DEVEX ESSENTIALS: IGNITING CHANGE, DELIVERING RESULTS

Nicole Forsgren
Partner, Microsoft Research

INDEX.HTML

HELLO!

/DEVEX

```
# cat devex.txt grep hotness
```

INDEX.HTML

/TODAY

/01 /DEVEX 101

> What is DevEx and why we should care

/02 /THE (NEW!) RESEARCH

> DevEx seems cool, but where's the data? RIGHT HERE.

/03 /SHIFTING CULTURE

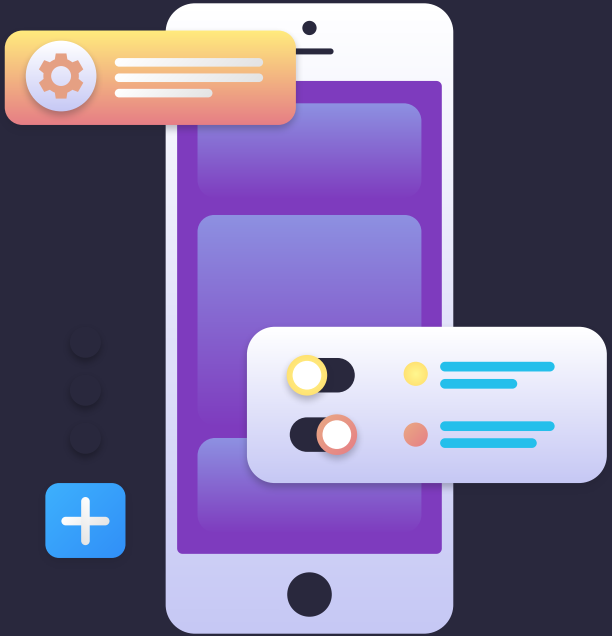
> How a DevEx lens can shift culture

/04 /DEVEX & AI

> How my team is using LLMs to help teams and orgs

/01

/DEVEX 101



/WHAT IS DEVEX?

Developer experience is the satisfaction and usability of tools, technologies, and processes that developers use in their work.

DevEx is **NOT just about tools**

/WHY SHOULD WE CARE ABOUT DEVEX?

/CENTERING USERS

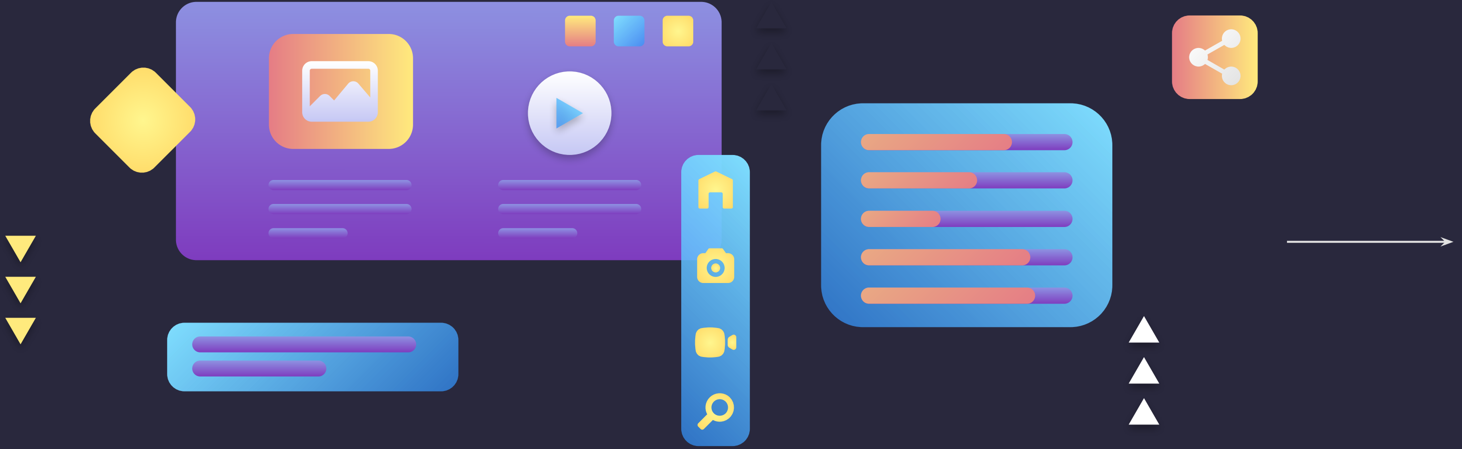
Focusing on developers helps identify system problems that are typically overlooked

/IMPROVING SYSTEMS

Good DevEx creates systems that enhance speed, reliability, and impact

/DIMENSIONS

/FLOW STATE	The mental state of being fully immersed; energized focus and enjoyment
/FEEDBACK LOOPS	The speed and quality of responses to activities performed
/COGNITIVE LOAD	The amount of mental processing required to complete a task



/02 /THE RESEARCH

/WHY MORE RESEARCH?

/IT'S NICE

Good vibes only

/IMPACTS

Impacts for *who?*

/ACTIONABLE INSIGHTS

Things anyone can do plus survey questions (soon!)

KEY FINDINGS

/DEVEX: MAKING AN IMPACT

DATA FROM 20 COMPANIES AND 200+ DEVELOPERS

/WHAT DO WE GET WITH BETTER DEVEX?

DEVELOPERS: Boost creativity, productivity, learning at work

TEAMS: Better code quality and less tech debt

ORGANIZATIONS: Drive innovation, retention, goals, and profit

A JOINT RESEARCH PROJECT BETWEEN MSR, MICROSOFT, GITHUB, AND DX. (2023)

KEY FINDINGS

/DEVEX: MAKING AN IMPACT

DATA FROM 20 COMPANIES AND 200+ DEVELOPERS

/WHAT DO WE GET WITH BETTER DEVEX?

DEVELOPERS: Boost creativity, productivity, learning at work

TEAMS: Better code quality and less tech debt

ORGANIZATIONS: Drive innovation, retention, goals, and profit

Support flow state: fewer interruptions, deeper work, and engaging tasks

⬆ 50% **//significant time for deep work** more productive vs those without dedicated time

//engaging work more productive vs those with boring work ⬆ 30%

/FLOW STATE

/DEVEX: MAKING AN IMPACT

DATA FROM 20 COMPANIES AND 200+ DEVELOPERS

/WHAT DO WE GET WITH BETTER DEVEX?

DEVELOPERS: Boost creativity, productivity, learning at work

TEAMS: Better code quality and less tech debt

ORGANIZATIONS: Drive innovation, retention, goals, and profit

Support flow state: fewer interruptions, deeper work, and engaging tasks

50% **//significant time for deep work** more productive vs those without dedicated time

//engaging work more productive vs those with boring work 30%

/FLOW STATE

Reduce cognitive load with intuitive processes, understandable code, and easier deployment

42% **//great code understandability** more productive vs those without it

//intuitive process more innovation vs those with hard-to-understand process 50%

/COGNITIVE LOAD

A JOINT RESEARCH PROJECT BETWEEN MSR, MICROSOFT, GITHUB, AND DX. (2023)

KEY FINDINGS

/DEVEX: MAKING AN IMPACT

DATA FROM 20 COMPANIES AND 200+ DEVELOPERS

/WHAT DO WE GET WITH BETTER DEVEX?

