

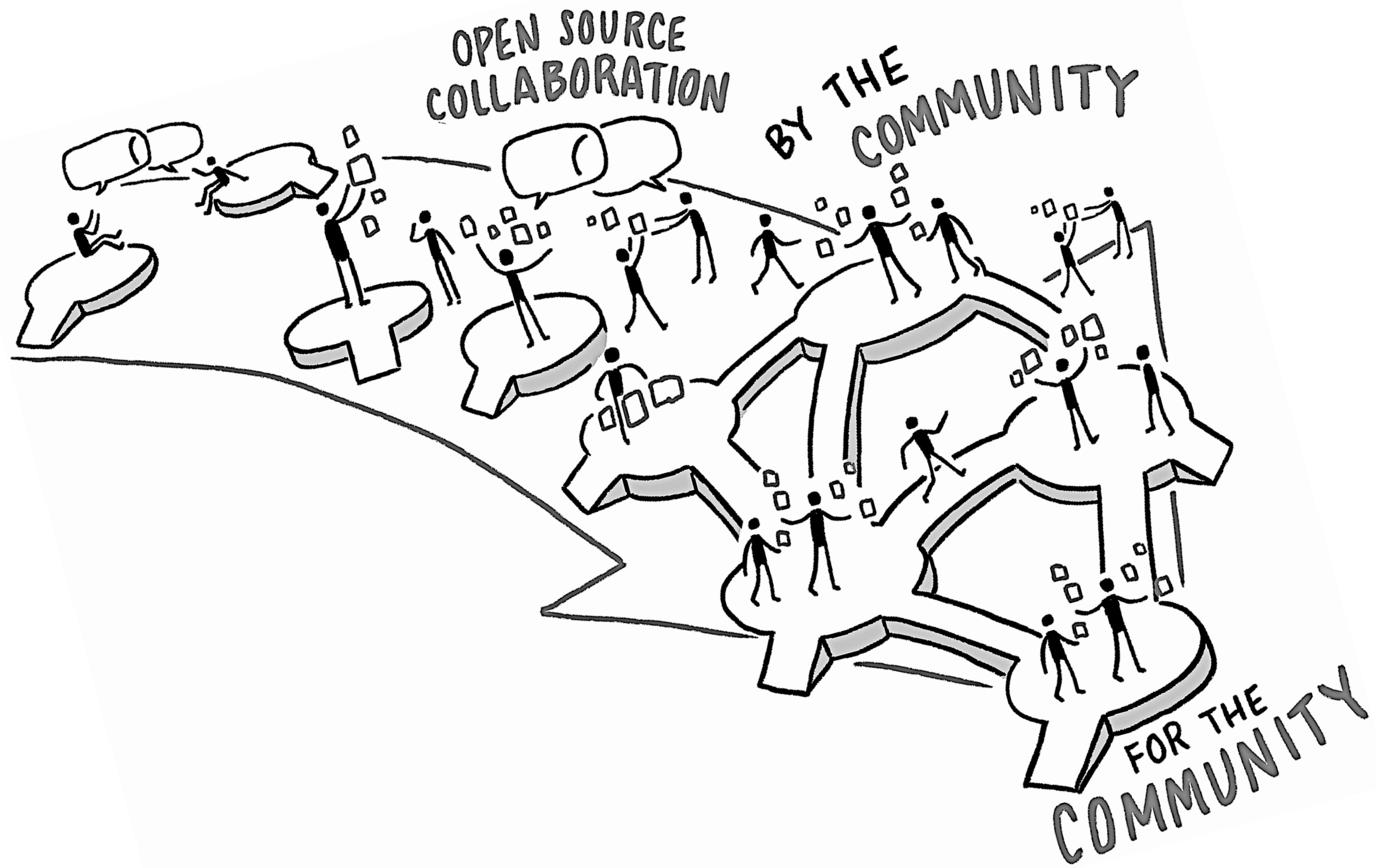
Pull Request Abandonment in Open-Source Projects

SayedHassan Khatoonabadi

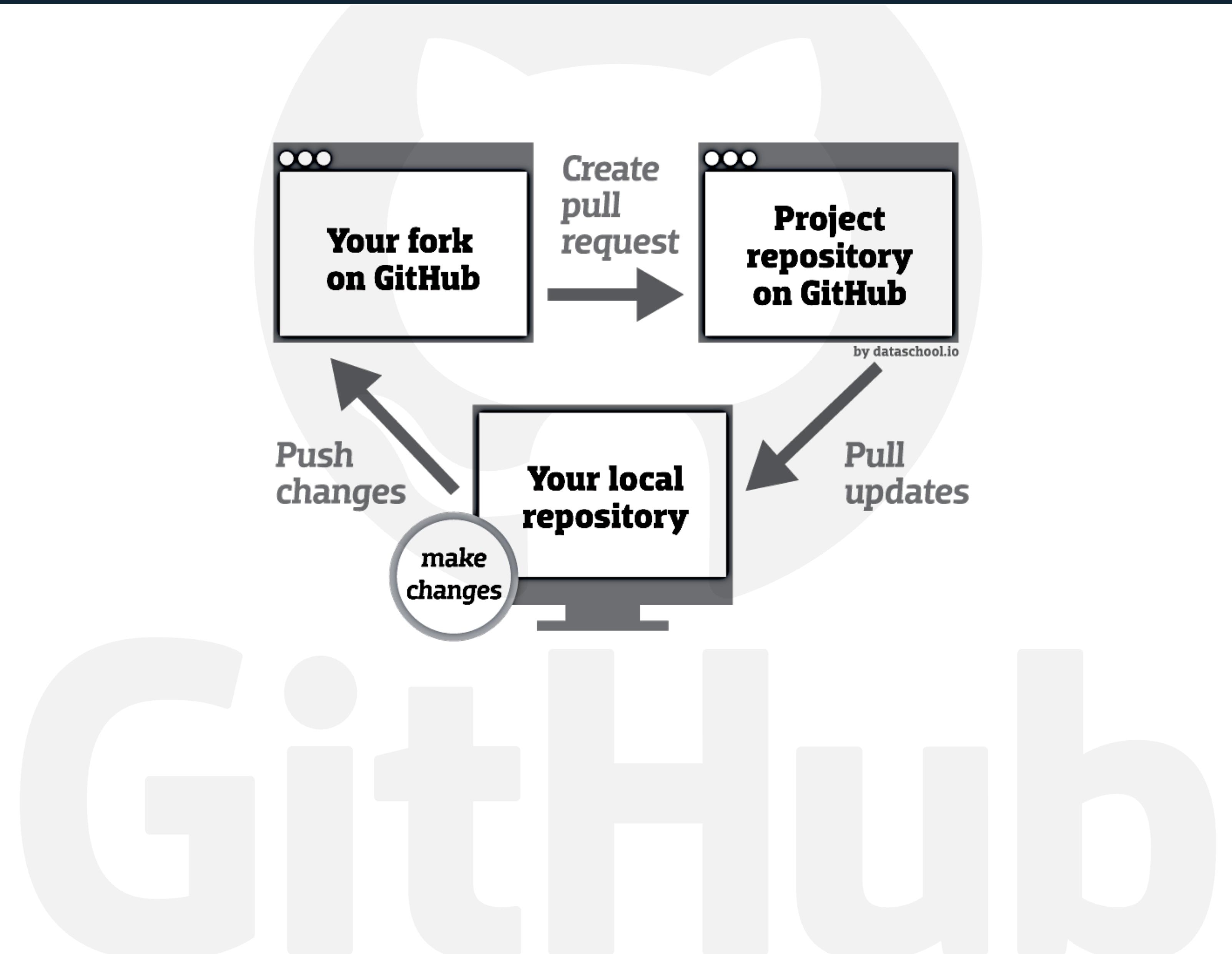
A Doctoral Thesis
in
the Department
of
Computer Science and Software Engineering

Supervised by
Dr. Emad Shihab

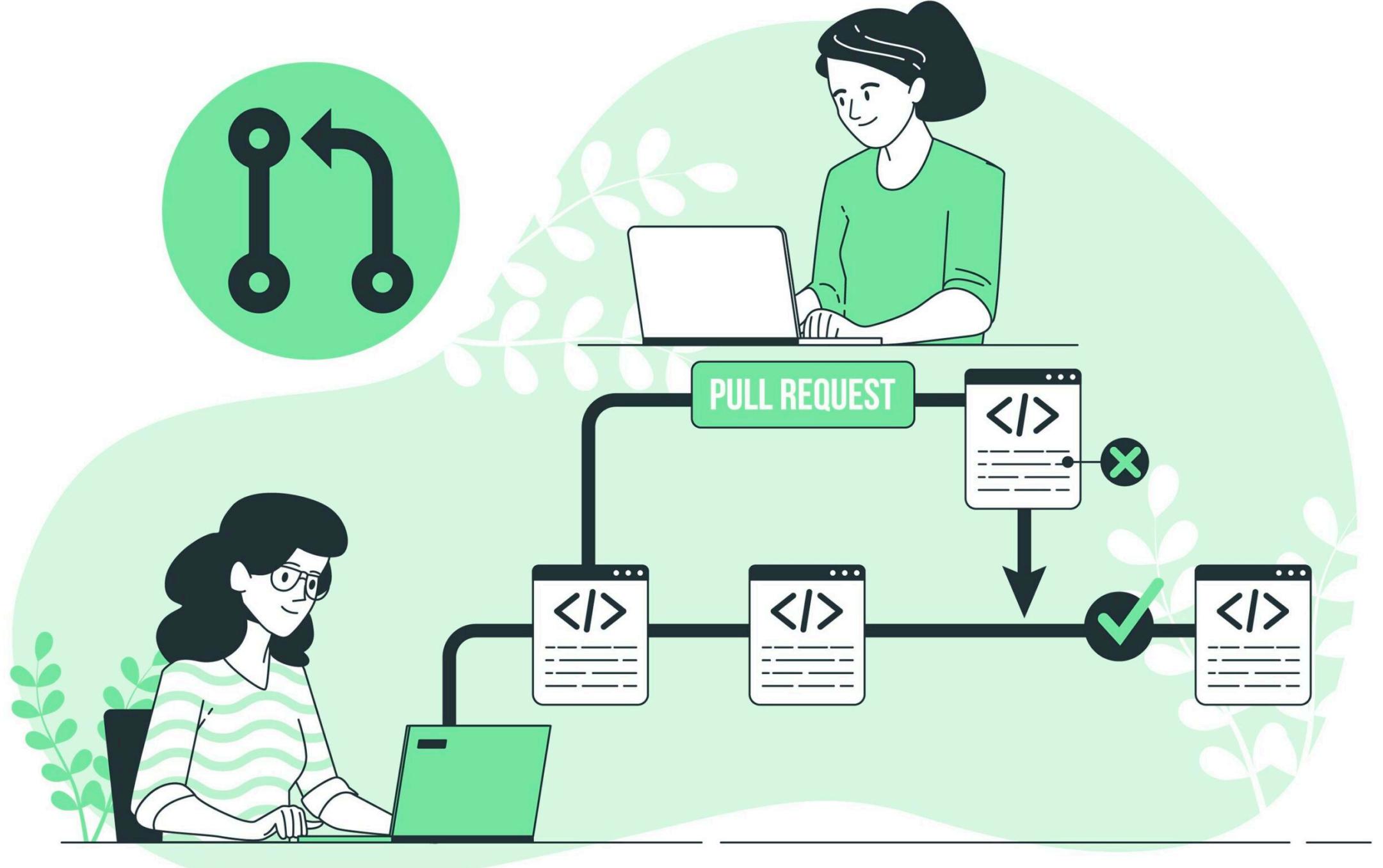
OSS Projects **Thrive** on Community Contributions



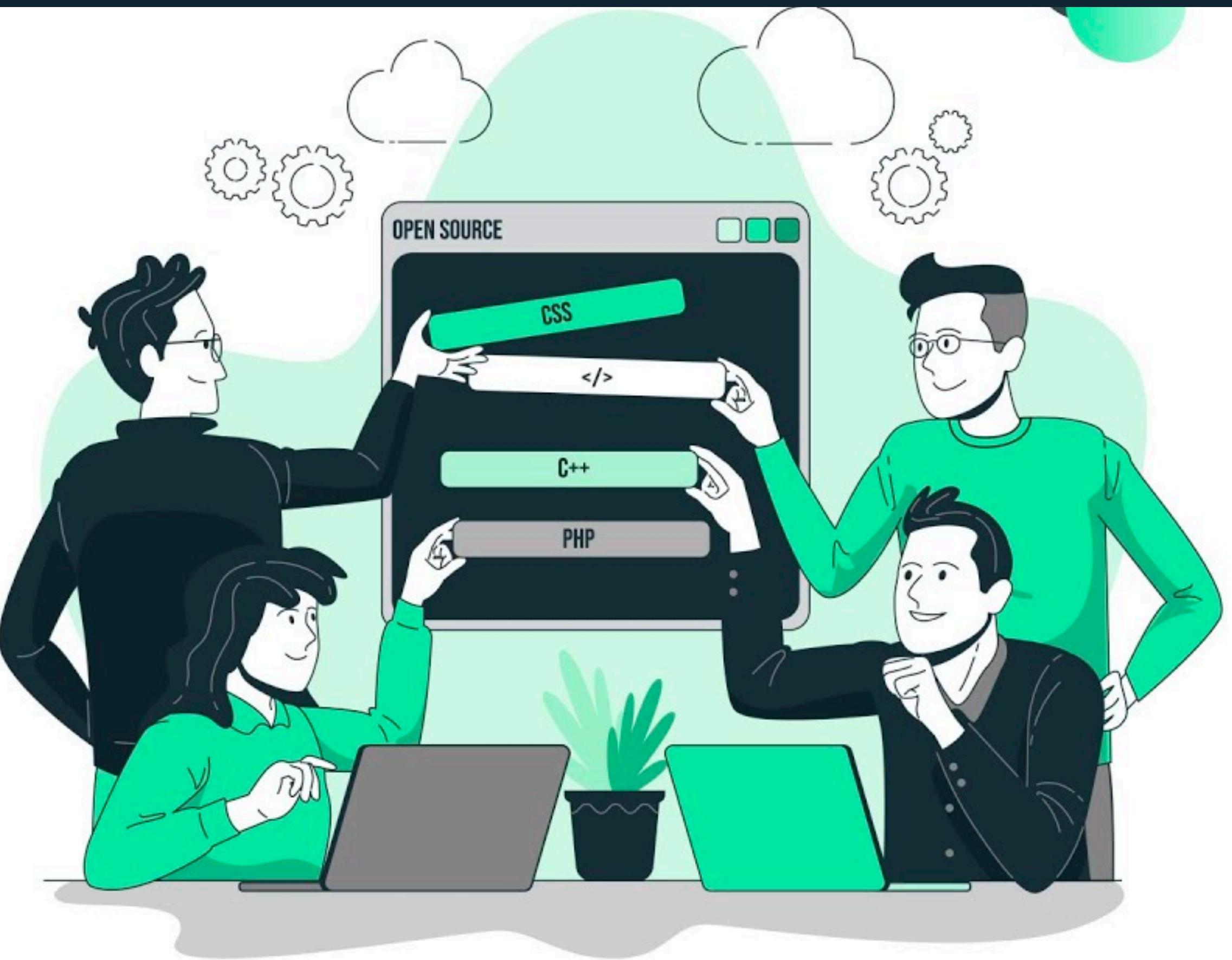
Pull Request (PR) Is **Widely Used** for Contributing and Reviewing Changes in OSS Projects



Pull Request (PR) Is **Widely Used** for Contributing and Reviewing Changes in OSS Projects

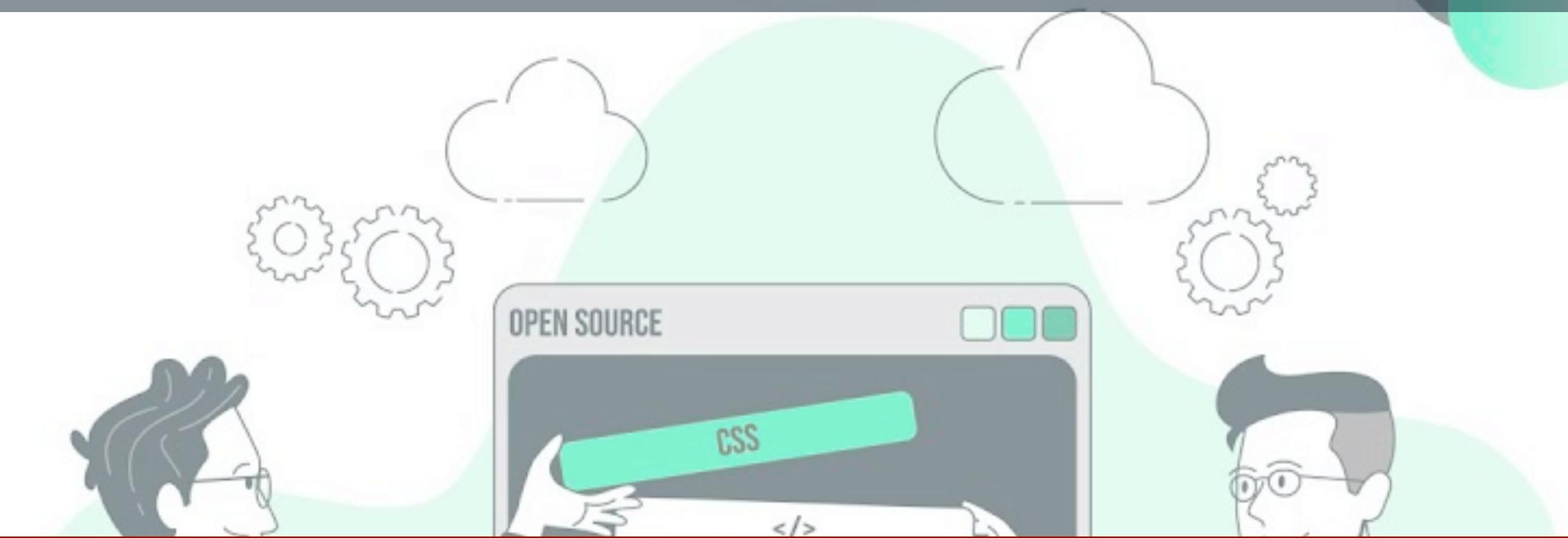
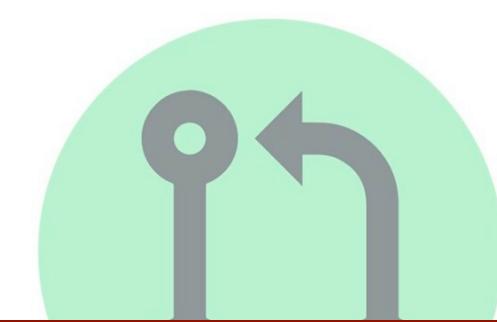


Streamlined Mechanism

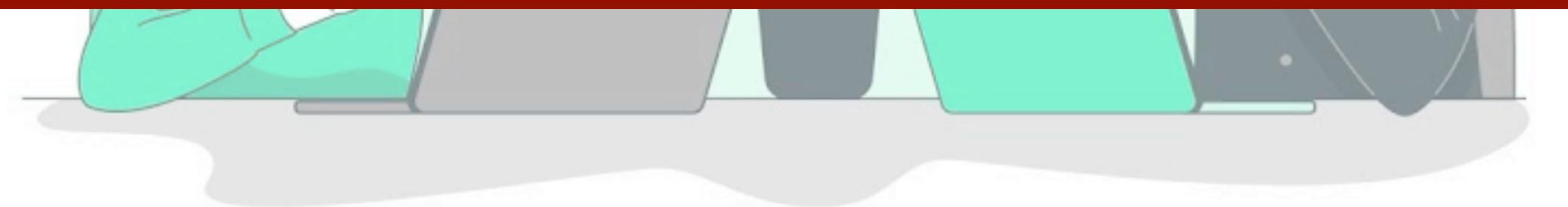


Lowered Barriers to Entry

Pull Request (PR) Is **Widely Used** for Contributing and Reviewing Changes in OSS Projects



OSS Maintainers Are Complaining About Abandoned PRs



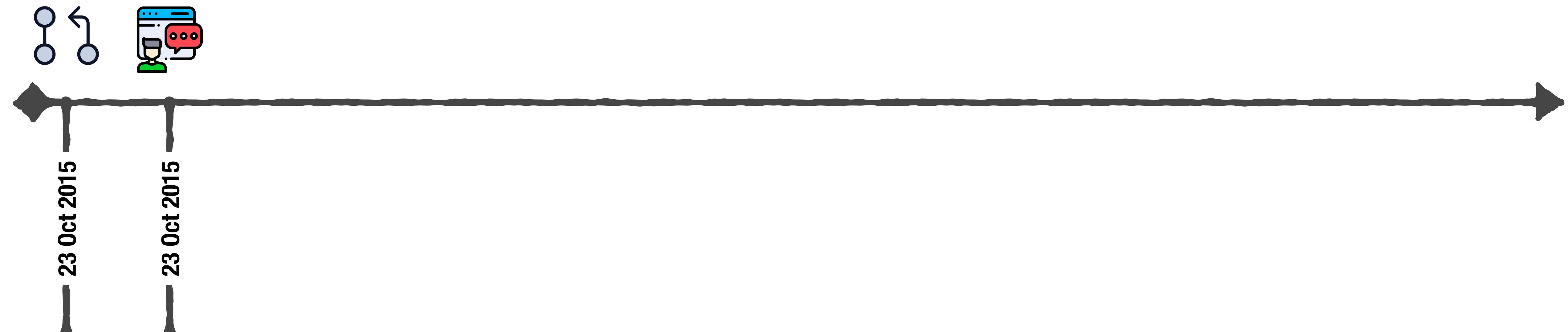
**Streamlined
Mechanism**

**Lowered
Barriers to Entry**

A Real-World **Example** of Abandoned PRs



A Real-World **Example** of Abandoned PRs



The diagram features a horizontal timeline with two vertical markers labeled "23 Oct 2015". Above the timeline are two icons: one showing a person connected to a network, and another showing a person at a computer with a speech bubble. A large, thin-lined rectangular box encloses the commit details.

thockin reviewed on Oct 23, 2015 View changes

`pkg/api/types.go`

```
... ... @@ -153,6 +153,15 @@ const (
153 153     TerminationMessagePathDefault string = "/dev/termination-log"
154 154 )
155 155
156 + // There is no namespace required for these kinds specified here.
157 + // If a kind is not enumerated here, it is assumed to have a namespace scope.
158 + var NoneNamespacedKinds = []string{
```

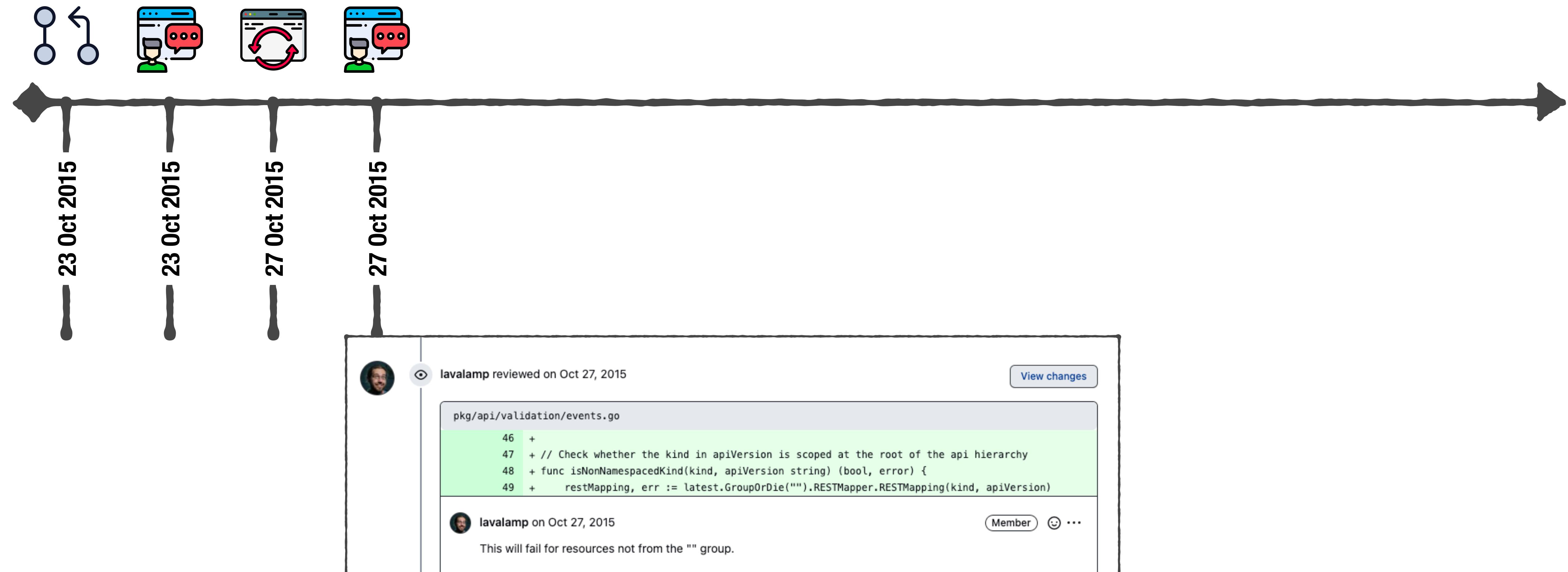
thockin on Oct 23, 2015 Member ...
The disadvantage of this is that the list is now far away from the place where types are actually installed, which means it will probably get forgotten.
@lavalamp @smarterclayton @derekwayncarr for ideas on how to do this best

thockin on Oct 23, 2015 Member ...
Also s/None/Non/

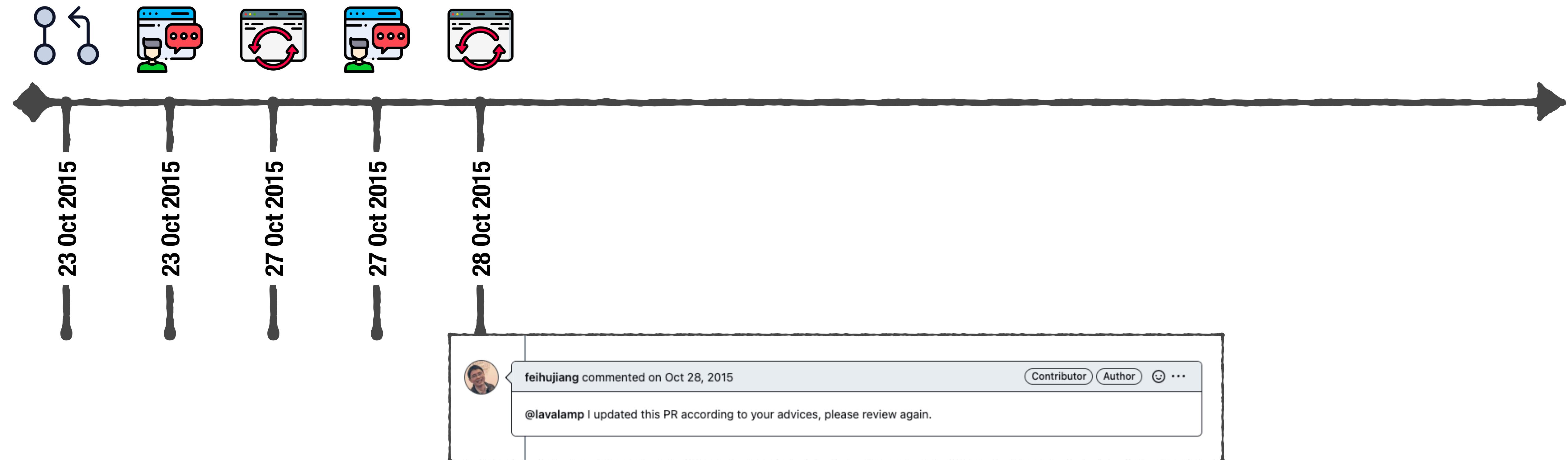
A Real-World **Example** of Abandoned PRs



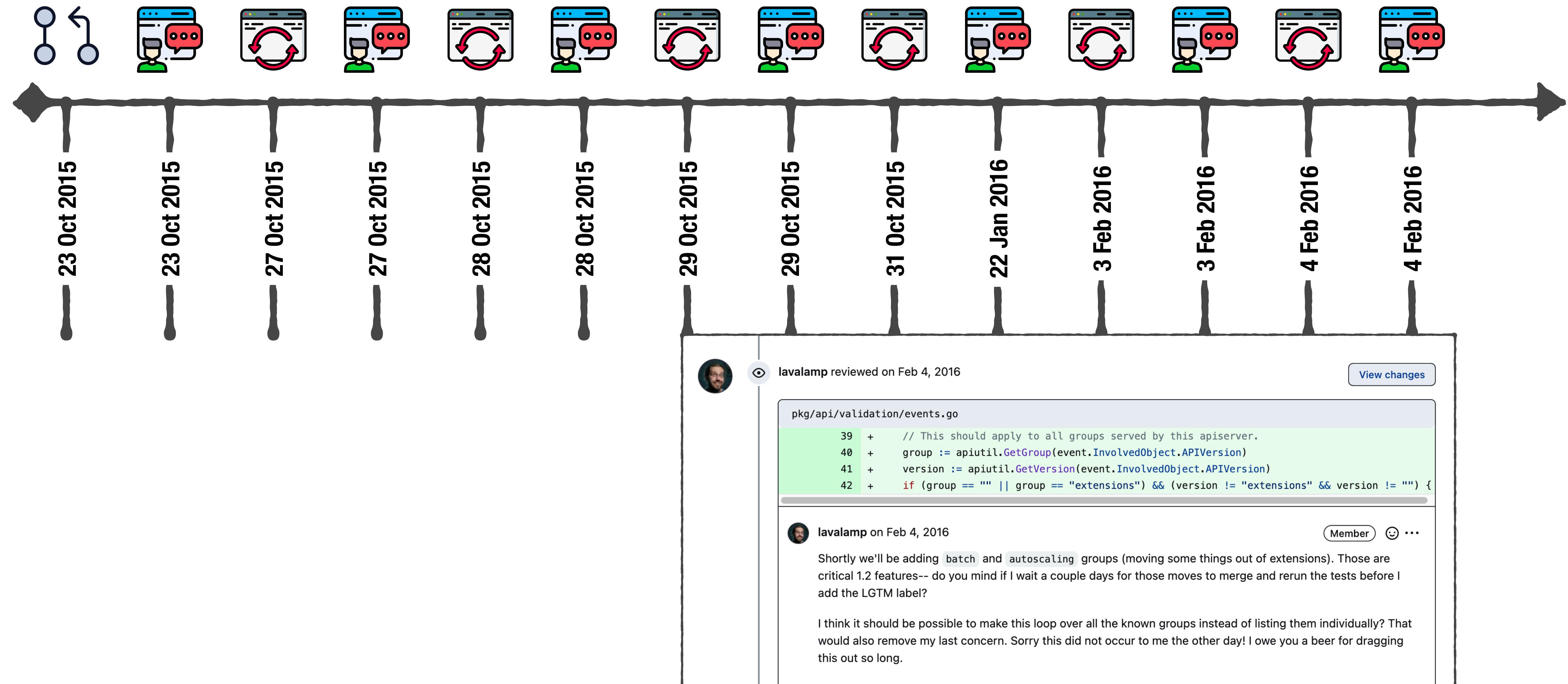
A Real-World **Example** of Abandoned PRs



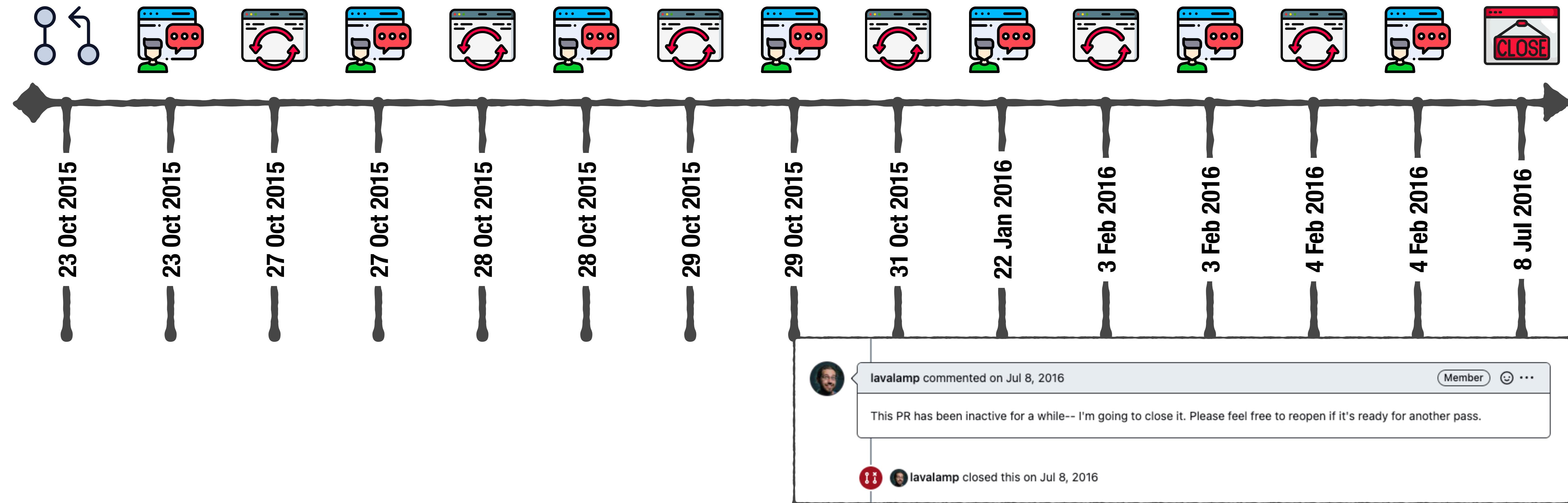
A Real-World **Example** of Abandoned PRs



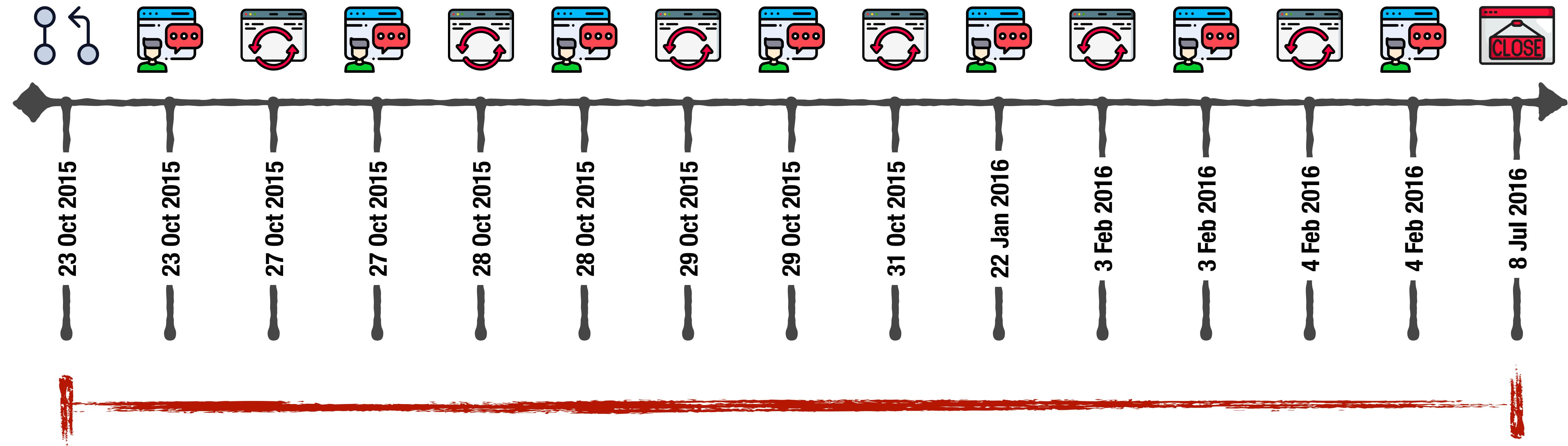
A Real-World **Example** of Abandoned PRs



A Real-World **Example** of Abandoned PRs



A Real-World **Example** of Abandoned PRs



8 Months of Work
7 Rounds of Reviews

A Real-World **Example** of Abandoned PRs



Abandoned PRs Make It **Difficult** To Manage and Prioritize PRs



DefinitelyTyped



RyanCavanaugh commented on Jun 29, 2016

Member

...

