

A comprehensive study of issue labeling in GitHub repositories



Joselito Mota Júnior (joselito.mota@ufba.br)

Orientador: Ivan do Carmo Machado

(ivan.machado@ufba.br)



WEPGCOMP 2023



Ficha do trabalho*

Título	ISSUE LABELING IN GITHUB REPOSITORIES
Nome	Joselito Mota Júnior
Curso	Doutorado
Orientador(es)	Ivan do Carmo Machado
Ingresso	2022.1
Qualificação	2024.1
Defesa	2026.1
Bolsista?	Sim

Introduction

- The open-source initiative evolves both at the community, and the software levels (K. Nakakoji et al.)



Introduction

The project repositories should encompass an infrastructure capable of:

- hosting the source code
- enabling its users (developers, testers, etc.) to provide information about the identified issues

Enabling issue reporting



Users could follow the issue life-cycle process.

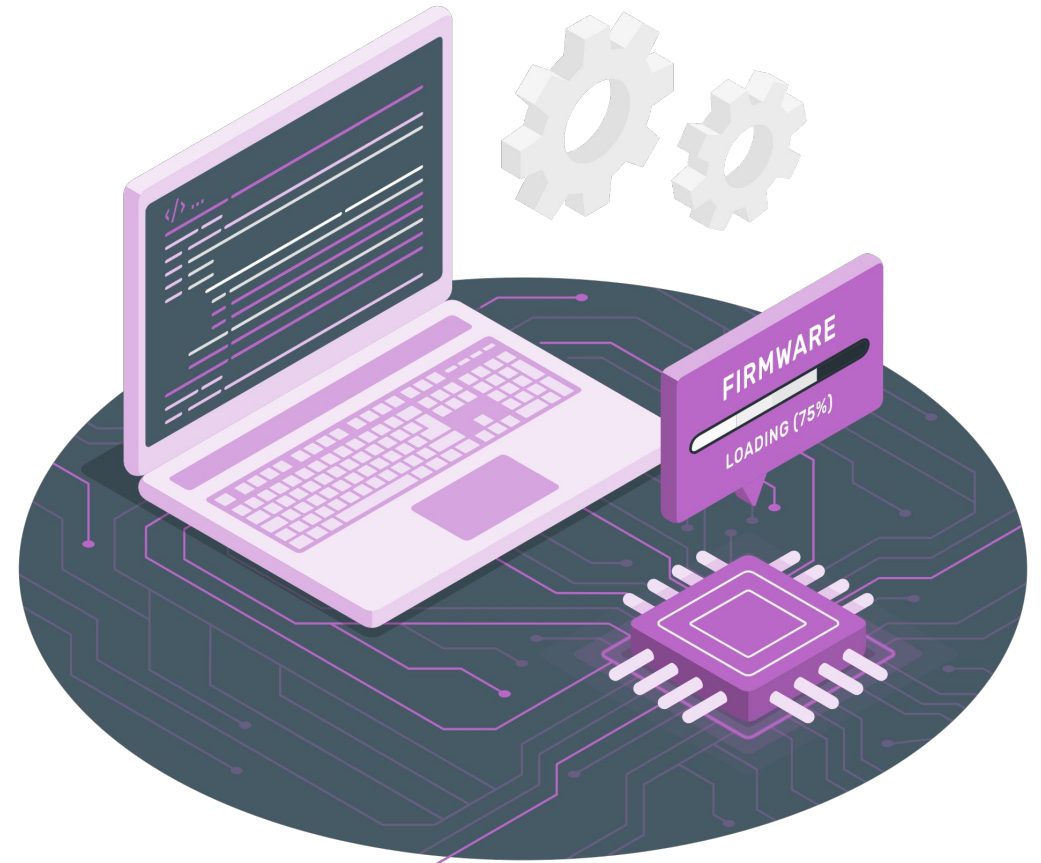


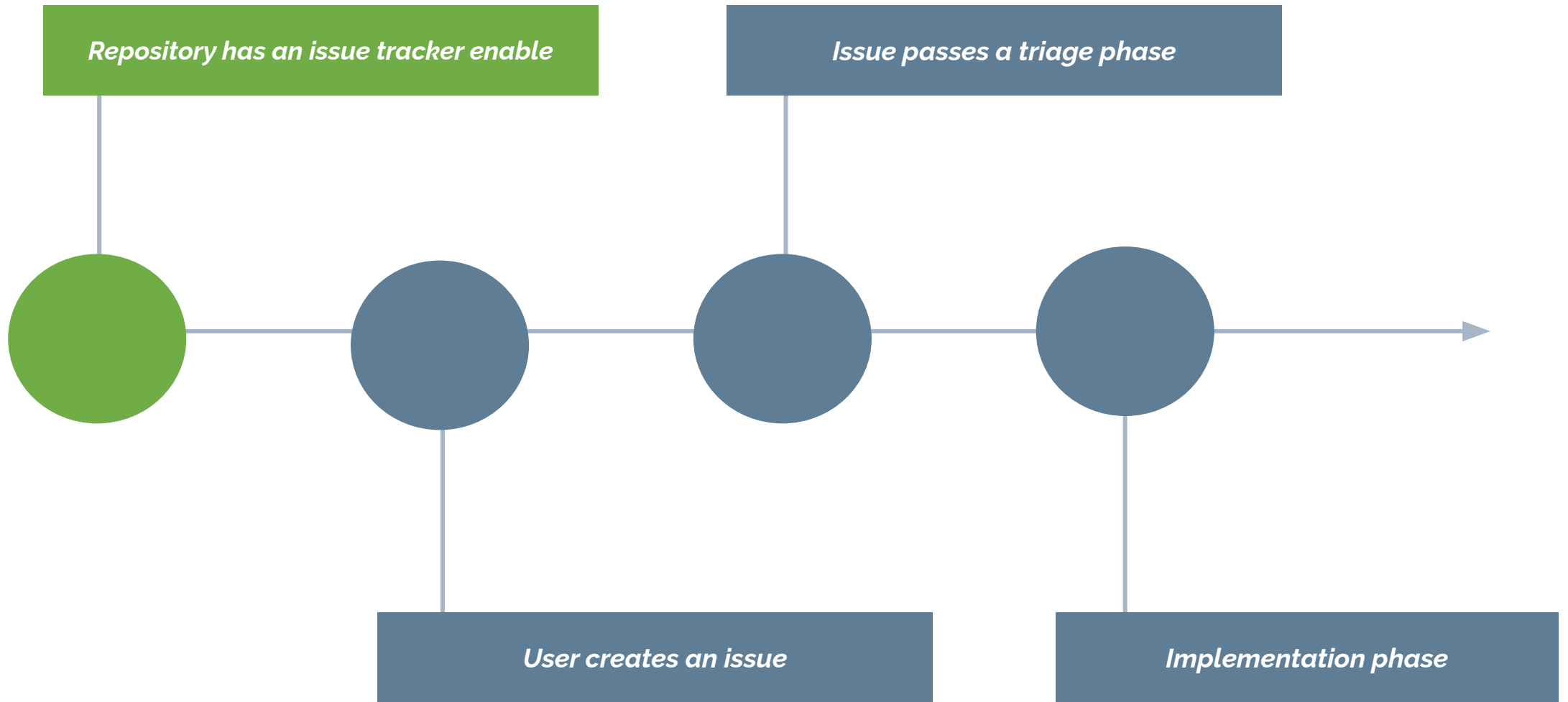
Crucial element for project management

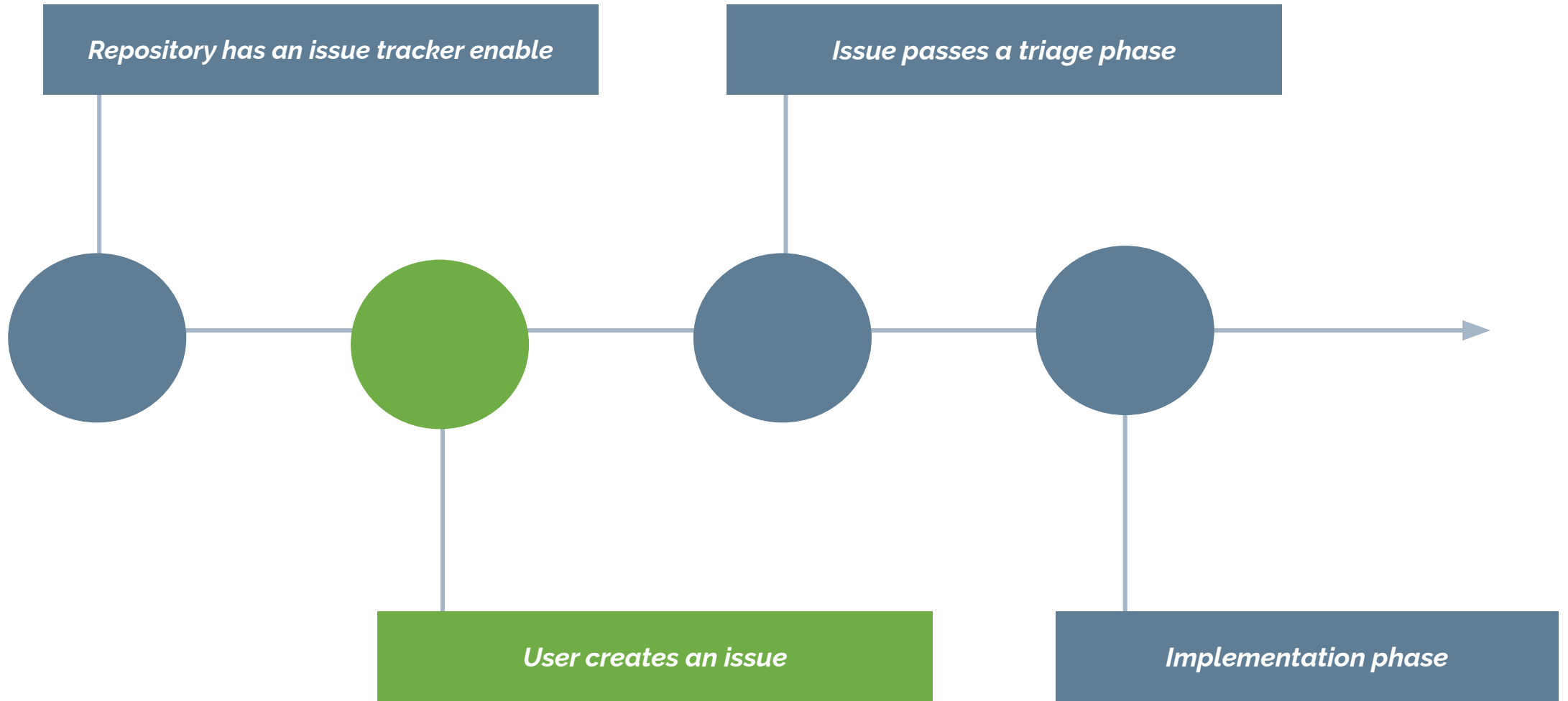


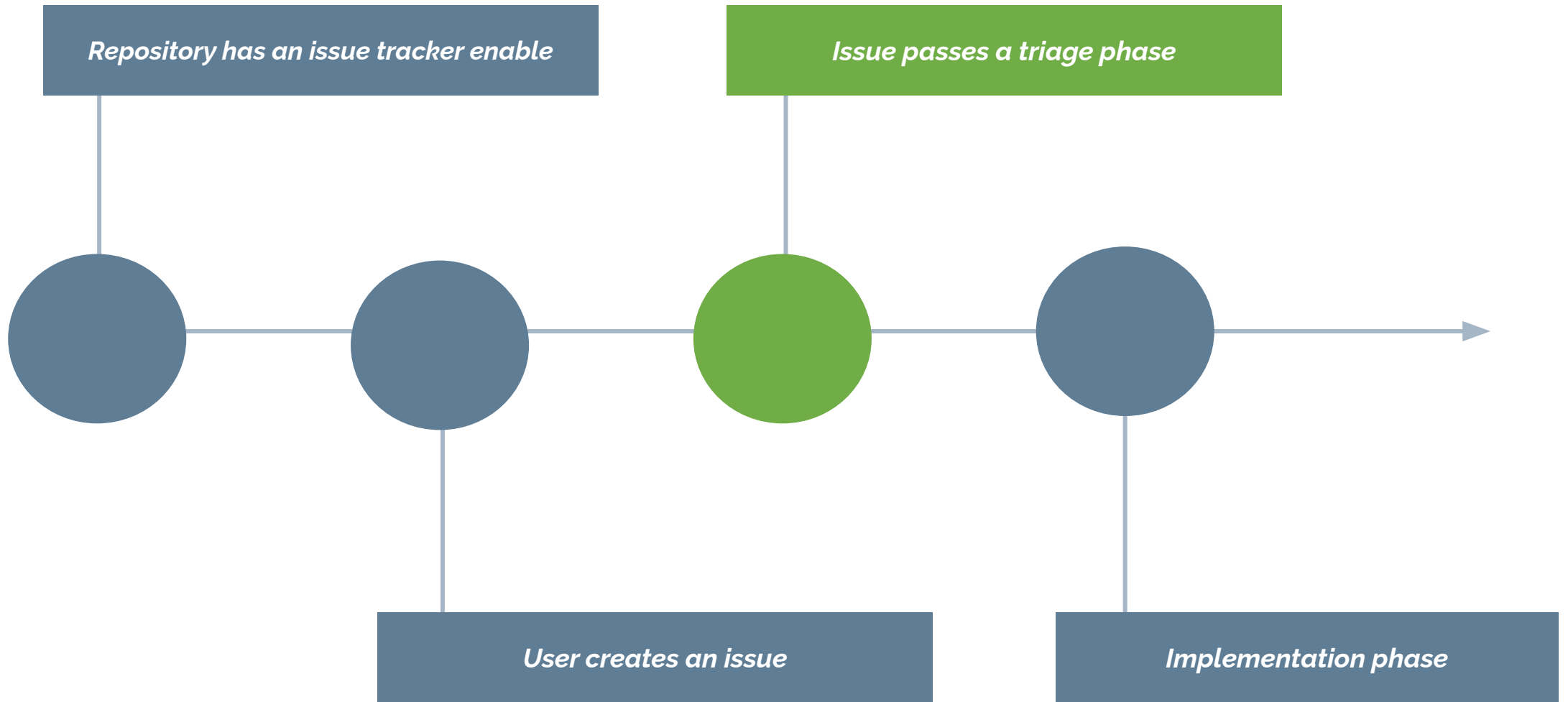
Essential for software maintenance and evolution

