [0%, 50%)
    - **Mid range**: [50%,75%)
    - **High range**: [75%, 100%]
- **Gauge 2:** Mean percentage of the **CORRELATION_INPROGRESSISSUES_NEWBRANCH** on the selected period in Grafana.
    - **Low range**: [0%, 50%)
    - **Mid range**: [50%,75%)
    - **High range**: [75%, 100%]
- **Timegraph (team)**: Evolution of **CORRELATION_INPROGRESSISSUES_NEWBRANCH** over time.
    - **Acceptable threshold** greater or equal than 75%
    - **Unacceptable threshold**, lower than 75%
- **Correlation**: Correlation between the metrics **COUNT_INPROGRESSISSUES_NEWBRANCH** and **NUMBER_GH_NEWBRANCH** on the period selected in Grafana.
    - X axis: **NUMBER_GH_NEWBRANCH**
    - Y axis: **COUNT_INPROGRESSISSUES_NEWBRANCH**

TP2: Percentage of PR related to “In Review” issues (by Team, Hourly)

When an issue is ready for review, a PR should be opened to facilitate that review.

This TP provides information about the team in relation to creating a pull request right after moving an issue to "In Review". The correct application of this practice is as follows:

1. An **issue** is moved from the "In Progress" column to the "In Review" column.
2. A **pull request** that contains the number of the **issue** moved to "In Review" in the title, in the format **/number**, is **created**.

> **Example**
>
> 1. The issue called “Creation of pets for vets **#47**” is moved from “In Progress” to “In Review”.
> 2. Then, a new pull request is created with the number of the issue included in its name as follows: “**/47** - Creation pets hotel”, “Pets hotel creation **/47**”, “(**/47**) Pets hotel creation”, etc.

Calculation Pre-Requirements (if not met, a point won’t be created in the corresponding period):

- A new pull request was created during the last hour.

Metrics:

- **COUNT_INREVIEWISSUES_OPENPR**: number of pull requests created including the number of an issue “In Review” by the team in the last hour.
- **NUMBER_GH_OPENPR**: number of pull requests created by the team in the last hour.

Guarantee:

$$\frac{COUNT\_INREVIEWISSUES\_OPENPR}{NUMBER\_GH\_OPENPR} \times 100 \geq 75$$

HOURLY by **TEAM**

Possible False-Negative:

- Sample sequence:

  1) The issue is moved to the “In Review” column.

  2) The P/R is created with the right issue number within the same hour than step 1.

  3) The issue is moved to “Done” within the same hour as step 1.

Dashboard:

Derived metric

CORRELATION_INREVIEWISSUES_OPENPR: Ratio between **COUNT_INREVIEWISSUES_OPENPR** and **NUMBER_GH_OPENPR**

$$\textbf{CORRELATION\_INREVIEWISSUES\_OPENPR} = \frac{COUNT\_INREVIEWISSUES\_OPENPR}{NUMBER\_GH\_OPENPR} \times 100$$

Blocks

- **Gauge 1:** Mean percentage of **CORRELATION_INREVIEWISSUES_OPENPR** since the beginning of the project.
  - **Low range**: [0%, 50%)
  - **Mid range**: [50%,75%)
  - **High range**: [75%, 100%]
- **Gauge 2:** Mean percentage of the **CORRELATION_INREVIEWISSUES_OPENPR** on the selected period in Grafana.
  - **Low range**: [0%, 50%)
  - **Mid range**: [50%,75%)
  - **High range**: [75%, 100%]
- **Timegraph (team)**: Evolution of **CORRELATION_INREVIEWISSUES_OPENPR** over time.
  - **Acceptable threshold** greater or equal than 75%
  - **Unacceptable threshold**, lower than 75%
- **Correlation**: Correlation between the metrics **COUNT_INREVIEWISSUES_OPENPR** and **NUMBER_GH_OPENPR** on the period selected in Grafana.
  - X axis: **NUMBER_GH_OPENPR**
  - Y axis: **COUNT_INREVIEWISSUES_OPENPR**

TP3: Percentage of mergedPR related to “Done” issues (by Team, Hourly)

Merging a PR should be followed by marking the related issue to “Done”

This TP provides information about the team in relation to creating a pull request right after moving an issue to "In Review". The correct application of this practice is as follows:

1. A **pull request** that contains in the title the number (in the format **/number**) of an **issue** located in the “In Review” column,, is **merged**.
2. The **issue** is moved from the "In Review" column to the "Done" column.