Hello and thank you for your contribution!

Due to an excessively long queue of pull requests that have become stale, we are "declaring bankruptcy" and closing all PRs opened before May 1, 2016. If you'd still like to merge this code in, please open a new PR that has been merged and rebased with the `master` branch.

Going forward, we are committing to review or merge all PRs on a regular basis so this bankruptcy will not occur again. We apologize for the inconvenience and hope you will continue to contribute to DefinitelyTyped in the future.

Thanks

The TypeScript and DefinitelyTyped teams



RyanCavanaugh closed this on Jun 29, 2016

PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li^{ID}, Yue Yu^{ID}, Tao Wang, Gang Yin, ShanShan Li^{ID}, and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers' willingness to remain long-term and established contributors to

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403



Waste Review Efforts

decrease the likelihood of developer disengagement and turnover [41], [43], [47], [54], [59], [78].

In this study, we shed light on OSS contributors' abandonment, focusing on the individual and single level in pull-based development [23], i.e., every single pull request that is unfinished but abandoned by the author. We clarify our research motivation as follows. The pull request mechanism [23], [71] provides a synthesized collaboration environment for OSS distributed development by coupling the code repository, modern issue tracker, code review and automatic tools of DevOps. This has unprecedentedly lowered the barrier to entry for potential contributors and simplified the collaborative development process compared to traditional patch-based methods [80]. It is interesting to understand the reasons why there are considerable unfinished pull requests whose contributors have spent time programming, submitting and discussing at the beginning but walk away, leaving their contributions abandoned. As project integrators mentioned in our survey, “*We see quite a lot of valuable contributions abandoned by authors, just applying a few demanded changes might make their pull requests good enough to be eventually accepted*” [SC224]. Furthermore, pull request abandonment is significant for the health of OSS projects and requires extra project maintenance efforts. Two representative quotes from our survey respondents indicate this: “*Opening a PR and walking away is bad form*” [SI3], and “*It's frustrating to see a bug almost fixed and then abandoned*” [SI6]. A more complete picture of pull request abandonment can help OSS communities obtain great benefits from designing appropriate strategies to overcome avoidable abandonment, building more efficient tools for distributed collaboration, and helping individual participants understand the mindset of abandonment, and find the best ways to contribute.

PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li^{ID}, Yue Yu^{ID}, Tao Wang, Gang Yin, ShanShan Li^{ID}, and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers’ willingness to remain long-term and established contributors to

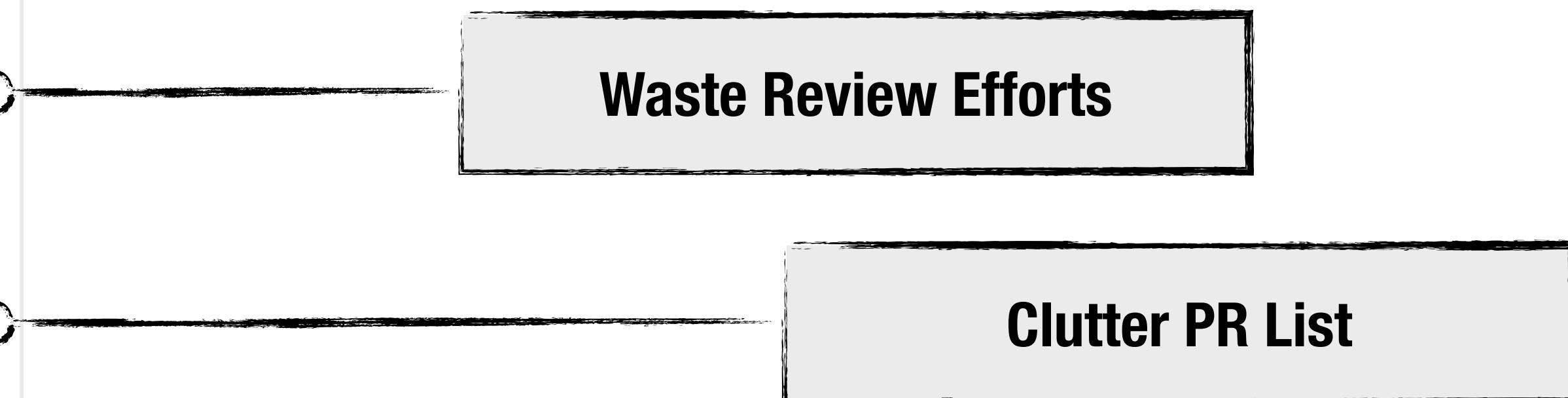
decrease the likelihood of developer disengagement and turnover [41], [43], [47], [54], [59], [78].

In this study, we shed light on OSS contributors’ abandonment, focusing on the individual and single level in pull-based development [23], i.e., every single pull request that is unfinished but abandoned by the author. We clarify our research motivation as follows. The pull request mechanism [23], [71] provides a synthesized collaboration environment for OSS distributed development by coupling the code repository, modern issue tracker, code review and automatic tools of DevOps. This has unprecedentedly lowered the barrier to entry for potential contributors and simplified the collaborative development process compared to traditional patch-based methods [80]. It is interesting to understand the reasons why there are considerable unfinished pull requests whose contributors have spent time programming, submitting and discussing at the beginning but walk away, leaving their contributions abandoned. As project integrators mentioned in our survey, “*We see quite a lot of valuable contributions abandoned by authors, just applying a few demanded changes might make their pull requests good enough to be eventually accepted*” [SC224]. Furthermore, pull request abandonment is significant for the health of OSS projects and requires extra project maintenance efforts. Two representative quotes from our survey respondents indicate this: “*Opening a PR and walking away is bad form*” [SI3], and “*It’s frustrating to see a bug almost fixed and then abandoned*” [SI6]. A more complete picture of pull request abandonment can help OSS communities obtain great benefits from designing appropriate strategies to overcome avoidable abandonment, building more efficient tools for distributed collaboration, and helping individual participants understand the mindset of abandonment, and find the best ways to contribute.

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403



PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li[✉], Yue Yu[✉], Tao Wang, Gang Yin, ShanShan Li[✉], and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers’ willingness to remain long-term and established contributors to

decrease the likelihood of developer disengagement and turnover [41], [43], [47], [54], [59], [78].

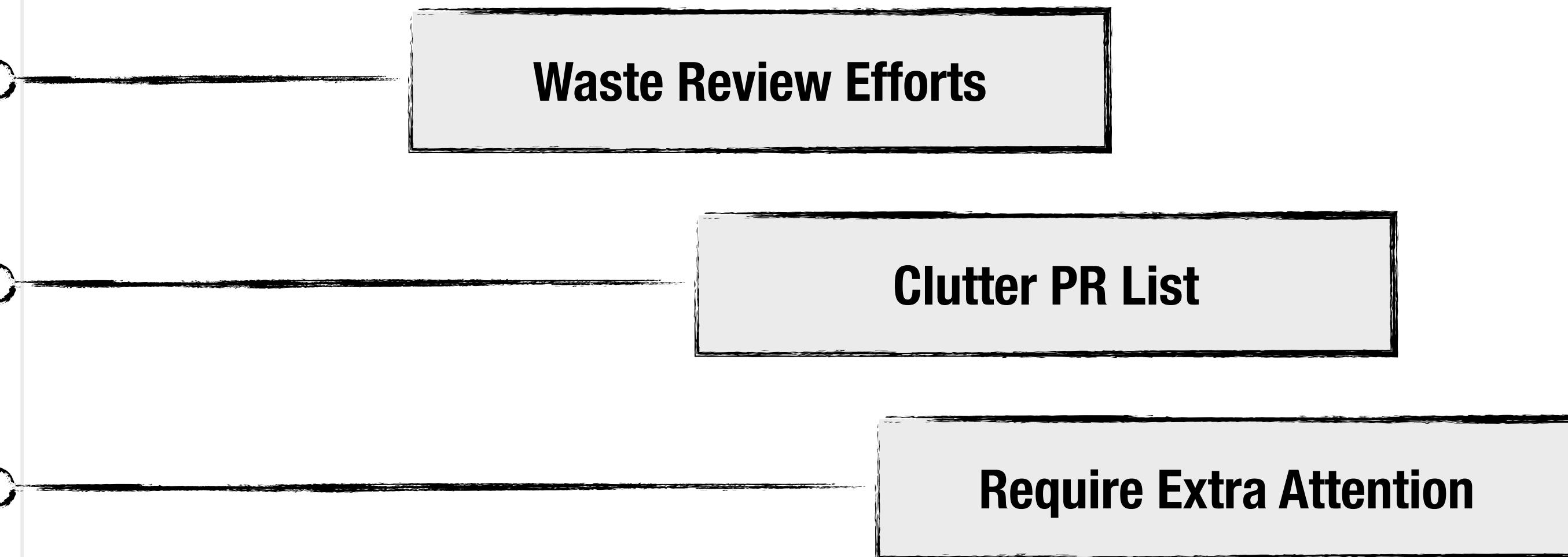
In this study, we shed light on OSS contributors’ abandonment, focusing on the individual and single level in pull-based development [23], i.e., every single pull request that is unfinished but abandoned by the author. We clarify our research motivation as follows. The pull request mechanism [23], [71] provides a synthesized collaboration environment for OSS distributed development by coupling the code repository, modern issue tracker, code review and automatic tools of DevOps. This has unprecedentedly lowered the barrier to entry for potential contributors and simplified the collaborative development process compared to traditional patch-based methods [80]. It is interesting to understand the reasons why there are considerable unfinished pull requests whose contributors have spent time programming, submitting and discussing at the beginning but walk away, leaving their contributions abandoned. As project integrators mentioned in our survey, “*We see quite a lot of valuable contributions abandoned by authors, just applying a few demanded changes might make their pull requests good enough to be eventually accepted*” [SC224]. Furthermore, pull request abandonment is significant for the health of OSS projects and requires extra project maintenance efforts. Two representative quotes from our survey respondents indicate this: “*Opening a PR and walking away is bad form*” [SI3], and “*It’s frustrating to see a bug almost fixed and then abandoned*” [SI6]. A more complete picture of pull request abandonment can help OSS communities obtain great benefits from designing appropriate strategies to overcome avoidable abandonment, building more efficient tools for distributed collaboration, and helping individual participants understand the mindset of abandonment, and find the best ways to contribute.

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.)

Recommended for acceptance by M. Nagappan.
Digital Object Identifier no. 10.1109/TSE.2021.3053403



PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li^{ID}, Yue Yu^{ID}, Tao Wang, Gang Yin, ShanShan Li^{ID}, and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

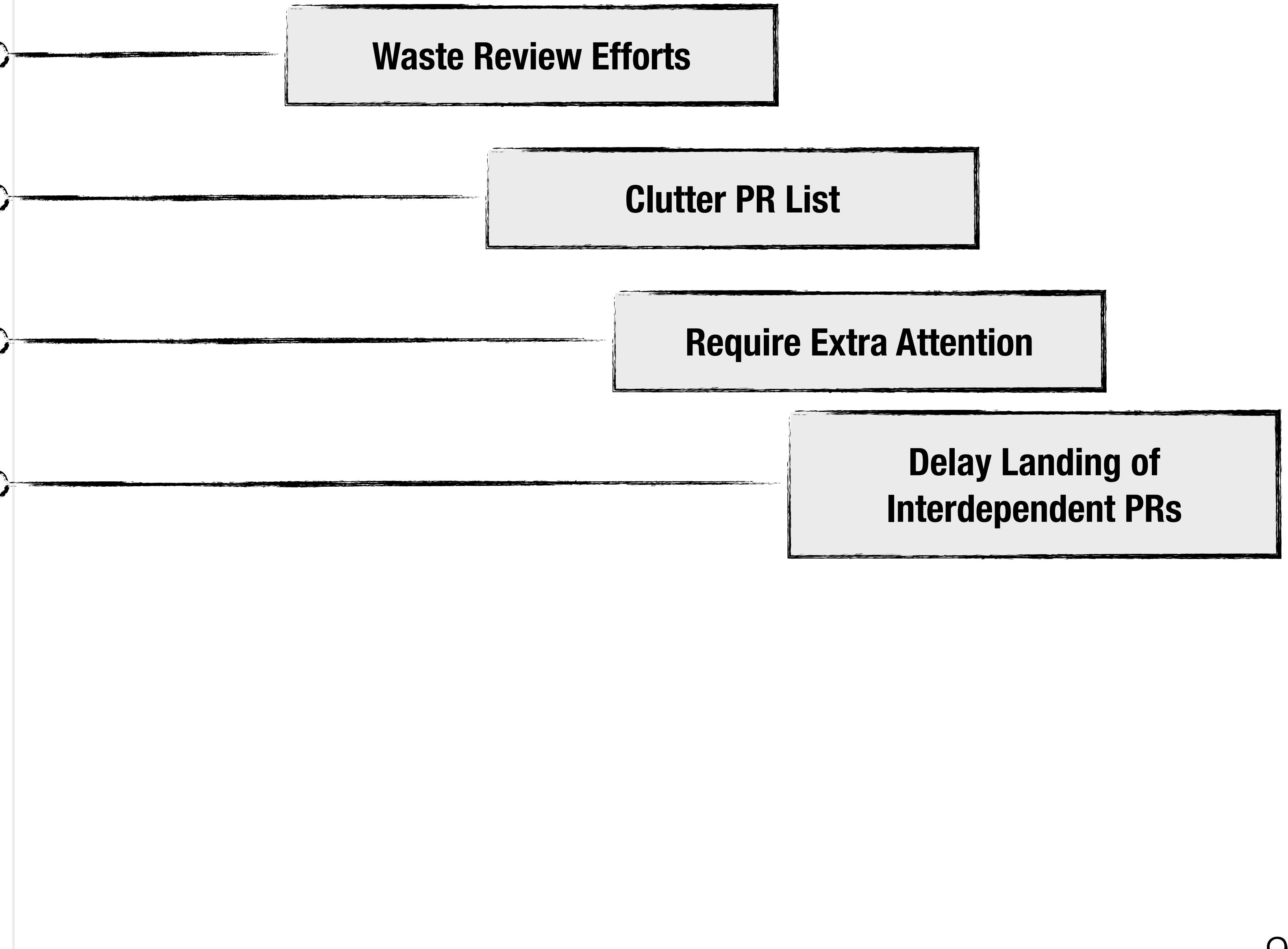
THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers’ willingness to remain long-term and established contributors to

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403



PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li^{ID}, Yue Yu^{ID}, Tao Wang, Gang Yin, ShanShan Li^{ID}, and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

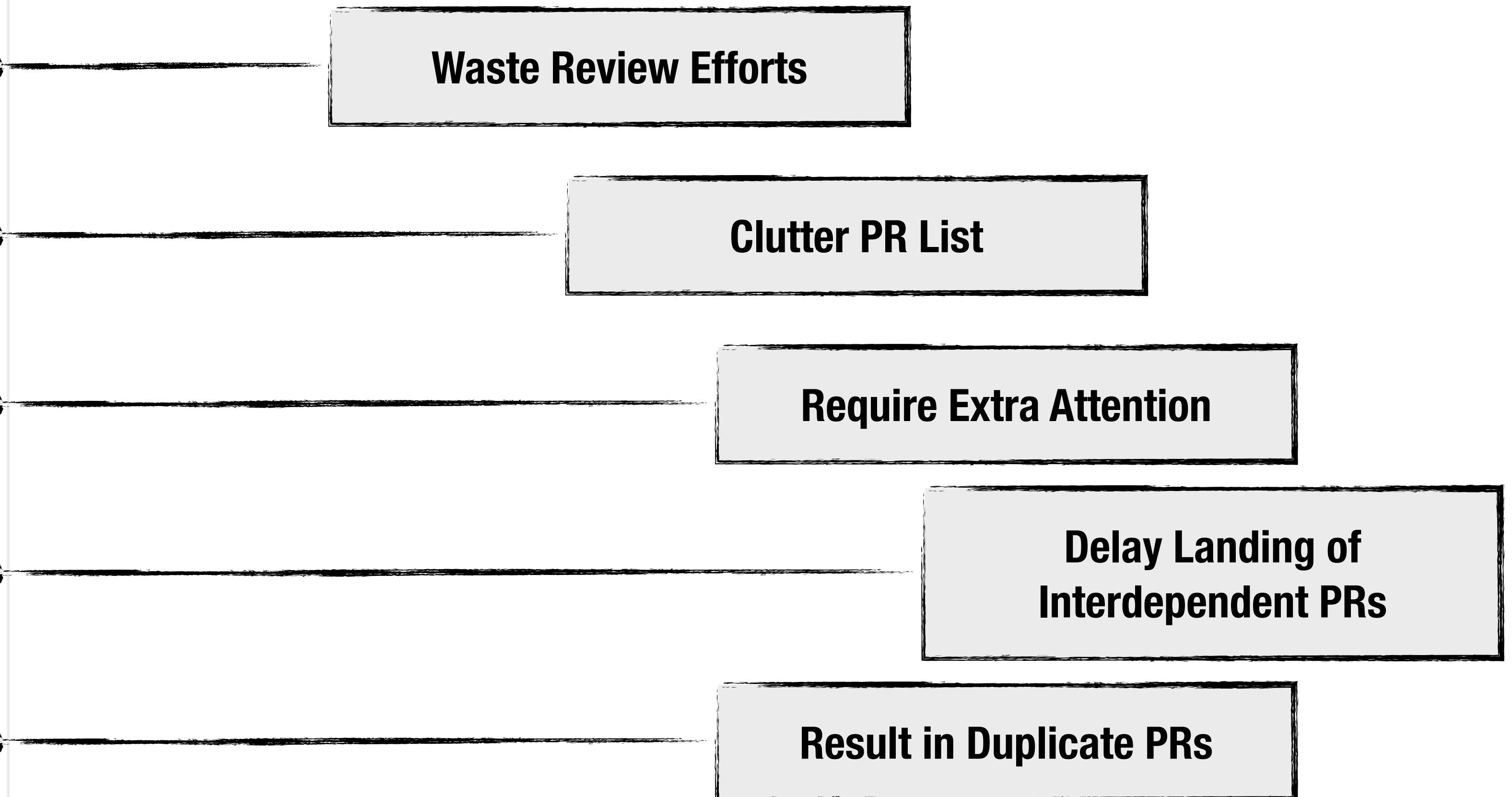
THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers’ willingness to remain long-term and established contributors to

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403



PR Abandonment Is a Non-Trivial Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li^{ID}, Yue Yu^{ID}, Tao Wang, Gang Yin, ShanShan Li^{ID}, and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

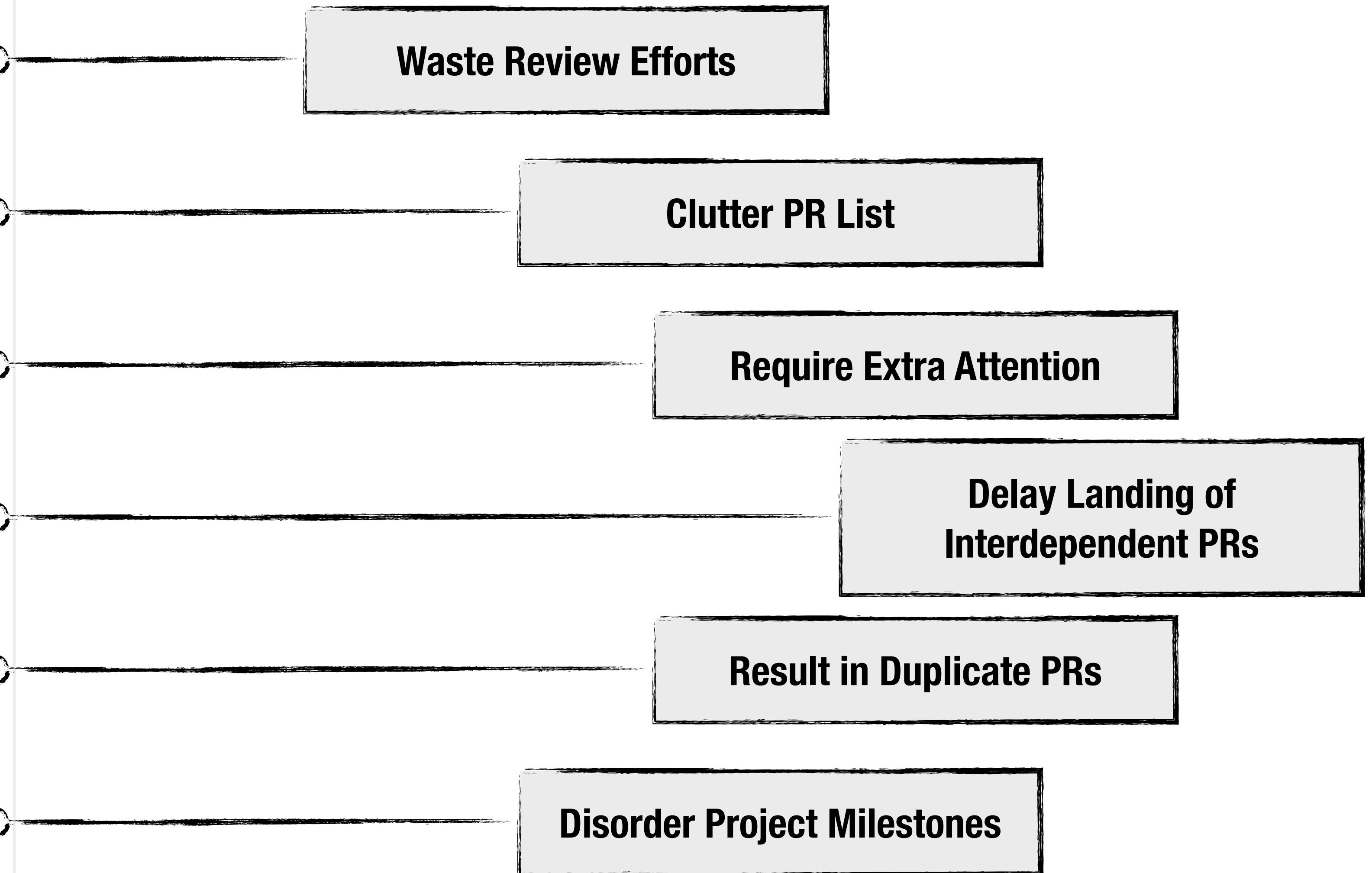
THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers' willingness to remain long-term and established contributors to

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403



PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li^{ID}, Yue Yu^{ID}, Tao Wang, Gang Yin, ShanShan Li^{ID}, and Huaimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well studied, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We manually examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show the significant impacts of pull request abandonment on project management and maintenance. Moreover, we elucidate the strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

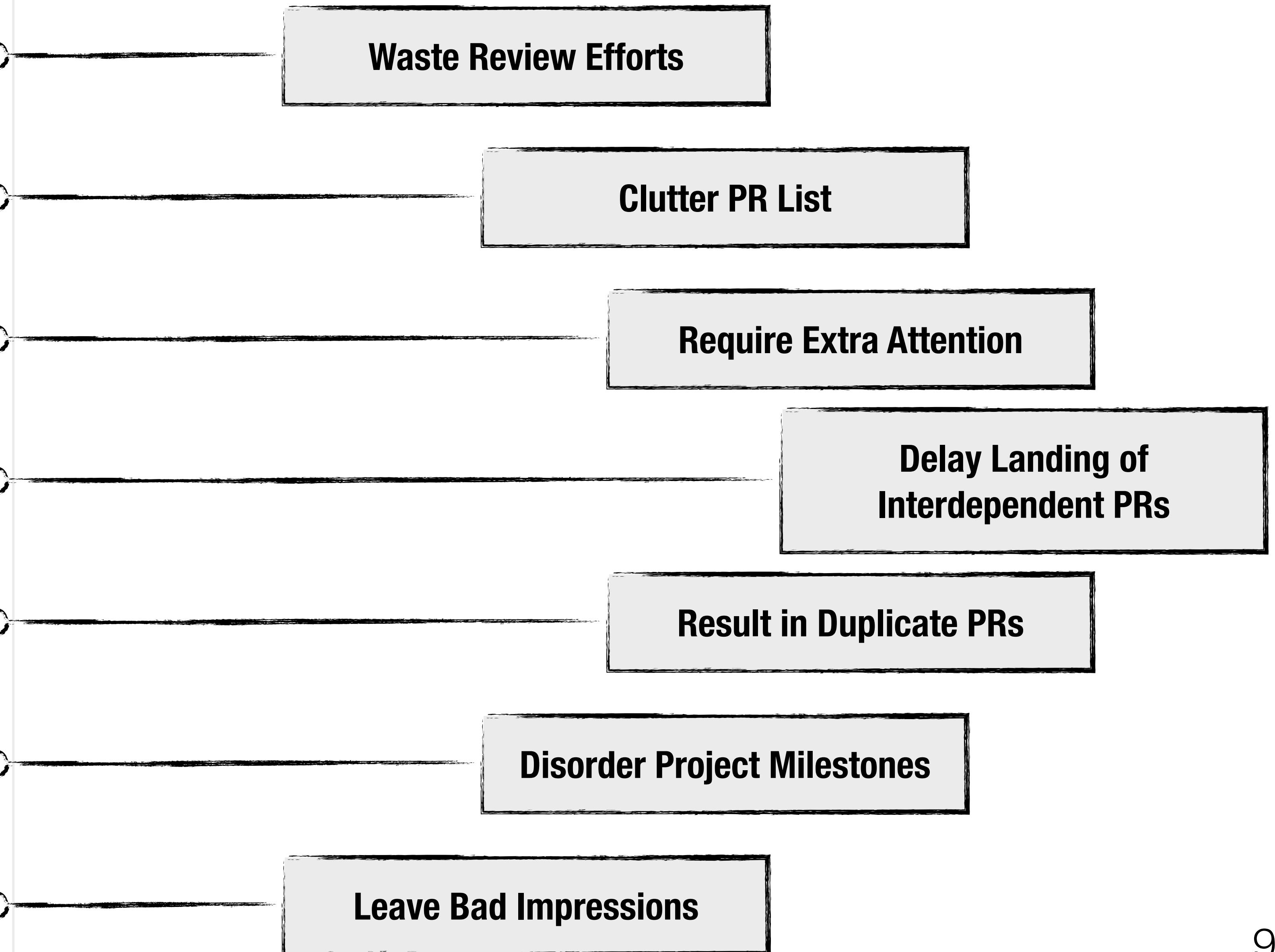
THE great success of open source software (OSS) is based on the theory that “many hands make light work” [57], [73]. For a popular community-based OSS project, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as the core-periphery structure [16], [17], and many external contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributors [9], [30], [40], [41], [58], [75]; ii) uncovering the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers' willingness to remain long-term and established contributors to

• Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Huaimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {lizhixing15, yuyue, taowang2005, yinggang, hmwang}@nudt.edu.cn.

• ShanShan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.

Manuscript received 8 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403



Research Statement

To **investigate** the underlying dynamics of abandoned PRs,
evaluate the helpfulness of common solutions to PR abandonment,
and **propose** ways to mitigate PR abandonment
in large OSS projects.

Thesis Organization

Thesis Organization

Chapter 3: Understanding the Dynamics of Contributor-Abandoned PRs

Thesis Organization

Chapter 3: Understanding the
Dynamics of
Contributor-Abandoned PRs

Chapter 4: Understanding the
Helpfulness of
Stale Bot for PRs

Thesis Organization

Chapter 3: Understanding the
Dynamics of
Contributor-Abandoned PRs

Chapter 4: Understanding the
Helpfulness of
Stale Bot for PRs

Chapter 5: Predicting the
First Response Latency of
Maintainers and Contributors in PRs

Thesis Organization

Chapter 3: Understanding the Dynamics of Contributor-Abandoned PRs

Chapter 4: Understanding the Helpfulness of Stale Bot for PRs

Chapter 5: Predicting the First Response Latency of Maintainers and Contributors in PRs

Thesis Organization

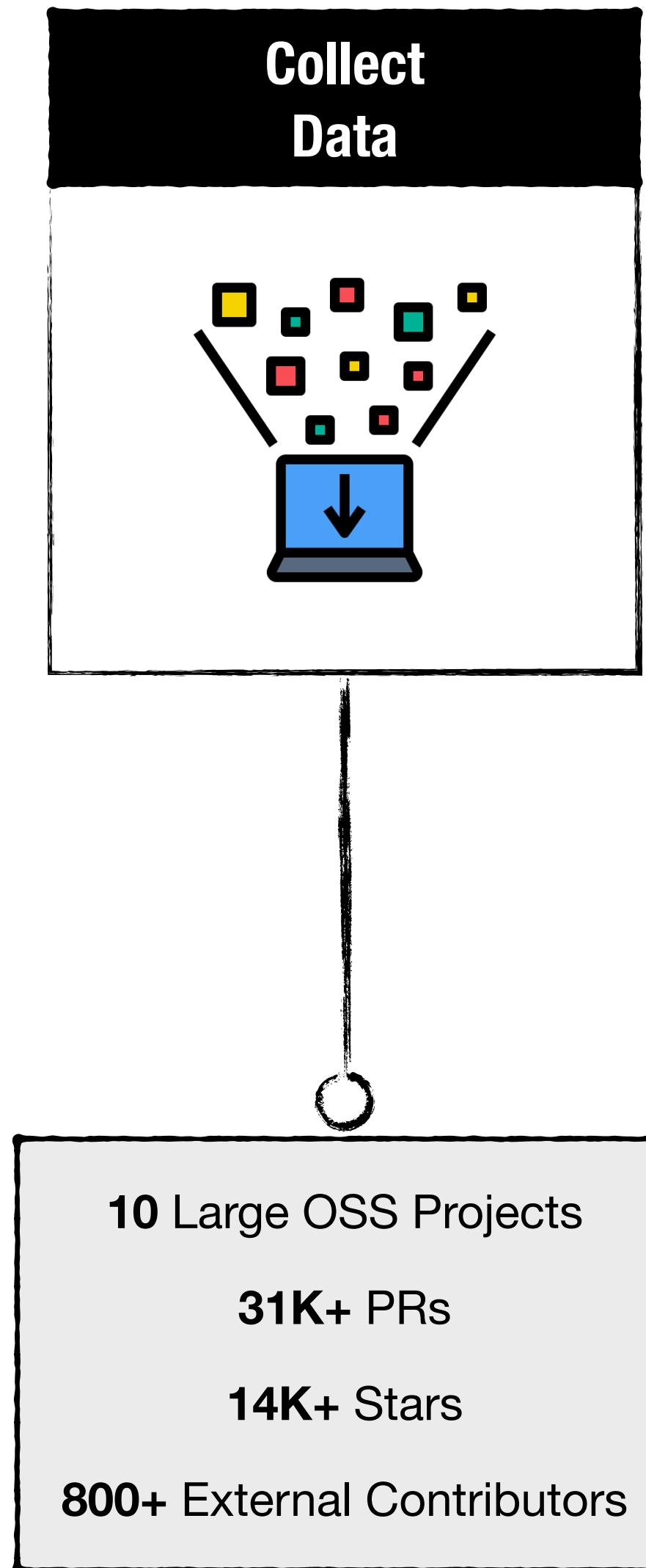
Chapter 3: Understanding the Dynamics of Contributor-Abandoned PRs

