Utrecht, The Netherlands, 19 – 22 March
REFSQ
24th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality
2018

Utrecht University

Evaluation of Techniques to Detect Wrong Interaction Based Trace Links

software engineering heidelberg

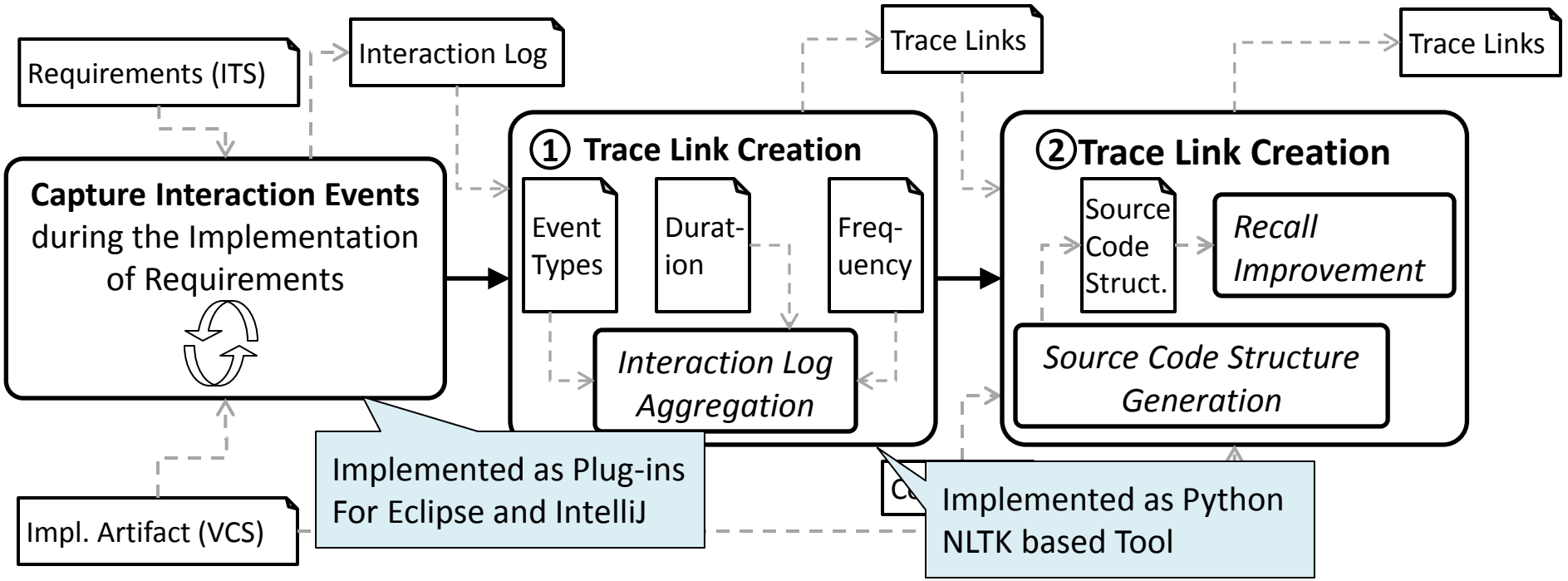
UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Paul Hübner and Barbara Paech
Institute for Computer Science
Heidelberg University

Motivation: Why Do We Need Another Trace Link Creation Approach?

- Our application context for trace links
  - Unstructured requirements in an issue tracking system (ITS)
  - Source code in a version control system
  - Direct usage of trace links during development
- Usage of trace links during the development requires good precision of trace links
  - Wrong links will prevent developers from using the links
  - Our prime research goal: **optimize precision up to 100%**
- Precision of existing trace link creation approaches is typically below 50%
  - Focus is the increase of recall to use trace links retrospectively (bad precision is tolerated)

Interaction based Trace Link Creation Approach (IL)