> **Example**
>
> 1. A pull request that contains the number of the issue in its title, as follows: *“**/51** - Creation reserves”, “Reserves creation **/51**”, “(**/51**) Reserves creation”, etc.*, is merged.
> 2. The issue called “Creation of pets for vets **#51**” is moved from the “In Review” column to the “Done” column.

Calculation Pre-Requirements (if not met, a point won’t be created in the corresponding period):

- A new pull request was merged during the last hour.

Metrics:

- **COUNT_DONEISSUES_MERGEDPR**: number of pull requests merged that contain the number of an issue that is in “Done” by the team in the last hour.
- **NUMBER_GH_MERGEDPR**: number of pull requests merged by the team in the last hour.

Guarantee:

$$\frac{COUNT\_DONEISSUES\_MERGEDPR}{NUMBER\_GH\_MERGEDPR} \times 100 \geq 75$$

HOURLY by **TEAM**

Dashboard:

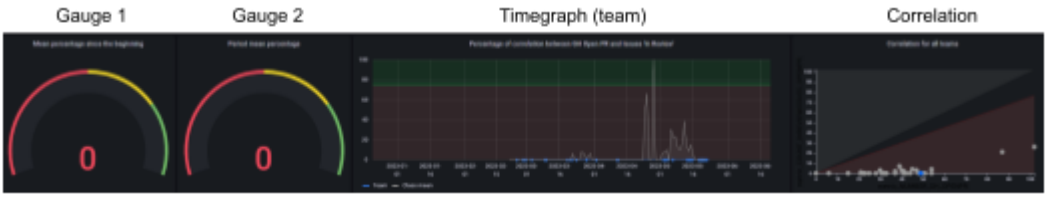
Derived metric

CORRELATION_DONEISSUES_MERGEDPR: Ratio between **COUNT_DONEISSUES_MERGEDPR** and **NUMBER_GH_MERGEDPR**

CORRELATION_DONEISSUES_MERGEDPR =

$$\frac{COUNT\_DONEISSUES\_MERGEDPR}{NUMBER\_GH\_MERGEDPR} \times 100$$

Blocks



- **Gauge 1:** Mean percentage of **CORRELATION_DONEISSUES_MERGEDPR** since the beginning of the project.
    - **Low range**: [0%, 50%)
    - **Mid range**: [50%,75%)
    - **High range**: [75%, 100%]
- **Gauge 2:** Mean percentage of the **CORRELATION_DONEISSUES_MERGEDPR** on the selected period in Grafana.
    - **Low range**: [0%, 50%)
    - **Mid range**: [50%,75%)
    - **High range**: [75%, 100%]
- **Timegraph (team)**: Evolution of **CORRELATION_DONEISSUES_MERGEDPR** over time.
    - **Acceptable threshold** greater or equal than 75%
    - **Unacceptable threshold**, lower than 75%
- **Correlation**: Correlation between the metrics **COUNT_DONEISSUES_MERGEDPR** and **NUMBER_GH_MERGEDPR** on the period selected in Grafana.
    - X axis: **NUMBER_GH_MERGEDPR**
    - Y axis: **COUNT_DONEISSUES_MERGEDPR**

TP4: Number of “In Progress” issues (by Member, Hourly)

A developer should only be working on one issue at a time

This TP provides information about team’s members in relation to the number of “In Progress” issues. The correct application of this practice is as follows:

Example

1. The member Alexander does not have any issues “In Progress”.
2. He decides to move an issue that is assigned to him into the "In Progress" column.
3. He wants to take on a new issue, but prior to moving it into "In Progress", he must move the issue that is already "In Progress" into the “In Review” column. Then, he will be able to start a new task.

Metrics:

- $\mathbf{A_{TP4}}$: number of issues in the “In Progress” column by member in the last hour.

Guarantee:

$$COUNT\_INPROGRESSISSUES\_MEMBER \leq 1$$

HOURLY by **MEMBER**

Possible False-Positive:

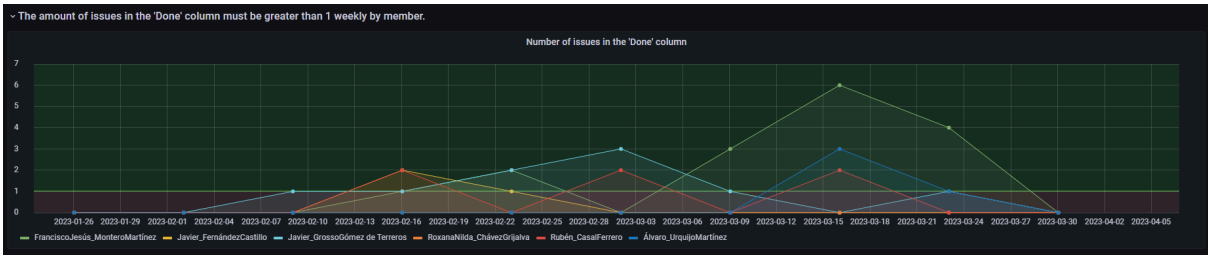
- Sample sequence:

  1) The issue is moved to the “In Progress” column.