Chapter 4: Understanding the Helpfulness of Stale Bot for PRs

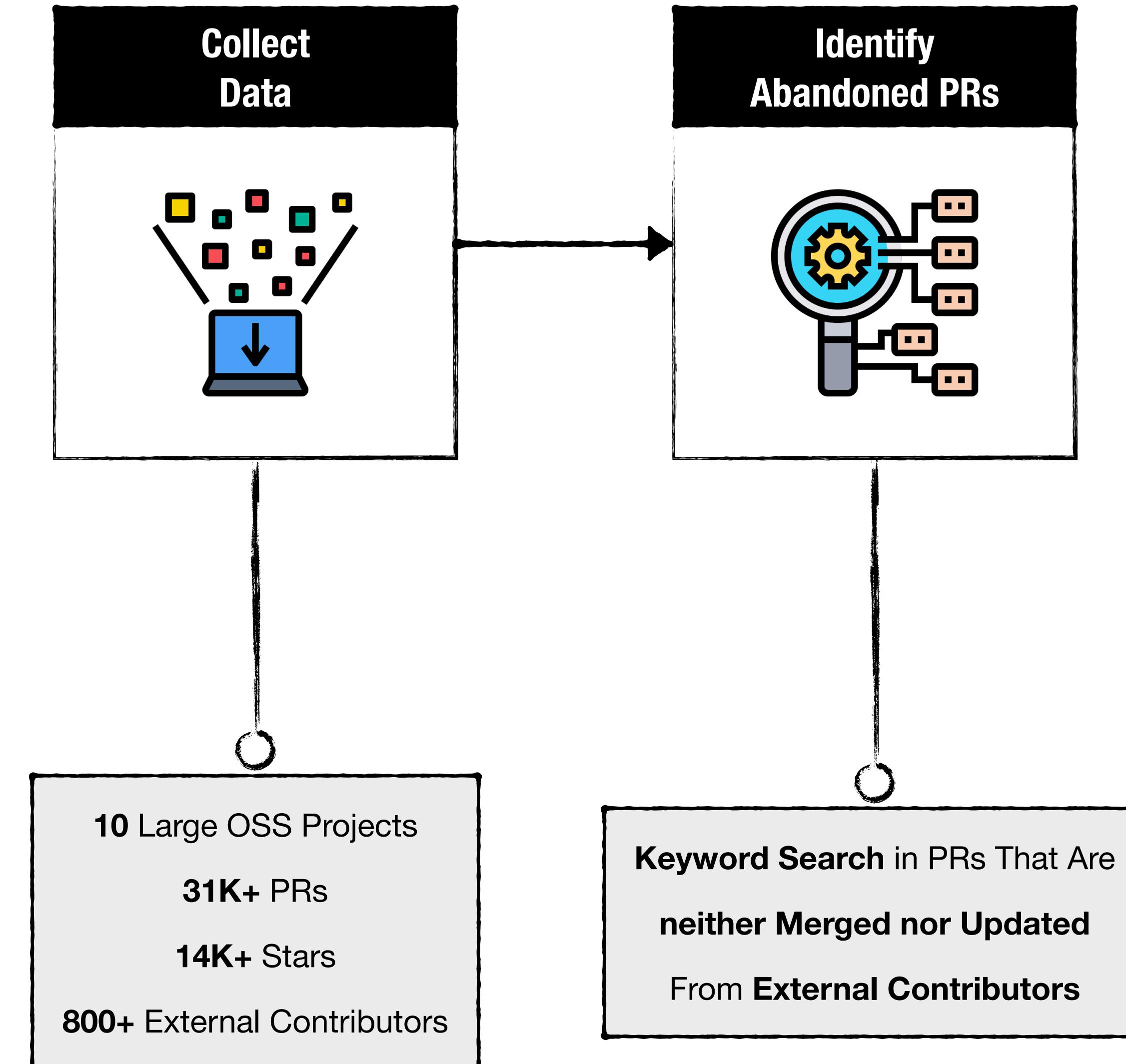
Chapter 5: Predicting the First Response Latency of Maintainers and Contributors in PRs

Understanding the **Dynamics of Contributor-Abandoned PRs**

Understanding the **Dynamics of Contributor-Abandoned PRs**

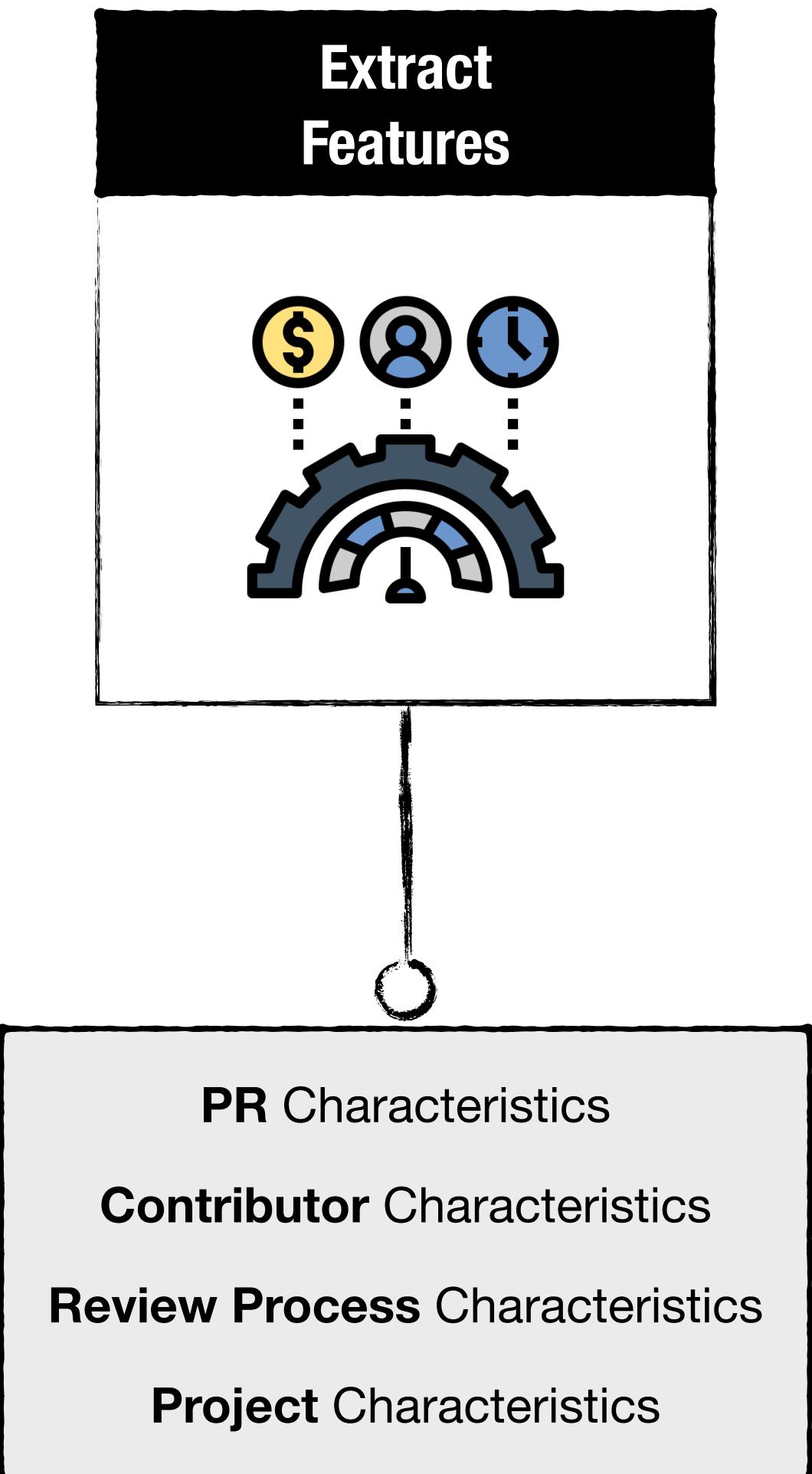


Understanding the Dynamics of Contributor-Abandoned PRs

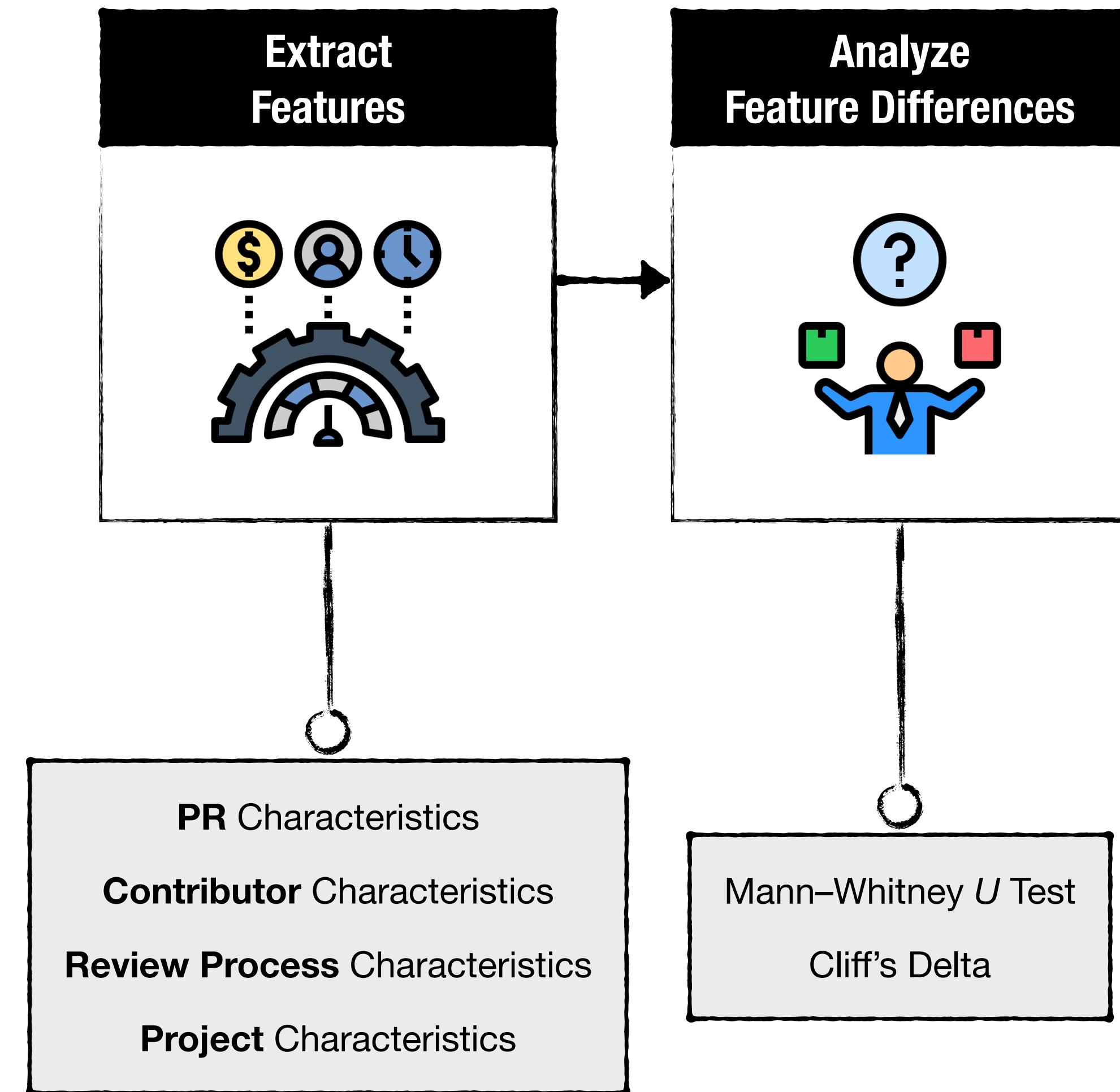


What Are the **Characteristics** of Abandoned PRs?

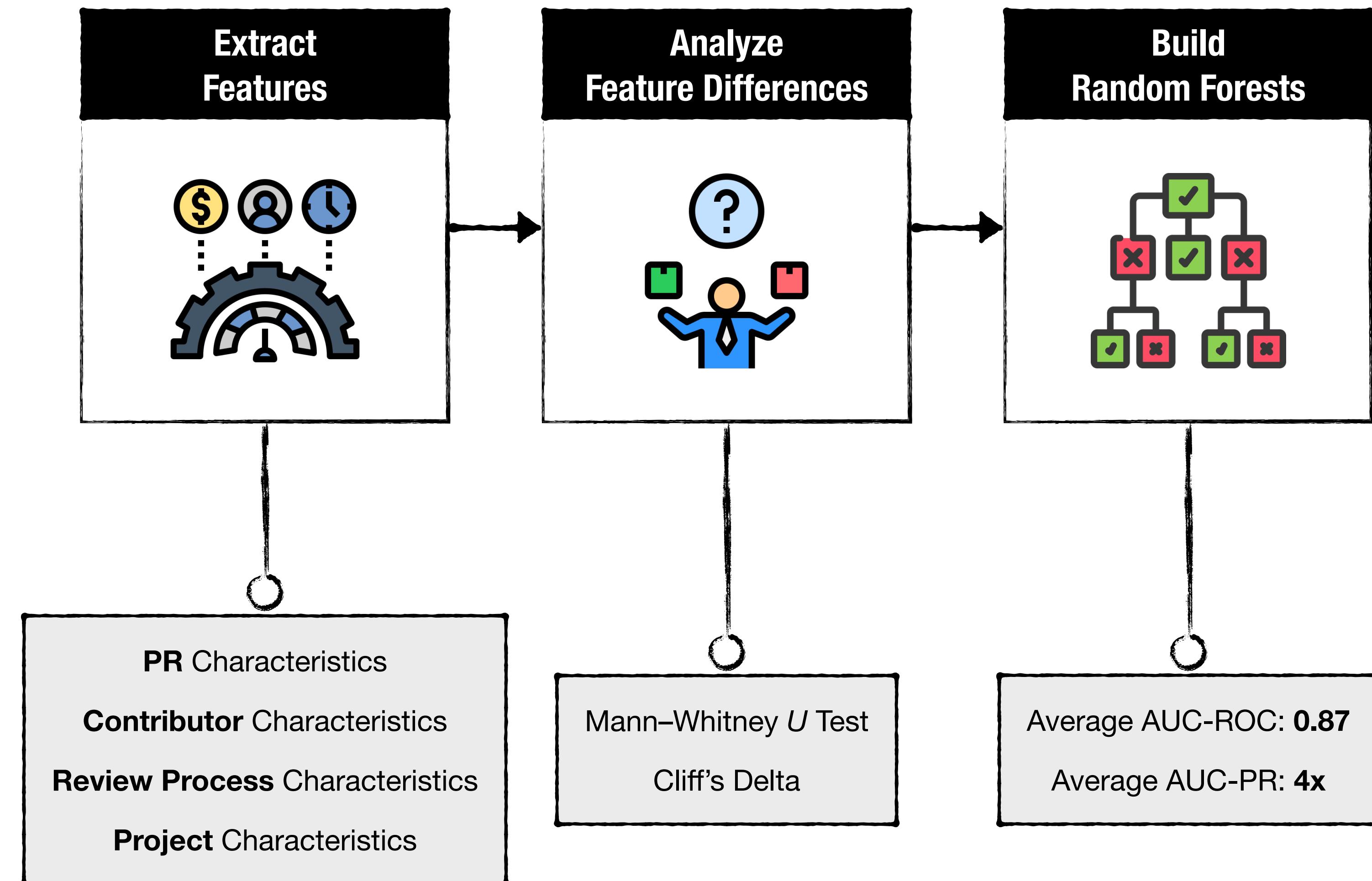
What Are the **Characteristics** of Abandoned PRs?



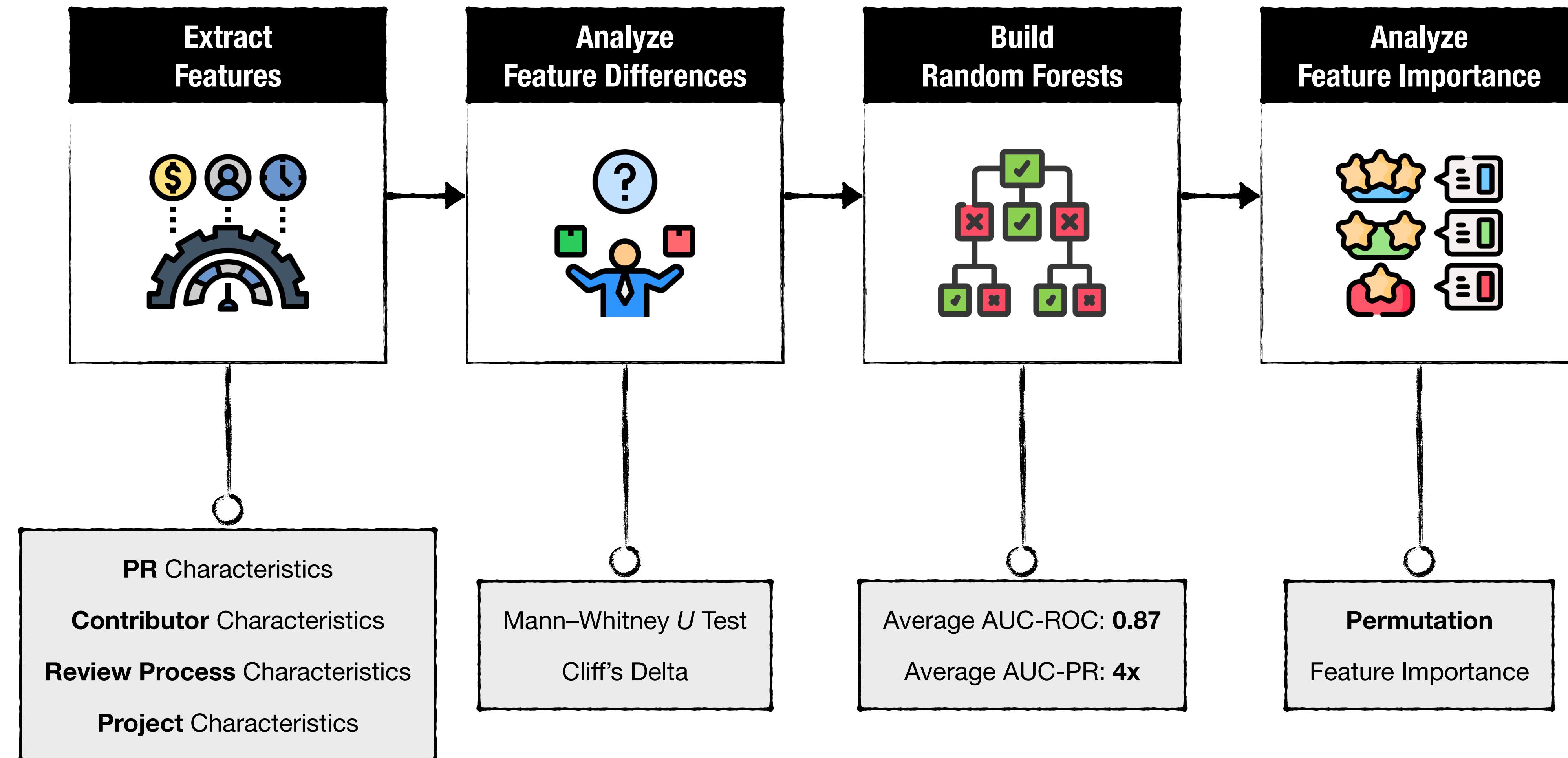
What Are the **Characteristics** of Abandoned PRs?



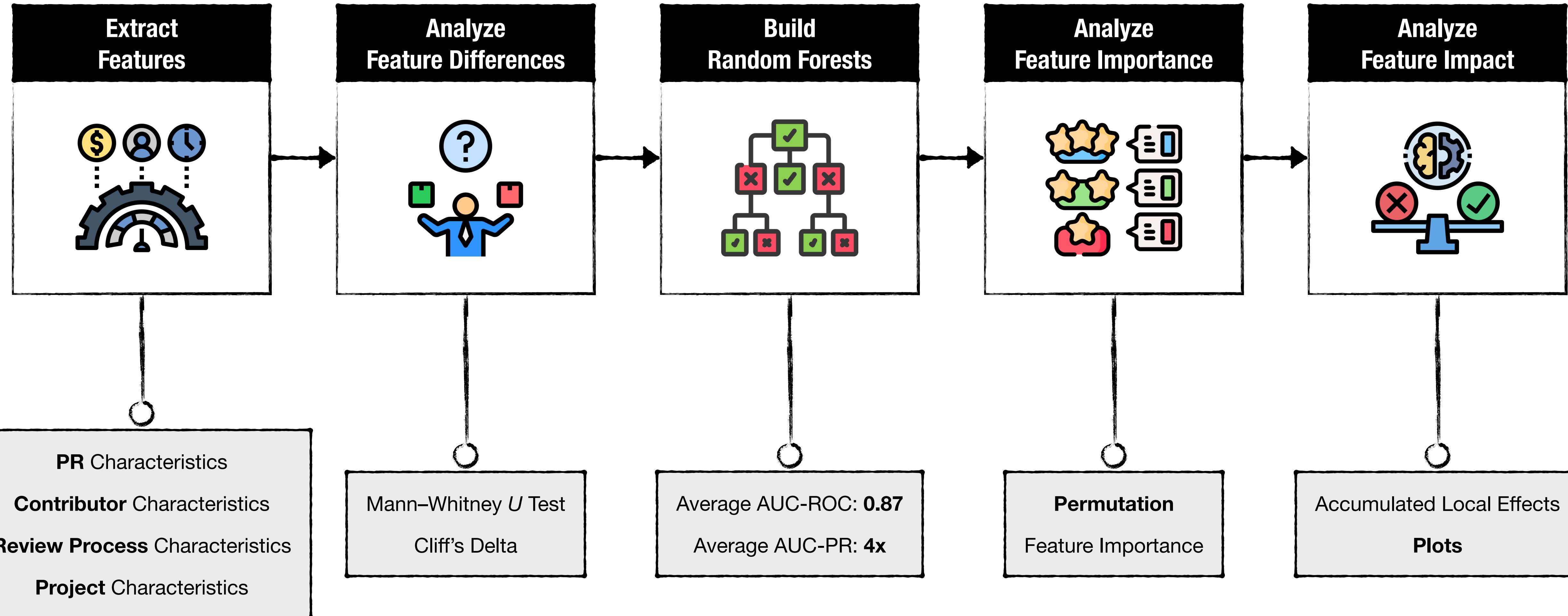
What Are the **Characteristics** of Abandoned PRs?



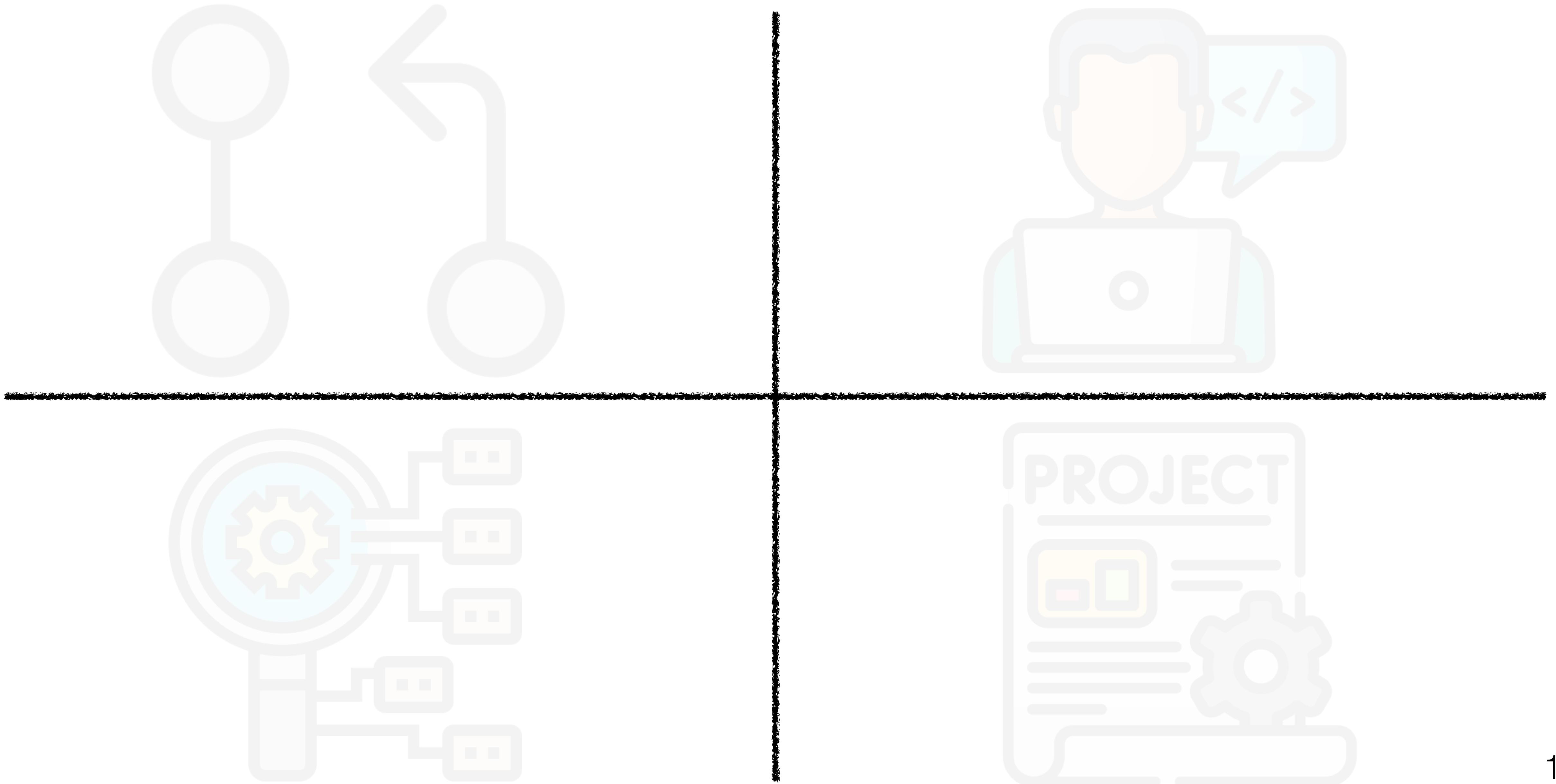
What Are the **Characteristics** of Abandoned PRs?



What Are the **Characteristics** of Abandoned PRs?



What Are the **Characteristics** of Abandoned PRs?



What Are the **Characteristics** of Abandoned PRs?

More Complex PRs

- Lengthier Descriptions
- More Commits
- More Changed Lines

What Are the **Characteristics** of Abandoned PRs?

More Complex PRs

- Lengthier Descriptions
- More Commits
- More Changed Lines

Less Experienced Contributors

- Fewer Submitted PRs
- Lower Acceptance Rate
- Lower Contribution Period
- Higher Abandonment Rate

What Are the **Characteristics** of Abandoned PRs?

More Complex PRs

- Lengthier Descriptions
- More Commits
- More Changed Lines

Less Experienced Contributors

- Fewer Submitted PRs
- Lower Acceptance Rate
- Lower Contribution Period
- Higher Abandonment Rate

Lengthier Review Processes

- More Participants
- More Participant Responses
- More Contributor Responses
- Higher Response Latency

What Are the **Characteristics** of Abandoned PRs?

More Complex PRs

- Lengthier Descriptions
- More Commits
- More Changed Lines

Less Experienced Contributors

- Fewer Submitted PRs
- Lower Acceptance Rate
- Lower Contribution Period
- Higher Abandonment Rate

Lengthier Review Processes

- More Participants
- More Participant Responses
- More Contributor Responses
- Higher Response Latency

Projects Play Both Positive and Negative Roles

- Both More and Less Maturity
- Both More and Less Workload

What Are the **Characteristics** of Abandoned PRs?

More Complex PRs

Lengthier Descriptions
More Commits
More Changed Lines

Less Experienced Contributors

Fewer Submitted PRs
Lower Acceptance Rate
Lower Contribution Period
Higher Abandonment Rate

Lengthier Review Processes

More Participants

More Participant Responses (3x)

More Contributor Responses

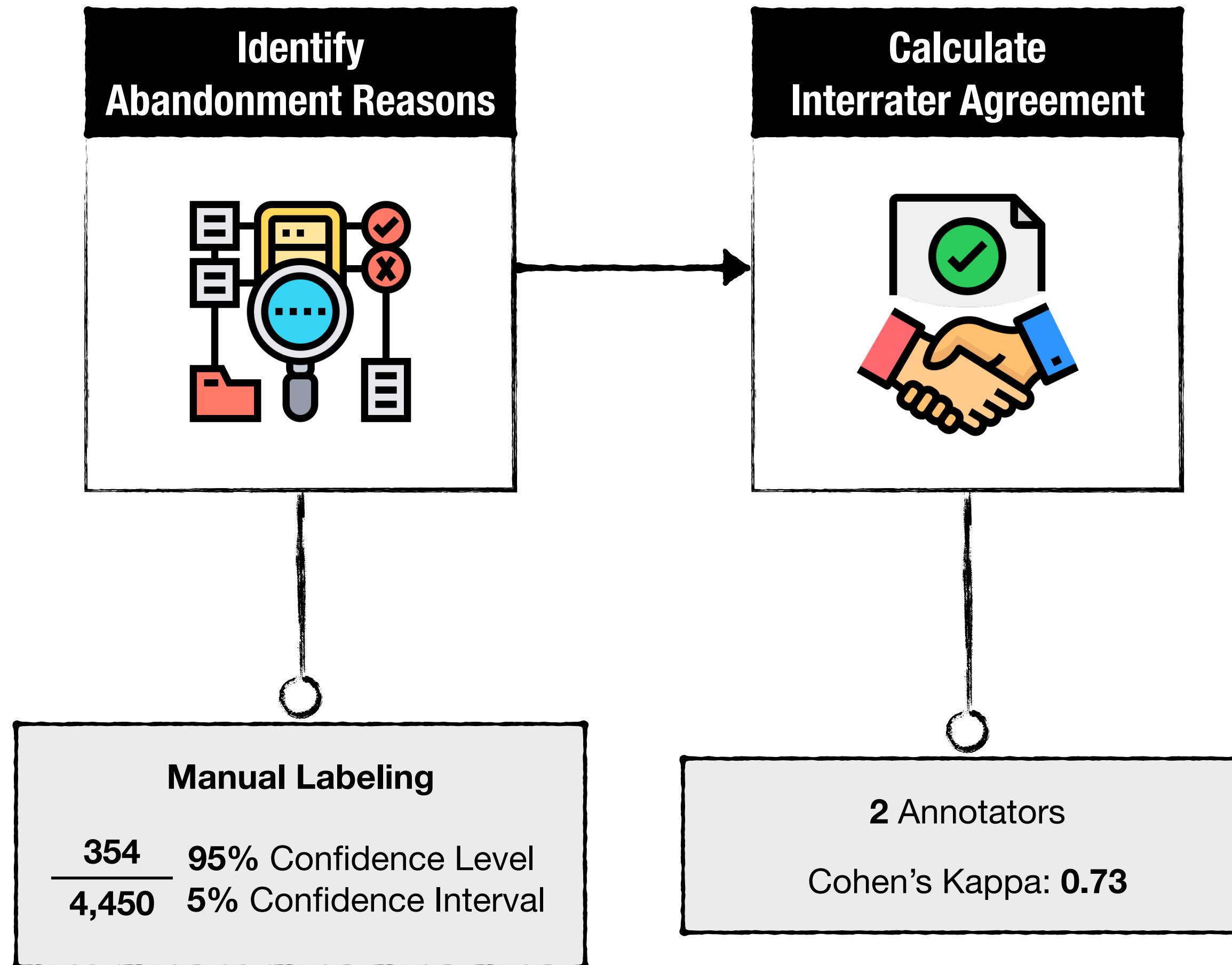
Higher Response Latency

Projects Play Both Positive and Negative Roles

Both More and Less Maturity
Both More and Less Workload

Why Do Contributors Abandon Their PRs?

Why Do Contributors Abandon Their PRs?



Why Do Contributors Abandon Their PRs?

Obstacles Faced by **Contributors**

Hurdles Imposed by **Maintainers**

Why Do Contributors Abandon Their PRs?

Obstacles Faced by **Contributors**

Difficulty addressing the
maintainers' comments
(45.8%)

Difficulty resolving the
CI failures
(20.9%)

Difficulty resolving the
merge issues
(14.1%)

Hurdles Imposed by **Maintainers**

Why Do Contributors Abandon Their PRs?

Obstacles Faced by **Contributors**

Difficulty addressing the
maintainers' comments
(45.8%)

Difficulty resolving the
CI failures
(20.9%)

Difficulty resolving the
merge issues
(14.1%)

Hurdles Imposed by **Maintainers**

Lack of review
from the maintainers
(22.6%)

Lack of answer
from the maintainers
(9.3%)

Lack of integration
by the maintainers
(6.5%)

Chapter3: Understanding the Dynamics of Contributor-Abandoned PRs



On Wasted Contributions: Understanding the Dynamics of Contributor-Abandoned Pull Requests—A Mixed-Methods Study of 10 Large Open-Source Projects

SAYEDHASSAN KHATOONABADI and DIEGO ELIAS COSTA, Concordia University, Canada
RABE ABDALKAREEM, Carleton University, Canada
EMAD SHIHAB, Concordia University, Canada

Pull-based development has enabled numerous volunteers to contribute to open-source projects with fewer barriers. Nevertheless, a considerable amount of pull requests (PRs) with valid contributions are abandoned by their *contributors*, wasting the effort and time put in by both the contributors and maintainers. To better understand the underlying dynamics of contributor-abandoned PRs, we conduct a mixed-methods study using both quantitative and qualitative methods. We curate a dataset consisting of 265,325 PRs including 4,450 abandoned ones from ten popular and mature GitHub projects and measure 16 features characterizing PRs, contributors, review processes, and projects. Using statistical and machine learning techniques, we find that complex PRs, novice contributors, and lengthy reviews have a higher probability of abandonment and the rate of PR abandonment fluctuates alongside the projects' maturity or workload. To identify why contributors abandon their PRs, we also manually examine a random sample of 354 abandoned PRs. We observe that the most frequent abandonment reasons are related to the obstacles faced by contributors, followed by the hurdles imposed by maintainers during the review process. Finally, we survey the top core maintainers of the studied projects to understand their perspectives on dealing with PR abandonment and on our findings.