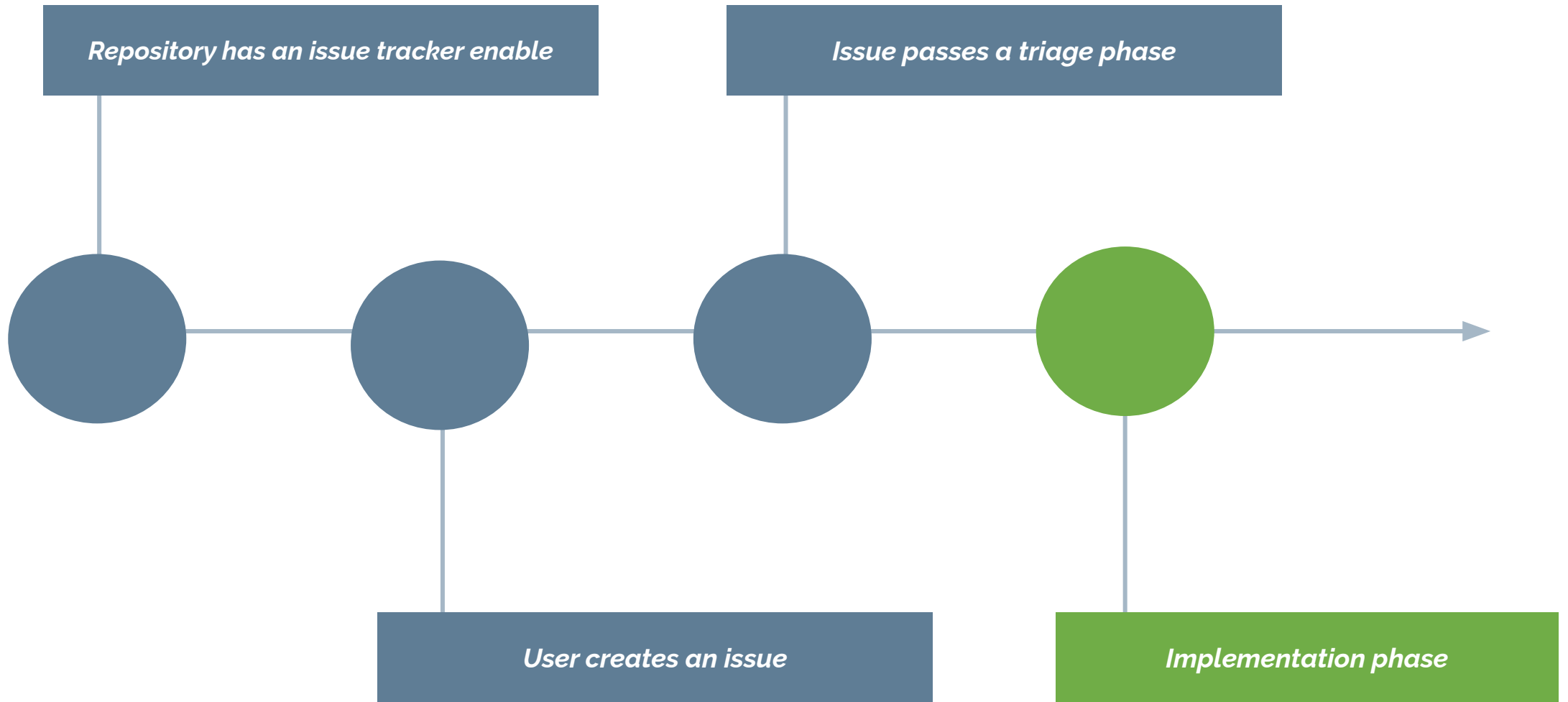
(T. Koponen et al.)











Possible typo in the instruction of Cat Photo App Step 42 #47055

New issue

Open sidemt opened this issue 6 hours ago · 3 comments



sidemt commented 6 hours ago

Contributor

Describe the Issue

At the end of the instruction, it says:

Go ahead and do this to specify where this button should submit the form.

But this step is just asking the camper to add `type="submit"` to the button element, and not asking them to set the destination where the form should be sent.

Affected Page

<https://www.freecodecamp.org/learn/2022/responsive-web-design/learn-html-by-building-a-cat-photo-app/step-42>

Your code

```
<button type="submit">Submit</button>
```

Expected behavior

Body

Title

Labels

Assignees

No one assigned

Labels

- scope: curriculum
- status: waiting triage
- type: bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants





Problem

The issue **should include high-quality information**

Help developers **understand** the **cause** of the
reported problem

Problem

An **incomplete** issue **report** is a rather **severe problem**
a developer could come across in software projects

(N. Bettenburg et al., 2008) (J. Aranda and G. Venolia, 2009) (N. Bettenburg et al., 2007)

