What We Can Learn From Your Mistakes: A Study of Rust Errors

Molly MacLaren, Ruochen Wang, Michael Coblenz

New to Rust ?

0 How hard is it to contribute to a rust-embedd

seeking help & advice

Hi all,

I'm a newbie Rust developer. Previously I wanted to build micropython (I gave up since the device firmware does n mesh). But after learning some rust, I accidentally search

Posted by u/CrafterJunkie1 6 hours ago

0

Calling a function using the elements of a

Hey! I'm new to rust and found myself tinkering with

Posted by u/a-dev-in-space 1 day ago

71

How often do you write C and C++

seeking help & advice

Hello,

I am a new rust developer coming from a 5 year background with languages. I really enjoy rust and would like to have my next pos

Posted by u/PythonPizzaDE 1 day ago

22

A little dmenu alternative I wrote

Hello there I do learn rust at the moment and f software launcher like dmenu. I would be more some feedback (roast me) on the code

Posted by u/appinv 22 hours ago

40

OpenSource: Yet Another Way To Learn Rust

I read the Rust book twice. Also Comprehensive Rust from as i forgot ownership fast when digesting other topics. Th concepts to cover. I lost a lot of time starting again. I reall and started programming on some playgrounds as well a

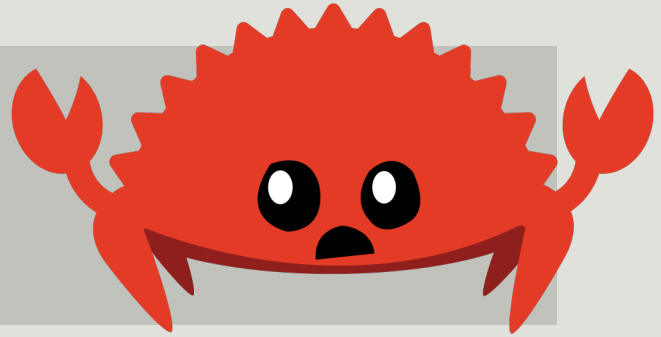
Posted by u/platonicjs 1 day ago

3

the trait `Responder` is implemented

I am a new comer on Rust and trying to get han custom struct for my response, the cargo check

Ownership Rules

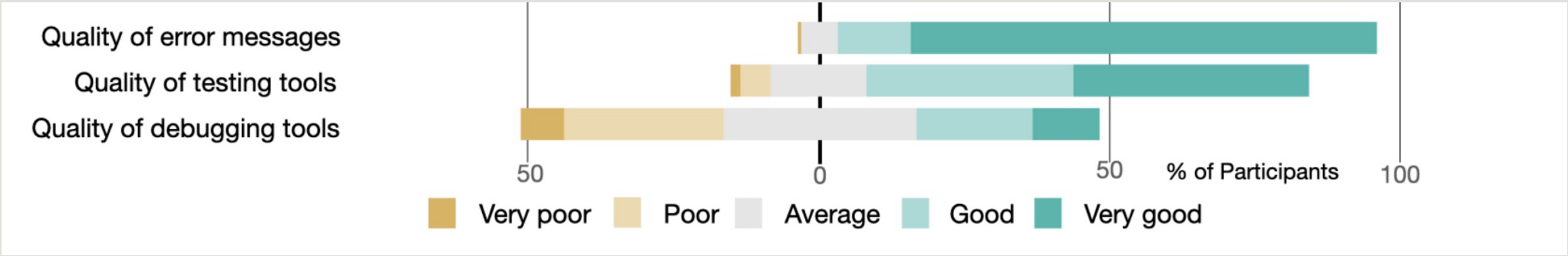


Ownership: Every variable must have exactly ONE owner at a time.

- References are valid for a scope or *lifetime* determined by the compiler
- Ownership can be temporarily leased through *borrowing*