  2) The same issue is moved to another column within the same hour.

Dashboard:

Blocks



- **Timegraph (member)**: *Amount of issues in the “In Progress” column* over time by member (**COUNT_INPROGRESSISSUES_MEMBER**).

  - **Acceptable threshold** lower or equal than 1
  - **Unacceptable threshold**, greater than 1

TP5: Number of “Done” issues (by Member, Weekly)

Each developer should be finishing at least 1 issue per week

This TP provides information about team’s members in relation to the number of “Done” issues each week. The correct application of this practice is as follows:

> **Example**
>
> 1. The member Paul has not put any issues into the “Done” column this week yet. So he must finish an issue before the weekends in order to comply with the established team practice.
> 2. Paul has a pull request approved by his team, so he proceeds to move the issue related to that pull request from the “In Review” column to the “Done” column, thus complying with the team practice.

Metrics:

- **COUNT_DONEISSUES_MEMBER**: number of issues in the “Done” column by member in the last week.

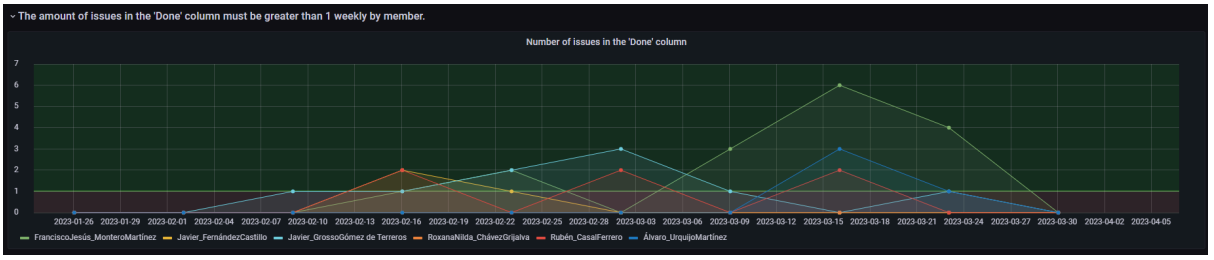
Guarantee:

$$COUNT\_DONEISSUES\_MEMBER \geq 1$$

WEEKLY by **MEMBER**

Dashboard:

Blocks



- **Timegraph (member)**: *Amount of issues in the “Done” column* over time by member (**COUNT_DONEISSUES_MEMBER**).
  - **Acceptable threshold** greater or equal than 1
  - **Unacceptable threshold**, lower than 1

TP6: Percentage of approved mergedPR (by Team, Hourly)

Most merged PRs should be approved

This TP provides information about the team in relation to the reviews on each merged pull request. The correct application of this practice is as follows:

1. A **pull request** is created by a member.
2. A different member **reviews** the pull request with an **approval**.
3. The pull request is **merged**.

> **Example**
>
> John creates a new pull request. Then, Elon reviews the pull request with approval, so that the pull request can now be merged.

Calculation Pre-Requirements (if not met, a point won't be created in the corresponding period):

- Some pull request was merged during the last week.

Metrics:

- **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_TEAM**: number of merged pull requests with at least one positive review by the team in the last week..
- **COUNT_PR_MERGED_TEAM**: number of merged pull requests by the team in the last week.