CCS Concepts: • Software and its engineering → Collaboration in software development; • Human-centered computing → Empirical studies in collaborative and social computing;

Additional Key Words and Phrases: Socio-technical factors, pull-based development, modern code review, social coding platforms, open-source software, mixed-methods research

ACM Reference format:

SayedHassan Khatoonabadi, Diego Elias Costa, Rabe Abdalkareem, and Emad Shihab. 2023. On Wasted Contributions: Understanding the Dynamics of Contributor-Abandoned Pull Requests—A Mixed-Methods Study of 10 Large Open-Source Projects. *ACM Trans. Softw. Eng. Methodol.* 32, 1, Article 15 (February 2023), 39 pages.
<https://doi.org/10.1145/3530785>

Authors' addresses: S. Khatoonabadi, D. E. Costa, and E. Shihab, Data-driven Analysis of Software (DAS) Lab, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC, H3H 2L9, Canada; emails: {sayedhassan.khatoonabadi, diego.costa, emad.shihab}@concordia.ca; R. Abdalkareem, School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6, Canada; email: rabe.abdalkareem@carleton.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

1049-331X/2023/02-ART15 \$15.00

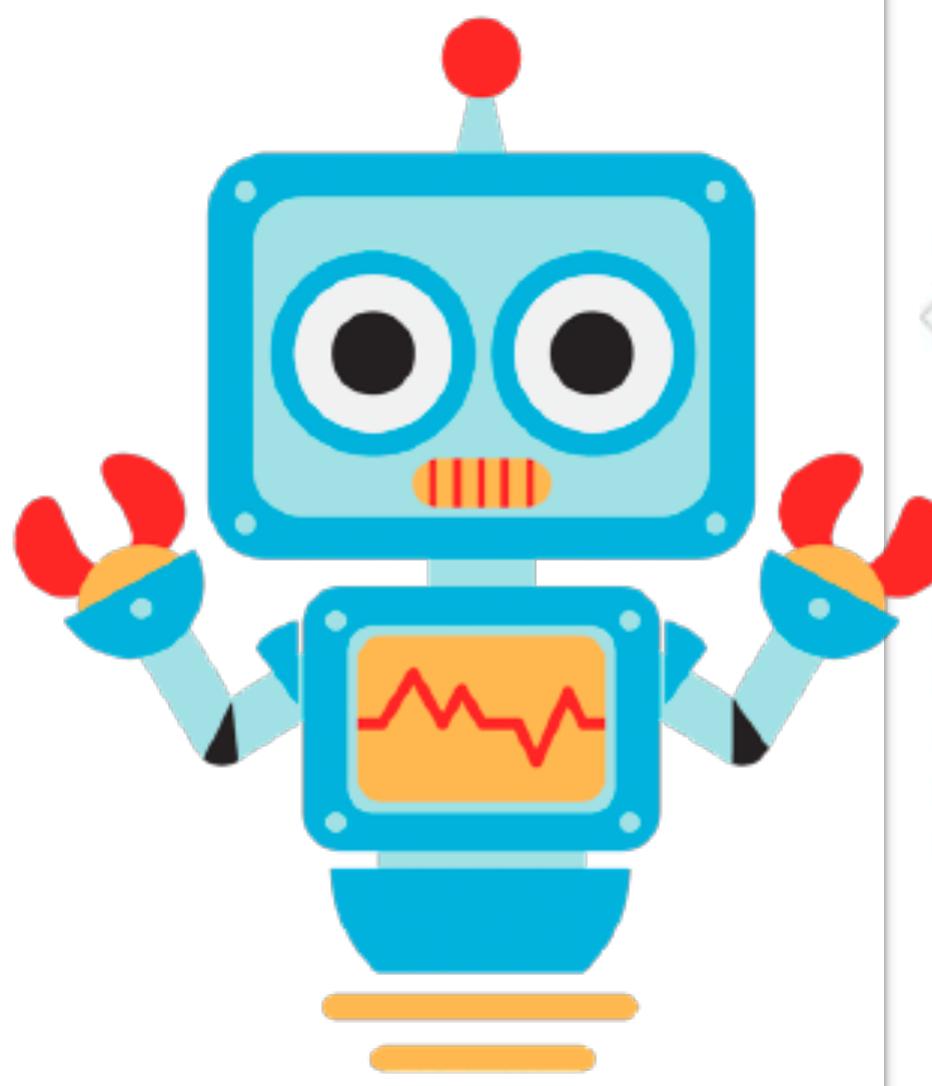
<https://doi.org/10.1145/3530785>

ACM Transactions on Software Engineering and Methodology, Vol. 32, No. 1, Article 15. Publication date: February 2023.

Contributor experience and review process length play the biggest roles in PR abandonment.

[Khatoonabadi et al., TOSEM 2023]

Stale Bot Is a **Common** Solution to PR Abandonment



probot-stale bot added the `stale` label on May 9

probot-stale bot commented on May 9

+ 😊 🖍 ✕

This issue has been automatically marked as stale because it has not had recent activity. It will be closed if no further activity occurs. Thank you for your contributions.

probot-stale bot closed this 23 days ago

However, Stale Bot Is a **Controversial** Solution

helm/helm

#11474 stop using stale bot



1 comment

 clintmod opened on October 26, 2022



NuGet/docs.microsoft.com-nuget

#2449 Stale bot is closing issues inappropriately

1 comment

 StingyJack opened on June 2, 2021



OWASP/glue

#111 Please review your use of stale bot



4 comments

 psiilon opened on May 9, 2018



kubernetes/community

#6418 Please 🛡️ remove the stale bot



3 comments

 ghostsquad opened on February 1, 2022



probot/stale

#343 Stale-bot closing issues which should remain open



9 comments

 NiklasBr opened on June 19, 2021



mockk/mockk

#356 Discussion: Deactive stale bot



3 comments

 krichter722 opened on September 17, 2019



jitsi/jitsi-meet

#5332 switch off stale bot



2 comments

 WolfgangFahl opened on March 23, 2020



mlflow/mlflow

#2290 Maybe tone tone down the stale bot?



2 comments

 jpambrun opened on January 10, 2020



lando/lando

#3376 STOP using stale bot



1 comment

 allanaal opened on April 9, 2022



Thesis Organization

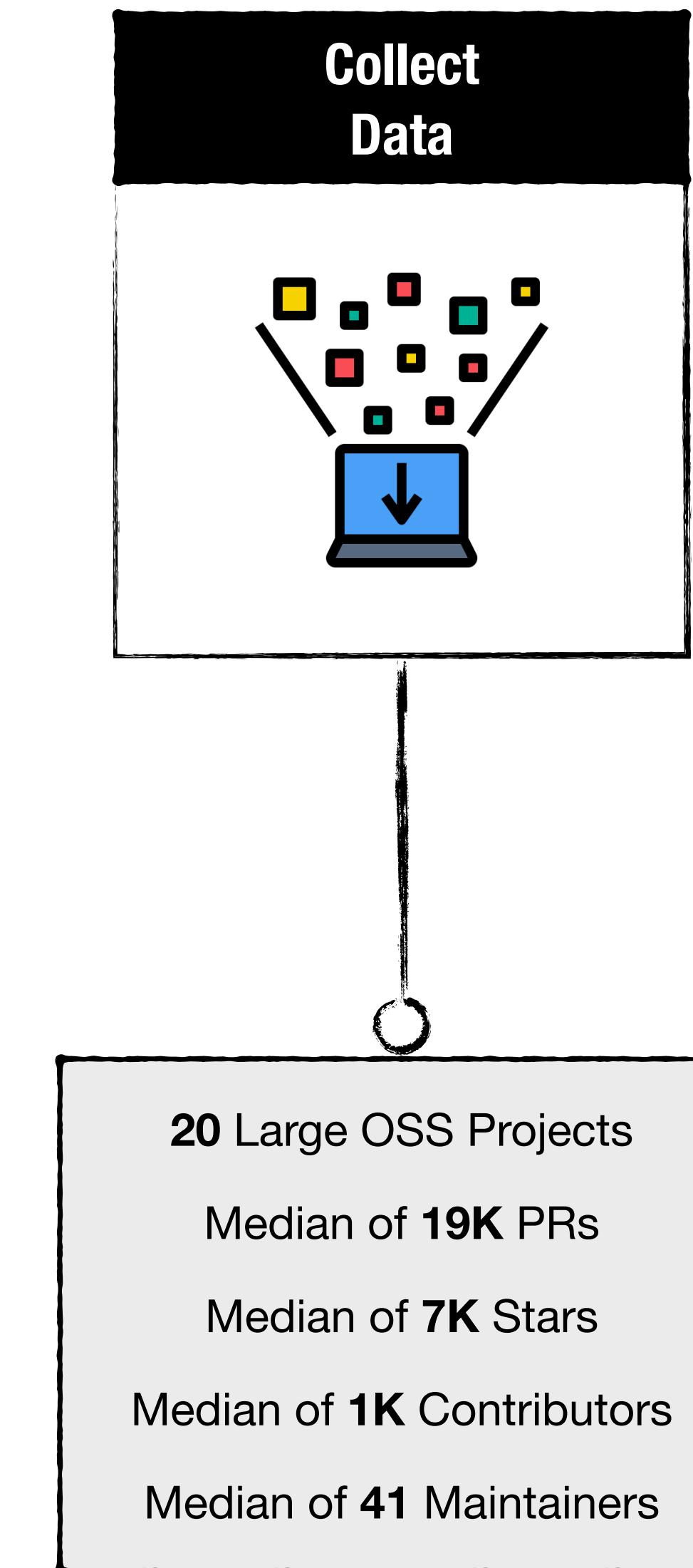
Chapter 3: Understanding the
Dynamics of
Contributor-Abandoned PRs

Chapter 4: Understanding the
Helpfulness of
Stale Bot for PRs

Chapter 5: Predicting the
First Response Latency of
Maintainers and Contributors in PRs

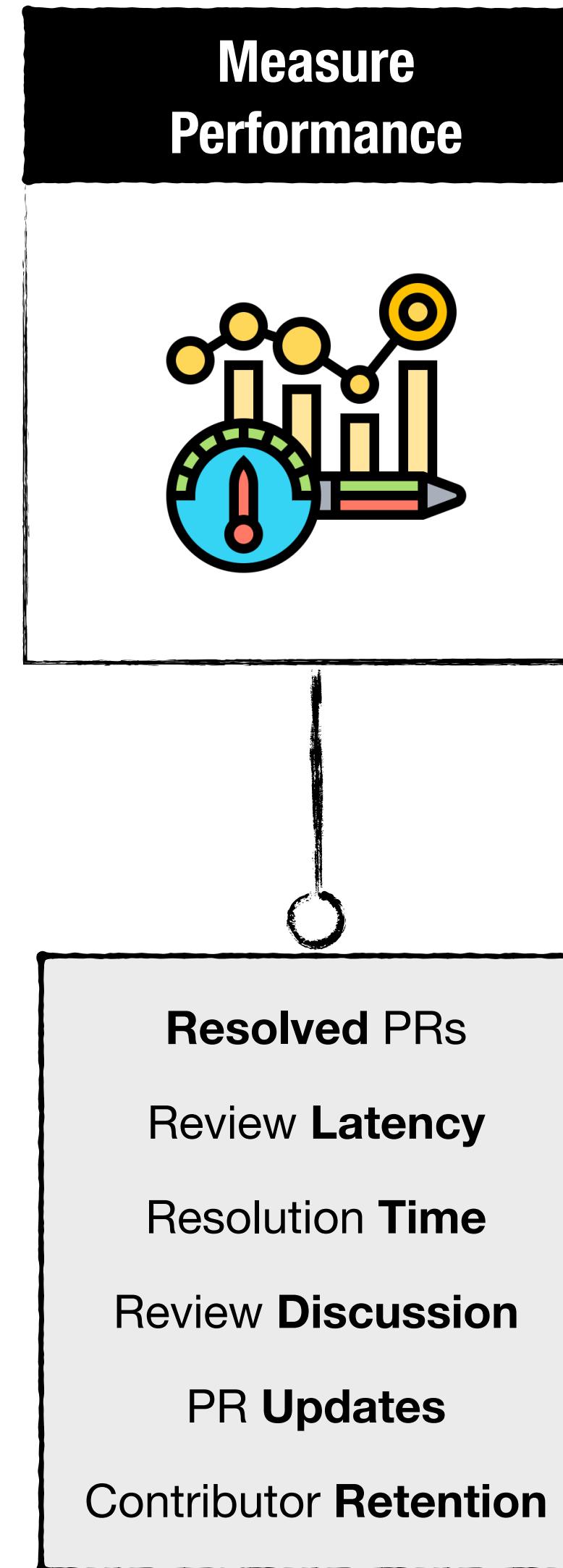
Understanding the **Helpfulness** of Stale Bot

Understanding the **Helpfulness** of Stale Bot

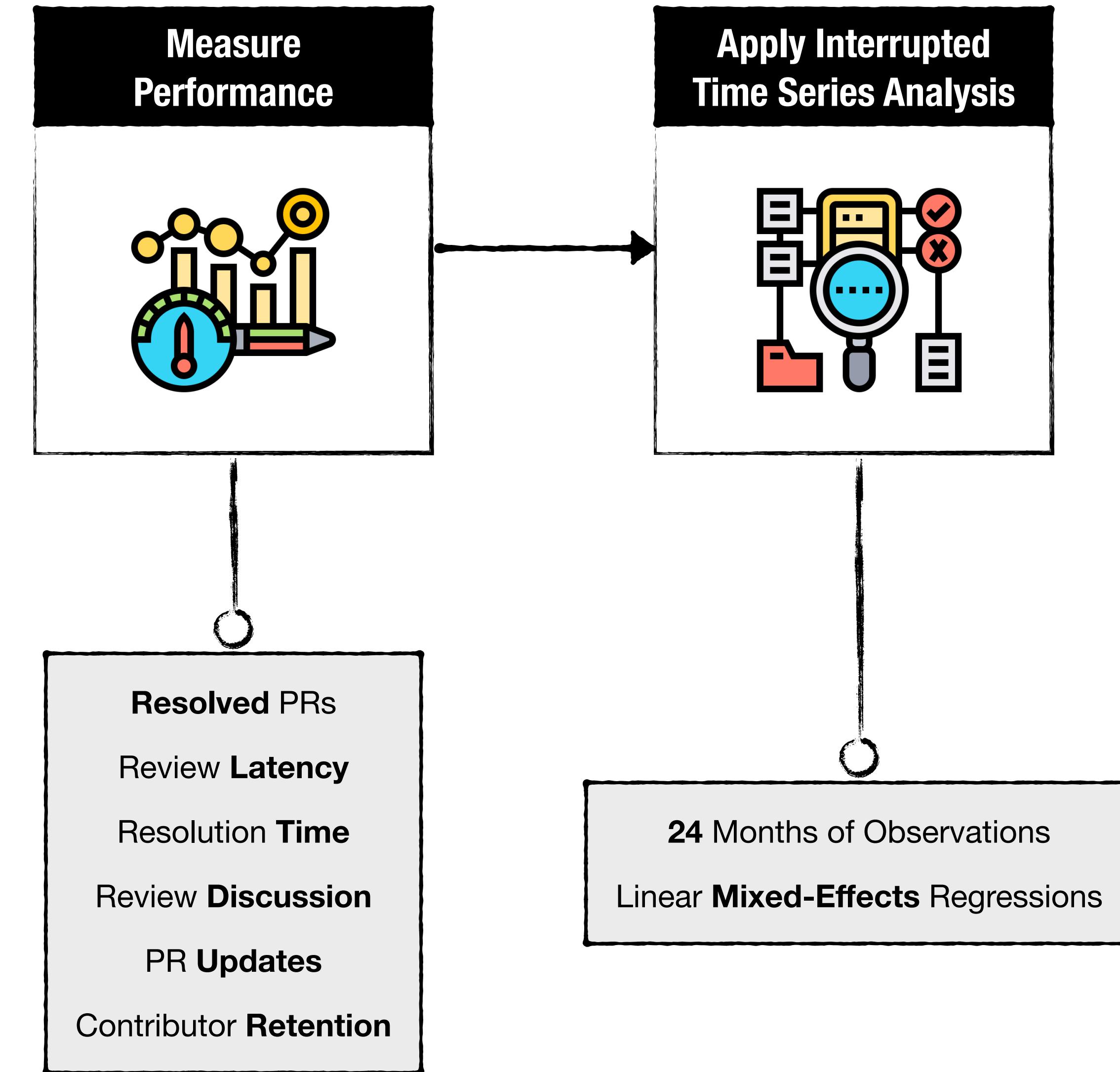


What Is the **Impact** of Stale Bot on Pull-Based Development?

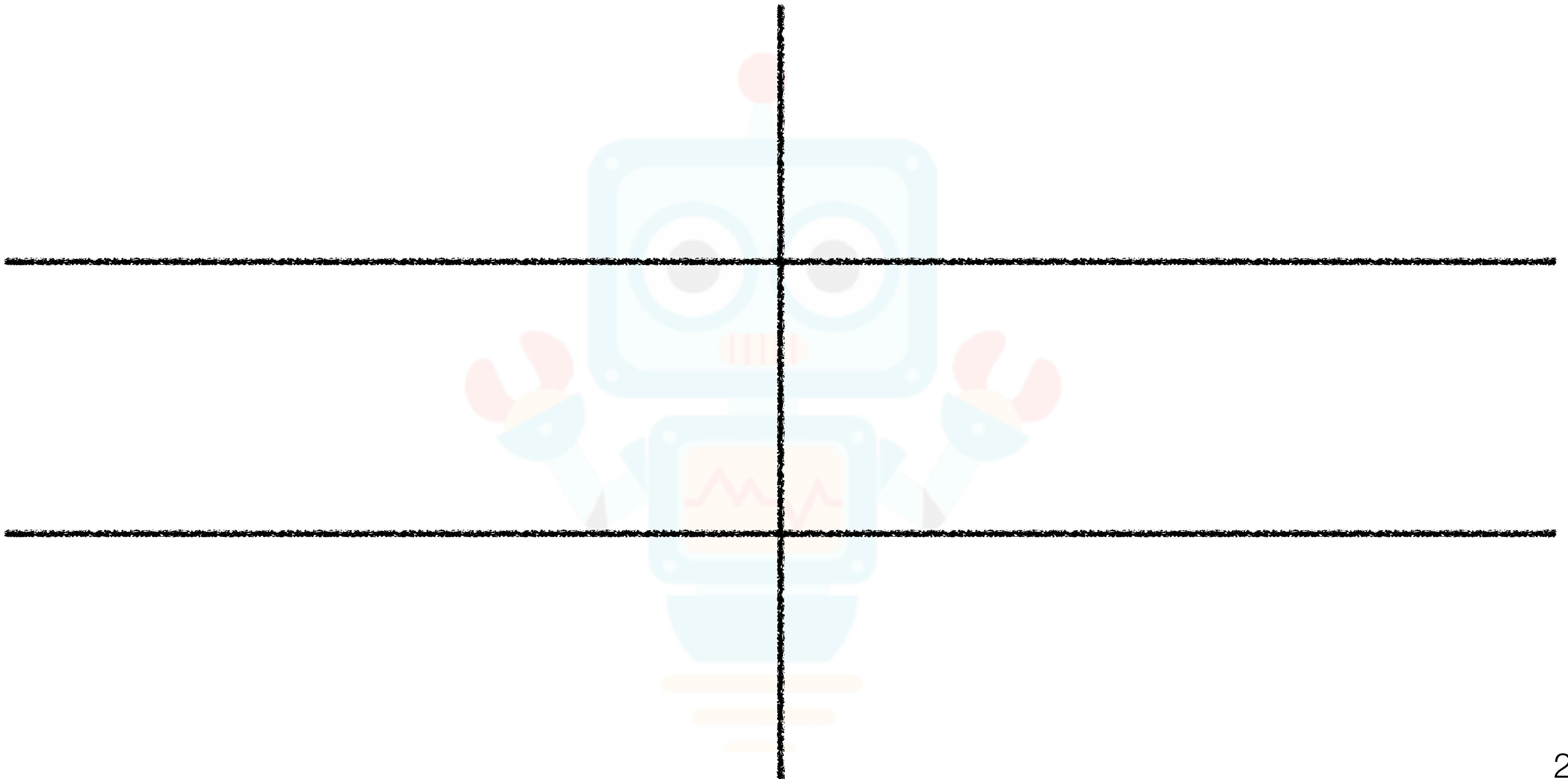
What Is the **Impact** of Stale Bot on Pull-Based Development?



What Is the **Impact** of Stale Bot on Pull-Based Development?



What Is the **Impact** of Stale Bot on Pull-Based Development?



What Is the **Impact** of Stale Bot on Pull-Based Development?

Closed PRs

+15% in 1st Month
-10% in 1st Year

What Is the **Impact** of Stale Bot on Pull-Based Development?

Closed PRs

+15% in 1st Month
-10% in 1st Year

Merged PRs

-24% in 1st Year

What Is the **Impact** of Stale Bot on Pull-Based Development?

More PRs Are Closed in the First Months,
But Fewer Are Resolved Afterward

What Is the **Impact** of Stale Bot on Pull-Based Development?

More PRs Are Closed in the First Months,
But **Fewer** Are Resolved Afterward

Review Latency

-21% in Merged PRs in 1st Year

What Is the **Impact** of Stale Bot on Pull-Based Development?

More PRs Are Closed in the First Months,
But **Fewer** Are Resolved Afterward

Review Latency

-21% in Merged PRs in 1st Year

Resolution Time

-22% in Closed PRs in 12th Month

What Is the **Impact** of Stale Bot on Pull-Based Development?

More PRs Are Closed in the First Months,
But Fewer Are Resolved Afterward

PRs Tend To Have **Faster** Reviews and Resolutions

What Is the **Impact** of Stale Bot on Pull-Based Development?

More PRs Are Closed in the First Months,
But **Fewer** Are Resolved Afterward

PRs Tend To Have **Faster** Reviews and Resolutions

PR Updates

-11% in Merged PRs in 1st Year

What Is the **Impact** of Stale Bot on Pull-Based Development?

More PRs Are Closed in the First Months,
But **Fewer** Are Resolved Afterward

PRs Tend To Have **Faster** Reviews and Resolutions

PR Updates

-11% in Merged PRs in 1st Year

Active Contributors

-14% in 1st Year

What Is the **Impact** of Stale Bot on Pull-Based Development?

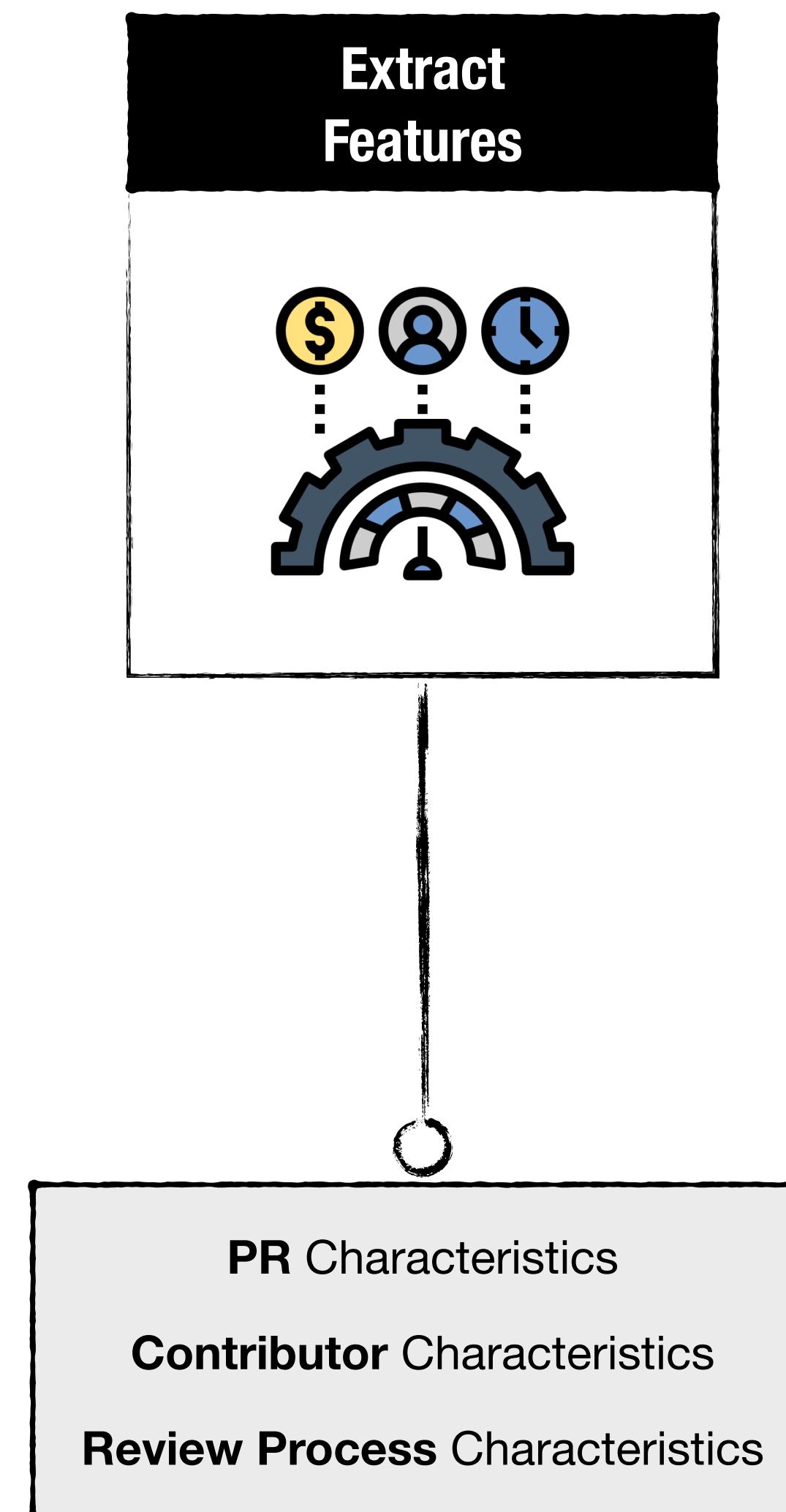
More PRs Are Closed in the First Months,
But Fewer Are Resolved Afterward

PRs Tend To Have **Faster** Reviews and Resolutions

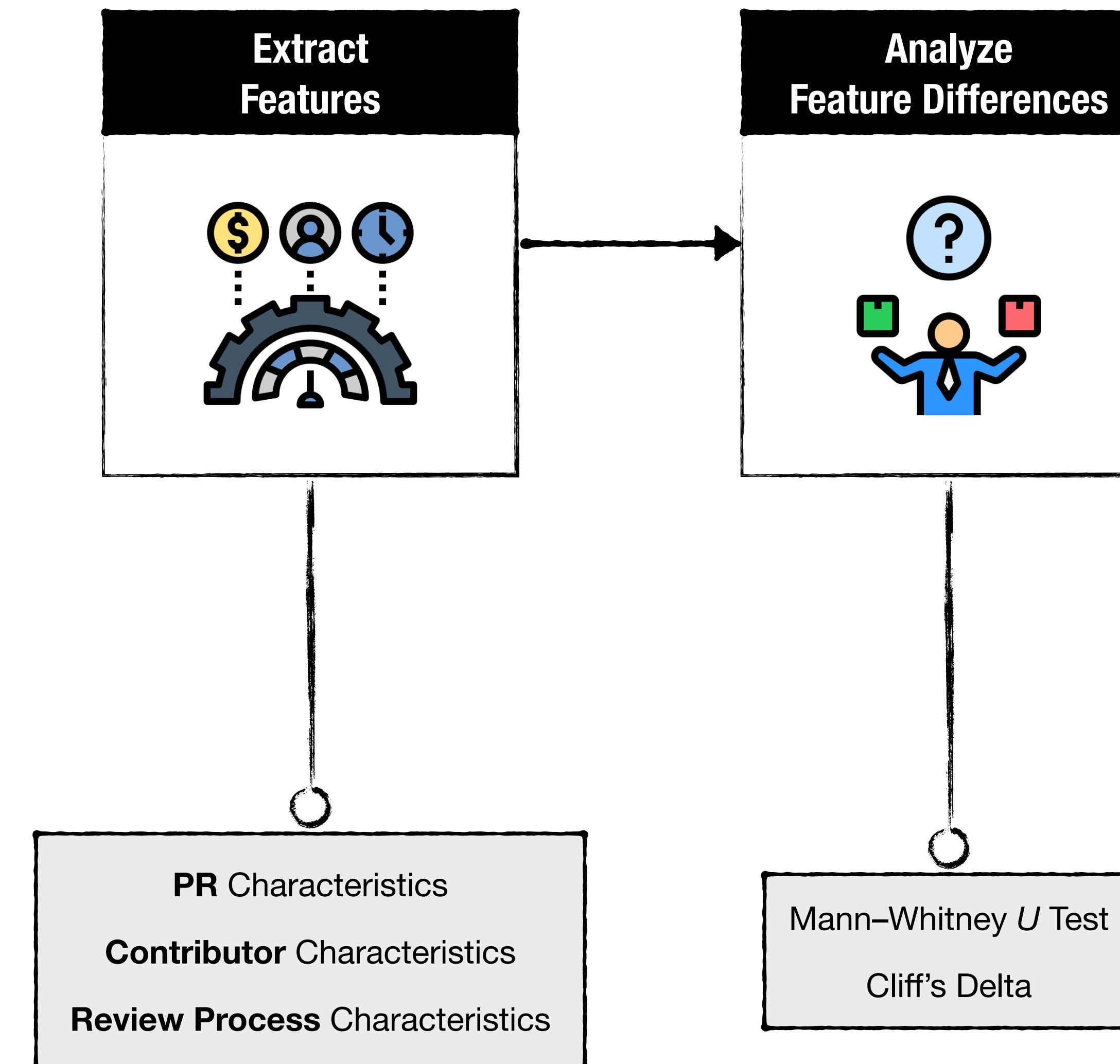
The Adoption **Negatively** Affects Contributor Retention

What **Kind** of PRs Are Usually Intervened by Stale Bot?

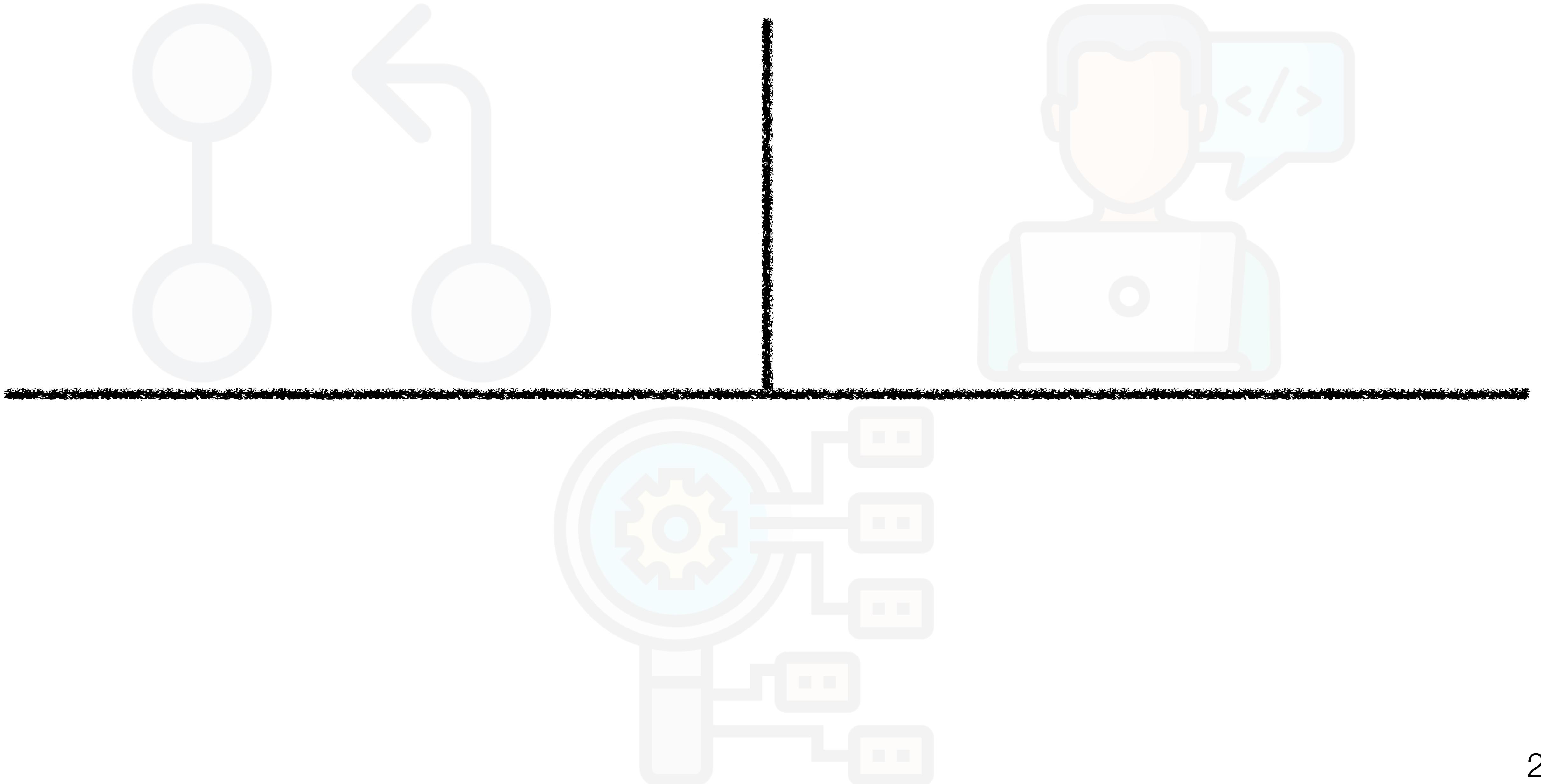
What **Kind** of PRs Are Usually Intervened by Stale Bot?



What **Kind** of PRs Are Usually Intervened by Stale Bot?



What **Kind** of PRs Are Usually Intervened by Stale Bot?



What **Kind** of PRs Are Usually Intervened by Stale Bot?

More Complex PRs

- Lengthier Descriptions
- More Followup Commits
- More Followup Changed Lines
- More Followup Changed Files

What **Kind** of PRs Are Usually Intervened by Stale Bot?

More Complex PRs

- Lengthier Descriptions
- More Followup Commits
- More Followup Changed Lines
- More Followup Changed Files

Less Experienced Contributors

- Fewer Submitted PRs
- Lower Acceptance Rate
- Lower Contribution Period

What **Kind** of PRs Are Usually Intervened by Stale Bot?

More Complex PRs

- Lengthier Descriptions
- More Followup Commits
- More Followup Changed Lines
- More Followup Changed Files

Less Experienced Contributors

- Fewer Submitted PRs
- Lower Acceptance Rate
- Lower Contribution Period

Lengthier Review Processes

- More Participants
- More Discussion Comments
- Higher First Response Latency
- Higher Mean Response Latency

What **Kind** of PRs Are Usually Intervened by Stale Bot?

More Complex PRs

Lengthier Descriptions

More Followup Commits

More Followup Changed Lines

More Followup Changed Files

Less Experienced Contributors

Fewer Submitted PRs

Lower Acceptance Rate

Lower Contribution Period

Lengthier Review Processes

More Participants

More Discussion Comments

Higher First Response Latency

Higher Mean Response Latency

Chapter4: Understanding the Helpfulness of Stale Bot for PRs



Understanding the Helpfulness of Stale Bot for Pull-Based Development: An Empirical Study of 20 Large Open-Source Projects

SAYEDHASSAN KHATOONABADI and DIEGO ELIAS COSTA, Concordia University, Canada
SUHAIB MUJAHID, Mozilla Corporation, Canada
EMAD SHIHAB, Concordia University, Canada

Pull Requests (PRs) that are neither progressed nor resolved clutter the list of PRs, making it difficult for the maintainers to manage and prioritize unresolved PRs. To automatically track, follow up, and close such inactive PRs, Stale bot was introduced by GitHub. Despite its increasing adoption, there are ongoing debates on whether using Stale bot alleviates or exacerbates the problem of inactive PRs. To better understand if and how Stale bot helps projects in their pull-based development workflow, we perform an empirical study of 20 large and popular open source projects. We find that Stale bot can help deal with a backlog of unresolved PRs, as the projects closed more PRs within the first few months of adoption. Moreover, Stale bot can help improve the efficiency of the PR review process as the projects reviewed PRs that ended up merged and resolved PRs that ended up closed faster after the adoption. However, Stale bot can also negatively affect the contributors, as the projects experienced a considerable decrease in their number of active contributors after the adoption. Therefore, relying solely on Stale bot to deal with inactive PRs may lead to decreased community engagement and an increased probability of contributor abandonment.

