Assessing the Quality of the Steps to Reproduce in Bug Reports

William & Mary
University of Texas at Dallas
University of Sannio

27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'19)

Wednesday, August 28th, 2019



Oscar Chaparro



Carlos Bernal



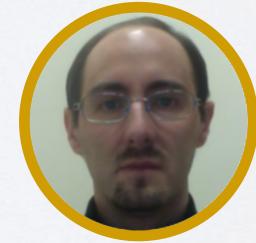
Jing Lu



<u>Kevin</u> <u>Moran</u>



Andrian Marcus



Massimiliano Di Penta

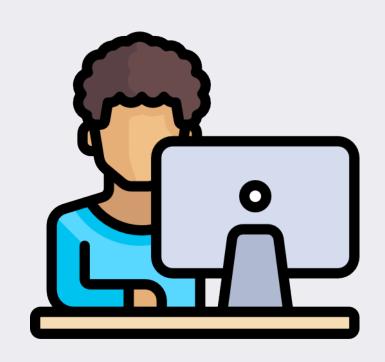


Denys Poshyvanyk



Vincent Ng

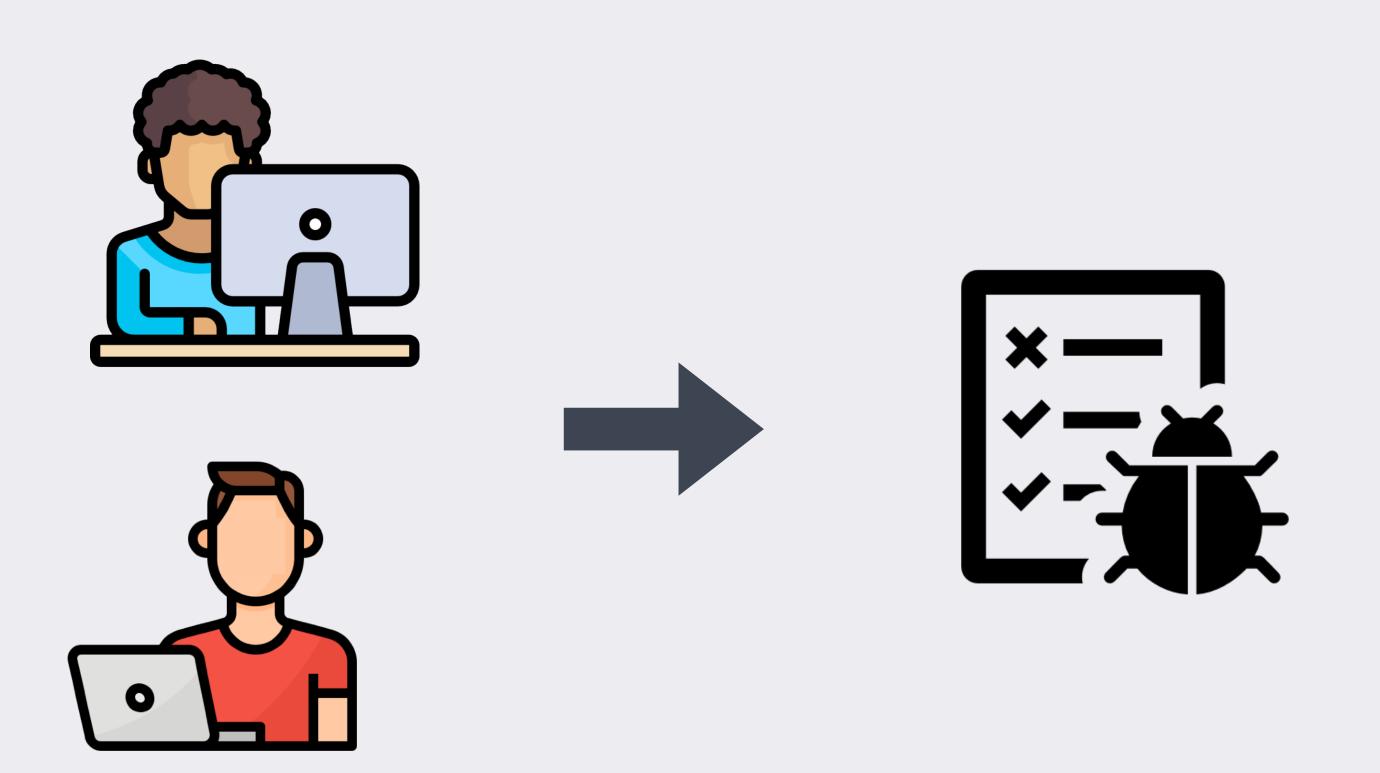
REPORTING SOFTWARE BUGS





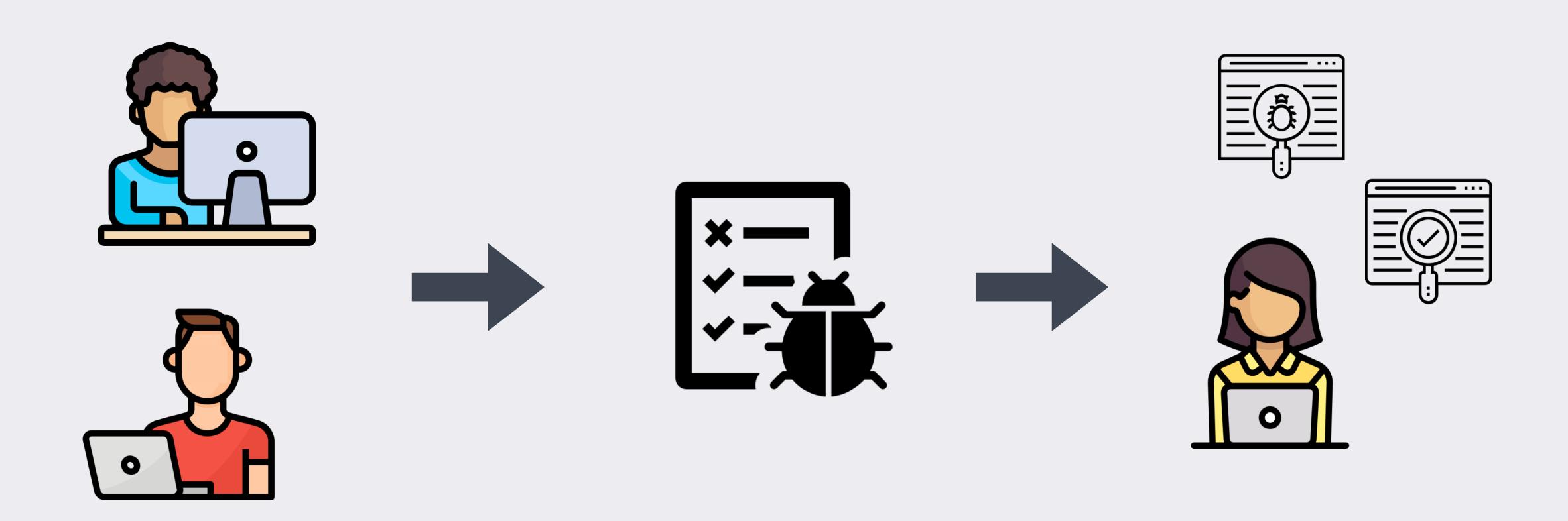
Users and developers report bugs when software does not behave as expected

REPORTING SOFTWARE BUGS



Users and developers report bugs when software does not behave as expected

REPORTING SOFTWARE BUGS



Users and developers report bugs when software does not behave as expected

COMPONENTS OF A BUG REPORT

MetaData

Bug 101434 - [Contexts] performance: Slow cursor navigation in Text fields

Status: VERIFIED FIXED Reported: 2005-06-23 09:08 EDT by Tom Hofmann

Alias: None Modified: 2005-09-26 13:45 EDT (History)

CC List: 6 users (show)

Product: Platform

Component: UI (show other bugs) See Also:
Version: 3.1

Hardware: PC Linux-GTK

Importance: P2 major (vote)

<u>Target</u> 3.1.1

Assignee: Douglas Pollock (

Attachments		
updateShellKludge.zip (5.07 KB, application/x-zip-compressed) 2005-06-23 10:30 EDT, Tom Hofmann ECA	no flags	<u>Details</u>
Patch to "org.eclipse.ui.workbench" (3.96 KB, patch)	no flags	Details Diff

COMPONENTS OF A BUG REPORT

Attachments

Attachments		
updateShellKludge.zip (5.07 KB, application/x-zip-compressed) 2005-06-23 10:30 EDT, Tom Hofmann	no flags	<u>Details</u>
Patch to "org.eclipse.ui.workbench" (3.96 KB, patch) 2005-06-23 11:44 EDT, Douglas Pollock ECA		Details Diff
Add an attachment (proposed patch, testcase, etc.) Show	w Obsolet	e (1) View All

— Note -

You need to log in before you can comment on or make changes to this bug.

Tom Hofmann 2005-06-23 09:08:52 EDT <u>Description</u>

N20050623-0010-gtk

Not sure whether this is my particular install or a new problem.

With the above build, cursor navigation in any text field is extremely slow.

Steps:

- open the Find dialog (alternatively: open the preference dialog, or the Team>Create Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / _RIGHT keys

COMPONENTS OF A BUG REPORT

Description

Tom Hofmann (



2005-06-23 09:08:52 EDT

Description

N20050623-0010-gtk

Not sure whether this is my particular install or a new problem.

With the above build, cursor navigation in any text field is extremely slow.

Steps:

- open the Find dialog (alternatively: open the preference dialog, or the Team>Create Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / _RIGHT keys

Expected: I see the caret move through the entered text Actual: The cursor does not visibly change its position until after releasing the key plug some delay.

Notes:

- keyboard repeat rate is at the default
- the same works fine (+/- at least nowhere near the slowness of Text fields) in text editors
- other applications do not show this behavior

This is quite bad as the caret cannot be placed reliably this way. Will try to reproduce with an SWT-only example.

STEPS TO REPRODUCE (S2RS)

Description

Tom Hofmann 2005-06-23 09:08:52 EDT

Description

N20050623-0010-gtk

Not sure whether this is my particular install or a new problem.

With the above build, cursor navigation in any text field is extremely slow.

Steps:

- open the Find dialog (alternatively: open the preference dialog, or the Team>Create Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / _RIGHT keys

Expected: I see the caret move through the entered text Actual: The cursor does not visibly change its position until after releasing the key plug some delay.

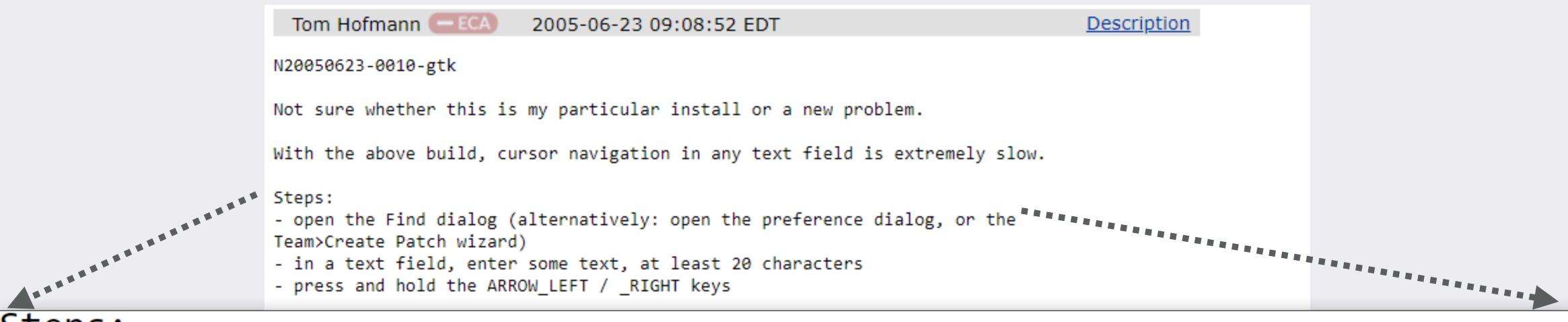
Notes:

- keyboard repeat rate is at the default
- the same works fine (+/- at least nowhere near the slowness of Text fields) in text editors
- other applications do not show this behavior

This is quite bad as the caret cannot be placed reliably this way. Will try to reproduce with an SWT-only example.

STEPS TO REPRODUCE (S2RS)

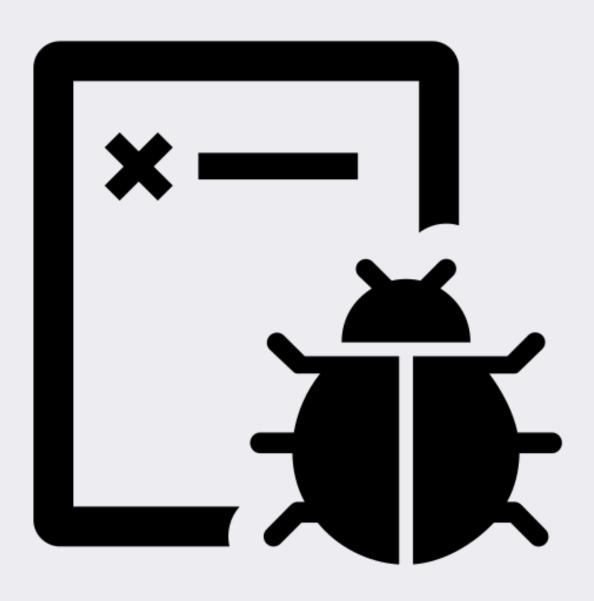
Description



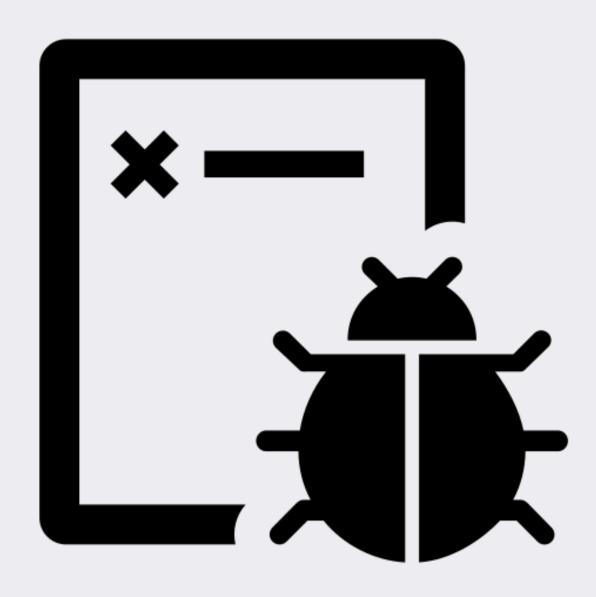
Steps:

- open the Find dialog (alternatively: open the preference dialog, or the Team>Create Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / _RIGHT keys

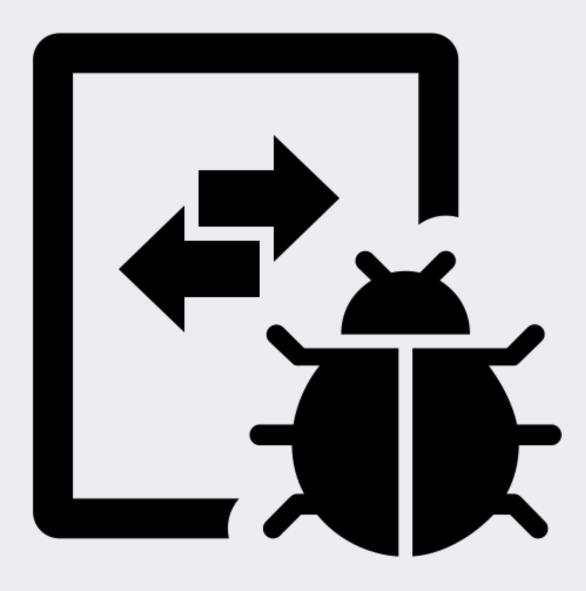
This is quite bad as the caret cannot be placed reliably this way. Will try to reproduce with an SWT-only example.



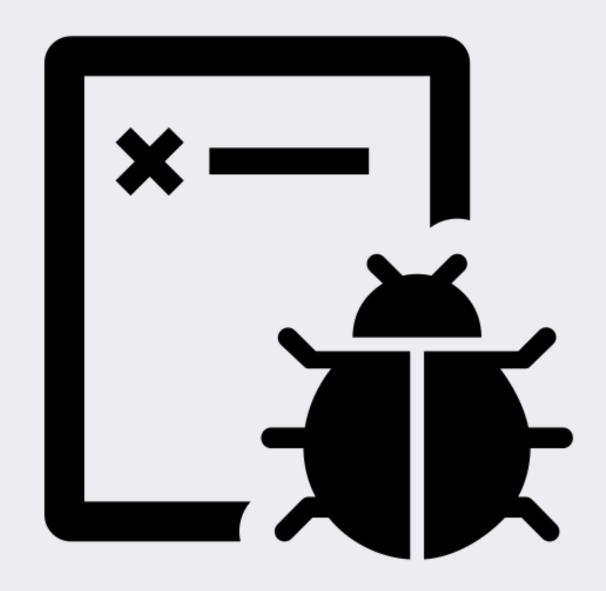
Missing



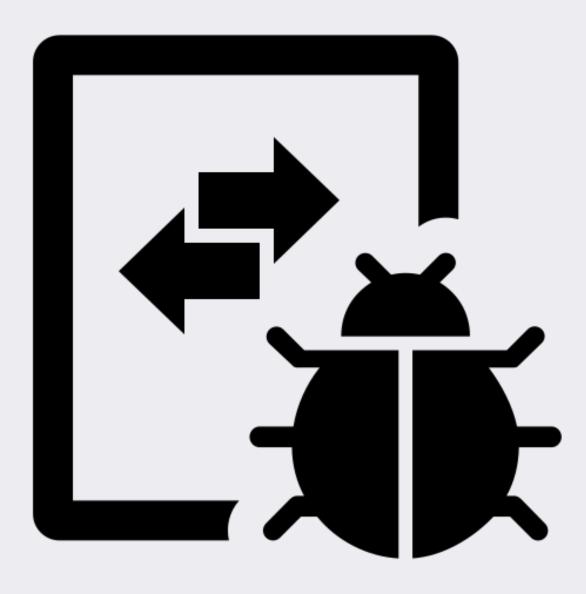
Missing



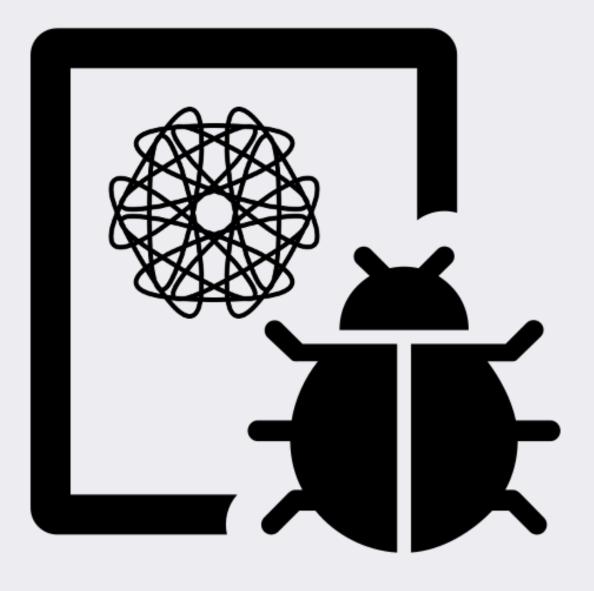
Ambiguous



Missing



Ambiguous



Incomplete

THE GITHUB ISSUE PETITION

Dear GitHub,

You have done so much to grow the open source community and make it really accessible to users. Somehow you have us chasing stars and filling up squares, improving the world's software in the process.

However, many of us are frustrated. Those of us who run some of the most popular projects on GitHub feel completely ignored by you. We've gone through the only support channel that you have given us either to receive an empty response or even no response at all. We have no visibility into what has happened with our requests, or whether GitHub is working on them. Since our own work is usually done in the open and everyone has input into the process, it seems strange for us to be in the dark about one of our most important project dependencies.

The problems we most frequently have, and our best ideas for how to address them, are:

- Issues are often filed missing crucial information like reproduction steps or version tested. We'd like issues to gain custom fields, along with a mechanism (such as a mandatory issue template, perhaps powered by a newissue.md in root as a likely-simple solution) for ensuring they are filled out in every issue.
- Iccurs often accumulate content loss " 1" comments which carve only to snow the maintainers and any others

THE GITHUB ISSUE PETITION

The problems we most frequently have, and our best ideas for how to address them, are:

- Issues are often filed missing crucial information like reproduction steps or version tested. We'd like issues to gain custom fields, along with a mechanism (such as a mandatory issue template, perhaps powered by a newissue.md in root as a likely-simple solution) for ensuring they are filled out in every issue.
- Issues often accumulate content-less "+1" comments which serve only to spam the maintainers and any others subscribed to the issue. These +1s serve a valuable function in letting maintainers know how widespread an issue is, but their drawbacks are too great. We'd like issues to gain a first-class voting system, and for content-less comments like "+1" or "♣" or "me too" to trigger a warning and instructions on how to use the voting mechanism.
- Issues and pull requests are often created without any adherence to the CONTRIBUTING.md contribution guidelines, due to the inconspicuous nature of the "guidelines for contributing" link when creating an issue and the fact that it often contains a lot of information that isn't relevant to opening issues (such as information about hacking on the project). Maintainers should be able to configure a file in the repo (interpreted as GFM) to be displayed at the top of the new issue / PR page instead of that link. Maintainers can choose to inline content there and / or link to other pages as appropriate.