Guarantee:

$$\frac{COUNT\_MERGED\_PR\_WITH\_POSITIVE\_REVIEWS\_TEAM}{COUNT\_PR\_MERGED\_TEAM} \times 100 \geq 75$$

WEEKLY by **TEAM**

Dashboard:

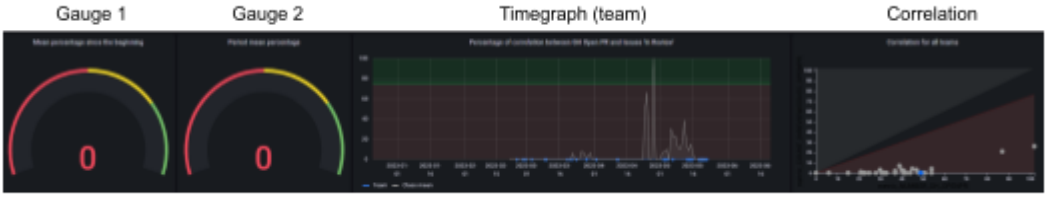
Derived metric

CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_TEAM: Ratio between **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_TEAM** and **COUNT_PR_MERGED_TEAM**

CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_TEAM =

$$\frac{COUNT\_MERGED\_PR\_WITH\_POSITIVE\_REVIEWS\_TEAM}{COUNT\_PR\_MERGED\_TEAM} \times 100$$

Blocks



- **Gauge 1:** Mean percentage of **CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_TEAM** since the beginning of the project.
    - **Low range**: [0%, 50%)
    - **Mid range**: [50%,75%)
    - **High range**: [75%, 100%]
- **Gauge 2:** Mean percentage of the **CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_TEAM** on the selected period in Grafana.
    - **Low range**: [0%, 50%)
    - **Mid range**: [50%,75%)
    - **High range**: [75%, 100%]
- **Timegraph (team)**: Evolution of **CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_TEAM** over time.
    - **Acceptable threshold** greater or equal than 75%
    - **Unacceptable threshold**, lower than 75%
- **Correlation**: Correlation between the metrics **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_TEAM** and **COUNT_PR_MERGED_TEAM** on the period selected in Grafana.
    - X axis: **COUNT_PR_MERGED_TEAM**
    - Y axis: **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_TEAM**

TP7: Percentage of approved mergedPR (By Member, Hourly)

Most of your merged PRs should be approved by your teammates.

This TP provides information about members in relation to the reviews on each merged pull request. The correct application of this practice is as follows:

1. A **pull request** is created by a member.
2. A different member **reviews** the pull request with an **approval**.
3. The pull request is **merged**.

> **Example**
>
> John creates a new pull request. Then, Elon reviews the pull request with approval, so that the pull request can now be merged.

Calculation Pre-Requirements (if not met, a point won't be created in the corresponding period):

- Some pull requests were merged during the last week.

Metrics (for a given member M)

- **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_MEMBER**: number of merged pull requests by member M with at least one positive review during the last week.
- **COUNT_PR_MERGED_MEMBER**: number of merged pull requests by the same member M during the last week.

Guarantee:

$$\frac{COUNT\_MERGED\_PR\_WITH\_POSITIVE\_REVIEWS\_MEMBER}{COUNT\_PR\_MERGED\_MEMBER} \times 100 \geq 75$$

WEEKLY by **MEMBER**

Dashboard:

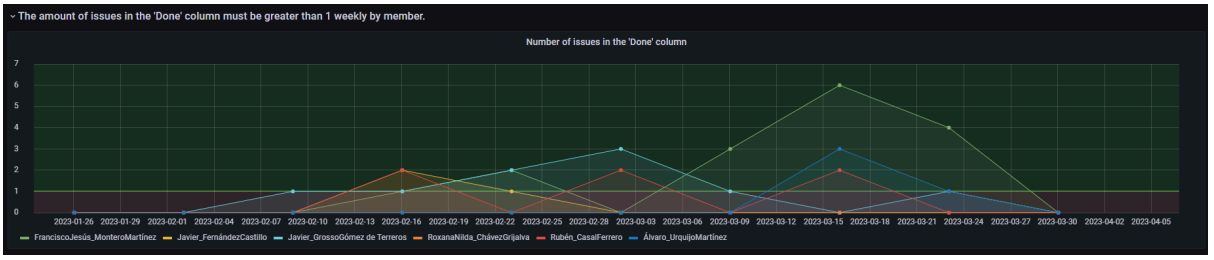
Derived metric

CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_MEMBER: Ratio between **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_MEMBER** and **COUNT_PR_MERGED_MEMBER**

CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_MEMBER=

$$\frac{COUNT\_MERGED\_PR\_WITH\_POSITIVE\_REVIEWS\_MEMBER}{COUNT\_PR\_MERGED\_MEMBER} \times 100$$

Blocks



- **Timegraph (member)**: Evolution of **CORRELATION_APPROVEDMERGEDPULLREQUEST_TOTALMERGEDPULLREQUEST_MEMBER** over time..