Evaluation of IL Approach

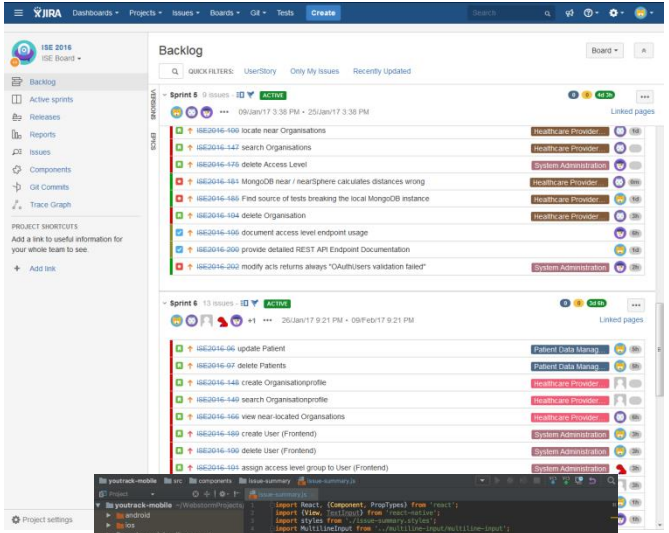
- RFESQ17: Study with open source project data
  - IL approach: **100% precision** and **~90% relative recall**
  - IR (VSM & LSI): in best cases **34% precision** and **53% relative recall**
- 2018: 2nd Study with student project data
  - Complete gold standard and real recall values
  - ... it turned out different

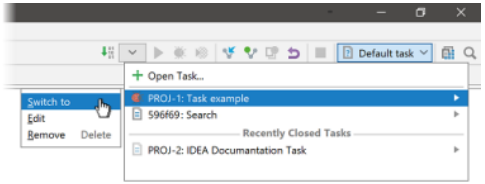
→ **New idea for this study:** Evaluate different techniques to avoid the creation of wrong links

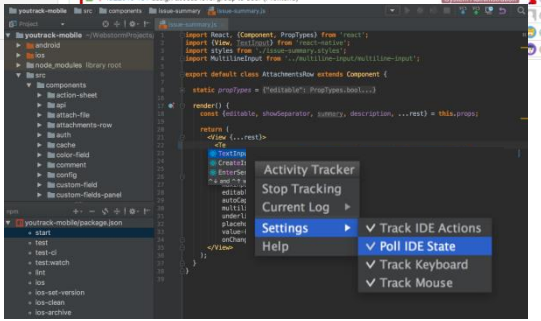
- The precision of created links was not as good as expected
  - IL approach: only **43% precision** and **73% recall**
  - IR (VSM & LSI): in best cases **34% precision** and **48% recall** (low thresholds)

The Student Project

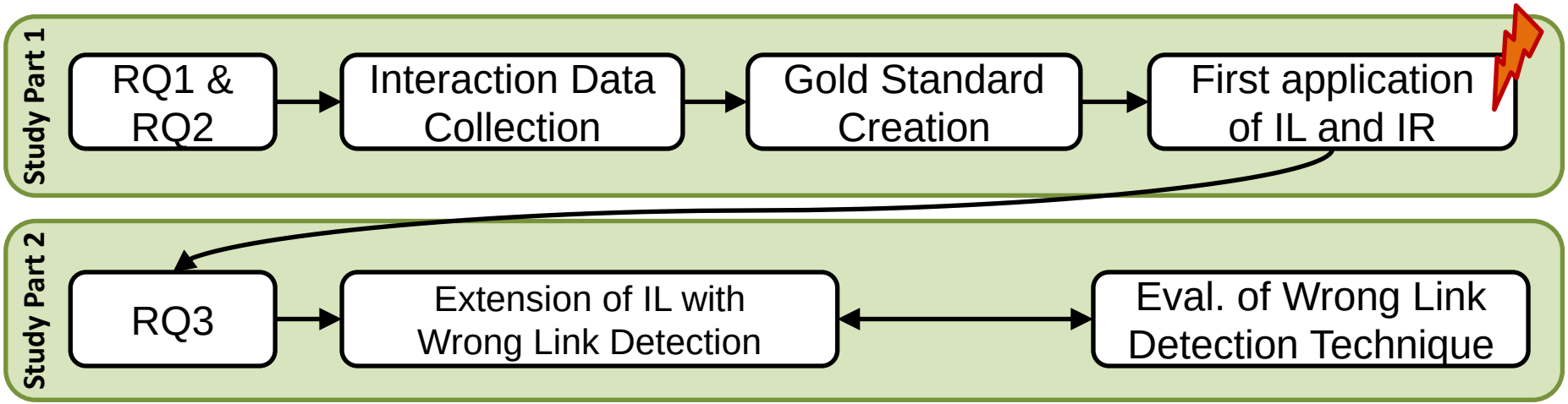
- 6 month (oct 2016 – mar 2017), scrum oriented with real world customer
- 7 sprints and 7 developers
- Development of an Open ID systems for patient healthcare data
- Infrastructure: Jira, Git, IntelliJ (Webstorm) IDE
- Interaction Tracking in IntelliJ by Activity Tracker
  - Manual selection of the issue worked on by developer







Experiment Design



RQ1: **Precision (P)** and **recall (R)** of **IL** created trace links?

RQ2: **P** and **R** of **IR** created trace links?

