26° Workshop RNP

SecDevias – Generative AI for Code Vulnerability Repair

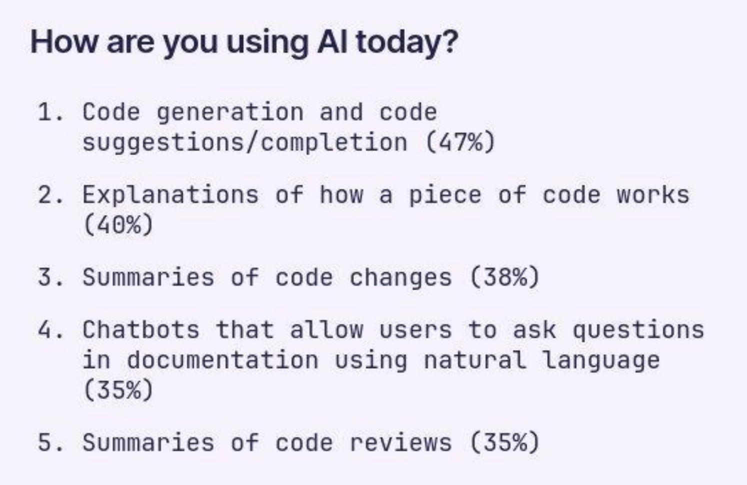
Cesar Marcondes

Instituto Tecnológico de Aeronáutica (ITA) /
Cientista Chefe Netconn

26°
Workshop
RNP

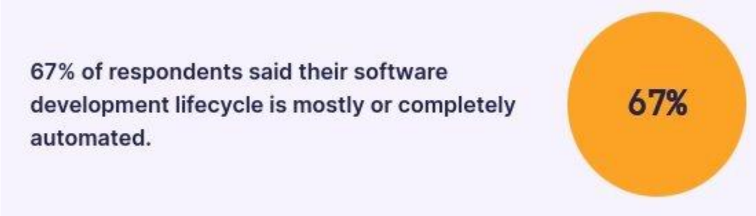
Onde o futuro se encontra.

2024
Global DevSecOps Report



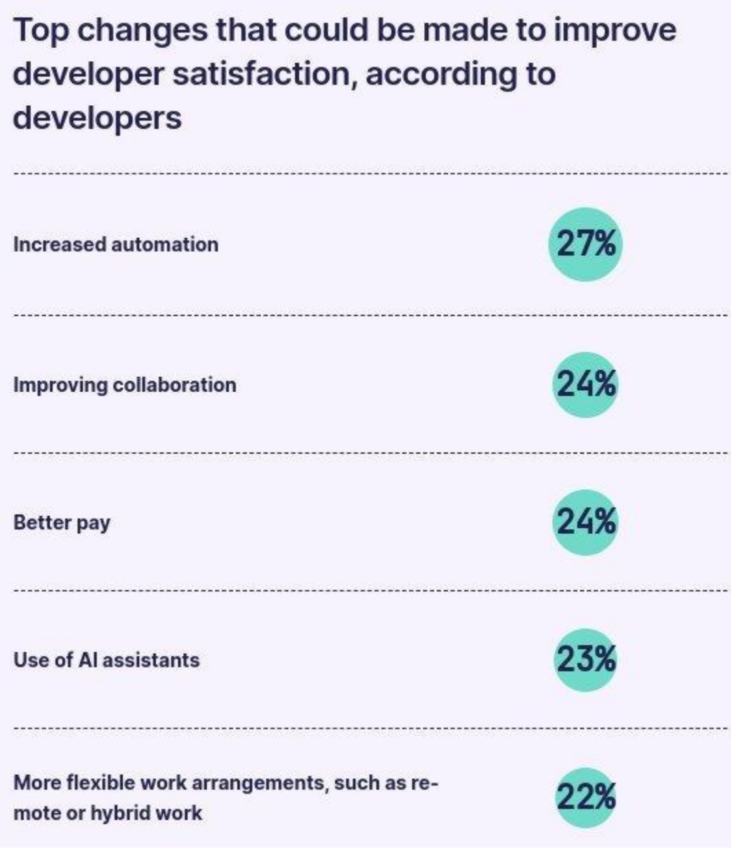
How are you using AI today?

1. Code generation and code suggestions/completion (47%)
2. Explanations of how a piece of code works (40%)
3. Summaries of code changes (38%)
4. Chatbots that allow users to ask questions in documentation using natural language (35%)
5. Summaries of code reviews (35%)



67% of respondents said their software development lifecycle is mostly or completely automated.