Motivation

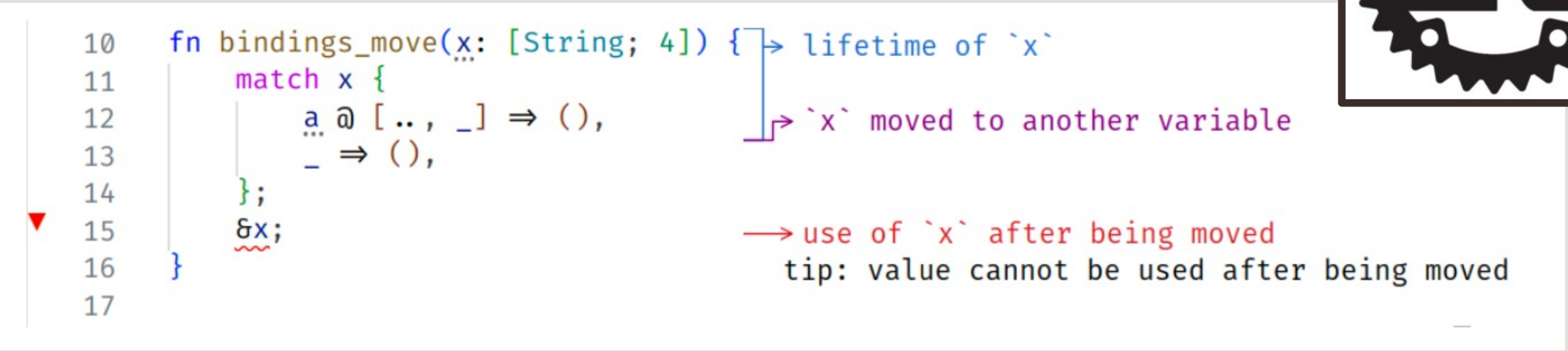
- Understanding concepts of ownership may be difficult for those coming from other languages that manage memory differently.
- Rust has great error messages but mediocre tools?



Fulton, K. R., Chan, A., Votipka, D., Hicks, M., & Mazurek, M. L. (2021). Benefits and drawbacks of adopting a secure programming language: Rust as a case study. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)* (pp. 597-616).

Prior Work

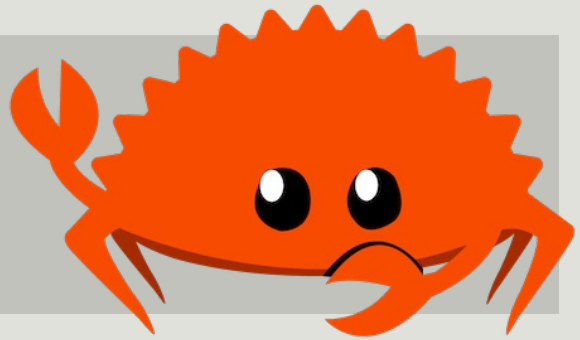
Rust Error Visualizer (REVIS)



Designed by Ruochen Wang, REVIS is a VSCode extension that generates diagrams representing lifetime and ownership errors using diagnostic data.

A study was conducted over Spring 2023 in CSE 131/231, a course on compiler construction.

Research Questions



In order to design and improve Rust debugging tools, we need to know which errors are highest priority:

- **What are the most frequent errors Rust users make and which errors take the longest to resolve?**

We also need to evaluate tool effectiveness:

- **Do error resolution tools such as REVIS reduce time spent fixing errors?**
- **Can we track individuals' learning progress through resolution times?**

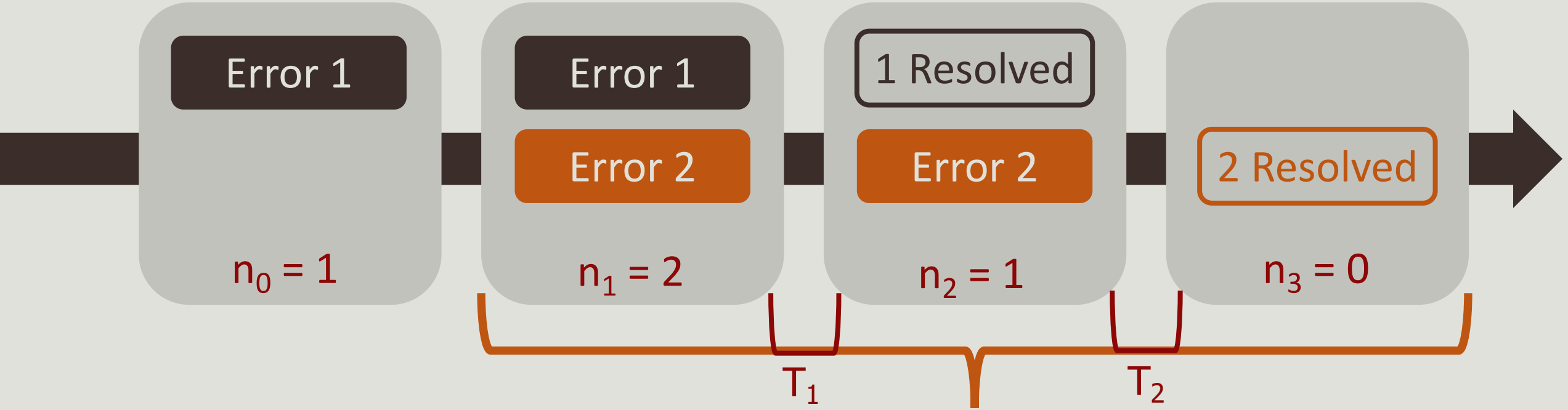
Analysis by Resolution Session

After rebuilding every code snapshot and logging errors, the time-to-repair errors was determined by:

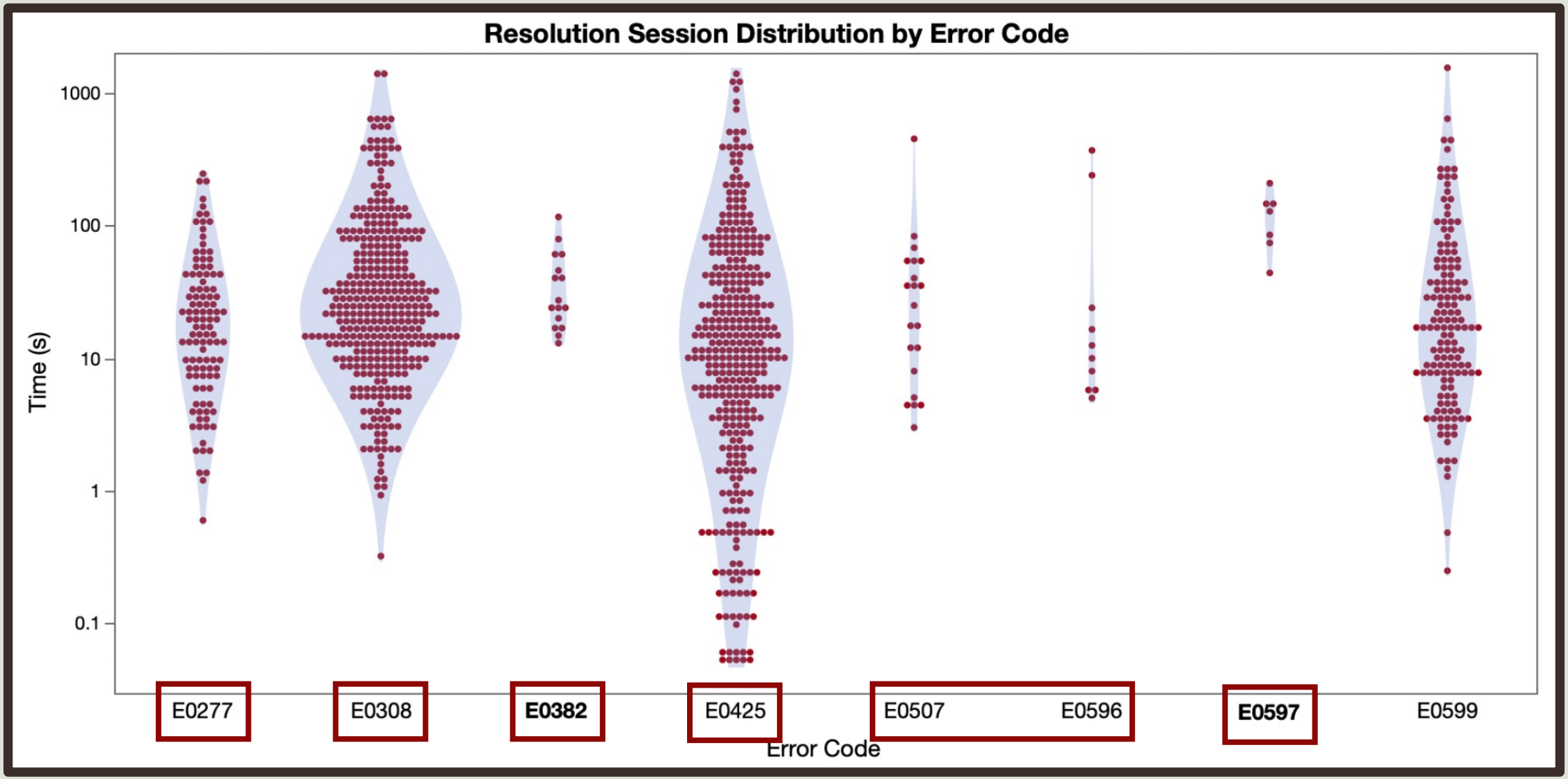
- **Resolution Session**: A series of consecutive builds containing a particular error.
- **Active Resolution Cost (ARC)**: Sum of the time between each build divided by the number of errors present in each build.