RQ3: P and R of IL created trace links **with wrong link detection techniques**?

Interaction Log Recording, Gold Standard Creation, RQ1 & RQ2

- Recorded Interactions during the project
  - ~2 Mio interactions between 42 user stories and 312 source code files
- Gold Standard created by link candidate questionnaire
  - 217 correct links between 19 user stories and 151 source code files
- Initial application of IL (RQ1 & RQ2):

Approach	Link Cand.	Correct links	Wrong links	Not found	Precision	Recall
IL	372	160	212	57	0.430	0.737
$IR_{VSM(0.2)}$	642	104	538	113	0.162	0.480
$IR_{LSI(0.1)}$	102	35	67	182	0.343	0.161

RQ3: Detection Techniques for Wrong Trace Links - Duration

Observations

- **Select interactions** create **more wrong** links than edit
- **Wrong** selects are based in the **IDE navigator** not in the code editor

→ Focus on edit interactions

Characteristics of wrong interactions

- **Duration**: interactions with **longer duration** are **more likely** to be **correct**
  - Get the duration based on the deltas of interaction timestamps
  - How to set the threshold value for duration?
    - Ensure that **no correct links** get **lost**
    - Use a **fixed value** (e.g. 2.0 sec) or **average valu**

$duration_{01} = timestamp_{02} - timestamp_{01}$
$duration_{01} = 28.720$ sec

```
01: 2016-10-07T15:33:42.270+02:00;Dev1;Action;$EditorEnter;ise;Hierarchie;
        /WebstormProjects/ise/test/patient_controller_test.js
02: 2016-10-07T15:34:11.090+02:00;Dev1;Action;EditorBackSpace;ise;Editor;
        /WebstormProjects/ise/lib/patient_controller.js
```

RQ3: Detection Techniques for Wrong Trace Links - Frequency

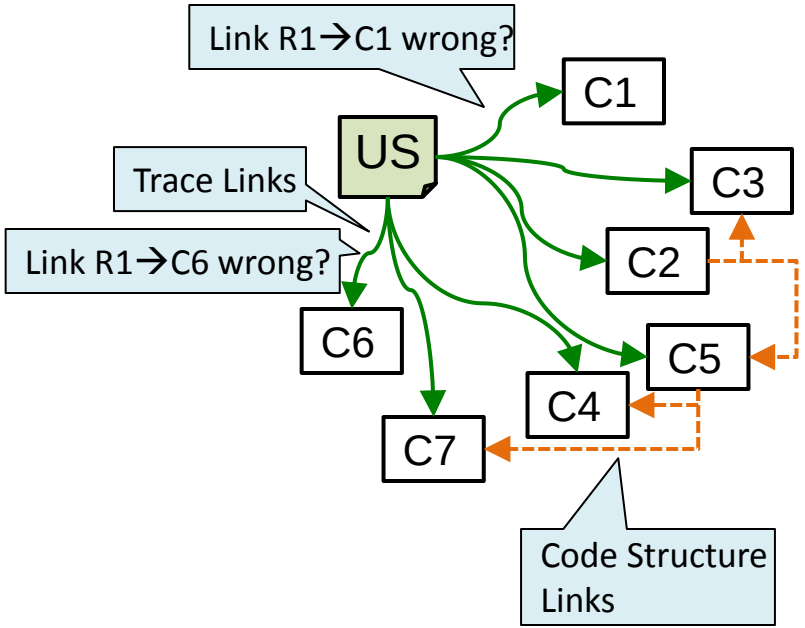
Characteristics of wrong interactions

- **Frequency:** links to **frequently interacted** files are **more likely** to be **correct**
  - Count the number of interactions with source code files during the implementation of a requirement

frequency for `patient_controller.js` in `ISE2016-38` = 2

```
01: 2016-10-07T15:32:17.120+02:00;Task Activation;ISE2016-38: Validate JSON;
          ise;Dialog
02: 2016-10-07T15:33:28.740+02:00;Dev1;Action;EditorBackSpace;ise;Dialog;
          /WebstormProjects/ise/lib/patient_controller.js
03: 2016-10-07T15:33:42.270+02:00;Dev1;Action;$EditorEnter;ise;Hierarchie;
          /WebstormProjects/ise/lib/patient_controller.js
04: 2016-10-07T15:35:11.090+02:00;Dev1;Action;EditorBackSpace;ise;Editor;
          /WebstormProjects/ise/test/patient_controller_test.js
...
42: 2016-10-07T15:21:17.120+02:00;Task Deactivation;ISE2016-38: Validate JSON;
          ise;Dialog
```

