https://github.com/governify/zoo/tree/main/bluejay/tpa/seville/PSG2-2223/v1.0

Changelog

Version	Date	Description	Authors
v0.1	2023-06-19	Initial draft	M. Torrado, J. Fernández
v0.2	2023-06-19	Initial draft Review	M. Torrado, J. Fernández, A. Santisteban, M. Otero, J.M. Garcia, P. Fernandez
v0.3	2023-06-20	Calculation Pre-Requirements Derived metric Blocks images Minor editing fixes	M. Torrado, J. Fernandez, J.M. Garcia, P. Fernandez
v1.0	2024-05-31	Minor editing fixes	J. Fernandez

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:

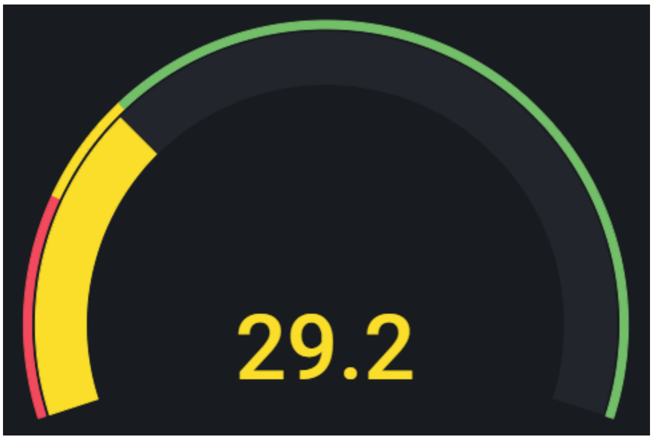
1. A **TPA (Team Practices Agreement)** is made up of **TPs (Team Practices)**.
2. Each **TP** is made up of one **guarantee.**
3. A **guarantee** defines what a work team must comply with to ensure its proper functioning.
4. Each **guarantee** needs an **objective**, which will be using one or more **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.

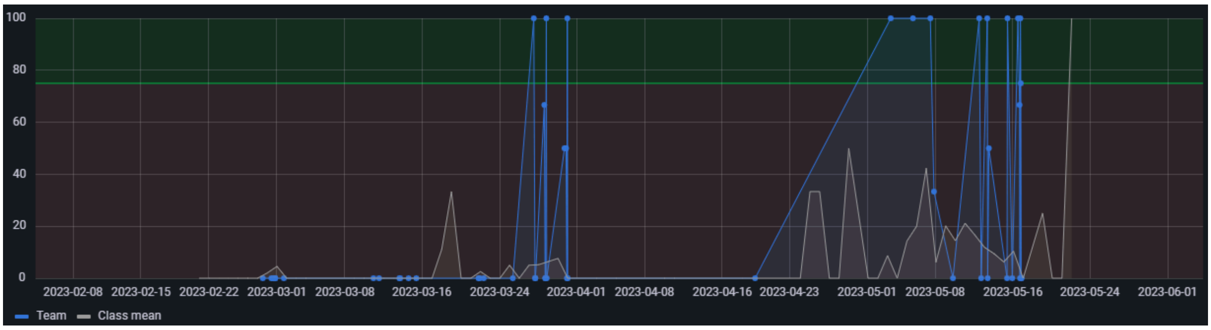
Given all the previous context, our **guarantee's objective** would be "NUMBER_PR_CREATED >= 1" and the **period** would be "weekly".

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.