Mesbah, A., Rice, A., Johnston, E., Glorioso, N., & Aftandilian, E. (2019, August). Deepdelta: learning to repair compilation errors. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 925-936).

Resolution Session Example



Error 2 Active Resolution Cost = $\frac{T_1}{n_1} + \frac{T_2}{n_2}$



Type does not implement trait

Type mismatch

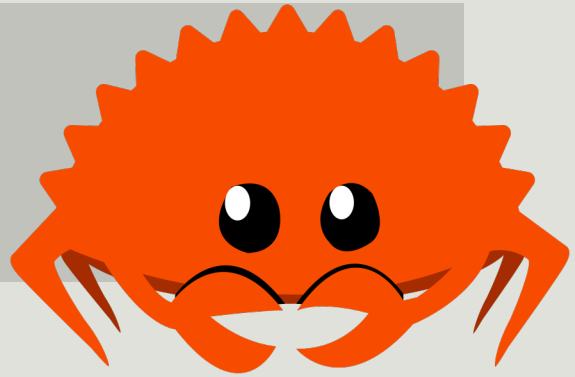
Variable used after value moved

Unresolved name

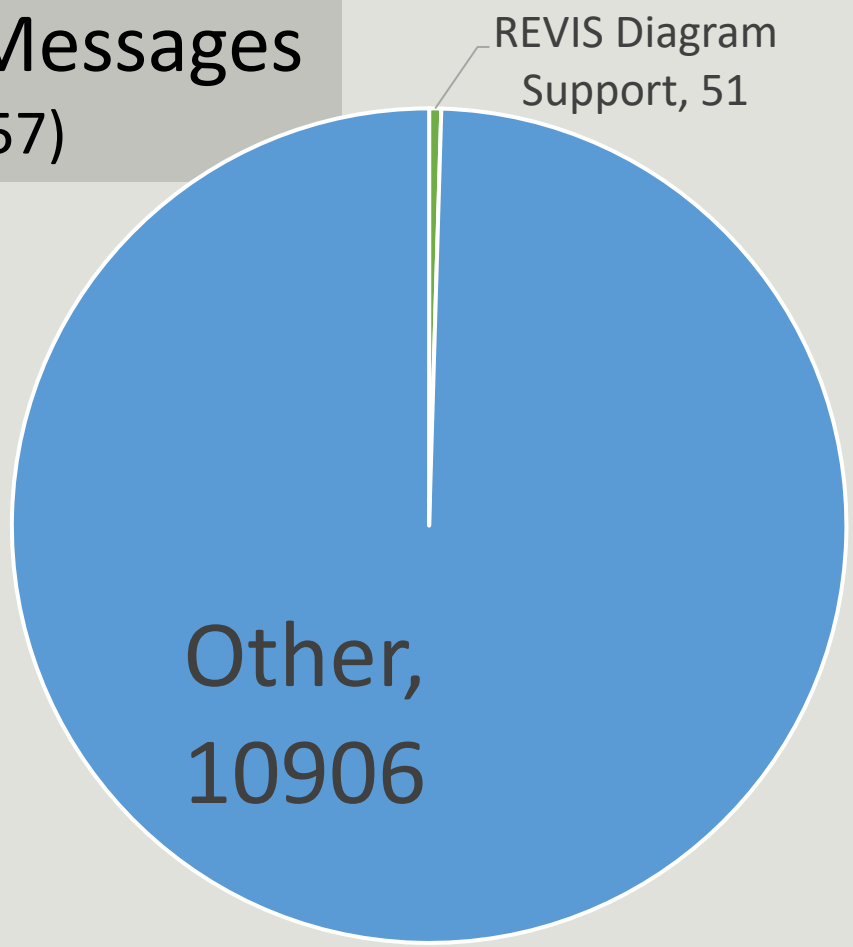