67%



Top changes that could be made to improve developer satisfaction, according to developers

Increased automation	27%
Improving collaboration	24%
Better pay	24%
Use of AI assistants	23%
More flexible work arrangements, such as remote or hybrid work	22%

Today:

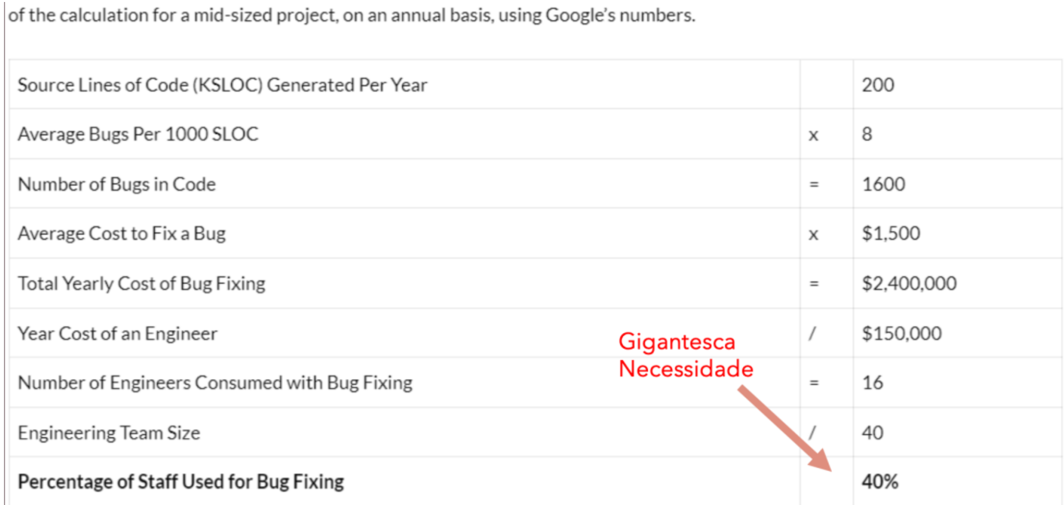
- Code Security Processes
- Devs need to deal with Features and Vulnerabilities
- Need for Custom CI/CD pipelines
- Need for Rare DevSecOps Professionals
- Security Focus is on Firewalls

26

Workshop
RNP

Onde o futuro
se encontra.

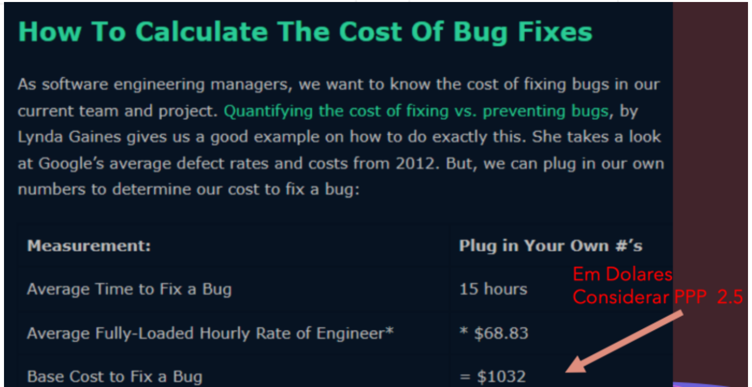
Economic Value: Study by Google (Vuln. Fixes): 40% addition team



of the calculation for a mid-sized project, on an annual basis, using Google's numbers.

Source Lines of Code (KSLOC) Generated Per Year		200
Average Bugs Per 1000 SLOC	x	8
Number of Bugs in Code	=	1600
Average Cost to Fix a Bug	x	$1,500
Total Yearly Cost of Bug Fixing	=	$2,400,000
Year Cost of an Engineer	/	$150,000
Number of Engineers Consumed with Bug Fixing	=	16
Engineering Team Size	/	40
Percentage of Staff Used for Bug Fixing		**40%**

Gigantesca
Necessidade



How To Calculate The Cost Of Bug Fixes

As software engineering managers, we want to know the cost of fixing bugs in our current team and project. Quantifying the cost of fixing vs. preventing bugs, by Lynda Gaines gives us a good example on how to do exactly this. She takes a look at Google's average defect rates and costs from 2012. But, we can plug in our own numbers to determine our cost to fix a bug:

Measurement:	**Plug in Your Own #'s**
Average Time to Fix a Bug	15 hours
Average Fully-Loaded Hourly Rate of Engineer*	* $68.83
Base Cost to Fix a Bug	= $1032