THE GITHUB ISSUE PETITION

The problems we most frequently have, and our best ideas for how to address them, are:

Issues are often filed missing crucial information like reproduction steps or version tested. We'd like issues to gain custom fields, along with a mechanism (such as a mandatory issue template, perhaps powered by a newissue.md in root as a likely-simple solution) for ensuring they are filled out in every issue.

Issues are often filed missing crucial information like reproduction steps

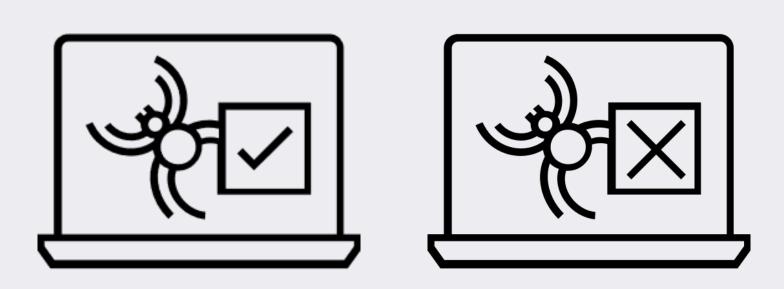
issue is, but their drawbacks are too great. We'd like issues to gain a first-class voting system, and for content-less comments like "+1" or "

ightharpoonup of the system of the syst

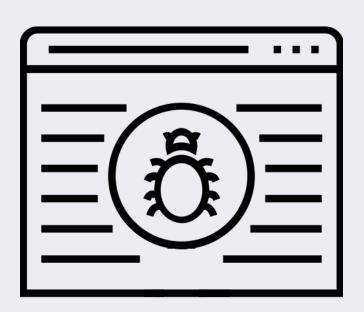
• Issues and pull requests are often created without any adherence to the CONTRIBUTING.md contribution guidelines, due to the inconspicuous nature of the "guidelines for contributing" link when creating an issue and the fact that it often contains a lot of information that isn't relevant to opening issues (such as information about hacking on the project). Maintainers should be able to configure a file in the repo (interpreted as GFM) to be displayed at the top of the new issue / PR page instead of that link. Maintainers can choose to inline content there and / or link to other pages as appropriate.

THE IMPACT OF LOW-QUALITY S2Rs

THE IMPACT OF LOW-QUALITY S2Rs

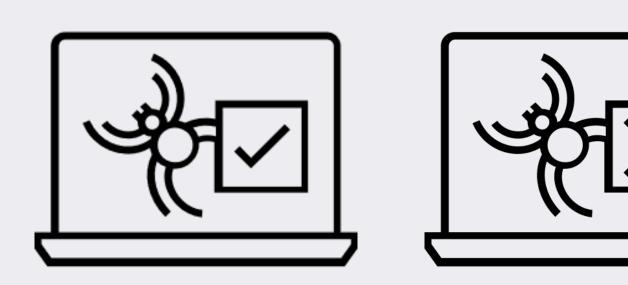


Non-Reproducible Bugs
[Erfani Joorbachi et.al., MSR'14]



Unfixed Bugs
[Zimmerman et.al., ICSE'12]

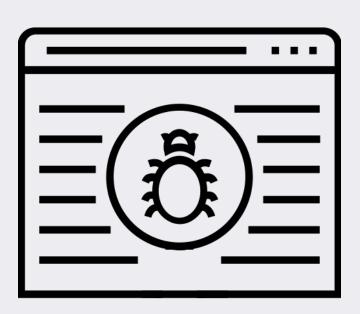
THE IMPACT OF LOW-QUALITY S2Rs



Non-Reproducible Bugs
[Erfani Joorbachi et.al., MSR'14]



Additional Bug Triage
[Breu, et.al., CSCW'10
& Zimmerman, et. al., ICSE'12]

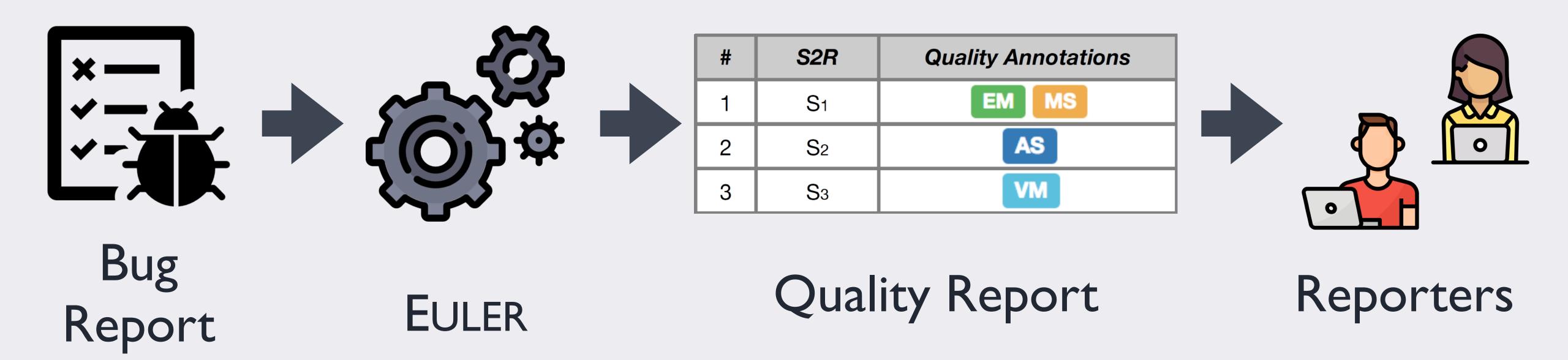


Unfixed Bugs
[Zimmerman et.al., ICSE'12]



Delay in Bug Resolution
[Guo, et. al., ICSE'10
& Zimmerman, et. al., ICSE'12]

EULER: ASSESSING THE QUALITY OF S2RS



EULER: ASSESSING THE QUALITY OF S2Rs



Euler is able to:

- 1) Automatically identify S2Rs
- 2) Associate the S2Rs with program information
- 3) Asses the quality of given S2Rs

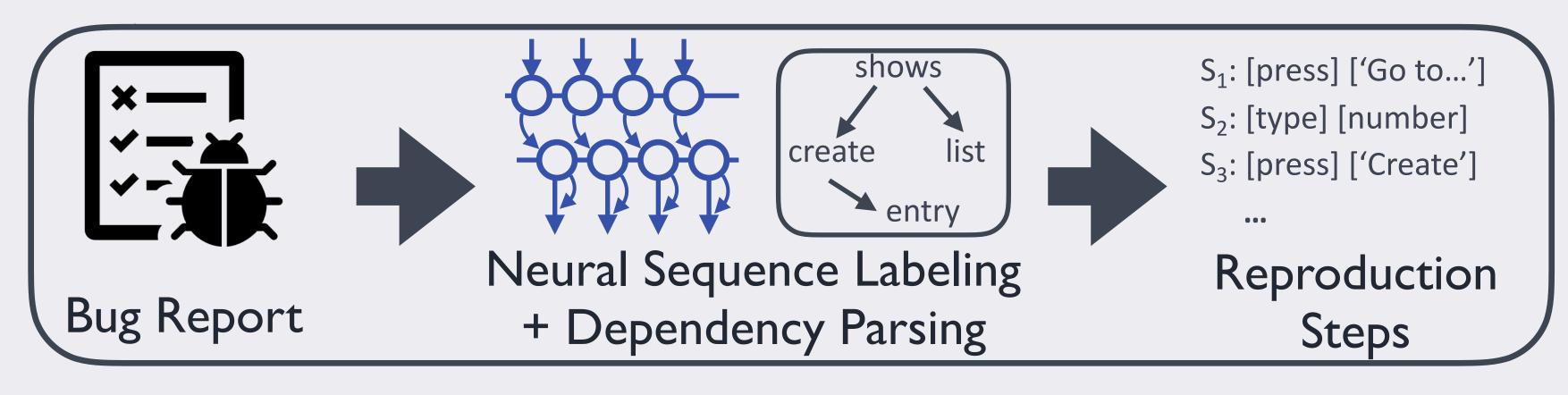
EULER: ASSESSING THE QUALITY OF S2Rs



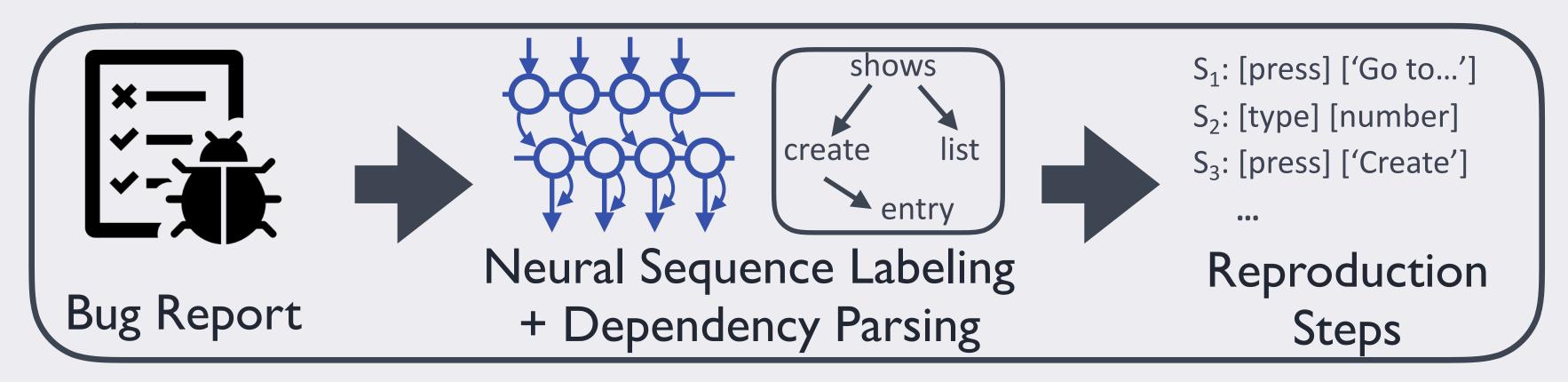
Euler is able to detect:

- 1) Missing Steps
- 2) Ambiguous Steps
- 3) Steps with Unexpected Vocabulary

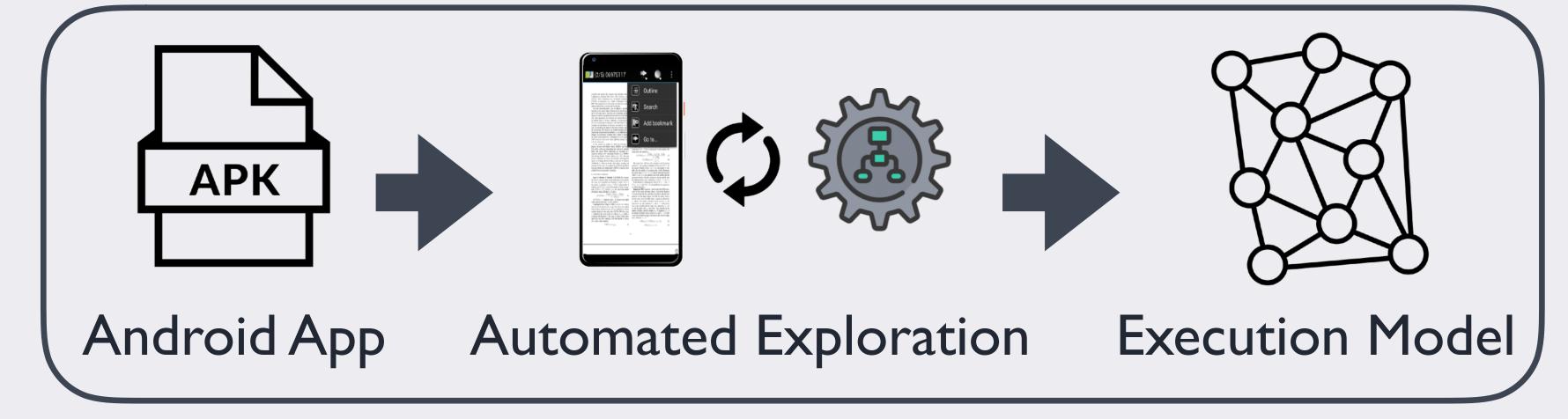
1) S2R Identification



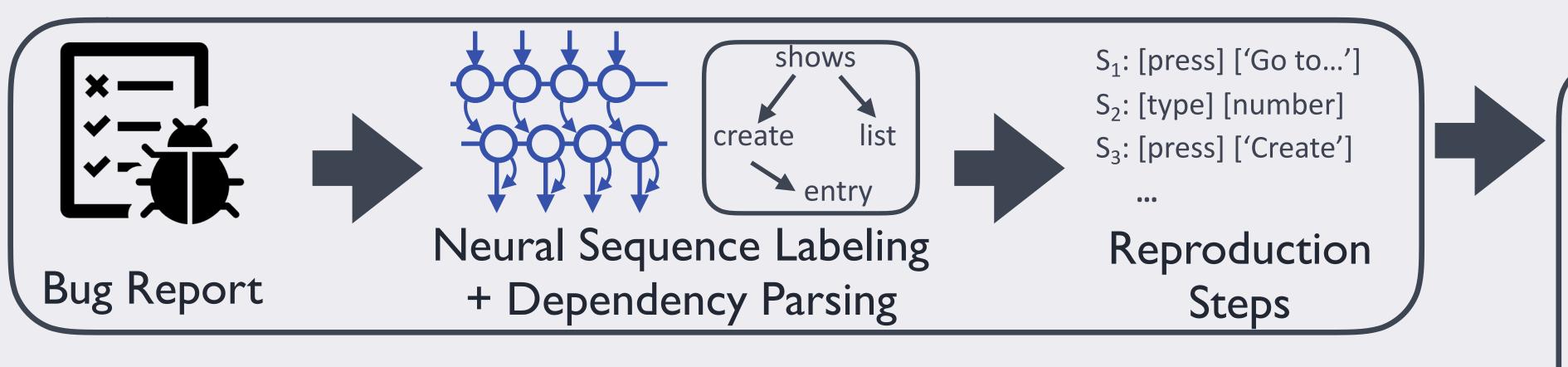
1) S2R Identification



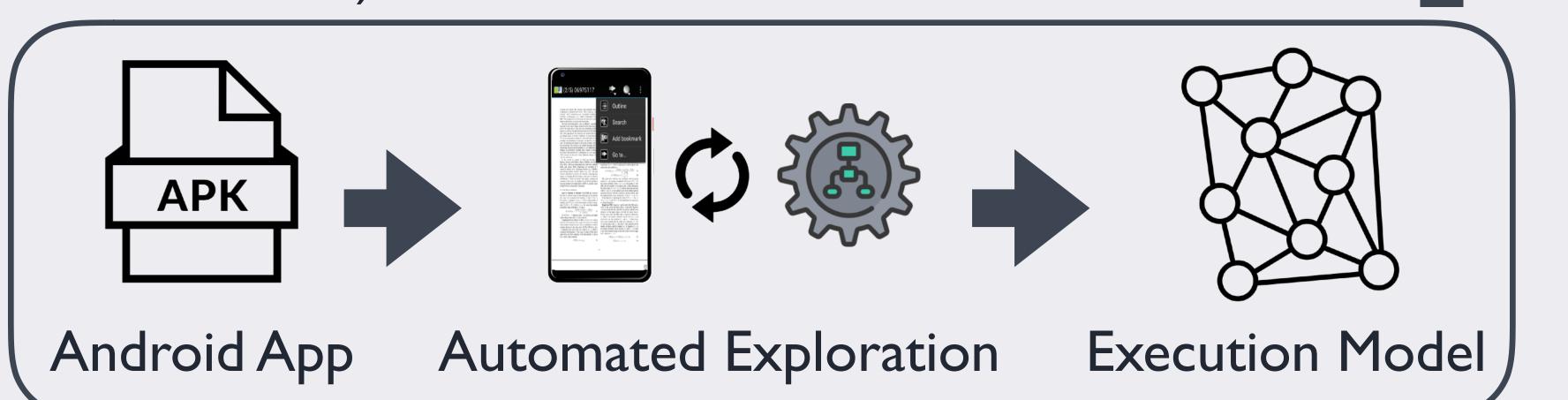
2) Execution Model Generation







2) Execution Model Generation

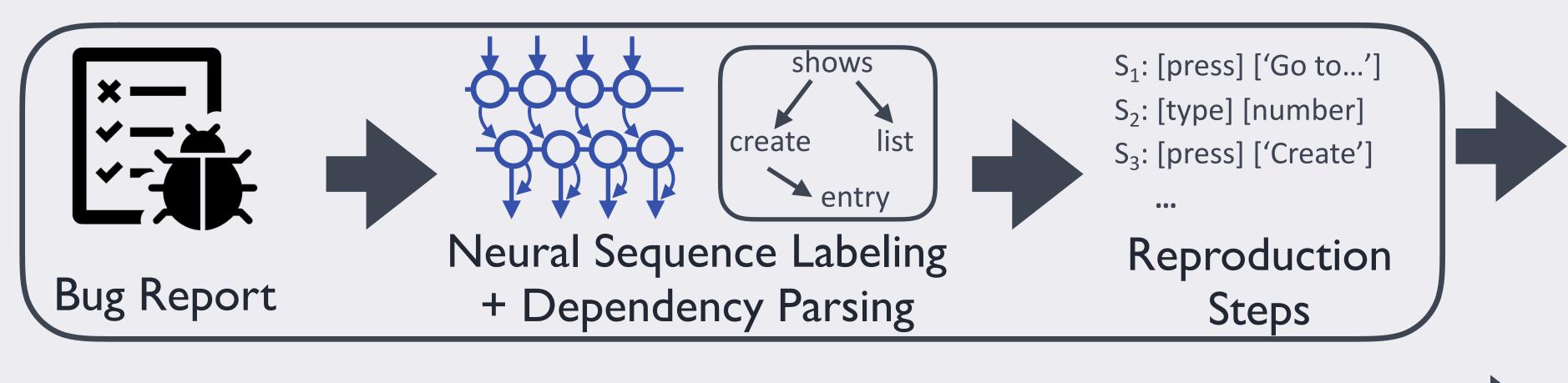


3) Quality
Assessment

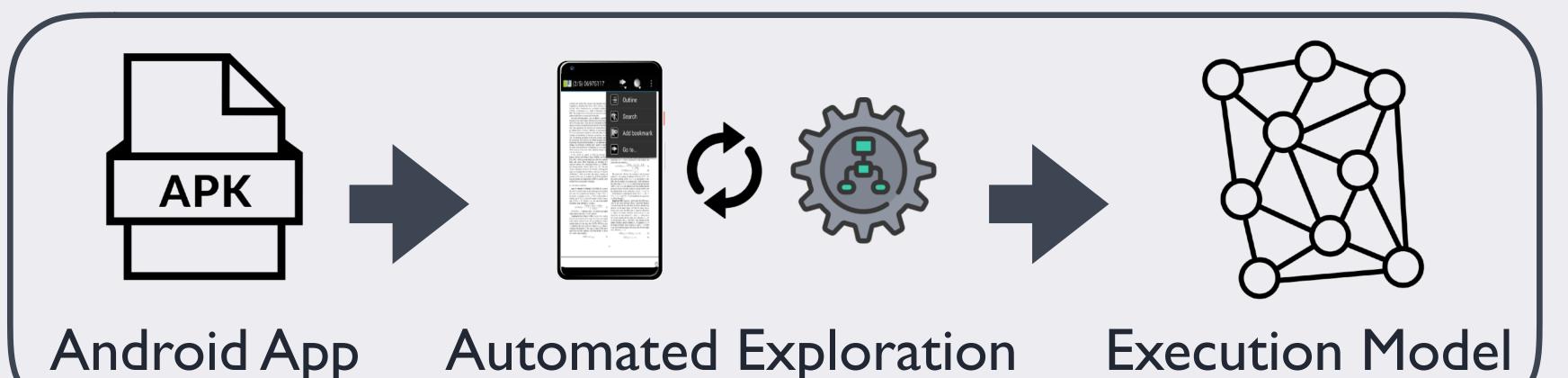
S2R Matching

S2R Inference





2) Execution Model Generation



3) Quality
Assessment

S2R Matching

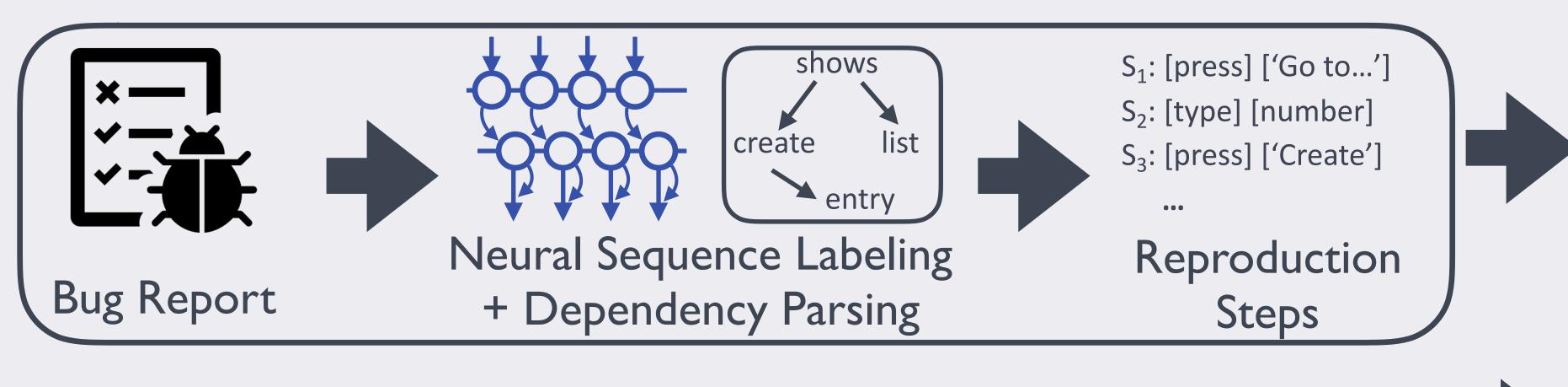
S2R Inference





#	S2R	Quality Annotations
1	S ₁	EM MS
2	S ₂	AS
3	S ₃	VM





2) Execution Model Generation



3) Quality
Assessment

S2R Matching

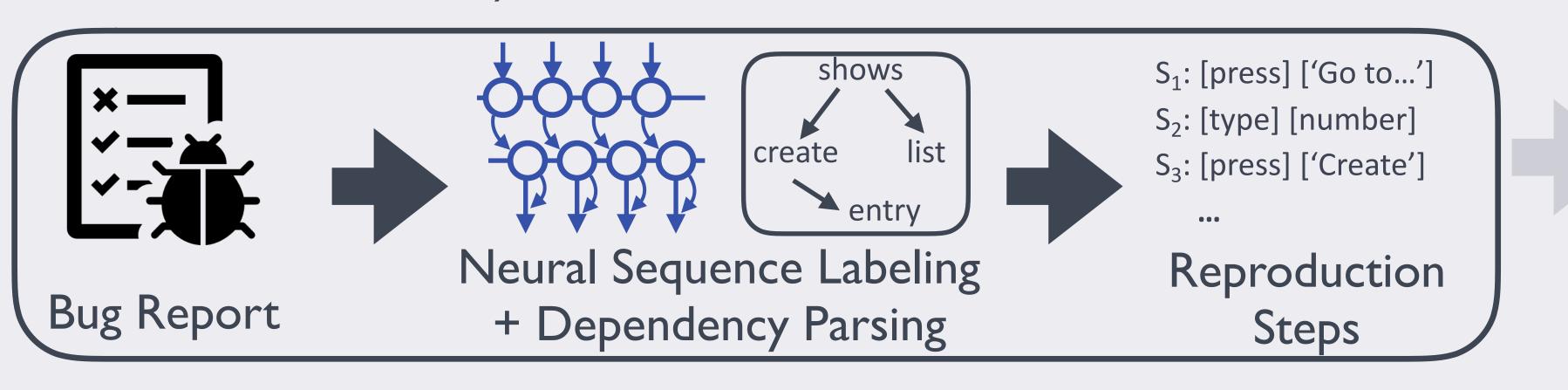
S2R Inference



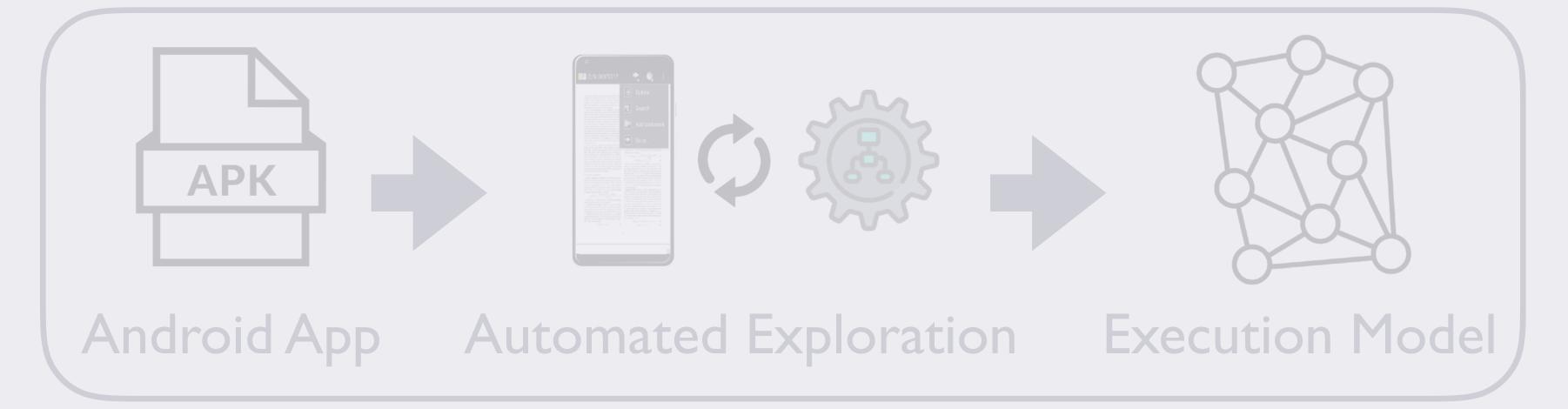


#	S2R	Quality Annotations
1	S ₁	EM MS
2	S ₂	AS
3	S ₃	VM

1) S2R Identification



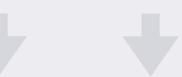
2) Execution Model Generation



3) Quality
Assessment

SZK Matching

nference



		Quality Annotations
1	S ₁	
2	S ₂	
3	S ₃	

S2R IDENTIFICATION IN BUG REPORTS

Bug Report

• • •

Usually when I press 'Go to...' and type a number and press 'Go', it simply refreshes ...

S2R Sentences

Usually when I press 'Go to....' and type a number and press 'Go', it simply refreshes ...

Individual S2R

 S_1 : [pres

S₁: [press] ['Go to...']

S₂: [type] [number]

S₃: [press] ['Go']

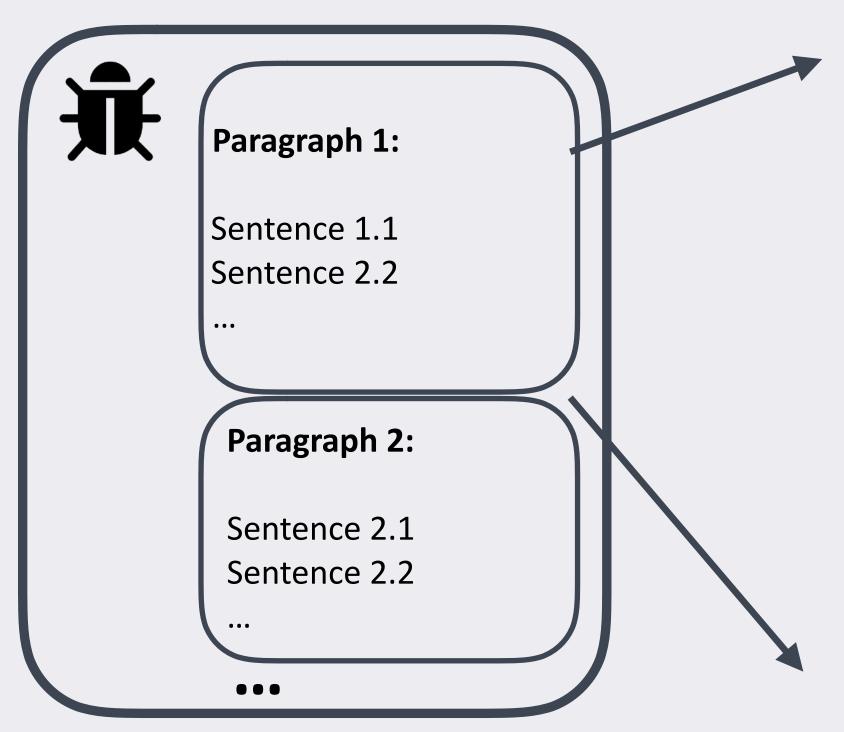
S2R IDENTIFICATION IN BUG REPORTS

Bug Report S2R Sentences Individual S2R Usually when I press 'Go to...' and type a number and press 'Go', it simply refreshes ... Usually when I press 'Go to...' and type a number and press 'Go', it simply refreshes ... S1: [press] ['Go to...'] S2: [type] [number] S3: [press] ['Go']

Approach: neural sequence labeling + dependency parsing

MODEL INPUT + DATASET LABELS

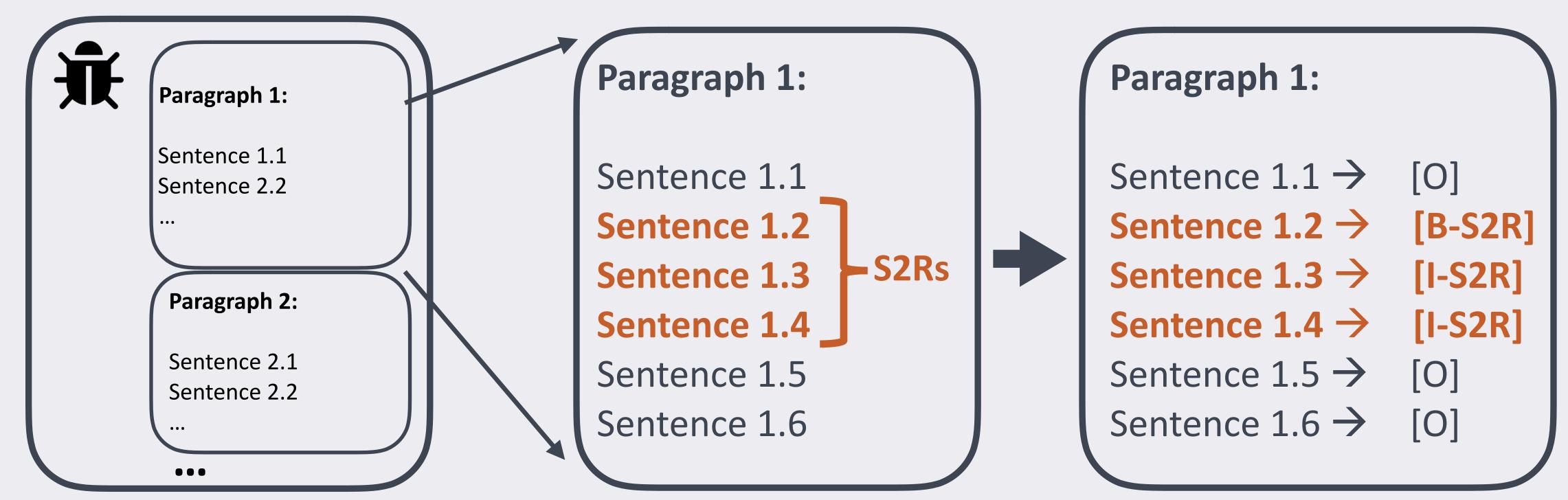
Bug Report: Paragraphs, Sentences, Words



MODEL INPUT + DATASET LABELS

Bug Report:
Paragraphs,
Sentences, Words

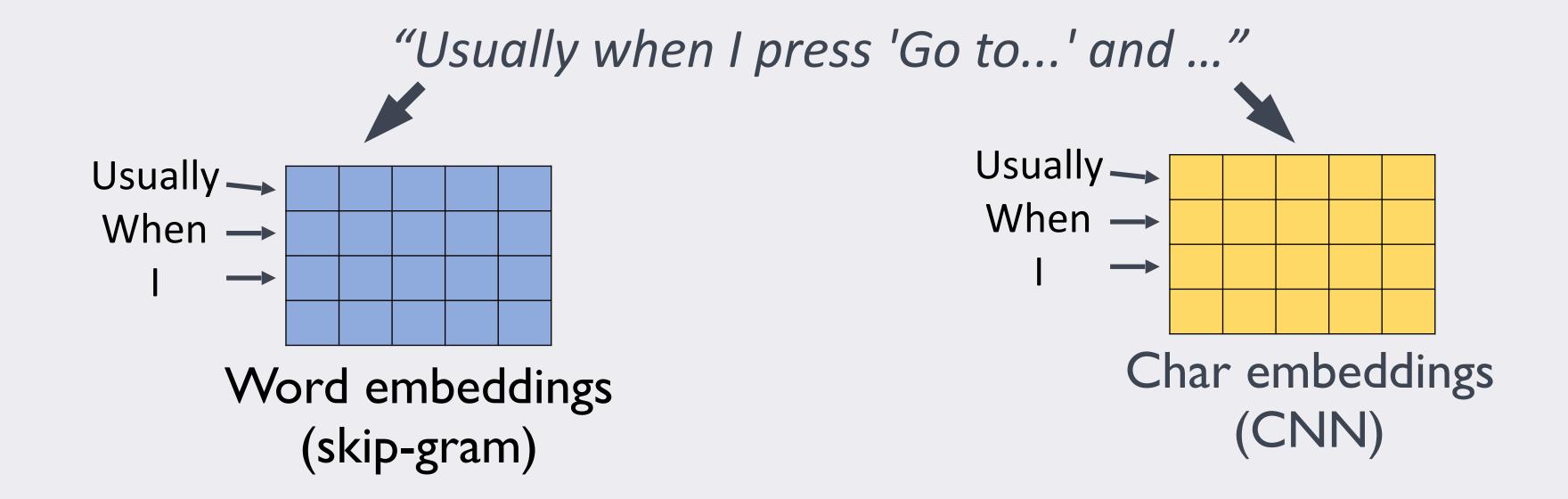
Beginning-Inside-Outside (BIO) tagging



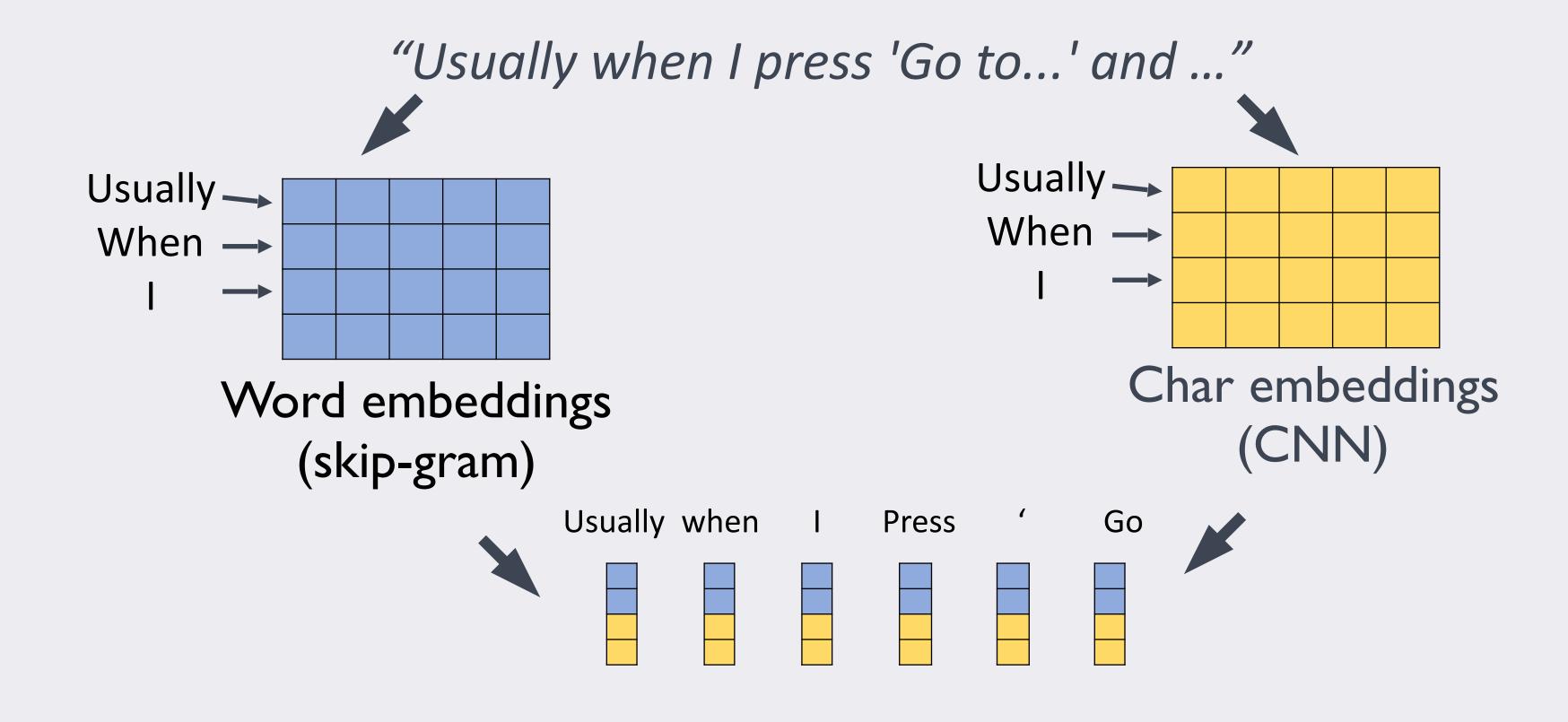
WORD REPRESENTATIONS

"Usually when I press 'Go to...' and ..."