Borrowing Errors

Value dropped while still borrowed

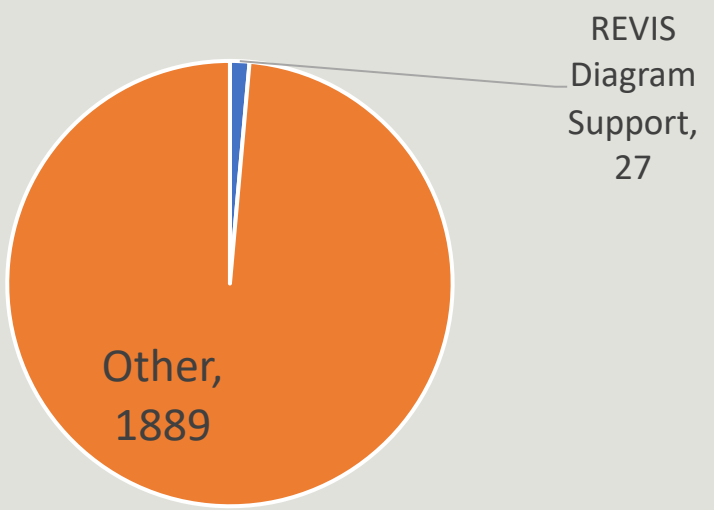
Stats Across 6 Participants



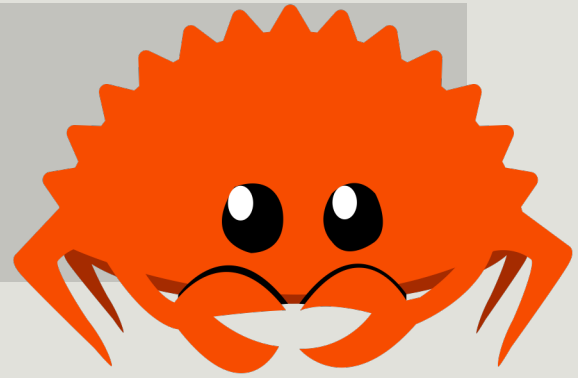
Error Messages (n = 10957)



Resolution Sessions (n = 1916)



Limitations



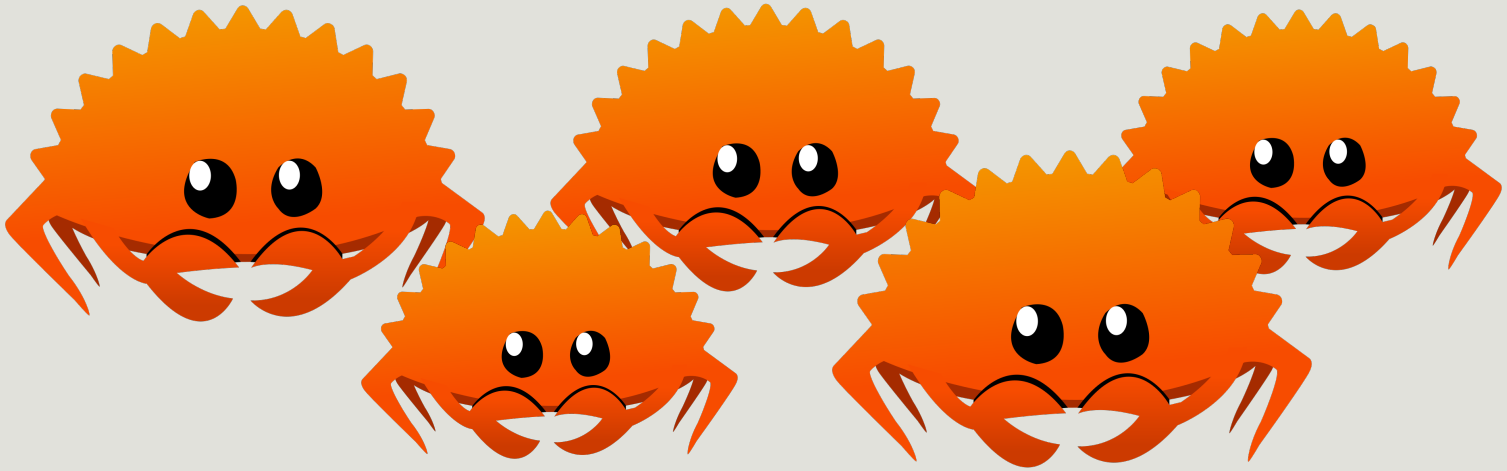
- Hard to gauge efficiency of tool
  - Most data came from those who didn’t use REVIS!
- Error reproduction may be inaccurate
- Study took place halfway through the quarter