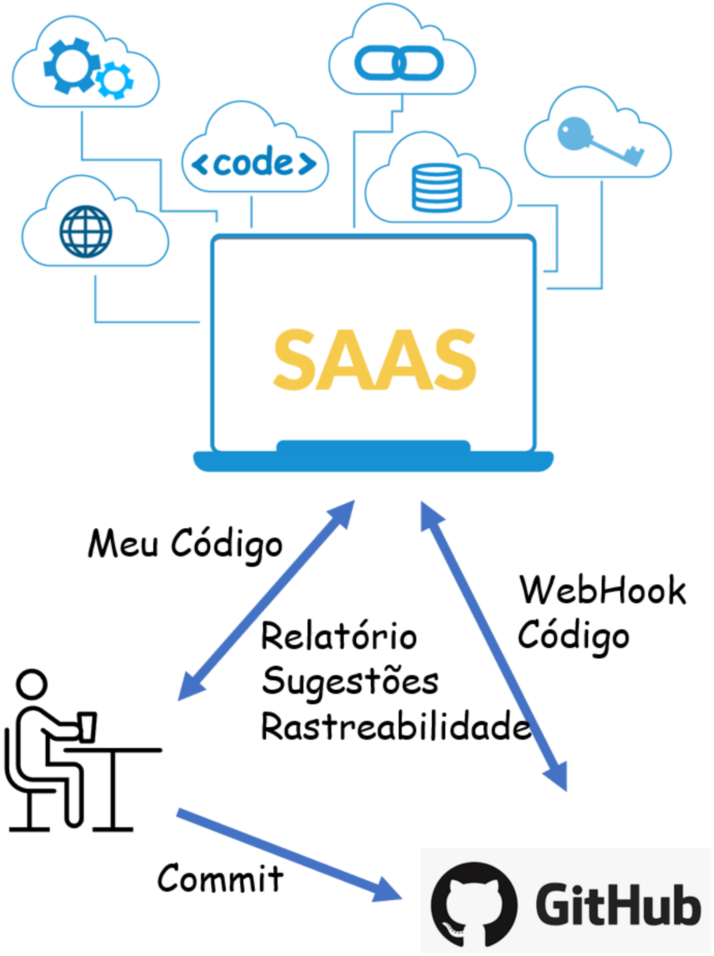
Em Dolares
Considerar PPP 2.5

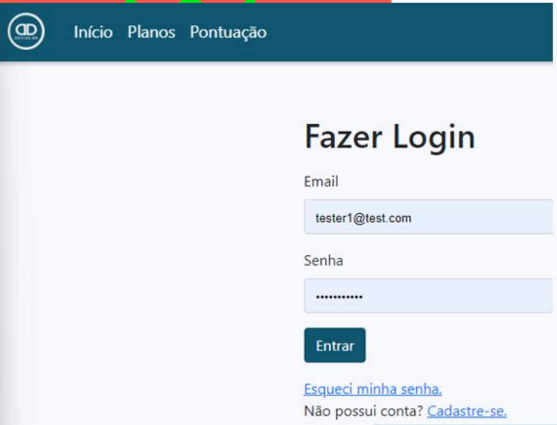
26
Workshop
RNP

Onde o futuro se encontra.

SECDEVIAS

- Automatic Pipeline Creation
- Natural Language Improved with Examples Reports (CWE/CVE)
- GPT and Transformer Based AI Scan of Vulnerability
- Suggestions to Improve Code
- Flexible SaaS: On-Prem and Cloud-Based





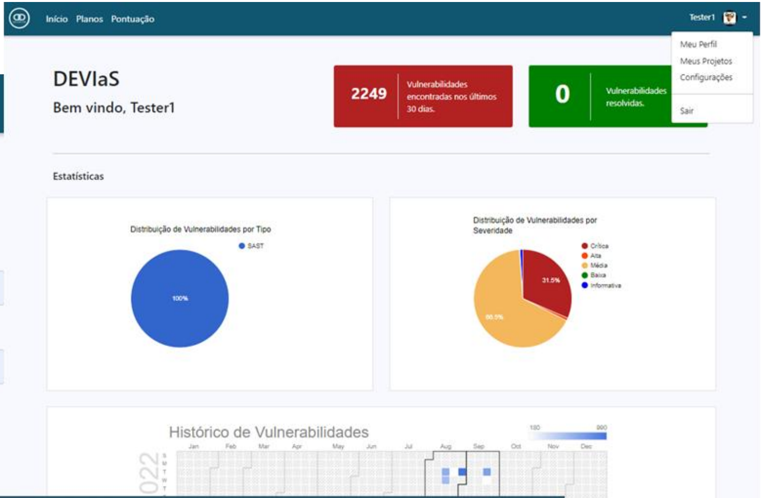
Fazer Login

Email: tester1@test.com

Senha: ••••••••••

Esqueci minha senha.

Não possui conta? Cadastre-se.