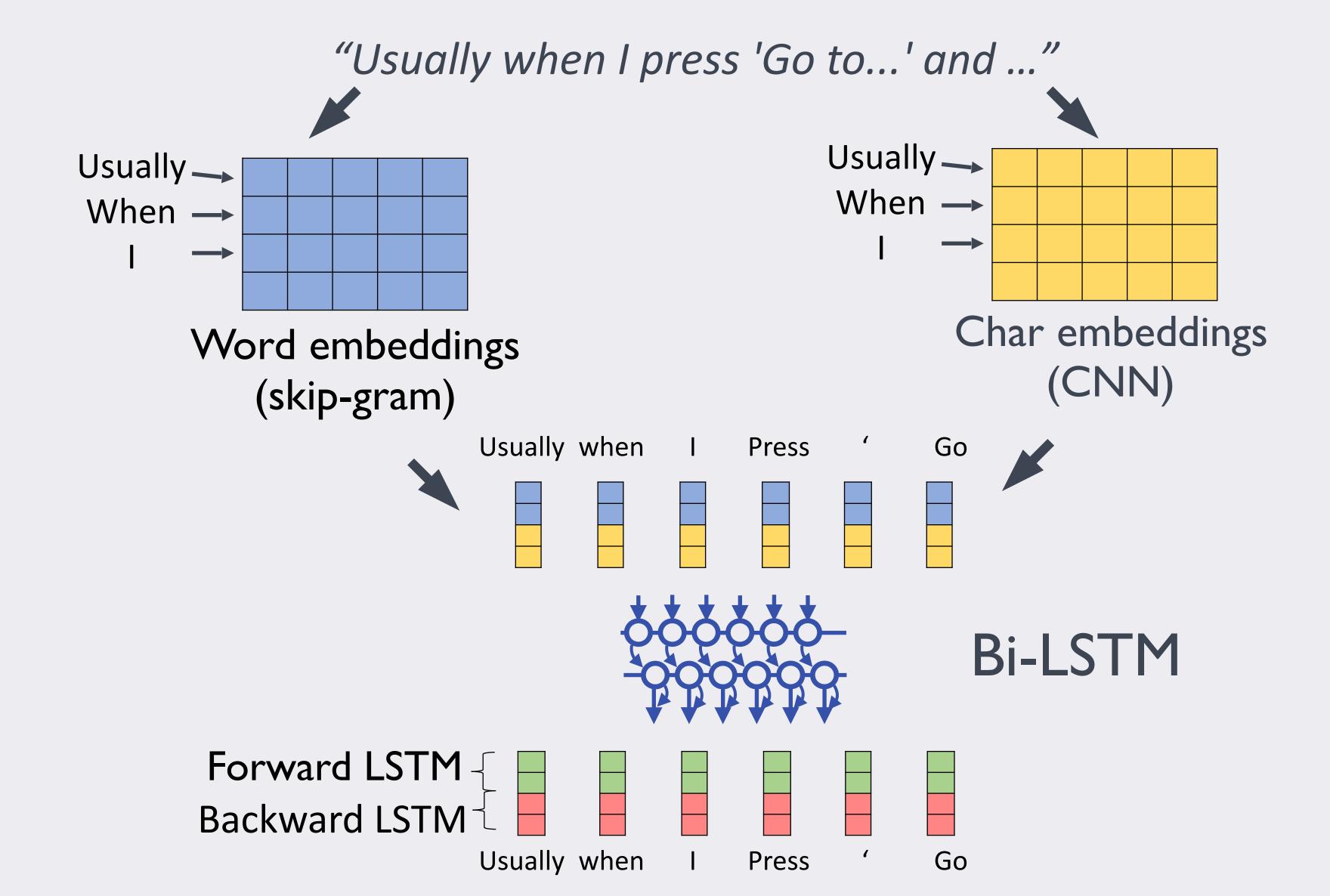
WORD REPRESENTATIONS



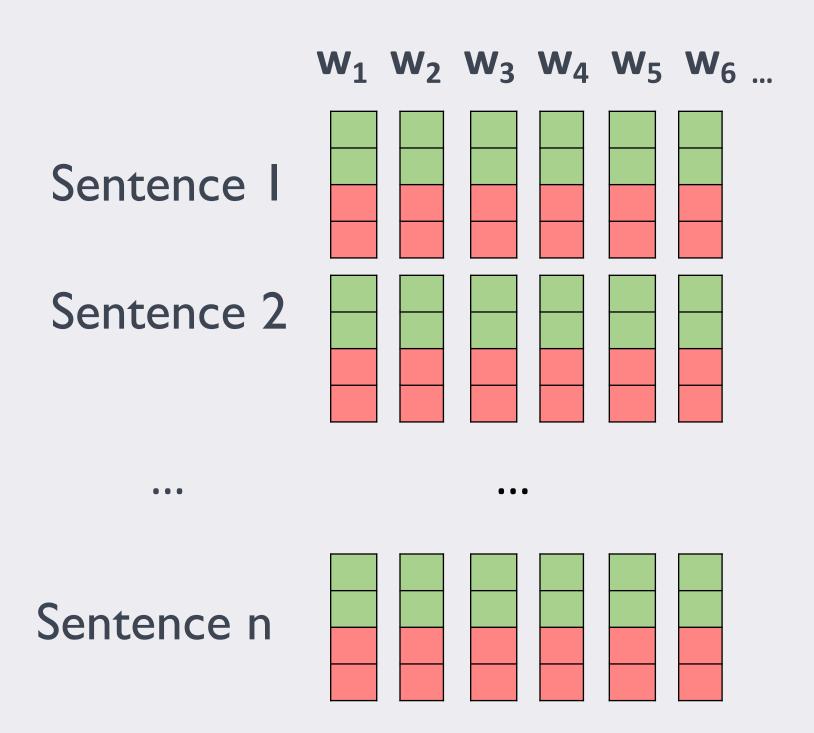
WORD REPRESENTATIONS



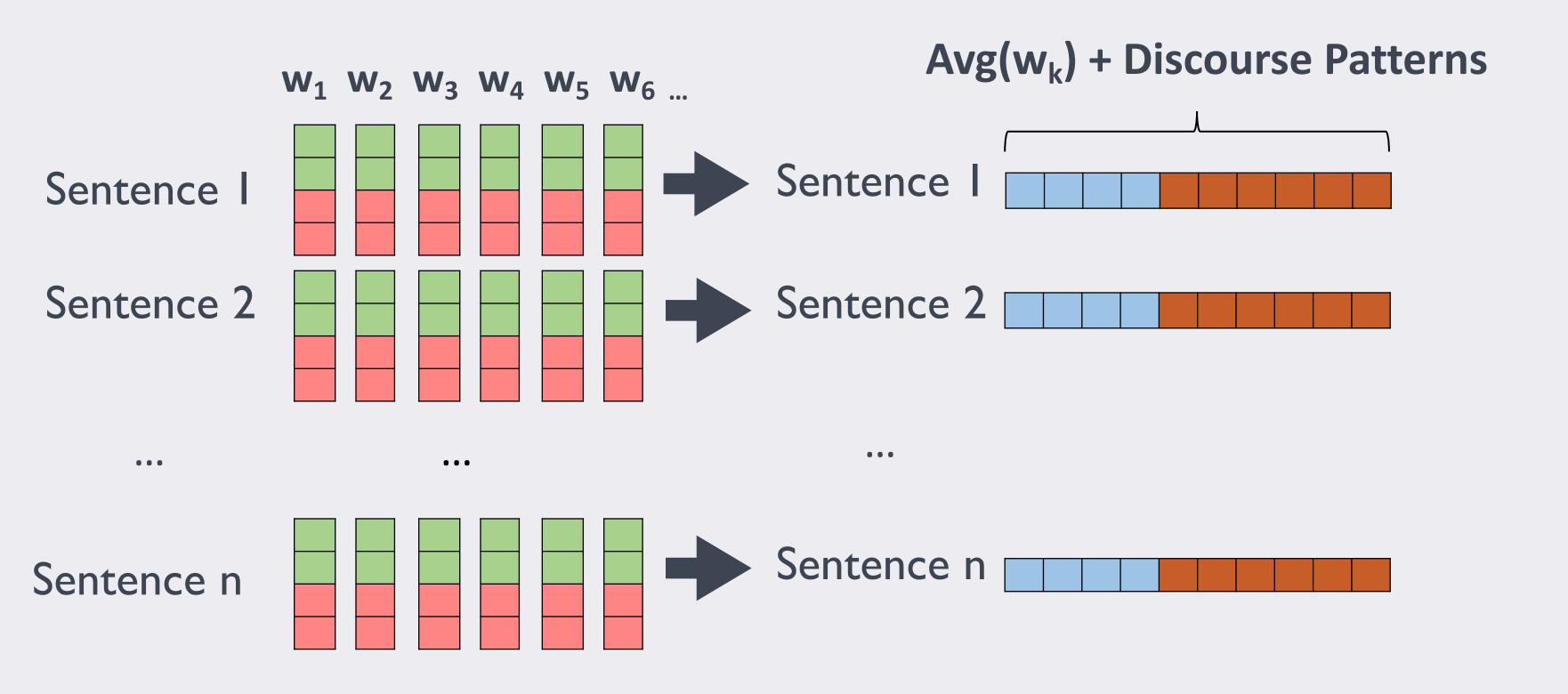
WORD REPRESENTATIONS



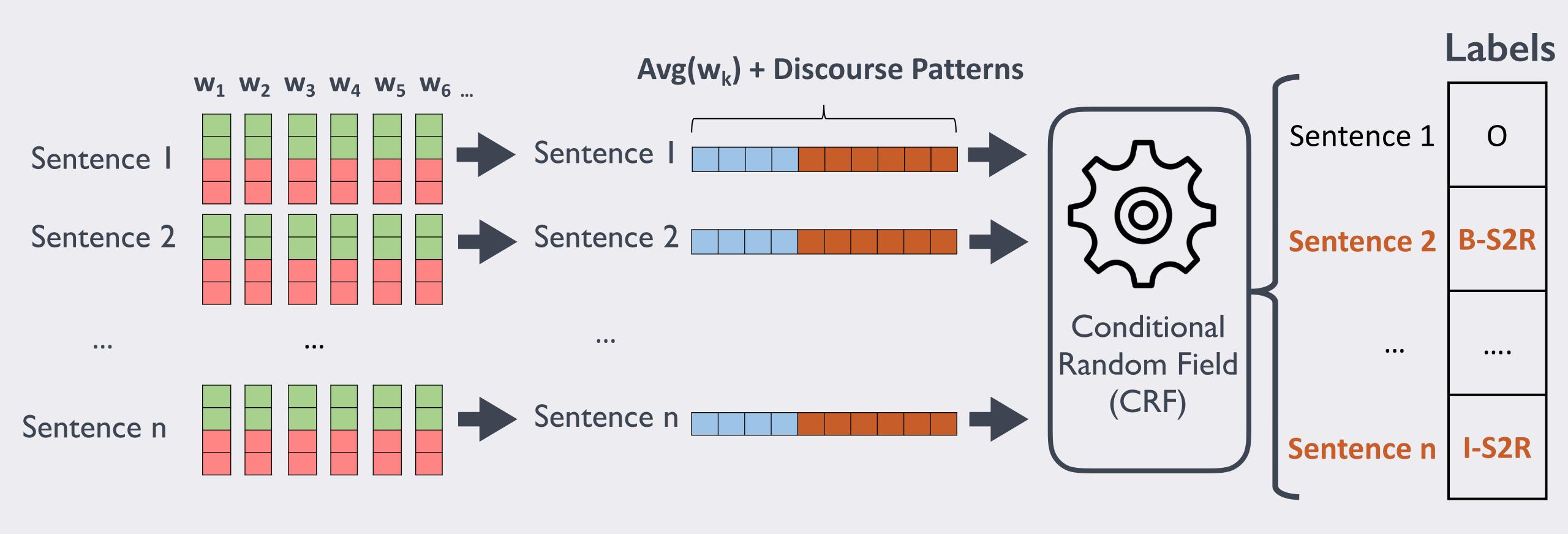
SENTENCE REPRESENTATION & INFERENCE



SENTENCE REPRESENTATION & INFERENCE



SENTENCE REPRESENTATION & INFERENCE



S2R FORMAT

S2R: [action] [object] [preposition] [object2]

User operation Direct entity Relationship Related entity

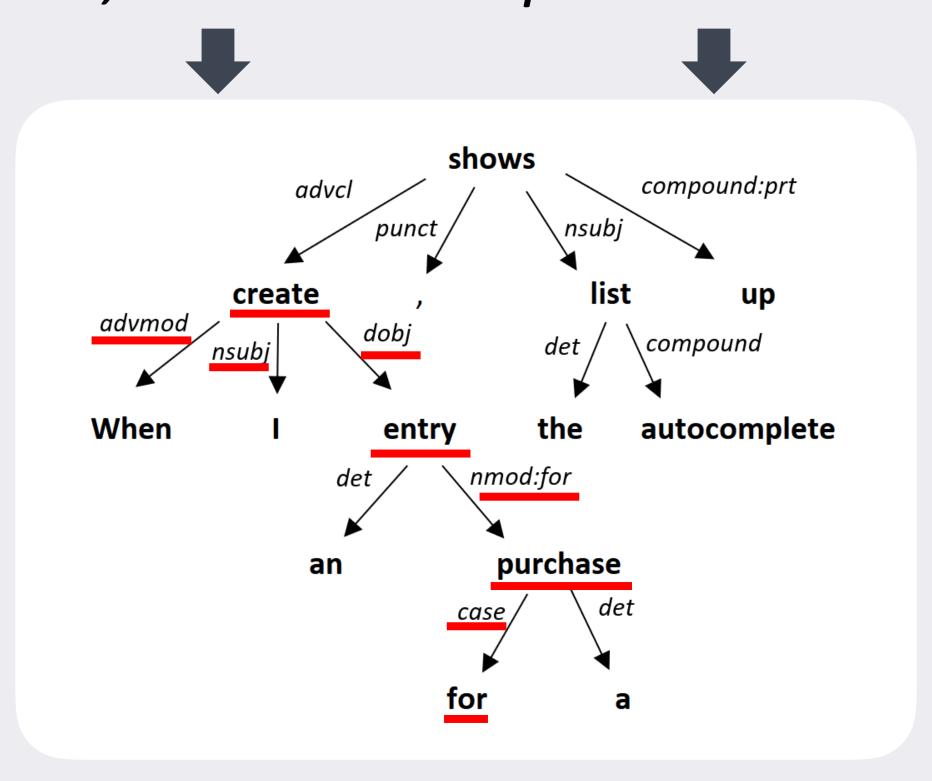
"When I **create** an **entry for** a **purchase**, the autocomplete list shows up"



S2R: [create] [entry] [for] [purchase]

DEPENDENCY PARSING

"When I **create** an **entry for** a **purchase**, the autocomplete list shows up"



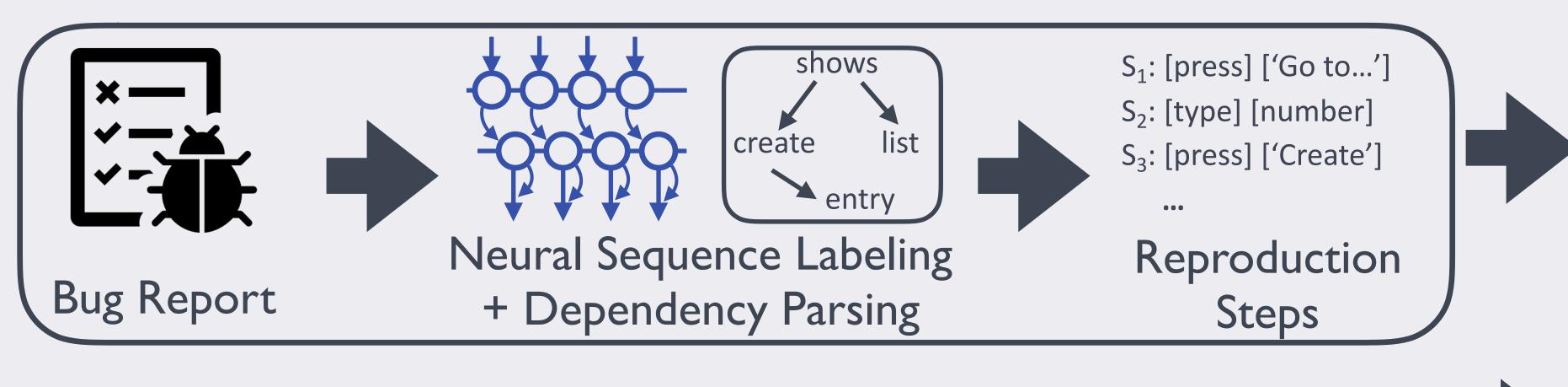




S2R: [create] [entry] [for] [purchase]

EULER'S COMPONENTS & WORKFLOW





2) Execution Model Generation



3) Quality
Assessment

S2R Matching

S2R Inference

Random Exploration

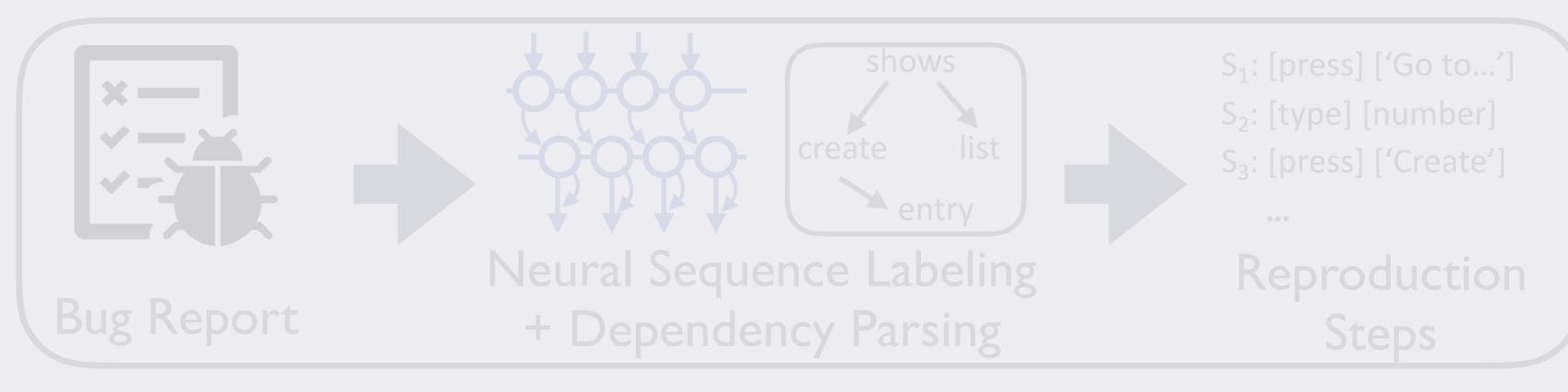




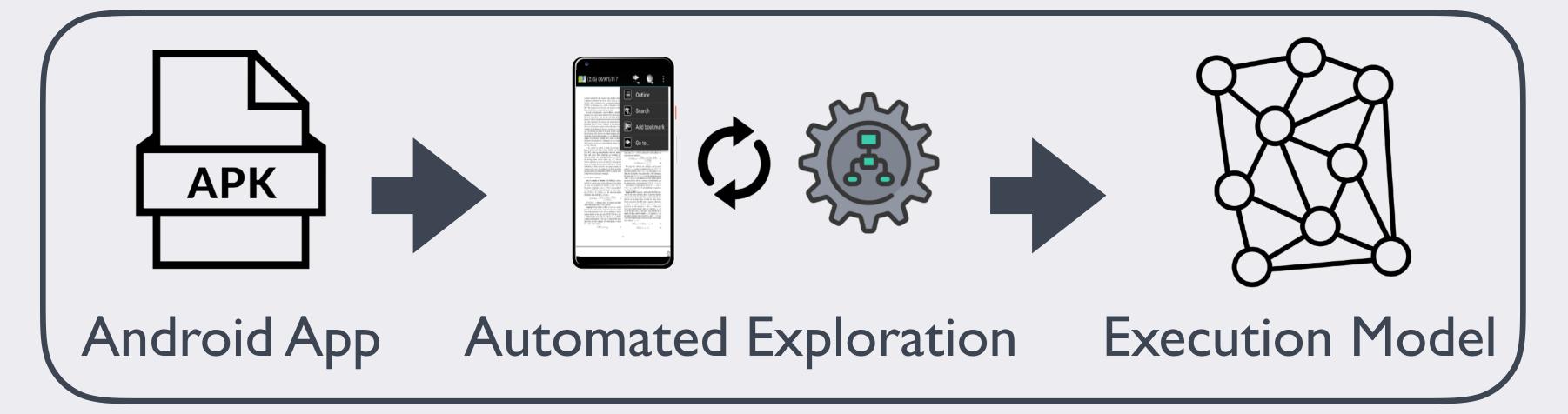
#	S2R	Quality Annotations
1	S ₁	EM MS
2	S ₂	AS
3	S ₃	VM