CCS Concepts: • Human-centered computing → Empirical studies in collaborative and social computing; Open source software; • Software and its engineering → Collaboration in software development; Open source model;

Additional Key Words and Phrases: Software development bots, pull request abandonment, pull-based development, modern code review, social coding platforms, open source software

ACM Reference format:

SayedHassan Khatoonabadi, Diego Elias Costa, Suhaib Mujahid, and Emad Shihab. 2023. Understanding the Helpfulness of Stale Bot for Pull-Based Development: An Empirical Study of 20 Large Open-Source Projects. *ACM Trans. Softw. Eng. Methodol.* 33, 2, Article 36 (December 2023), 43 pages.
<https://doi.org/10.1145/3624739>

Authors' addresses: S. Khatoonabadi and E. Shihab, Data-driven Analysis of Software (DAS) Lab, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC, H3H 2L9, Canada; e-mails: sayedhassan.khatoonabadi@mail.concordia.ca, emad.shihab@concordia.ca; D. E. Costa, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC, H3H 2L9, Canada; e-mail: diego.costa@concordia.ca; S. Mujahid, Mozilla Corporation, Montreal, QC, Canada; e-mail: smujahid@mozilla.com. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

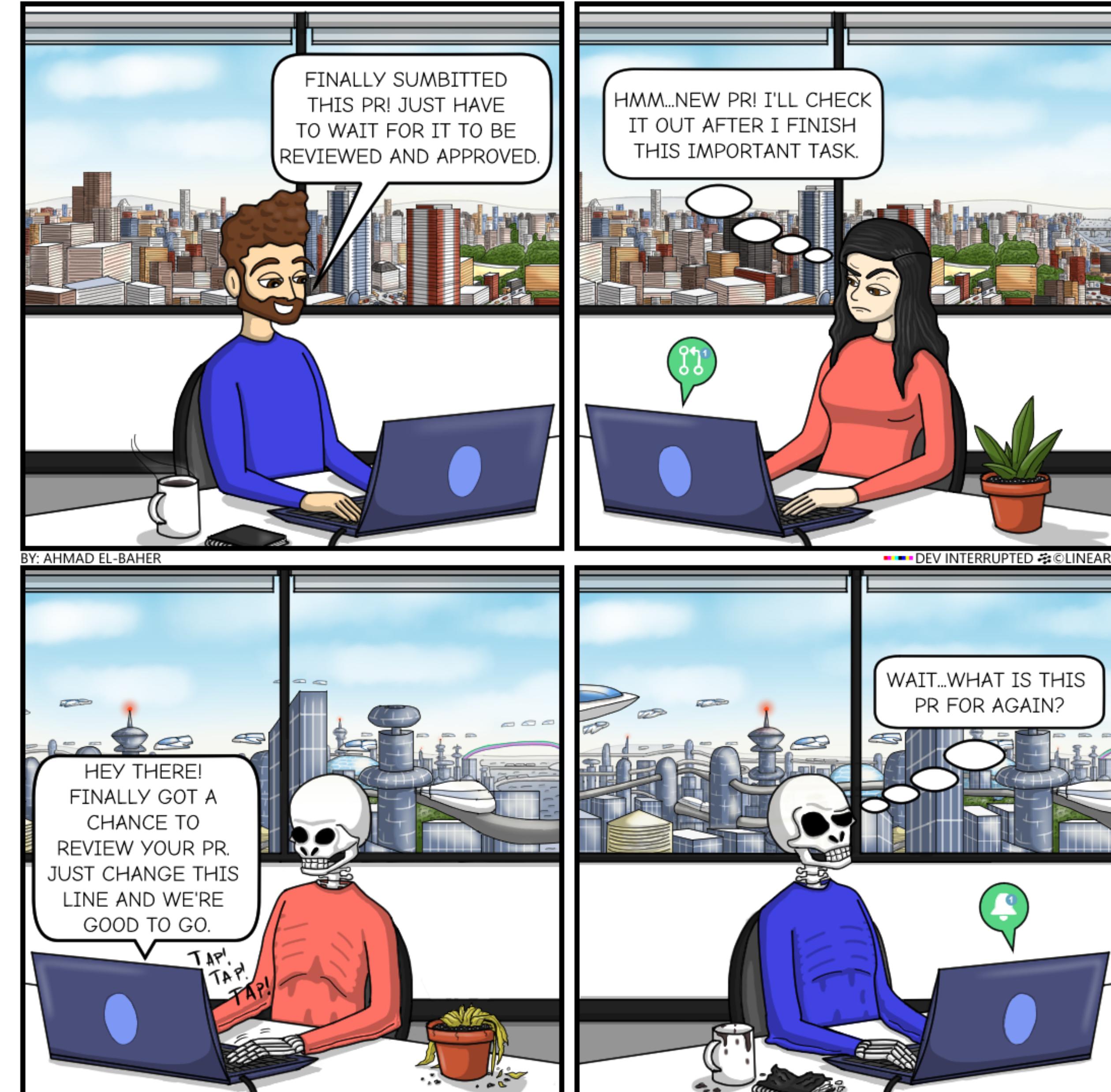
1049-331X/2023/12-ART36 \$15.00

<https://doi.org/10.1145/3624739>

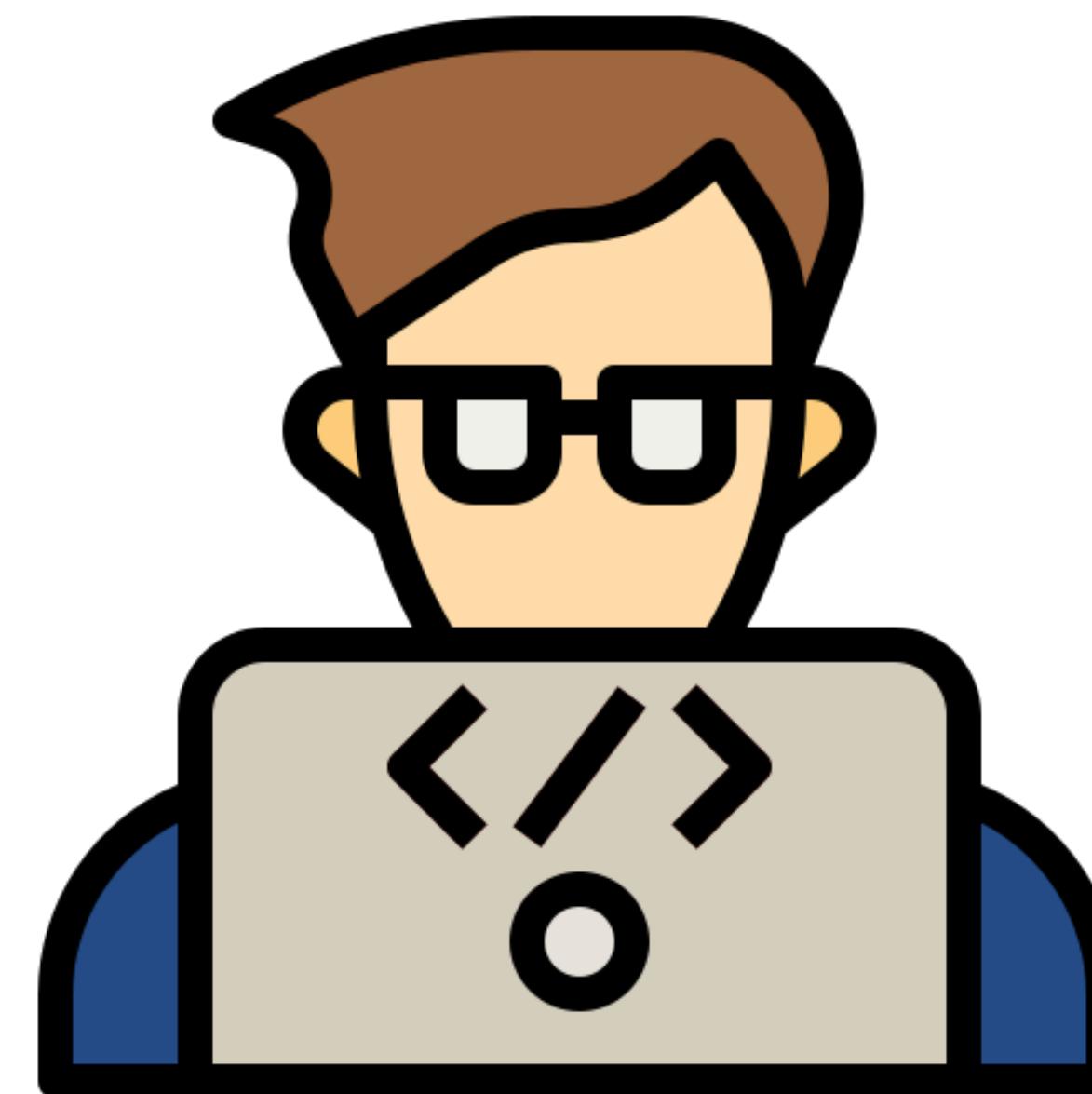
Despite its helpfulness in dealing with a backlog of unresolved PRs, Stale bot tends to further aggravate PR abandonment and decrease contributor engagement.

[Khatoonabadi et al., TOSEM 2023]

PR Success Depends on the Responsiveness of **Both** Maintainers and Contributors

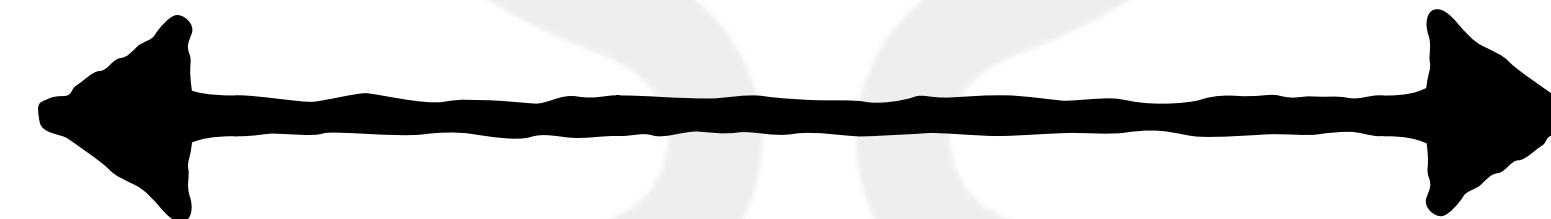


Anticipating Waiting Times **Facilitates** Collaboration

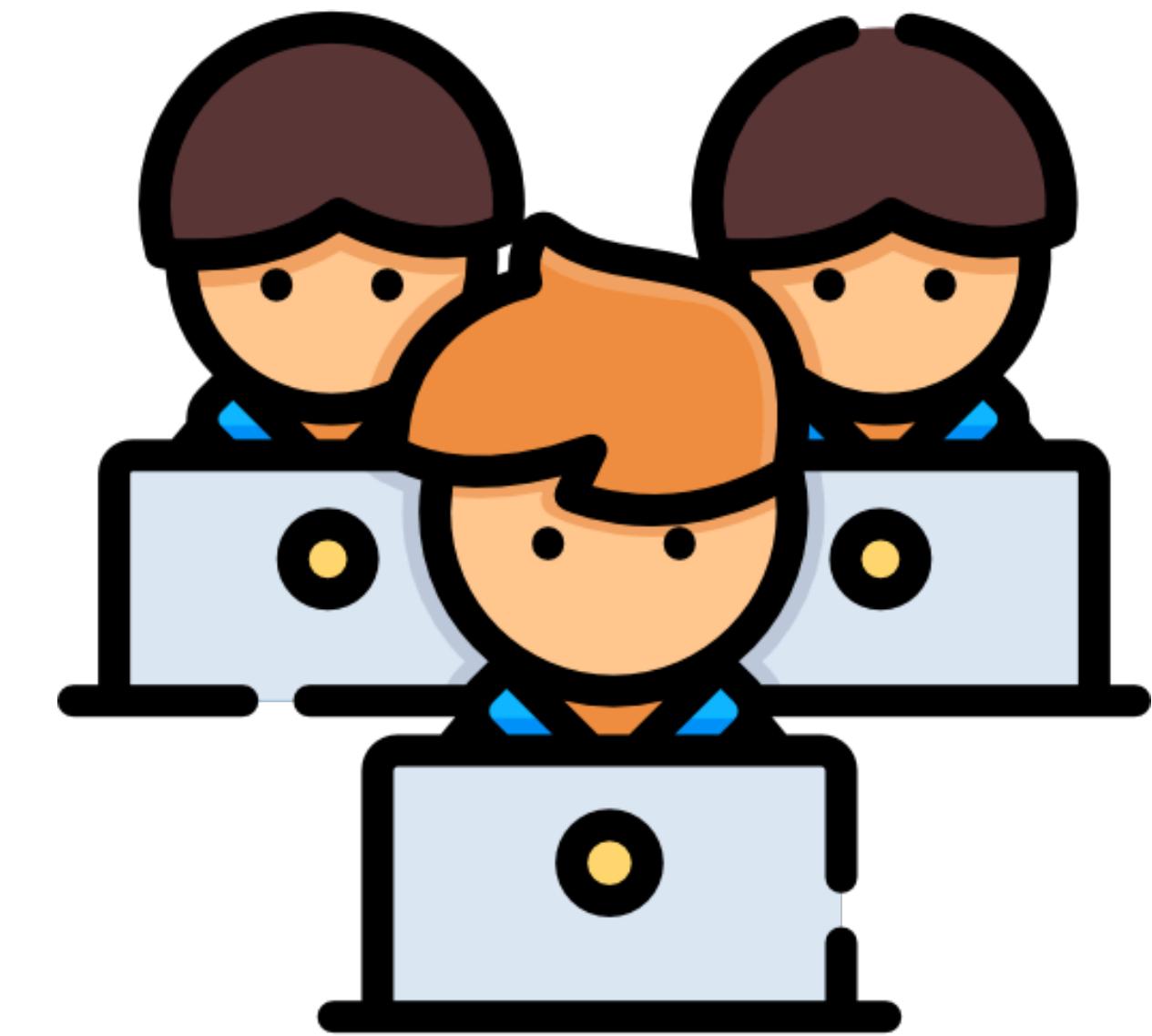


Contributor

Estimated Response Latencies



Potential Challenges



Maintainers

Thesis Organization

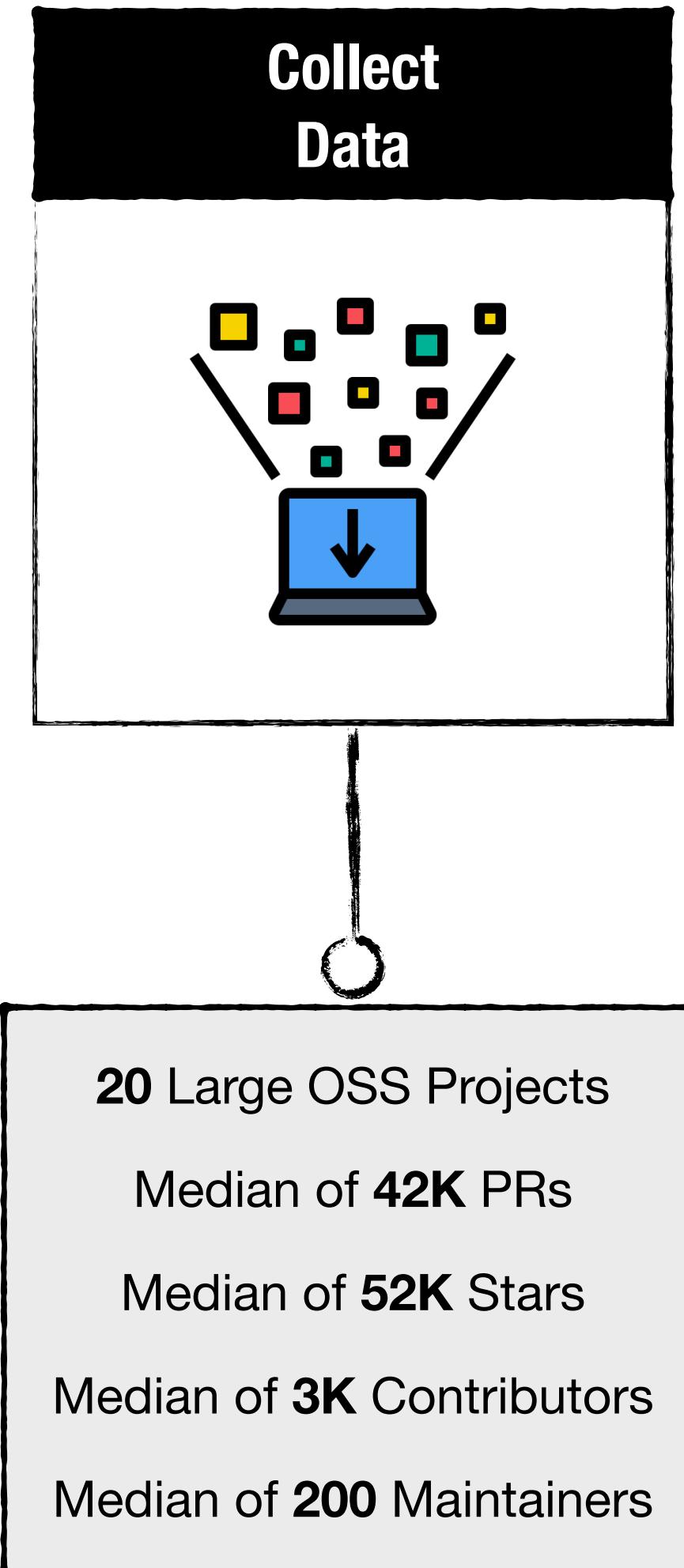
Chapter 3: Understanding the
Dynamics of
Contributor-Abandoned PRs

Chapter 4: Understanding the
Helpfulness of
Stale Bot for PRs

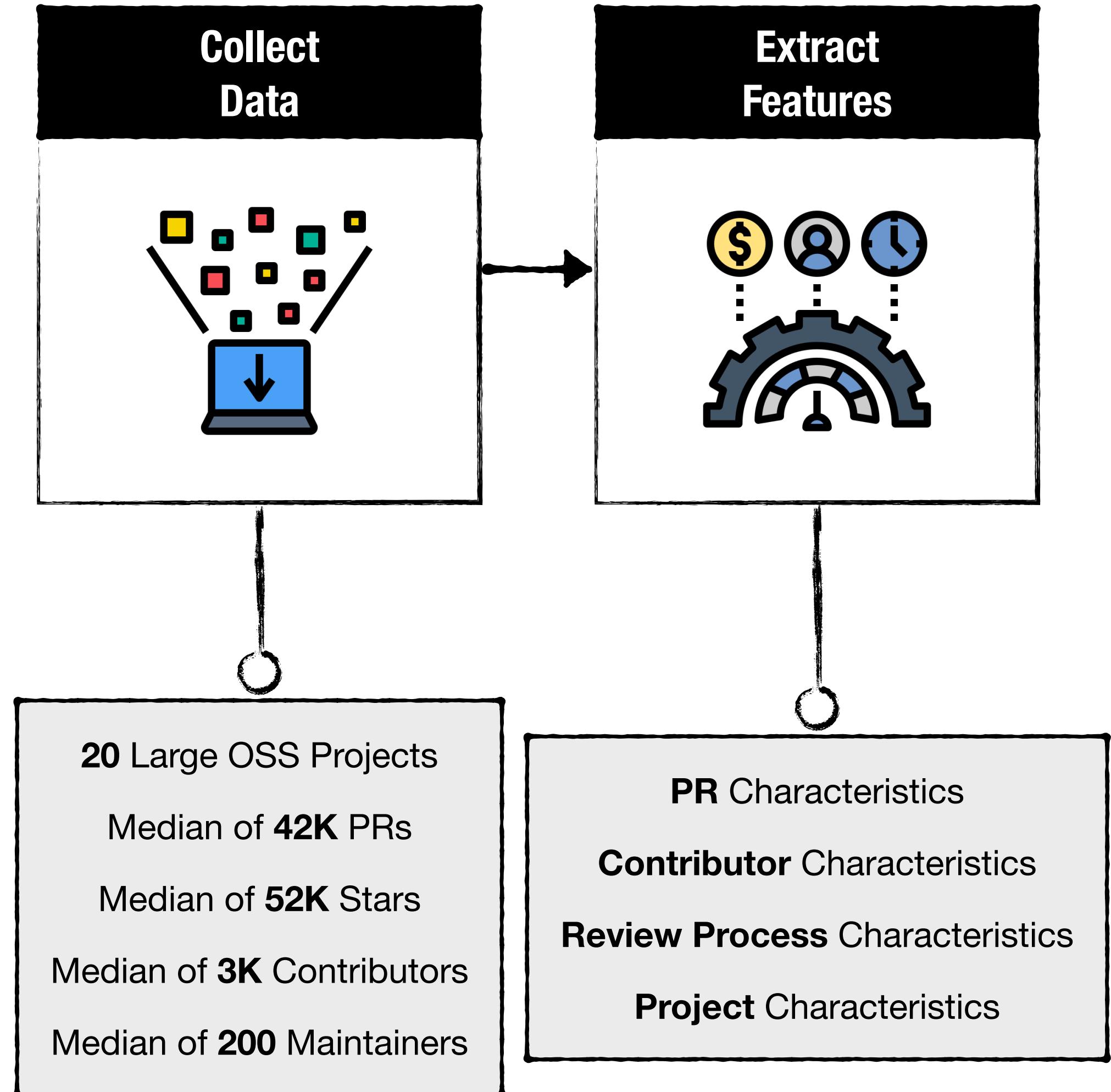
Chapter 5: Predicting the
First Response Latency of
Maintainers and Contributors in PRs

Predicting the **First Response Latency** of Maintainers and Contributors in PRs

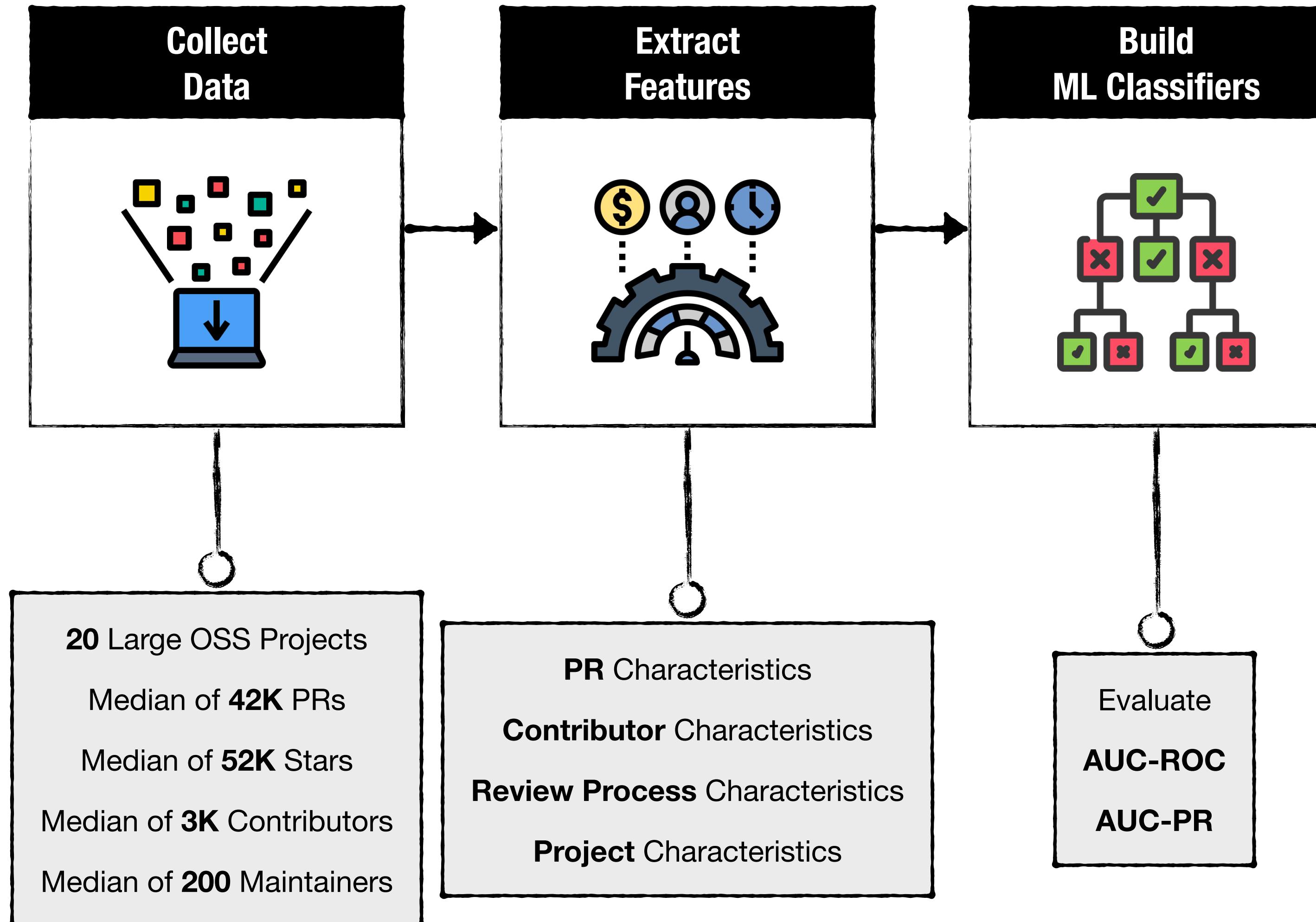
Predicting the **First Response Latency** of Maintainers and Contributors in PRs



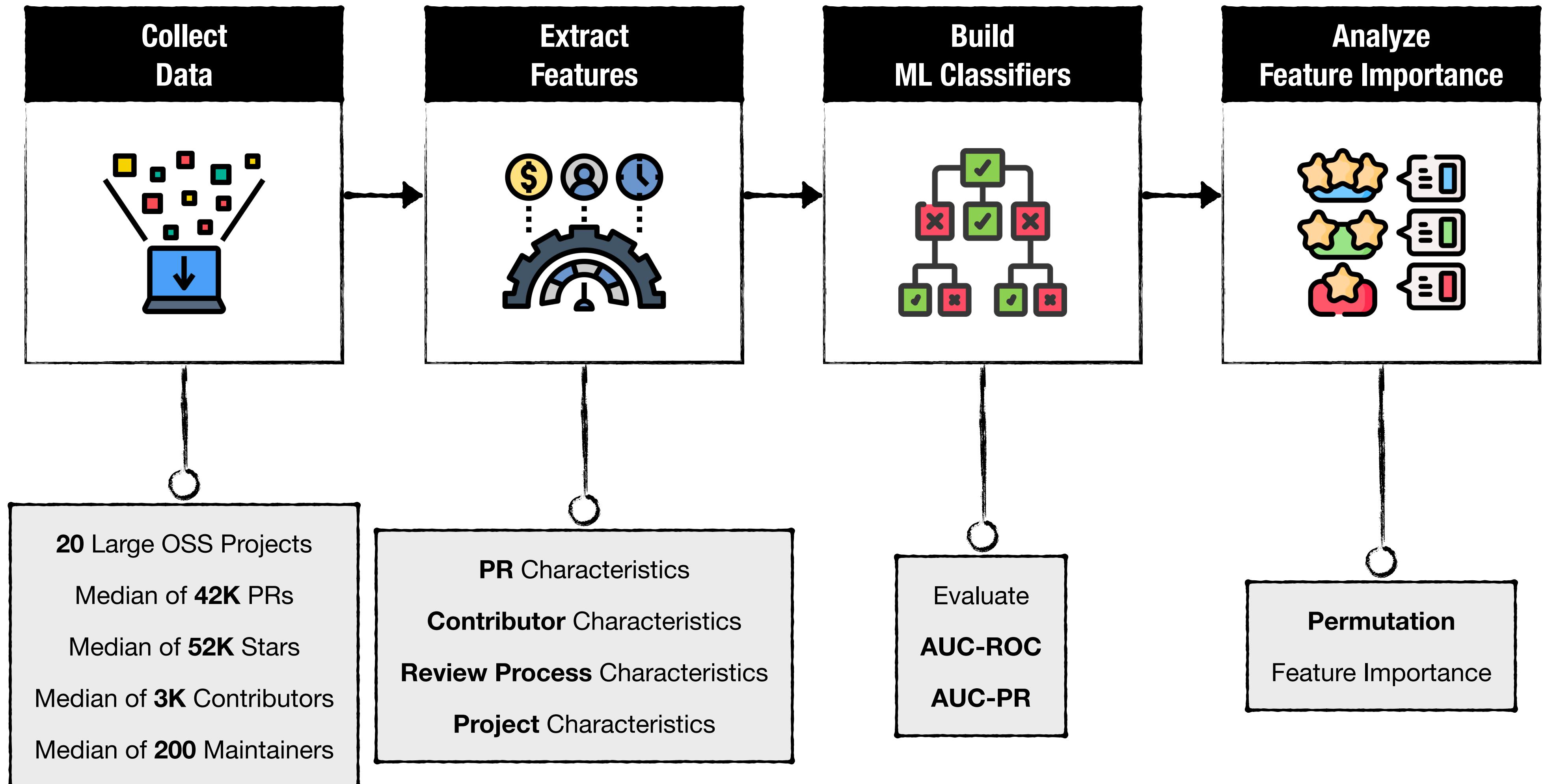
Predicting the First Response Latency of Maintainers and Contributors in PRs



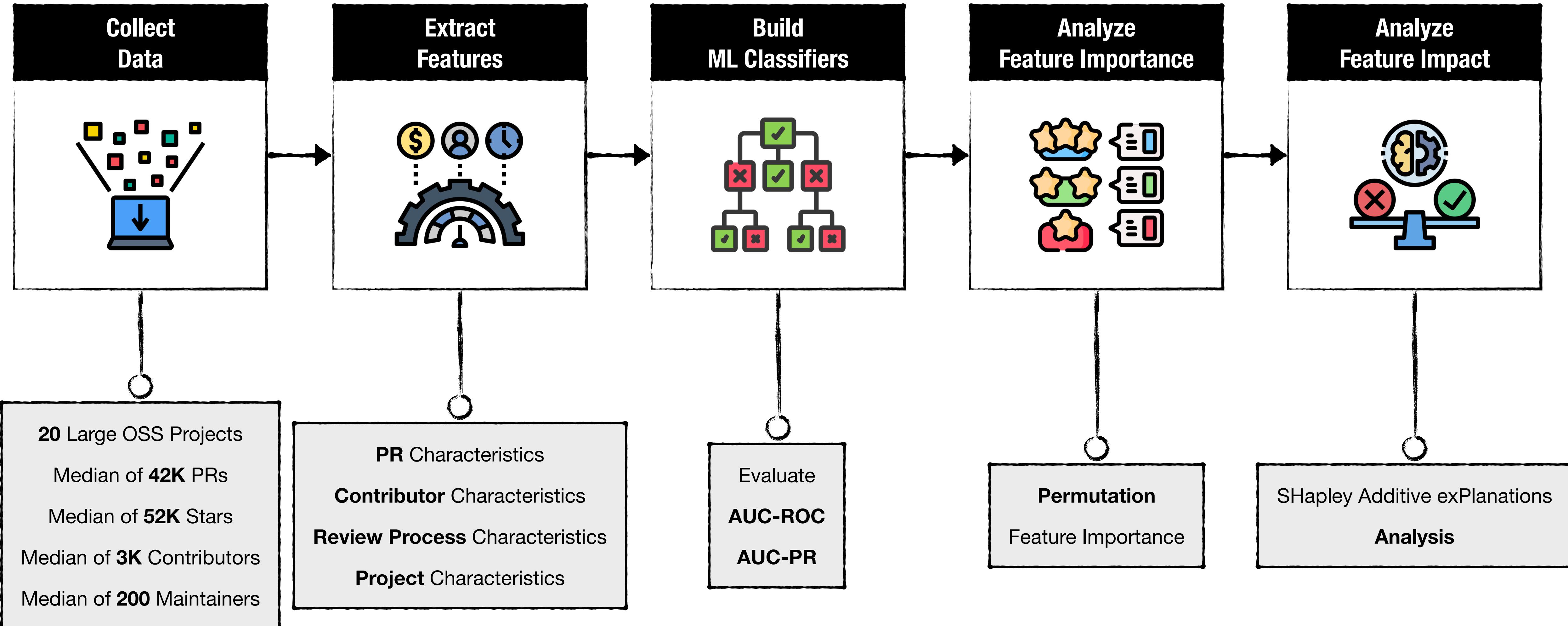
Predicting the First Response Latency of Maintainers and Contributors in PRs



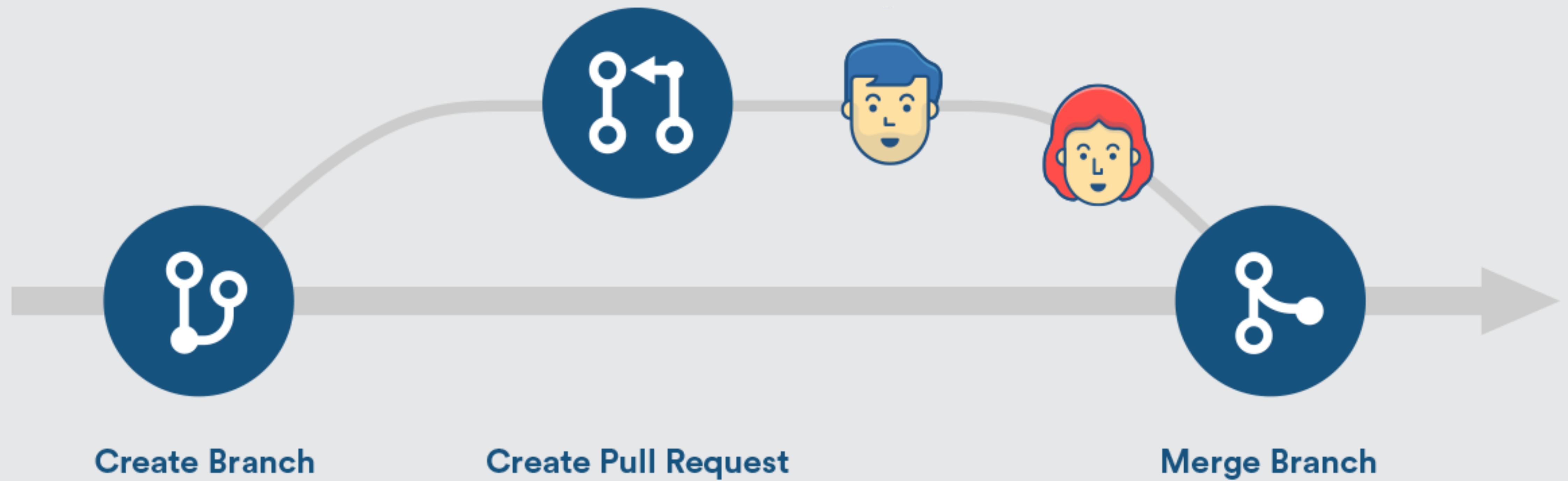
Predicting the First Response Latency of Maintainers and Contributors in PRs