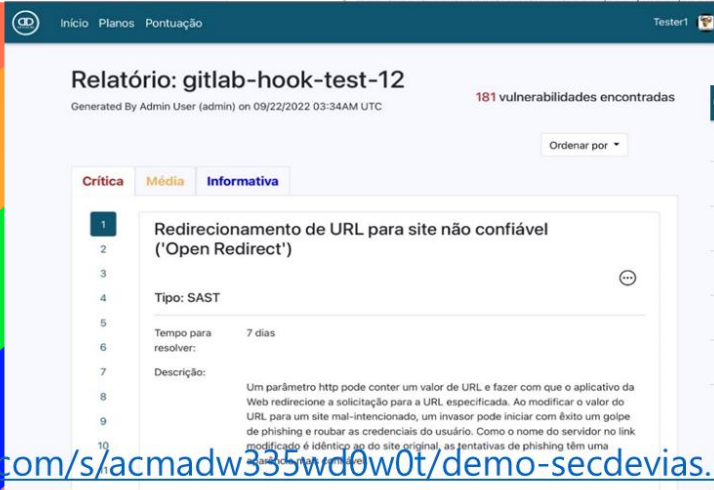
DEVIaS

Bem vindo, Tester1

2249 Vulnerabilidades encontradas nos últimos 30 dias.

0 Vulnerabilidades resolvidas.

Estatísticas



Relatório: gitlab-hook-test-12

Generated By Admin User (admin) on 09/22/2022 03:34AM UTC

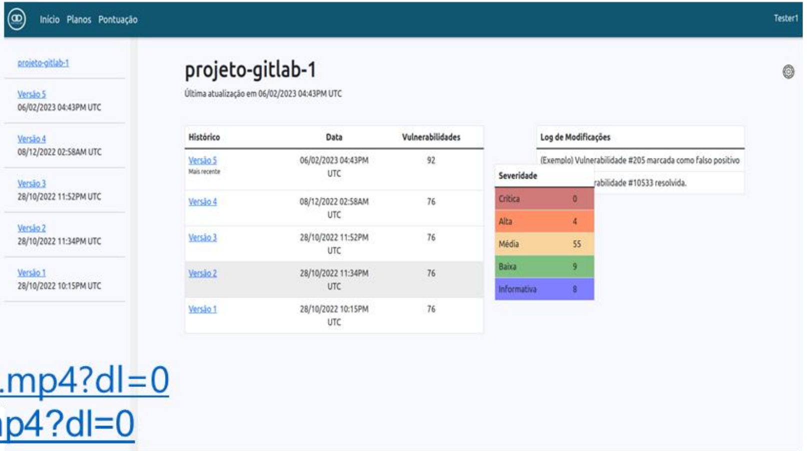
181 vulnerabilidades encontradas

Redirecionamento de URL para site não confiável ('Open Redirect')

Tipo: SAST

Tempo para resolver: 7 dias

Descrição: Um parâmetro http pode conter um valor de URL e fazer com que o aplicativo da Web redirecione a solicitação para a URL especificada. Ao modificar o valor do URL para um site mal-intencionado, um invasor pode iniciar com êxito um golpe de phishing e roubar as credenciais do usuário. Como o nome do servidor no link modificado é idêntico ao do site original, as tentativas de phishing têm uma aparência mais confiável.



projeto-gitlab-1

Versão 5 06/02/2023 04:43PM UTC

Versão 4 08/12/2022 02:58AM UTC

Versão 3 28/10/2022 11:52PM UTC

Versão 2 28/10/2022 11:34PM UTC

Versão 1 28/10/2022 10:15PM UTC

projeto-gitlab-1

Última atualização em 06/02/2023 04:43PM UTC

Histórico	Data	Vulnerabilidades
Versão 5 Mais recente	06/02/2023 04:43PM UTC	92
Versão 4	08/12/2022 02:58AM UTC	76
Versão 3	28/10/2022 11:52PM UTC	76
Versão 2	28/10/2022 11:34PM UTC	76
Versão 1	28/10/2022 10:15PM UTC	76

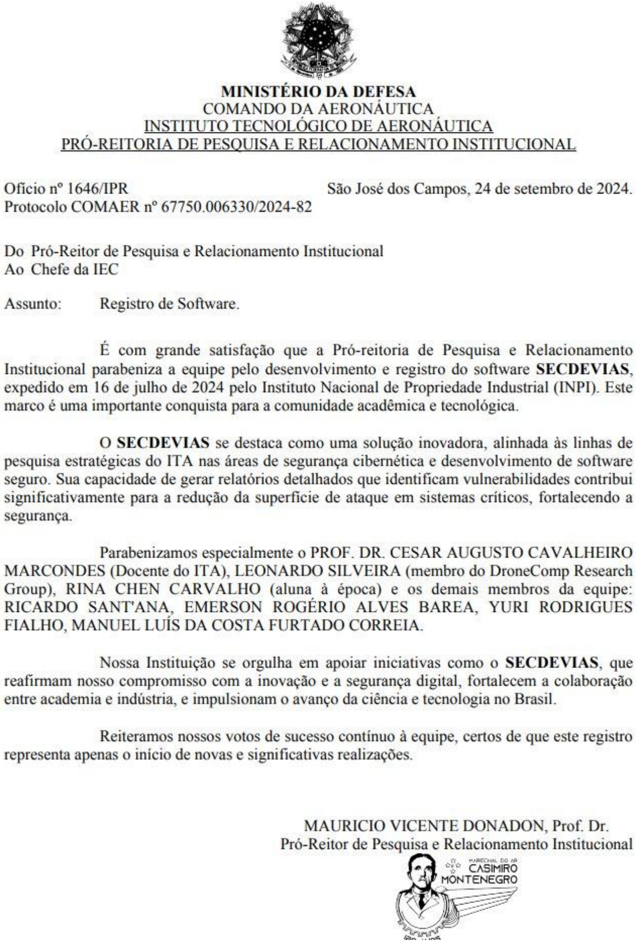
Severidade	
Crítica	0
Alta	4
Média	55
Baixa	9
Informativa	8

Log de Modificações
(Exemplo) Vulnerabilidade #205 marcada como falso positivo
rabilidade #10533 resolvida.

1. Multiplas Ferramentas
2. Esteira para Esteira
3. Dashboard Holistico do Desempenho
4. Relatório em Portugues
5. Classificado por Criticidade
6. Evolução do Projeto

https://www.dropbox.com/s/acmadw335wd0w0t/demo-secdevias.mp4?dl=0

https://www.dropbox.com/s/5vxsvg4ndlz9zhk/demo3-secdevias.mp4?dl=0



MINISTÉRIO DA DEFESA
COMANDO DA AERONÁUTICA
INSTITUTO TECNOLÓGICO DE AERONÁUTICA
PRÓ-REITORIA DE PESQUISA E RELACIONAMENTO INSTITUCIONAL

Ofício nº 1646/IPR São José dos Campos, 24 de setembro de 2024.
Protocolo COMAER nº 67750.006330/2024-82

Do Pró-Reitor de Pesquisa e Relacionamento Institucional
Ao Chefe da IEC

Assunto: Registro de Software.