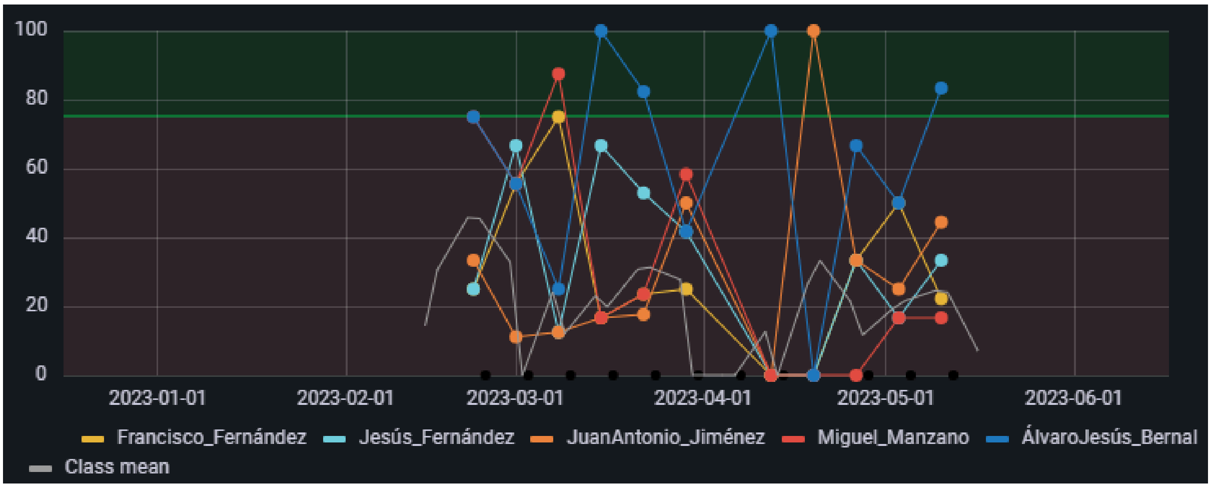


b. **Timegraph by team/member**: Evolution of the value that a guarantee takes over time. This block requires 1 guarantee and can be used for displaying data by team (blue line) or by each member (colored line), both along with the class mean (gray line).

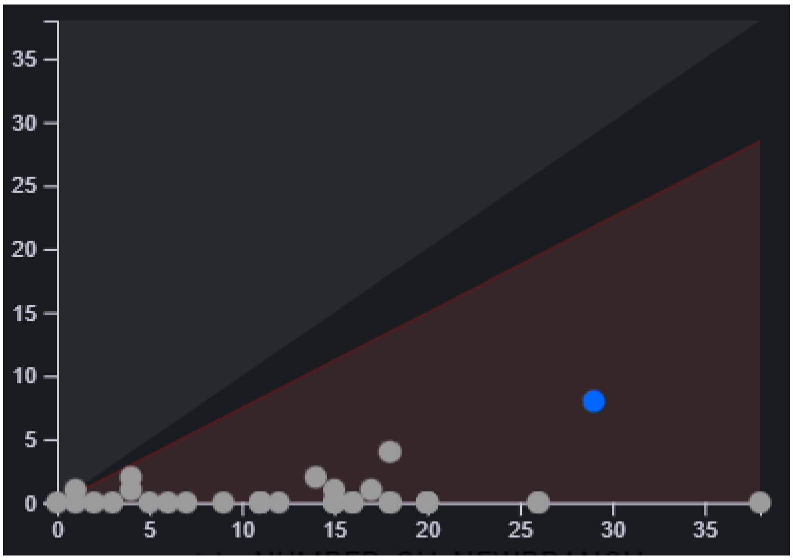
Team



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 correlated with issues moved to “In Progress” hourly by team

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

Caution: This TP calculation must be done in the current time (present). While GithubAPI provides the "updated at" information, it doesn't include the date when the issue transitions to "in Progress". If the TP is calculated in a past period, the results will be inaccurate..

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** in the format **descriptiveName /number** moved to "In Progress" in the title is **created**. (^[A-Za-z][A-Za-z0-9-]*/[0-9]+$)

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**”, “Pets creation **/35**”, 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:

- $\mathbf{A_{TP1}}$: 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.
- $\mathbf{B_{TP1}}$: number of branches created by the team in the last hour.

Guarantee:

$$\frac{A_{TP1}}{B_{TP1}} \times 100 \geq 75$$

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.

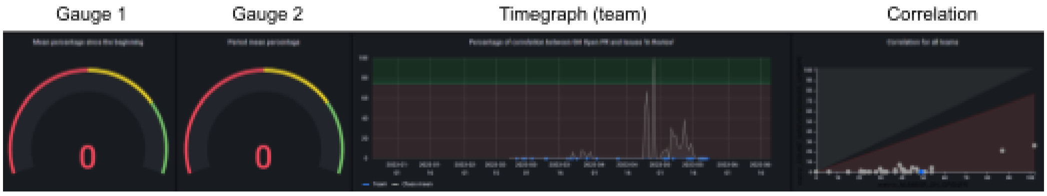
Dashboard:

Derived metric

C_{TP1}: Ratio between A_{TP1} and B_{TP1}

$$\boxed{C_{TP1} = \frac{A_{TP1}}{B_{TP1}} \times 100}$$

Blocks



- **Gauge 1:** Mean percentage of C_{TP1} since the beginning of the project.
  - **Low range**: [0%, 50%)
  - **Mid range**: [50%,75%)
  - **High range**: [75%, 100%]

- **Gauge 2:** Mean percentage of the C_{TP1} on the selected period in Grafana.
  - **Low range**: [0%, 50%)
  - **Mid range**: [50%,75%)
  - **High range**: [75%, 100%]

- **Timegraph (team)**: Evolution of C_{TP1} over time.
  - **Acceptable threshold** greater or equal than 75%
  - **Unacceptable threshold**, lower than 75%

- **Correlation**: Correlation between the metrics A_{TP1} and B_{TP1} on the period selected in Grafana.
  - X axis: B_{TP1}
  - Y axis: A_{TP1}

TP2: Percentage of new pull requests correlated with issues moved to “In Review” hourly by team

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

Caution: This TP calculation must be done in the current time (present). While GithubAPI provides the "updated at" information, it doesn't include the date when the issue transitions to "in Review". If the TP is calculated in a past period, the results will be inaccurate.

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** in the format **/number** moved to "In Review" in the title 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:

- **A_{TP2}**: number of pull requests created including the number of an issue “In Review” by the team in the last hour.
- **B_{TP2}**: number of pull requests created by the team in the last hour.

Guarantee:

$$\frac{A_{TP2}}{B_{TP2}} \times 100 \geq 75$$

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.

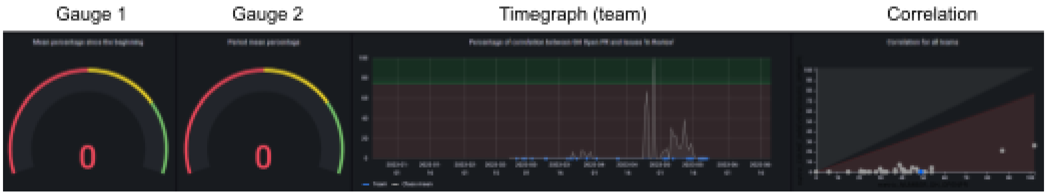
Dashboard:

Derived metric

C_{TP2}: Ratio between **A_{TP2}** and **B_{TP2}**

$$\boxed{C_{TP2} = \frac{A_{TP2}}{B_{TP2}} \times 100}$$

Blocks



- **Gauge 1:** Mean percentage of $\mathbf{C_{TP2}}$ since the beginning of the project.
  - Low range: [0%, 50%)
  - Mid range: [50%,75%)
  - High range: [75%, 100%]
- **Gauge 2:** Mean percentage of the $\mathbf{C_{TP2}}$ on the selected period in Grafana.
  - Low range: [0%, 50%)
  - Mid range: [50%,75%)
  - High range: [75%, 100%]
- **Timegraph (team)**: Evolution of $\mathbf{C_{TP2}}$ over time.
  - Acceptable threshold greater or equal than 75%
  - Unacceptable threshold, lower than 75%
- **Correlation**: Correlation between the metrics $\mathbf{A_{TP2}}$ and $\mathbf{B_{TP2}}$ on the period selected in Grafana.
  - X axis: $\mathbf{B_{TP2}}$
  - Y axis: $\mathbf{A_{TP2}}$

TP3: Percentage of merged pull requests correlated ith issues moved to “Done” hourly by team

An issue being marked "done" should be quickly followed by merging its associated PR

Caution: This TP calculation must be done in the current time (present). While GithubAPI provides the "updated at" information, it doesn't include the date when the issue transitions to "Done". If the TP is calculated in a past period, the results will be inaccurate.

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 Review" column to the "Done" column.
2. A **pull request** that contains the number of the **issue** in the format **/number** moved to "Done" in the tittle is **merged**.

Example

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

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:

- **A_{TP3}**: number of pull requests merged that contain the number of an issue that is in “Done” by the team in the last hour.
- **B_{TP3}**: number of pull requests merged by the team in the last hour.