DEVELOPERS: Boost creativity, productivity, learning at work

TEAMS: Better code quality and less tech debt

ORGANIZATIONS: Drive innovation, retention, goals, and profit

Support flow state: fewer interruptions, deeper work, and engaging tasks

↑ 50% **//significant time for deep work** more productive vs those without dedicated time

//engaging work more productive vs those with boring work ↑ 30%

/FLOW STATE

Reduce cognitive load with intuitive processes, understandable code, and easier deployment

↑ 42% **//great code understandability** more productive vs those without it

//intuitive process more innovation vs those with hard-to-understand process ↑ 50%

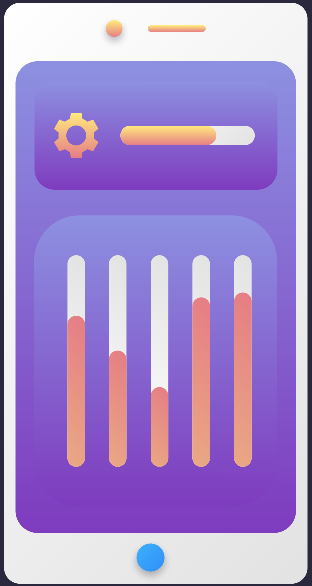
/COGNITIVE LOAD

Fast feedback loops: Fast responses for developer questions and code reviews.

↑ 20% **//fast code review turnaround times** increased innovation vs slow turnaround times

//fast responses for developer questions less tech debt vs slow responses ↓ 50%

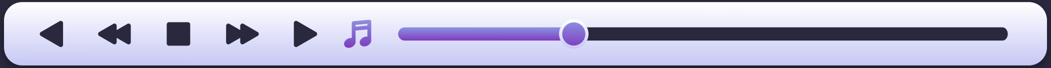
/FEEDBACK LOOPS



/03

CULTURE

INDEX.HTML



> “Culture is simply a shared way of doing something with a passion.”

—BRIAN CHESKY

/METRICS AS COMMUNICATION

/DATA

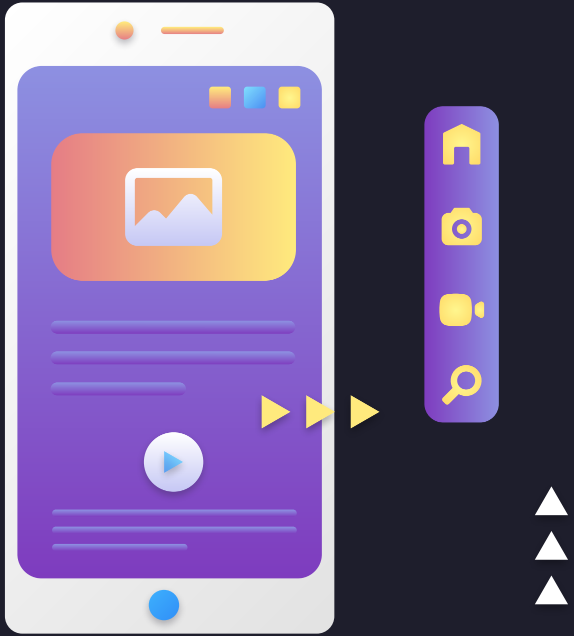
Provides opportunities to clarify and define metrics

/LANGUAGE

Can create a shared language among teams

/INSIGHTS

Can move teams from intuition to data-informed insights



/CREATING A DEVEX CULTURE AT MICROSOFT: ENGINEERING THRIVE

Anchoring on pillars of **Speed**, **Ease**, and **Quality**

/NOTICE

/NOT ONE METRIC

Suite of metrics across durable categories

/VISUAL

Metrics across categories show tradeoffs & constraints

/CATEGORIES

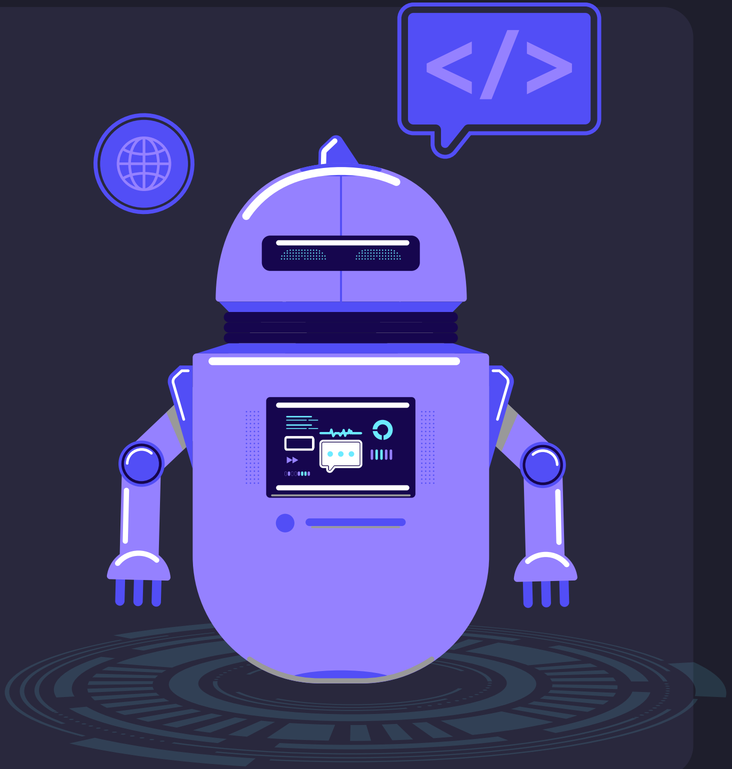
Categories are durable, metrics can evolve

/SHARED LANGUAGE

This creates shared language & understanding for change

/04

/DEVEX & AI



INDEX.HTML

/IS THIS FAMILIAR?

Efficient infrastructure is vital for software companies.

Complexity:

- Slows development
- Makes problem diagnosis hard
- Leads to repeated solutions
- Makes it difficult to understand code and context

/WHAT IF YOU HAD A “DREAM TEAM”?

/WHAT IF *EVERY ENGINEER AND TEAM* HAD A “DREAM TEAM”?

/YOUR PERSONAL LLM-POWERED EXPERTS

Giving teams expertise to unblock so they can **focus on product and strategy to drive business outcomes**

/CONSULTANT

Expert guidance, insightful system analysis, performance measurement tips

/TECH LEAD

Codebase tour, stack trace insights, and whiteboard sessions for a faster learning curve

/DATA SCIENTIST

Software engineering expertise + data analysis for actionable insights

/LLM EXPERTS TL;DR: QUESTIONS YOU CAN ASK

/CONSULTANT

- How can I improve onboarding?
- Who else has improved build times in similar contexts?
- What’s happening in my systems I should know about?