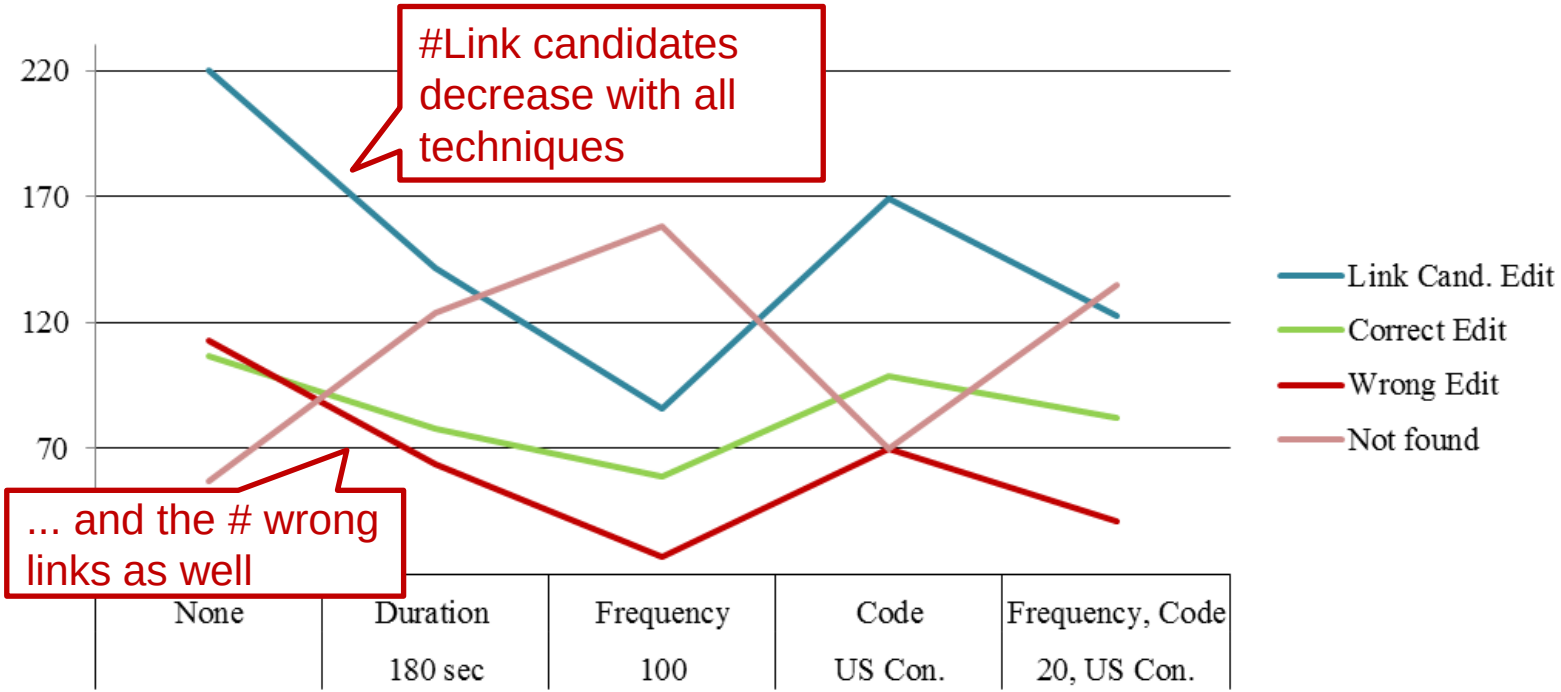
RQ3: Detection Techniques for Wrong Trace Links – Code Connection for User Story