In addition to using the **issue structure** information of **title** and **description**



Repositories **enabled** the **usage** of **labels** to **incorporate** more contextual **information** to the issue

Labels creation

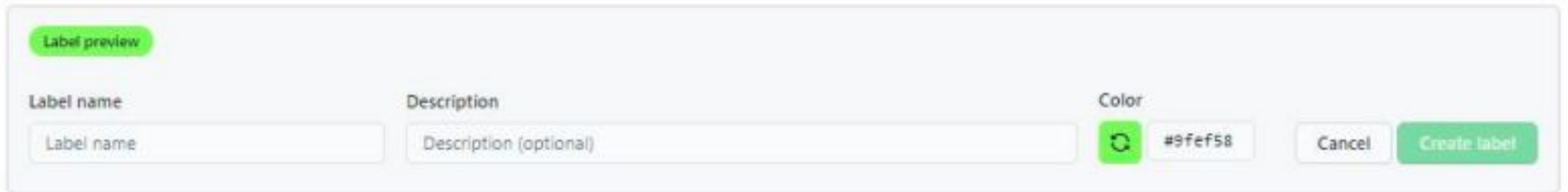
Standard labels **X** Custom labels

Label	Description
bug	Something isn't working.
documentation	Improvements or additions to documentation.
duplicate	This issue or pull request already exists.
enhancement	New feature or request.
good first issue	Good for newcomers.
help wanted	Extra attention is needed.
invalid	This doesn't seem right.
question	Further information is requested.
wontfix	This will not be worked on.

Table 1 GitHub default labels.

Labels creation

Standard labels **X** Custom labels



The screenshot shows the GitHub label creation form. At the top left, there is a green pill-shaped button labeled "Label preview". Below this, the form is organized into three main sections: "Label name", "Description", and "Color". The "Label name" section contains a text input field with the placeholder text "Label name". The "Description" section contains a text input field with the placeholder text "Description (optional)". The "Color" section features a green circular icon with a refresh symbol, followed by a text input field containing the hex color code "#9fef58". To the right of these sections are two buttons: a light gray "Cancel" button and a green "Create label" button.

Fig. 2 The GitHub label creation form.

The main contribution has already been achieved.

1. A database of **10,673,459 issues** of the **180 GitHub** featured **topics** from **13,280 repositories**.
2. A study about issue labeling with a focus on the maintainability and evolution processes
3. A curated list of labels related to triage, life cycle, and defect detection processes
4. A list of label creation trends from the most featured repositories

Research objectives (Objetivo(s) da pesquisa*)

The **impacts** of **labels related** to the **maintainability** and **evolution** of **GitHub** issue trackers

RQ1: How often developers use labels in their open-source repositories?

RQ2: Are there occurrences of labels to assist developers in evolution and maintenance activities of triage, issue life cycle?

RQ3: Can the triage time be derived? from the labeling of practice-related labels?

Preliminary results

RQ1: How often developers use labels in their open-source repositories?