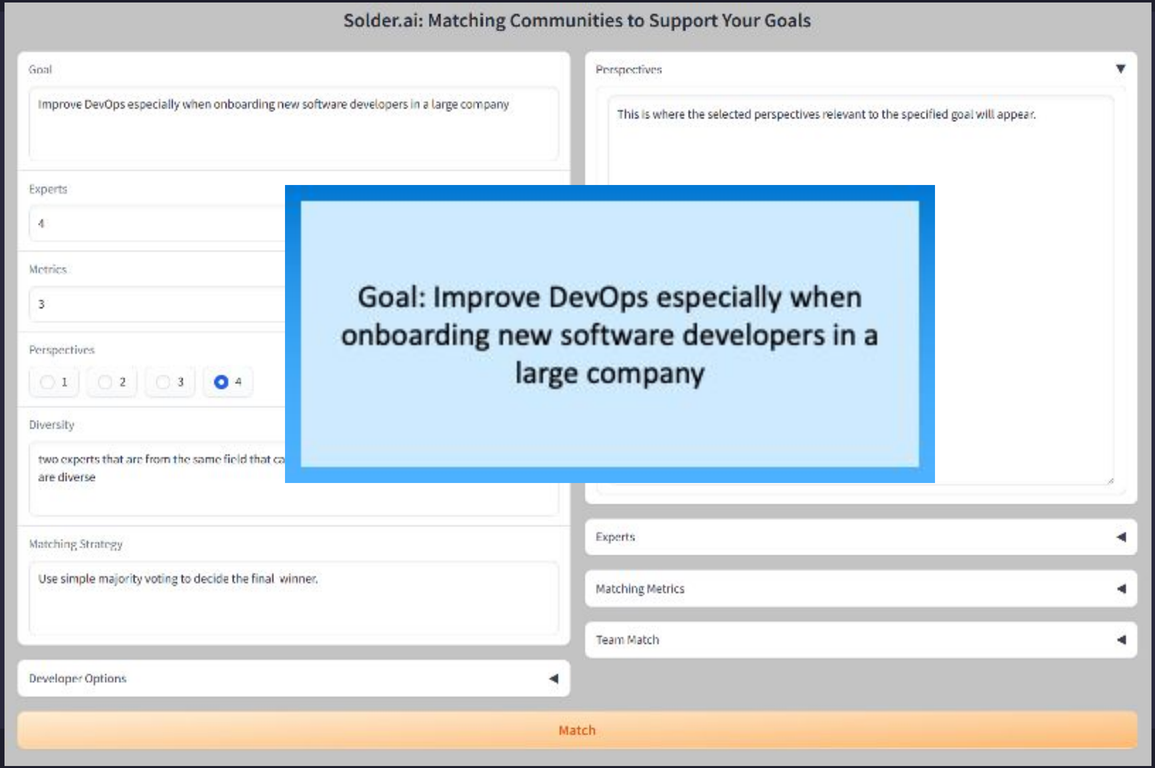
/TECH LEAD

- How does authorization work in this codebase?
- Walk me through the build process in this repo.
- Help me understand this stack trace.

/DATA SCIENTIST

- Does distributed development affect code quality in my organization?
- How does build time affect developer satisfaction?
- How does branch strategy affect PR time?

/CONSULTING EXPERT



Solder.ai: Matching Communities to Support Your Goals

Goal: Improve DevOps especially when onboarding new software developers in a large company

Experts: 4

Metrics: 3

Perspectives: ○ 1 ○ 2 ○ 3 ● 4

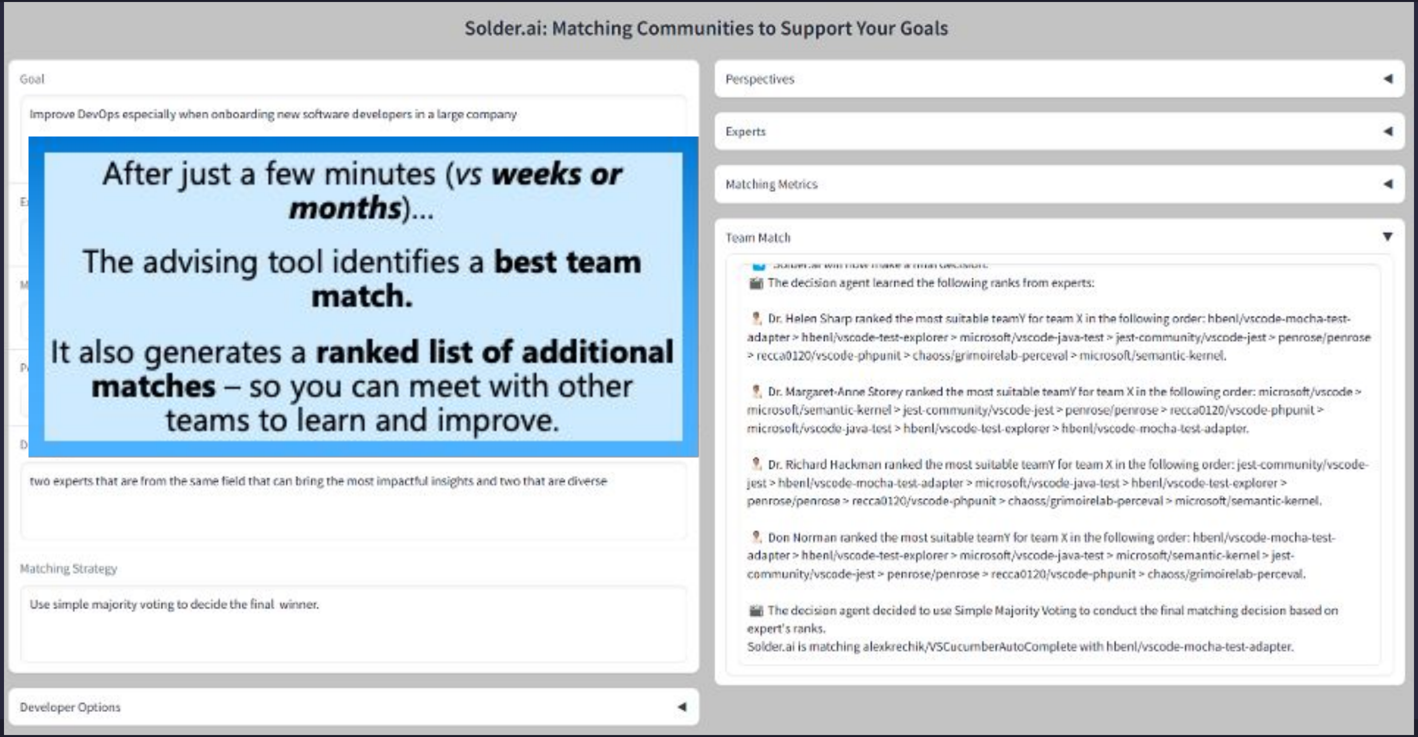
Diversity: two experts that are from the same field that ca
are diverse

Matching Strategy: Use simple majority voting to decide the final winner.

Perspectives: This is where the selected perspectives relevant to the specified goal will appear.

Goal: Improve DevOps especially when onboarding new software developers in a large company

/CONSULTING EXPERT



Solder.ai: Matching Communities to Support Your Goals

Goal

Improve DevOps especially when onboarding new software developers in a large company

After just a few minutes (*vs* ***weeks or months***)...

