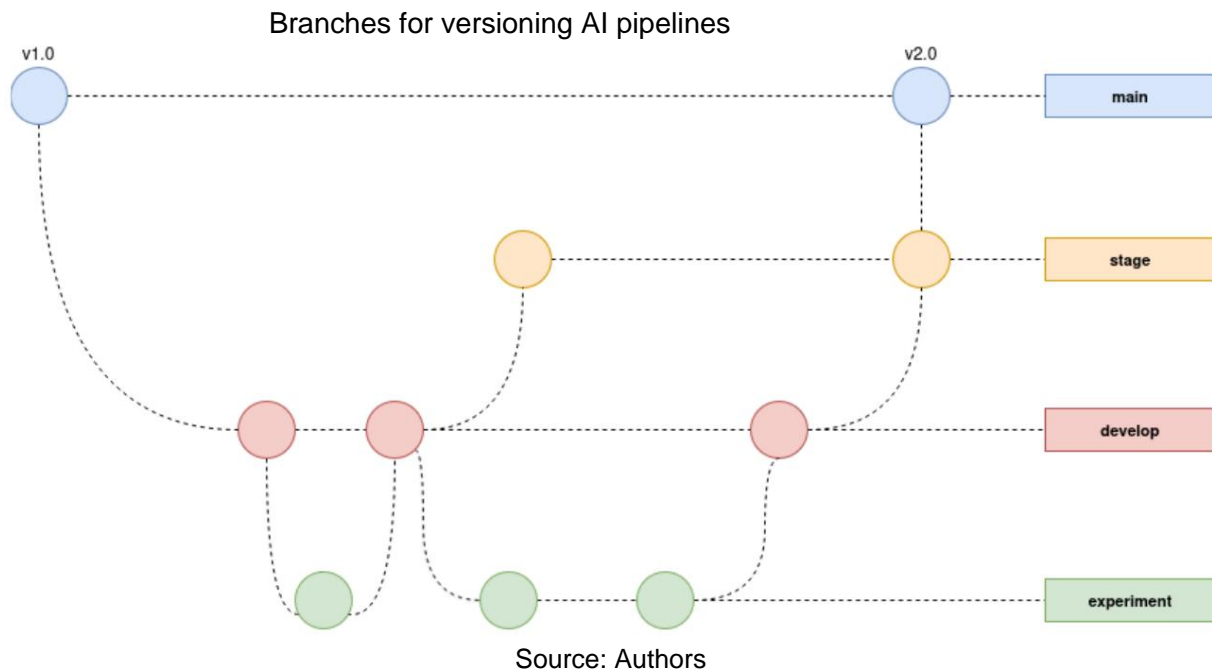


Software Configuration Management Guide for the AI Team

This guide aims to provide a detailed step-by-step guide for the AI team, following the standards and policies defined in the **Software Configuration Management (SCM) document**. The guide uses practical examples to illustrate how tasks should be performed, from opening an **issue** to merging changes into the main **branch**. In addition, the guide covers semantic versioning and the use of **tags** to facilitate the management of AI **pipeline** versions.



1. Opening an Issue

1

Example Situation:

A decrease in the model's accuracy rate when extracting variables from initial petitions was identified. The cause was a change in the format of these documents, which affected the model's accuracy. The AI team needs to improve the model's accuracy to handle the new format.

Step-by-Step Guide to Opening an Issue:

1. Title: •

Example: "Improving the accuracy of the model for extracting variables from initial petitions with a new format."

2. Context: •

Describe the current scenario and the identified problem. • **Example:**

• "Recently, the

format of the initial petitions was changed, which resulted in a decrease in the model's accuracy rate. The extracted variables are being incorrectly identified in

documents with the new format, which impacts the accuracy of the system."

3. Description: •

Explain in detail the necessary change. • **Example:** • "The current model was

trained with a specific format of initial petitions. With the change in format, the model is extracting incorrect information. The proposed solution is to adjust the model to recognize the new format and improve the accuracy in extracting variables."

4. Proposed Solution:

• Describe possible solutions or methods to solve the problem. • **Example:** • "1. Collect a new data set with the updated format.

2. Revise text preprocessing to adapt to the new format.

3. Conduct experiments with different classifiers to improve accuracy."

5. Acceptance Criteria: • List the

requirements that must be met to consider the **issue** completed.

• **Example:** •

"The model must achieve a minimum accuracy of 90% in extracting variables from initial petitions with the new format."

¹ See the "Change Management" section in the Change Management Document. Software Configuration (GCS)

The model response time should not increase significantly."

6. **Additional Information:** • Include

links, references or observations that may assist in development.

• **Example:** ÿ

"Link to current dataset: [link]

Documentation of the classifier used: [link]

Related issue: [link]"

2. Creating Experimental Branches

2

Step-by-Step Guide to Creating Branches:

1. **Create a new branch:**

- From the **develop branch**, create a new **branch** to perform experiments. • **Naming example:**

experiment/123/improvement-accuracy-initial-petition.

2. **Implement the changes:** • In the

created **branch**, make the necessary changes to the code and models. • Make frequent and descriptive **commits**, following the messaging pattern of **commit** defined in GCS. •

Example commit: ÿ feat: adds

new preprocessing for requests
initials

ÿ tune: adjusts hyperparameters of the X classifier.

3. **Open a Pull Request (PR):** • When

the experiment is complete and the results are satisfactory, open a PR to merge the changes into the **develop branch**. • **Example PR title:** "Improving model accuracy for petitions

initials".