- **Acceptable threshold** greater or equal than 75%
- **Unacceptable threshold**, lower than 75%

TP8: Number of approved mergedPR (By Member, Weekly)

If you merge a PR it should be approved by at least 1 teammate

This TP provides information about members in relation to the reviews on each merged pull request. The correct application of this practice is as follows:

1. A **pull request** is created by a member.
2. A different member **reviews** the pull request with **approval**.
3. The pull request is **merged**.

> **Example**
>
> John creates a new pull request. Then, Elon reviews the pull request with approval, so that the pull request can now be merged.

Calculation Pre-Requirements (if not met, a point won't be created in the corresponding period):

- Some pull requests were merged during the last week.

Metrics:

- **COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_MEMBER**: number of merged pull requests by a member with at least one positive review during the last week.

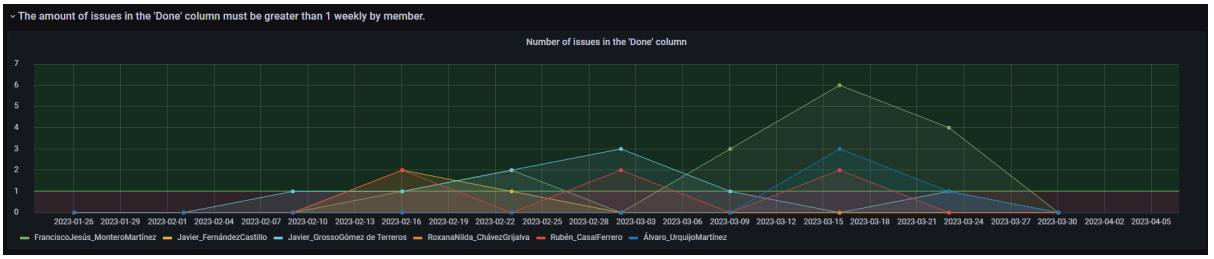
Guarantee:

$$COUNT\_MERGED\_PR\_WITH\_POSITIVE\_REVIEWS\_MEMBER \geq 1$$

WEEKLY by **MEMBER**

Dashboard:

Blocks



- **Timegraph (member)**: Number of *merged pull requests* by a member *with at least one positive review* over time (**COUNT_MERGED_PR_WITH_POSITIVE_REVIEWS_MEMBER**).
    - **Acceptable threshold** greater or equal than 1
    - **Unacceptable threshold**, lower than 1

TP9: Percentage of teammatePRs approved or commented (By Member, Weekly)

You should comment or approve your teammatesPRs

This TP provides information about the team members in relation to the open pull requests with at least one comment. The correct application of this practice is as follows:

1. A **pull request** is created by a member.
2. A different member **comments** on the pull request at least one time every week the PR remains open.

> **Example**
>
> Andrew creates a new pull request during week 3. Then, Logan comments on the pull request in the same week, but the pull request is not closed until week 4, so Logan will need to comment again in week 4 too. That is, if a pull request is open for more than a week, Logan will need to make a comment for each week the pull request stays open.

Calculation Pre-Requirements (if not met, a point won't be created in the corresponding period):

- Some pull requests were open during the last week.

Metrics for a member M:

- **COUNT_PRS_WITH_AT_LEAST_ONE_COMMENT_OR_ONE_REVIEW_COMMENT_BY_MEMBER**: number of open pull requests (not created by member M) with at least one comment by member M during the last week.
- **COUNT_PR**: number of open pull requests not created by member M during the last week.

Guarantee:

$$\frac{COUNT\_PRS\_WITH\_AT\_LEAST\_ONE\_COMMENT\_OR\_ONE\_REVIEW\_COMMENT\_BY\_MEMBER}{COUNT\_PR} \times 100 \geq 20$$

WEEKLY by **MEMBER**

- The proposed threshold (20%) is based on the assumption that teams are composed of ~5 people so each member should do ~⅕ of P/R; with this threshold, member M should make at least 1 comment in one out of five P/R created by other members.