EULER'S COMPONENTS & WORKFLOW





2) Execution Model Generation



3) Quality
Assessment

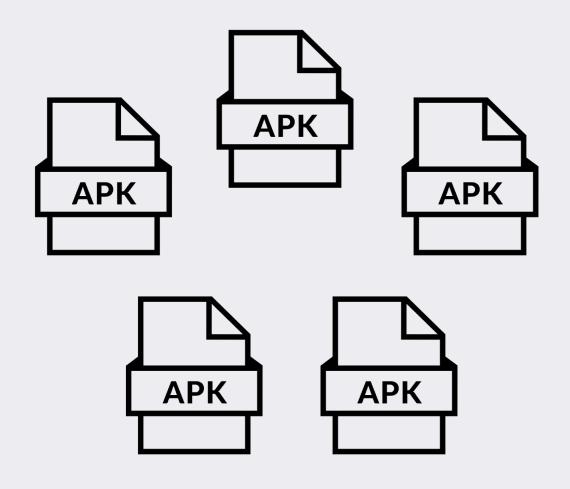
SZK Matching

32K nference

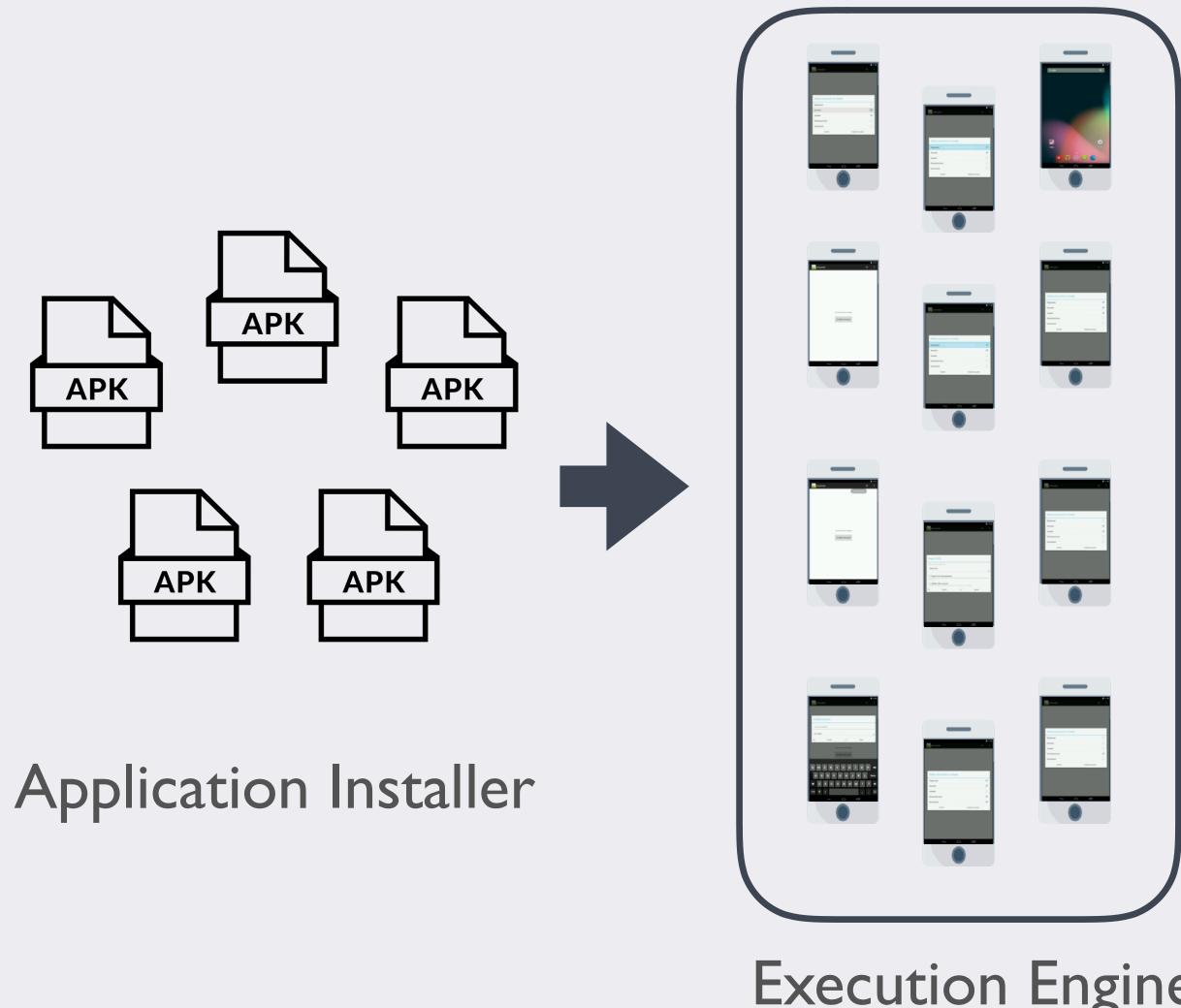
Random Exploration



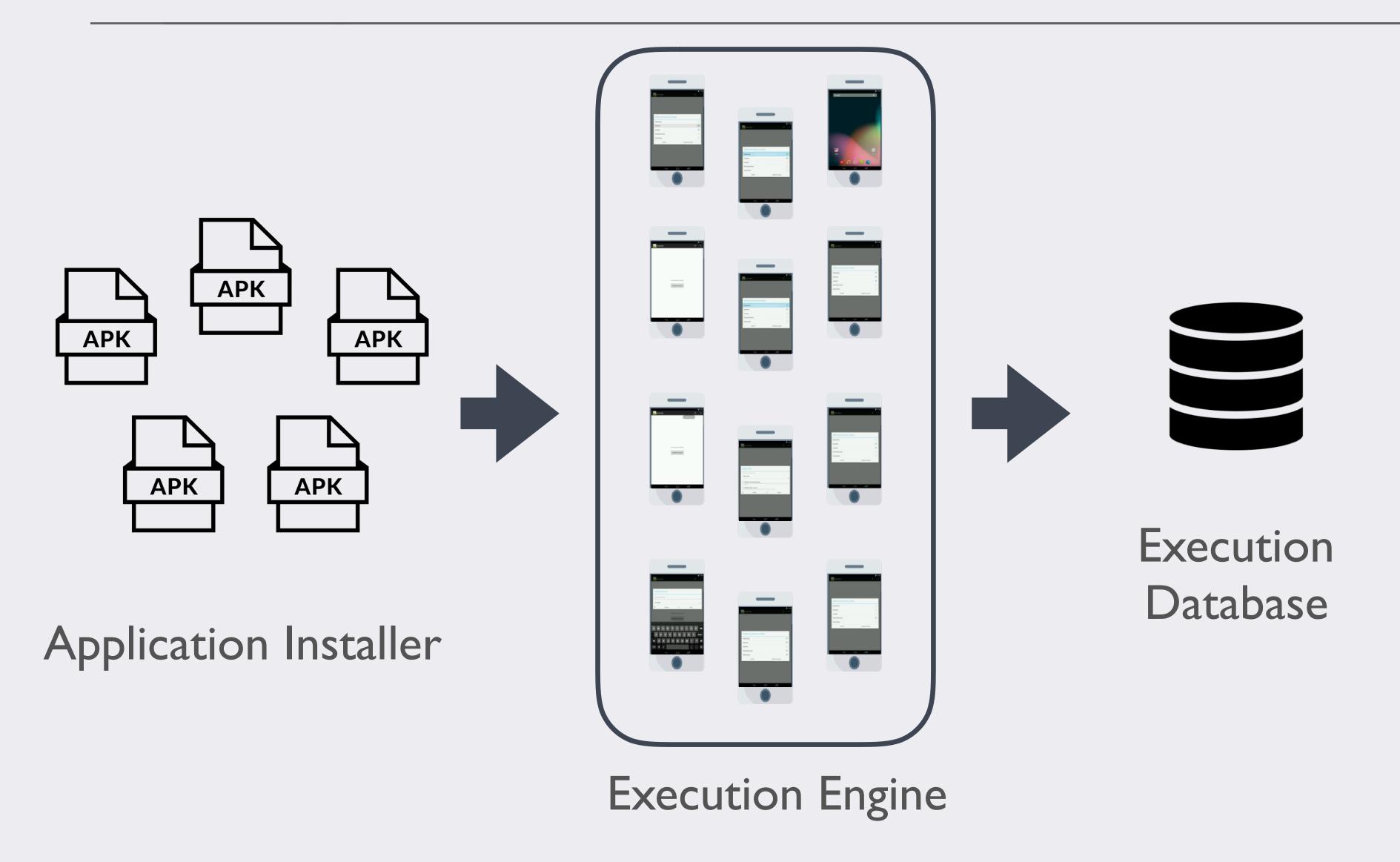
		Quality Annotations
1	S ₁	
2	S ₂	
3	S ₃	

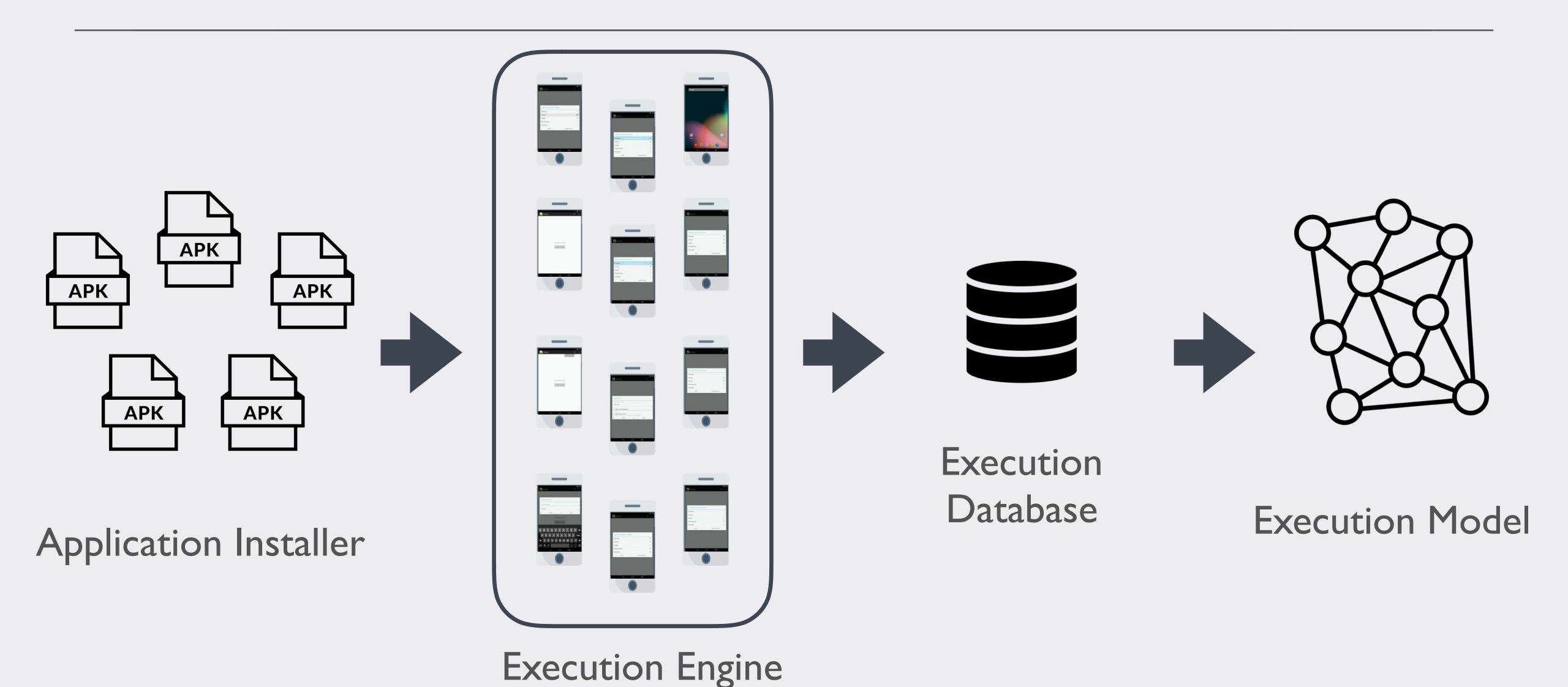


Application Installer



Execution Engine

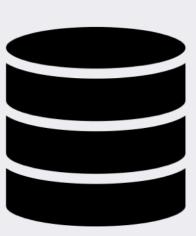


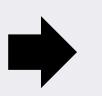


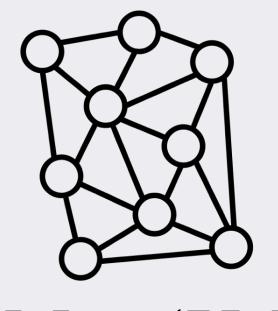
EXECUTION DATABASE TRANSFORMATION

Execution database







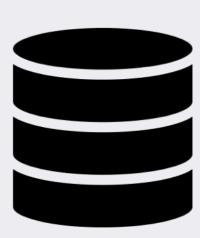


$$EM = (V, E)$$

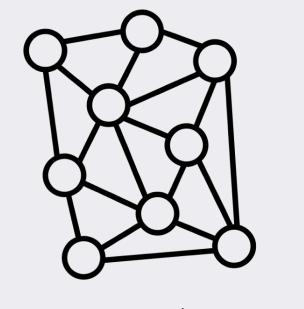
EXECUTION DATABASE TRANSFORMATION

Execution database









$$EM = (V, E)$$

V: App screens

$$v \in V = (GC, H)$$

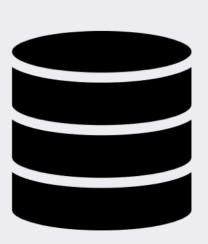
GC: Set of GUI-components

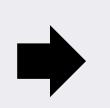
H: Component hierarchy

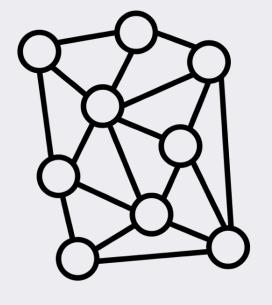
EXECUTION DATABASE TRANSFORMATION

Execution database

Execution Model (EM)







$$EM = (V, E)$$

V: App screens

$$v \in V = (GC, H)$$

GC: Set of GUI-components

H: Component hierarchy

E: App interactions

$$e \in E = (v_x, e, c, v_y)$$

 U_{χ} : Source screen

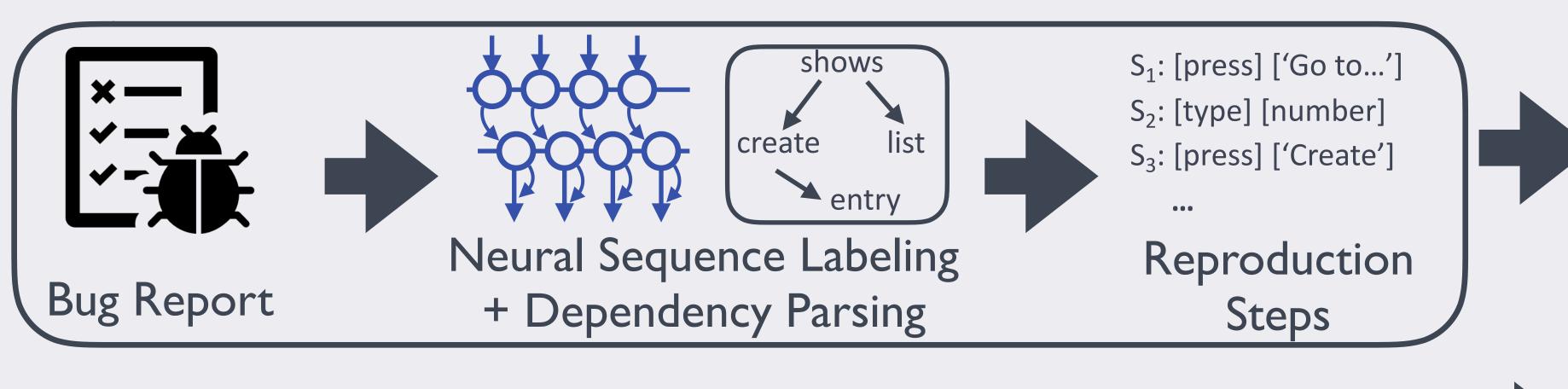
e: Event (tap, swipe, etc.)

c: Component from v_x

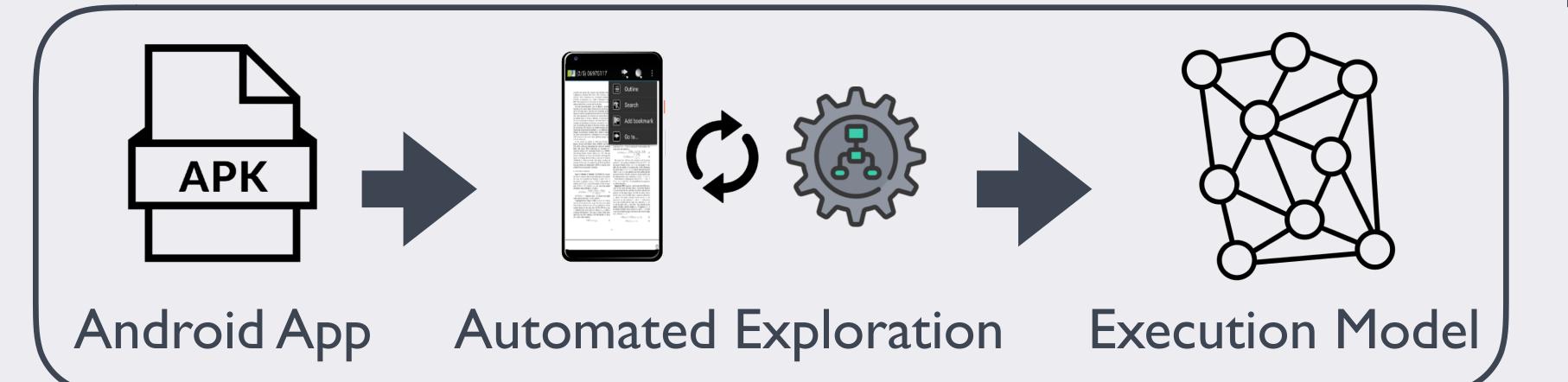
 v_v : Target screen

EULER'S COMPONENTS & WORKFLOW





2) Execution Model Generation



3) Quality Assessment

S2R Matching

S2R Inference

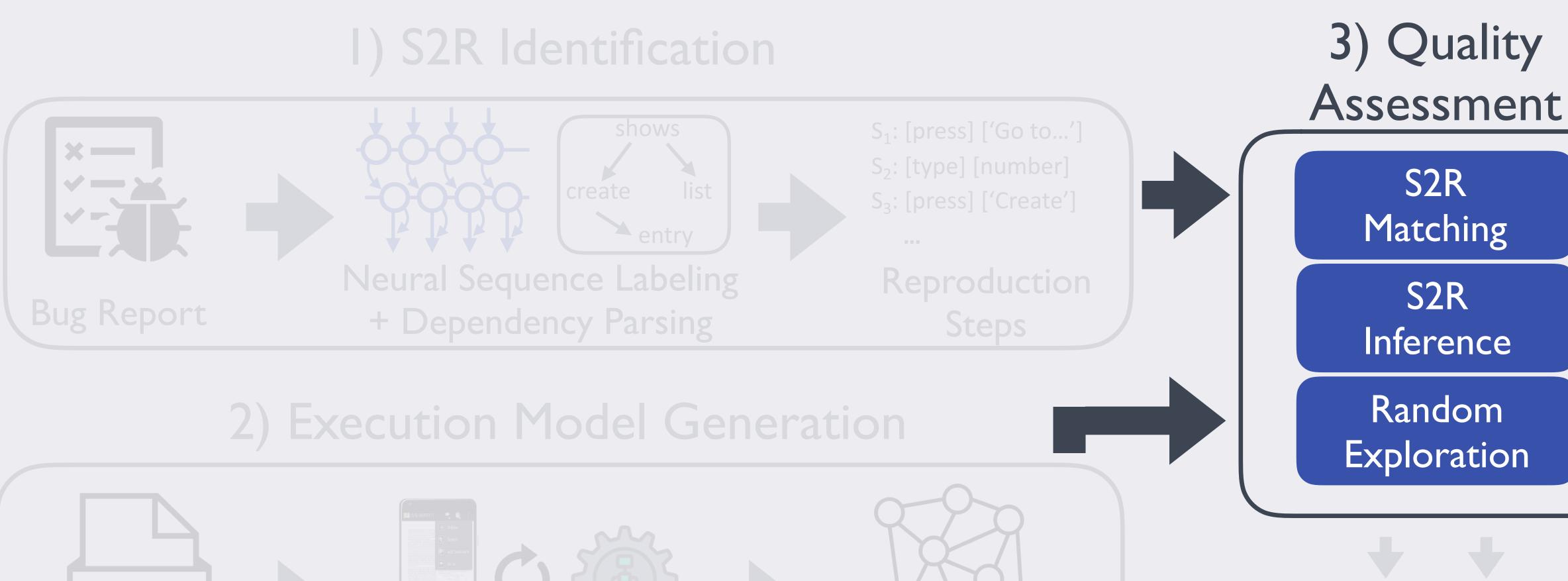
Random Exploration





#	S2R	Quality Annotations
1	S ₁	EM MS
2	S ₂	AS
3	S ₃	VM

EULER'S COMPONENTS & WORKFLOW



Automated Exploration

Android App

Execution Model

		Quality Annotations
1	S ₁	
2	S ₂	
3	S ₃	

COMPONENT RESOLUTION & QUERY FORMULATION

Individual S2R

S₁: [press] ['Go to...']