- The **PR** must reference the **issue** addressed by carrying out the experiments.

4. **PR Review:**

- Two other team members (not including the PR author) must review the code, provide feedback, and approve the PR. The approval or rejection of the PR must be based on technical comments, detailing the reasons for the approval. • **Approval criteria:** ÿ The model must meet the acceptance

criteria defined in the **issue**. ÿ The

code must be well documented and follow good practices.

development.

² See the "Branching Model" and "Pull Request Management" sections in the Software Configuration Management (SCM) Document

• The new version must not present **drawbacks** (measured based on documented criteria and metrics shared with the entire team) in relation to the pre-existing model in the target **branch**. • If it is a concurrent experiment with others carried out for address the same **issue**, this should present the best measured results. Examinations on this fact should be made explicit in analysis comments.

5. Merge the PR:

- After approval by at least two team members, the PR can be merged into the **develop branch** by any developer on the team.
- **Deployment:** • After merging, the **pipeline** will be orchestrated in the development environment for further testing.

3. Integration into the Stage Branch

Step-by-Step Guide to Integrate into the Stage Branch:

1. Create a PR for the **stage branch**:

- When a set of features or improvements form a planned increment, open a PR to merge the changes from the **develop branch** into the stage **branch**.
- **Example PR title:** "Accuracy Improvement Increment for initial petitions".

2. PR Review:

- Two other team members (not including the PR author) should review the code, provide feedback, and approve the PR. Preferably, at least one of the reviewers should be a teacher, leader, or senior professional on the team. The approval or rejection of the PR should be based on technical comments, detailing the motivations.
- **Approval criteria:** • The model must be stable and compatible with the development environment

approval. • Performance metrics (accuracy, precision, F1) must be satisfactory.

• The new version must not present **drawbacks** (measured using criteria and metrics documented and shared with the entire team) in relation to the pre-existing model in the target **branch**.

3. Merge the PR:

- After approval, if it is not a MAJOR approval, the PR can be merged into the **branch stage**.

³ See the "Branching Model" and "Pull Request Management" sections in Document Software Configuration Management (SCM)

- If it is a **MAJOR** change, the merge should not occur until confirmation is obtained from the affected teams regarding the preparation for the update in the **staging environment**. In this case, this aspect must be explicitly addressed by the developer responsible for implementing the merge, based on a comment in the PR, before performing it. • **Deployment:** The **pipeline** will be orchestrated in the staging environment, where other teams and end users can test the model.
-

4. Integration into the Main Branch

Step-by-Step Guide to Integrate into the Main Branch:

1. Create a PR for the **main branch**:

• After validation in the staging environment, open a PR to merge the changes from the **stage branch** to the main **branch**.

- **Example PR title:** "Petition Accuracy Improvement Release initials".

2. PR Review:

- Two other team members (not including the PR author) must review the code, provide feedback, and approve the PR. The two should preferably be teachers, leaders, or senior professionals on the team. The approval or rejection of the PR should be based on technical comments, detailing the motivations. • **Approval criteria:** The model must be stable and integrated with the rest of the system. There should be no

drawbacks regarding the perception of the

effectiveness of the model, by the stakeholders who tested it in the development **branch**.

- There should be no **drawbacks** that make the model unfeasible to use, or that are unjustifiable for the benefits brought by the improvements or corrections of the increment, such as increased latency or incompatibilities. The new version should not present

drawbacks (measured based on documented criteria and metrics shared with the entire team) in relation to the pre-existing model in the target **branch**.

3. Merge the PR:

- After approval, the PR can be merged into the main **branch**. • It is ideal that the merge is performed in sync with the increment updates from other project teams. Unless the increment is for a specific or emergency correction modification. Or unless the added features will not impact

⁴ See the "Branching Model" and "Pull Request Management" sections in Document Software Configuration Management (SCM)

negatively in the production environment, and its early delivery is relevant to meet some immediate need.

- If it is a MAJOR change , it should not be merged until the other modules in the system are adapted to use the new version. The update must be performed after, or in a synchronized manner with, the adapted version of the other modules (front-end and back-end). In this case, this aspect must be explicitly addressed by the developer responsible for completing the merge, through a comment in the PR, before performing it.

- **Deployment:** ÿ The *pipeline* will be orchestrated in the production environment, synchronized with the increments from other teams (front-end and back-end).

5. Semantic Versioning and Tag Usage

5

Semantic Versioning: 6

- **Format:** MAJOR.MINOR.PATCH
- **Example:** 1.2.3
ÿ 1 (MAJOR): Incompatible API changes. ÿ 2 (MINOR): Added new compatible features. ÿ 3 (PATCH): Bug fixes.

Tag Usage:

- **Tag example:**
 - *classifier-X-v1.2.3*: Identifies the classifier used and the version of *the pipeline*.
 - *dataset-2023-10*: Identifies the version of the dataset used for training.

⁵ See the “Commit Message Management” and “Semantic Versioning” sections in Software Configuration Management (SCM) Document

⁶ Semantic Versioning <<https://semver.org/lang/pt-BR/>>

6. Dataset Changes

⁷

- **Development branch:** If there are changes to the ***datasets*** of the develop branch , if there are experiments running, the experiments must be closed before integrating the new ***datasets***, or the ***datasets*** must be updated in all experiment branches.
- **Data versioning:** Data versioning must be performed within the model ***pipeline*** repository. Tools such as **DVC** must be used to version the ***datasets***.

⁷ See the “Branching Model” section in Software Configuration Management (SCM) Document