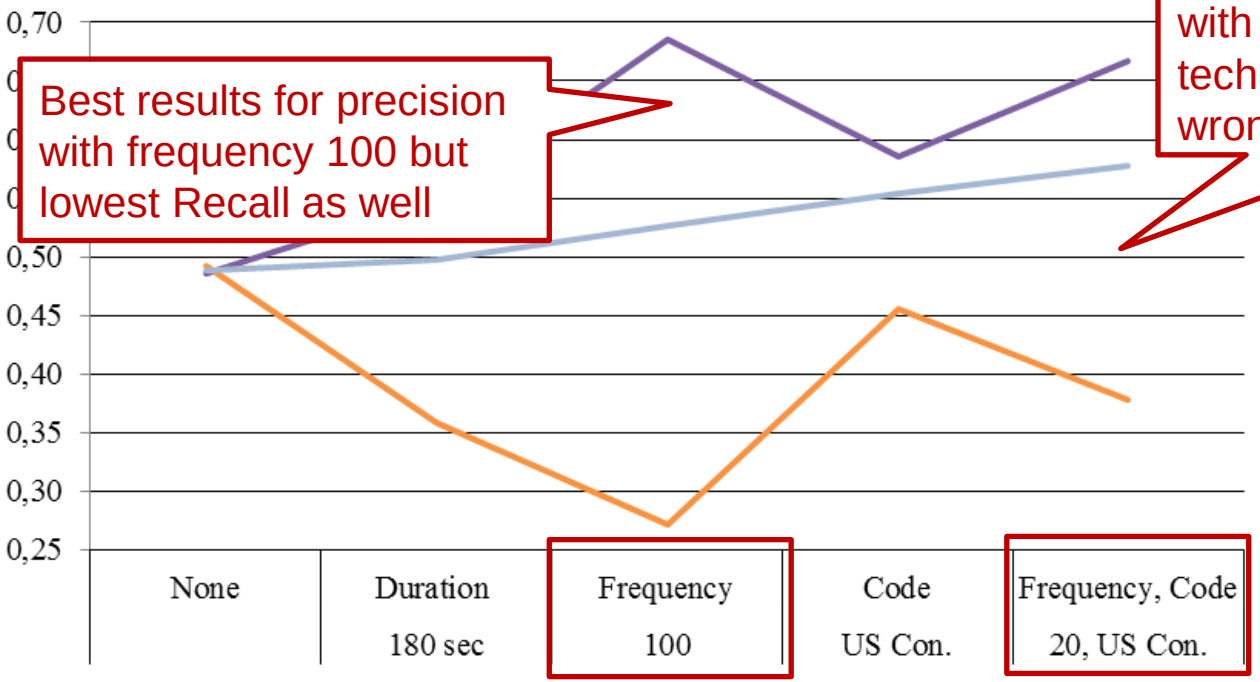
Use the source code structure for a user story

- Check if **source code files** are also **connected by** the **code structure**
- **Source Code Structure:** links to **connected code** files are **more likely** to be **correct**

RQ3: Results for Application of Wrong Link Detection Techniques Improve Precision