S₁: [enter] ['10'] [on] [price]

S₁: [set] [price] [to] ['10']

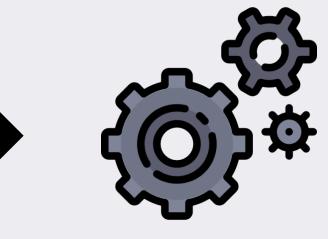
Queries

q₁: *press* + *Go* to...

q₂: *Go to...*

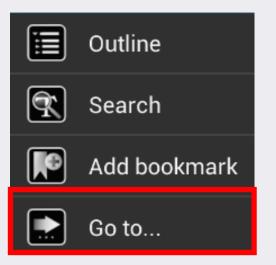
q₁: price

Matching Algorithm





Resolved Component



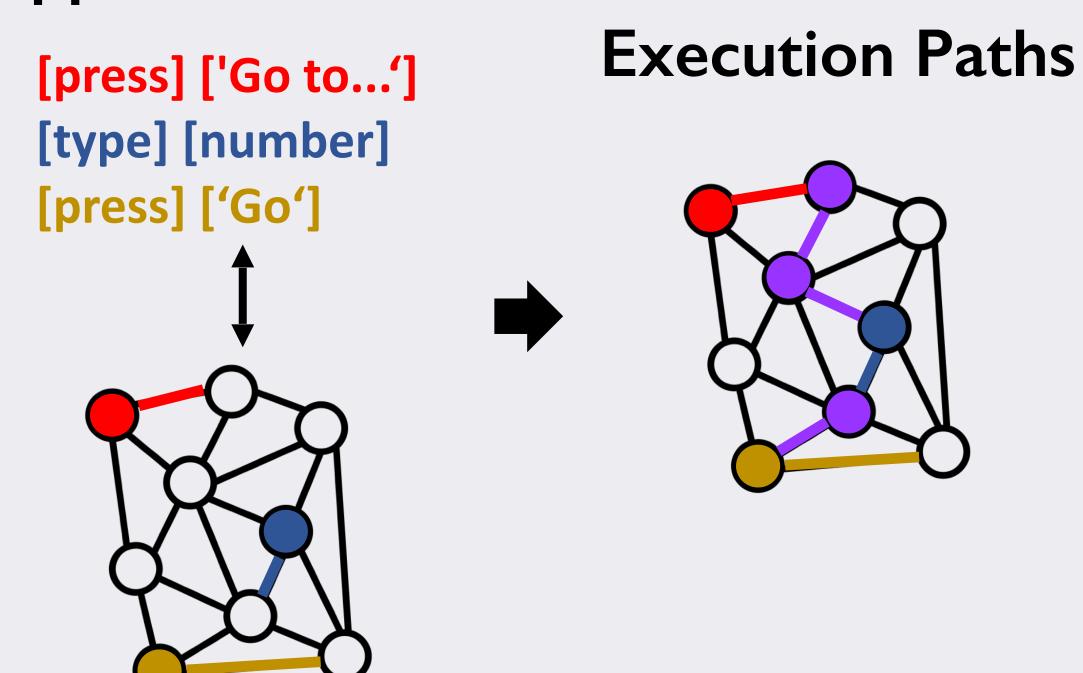
S2R EXECUTION & INFERENCE

Matched S2Rs to app interactions

```
[press] ['Go to...']
[type] [number]
[press] ['Go']
```

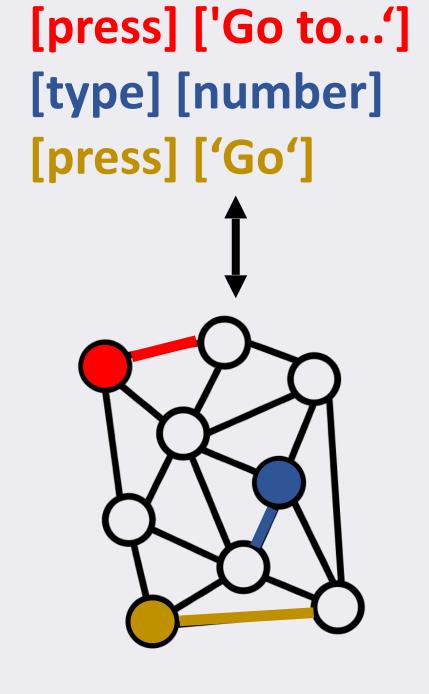
S2R EXECUTION & INFERENCE

Matched S2Rs to app interactions

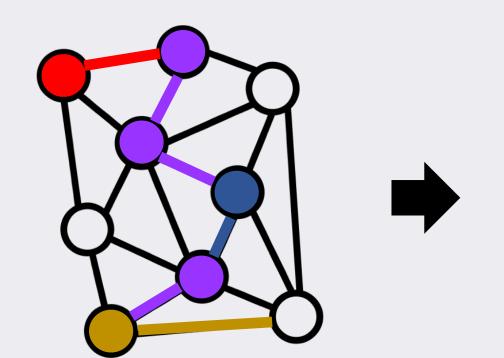


S2R EXECUTION & INFERENCE

Matched S2Rs to app interactions



Execution Paths

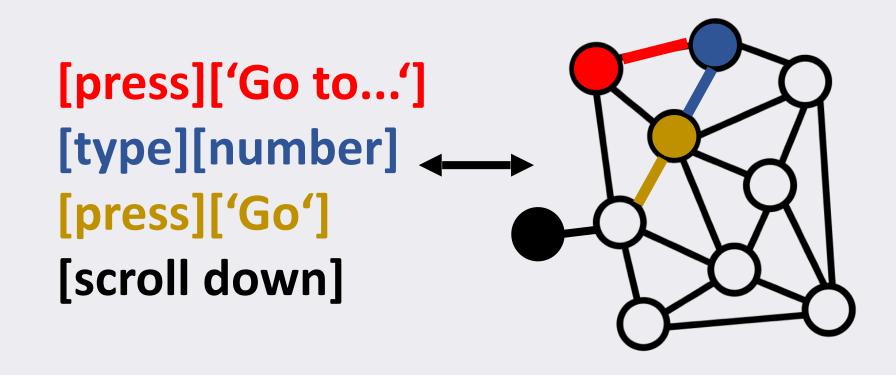


Inferred Missing
Steps

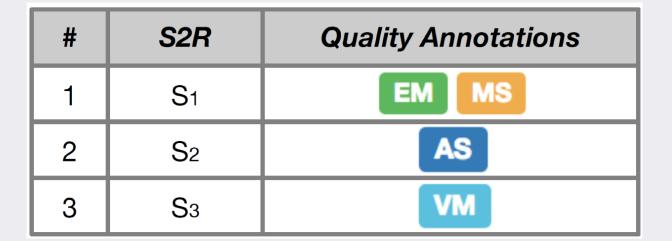
[press]['Go to...']
[scroll down]
[tap]['Enter number']
[type][number]
[scroll up]
[press]['Go']

S2R QUALITY ANNOTATION

Matched S2Rs to app interactions



Quality Report

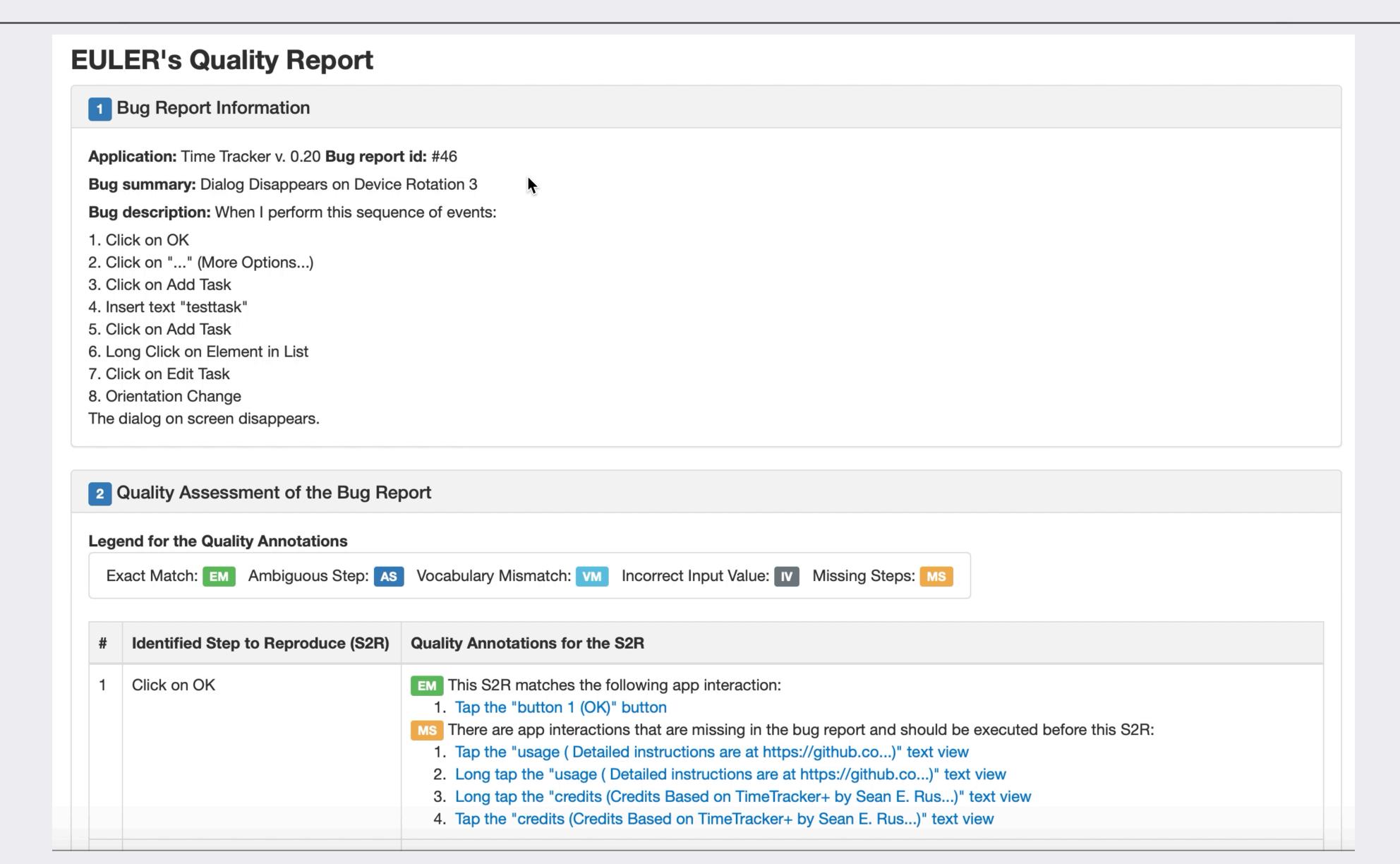


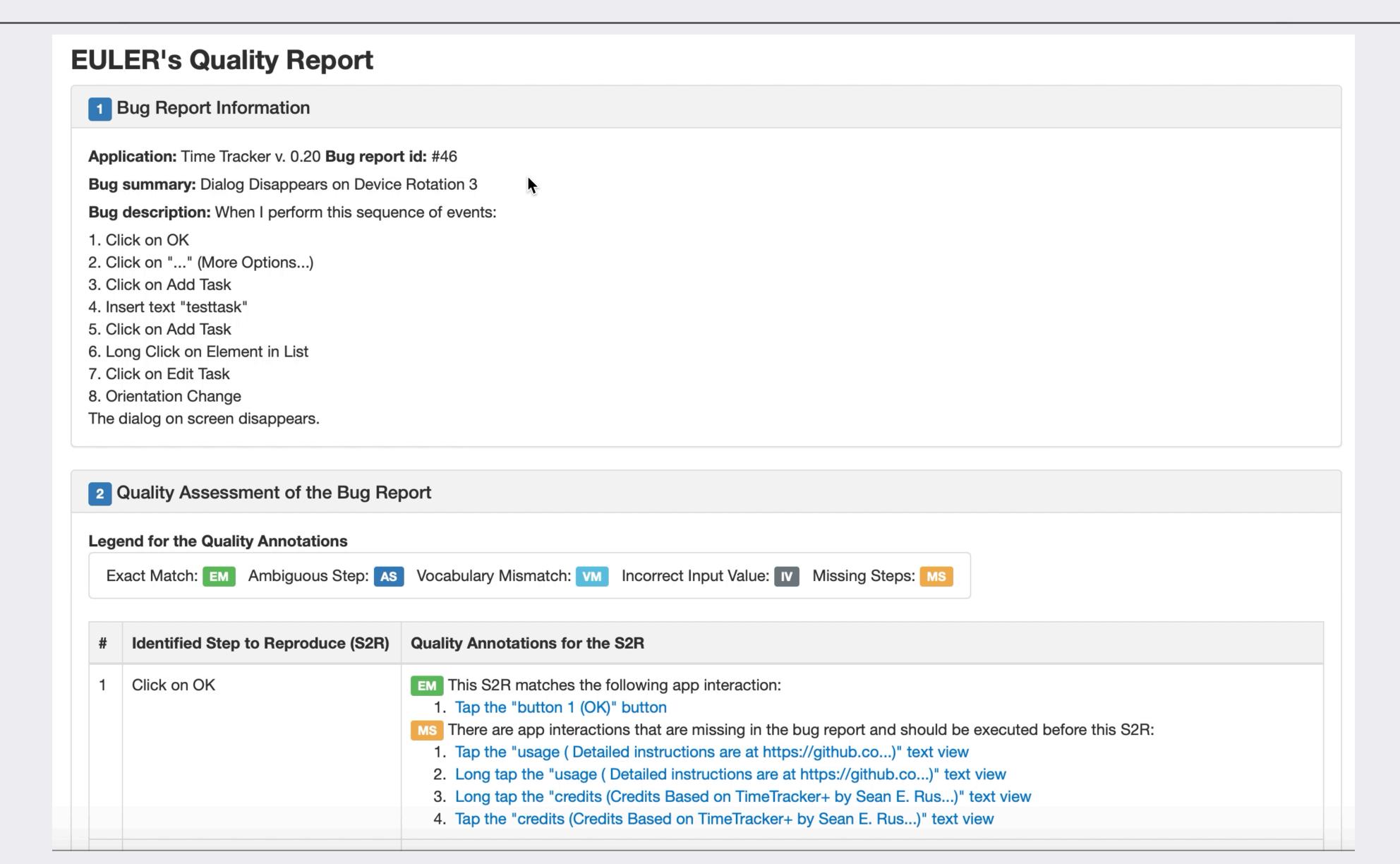
Lege	Legend for the Quality Annotations		
Ex	Exact Match: EM Ambiguous Step: AS Vocabulary Mismatch: VM Missing Steps: MS		
_			
#	Identified S2R	Quality Annotations	
1	Add favorites	This S2R matches the following app interaction: 1. Tap the "item fav (Add to favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the image button 2. Tap the "Chaos Communication Camp Opening" view	
2	Go into favorites	This S2R matches the following app interaction: 1. Tap the "item starred list (Show favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the "Navigate up" image button 2. Tap the image button 3. Tap the drop down list 4. Tap the list view	
3	Select event	This S2R matches multiple actions (e.g., "long click" or "click").	
4	Remove event in event details screen	The term "event in event details screen" does not match a GUI component from the app.	
5	Hit BACK button to return	This S2R matches the following app interaction: 1. Tap the back button	

Lege	Legend for the Quality Annotations		
Ex	Exact Match: EM Ambiguous Step: AS Vocabulary Mismatch: VM Missing Steps: MS		
#	Identified S2R	Quality Annotations	
1	Add favorites	This S2R matches the following app interaction: 1. Tap the "item fav (Add to favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the image button 2. Tap the "Chaos Communication Camp Opening" view	
2	Go into favorites	This S2R matches the following app interaction: 1. Tap the "item starred list (Show favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the "Navigate up" image button 2. Tap the image button 3. Tap the drop down list 4. Tap the list view	
3	Select event	This S2R matches multiple actions (e.g., "long click" or "click").	
4	Remove event in event details screen	The term "event in event details screen" does not match a GUI component from the app.	
5	Hit BACK button to return	This S2R matches the following app interaction: 1. Tap the back button	

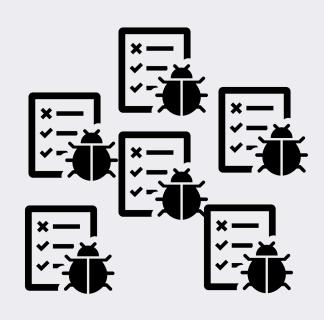
Lege	Legend for the Quality Annotations		
Ex	Exact Match: EM Ambiguous Step: AS Vocabulary Mismatch: VM Missing Steps: MS		
-			
#	Identified S2R	Quality Annotations	
1	Add favorites	This S2R matches the following app interaction: 1. Tap the "item fav (Add to favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the image button 2. Tap the "Chaos Communication Camp Opening" view	
2	Go into favorites	This S2R matches the following app interaction: 1. Tap the "item starred list (Show favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the "Navigate up" image button 2. Tap the image button 3. Tap the drop down list 4. Tap the list view	
3	Select event	This S2R matches multiple actions (e.g., "long click" or "click").	
4	Remove event in event details screen	The term "event in event details screen" does not match a GUI component from the app.	
5	Hit BACK button to return	This S2R matches the following app interaction: 1. Tap the back button	

Legend for the Quality Annotations			
Ex	Exact Match: EM Ambiguous Step: AS Vocabulary Mismatch: VM Missing Steps: MS		
#	Identified S2R	Quality Annotations	
1	Add favorites	This S2R matches the following app interaction: 1. Tap the "item fav (Add to favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the image button 2. Tap the "Chaos Communication Camp Opening" view	
2	Go into favorites	This S2R matches the following app interaction: 1. Tap the "item starred list (Show favorites)" text view MS There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the "Navigate up" image button 2. Tap the image button 3. Tap the drop down list 4. Tap the list view	
3	Select event	This S2R matches multiple actions (e.g., "long click" or "click").	
4	Remove event in event details screen	The term "event in event details screen" does not match a GUI component from the app.	
5	Hit BACK button to return	This S2R matches the following app interaction: 1. Tap the back button	





EVALUATING EULER

















24 bug reports:

- 5 crashes
- 14 functional
- 5 look-n-feel

6 Android apps:

- GnuCash
- Mileage
- Schedule

•

12 evaluators:

- MS and PHD students
- B. Analyst

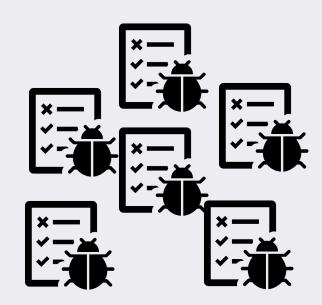
•

RESEARCH QUESTIONS

RQ₁: What is the *accuracy* of Euler in Identifying and Assessing S2Rs?

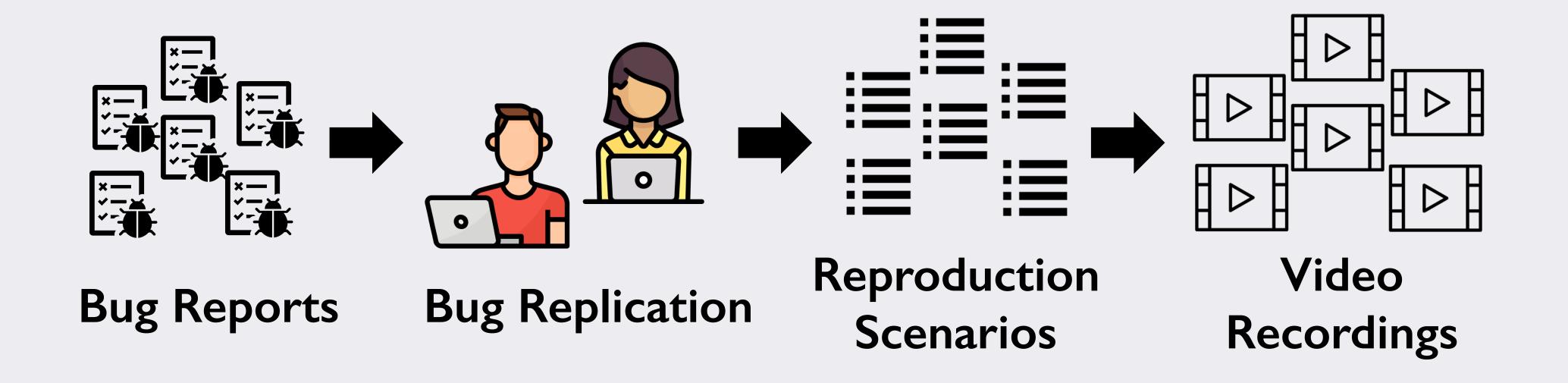
RQ2: How useful are Euler's Reports?