Takeaways

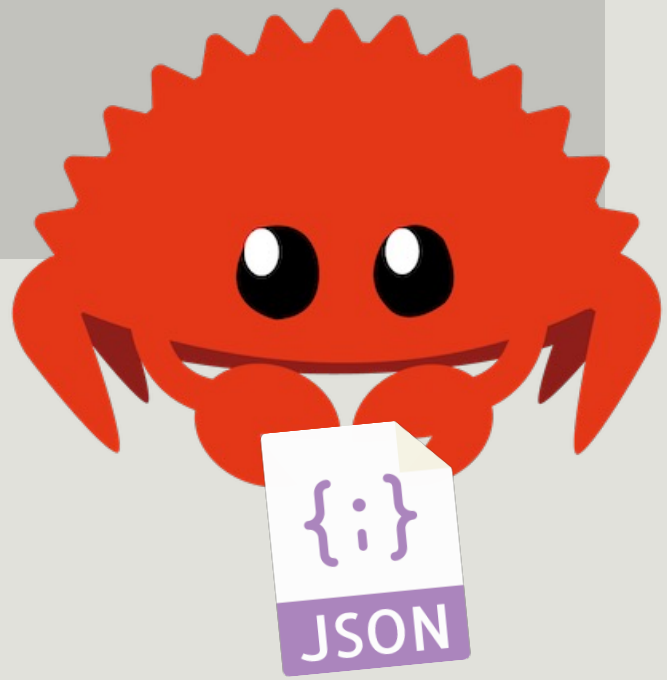
- Ownership errors took longer to resolve on average compared to other errors
- Common errors in Rust are also common in other languages
  - Future tools could be replicated for other PLs
- There are other common ownership-related errors that REVIS should support

Next Steps

- We want our error resolution tools to be used in a more general context than a grad-level compilers course.
- How can we gain a greater understanding of the errors Rust users face in more **diverse experience levels** and **programming contexts**



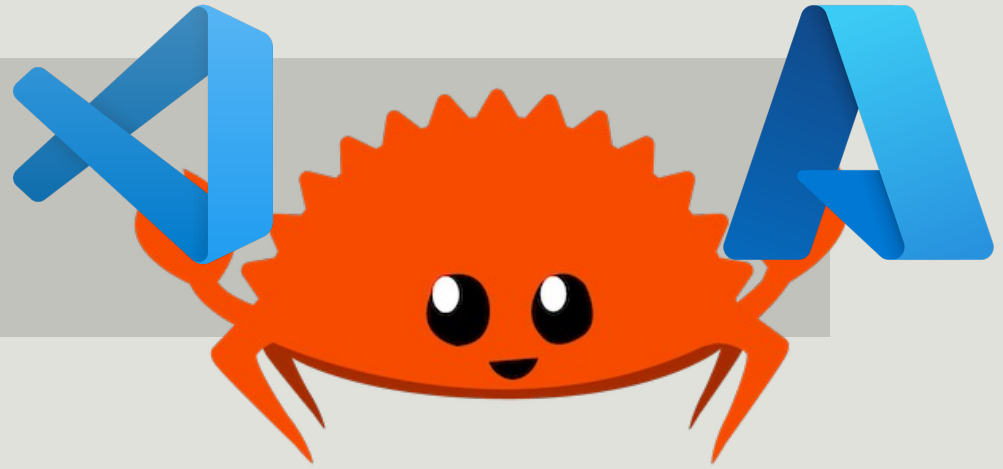
Minimally Invasive Data



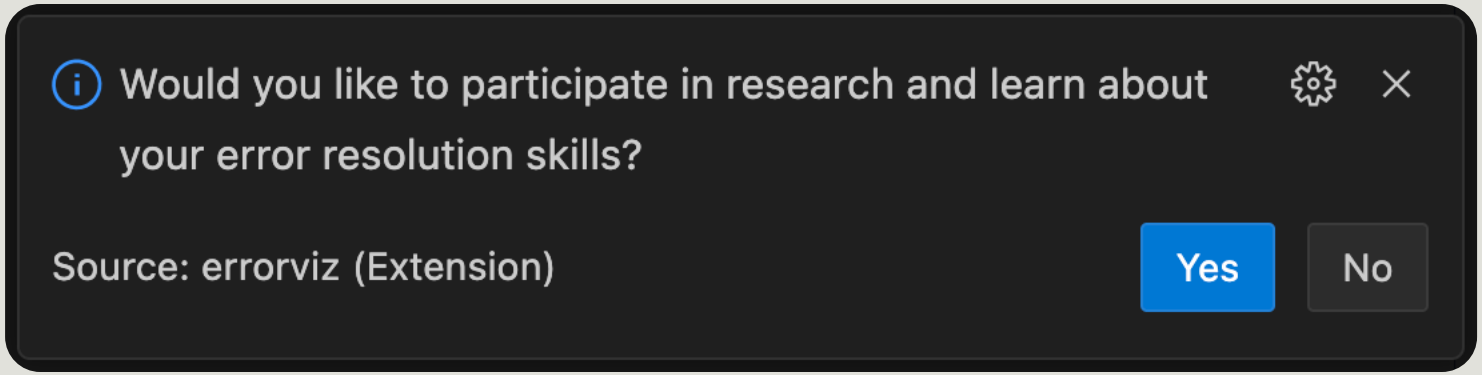
All you need to analyze error resolution times is:

- Error code
- Unique error message identifier
- Time difference

Work in Progress



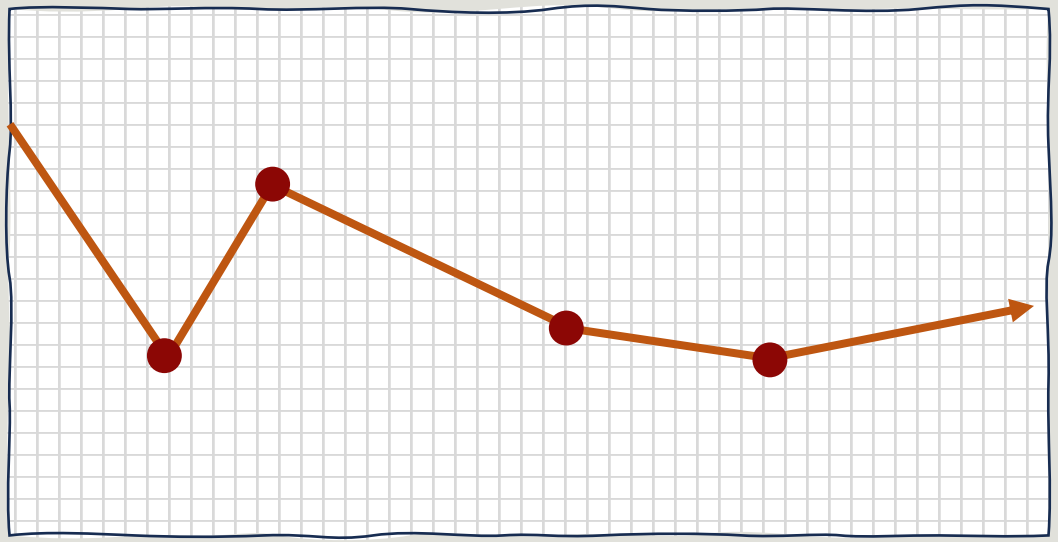
- Combined tool and telemetry!
- Error logs sent directly to server for analysis
- Users of the extension will be able to opt-in to our study
- Currently working on Institutional Review Board (IRB) paperwork for study approval



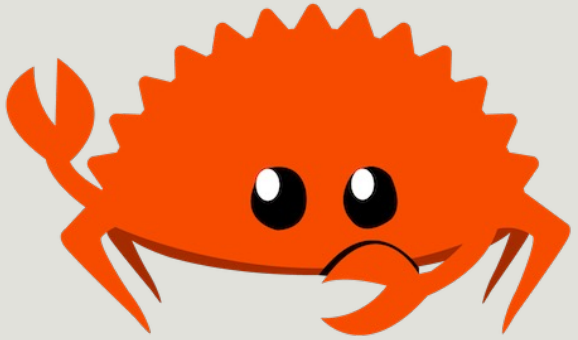
Community Impact

Iterative analysis will allow the user to track their progress over time:

- Will they become quicker at fixing the same types of errors?
- Or will they encounter more difficult instances of the same error code?



Your resolution cost over time for E0382



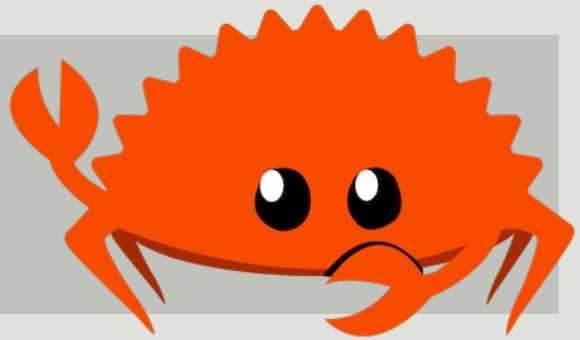
Future Analysis

As developers gain more experience:

- **Are there any errors that become more difficult to solve?**
- **Which errors show up more frequently over time?**

Focusing on these errors would help us develop tools that would assist seasoned developers, not just newcomers.