The advising tool identifies a **best team match.**

It also generates a **ranked list of additional matches** – so you can meet with other teams to learn and improve.

two experts that are from the same field that can bring the most impactful insights and two that are diverse

Matching Strategy

Use simple majority voting to decide the final winner.

The decision agent learned the following ranks from experts:

Dr. Helen Sharp ranked the most suitable teamY for team X in the following order: hbenl/vscode-mocha-test-adapter > hbenl/vscode-test-explorer > microsoft/vscode-java-test > jest-community/vscode-jest > penrose/penrose > recca0120/vscode-phpunit > chaoss/grimoirelab-perceval > microsoft/semantic-kernel.

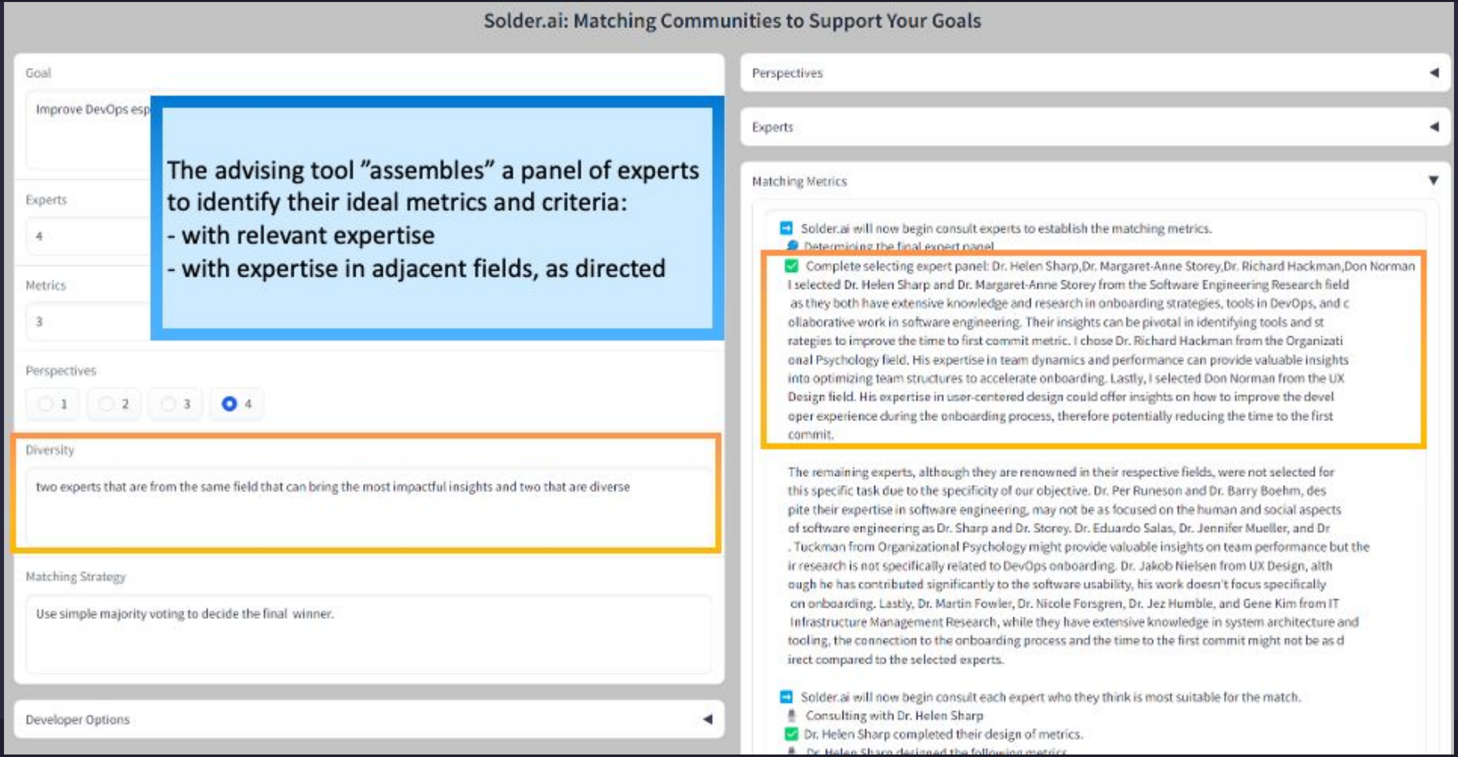
Dr. Margaret-Anne Storey ranked the most suitable teamY for team X in the following order: microsoft/vscode > microsoft/semantic-kernel > jest-community/vscode-jest > penrose/penrose > recca0120/vscode-phpunit > microsoft/vscode-java-test > hbenl/vscode-test-explorer > hbenl/vscode-mocha-test-adapter.

Dr. Richard Hackman ranked the most suitable teamY for team X in the following order: jest-community/vscode-jest > hbenl/vscode-mocha-test-adapter > microsoft/vscode-java-test > hbenl/vscode-test-explorer > penrose/penrose > recca0120/vscode-phpunit > chaoss/grimoirelab-perceval > microsoft/semantic-kernel.

Don Norman ranked the most suitable teamY for team X in the following order: hbenl/vscode-mocha-test-adapter > hbenl/vscode-test-explorer > microsoft/vscode-java-test > microsoft/semantic-kernel > jest-community/vscode-jest > penrose/penrose > recca0120/vscode-phpunit > chaoss/grimoirelab-perceval.

The decision agent decided to use Simple Majority Voting to conduct the final matching decision based on expert's ranks.
Solder.ai is matching alexkrechik/VSCucumberAutoComplete with hbenl/vscode-mocha-test-adapter.

/CONSULTING EXPERT



Solder.ai: Matching Communities to Support Your Goals

Goal: Improve DevOps esp

Experts: 4

Metrics: 3

Perspectives: ○ 1 ○ 2 ○ 3 ● 4

Diversity: two experts that are from the same field that can bring the most impactful insights and two that are diverse

Matching Strategy: Use simple majority voting to decide the final winner.

The advising tool "assembles" a panel of experts to identify their ideal metrics and criteria:
- with relevant expertise
- with expertise in adjacent fields, as directed

Solder.ai will now begin consult experts to establish the matching metrics.

Determining the final expert panel

☑ Complete selecting expert panel: Dr. Helen Sharp,Dr. Margaret-Anne Storey,Dr. Richard Hackman,Don Norman

I selected Dr. Helen Sharp and Dr. Margaret-Anne Storey from the Software Engineering Research field as they both have extensive knowledge and research in onboarding strategies, tools in DevOps, and collaborative work in software engineering. Their insights can be pivotal in identifying tools and strategies to improve the time to first commit metric. I chose Dr. Richard Hackman from the Organizational Psychology field. His expertise in team dynamics and performance can provide valuable insights into optimizing team structures to accelerate onboarding. Lastly, I selected Don Norman from the UX Design field. His expertise in user-centered design could offer insights on how to improve the developer experience during the onboarding process, therefore potentially reducing the time to the first commit.