É com grande satisfação que a Pró-reitoria de Pesquisa e Relacionamento Institucional parabeniza a equipe pelo desenvolvimento e registro do software **SECDEVIAS**, expedido em 16 de julho de 2024 pelo Instituto Nacional de Propriedade Industrial (INPI). Este marco é uma importante conquista para a comunidade acadêmica e tecnológica.

O **SECDEVIAS** se destaca como uma solução inovadora, alinhada às linhas de pesquisa estratégicas do ITA nas áreas de segurança cibernética e desenvolvimento de software seguro. Sua capacidade de gerar relatórios detalhados que identificam vulnerabilidades contribui significativamente para a redução da superfície de ataque em sistemas críticos, fortalecendo a segurança.

Parabenizamos especialmente o PROF. DR. CESAR AUGUSTO CAVALHEIRO MARCONDES (Docente do ITA), LEONARDO SILVEIRA (membro do DroneComp Research Group), RINA CHEN CARVALHO (aluna à época) e os demais membros da equipe: RICARDO SANT'ANA, EMERSON ROGÉRIO ALVES BAREA, YURI RODRIGUES FIALHO, MANUEL LUÍS DA COSTA FURTADO CORREIA.

Nossa Instituição se orgulha em apoiar iniciativas como o **SECDEVIAS**, que reafirmam nosso compromisso com a inovação e a segurança digital, fortalecem a colaboração entre academia e indústria, e impulsionam o avanço da ciência e tecnologia no Brasil.

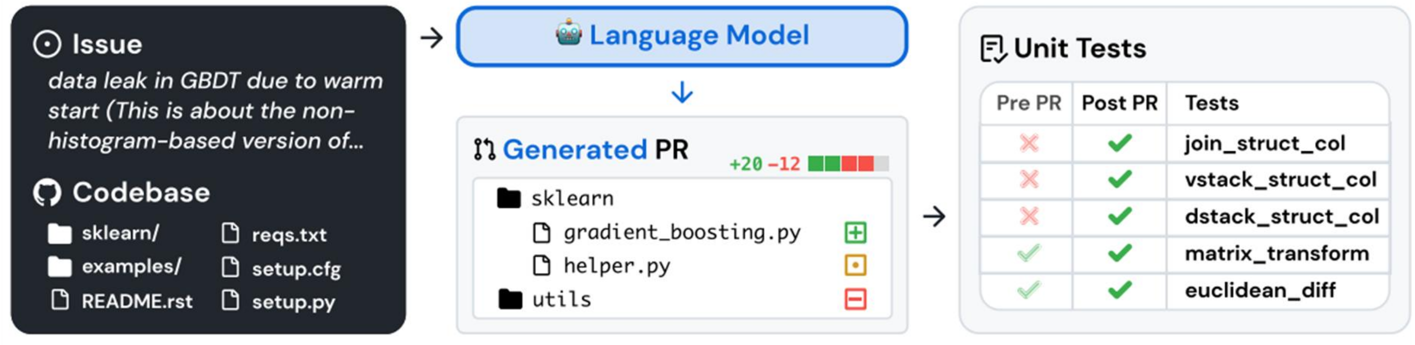
Reiteramos nossos votos de sucesso contínuo à equipe, certos de que este registro representa apenas o início de novas e significativas realizações.

MAURICIO VICENTE DONADON, Prof. Dr.
Pró-Reitor de Pesquisa e Relacionamento Institucional