Guarantee:

$$\frac{A_{TP3}}{B_{TP3}} \times 100 \geq 75$$

HOURLY by **TEAM**

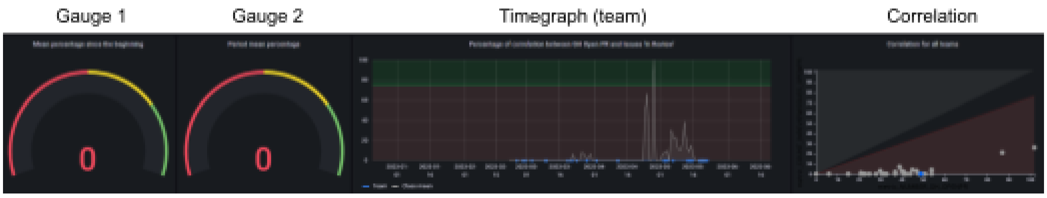
Dashboard:

Derived metric

C_{TP3}: Ratio between **A_{TP3}** and **B_{TP3}**

$$C_{TP3} = \frac{A_{TP3}}{B_{TP3}} \times 100$$

Blocks



- **Gauge 1:** Mean percentage of $\mathbf{C}_{\mathbf{TP3}}$ since the beginning of the project.
  - **Low range**: [0%, 50%)
  - **Mid range**: [50%,75%)
  - **High range**: [75%, 100%]

- **Gauge 2:** Mean percentage of the $\mathbf{C}_{\mathbf{TP3}}$ on the selected period in Grafana.
  - **Low range**: [0%, 50%)
  - **Mid range**: [50%,75%)
  - **High range**: [75%, 100%]

- **Timegraph (team)**: Evolution of $\mathbf{C}_{\mathbf{TP3}}$ over time.
  - **Acceptable threshold** greater or equal than 75%
  - **Unacceptable threshold**, lower than 75%

- **Correlation**: Correlation between the metrics $\mathbf{A}_{\mathbf{TP3}}$ and $\mathbf{B}_{\mathbf{TP3}}$ on the period selected in Grafana.
  - X axis: $\mathbf{B}_{\mathbf{TP3}}$
  - Y axis: $\mathbf{A}_{\mathbf{TP3}}$

TP4: Number of issues “In Progress” hourly by member

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

Caution: This TP calculation must be done in the current time (present). While GithubAPI provides the "updated at" information, it doesn't include the date when the issue transitions to "In Progress". If the TP is calculated in a past period, the results will be inaccurate.

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:

- **A$_{TP4}$**: number of issues in the “In Progress” column by member in the last hour.

Guarantee:

$$A_{TP4} \leq 1$$

HOURLY by **MEMBER**

Possible False-Positive:

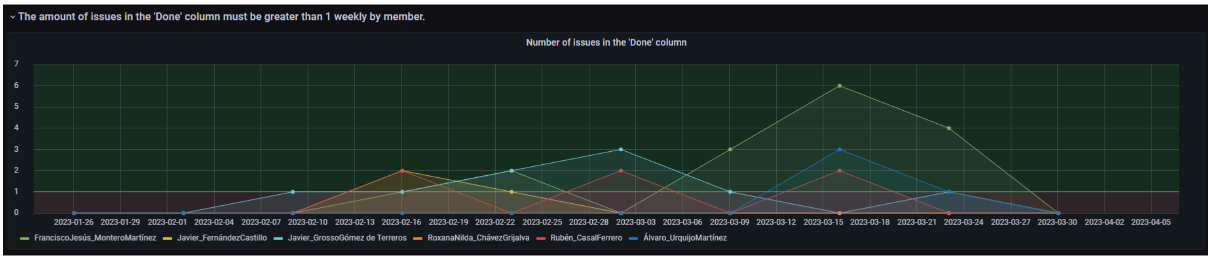
- Sample sequence:

  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 (**A$_{TP4}$**).
  - **Acceptable threshold** lower or equal than 1
  - **Unacceptable threshold**, greater than 1

TP5: Number of issues “Done” weekly by member

Each developer should be finishing at least 1 issue per week

Caution: This TP calculation must be done in the current time (present). While GithubAPI provides the "updated at" information, it doesn't include the date when the issue transitions to "Done". If the TP is calculated in a past period, the results will be inaccurate.

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 week ends 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:

- **A_{TP5}**: number of issues in the “Done” column by member in the last week.