The remaining experts, although they are renowned in their respective fields, were not selected for this specific task due to the specificity of our objective. Dr. Per Runeson and Dr. Barry Boehm, despite their expertise in software engineering, may not be as focused on the human and social aspects of software engineering as Dr. Sharp and Dr. Storey. Dr. Eduardo Salas, Dr. Jennifer Mueller, and Dr. Tuckman from Organizational Psychology might provide valuable insights on team performance but their research is not specifically related to DevOps onboarding. Dr. Jakob Nielsen from UX Design, although he has contributed significantly to the software usability, his work doesn't focus specifically on onboarding. Lastly, Dr. Martin Fowler, Dr. Nicole Forsgren, Dr. Jez Humble, and Gene Kim from IT Infrastructure Management Research, while they have extensive knowledge in system architecture and tooling, the connection to the onboarding process and the time to the first commit might not be as direct compared to the selected experts.

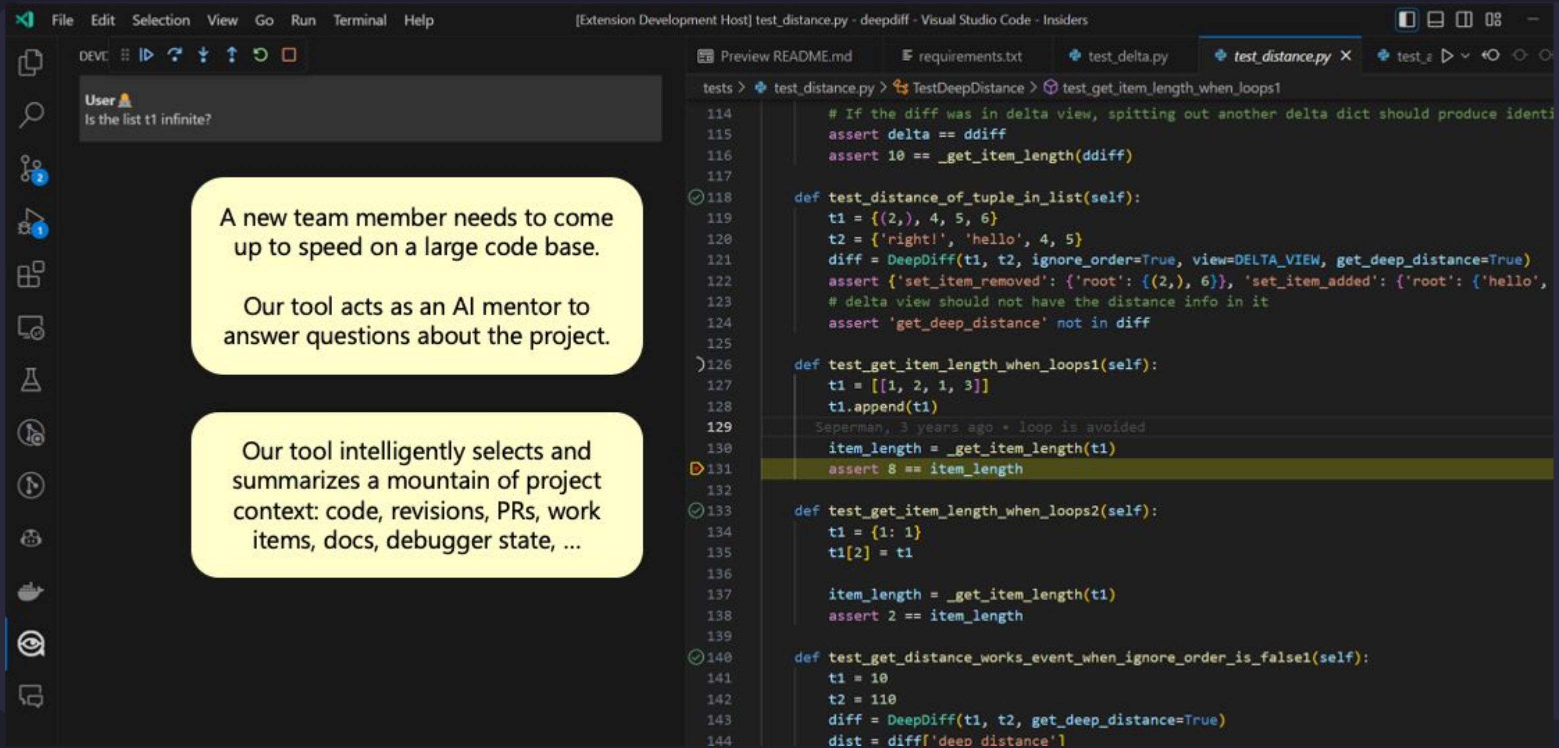
Solder.ai will now begin consult each expert who they think is most suitable for the match.

Consulting with Dr. Helen Sharp

☑ Dr. Helen Sharp completed their design of metrics.

Dr. Helen Sharp designed the following metrics

/TECH LEAD EXPERT


File
Edit
Selection
View
Go
Run
Terminal
Help
[Extension Development Host] test_distance.py - deepdiff - Visual Studio Code - Insiders
DEVC
Preview README.md
requirements.txt
test_delta.py
test_distance.py
test_d
tests > test_distance.py > TestDeepDistance > test_get_item_length_when_loops1

User
Is the list t1 infinite?

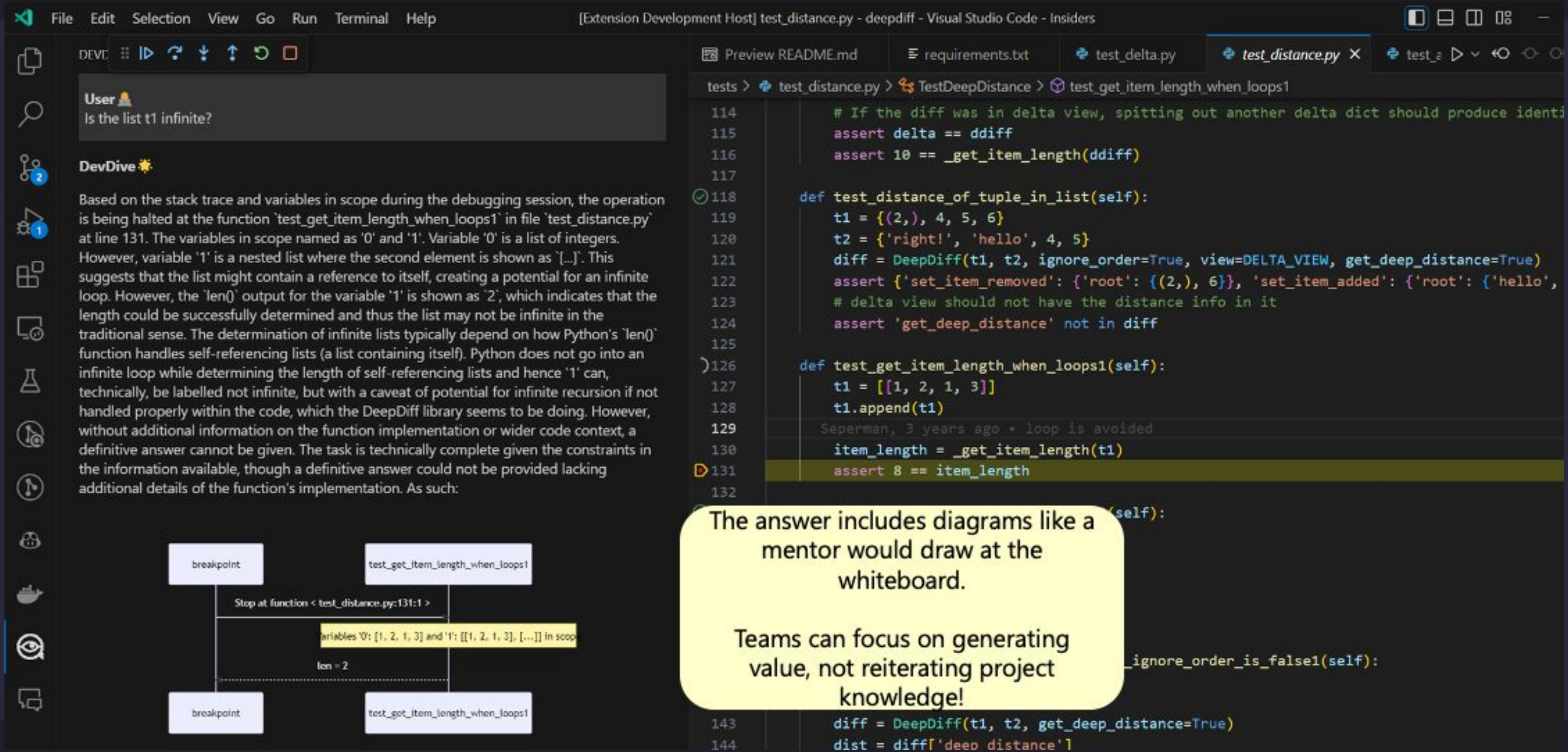
A new team member needs to come up to speed on a large code base.