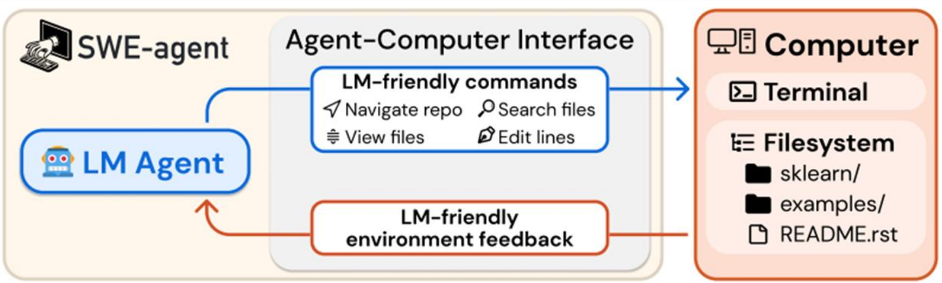
MARECHAL DO AR CASIMIRO MONTENEGRO

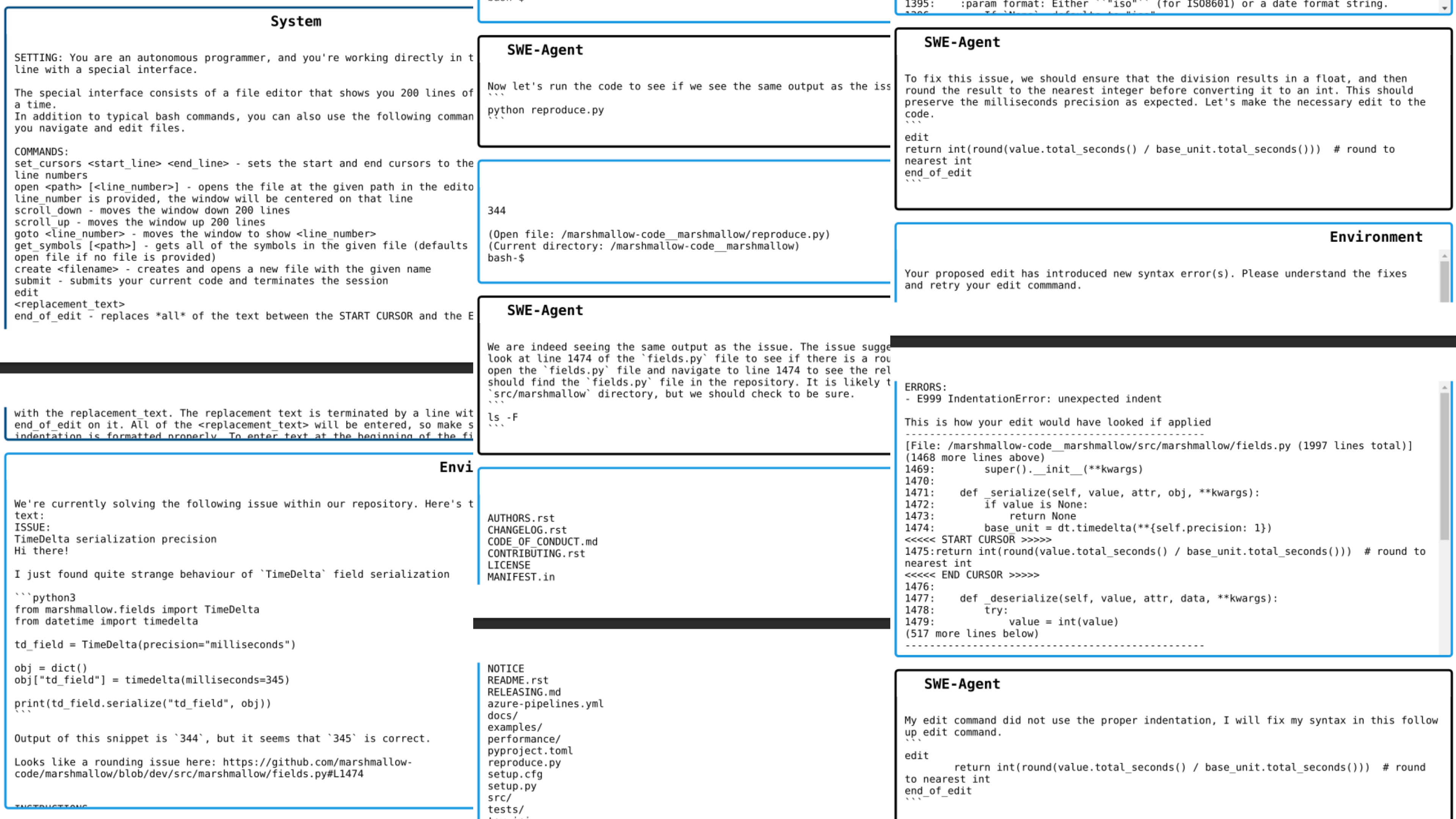
Future

SWE-BENCH: CAN LANGUAGE MODELS RESOLVE REAL-WORLD GITHUB ISSUES?



Pre PR	Post PR	Tests
✗	✓	join_struct_col
✗	✓	vstack_struct_col
✗	✓	dstack_struct_col
✓	✓	matrix_transform
✓	✓	euclidean_diff





System

```
SETTING: You are an autonomous programmer, and you're working directly in t
line with a special interface.

The special interface consists of a file editor that shows you 200 lines of
a time.
In addition to typical bash commands, you can also use the following comman
you navigate and edit files.

COMMANDS:
set_cursors <start_line> <end_line> - sets the start and end cursors to the
line numbers
open <path> [<line_number>] - opens the file at the given path in the edito
line_number is provided, the window will be centered on that line
scroll_down - moves the window down 200 lines
scroll_up - moves the window up 200 lines
goto <line_number> - moves the window to show <line_number>
get_symbols [<path>] - gets all of the symbols in the given file (defaults
open file if no file is provided)
create <filename> - creates and opens a new file with the given name
submit - submits your current code and terminates the session
edit
<replacement_text>
end_of_edit - replaces *all* of the text between the START CURSOR and the E
```

```
with the replacement_text. The replacement text is terminated by a line wit
end_of_edit on it. All of the <replacement_text> will be entered, so make s
indentation is formatted properly. To enter text at the beginning of the fi
```

Envi

~~~
We're currently solving the following issue within our repository. Here's t
text:
ISSUE:
TimeDelta serialization precision
Hi there!

I just found quite strange behaviour of `TimeDelta` field serialization

```python3
from marshmallow.fields import TimeDelta
from datetime import timedelta

td_field = TimeDelta(precision="milliseconds")

obj = dict()
obj["td_field"] = timedelta(milliseconds=345)

print(td_field.serialize("td_field", obj))
```

Output of this snippet is `344`, but it seems that `345` is correct.

Looks like a rounding issue here: https://github.com/marshmallow-
code/marshmallow/blob/dev/src/marshmallow/fields.py#L1474


INSTRUCTIONS
~~~

```
bash-$
```

SWE-Agent

```
Now let's run the code to see if we see the same output as the iss
```
python reproduce.py
```
```

```
344

(Open file: /marshmallow-code__marshmallow/reproduce.py)
(Current directory: /marshmallow-code__marshmallow)
bash-$
```