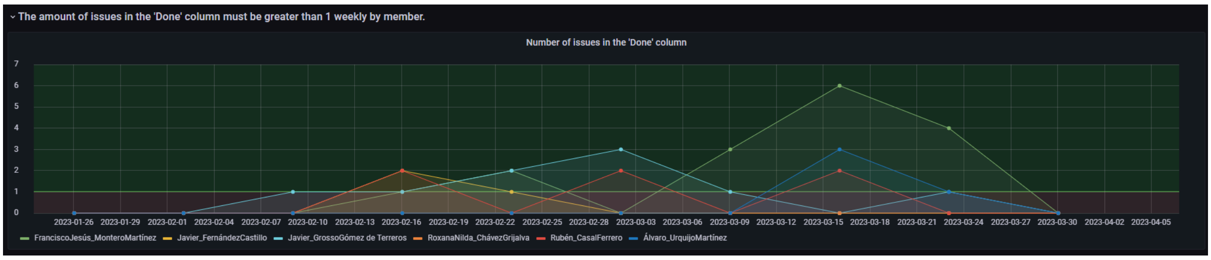
Guarantee:

$$A_{TP5} \geq 1$$

WEEKLY by **MEMBER**

Dashboard:

Blocks



⌄ The amount of issues in the 'Done' column must be greater than 1 weekly by member.

- **Timegraph (member)**: *Amount of issues in the “Done” column* over time by member (**A_{TP5}**).
    - **Acceptable threshold** greater or equal than 1
    - **Unacceptable threshold**, lower than 1

TP6: Percentage of merged pull requests with at least one positive review correlated with total merged pull requests weekly by team

Most PRs should have positive reviews

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 and adds Elon as a reviewer to it. 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:

- **A$_{\mathbf{TP6}}$**: number of merged pull requests with at least one positive review by the team in the last week.
  - Assuming the recurring calculation starts on Monday at 10:00 AM (Director Script launch)
  - E.g. The week is from Monday 8th - to Sunday 14th
  - If the calculation is on Monday 15th at 10:00, this metric returns "the number of P/R with at least one positive review by the team" from Monday 8th 10:00 AM to Monday 15th at 9:59 AM.
- **B$_{\mathbf{TP6}}$**: number of merged pull requests by the team in the last week.

Guarantee:

$$\frac{A_{TP6}}{B_{TP6}} \times 100 \geq 75$$

WEEKLY by **TEAM**

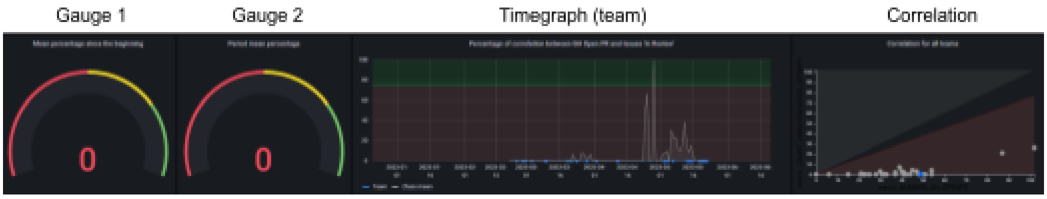
Dashboard:

Derived metric

C$_{\mathbf{TP6:}}$ Ratio between **A$_{\mathbf{TP6}}$** and **B$_{\mathbf{TP6}}$**

$$\mathbf{C_{TP6}} = \frac{A_{TP6}}{B_{TP6}} \times 100$$

Blocks



- **Gauge 1:** Mean percentage of **C_{TP6}** since the beginning of the project.
  - Low range: [0%, 50%)
  - Mid range: [50%,75%)
  - High range: [75%, 100%]

- **Gauge 2:** Mean percentage of the **C_{TP6}** on the selected period in Grafana.
  - Low range: [0%, 50%)
  - Mid range: [50%,75%)
  - High range: [75%, 100%]

- **Timegraph (team)**: Evolution of **C_{TP6}** over time.
  - Acceptable threshold greater or equal than 75%
  - Unacceptable threshold, lower than 75%

- **Correlation**: Correlation between the metrics **A_{TP6}** and **B_{TP6}** on the period selected in Grafana.
  - X axis: **B_{TP6}**
  - Y axis: **A_{TP6}**

TP7: Percentage of merged pull requests by a member with at least one positive review in relation to the total amount of merged pull requests by the same member weekly

Most of your PRs should end up with positive reviews

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 and adds Elon as a reviewer to it. 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 (for a given member M)

- **A_{TP7}**: number of merged pull requests by member M with at least one positive review during the last week.
- **B_{TP7}**: number of merged pull requests by the same member M during the last week.