RQ3: Results Application of Wrong Link Detection Techniques Improve Precision

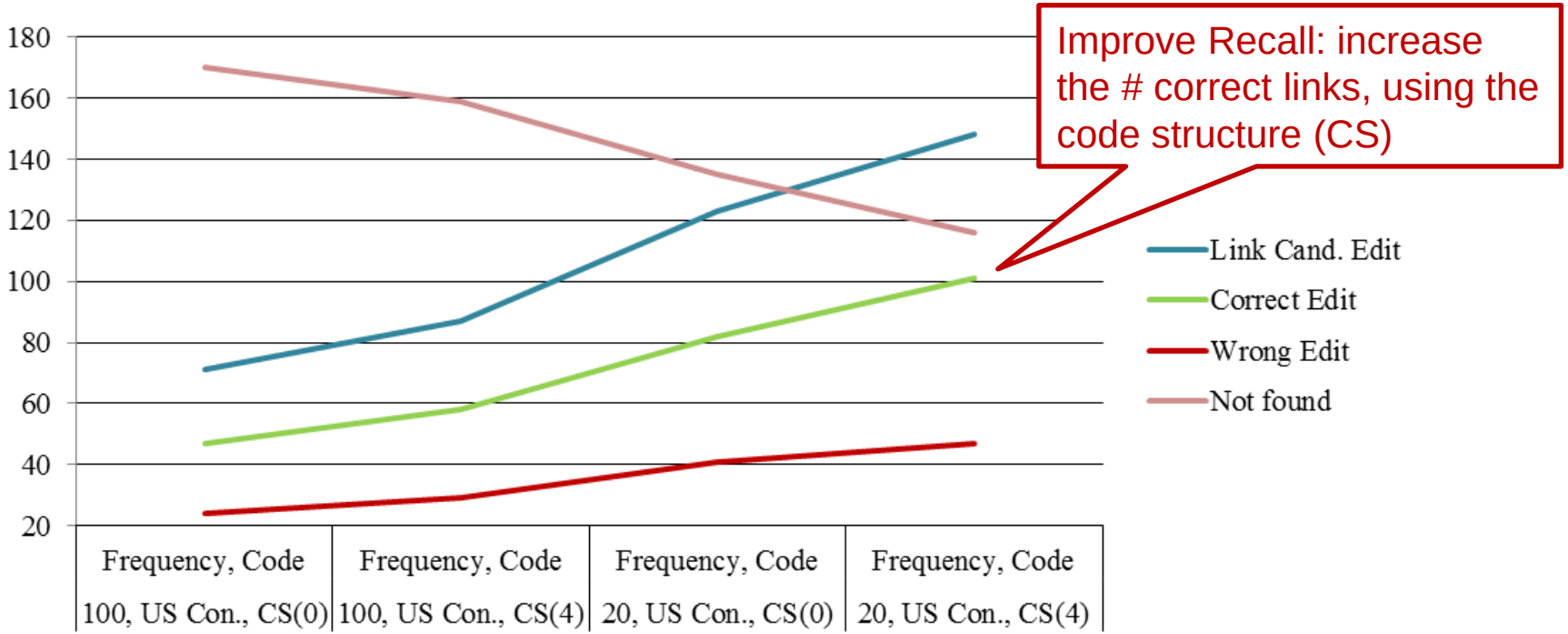


Best results for precision with frequency 100 but lowest Recall as well

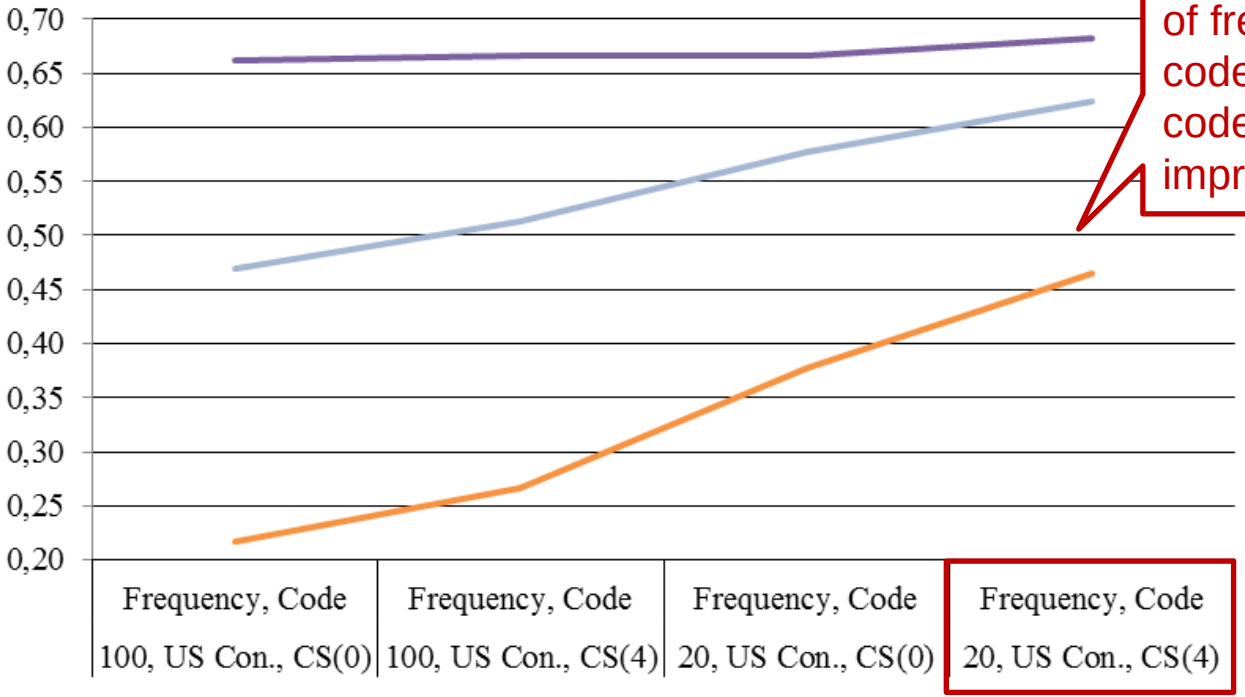
Best combined results with combination of techniques (filter out wrong but keep correct)

Precision and F0.5 measure to rate achieved improvement

RQ3: Results Application of Wrong Link Detection Techniques Improve Precision and Recall