Our tool acts as an AI mentor to answer questions about the project.

Our tool intelligently selects and summarizes a mountain of project context: code, revisions, PRs, work items, docs, debugger state, ...

```
        # If the diff was in delta view, spitting out another delta dict should produce identi
        assert delta == ddiff
        assert 10 == _get_item_length(ddiff)

    def test_distance_of_tuple_in_list(self):
        t1 = {(2,), 4, 5, 6}
        t2 = {'right!', 'hello', 4, 5}
        diff = DeepDiff(t1, t2, ignore_order=True, view=DELTA_VIEW, get_deep_distance=True)
        assert {'set_item_removed': {'root': {(2,), 6}}, 'set_item_added': {'root': {'hello',
        # delta view should not have the distance info in it
        assert 'get_deep_distance' not in diff

    def test_get_item_length_when_loops1(self):
        t1 = [[1, 2, 1, 3]]
        t1.append(t1)
      Seperman, 3 years ago • loop is avoided
        item_length = _get_item_length(t1)
        assert 8 == item_length

    def test_get_item_length_when_loops2(self):
        t1 = {1: 1}
        t1[2] = t1

        item_length = _get_item_length(t1)
        assert 2 == item_length

    def test_get_distance_works_event_when_ignore_order_is_false1(self):
        t1 = 10
        t2 = 110
        diff = DeepDiff(t1, t2, get_deep_distance=True)
        dist = diff['deep_distance']
```

/TECH LEAD EXPERT


File
Edit
Selection
View
Go
Run
Terminal
Help
[Extension Development Host] test_distance.py - deepdiff - Visual Studio Code - Insiders
DEVC
Preview README.md
requirements.txt
test_delta.py
test_distance.py
test_a
tests > test_distance.py > TestDeepDistance > test_get_item_length_when_loops1

User
Is the list t1 infinite?

DevDive

Based on the stack trace and variables in scope during the debugging session, the operation is being halted at the function `test_get_item_length_when_loops1` in file `test_distance.py` at line 131. The variables in scope named as '0' and '1'. Variable '0' is a list of integers. However, variable '1' is a nested list where the second element is shown as `[...]`. This suggests that the list might contain a reference to itself, creating a potential for an infinite loop. However, the `len()` output for the variable '1' is shown as `2`, which indicates that the length could be successfully determined and thus the list may not be infinite in the traditional sense. The determination of infinite lists typically depend on how Python's `len()` function handles self-referencing lists (a list containing itself). Python does not go into an infinite loop while determining the length of self-referencing lists and hence '1' can, technically, be labelled not infinite, but with a caveat of potential for infinite recursion if not handled properly within the code, which the DeepDiff library seems to be doing. However, without additional information on the function implementation or wider code context, a definitive answer cannot be given. The task is technically complete given the constraints in the information available, though a definitive answer could not be provided lacking additional details of the function's implementation. As such:

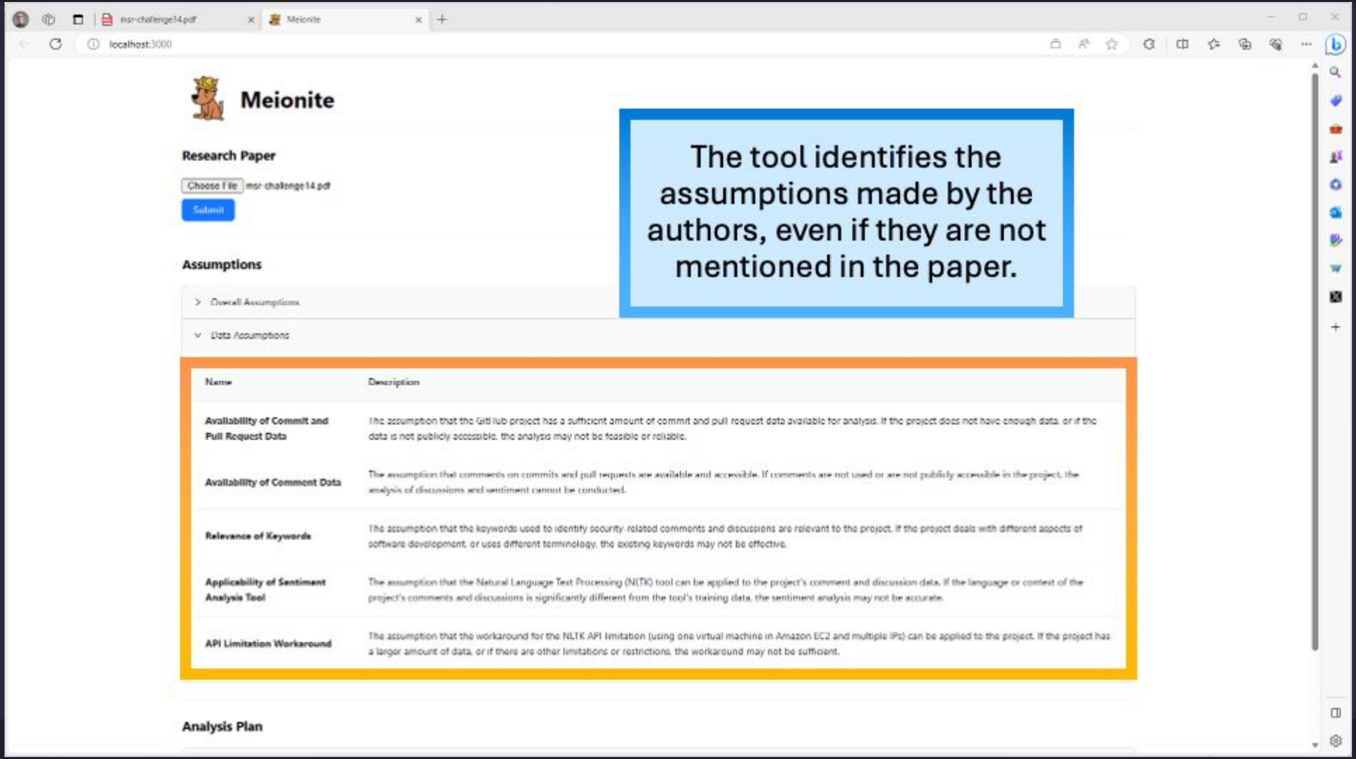
breakpoint
test_get_item_length_when_loops1
Stop at function < test_distance.py:131:1 >
ariables '0': [1, 2, 1, 3] and '1': [[1, 2, 1, 3], [...]] in scop
len = 2
breakpoint
test_get_item_length_when_loops1

```
114            # If the diff was in delta view, spitting out another delta dict should produce identi
115            assert delta == ddiff
116            assert 10 == _get_item_length(ddiff)
117
118        def test_distance_of_tuple_in_list(self):
119            t1 = {(2,), 4, 5, 6}
120            t2 = {'right!', 'hello', 4, 5}
121            diff = DeepDiff(t1, t2, ignore_order=True, view=DELTA_VIEW, get_deep_distance=True)
122            assert {'set_item_removed': {'root': {(2,), 6}}, 'set_item_added': {'root': {'hello',
123            # delta view should not have the distance info in it
124            assert 'get_deep_distance' not in diff
125
126        def test_get_item_length_when_loops1(self):
127            t1 = [[1, 2, 1, 3]]
128            t1.append(t1)
129          Seperman, 3 years ago • loop is avoided
130            item_length = _get_item_length(t1)
131            assert 8 == item_length
132
                                            (self):

                                            _ignore_order_is_false1(self):

143            diff = DeepDiff(t1, t2, get_deep_distance=True)
144            dist = diff['deep_distance']
```