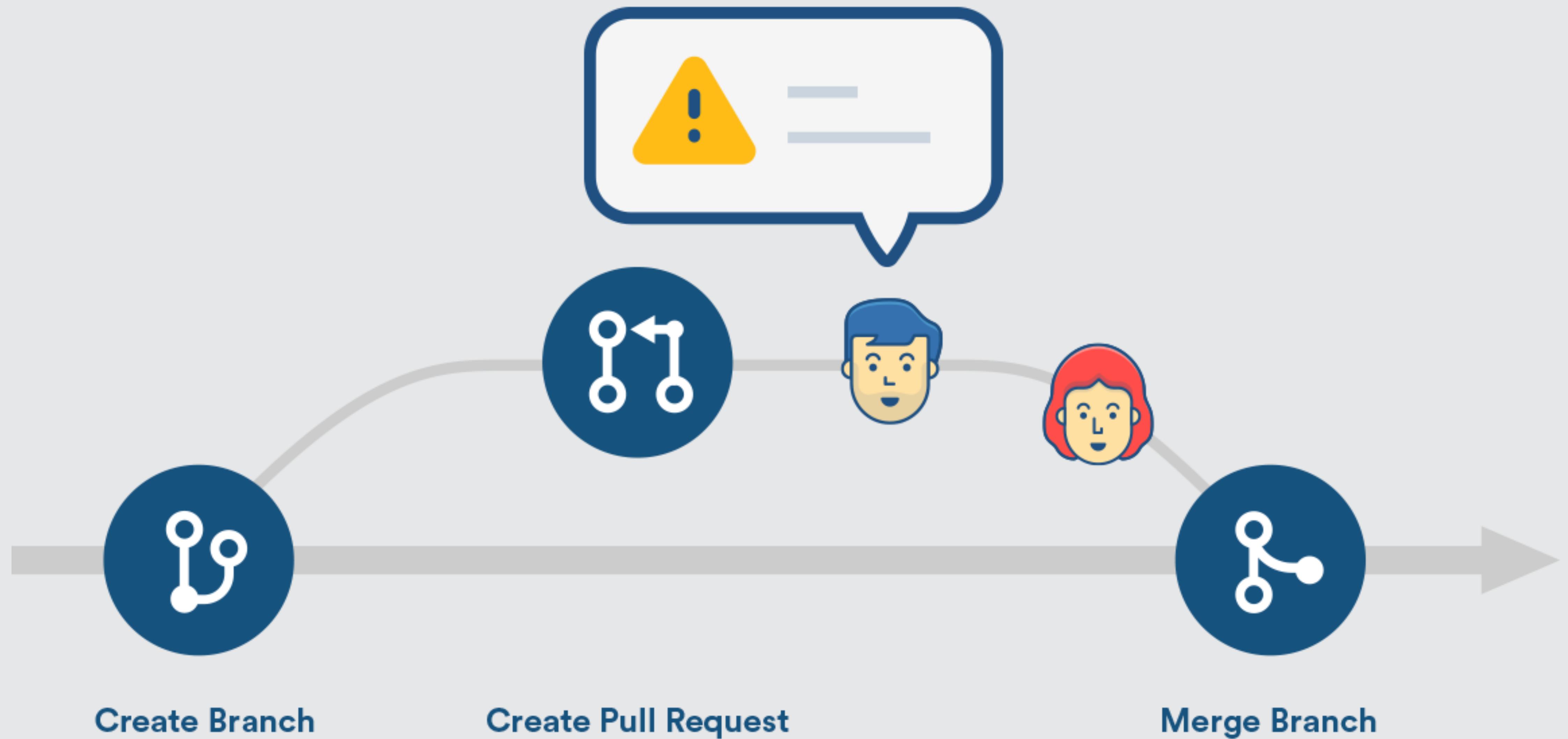
Predicting the First Response Latency of Maintainers and Contributors in PRs



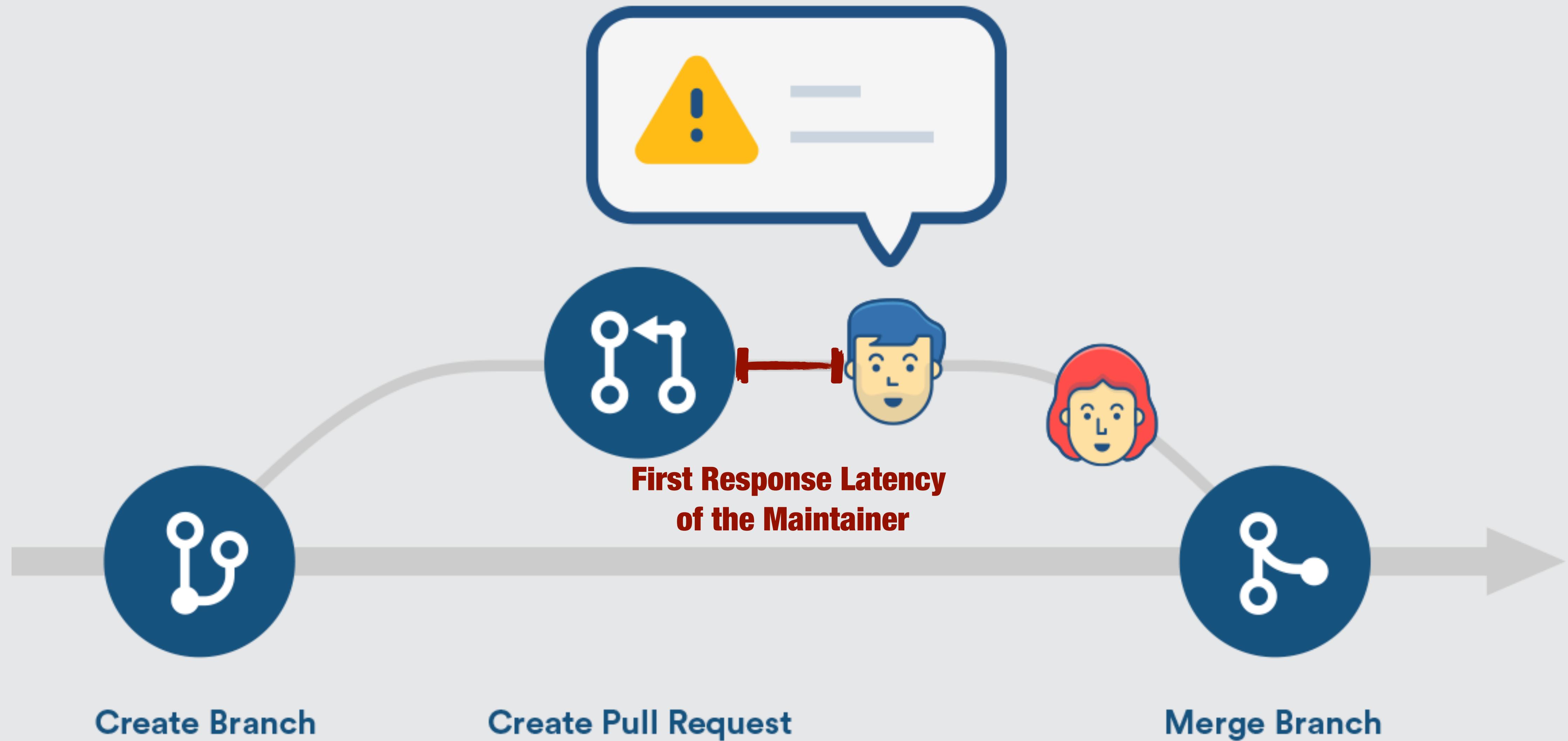
Predicting the **First Response Latency** of Maintainers and Contributors in PRs



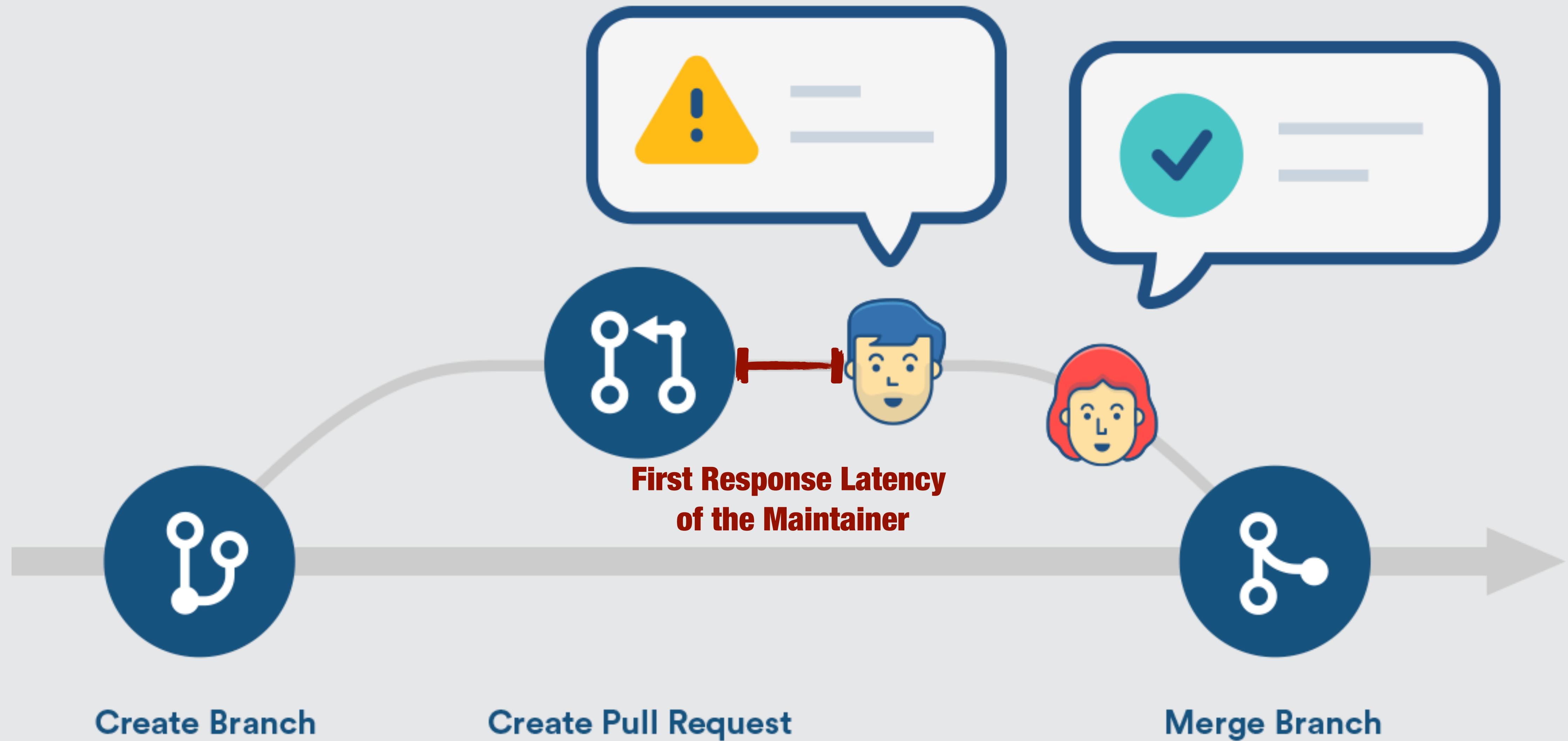
Predicting the **First Response Latency** of Maintainers and Contributors in PRs



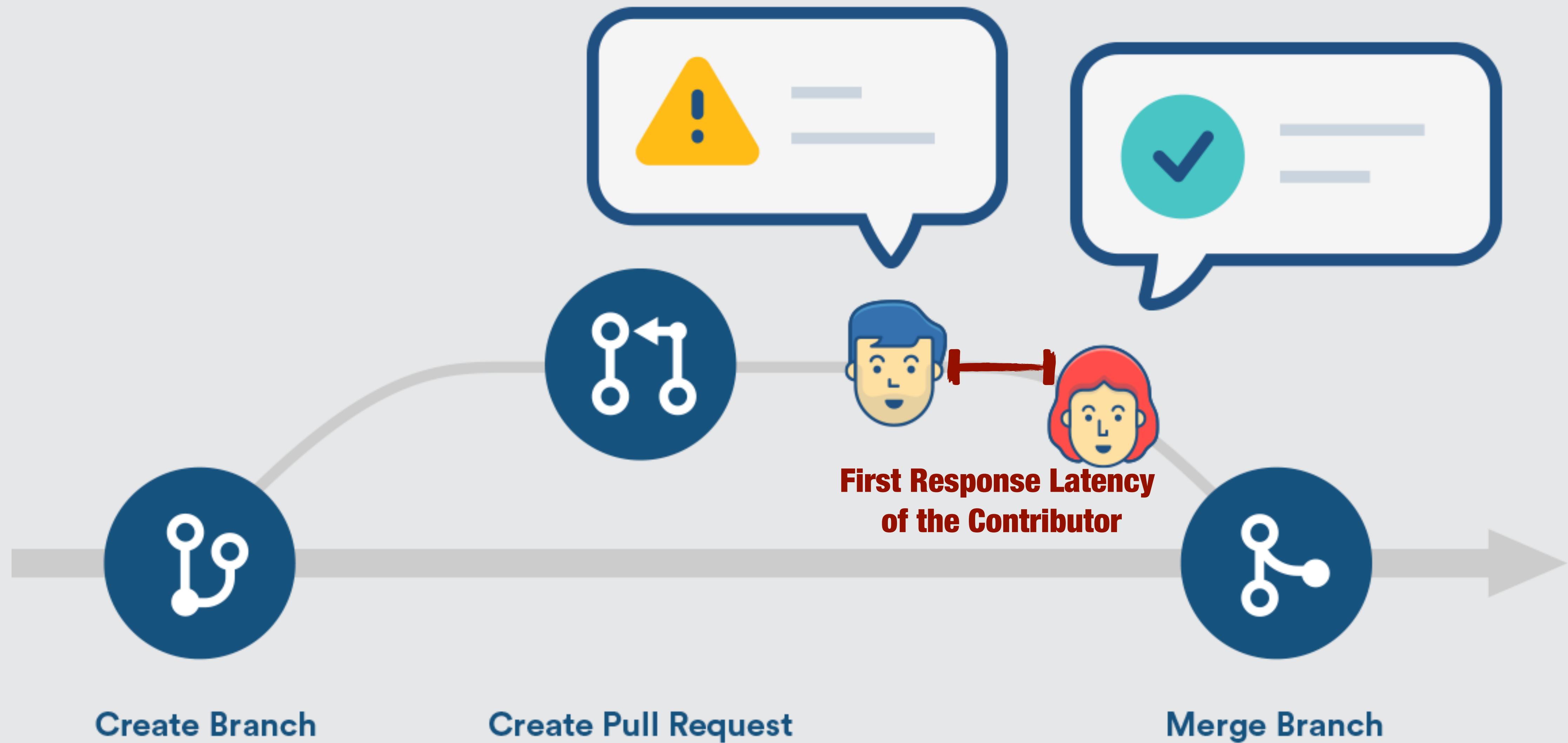
Predicting the First Response Latency of Maintainers and Contributors in PRs



Predicting the First Response Latency of Maintainers and Contributors in PRs

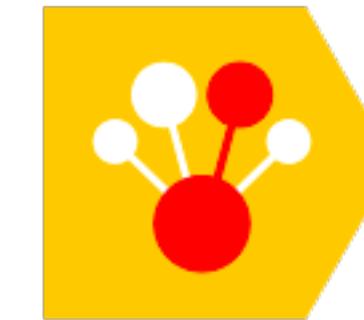


Predicting the First Response Latency of Maintainers and Contributors in PRs



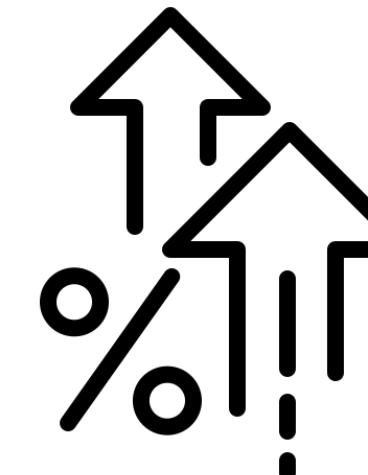
First Response Latency of **Maintainers**

First Response Latency of **Maintainers**

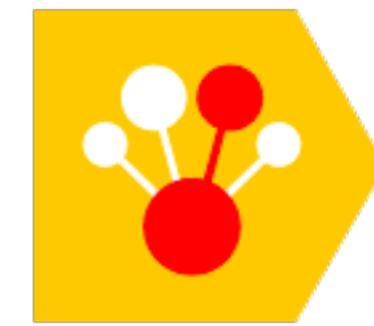


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

Two black arrows pointing upwards, one above the other, indicating an increase in both AUC-ROC and AUC-PR values.

First Response Latency of **Maintainers**



CatBoost

AUC-ROC: +29
AUC-PR: +51 %

High

Value

Low

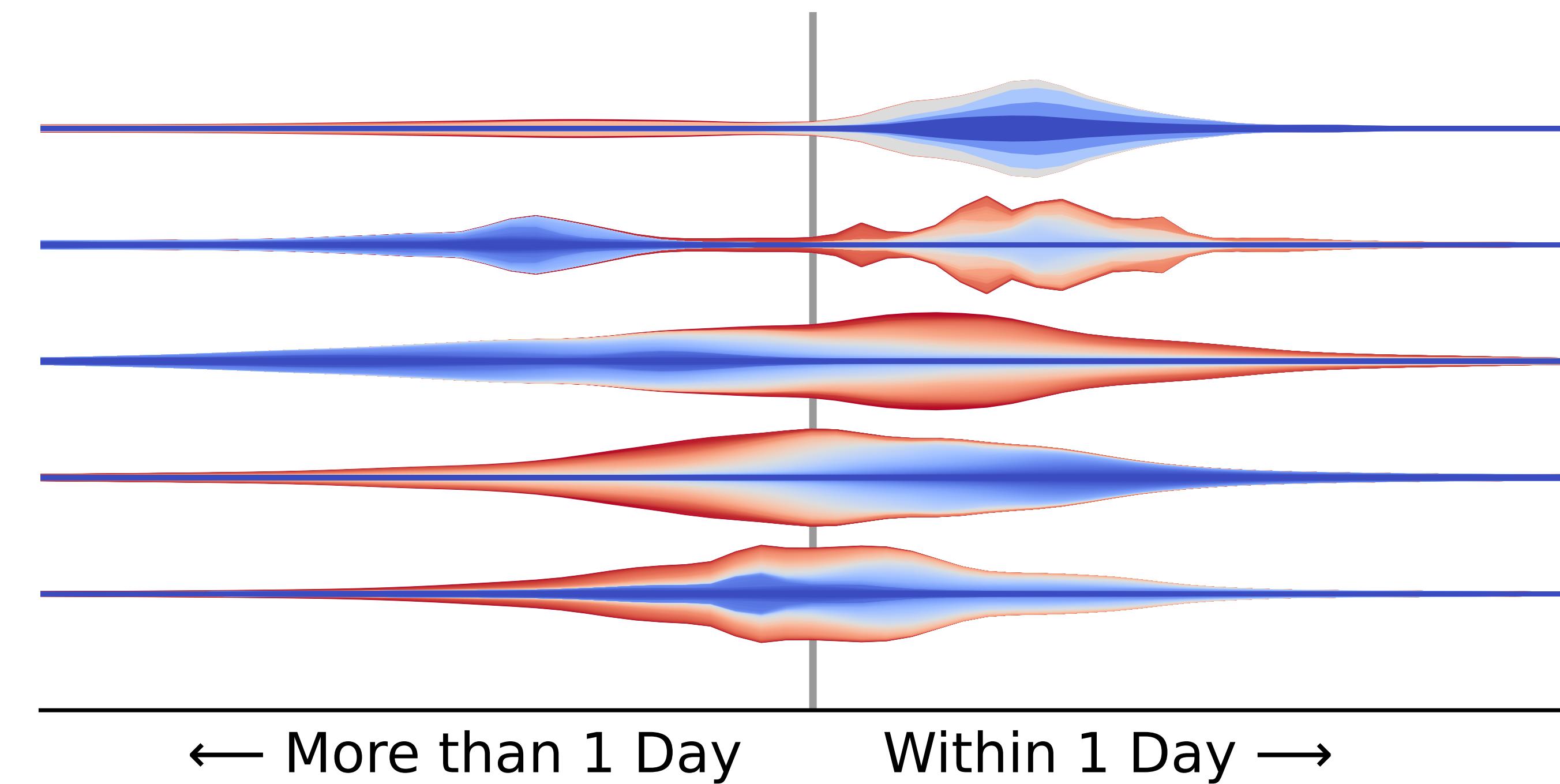
Submission Day

Commits

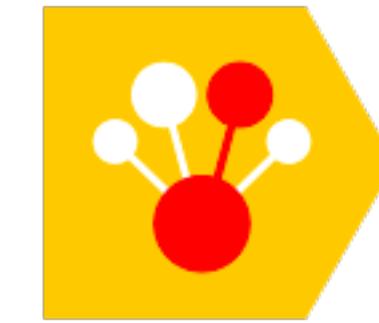
Contributor Performance

Description Length

Contributor Responsiveness

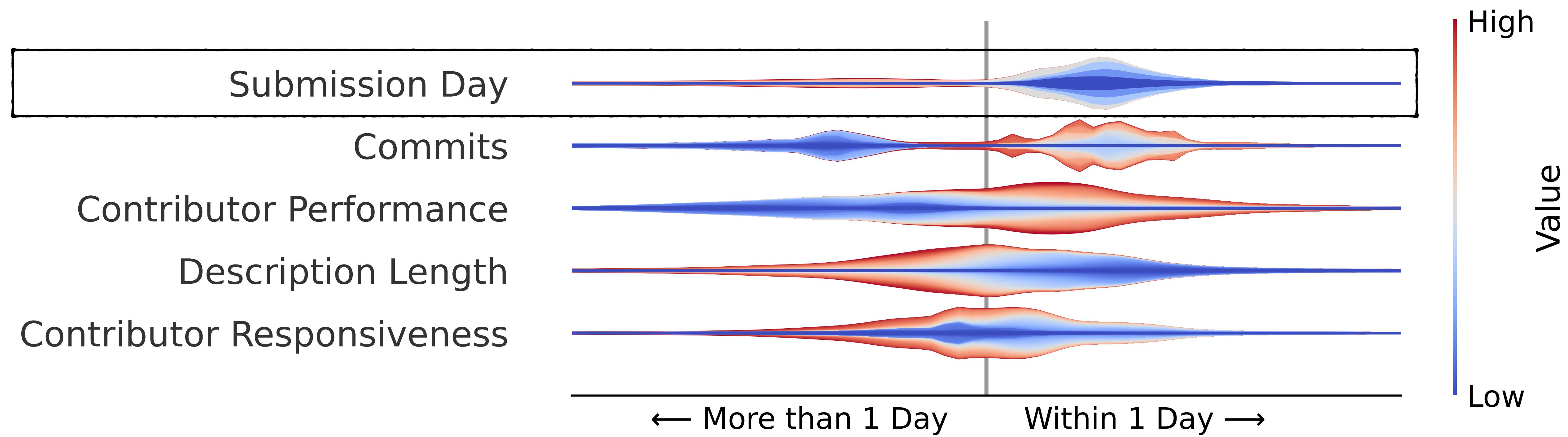


First Response Latency of **Maintainers**

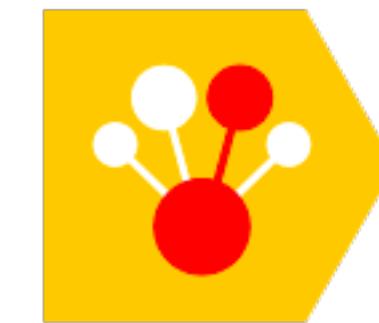


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

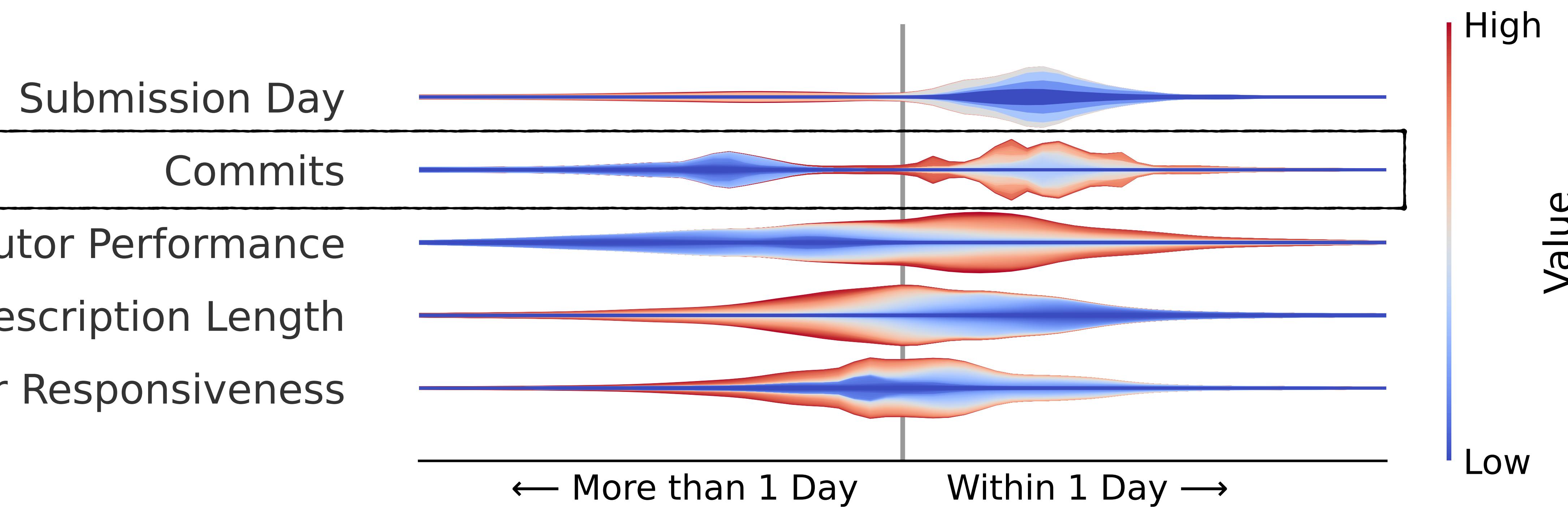


First Response Latency of **Maintainers**

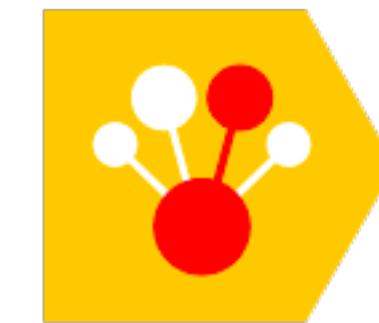


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

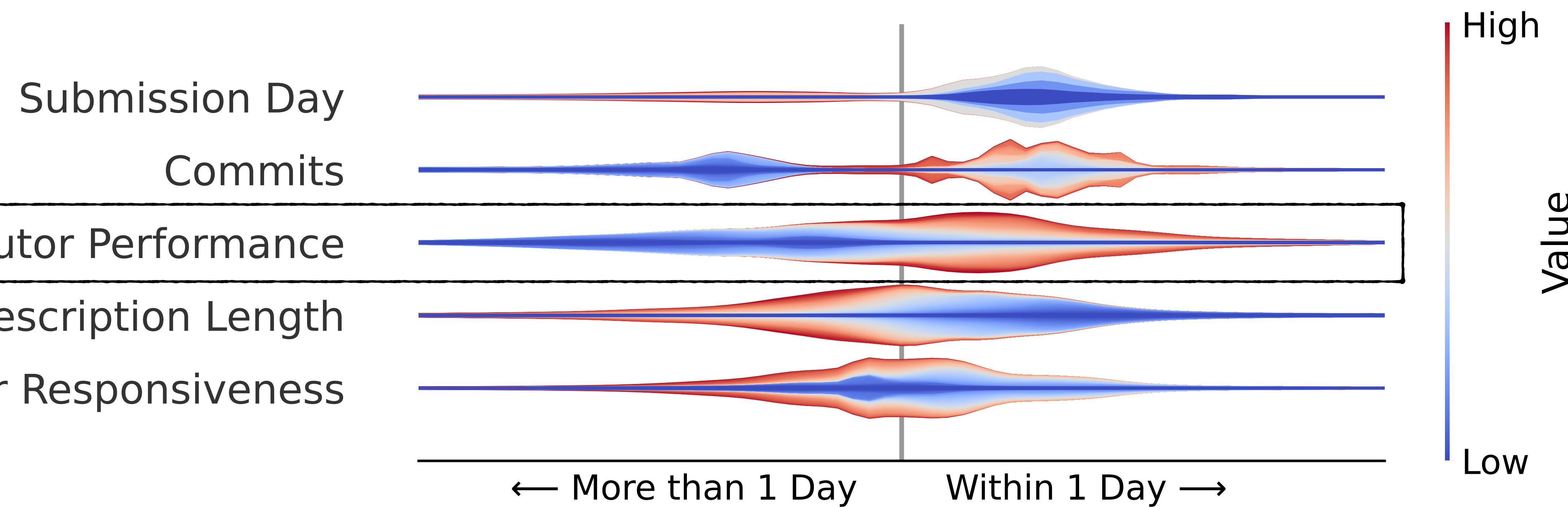
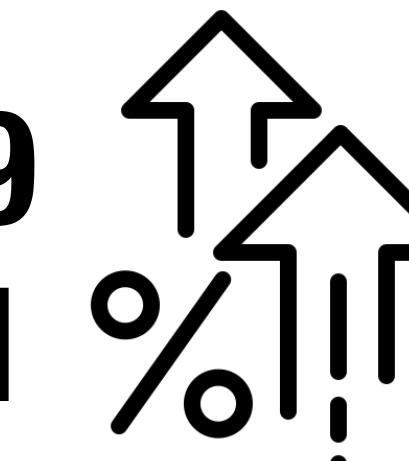


First Response Latency of **Maintainers**

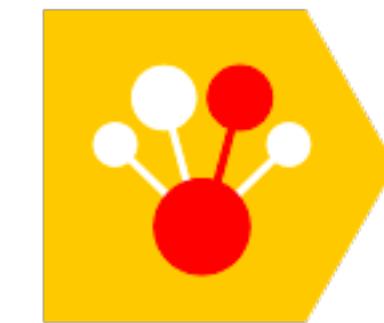


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

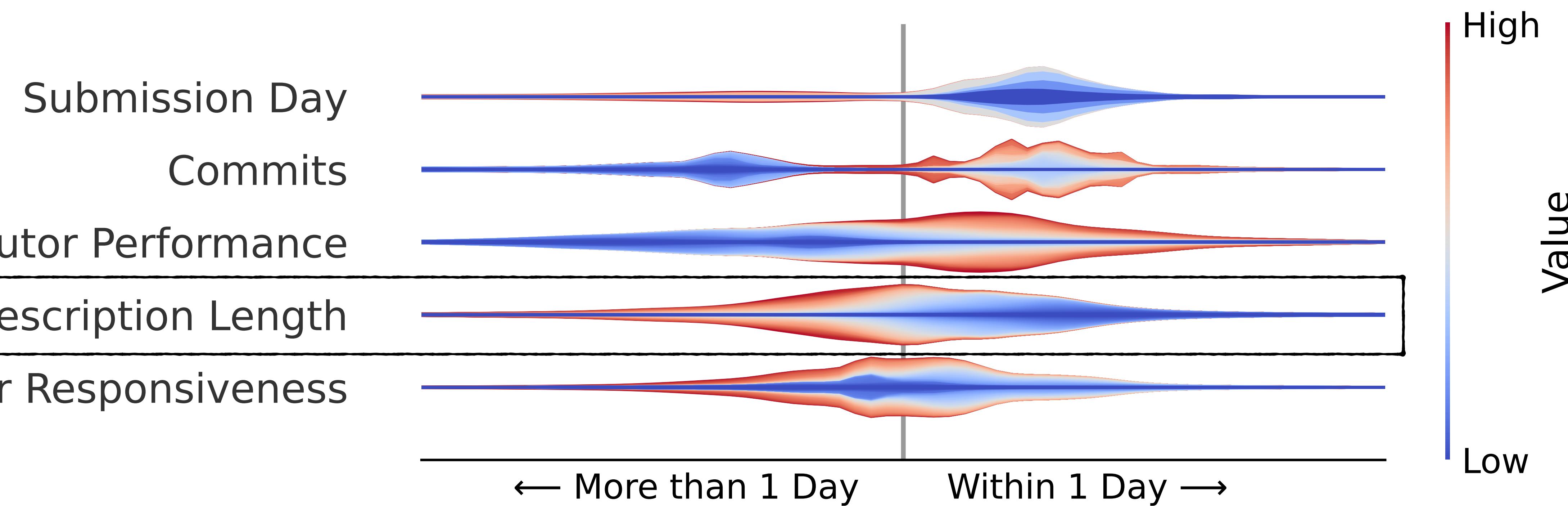


First Response Latency of **Maintainers**

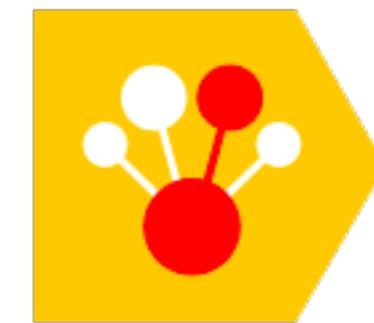


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

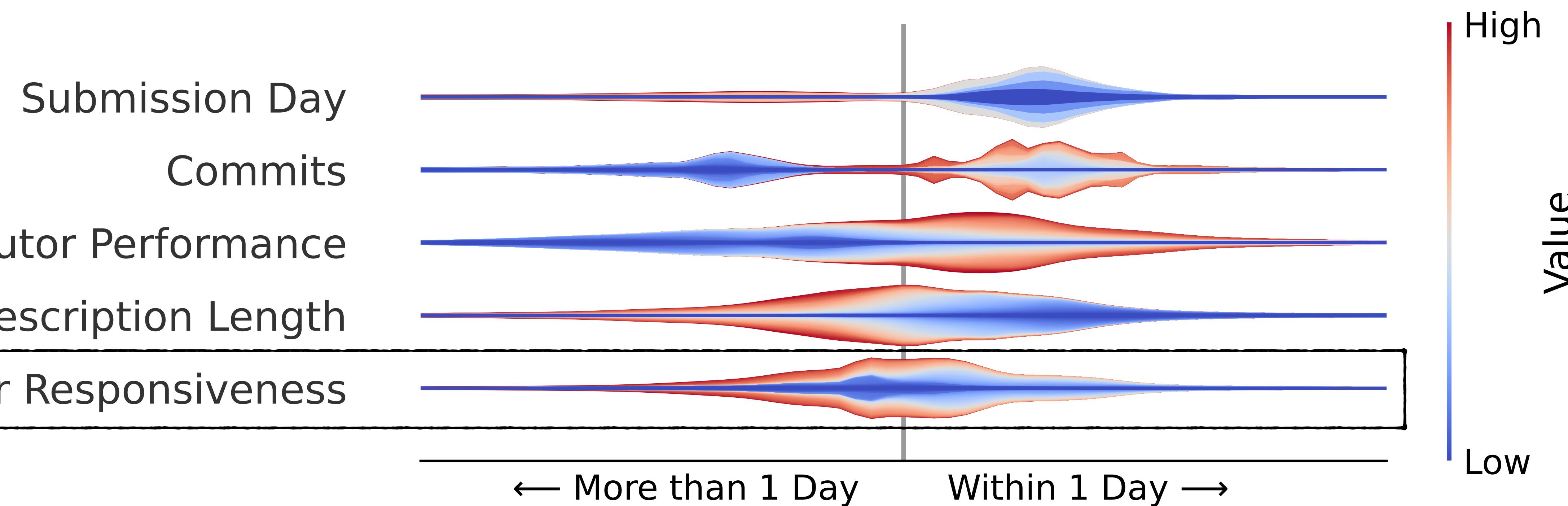


First Response Latency of **Maintainers**



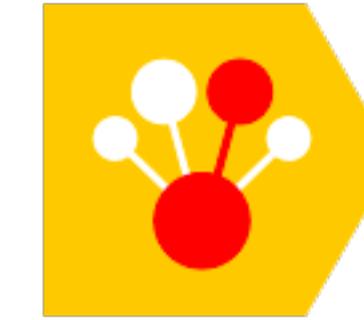
CatBoost

AUC-ROC: +29
AUC-PR: +51 %



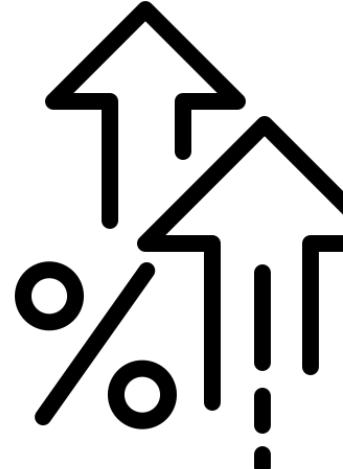
First Response Latency of **Contributors**