SWE-Agent

```
We are indeed seeing the same output as the issue. The issue sugge
look at line 1474 of the `fields.py` file to see if there is a rou
open the `fields.py` file and navigate to line 1474 to see the rel
should find the `fields.py` file in the repository. It is likely t
`src/marshmallow` directory, but we should check to be sure.
```
ls -F
```
```

```
AUTHORS.rst
CHANGELOG.rst
CODE_OF_CONDUCT.md
CONTRIBUTING.rst
LICENSE
MANIFEST.in
```

```
NOTICE
README.rst
RELEASING.md
azure-pipelines.yml
docs/
examples/
performance/
pyproject.toml
reproduce.py
setup.cfg
setup.py
src/
tests/
tox.ini
```

```
1395:    :param format: Either ``"iso"`` (for ISO8601) or a date format string.
1396:        If `None`, defaults to "iso".
```

SWE-Agent

```
To fix this issue, we should ensure that the division results in a float, and then
round the result to the nearest integer before converting it to an int. This should
preserve the milliseconds precision as expected. Let's make the necessary edit to the
code.
```
edit
return int(round(value.total_seconds() / base_unit.total_seconds())) # round to
nearest int
end_of_edit
```
```

Environment

```
Your proposed edit has introduced new syntax error(s). Please understand the fixes
and retry your edit commmand.
```

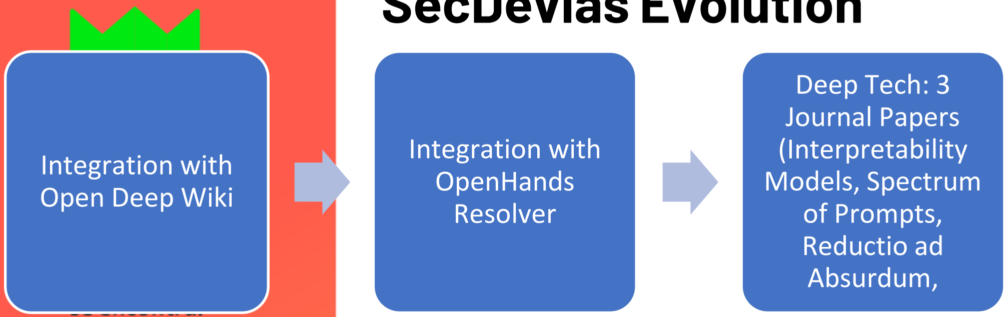
```
ERRORS:
- E999 IndentationError: unexpected indent

This is how your edit would have looked if applied
-------------------------------------------------
[File: /marshmallow-code__marshmallow/src/marshmallow/fields.py (1997 lines total)]
(1468 more lines above)
1469:        super().__init__(**kwargs)
1470:
1471:    def _serialize(self, value, attr, obj, **kwargs):
1472:        if value is None:
1473:            return None
1474:        base_unit = dt.timedelta(**{self.precision: 1})
<<<<< START CURSOR >>>>>
1475:return int(round(value.total_seconds() / base_unit.total_seconds()))  # round to
nearest int
<<<<< END CURSOR >>>>>
1476:
1477:    def _deserialize(self, value, attr, data, **kwargs):
1478:        try:
1479:            value = int(value)
(517 more lines below)
-------------------------------------------------
```

SWE-Agent

```
My edit command did not use the proper indentation, I will fix my syntax in this follow
up edit command.
```
edit
        return int(round(value.total_seconds() / base_unit.total_seconds())) # round
to nearest int
end_of_edit
```
```

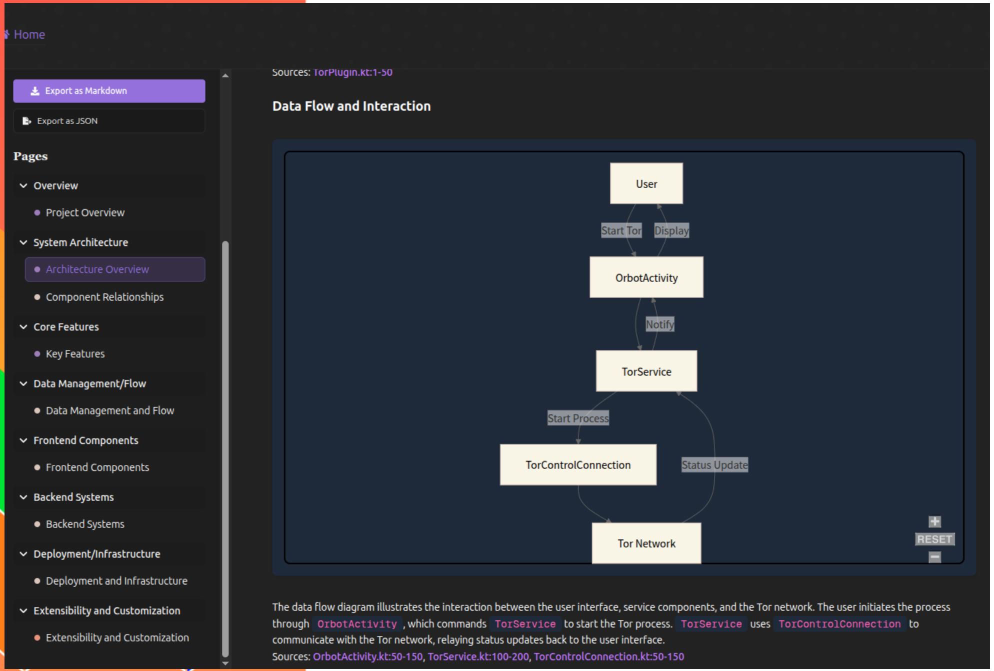
SecDevias Evolution



Integration with Open Deep Wiki

Integration with OpenHands Resolver

Deep Tech: 3 Journal Papers (Interpretability Models, Spectrum of Prompts, Reductio ad Absurdum,