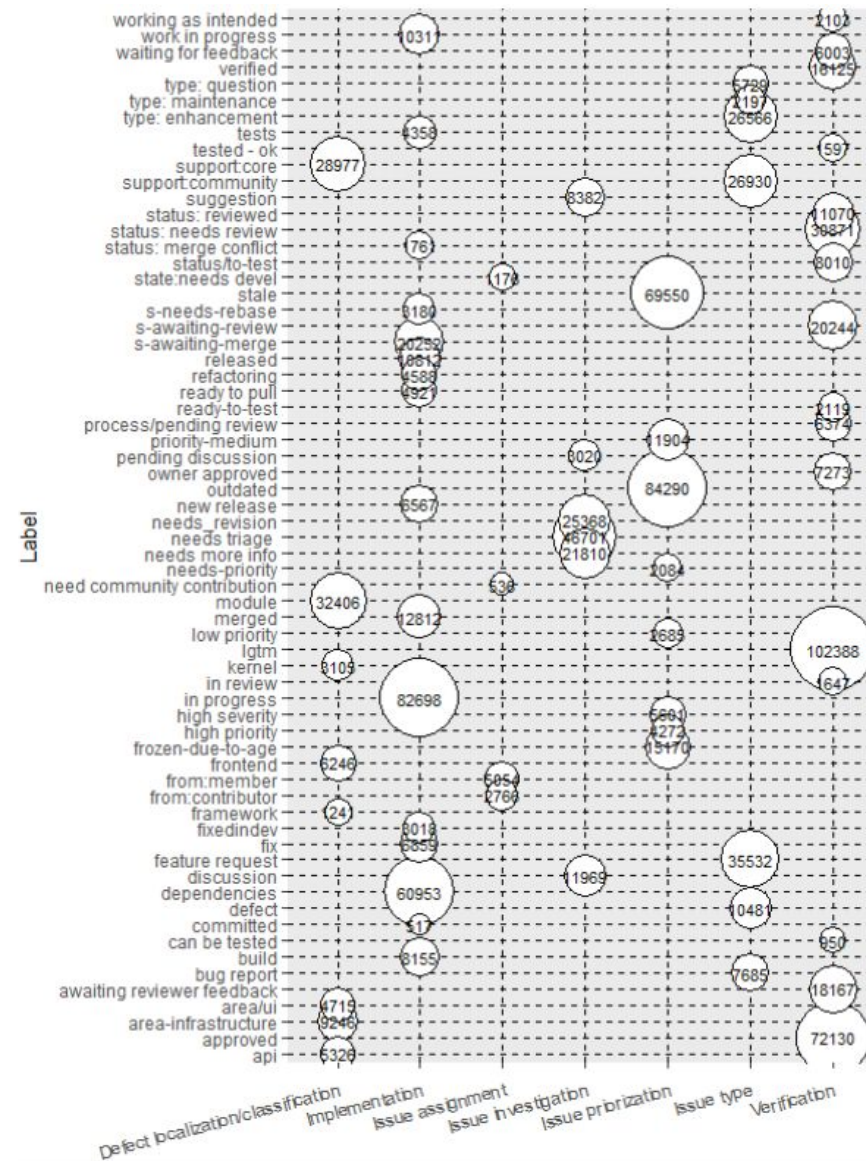
We observed that 10,458 (78.75%) labeled their issues, while 2,822 (21.25%) did not label any issues

66.48% (10,127,463 subscribers) in labeled issues and 33.52% (5,106,845 subscribers) of unlabeled issues

81.72% (1,736,294 assigns) in labeled issue and 18.27% (388,224 assigns) in unlabeled issues.

Preliminary results (Resultados parciais)*

RQ2: Can we define the triage phase from the labeling practice?



Preliminary results (Resultados parciais)*

RQ3: Can the triage time be derived from the labeling of practice-related labels?

Labeling time is concentrated in a shorter time of days

The vast majority of the triage performed by the labels takes little time:

From 1 to 100 days to complete.