Concluding Remarks

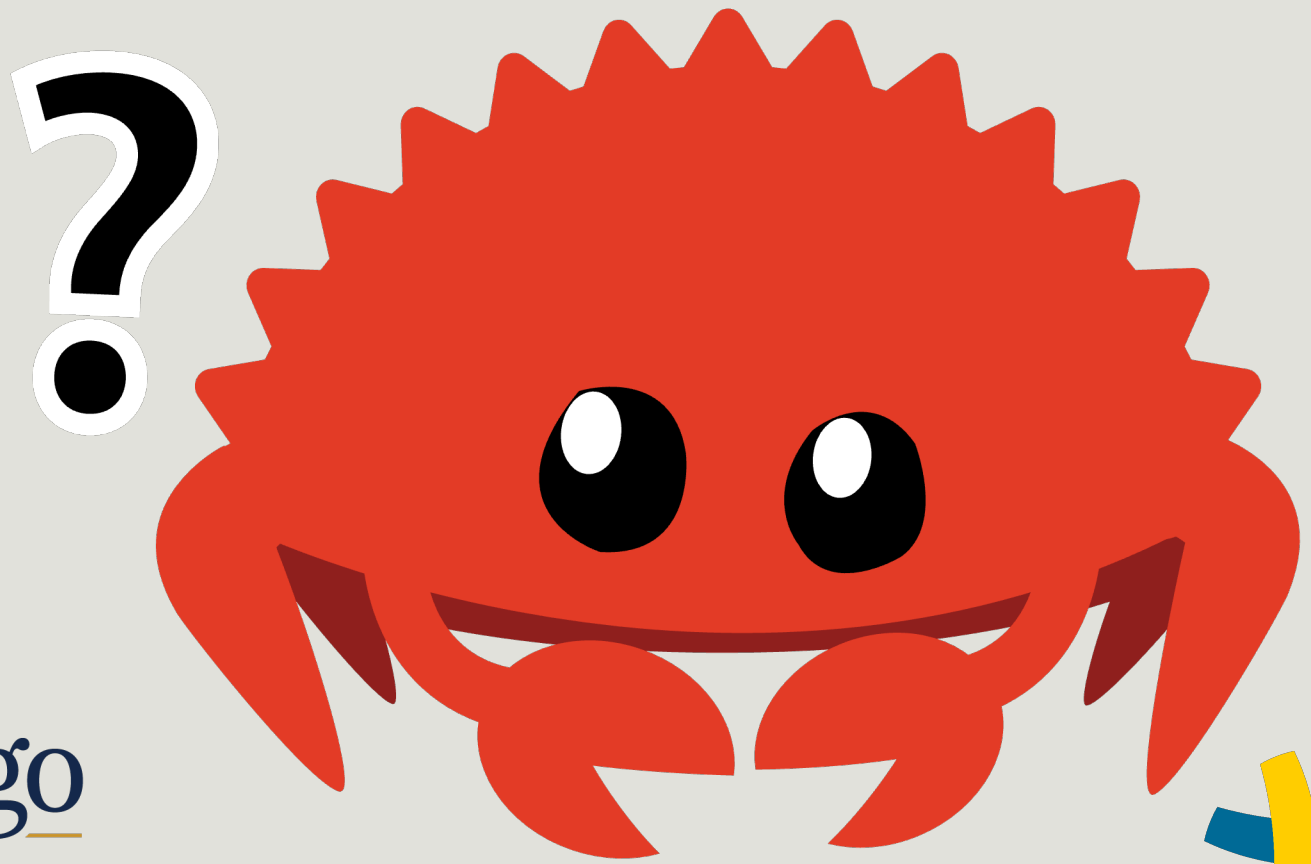


- Error analysis can help us build tools beyond just ownership and Rust
- Better tooling would help flatten the learning curve and encourage adoption better coding practices!

Thank you!

- Ruochen Wang, Michael Coblenz, and the Kale lab
- Mai Elsherief and Christine Alvarado
- The ERSP summer cohort!

Questions?



UC San Diego
JACOBS SCHOOL OF ENGINEERING
Computer Science and Engineering

ERSP
UC San Diego
Early Research Scholars Program