Guarantee:

$$\frac{A_{TP7}}{B_{TP7}} \times 100 \geq 75$$

WEEKLY by **MEMBER**

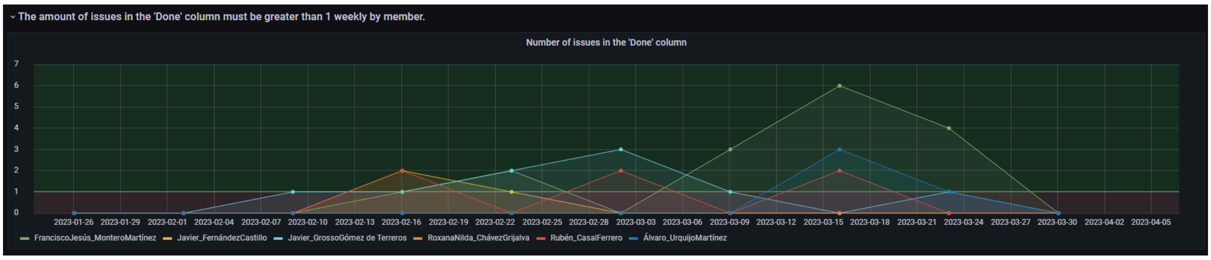
Dashboard:

Derived metric

C_{TP7}: Ratio between **A_{TP7}** and **B_{TP7}**

$$C_{TP7} = \frac{A_{TP7}}{B_{TP7}} \times 100$$

Blocks



The amount of issues in the 'Done' column must be greater than 1 weekly by member.

- **Timegraph (member)**: Evolution of $\mathbf{C_{TP7}}$ over time..

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

TP8: Number of merged pull requests by a member with at least one positive review weekly

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 and adds Elon as a reviewer to it. 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:

- **A$_{TP8}$**: number of merged pull requests by a member with at least one positive review during the last week.

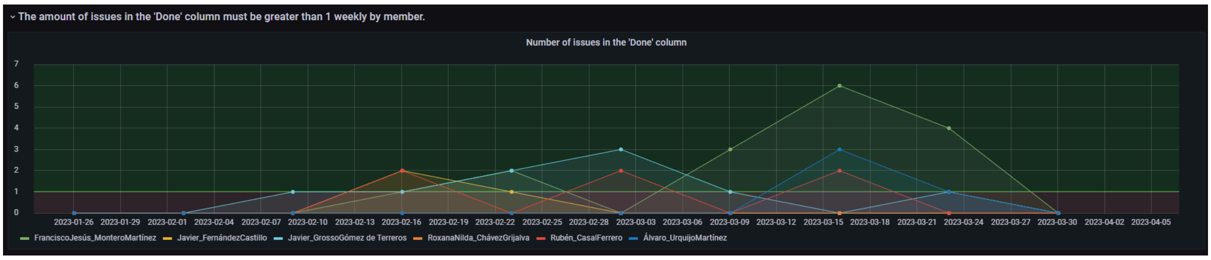
Guarantee:

$$A_{TP8} \geq 1$$

WEEKLY by **MEMBER**

Dashboard:

Blocks



- **Timegraph (member)**: Number of *merged pull requests* by a member *with at least one positive review* over time (**A$_{TP8}$**).
  - **Acceptable threshold** greater or equal than 1
  - **Unacceptable threshold**, lower than 1

TP9: Percentage of open pull requests with comments (including comments from positive reviews) from non-creators in relation to open pull requests from different members, by member weekly

Most PRs should be either active or closed

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 request was open during the last week.