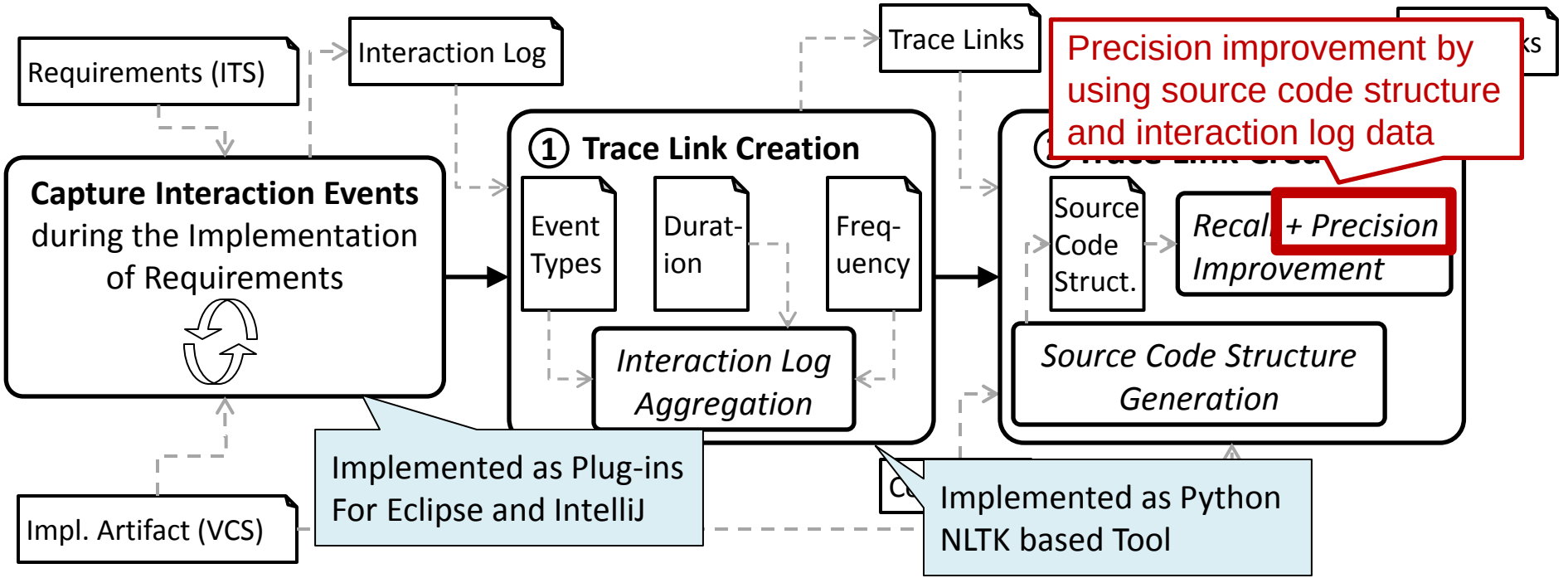
RQ3: Results Application of Wrong Link Detection Techniques Improve Precision and Recall



Best results for combination of freq. 20, connection of code files in user stories and code structure based recall improvement

Recall and F0.5 measure to rate achieved improvement and keep Precision

Extended IL Approach



Conclusion and Outlook

- IL performed insufficient in context of our student project
  - Developers have only little awareness of interaction logging effects
  - This might be the more common case in development projects

- Only frequency and removal of non-connected files improved the precision considerably up to almost 70%

- Further directions:
  - Support developers to apply interaction log recording more disciplined
  - Recommend the link candidates to the developers instead of automatic creation

Thank your for your attention!

Questions ... ???

Further Detailed Results

Interaction Log Recording and Gold Standard Creation

	Interaction logs			Gold standard creation					
	#Req.	#Interactions	#Code. Files	#Req.	#Code. Files	#Link Cand.	#Rated Correct	#Rated Wrong	#Rated Unknown
Dev_1	12	628.502	155	3	99	139	37	90	2
Dev_2	20	506.726	273	11	141	374	128	241	5
Dev_3	16	893.390	256	5	83	189	52	123	14
Sum	42[a]	2.028.618	312[a]	19	151[a]	692	217	454	21

[a]Same issues and source files used by different developers have been accumulated

Initial Application of IL and IR

Approach	GS links	Link Cand.	Correct links	Wrong links	Not found	Precision	Recall	$F_{0.5}$	F_1
IL	217	372	160	212	57	0.430	0.737	0.469	0.543
$IR_{VSM(0.3)}$	217	191	38	153	179	0.199	0.175	0.194	0.186
$IR_{VSM(0.2)}$	217	642	104	538	113	0.162	0.480	0.187	0.242
$IR_{LSI(0.1)}$	217	102	35	67	182	0.343	0.161	0.280	0.219
$IR_{LSI(0.05)}$	217	363	77	286	140	0.212	0.355	0.231	0.266

Wrong Link Detection: Duration Improvements

Dur. (sec)	GS links	Link Cand.		Correct		Wrong		Not found	Precision		Recall		$F_{0.5}$	F_1
		All	Edit	All	Edit	All	Edit		All	Edit	All	Edit		
1	217	372	220	160	107	212	113	57	0.430	0.486	0.737	0.493	0.488	0.490
10	217	317	199	144	104	173	95	73	0.454	0.523	0.664	0.479	0.513	0.500
60	217	231	167	113	90	118	77	104	0.489	0.539	0.521	0.415	0.508	0.469
180	217	183	142	93	78	90	64	124	0.508	0.549	0.429	0.359	0.497	0.435
300	217	154	122	81	70	73	52	136	0.526	0.574	0.373	0.323	0.496	0.413