Dashboard:

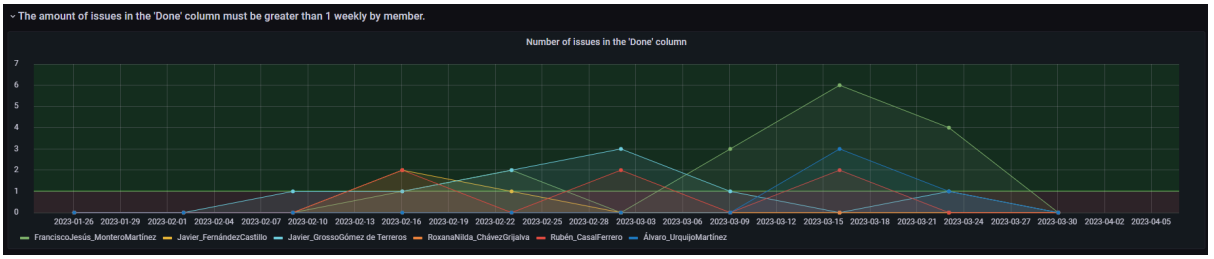
Derived metric

CORRELATION_COUNTPRSWITHATLEASTONECOMMENTORONEREVIEWCOMMENTBYMEMBER_AND_COUNTPR: Ratio between **COUNT_PRS_WITH_AT_LEAST_ONE_COMMENT_OR_ONE_REVIEW_COMMENT_BY_MEMBER** and **COUNT_PR**

CORRELATION_COUNTPRSWITHATLEASTONECOMMENTORONEREVIEWCOMMENTBYMEMBER_AND_COUNTPR =

$$\frac{COUNT\_PRS\_WITH\_AT\_LEAST\_ONE\_COMMENT\_OR\_ONE\_REVIEW\_COMMENT\_BY\_MEMBER}{COUNT\_PR} \times 100$$

Blocks



- **Timegraph (member)**: Evolution of **CORRELATION_COUNTPRSWITHATLEASTONECOMMENTORONEREVIEWCOMMENTBYMEMBER_AND_COUNTPR** over time.
    - **Acceptable threshold** greater or equal than 20%
    - **Unacceptable threshold**, lower than 20%

TP10: Number of teammatePRs approved or commented (By Member, Weekly)

You should comment or approve at least one of your teammatesPRs in a week.

This TP provides information about the team members in relation to the open pull requests with at least one comment. The correct application of this practice is as follows:

1. A **pull request** is created by a member.
2. A different member **comments** on the pull request at least one time every week the PR remains open.

> **Example**
>
> Andrew creates a new pull request during week 3. Then, Logan comments on the pull request in the same week, but the pull request is not closed until week 4, so Logan will need to comment again in week 4. That is, if a pull request is open for more than a week, Logan will need to make a comment for each week the pull request stays open.

Calculation Pre-Requirements (if not met, a point won't be created in the corresponding period):

- Some pull requests were open during the last week.

Metrics:

- **COUNT_PRS_WITH_AT_LEAST_ONE_COMMENT_OR_ONE_REVIEW_COMMENT_BY_MEMBER**: number of pull requests not created by the member with at least one comment by the member during the last week.

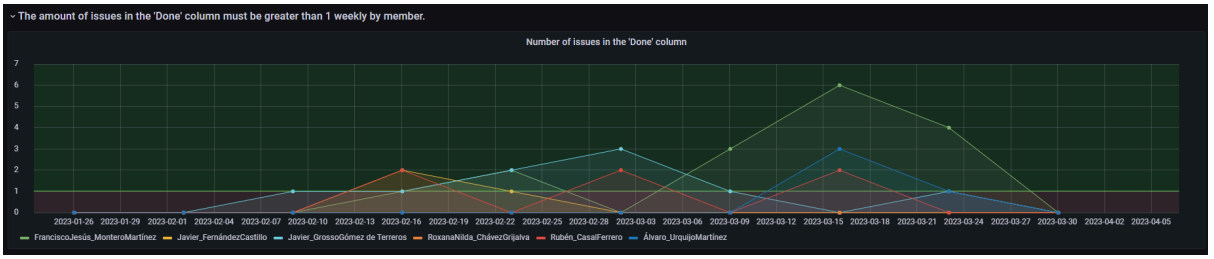
Guarantee:

$$COUNT\_PRS\_WITH\_AT\_LEAST\_ONE\_COMMENT\_OR\_ONE\_REVIEW\_COMMENT\_BY\_MEMBER \geq 1$$

WEEKLY by **MEMBER**

Dashboard:

Blocks



- **Timegraph (member)**: *Number of pull requests not created by the member in which the member has made a comment* over time (**COUNT_PRS_WITH_AT_LEAST_ONE_COMMENT_OR_ONE_REVIEW_COMMENT_BY_MEMBER**) .
  - **Acceptable threshold** greater or equal than 1
  - **Unacceptable threshold**, lower than 1

Support: https://app.gitter.im/#/room/#governify_community:gitter.im