First Response Latency of **Contributors**

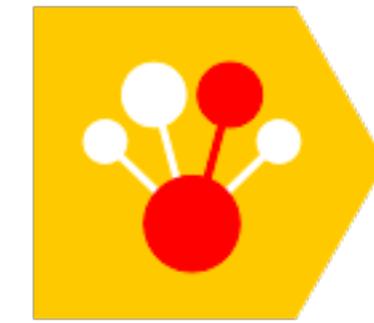


CatBoost

AUC-ROC: +29
AUC-PR: +51 %



First Response Latency of Contributors



CatBoost

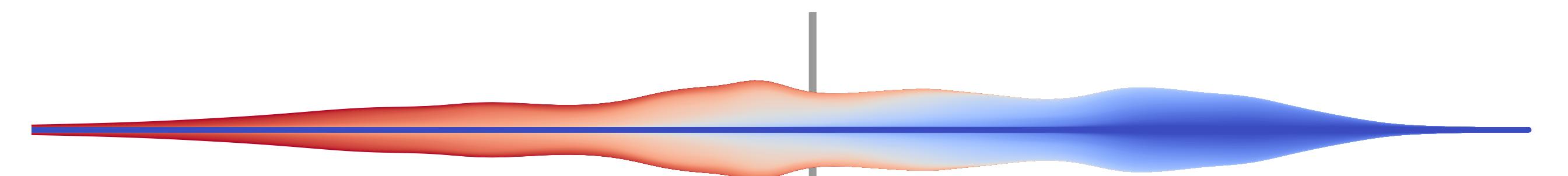
AUC-ROC: +29
AUC-PR: +51 %

High

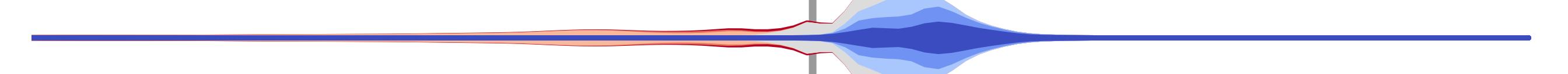
Value

Low

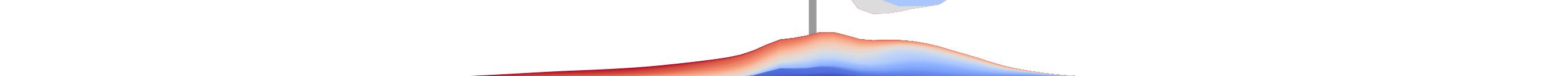
Review Latency



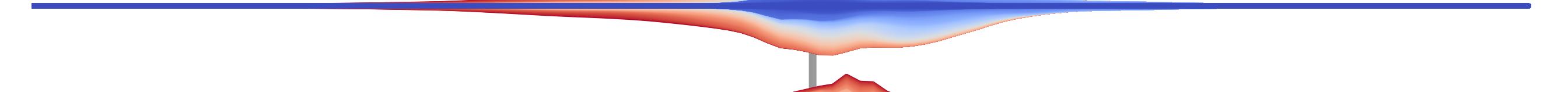
Review Day



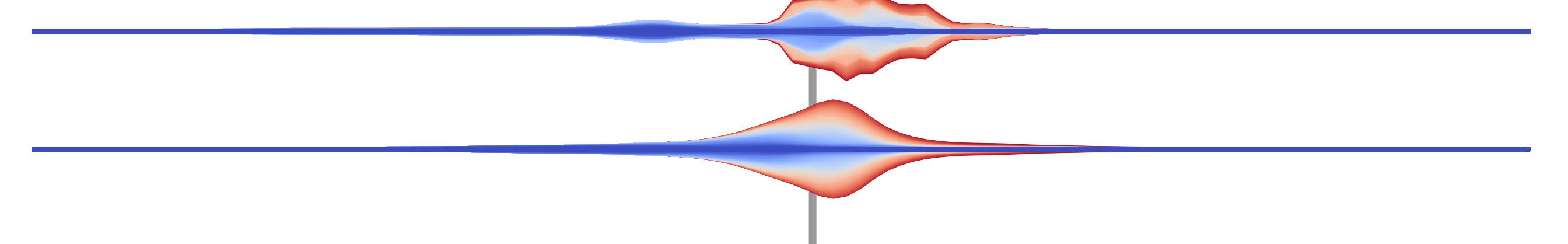
Contributor Responsiveness



Commits



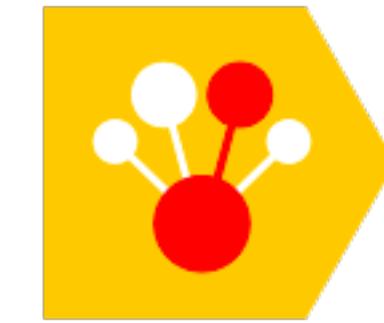
Contributor Performance



← More than 1 Day

Within 1 Day →

First Response Latency of Contributors

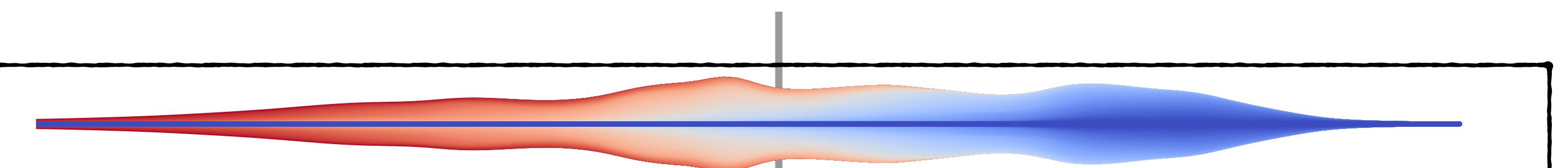


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

High

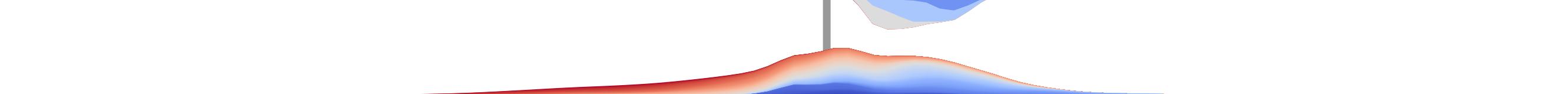
Review Latency



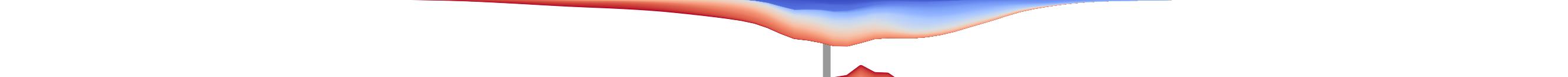
Review Day



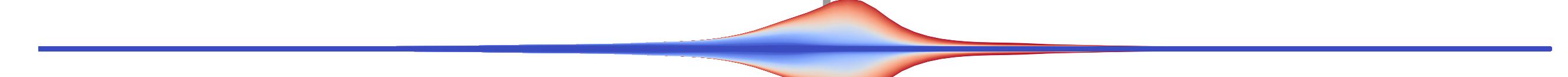
Contributor Responsiveness



Commits



Contributor Performance



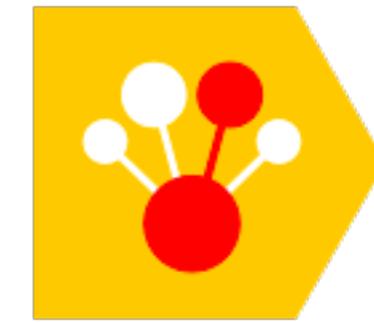
Value

Low

← More than 1 Day

Within 1 Day →

First Response Latency of Contributors

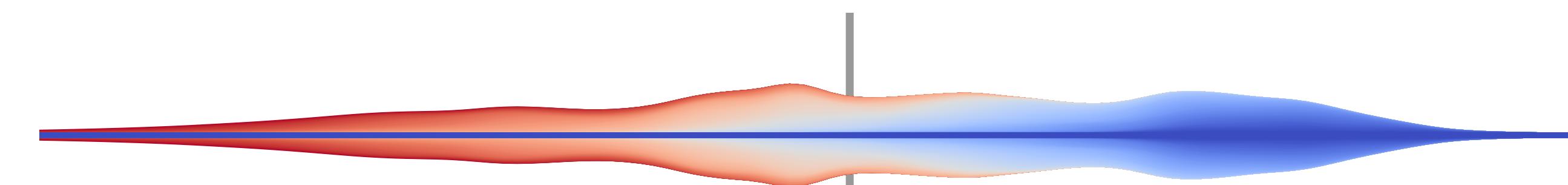


CatBoost

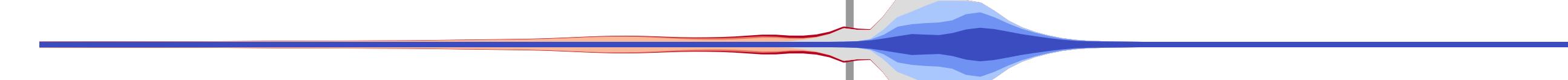
AUC-ROC: +29
AUC-PR: +51 %

High

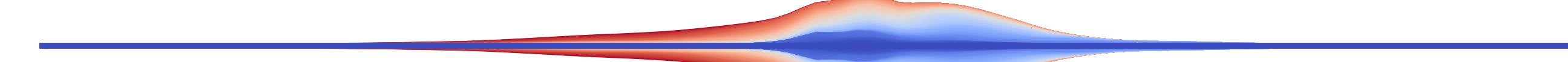
Review Latency



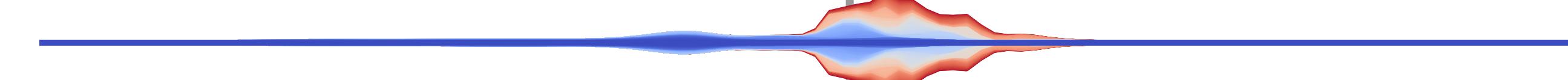
Review Day



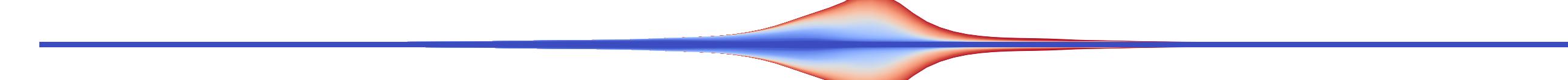
Contributor Responsiveness



Commits



Contributor Performance



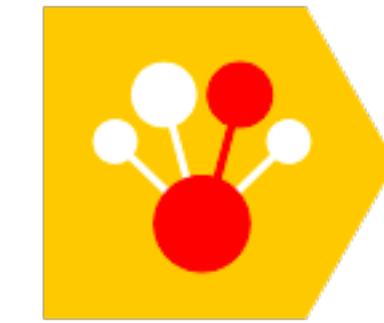
Value

Low

← More than 1 Day

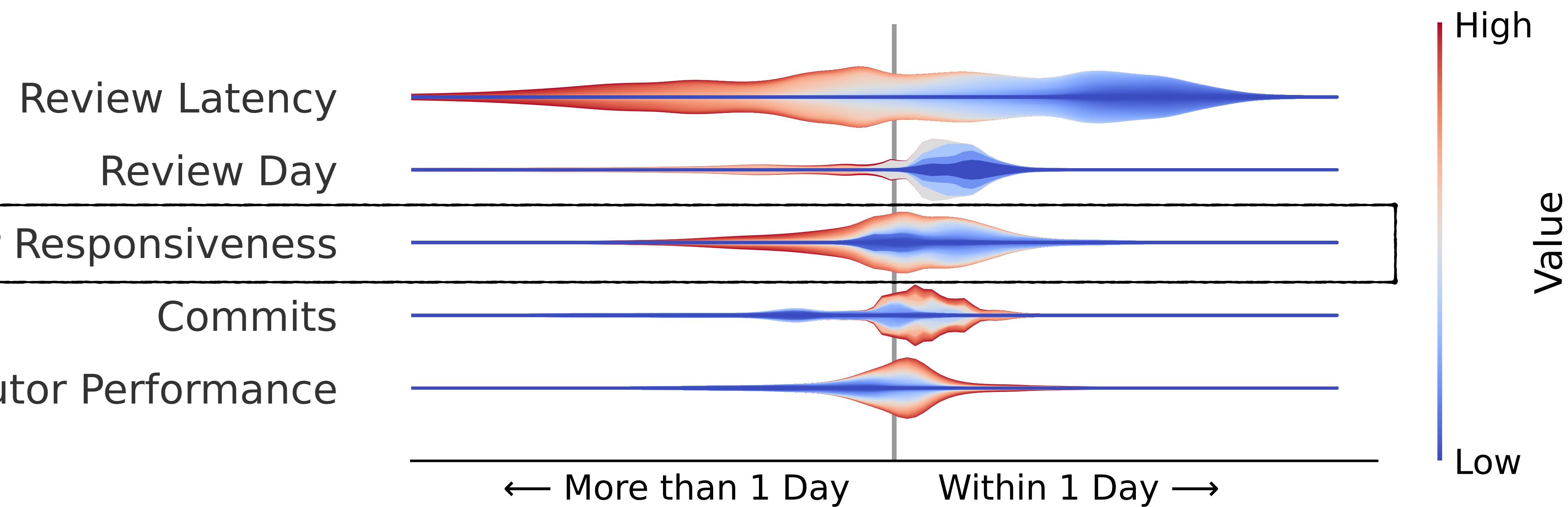
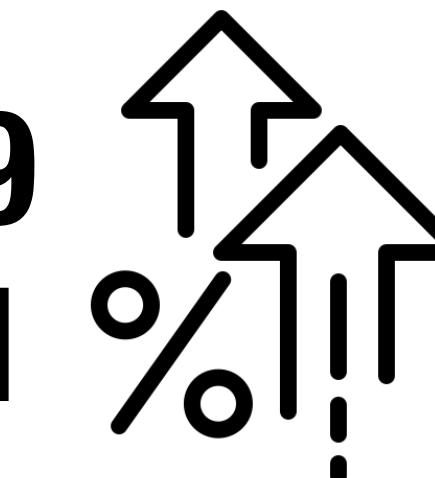
Within 1 Day →

First Response Latency of Contributors

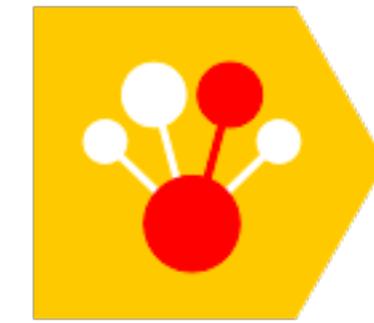


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

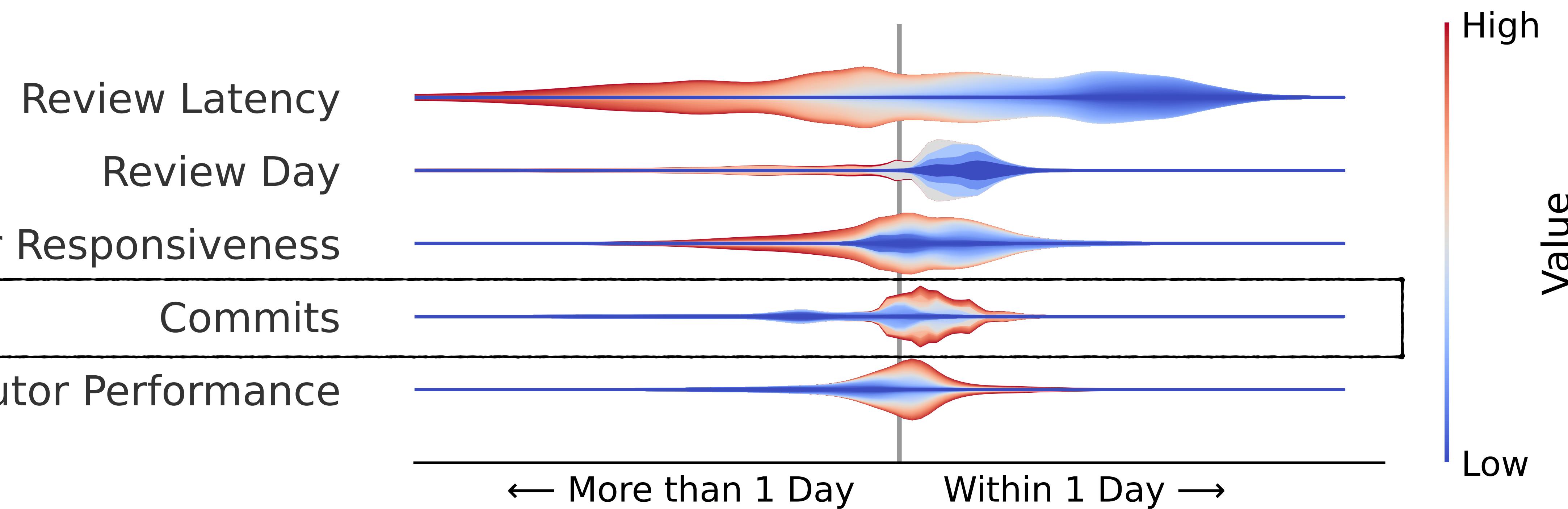


First Response Latency of Contributors

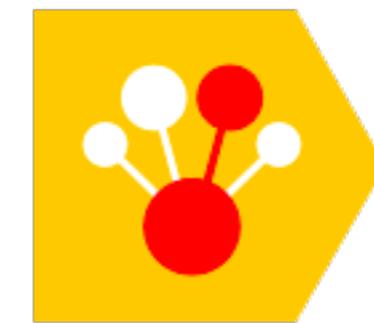


CatBoost

AUC-ROC: +29
AUC-PR: +51 %

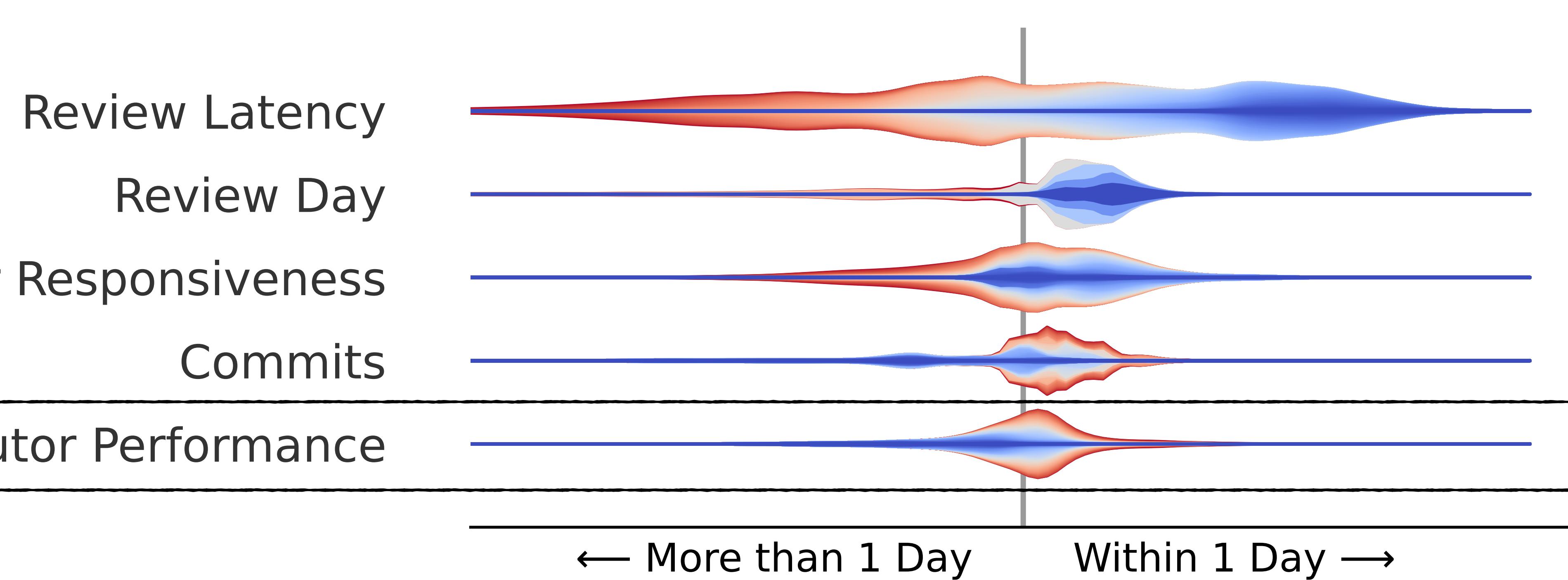


First Response Latency of Contributors



CatBoost

AUC-ROC: +29
AUC-PR: +51 %



Chapter5: Predicting the First Response Latency of Maintainers and Contributors in PRs

Page 1 of 16

*****For Peer Review Only*****

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING

Predicting the First Response Latency of Maintainers and Contributors in Pull Requests

SayedHassan Khatoonabadi, Ahmad Abdellatif, Diego Elias Costa, and Emad Shihab, *Senior Member, IEEE*

Abstract—The success of a Pull Request (PR) depends on the responsiveness of the maintainers and the contributor during the review process. Being aware of the expected waiting times can lead to better interactions and managed expectations for both the maintainers and the contributor. In this paper, we propose a machine-learning approach to predict the first response latency of the maintainers following the submission of a PR, and the first response latency of the contributor after receiving the first response from the maintainers. We curate a dataset of 20 large and popular open-source projects on GitHub and extract 21 features to characterize projects, contributors, PRs, and review processes. Using these features, we then evaluate seven types of classifiers to identify the best-performing models. We also perform permutation feature importance and SHAP analyses to understand the importance and impact of different features on the predicted response latencies. Our best-performing models achieve an average improvement of 33% in AUC-ROC and 58% in AUC-PR for maintainers, as well as 42% in AUC-ROC and 95% in AUC-PR for contributors compared to a no-skilled classifier across the projects. Our findings indicate that PRs submitted earlier in the week, containing an average or slightly above-average number of commits, and with concise descriptions are more likely to receive faster first responses from the maintainers. Similarly, PRs with a lower first response latency from maintainers, that received the first response of maintainers earlier in the week, and containing an average or slightly above-average number of commits tend to receive faster first responses from the contributors. Additionally, contributors with a higher acceptance rate and a history of timely responses in the project are likely to both obtain and provide faster first responses. Finally, we evaluate our approach in a cross-project setting, where the models achieve an average improvement of 33% in AUC-ROC and 58% in AUC-PR for maintainers, as well as an average improvement of 42% in AUC-ROC and 95% in AUC-PR for contributors. We believe that by predicting the first response latencies, our approach helps open-source projects facilitate collaboration between maintainers and contributors during their review processes.

Index Terms—Pull request abandonment, pull-based development, modern code review, social coding, open source software.

1 INTRODUCTION

PULL-based development has become a common paradigm for contributing to and reviewing code changes in numerous open-source projects [1, 2]. Pull Requests (PRs) are the driving force behind the maintenance and evolution of these projects, encompassing everything from bug fixes to new features. Contributors initiate this collaborative process by submitting a PR that proposes changes for integration into the project. The PR then undergoes a review process, during which the contributor revises the changes based on feedback from the project maintainers. This cycle repeats until the PR satisfies the maintainers' requirements for getting merged [3, 4].

The success of the PR depends on the responsiveness of both the maintainers and the contributor during the review process [5, 6, 7, 8]. Timely responses from the maintainers set a positive tone for the entire review process, increasing the likelihood of the contributor continuing the review process towards completion [5, 4]. Conversely, delayed responses are often perceived as negligence, increasing the risk of the contributor abandoning the PR [5, 6, 9]. Once the

maintainers have responded, the contributor's promptness in addressing the feedback is equally crucial. Timely responses help maintain the momentum of the review process, whereas delayed responses can cause it to stale [5, 4, 10].

Knowing the expected waiting times can lead to better interactions and managed expectations for both sides. Contributors, when aware of anticipated waiting times, can adjust their schedules accordingly, reducing uncertainty and preserving their motivation throughout the review process [5, 7]. Maintainers, aware of possible delays in contributor responses, can proactively offer additional support or take action to mitigate potential blockers [5]. This awareness also allows maintainers to better allocate their time and resources and prioritize PR reviews [8]. Furthermore, analyzing response time trends can help projects pinpoint and rectify bottlenecks, thereby enhancing the efficiency and effectiveness of their PR review workflows.

The first responses are of particular importance as they not only directly influence the duration [11, 12] and the outcome [13, 5, 6] of the review process, but also the likelihood of future contributions by the contributor [14, 11]. Despite the critical role of first responses, existing approaches only aim to predict the completion time of PRs [15, 16] or nudge overdue PRs [17]. Our study bridges this gap by proposing a machine learning approach to predict: (1) the first response latency of the maintainers following the submission of a PR, and (2) the first response latency of the contributor after receiving the first response from the maintainers.

- S. Khatoonabadi, A. Abdellatif, and E. Shihab are with the Data-driven Analysis of Software (DAS) Lab at the Department of Computer Science & Software Engineering, Concordia University, Montreal, QC, Canada. E-mail: {sayedhassan.khatoonabadi, ahmad.abdellatif, emad.shihab}@concordia.ca.
- D.E. Costa is with the Department of Computer Science & Software Engineering, Concordia University, Montreal, QC, Canada. E-mail: diego.costa@concordia.ca

Predicting response latencies enables maintainers and contributors to take proactive actions to mitigate potential challenges before the PR gets abandoned.

[Khatoonabadi et al., Under Review in TSE]

Future Work

Future Work

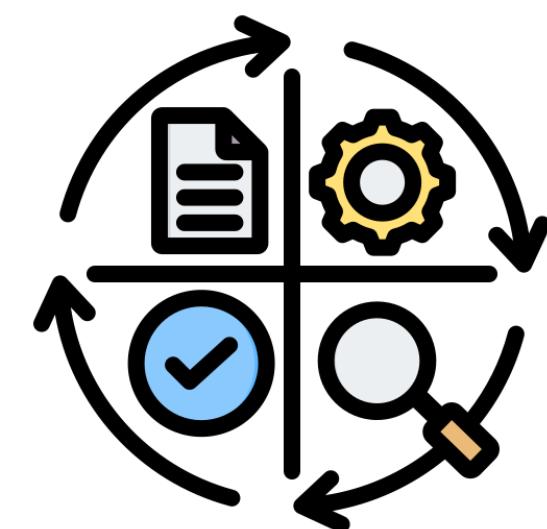


Assistance in Addressing
Change Requests

Future Work



Assistance in Addressing
Change Requests

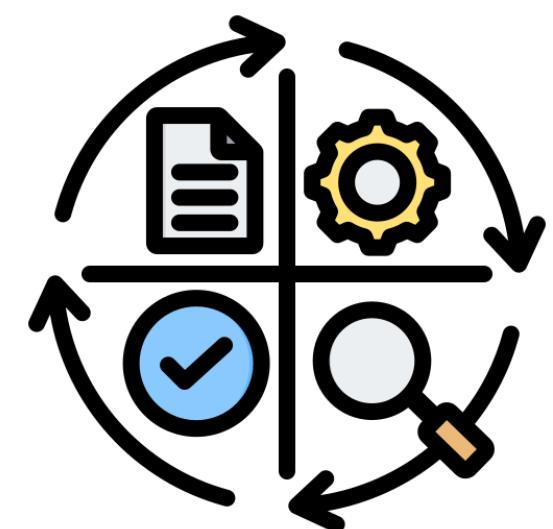


Assistance in Resolution of
CI Failures

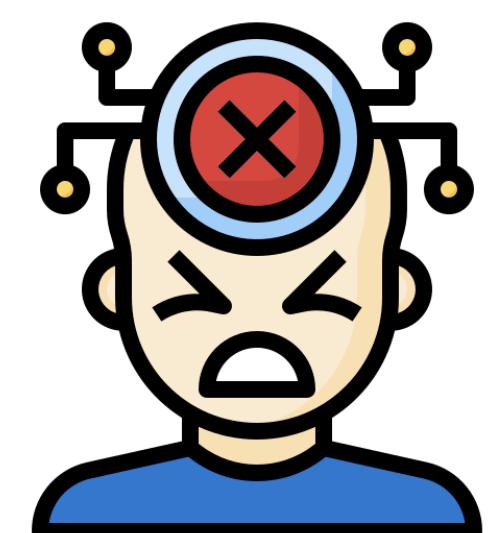
Future Work



Assistance in Addressing
Change Requests



Assistance in Resolution of
CI Failures

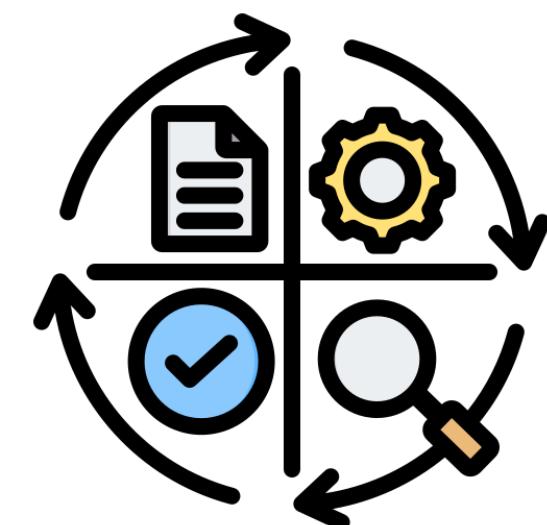


Automatic Resolution of
Merge Conflicts

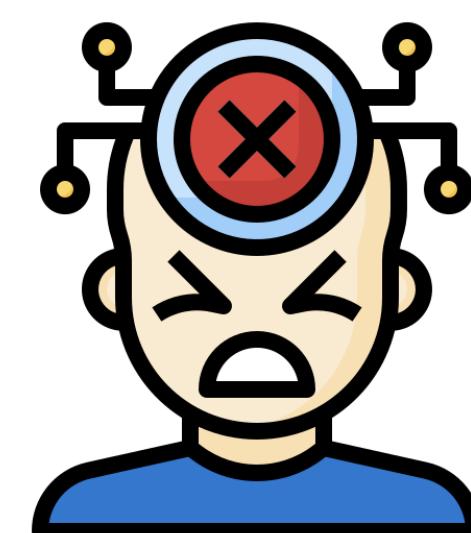
Future Work



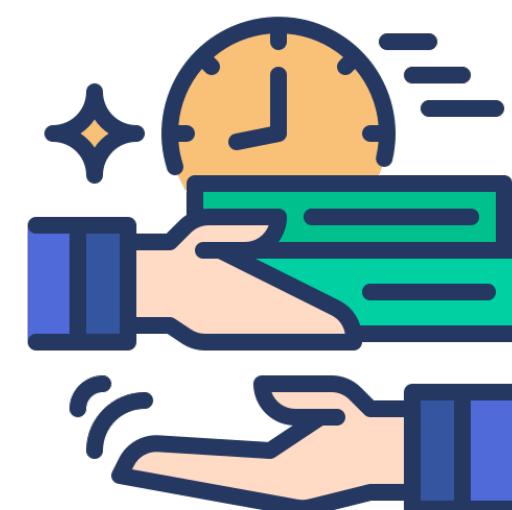
Assistance in Addressing
Change Requests



Assistance in Resolution of
CI Failures



Automatic Resolution of
Merge Conflicts



Smooth
Handover Mechanism

PR Abandonment Is a **Non-Trivial** Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li¹, Yue Yu², Tao Wang, Gang Yin, ShanShan Li³, and Hualimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not been well understood, especially for pull-based development. We present an empirical study on pull request abandonment based on a sizable dataset. We specifically examine 521 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show that the abandonment of pull requests is associated with the number of comments and the number of reviews. We discuss the avoidance strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

The great success of open source software (OSS) is based on the theory that many hands make light work [57]. For example, community-based OSS projects, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as a meritocracy structure [16], [17], and many general contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributions [15], [19], [29], [64], [66], [74]; and ii) investigating the factors that affect OSS developers' willingness to remain long-term and established contributors to

decrease the likelihood of developer disengagement and turnover [41], [43], [47], [54], [59], [78].