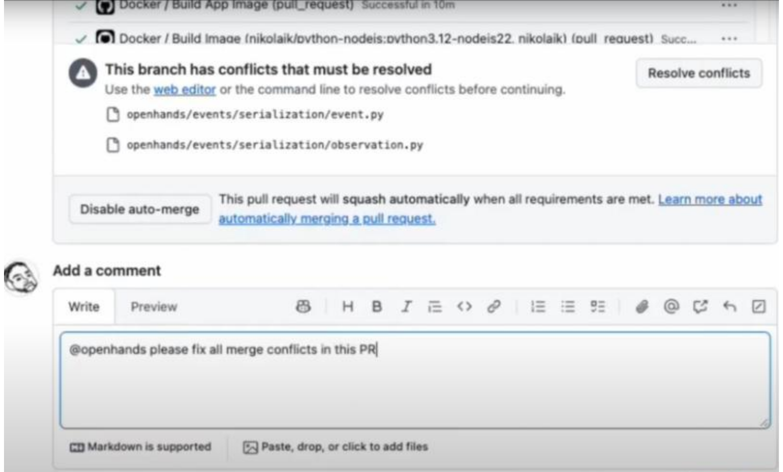


Sources: TorPlugin.kt:1-50

Data Flow and Interaction

The data flow diagram illustrates the interaction between the user interface, service components, and the Tor network. The user initiates the process through `OrbotActivity`, which commands `TorService` to start the Tor process. `TorService` uses `TorControlConnection` to communicate with the Tor network, relaying status updates back to the user interface.

Sources: OrbotActivity.kt:50-150, TorService.kt:100-200, TorControlConnection.kt:50-150


Docker / Build App Image (pull_request) Successful in 10m
Docker / Build Image (nikolaik/python-nodejs:python3.12-nodejs22, nikolaik) (pull_request) Succ...
Resolve conflicts
Disable auto-merge
Write
Preview
Markdown is supported
Paste, drop, or click to add files

This branch has conflicts that must be resolved

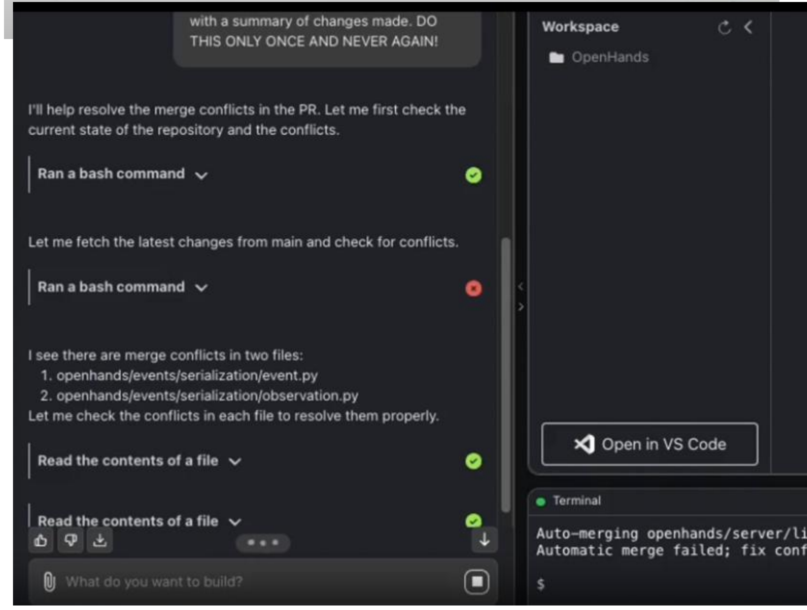
Use the web editor or the command line to resolve conflicts before continuing.

- `openhands/events/serialization/event.py`
- `openhands/events/serialization/observation.py`

This pull request will squash automatically when all requirements are met. Learn more about automatically merging a pull request.

Add a comment

@openhands please fix all merge conflicts in this PR



with a summary of changes made. DO THIS ONLY ONCE AND NEVER AGAIN!

I'll help resolve the merge conflicts in the PR. Let me first check the current state of the repository and the conflicts.

Ran a bash command ✓

Let me fetch the latest changes from main and check for conflicts.

Ran a bash command ✗

I see there are merge conflicts in two files:
1. openhands/events/serialization/event.py
2. openhands/events/serialization/observation.py

Let me check the conflicts in each file to resolve them properly.

Read the contents of a file ✓

Read the contents of a file ✓

```
Auto-merging openhands/server/li
Automatic merge failed; fix conf

$
```

26°
Workshop
RNP

Onde o futuro se encontra.

OBRIGADO(A)!

Cesar Marcondes

https://br.linkedin.com/in/cesarmarcondes