GENERATING IDEAL REPRODUCTION SCENARIOS

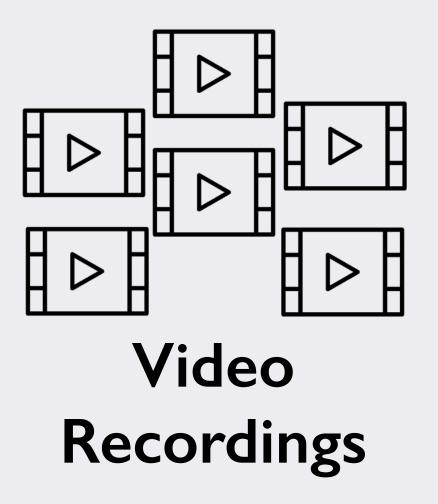


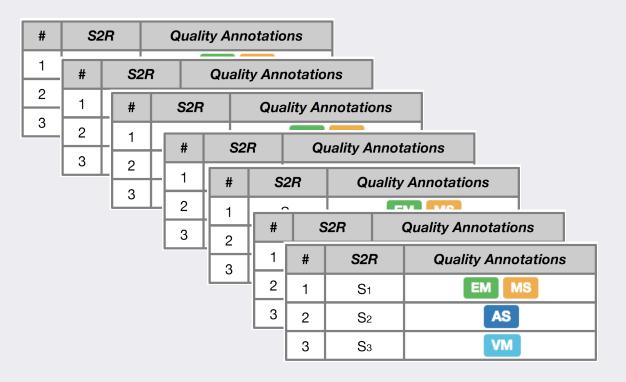
Bug Reports

GENERATING IDEAL REPRODUCTION SCENARIOS



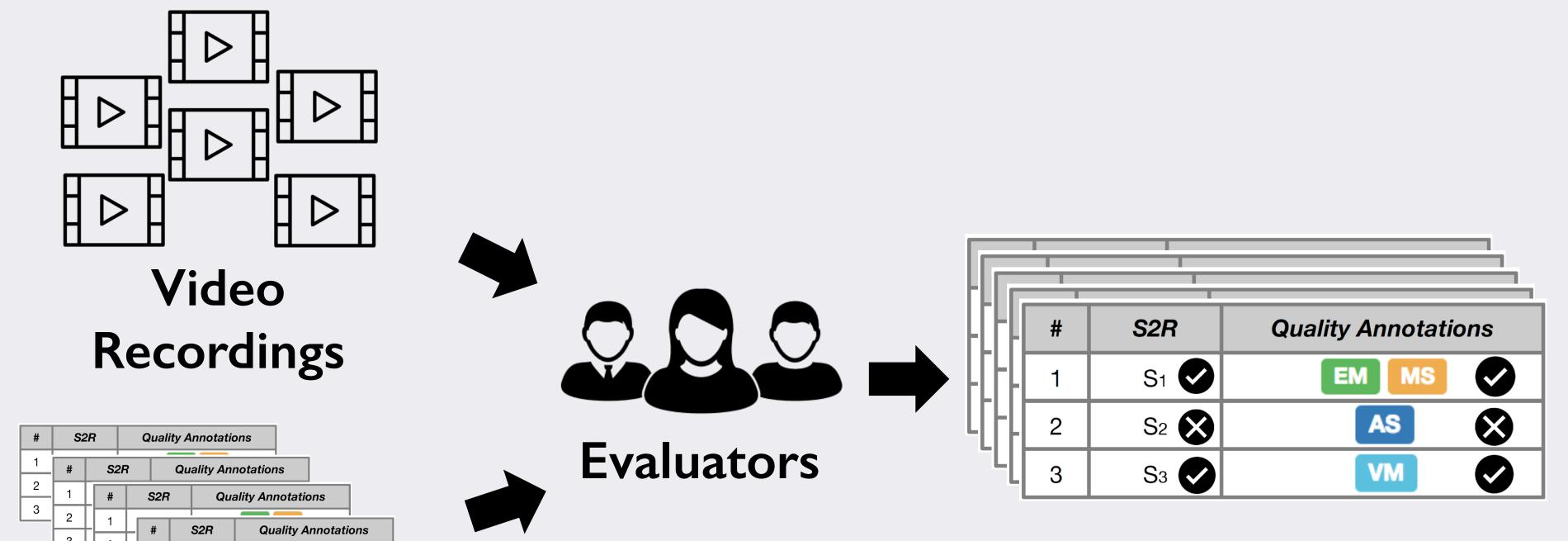
EXTERNAL EVALUATION OF EULER





Quality Reports

EXTERNAL EVALUATION OF EULER



Quality Reports

 S_2

Quality Annotations

Quality Annotations

Quality Annotations

AS

- Correct/Incorrect
 S2Rs and
 Annotations
- Usefulness/Quality of the Feedback

EULER'S ACCURACY

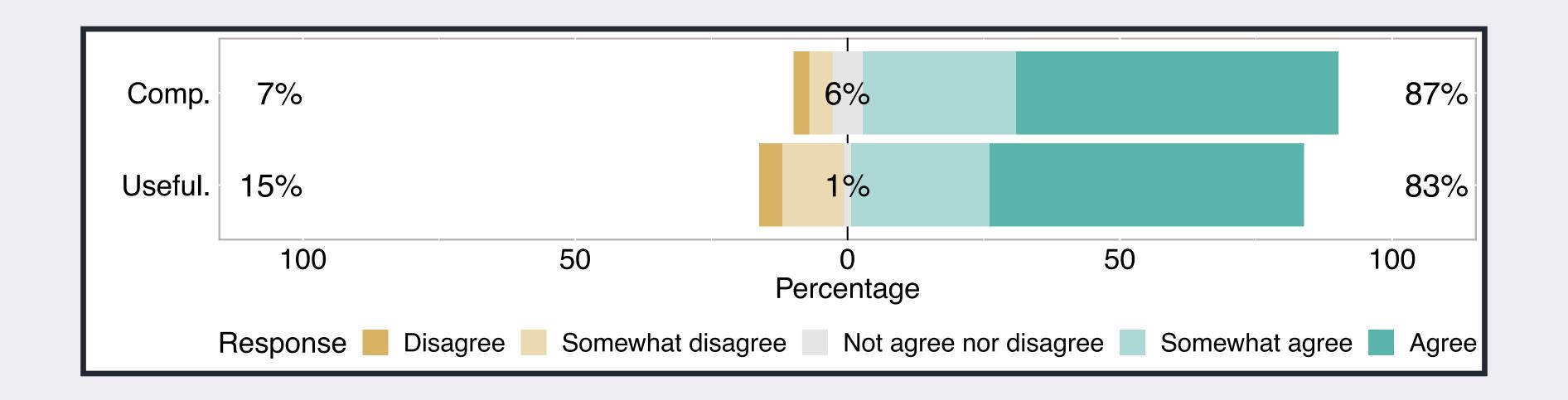
- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
 - 92/158 (58%) are correctly inferred
 - 5/7 steps for bug report (on avg)

EULER'S ACCURACY

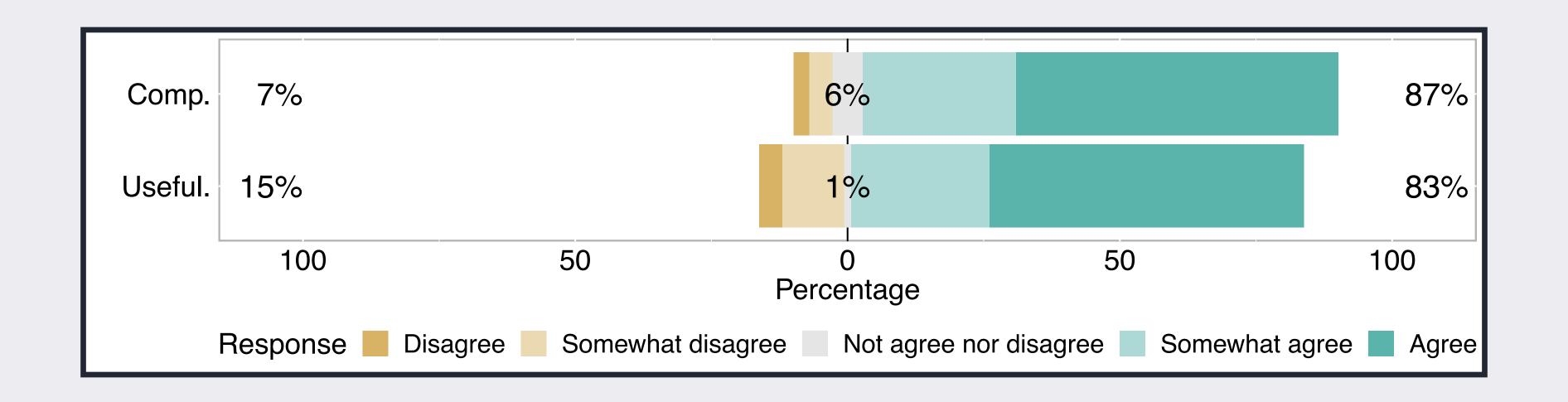
- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
 - 92/158 (58%) are correctly inferred
 - 5/7 steps for bug report (on avg)

Euler assesses the quality of S2Rs with high accuracy

EULER'S ACCURACY



EULER'S ACCURACY



Euler's quality reports are both <u>easy to understand</u> and would prove <u>helpful when writing bug reports</u>

PARTICIPANT FEEDBACK

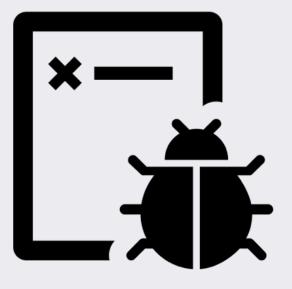
PARTICIPANT FEEDBACK

"EULER's S2Rs are pretty descriptive and would guide the user to complete better the bug description"

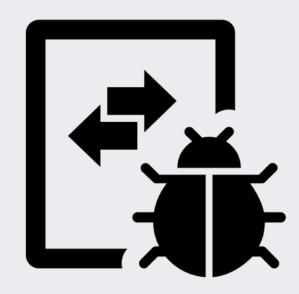
"EULER correctly flags words such as 'find' and 'fix' that do not directly translate to an app action".

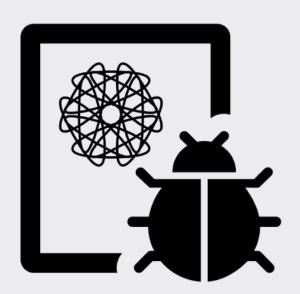
"Developers/maintainers would find this tool *very* useful for their debugging process"

"[The missing steps] help avoid the guessing part when reproducing the bug"



Missing

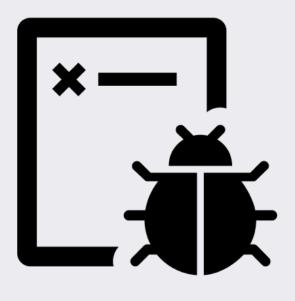


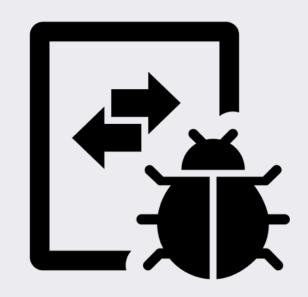


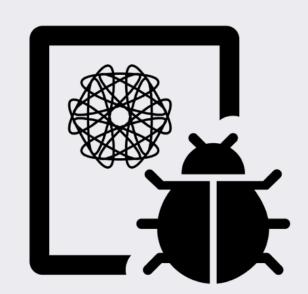
Ambiguous

Incomplete

T. Zimmermann, R. <u>Premraj</u>, N. <u>Bettenburg</u>, S. Just, A. <u>Schroter</u> and C. Weiss, "What Makes a Good Bug Report?," in *IEEE Transactions on Software Engineering*





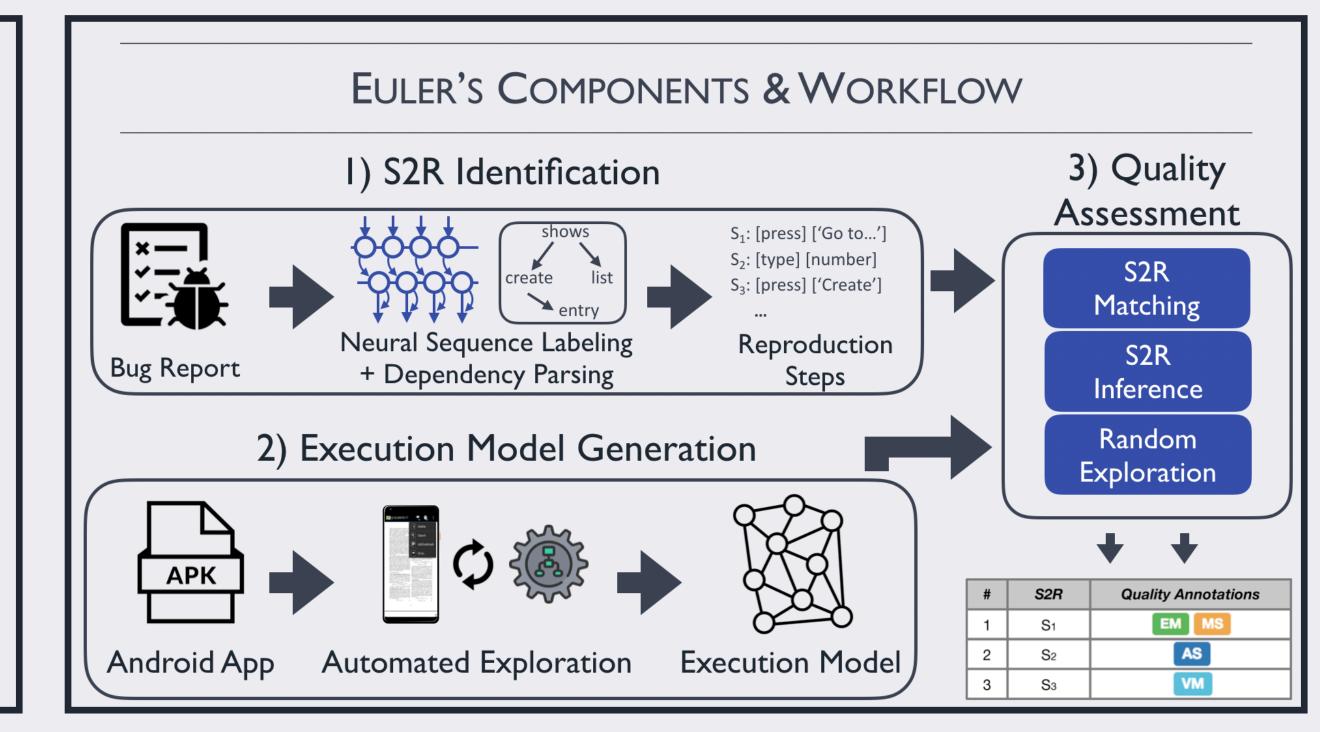


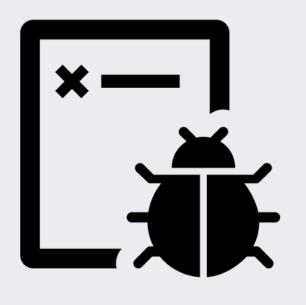
Missing

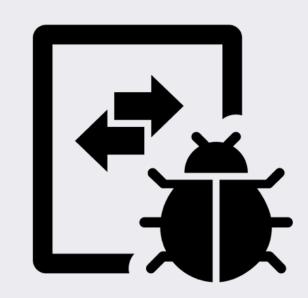
Ambiguous

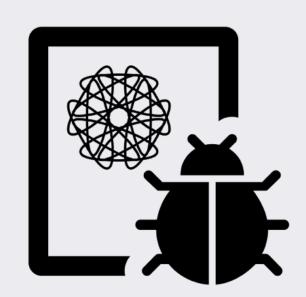
Incomplete

T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter and C. Weiss, "What Makes a Good Bug Report?," in IEEE Transactions on Software Engineering







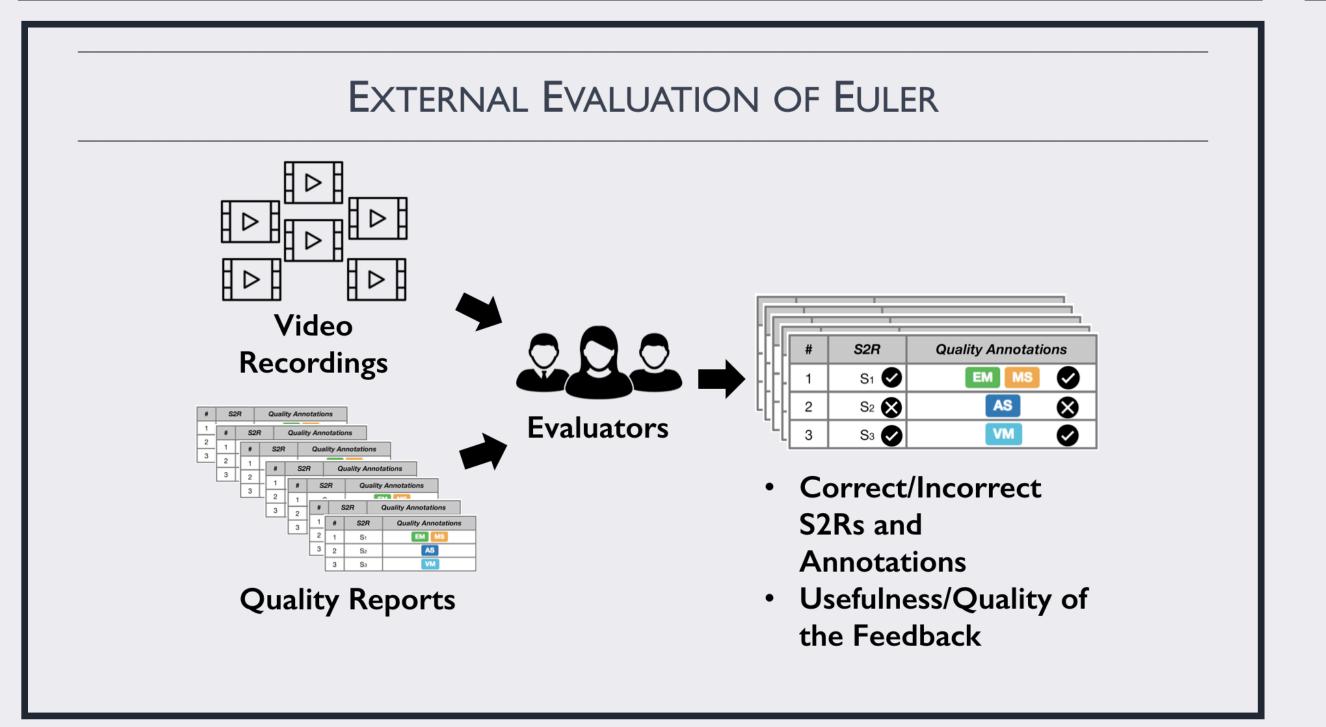


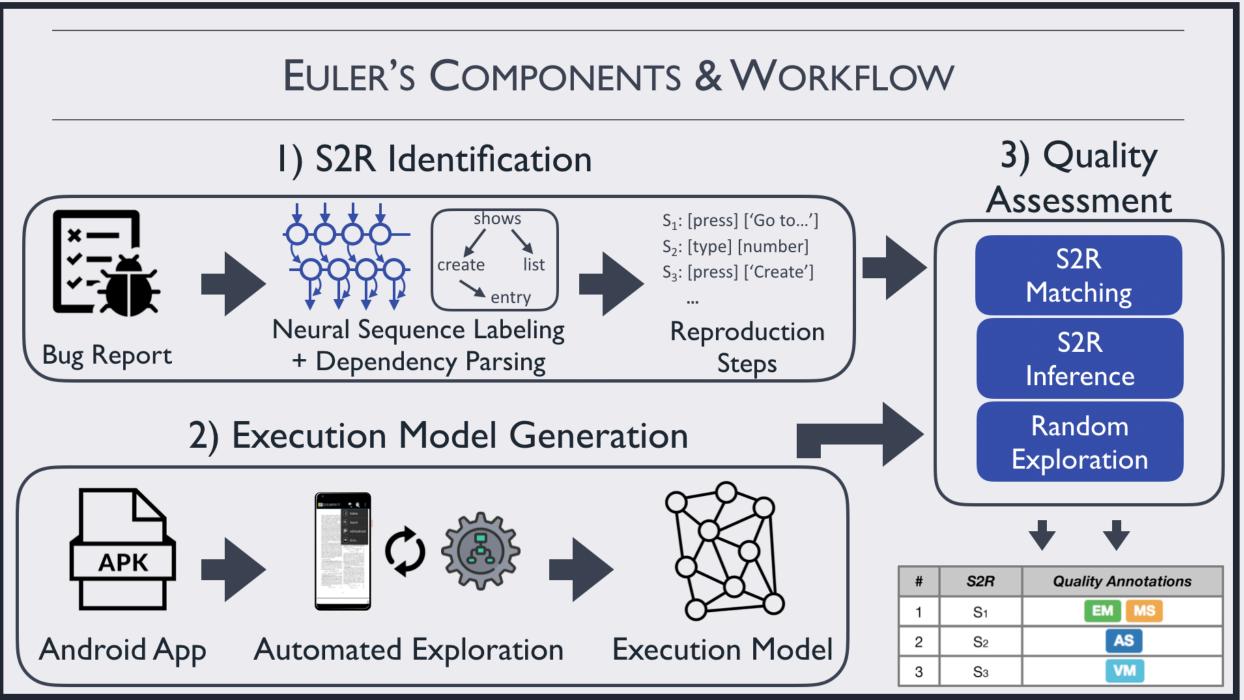
Missing

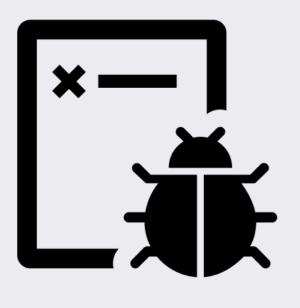
Ambiguous

Incomplete

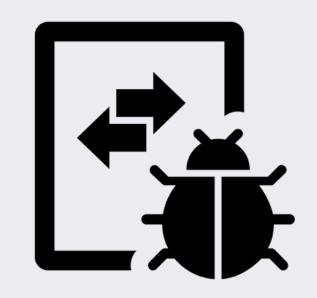
T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter and C. Weiss, "What Makes a Good Bug Report?," in IEEE Transactions on Software Engineering