The answer includes diagrams like a mentor would draw at the whiteboard.

Teams can focus on generating value, not reiterating project knowledge!

/DATA SCIENCE EXPERT



The tool identifies the assumptions made by the authors, even if they are not mentioned in the paper.

Meionite

Research Paper

Assumptions

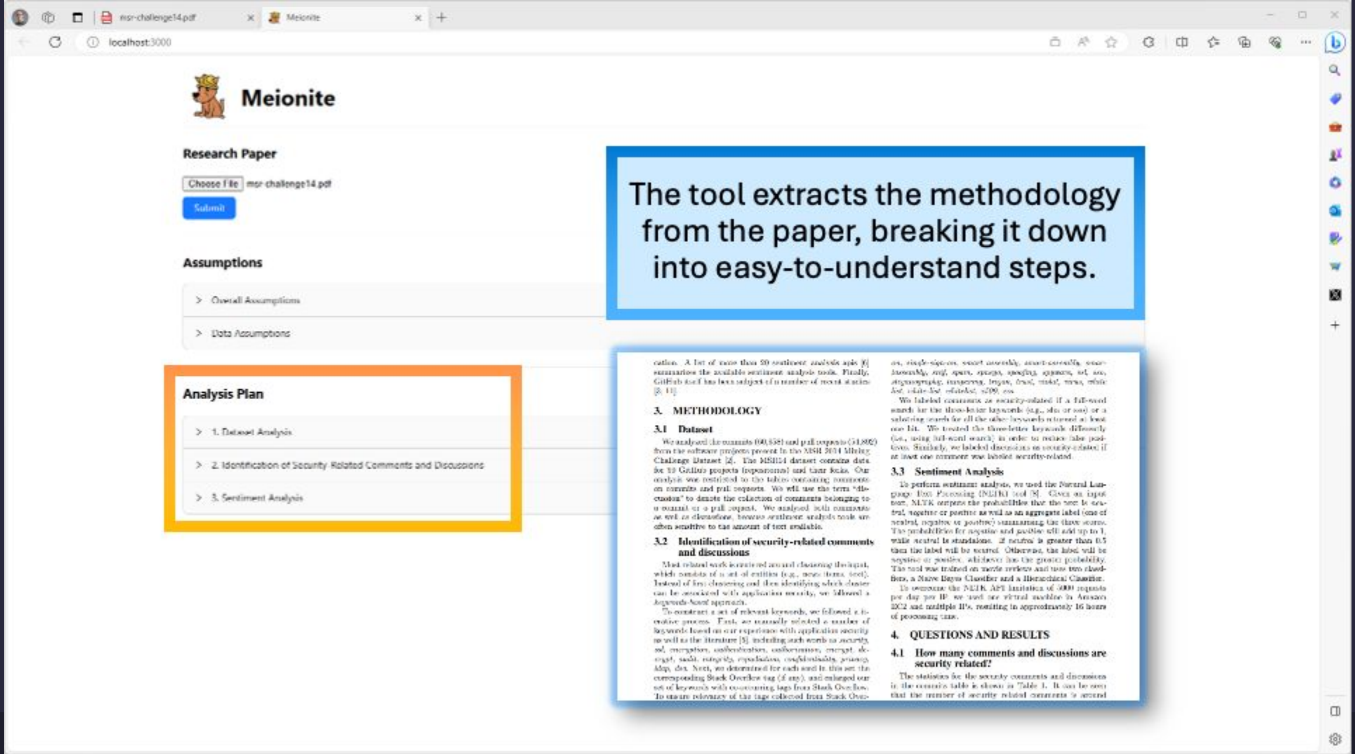
> Overall Assumptions

⌄ Data Assumptions

Name	Description
Availability of Commit and Pull Request Data	The assumption that the GitHub project has a sufficient amount of commit and pull request data available for analysis. If the project does not have enough data, or if the data is not publicly accessible, the analysis may not be feasible or reliable.
Availability of Comment Data	The assumption that comments on commits and pull requests are available and accessible. If comments are not used or are not publicly accessible in the project, the analysis of discussions and sentiment cannot be conducted.
Relevance of Keywords	The assumption that the keywords used to identify security-related comments and discussions are relevant to the project. If the project deals with different aspects of software development, or uses different terminology, the existing keywords may not be effective.
Applicability of Sentiment Analysis Tool	The assumption that the Natural Language Text Processing (NLTK) tool can be applied to the project's comment and discussion data. If the language or context of the project's comments and discussions is significantly different from the tool's training data, the sentiment analysis may not be accurate.
API Limitation Workaround	The assumption that the workaround for the NLTK API limitation (using one virtual machine in Amazon EC2 and multiple IPs) can be applied to the project. If the project has a larger amount of data, or if there are other limitations or restrictions, the workaround may not be sufficient.