Metrics for a member M:

- $\mathbf{A_{TP9}}$: number of open pull requests (not created by member M) with at least one comment by member M during the last week.
  - If the P/R is closed in the last week, it is still counted in $\mathbf{A_{TP9}}$ (since it was open during a certain part of the week)
  - Assuming the recurring calculation starts on Monday at 10:00AM (Director Script launch)
  - E.g. The week is from Monday 8th - to Sunday 14th
  - If the calculation is on Monday 15th at 10:00, this metric returns "number of open pull requests (not created by member M) with at least one comment by member M" from Monday 8th 10:00AM to Monday 15th at 9:59AM.
  - If a given P/R was open on Monday 8th 10:00AM but closed 1 hour later (11:00AM) it is still counted.
- $\mathbf{B_{TP9}}$: number of open pull requests not created by member M during the last week.

Guarantee:

$$\frac{A_{TP9}}{B_{TP9}} \times 100 \geq 20$$

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.

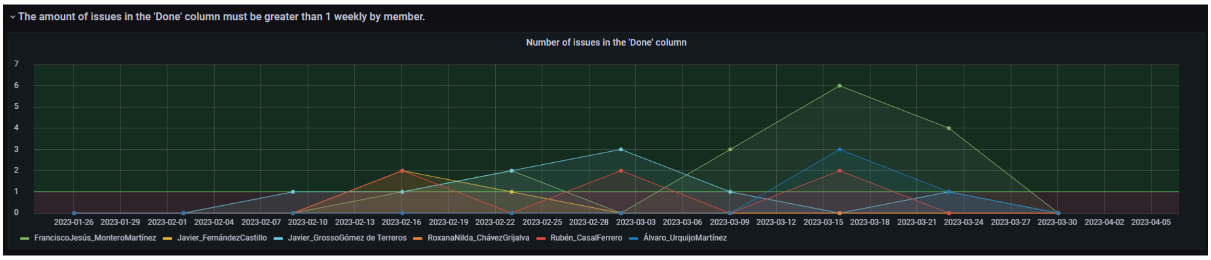
Dashboard:

Derived metric

C_{TP9}: Ratio between **A_{TP9}** and **B_{TP9}**

$$\boxed{\mathbf{C_{TP9}} = \frac{A_{TP9}}{B_{TP9}} \times 100}$$

Blocks



˅ The amount of issues in the 'Done' column must be greater than 1 weekly by member.

- **Timegraph (member)**: Evolution of **C_{TP9}** over time.
  - **Acceptable threshold** greater or equal than 20%
  - **Unacceptable threshold**, lower than 20%

TP10: Number of open pull requests with comments (including comments from positive reviews) from non-creators weekly

Open PRs should be getting regular comments from other team members

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 request was open during the last week.

Metrics:

- **A$_{TP10}$**: number of pull requests not created by the member with at least one comment by the member during the last week.
  - If the P/R is closed in the last week, it is still counted in **A$_{TP10}$** (since it was open during a certain part of the week)
  - Assuming the recurring calculation starts on Monday at 10:00AM (Director Script launch)
  - E.g. The week is from Monday 8th - to Sunday 14th
  - If the calculation is on Monday 15th at 10:00, this metric returns "number of open pull requests (not created by member M) with at least one comment by member M" from Monday 8th 10:00AM to Monday 15th at 9:59AM.
  - If a given P/R was open on Monday 8th 10:00AM but closed 1 hour later (11:00AM) it is still counted.

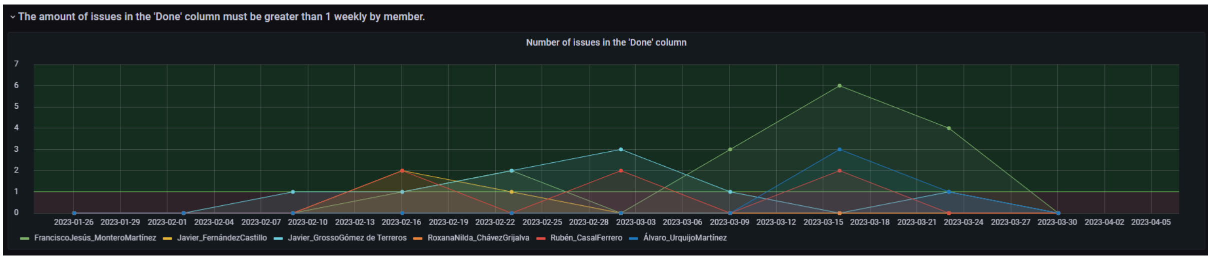
Guarantee:

$$A_{TP10} \geq \ 1$$

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 (**A_{TP10}**).

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