In this study, we shed light on OSS contributors' abandonment, focusing on the individual and single level in pull-based development [23], i.e., every single pull request that is unfinished but abandoned by the author. We clarify that the lack of responsiveness of project integrators to pull requests [23], [71] provides a synthesized collaboration environment for OSS distributed development by coupling the code repository, modern issue tracker, code review and automatic tools of DevOps. This has unprecedentedly lowered the barrier to entry for potential contributors and simplified their contribution process. However, it is common to abandon patch-based methods [69]. It is interesting to understand the reasons why there are considerable unfinished pull requests whose contributors have spent time programming, submitting and discussing at the beginning but walk away, leaving their contributions abandoned. As project integrators mentioned in our survey, "We see quite a few valuable contributions abandoned by authors just applying a few demanded changes might make their pull requests good enough to be eventually accepted" [SC224]. Furthermore, pull request abandonment is significant for the health of OSS projects and requires extra project maintenance efforts. Two representative quotes from our survey participants illustrate the negative impact of pull request abandonment [53], and "It's frustrating to set a bug almost fixed and then abandoned" [56]. A more complete picture of pull request abandonment can help OSS communities obtain great benefits from designing appropriate strategies to overcome avoidable abandonment, building more efficient tools for distributed collaboration, and helping individual participants understand the mindset of abandonment, and find the best ways to contribute.

¹ Zhitong Li, Yue Yu, Tao Wang, Gang Yin, and Hualimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {hizhixing5, yeyue, taowang2005, yinggang, hmlw}@nudt.edu.cn.
² Yue Yu is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshan@nudt.edu.cn.

Manuscript received 4 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of current version 14 June 2022. Corresponding author: Yue Yu.
Recommended for acceptance by M. Nagappan.
Digital Object Identifier no. 10.1109/TSE.2021.3053403

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications/rights/index.html for more information.

Waste Review Efforts

Clutter PR List

Require Extra Attention

Delay Landing of Interdependent PRs

Result in Duplicate PRs

Disorder Project Milestones

Leave Bad Impressions

PR Abandonment Is a Non-Trivial Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022

2173

Are You Still Working on This? An Empirical Study on Pull Request Abandonment

Zhixing Li¹, Yue Yu², Tao Wang, Gang Yin, ShanShan Li³, and Hualimin Wang

Abstract—The great success of numerous community-based open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment at the individual level have not received enough attention, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. We specifically examine 321 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show that the number of abandoned pull requests is correlated with the number of accepted ones. Finally, we discuss some effective strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

The great success of open source software (OSS) is based on the theory that “many hands make light work” [57]. Thus, for popular community-based OSS projects, a large number of volunteers continuously improve the project quality [52], [75] by submitting code patches, reporting bugs, and discussing new features. However, due to the nature of OSS collaboration (e.g., OSS projects are loosely organized as a peer-to-peer hierarchy structure [16], [17], and many casual contributors are motivated by indirect economic benefits [40], [48]), ensuring sustainability [21] is a persistent challenge in OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributions [15], [19], [29], [64], [66], [74]; and ii) investigating the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers' willingness to remain long-term and established contributors to traditional patch-based methods [68]. It is interesting to understand the reasons why there are considerable unfinished pull requests whose contributors have spent time programming, submitting and discussing at the beginning but walk away, leaving their contributions abandoned. As project integrators mentioned in our survey, “We see quite a few ‘idle’ contributions already made, just applying a few demanded changes might make their pull requests good enough to be eventually accepted” [SC224]. Furthermore, pull request abandonment is significant for the health of OSS projects and requires extra project maintenance efforts. Two representative quotes from our survey participants illustrate the negative impact of pull request abandonment: “It’s frustrating to set a bug almost fixed and then abandoned” [S66]. A more complete picture of pull request abandonment can help OSS communities obtain great benefits from designing appropriate strategies to overcome avoidable abandonment, building more efficient tools for distributed collaboration, and helping individual participants understand the mindset of abandonment, and find the best ways to contribute.

¹ Zhixing Li, Yue Yu, Tao Wang, Gang Yin, and Hualimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer Science and Technology, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {zhixing5, yueyu, taowang2005, yinggang, hmlwang}@nudt.edu.cn.
² Yue Yu is also with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshan@nudt.edu.cn.

Manuscript received 4 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of current version 14 June 2022. (Corresponding author: Yue Yu.) Recommended for acceptance by M. Nagappan. Digital Object Identifier no. 10.1109/TSE.2021.3053403

© 2021 IEEE. Personal use is permitted, but republication or distribution requires IEEE permission. See http://www.ieee.org/publications/rights/index.html for more information.

Waste Review Efforts

Clutter PR List

Require Extra Attention

Delay Landing of Interdependent PRs

Result in Duplicate PRs

Disorder Project Milestones

Leave Bad Impressions

Chapter3: Understanding the Dynamics of Contributor-Abandoned PRs

On Wasted Contributions: Understanding the Dynamics of Contributor-Abandoned Pull Requests—A Mixed-Methods Study of 10 Large Open-Source Projects

SAYEDHASSAN KHATOONABADI and DIEGO ELIAS COSTA, Concordia University, Canada
RABE ABDALKAREEM, Carleton University, Canada
EMAD SHIHAB, Concordia University, Canada

Pull-based development has enabled numerous volunteers to contribute to open-source projects with fewer barriers. Nevertheless, a considerable amount of pull requests (PRs) with valid contributions are abandoned by their contributors, wasting the effort and time put in by both the contributors and maintainers. To better understand the underlying dynamics of contributor-abandoned PRs, we conduct a mixed-methods study using both quantitative and qualitative methods. We analyze a dataset consisting of 265,200 PRs, including 4,450 abandoned ones from ten popular and mature GitHub projects and measure various characteristics of PRs, contributors, reviewers, and projects. Using statistical and machine learning techniques, we find that complex PRs, novice contributors, and lengthy reviews have a higher probability of abandonment and the rate of PR abandonment fluctuates alongside the projects' maturity or workload. To identify why contributors abandon their PRs, we also conduct a qualitative analysis of 354 abandoned PRs. Our observations show that the most frequent reason for abandonment is related to the obstacles imposed by contributors, followed by the burden imposed by maintainers during the review process. Finally, we survey the top core maintainers of the studied projects to understand their perspectives on dealing with PR abandonment and on our findings.

CCS Concepts: • Software and its engineering → Collaboration in software development; • Human-centered computing → Empirical studies in collaborative and social computing;

Additional Key Words and Phrases: Socio-technical factors, pull-based development, modern code review, social coding platforms, open-source software, mixed-methods research

ACM Reference format:
Sayedhassan Khatoonabadi, Diego Elias Costa, Rabe Abdalkareem, and Emad Shihab. 2023. On Wasted Contributions: Understanding the Dynamics of Contributor-Abandoned Pull Requests—A Mixed-Methods Study of 10 Large Open-Source Projects. *ACM Trans. Softw. Eng. Methodol.* 32, 1, Article 15 (February 2023), 39 pages. <https://doi.org/10.1145/3530785>

Authors' addresses: S. Khatoonabadi, D. E. Costa, and E. Shihab, Data-driven Analysis of Software (DAS) Lab, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC H3B 2L8, Canada; emails: {sayedhassan.khatoonabadi, diego.costa, emad.shihab}@concordia.ca; R. Abdalkareem, School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada; email: rabe.abdalkareem@carleton.ca. Permissions to reuse: In the event that you wish to reuse part or all of this article, please contact us for permission without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2023 Association for Computing Machinery. 1049-331X/23/02-ART15 \$15.00. <https://doi.org/10.1145/3530785>

ACM Transactions on Software Engineering and Methodology, Vol. 32, No. 1, Article 15. Publication date: February 2023.

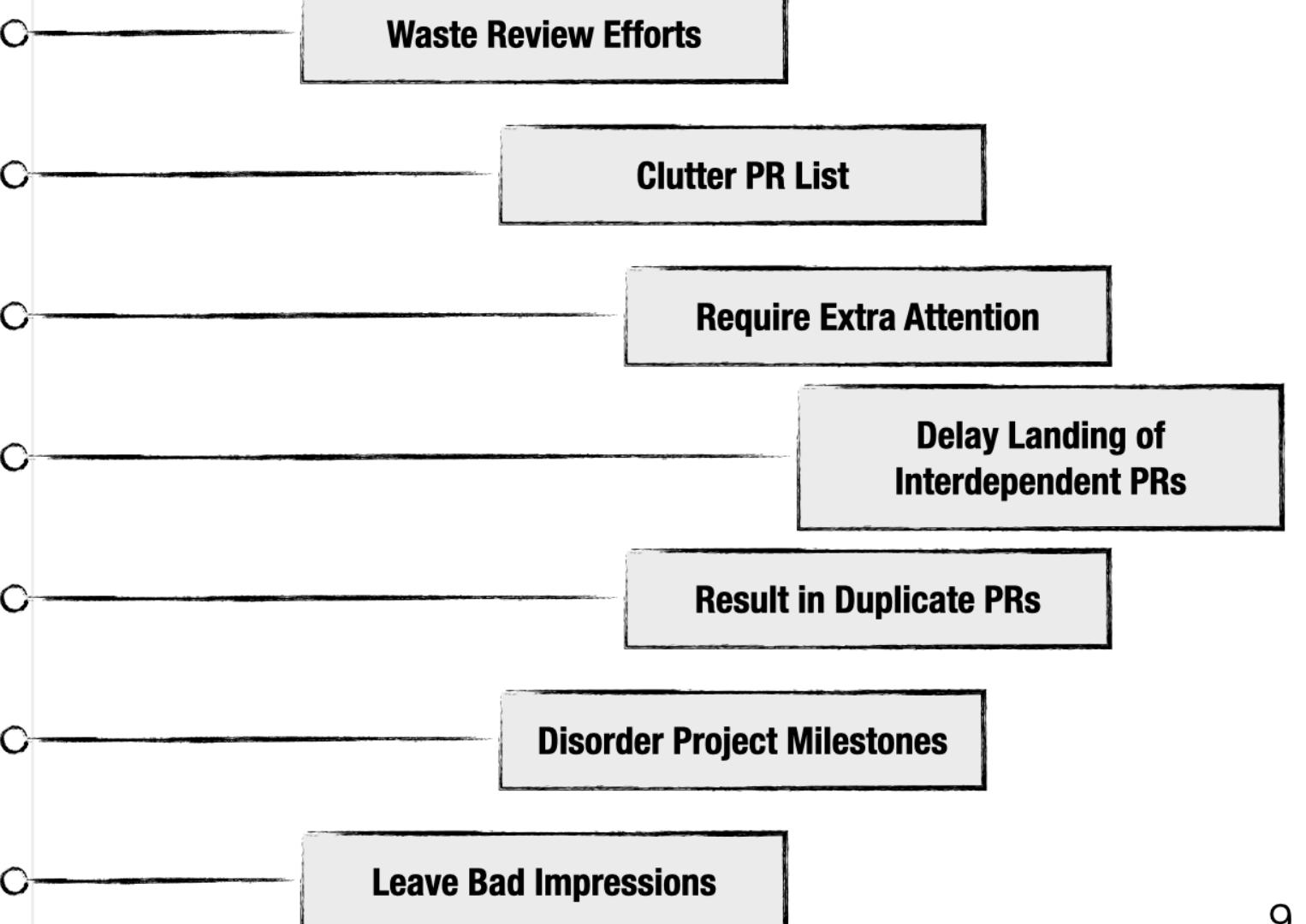
15

Contributor experience and review process length play the biggest roles in PR abandonment.

[Khatoonabadi et al., TOSEM 2023]

PR Abandonment Is a Non-Trivial Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022



Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

The great success of open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment have not been well explored, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. Specifically, we examine 521 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show that the adoption of GitHub's Stale bot can help mitigate abandonment. Finally, we propose some effective strategies used by project integrators to cope with abandoned pull request abandonment and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which could help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributions [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78]; ii) investigating the barriers and challenges faced by developers when contributing to OSS projects to assist the onboarding and participation of developers [15], [19], [29], [64], [66], [74]; and iii) investigating the factors that affect OSS developers' williness to remain long-term and established contributors to traditional patch-based methods [68]. It is interesting to understand the reasons why there are considerable unfinished pull requests whose contributors have spent time programming, submitting and discussing at the beginning but walk away, leaving their contributions abandoned. As project integrators stated in our survey, "We see quite a few submitted contributions abandoned by their authors just after a few demanded changes might make their pull requests good enough to be eventually accepted" [ISC224]. Furthermore, pull request abandonment is significant for the health of OSS projects and requires extra project maintenance efforts. Two representative quotes from our survey participants are as follows: "It's frustrating to see a bug almost fixed and then abandoned" [S6], and "It's frustrating to set a bug almost fixed and then abandoned" [S6]. A more complete picture of pull request abandonment can help OSS communities obtain great benefits from designing appropriate strategies to overcome avoidable abandonment, building more efficient tools for distributed collaboration, and helping individual participants understand the mindset of abandonment, and find the best ways to contribute.

• Zixing Li, Yue Yu, Tao Wang, Gang Yin, and Hualimin Wang are with the Key Laboratory of Parallel and Distributed Computing, College of Computer Science and Technology, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: {zixingli5, yueyu, taowing2005, yinggang, hmlwang}@nudt.edu.cn.
• Shanshan Li is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China. E-mail: shanshanli@nudt.edu.cn.
Manuscript received 4 July 2020; revised 18 Dec. 2020; accepted 17 Jan. 2021. Date of current version 14 June 2022.
(Corresponding author: Yue Yu.)
Recommended for acceptance by M. Nagappan.
Digital Object Identifier no. 10.1109/TSE.2021.3053403
© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications/rights/index.html for more information.

9

Chapter3: Understanding the Dynamics of Contributor-Abandoned PRs

On Wasted Contributions: Understanding the Dynamics of Contributor-Abandoned Pull Requests—A Mixed-Methods Study of 10 Large Open-Source Projects

SAYEDHASSAN KHATOONABADI and DIEGO ELIAS COSTA, Concordia University, Canada
RABE ABDALKAREEM, Carleton University, Canada
EMAD SHIHAB, Concordia University, Canada

Pull-based development has enabled numerous volunteers to contribute to open-source projects with fewer barriers. Nevertheless, a considerable amount of pull requests (PRs) with valid contributions are abandoned by their contributors, wasting the effort and time put in by both the contributors and maintainers. To better understand the underlying dynamics of contributor-abandoned PRs, we conduct a mixed-methods study using both quantitative and qualitative methods. We generate a dataset of 265,000 PRs, including 4,450 abandoned ones from ten popular and mature GitHub projects and measure various characteristics of PRs, contributors, reviewers, and projects. Using statistical and machine learning techniques, we find that complex PRs, novice contributors, and lengthy reviews have a higher probability of abandonment and the rate of PR abandonment fluctuates alongside the projects' maturity or workload. To identify why contributors abandon their PRs, we also conduct extensive interviews with 354 contributors. PRs are observed that the most frequent reason for abandonment is related to the obstacles imposed by contributors, followed by the burden imposed by maintainers during the review process. Finally, we survey the top core maintainers of the studied projects to understand their perspectives on dealing with PR abandonment and on our findings.

CCS Concepts—Software and its engineering → Collaboration in software development; Human-centered computing → Empirical studies in collaborative and social computing;

Additional Key Words and Phrases—Socio-technical factors, pull-based development, modern code review, social coding platforms, open-source software, mixed-methods research

ACM Reference format:
Sayedhassan Khatoonabadi, Diego Elias Costa, Rabe Abdalkareem, and Emad Shihab. 2023. On Wasted Contributions: Understanding the Dynamics of Contributor-Abandoned Pull Requests—A Mixed-Methods Study of 10 Large Open-Source Projects. *ACM Trans. Softw. Eng. Methodol.* 32, 1, Article 15 (February 2023), 39 pages.
<https://doi.org/10.1145/3530785>

Authors' addresses: S. Khatoonabadi, D. E. Costa, and E. Shihab, Data-driven Analysis of Software (DAS) Lab, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC H3H 2L8, Canada; e-mail: {sayedhassan.khatoonabadi, diego.costa, emad.shihab}@concordia.ca. R. Abdalkareem, School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada; e-mail: rabe.abdalkareem@carleton.ca. Permissions to reuse: (a) if your organization has granted you a license to reuse, you may follow the terms and conditions outlined in the license; (b) if your organization has not granted you a license to reuse, please contact us for permission. Copyright © 2023 Association for Computing Machinery. Abstracting with credit is permitted. Copying otherwise, or republishing, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Association for Computing Machinery.
1049-331X/2023/02-ART15 \$15.00
<https://doi.org/10.1145/3530785>

ACM Transactions on Software Engineering and Methodology, Vol. 32, No. 1, Article 15. Publication date: February 2023.

15

Contributor experience and review process length play the biggest roles in PR abandonment.

[Khatoonabadi et al., TOSEM 2023]

Chapter4: Understanding the Helpfulness of Stale Bot for PRs

Understanding the Helpfulness of Stale Bot for Pull-Based Development: An Empirical Study of 20 Large Open-Source Projects

SAYEDHASSAN KHATOONABADI and DIEGO ELIAS COSTA, Concordia University, Canada
SUHAIB MUJAHID, Mozilla Corporation, Canada
EMAD SHIHAB, Concordia University, Canada

Pull Requests (PRs) that are neither progressed nor resolved clutter the list of PRs, making it difficult for the maintainers to manage and prioritize unresolved PRs. To automatically track, follow up, and close such inactive PRs, Stale bot was introduced by GitHub. Despite its increasing adoption, there are ongoing debates on whether using Stale bot alleviates or exacerbates the problem of inactive PRs. To better understand if and how Stale bot helps projects in their pull-based development workflow, we perform an empirical study of 20 large open-source projects. Our results show that Stale bot can help maintainers to identify inactive PRs, as the projects closed more PRs within the first few months of adoption. Moreover, Stale bot can help improve the efficiency of the PR review process as the projects reviewed PRs that ended up merged and resolved PRs that ended up closed faster after the adoption. However, Stale bot can also negatively affect the contributors, as the projects experienced a considerable decrease in their number of active contributors after the adoption. Therefore, relying solely on Stale bot to deal with inactive PRs may lead to decreased community engagement and an increased probability of contributor abandonment.

CCS Concepts—Empirical computing → Empirical studies in collaborative and social computing; Open source software → Software and its engineering → Collaboration in software development; Modern code review; Social coding platforms; Open source software

Additional Key Words and Phrases—Software development bots, pull request abandonment, pull-based development, modern code review, social coding platforms, open source software
ACM Reference format:
Sayedhassan Khatoonabadi, Diego Elias Costa, Suhaib Mujahid, and Emad Shihab. 2023. Understanding the Helpfulness of Stale Bot for Pull-Based Development: An Empirical Study of 20 Large Open-Source Projects. *ACM Trans. Softw. Eng. Methodol.* 33, 2, Article 36 (December 2023), 43 pages.
<https://doi.org/10.1145/3624739>

Despite its helpfulness in dealing with a backlog of unresolved PRs, Stale bot tends to further aggravate PR abandonment and decrease contributor engagement.

[Khatoonabadi et al., TOSEM 2023]

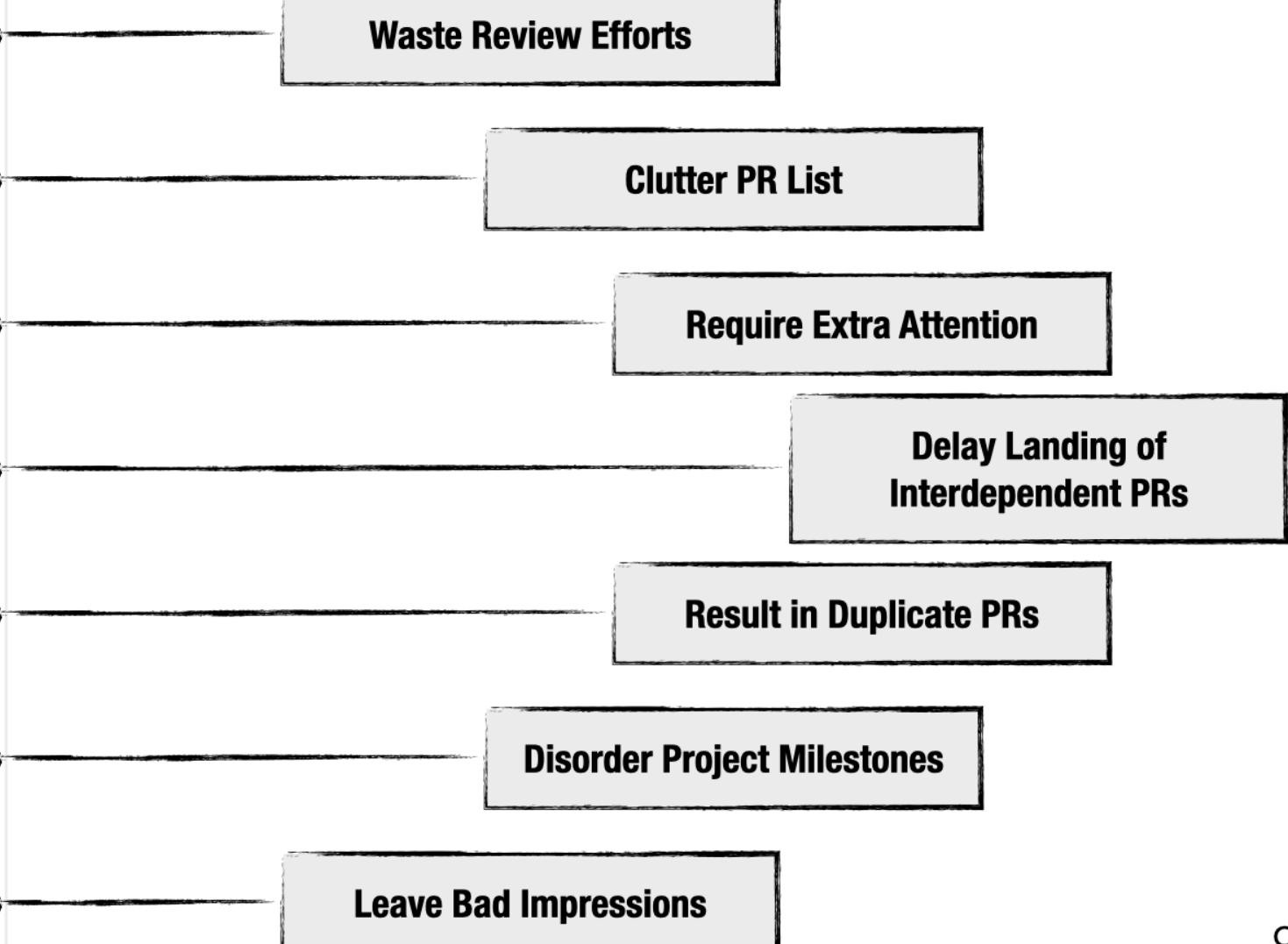
Author's address: S. Khatoonabadi, and E. Shihab, Data-driven Analysis of Software (DAS) Lab, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC H3H 2L8, Canada; e-mail: sayedhassan.khatoonabadi, diego.costa, emad.shihab@concordia.ca. D. E. Costa, Department of Computer Science & Software Engineering, Concordia University, 2155 Guy Street, Montreal, QC H3H 2L8, Canada; e-mail: diego.costa@concordia.ca. S. Mujahid, Mozilla Corporation, Montreal QC, Canada; e-mail: suhaib.mujahid@mozilla.com. Permissions to reuse: (a) if your organization has granted you a license to reuse, you may follow the terms and conditions outlined in the license; (b) if your organization has not granted you a license to reuse, please contact us for permission. Copyright © 2023 Association for Computing Machinery. Abstracting with credit is permitted. Copying otherwise, or republishing, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
1049-331X/2023/02-ART36 \$15.00
<https://doi.org/10.1145/3624739>

ACM Transactions on Software Engineering and Methodology, Vol. 33, No. 2, Article 36. Pub. date: December 2023.

36

PR Abandonment Is a Non-Trivial Challenge for OSS Projects

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 48, NO. 6, JUNE 2022



Index Terms—Pull request abandonment, pull-based development, open source software

1 INTRODUCTION

The great success of open source software (OSS) is based on volunteers continuously submitting contributions, but ensuring sustainability is a persistent challenge in OSS communities. Although the motivations behind and barriers to OSS contributors' joining and retention have been extensively studied, the impacts of, reasons for and solutions to contribution abandonment have not been well understood, especially for pull-based development. To bridge this gap, we present an empirical study on pull request abandonment based on a sizable dataset. Specifically, we examine 521 abandoned pull requests on GitHub and then quantify the manual observations by surveying 710 OSS developers. We find that while the lack of integrators' responsiveness and the lack of contributors' time and interest remain the main reasons that deter contributors from participation, limitations during the processes of patch updating and consensus reaching can also cause abandonment. We also show that the lack of communication and coordination among maintainers and contributors are the main reasons that contribute to the abandonment of PRs. Finally, we propose several practical strategies used by project integrators to cope with abandoned pull requests and highlight the need for a practical handover mechanism. We discuss the actionable suggestions and implications for OSS practitioners and tool builders, which can help to upgrade the infrastructure and optimize the mechanisms of OSS communities.

Prior work in this area has paid considerable attention to: i) revealing the motivations behind why developers make contributions to OSS projects to better recruit and encourage contributions [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], [121], [122], [123], [124], [125], [126], [127], [128], [129], [130], [131], [132], [133], [134], [135], [136], [137], [138], [139], [140], [141], [142], [143], [144], [145], [146], [147], [148], [149], [150], [151], [152], [153], [154], [155], [156], [157], [158], [159], [160], [161], [162], [163], [164], [165], [166], [167], [168], [169], [170], [171], [172], [173], [174], [175], [176], [177], [178], [179], [180], [181], [182], [183], [184], [185], [186], [187], [188], [189], [190], [191], [192], [193], [194], [195], [196], [197], [198], [199], [200], [201], [202], [203], [204], [205], [206], [207], [208], [209], [210], [211], [212], [213], [214], [215], [216], [217], [218], [219], [220], [221], [222], [223], [224], [225], [226], [227], [228], [229], [230], [231], [232], [233], [234], [235], [236], [237], [238], [239], [240], [241], [242], [243], [244], [245], [246], [247], [248], [249], [250], [251], [252], [253], [254], [255], [256], [257], [258], [259], [260], [261], [262], [263], [264], [265], [266], [267], [268], [269], [270], [271], [272], [273], [274], [275], [276], [277], [278], [279], [280], [281], [282], [283], [284], [285], [286], [287], [288], [289], [290], [291], [292], [293], [294], [295], [296], [297], [298], [299], [300], [301], [302], [303], [304], [305], [306], [307], [308], [309], [310], [311], [312], [313], [314], [315], [316], [317], [318], [319], [320], [321], [322], [323], [324], [325], [326], [327], [328], [329], [330], [331], [332], [333], [334], [335], [336], [337], [338], [339], [340], [341], [342], [343], [344], [345], [346], [347], [348], [349], [350], [351], [352], [353], [354], [355], [356], [357], [358], [359], [360], [361], [362], [363], [364], [365], [366], [367], [368], [369], [370], [371], [372], [373], [374], [375], [376], [377], [378], [379], [380], [381], [382], [383], [384], [385], [386], [387], [388], [389], [390], [391], [392], [393], [394], [395], [396], [397], [398], [399], [400], [401], [402], [403], [404], [405], [406], [407], [408], [409], [410], [411], [412], [413], [414], [415], [416], [417], [418], [419], [420], [421], [422], [423], [424], [425], [426], [427], [428], [429], [430], [431], [432], [433], [434], [435], [436], [437], [438], [439], [440], [441], [442], [443], [444], [445], [446], [447], [448], [449], [450], [451], [452], [453], [454], [455], [456], [457], [458], [459], [460], [461], [462], [463], [464], [465], [466], [467], [468], [469], [470], [471], [472], [473], [474], [475], [476], [477], [478], [479], [480], [481], [482], [483], [484], [485], [486], [487], [488], [489], [490], [491], [492], [493], [494], [495], [496], [497], [498], [499], [500], [501], [502], [503], [504], [505], [506], [507], [508], [509], [510], [511], [512], [513], [514], [515], [516], [517], [518], [519], [520], [521], [522], [523], [524], [525], [526], [527], [528], [529], [530], [531], [532], [533], [534], [535], [536], [537], [538], [539], [540], [541], [542], [543], [544], [545], [546], [547], [548], [549], [550], [551], [552], [553], [554], [555], [556], [557], [558], [559], [5510], [5511], [5512], [5513], [5514], [5515], [5516], [5517], [5518], [5519], [5520], [5521], [5522], [5523], [5524], [5525], [5526], [5527], [5528], [5529], [5530], [5531], [5532], [5533], [5534], [5535], [5536], [5537], [5538], [5539], [55310], [55311], [55312], [55313], [55314], [55315], [55316], [55317], [55318], [55319], [55320], [55321], [55322], [55323], [55324], [55325], [55326], [55327], [55328], [55329], [55330], [55331], [55332], [55333], [55334], [55335], [55336], [55337], [55338], [55339], [55340], [55341], [55342], [55343], [55344], [55345], [55346], [55347], [55348], [55349], [55350], [55351], [55352], [55353], [55354], [55355], [55356], [55357], [55358], [55359], [55360], [55361], [55362], [55363], [55364], [55365], [55366], [55367], [55368], [55369], [55370], [55371], [55372], [55373], [55374], [55375], [55376], [55377], [55378], [55379], [55380], [55381], [55382], [55383], [55384], [55385], [55386], [55387], [55388], [55389], [55390], [55391], [55392], [55393], [55394], [55395], [55396], [55397], [55398], [55399], [553100], [553101], [553102], [553103], [553104], [553105], [553106], [553107], [553108], [553109], [553110], [553111], [553112], [553113], [553114], [553115], [553116], [553117], [553118], [553119], [5531100], [5531101], [5531102], [5531103], [5531104], [5531105], [5531106], [5531107], [5531108], [5531109], [5531110], [5531111], [5531112], [5531113], [5531114], [5531115], [5531116], [5531117], [5531118], [5531119], [55311100], [55311101], [55311102], [55311103], [55311104], [55311105], [55311106], [55311107], [55311108], [55311109], [55311110], [55311111], [55311112], [55311113], [55311114], [55311115], [55311116], [55311117], [55311118], [55311119], [553111100], [553111101], [553111102], [553111103], [553111104], [553111105], [553111106], [553111107], [553111108], [553111109], [553111110], [553111111], [553111112], [553111113], [553111114], [553111115], [553111116], [553111117], [553111118], [553111119], [5531111100], [5531111101], [5531111102], [5531111103], [5531111104], [5531111105], [5531111106], [5531111107], [5531111108], [5531111109], [5531111110], [5531111111], [5531111112], [5531111113], [5531111114], [5531111115], [5531111116], [5531111117], [5531111118], [5531111119], [55311111100], [55311111101], [55311111102], [55311111103], [55311111104], [55311111105], [55311111106], [55311111107], [55311111108], [55311111109], [55311111110], [55311111111], [55311111112], [55311111113], [55311111114], [55311111115], [55311111116], [55311111117], [55311111118], [55311111119], [553111111100], [553111111101], [553111111102], [553111111103], [553111111104], [553111111105], [553111111106], [553111111107], [553111111108], [553111111109], [553111111110], [553111111111], [553111111112], [553111111113], [553111111114], [553111111115], [553111111116], [553111111117], [553111111118], [553111111119], [5531111111100], [5531111111101], [5531111111102], [5531111111103], [5531111111104], [5531111111105], [5531111111106], [5531111111107], [5531111111108], [5531111111109], [5531111111110], [5531111111111], [5531111111112], [5531111111113], [5531111111114], [5531111111115], [5531111111116], [5531111111117], [5531111111118], [5531111111119], [55311111111100], [55311111111101], [55311111111102], [55311111111103], [55311111111104], [55311111111105], [55311111111106], [55311111111107], [55311111111108], [55311111111109], [55311111111110], [55311111111111], [55311111111112], [55311111111113], [55311111111114], [55311111111115], [55311111111116], [55311111111117], [55311111111118], [55311111111119], [553111111111100], [553111111111101], [553111111111102], [553111111111103], [553111111111104], [553111111111105], [553111111111106], [553111111111107], [553111111111108], [553111111111109], [553111111111110], [553111111111111], [553111111111112], [553111111111113], [553111111111114], [553111111111115], [553111111111116], [553111111111117], [553111111111118], [553111111111119], [5531111111111100], [5531111111111101], [5531111111111102], [5531111111111103], [5531111111111104], [5531111111111105], [5531111111111106], [5531111111111107], [5531111111111108], [5531111111111109], [5531111111111110], [5531111111111111], [5531111111111112], [5531111111111113], [5531111111111114], [5531111111111115], [5531111111111116], [5531111111111117], [5531111111111118], [5531111111111119], [55311111111111100], [55311111111111101], [55311111111111102], [55311111111111103], [55311111111111104], [55311111111111105], [55311111111111106], [55311111111111107], [55311111111111108], [55311111111111109], [55311111111111110], [55311111111111111], [55311111111111112], [55311111111111113], [55311111111111114], [55311111111111115], [55311111111111116], [55311111111111117], [55311111111111118], [55311111111111119], [553111111111111100], [553111111111111101], [553111111111111102], [553111111111111103], [553111111111111104], [553111111111111105], [553111111111111106], [553111111111111107], [553111111111111108], [553111111111111109], [553111111111111110], [553111111111111111], [553111111111111112], [553111111111111113], [553111111111111114], [553111111111111115], [553111111111111116], [5531111

Thank You