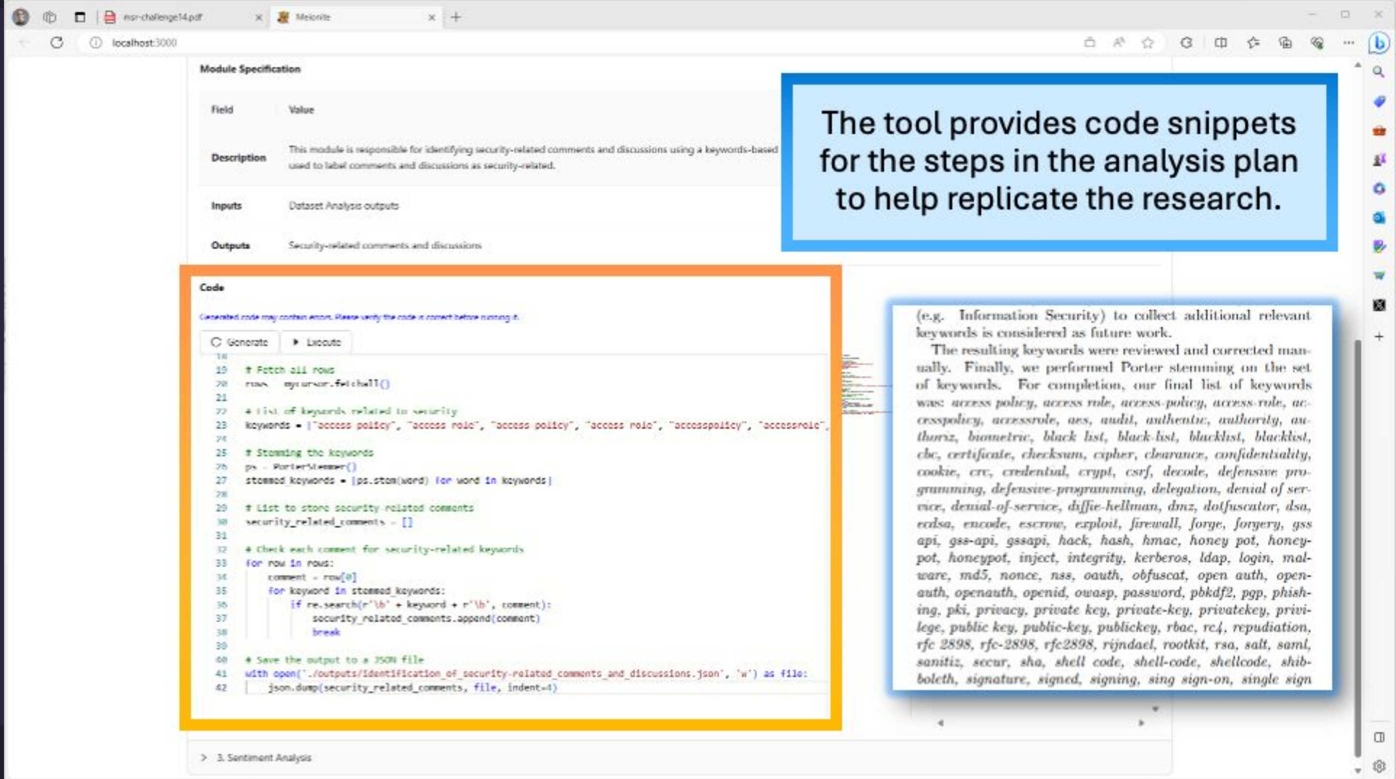
Analysis Plan

/DATA SCIENCE EXPERT



The tool extracts the methodology from the paper, breaking it down into easy-to-understand steps.

/DATA SCIENCE EXPERT



Module Specification

Field	Value
Description	This module is responsible for identifying security-related comments and discussions using a keywords-based used to label comments and discussions as security-related.
Inputs	Dataset Analysis outputs
Outputs	Security-related comments and discussions

Code

Generated code may contain errors. Please verify the code is correct before running it.

```
# Fetch all rows
rows = mycursor.fetchall()

# List of keywords related to security
keywords = ["access policy", "access role", "access policy", "access role", "accesspolicy", "accessrole",

# Stemming the keywords
ps = PorterStemmer()
stemmed_keywords = [ps.stem(word) for word in keywords]

# List to store security related comments
security_related_comments = []

# Check each comment for security-related keywords
for row in rows:
    comment = row[0]
    for keyword in stemmed_keywords:
        if re.search(r'\b' + keyword + r'\b', comment):
            security_related_comments.append(comment)
            break

# Save the output to a JSON file
with open('./outputs/identification_of_security-related_comments_and_discussions.json', 'w') as file:
    json.dump(security_related_comments, file, indent=4)
```

The tool provides code snippets for the steps in the analysis plan to help replicate the research.

(e.g. Information Security) to collect additional relevant keywords is considered as future work.

The resulting keywords were reviewed and corrected manually. Finally, we performed Porter stemming on the set of keywords. For completion, our final list of keywords was: *access policy, access role, access-policy, access-role, accesspolicy, accessrole, aes, audit, authentic, authority, authoriz, biometric, black list, black-list, blacklist, blacklist, cbc, certificate, checksum, cipher, clearance, confidentiality, cookie, crc, credential, crypt, csrf, decode, defensive programming, defensive-programming, delegation, denial of service, denial-of-service, diffie-hellman, dmz, dotfuscator, dsa, ecdsa, encode, escrow, exploit, firewall, forge, forgery, gss api, gss-api, gssapi, hack, hash, hmac, honey pot, honey-pot, honeypot, inject, integrity, kerberos, ldap, login, malware, md5, nonce, nss, oauth, obfuscat, open auth, open-auth, openauth, openid, owasp, password, pbkdf2, pgp, phishing, pki, privacy, private key, private-key, privatekey, privilege, public key, public-key, publickey, rbac, rc4, repudiation, rfc 2898, rfc-2898, rfc2898, rijndael, rootkit, rsa, salt, saml, sanitiz, secur, sha, shell code, shell-code, shellcode, shibboleth, signature, signed, signing, sing sign-on, single sign*

/WHAT'S NEXT – WHAT I'D LOVE FROM YOU!

/SHARE

What problems do you have? What LLM experts could help?

/CO-INNOVATE

Create and innovate solutions with us

/READ THE RESEARCH

Watch for the latest DevEx research and more details!

/LET'S CO-CREATE THE FUTURE OF DEVEX

/ACKNOWLEDGEMENTS*

/DEVEX RESEARCH

- MSR & Microsoft: Nicole Forsgren (Co-PI), Brian Houck
- GitHub: Eirini Kalliamvakou (Co-PI)
- DX: Michaela Greiler, Abi Noda, Margaret-Anne Storey

/ENG THRIVE

- Exec Sponsor: Tim Bozarth
- Co-Leads: Nicole Forsgren, Caitie McCaffrey, David Speirs
- Working Group: Folks across all business units

/AI FOR DEVEX

- SAINTES team: Carmen Badea, Christian Bird, Rob DeLine, Denae Ford, Nicole Forsgren, Tom Zimmermann
- Interns: Ebtesam Al Haque, Ti-Chung Cheng, Jenny Liang, Anastasia Ruvimova

*ABC order

/THANKS!

/DO YOU HAVE ANY QUESTIONS?

nicolefv@gmail.com
nicole.forsgren@microsoft.com

SAINT-studies@microsoft.com

https://bit.ly/devex-impacts