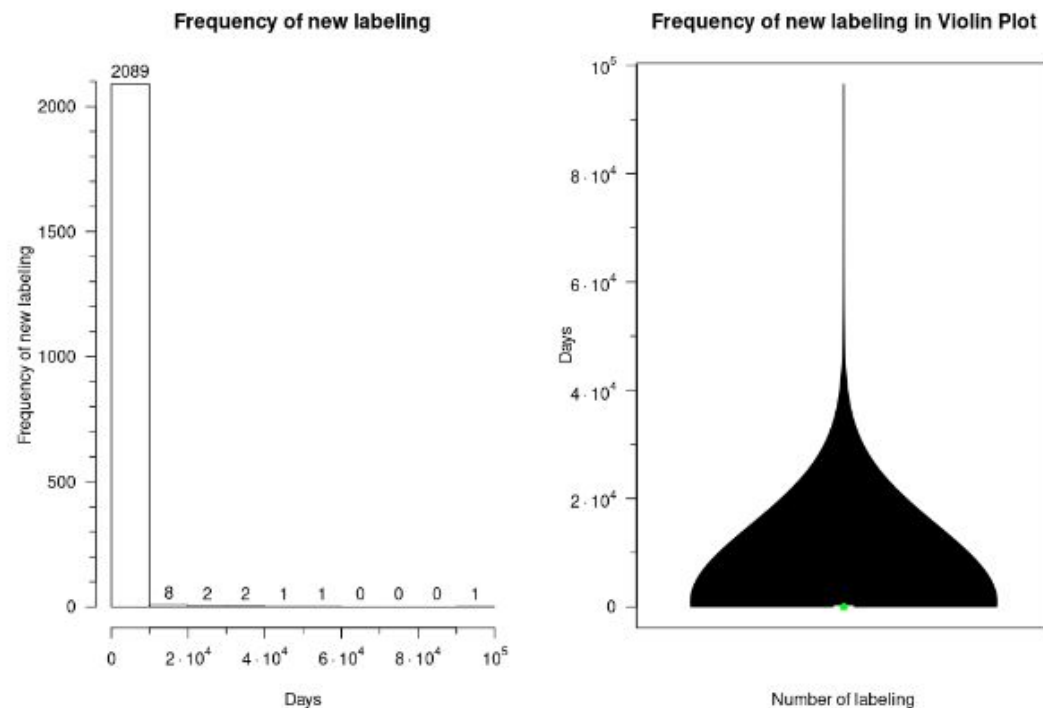


Fig. 15 Histogram and Violin Plot of labeling frequency versus days with no outliers

Future work (Estágio atual*)

Time between two labeling events

To what extent labels and issues components (title, body and comments) are related in terms of context and meaning in the issue conversation?

Automated labeling

Referências bibliográficas citadas nesta apresentação

1. Modern Software Engineering: Doing What Works to Build Better Software Faster. Addison-Wesley Professional, 1st edition, 2021.
2. J. M. Alonso-Abad, C. López-Nozal, J. M. Maudes-Raedo, and R. Marticorena-Sánchez. Label prediction on issue tracking systems using text mining. *Progress in Artificial Intelligence*, 8(3):325–342, 2019.
3. J. Anvik, L. Hiew, and G. C. Murphy. Coping with an open bug repository. In *Proceedings of the 2005 OOPSLA Workshop on Eclipse Technology EXchange, eclipse '05*, page 35–39, New York, NY, USA, 2005. ACM.
4. J. Anvik, L. Hiew, and G. C. Murphy. Who should fix this bug? In *Proceedings of the 28th International Conference on Software Engineering, ICSE '06*, page 361–370, New York, NY, USA, 2006. ACM.
5. J. Aranda and G. Venolia. The secret life of bugs: Going past the errors and omissions in software repositories. In *2009 IEEE 31st International Conference on Software Engineering*, pages 298–308, 2009.

Referências bibliográficas citadas nesta apresentação

1. M. L. Bernardi, G. Canfora, G. A. D. Lucca, M. D. Penta, and D. Distanto. Do developers introduce bugs when they do not communicate? the case of eclipse and mozilla. In 2012 16th European Conference on Software Maintenance and Reengineering, CSMR '12, pages 139–148, USA, 2012. IEEE Computer Society
2. N. Bettenburg, S. Just, A. Schr öter, C. Weiß, R. Premraj, and T. Zimmermann. Quality of bug reports in eclipse. In Proceedings of the 2007 OOPSLA Workshop on Eclipse Technology EXchange, eclipse '07, page 21–25, New York, NY, USA, 2007. ACM
3. N. Bettenburg, S. Just, A. Schr öter, C. Weiss, R. Premraj, and T. Zimmermann. What makes a good bug report? In Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, SIGSOFT '08/FSE-16, page 308–318, New York, NY, USA, 2008. ACM