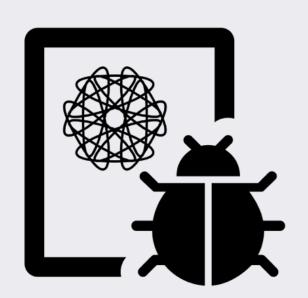






Missing

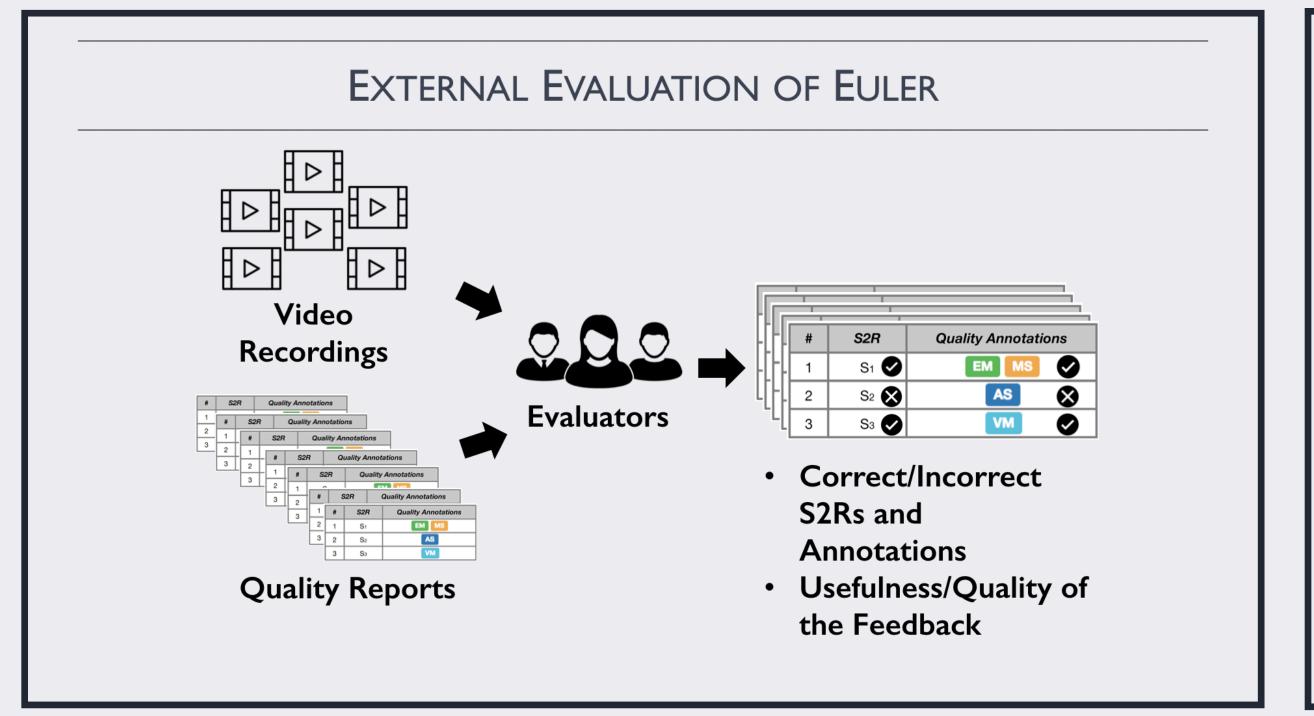


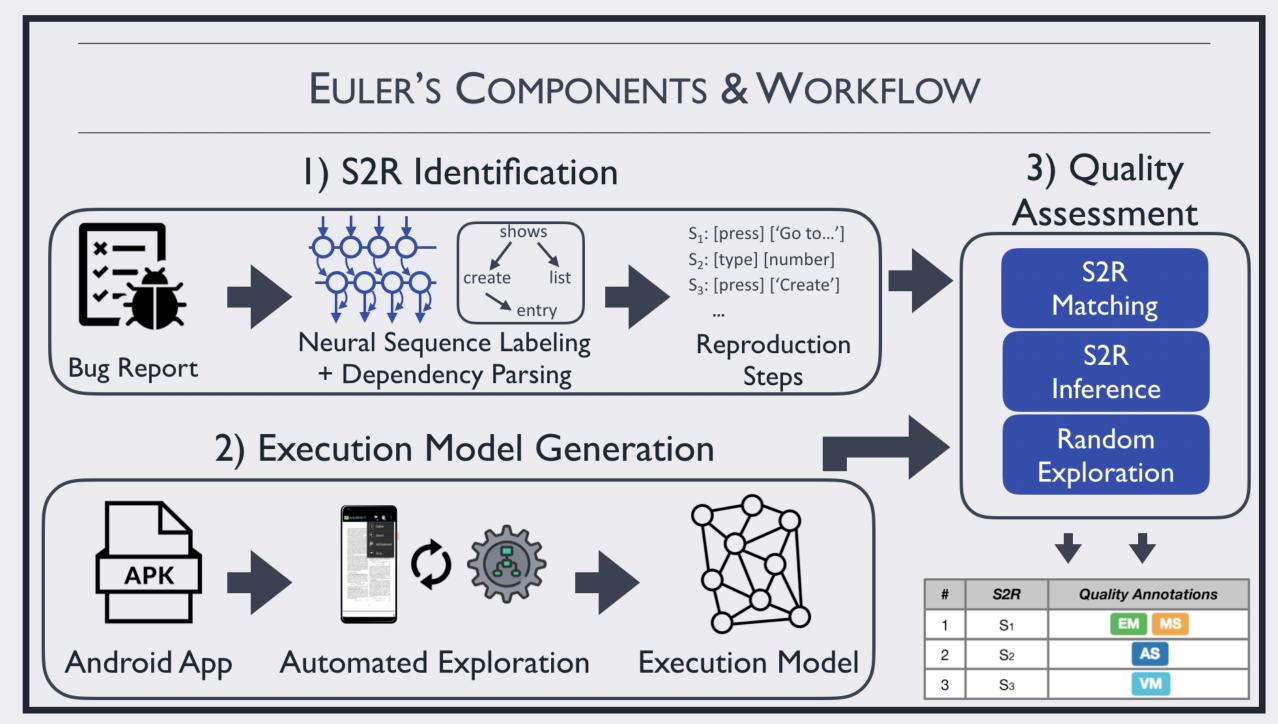


Ambiguous

Incomplete

T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter and C. Weiss, "What Makes a Good Bug Report?," in IEEE Transactions on Software Engineering





EULER'S ACCURACY

- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
- 92/158 (58%) are correctly inferred
- 5/7 steps for bug report (on avg)

Euler assesses the quality of S2Rs with high accuracy



Kevin Moran
William & Mary
@kevpmo
kpmoran@cs.wm.edu
https://www.kpmoran.com

Thank you! Any Questions?



Oscar Chaparro



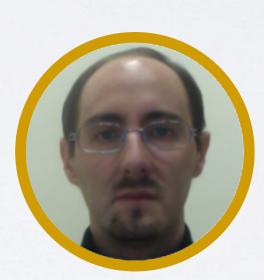
Carlos Bernal



Jing Lu



Andrian Marcus



Massimiliano Di Penta



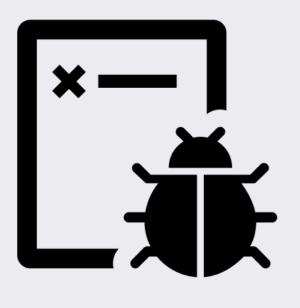
Denys Poshyvanyk



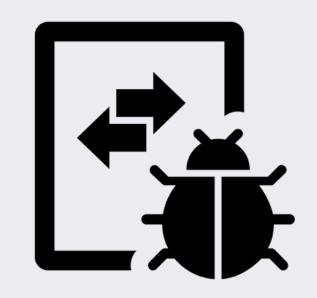
Vincent Ng

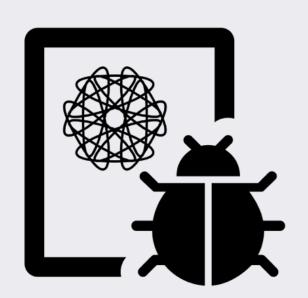






Missing

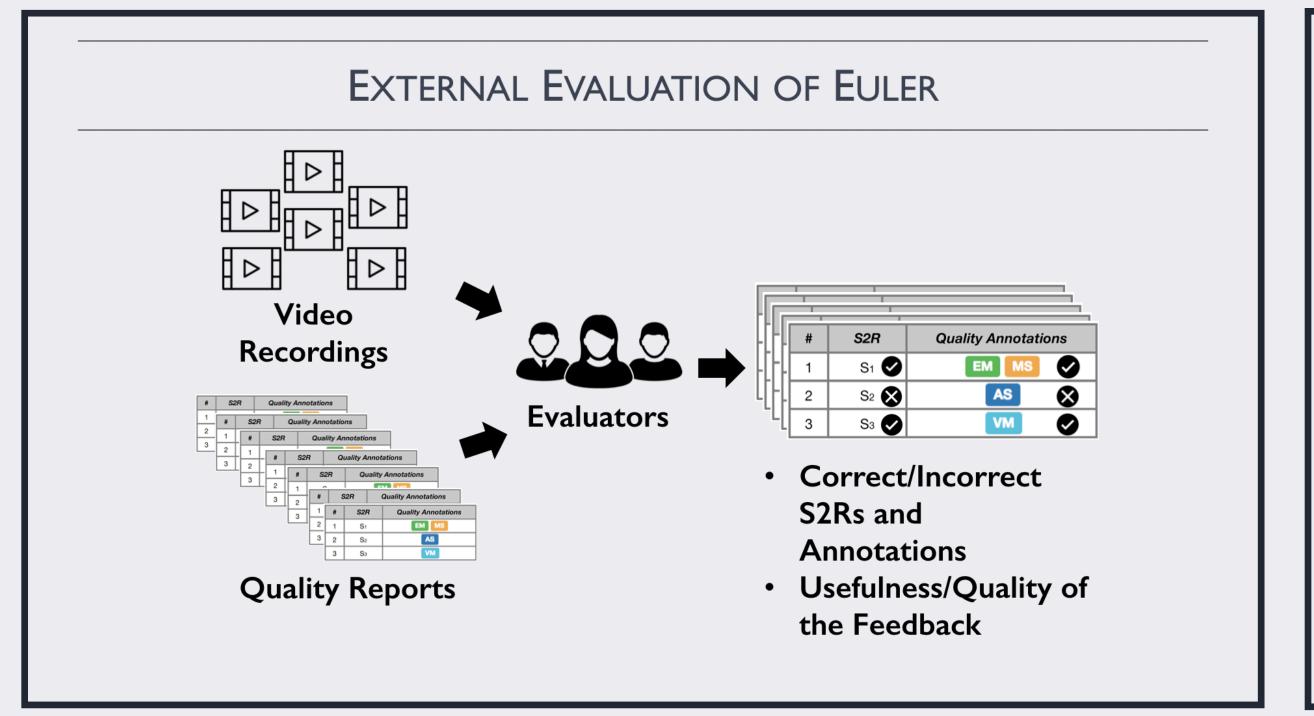


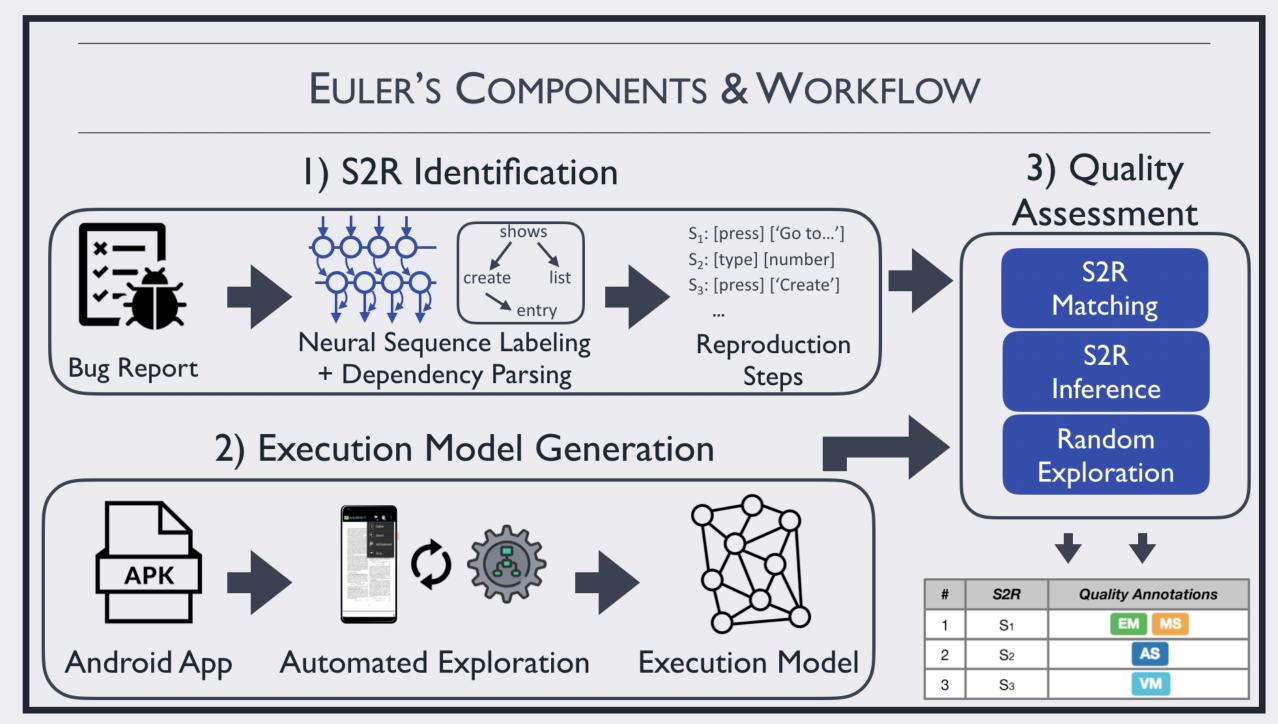


Ambiguous

Incomplete

T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter and C. Weiss, "What Makes a Good Bug Report?," in IEEE Transactions on Software Engineering





EULER'S ACCURACY

- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
- 92/158 (58%) are correctly inferred
- 5/7 steps for bug report (on avg)

Euler assesses the quality of S2Rs with high accuracy

ADDITIONAL SLIDES

S2R RESOLUTION

Individual S2R

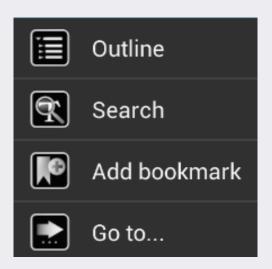
S₁: [press] ['Go to...']



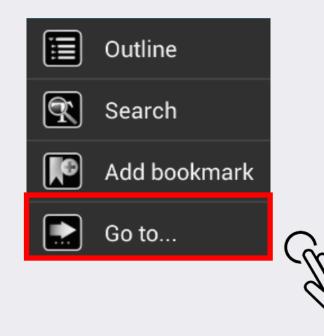
Resolved interaction

 $e_1 \in E = (v_x, e, c, v_y = null)$

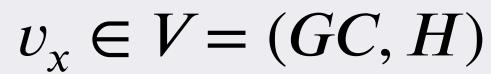
App state (screen)







S₁: [tap] on ['Go to...'] button



TEXTUAL SIMILARITY MEASURE

$$similarity(s_1, s_2) = \frac{\left| LCS(s_1, s_2) \right|}{avg(\left| s_1 \right|, \left| s_2 \right|)}$$

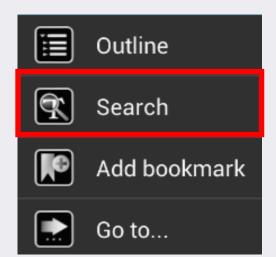
- s_1 and s_2 are two term sequences
- $LCS(s_1, s_2)$ is the Longest Common Substring (at term level)
- $avg(|s_1|, |s_2|)$ is the **average length** of both sequences

GUI COMPONENT RESOLUTION EXAMPLE

Screen

Individual S2R

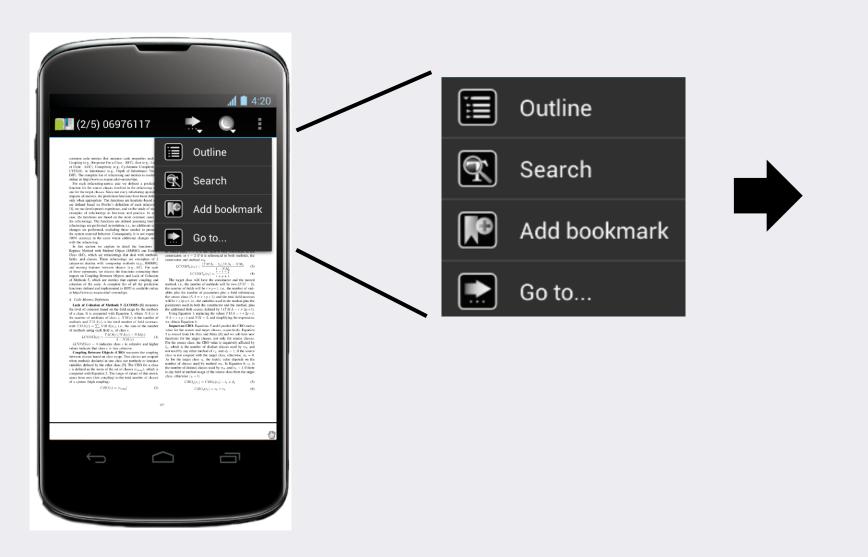




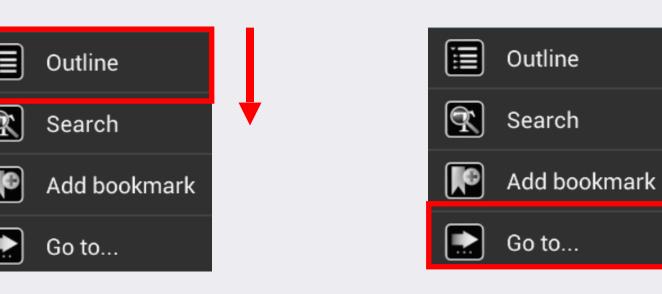
$$similarity('search number', 'search') = \frac{\left|'search'\right|}{avg(2,1)} = \frac{1}{1.5} = 0.67$$

SYSTEMATIC APP EXPLORATION

GUI Ripping GUI hierarchy







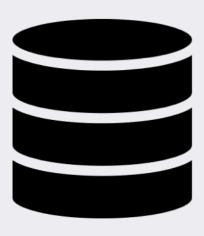
Top-down

Bottom-up



Execution database:

- App activities/screens
- Sequential GUI interactions



COMPONENT MATCHING ALGORITHM

Algorithm 1: Matching Algorithm

```
Data: Query (q), list of GUI components (GC)
   Result: Matched GUI component (c), or mismatch/multiple-match result (r)
   Function execute\_query(q, GC)
        gc\_candidates = [];
        foreach c in GC do
            S = \text{compute\_similarity}(q,c);
 4
            if S \geq 0.5 then
 5
                gc\_candidates.add(c)
 6
        end
        if gc\_candidates.size() == 1 then
            c = \text{gc\_candidates.first()};
            return c;
10
        if gc\_candidates.size() > 1 then
11
            return execute_heuristics(gc_candidates);
12
        if gc\_candidates.size() == 0 then
13
            q' = \text{reformulate\_query}(q);
14
            return execute_query(q', GC)
15
```

COMPONENT MATCHING ALGORITHM

```
    For each component in the screen:

    Compute its textual similarity with the query

 • If the similarity is high enough (>0.5)
   • The component is selected as candidate

    Return the top candidate or reformulate the query

 using synonyms
```

S2R MATCHING

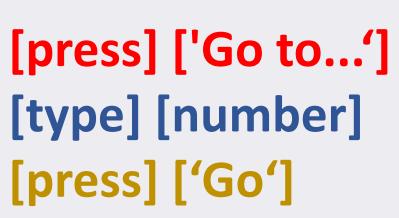
Individual S2R

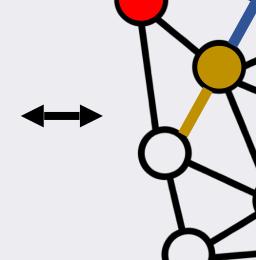
[press] ['Go to...'] [type] [number] [press] ['Go']



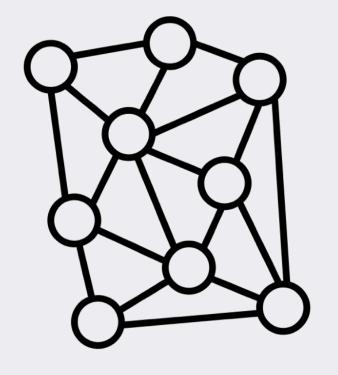
Matched S2Rs to app interactions

[type] [number]





Execution Model





S2R MATCHING ALGORITHM

Alg Starting from a state/screen in the graph: Select the adjacent states within n levels Resolve the S2R on each state Match the resolved actions against the graph transitions/interactions Return the closest matched interaction to the 11 starting state

S2R MATCHING ALGORITHM

 Starting from a state/screen in the graph: Select the adjacent states within n levels Resolve the S2R on each state Match the resolved actions against the graph transitions/interactions Return the closest matched interaction to the starting state