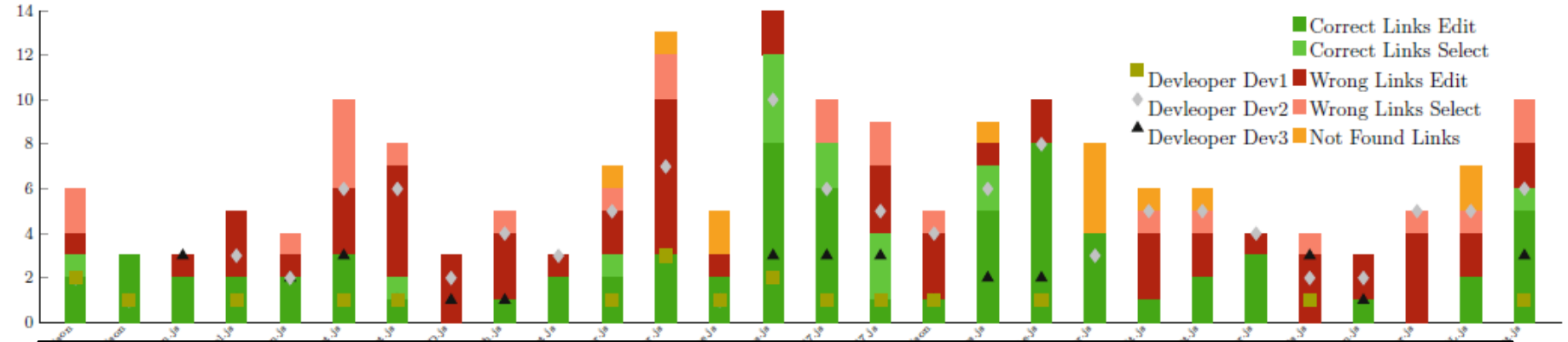
Wrong Link Detection: Frequency Improvements

Frequency	GS	Link Cand.		Correct		Wrong		Not	Precision		Recall		$F_{0.5}$	F_1
	links	All	Edit	All	Edit	All	Edit	found	All	Edit	All	Edit		
1	217	372	220	160	107	212	113	57	0.430	0.486	0.737	0.493	0.488	0.490
2	217	314	220	142	107	172	113	75	0.452	0.486	0.654	0.493	0.488	0.490
5	217	220	191	113	98	107	93	104	0.514	0.513	0.521	0.452	0.499	0.480
10	217	181	169	99	93	82	76	118	0.547	0.550	0.456	0.429	0.521	0.482
20	217	158	151	90	87	68	64	127	0.570	0.576	0.415	0.401	0.530	0.473
100	217	86	86	59	59	27	27	158	0.686	0.686	0.272	0.272	0.526	0.389

Wrong Link Detection: Developer Specific Differences

Developer	GS	Link Cand.		Correct		Wrong		Not	Precision		Recall		$F_{0.5}$	F_1
	links	All	Edit	All	Edit	All	Edit	found	All	Edit	All	Edit		
Dev_1	37	41	17	19	6	22	11	18	0.463	0.353	0.514	0.162	0.286	0.222
Dev_2	128	252	155	110	79	142	76	18	0.437	0.510	0.859	0.617	0.528	0.558
Dev_3	52	77	46	30	21	47	25	22	0.390	0.457	0.577	0.404	0.445	0.429

Results: Source code file occurrence and developer specific differences



Bar chart shows code files touched by >3 user stories

- 3 files with only wrong links (= bars with only red colors)
- #Developers touched a file differs (=marking in bars)
- Many link candidates → many correct links as well

→ No pattern

- **In addition:** developer specific precision is similar

Wrong Link Detection: Developer Specific Differences

Code Res.	GS links	Link Cand. All	Link Cand. Edit	Correct All	Correct Edit	Wrong All	Wrong Edit	Not found	Precision All	Precision Edit	Recall All	Recall Edit	$F_{0.5}$	F_1
none	217	372	220	160	107	212	113	57	0.430	0.486	0.737	0.493	0.488	0.490
>3 US	217	208	92	83	43	125	49	134	0.399	0.467	0.382	0.198	0.368	0.278
Only .js	186	327	203	129	99	198	104	57	0.394	0.488	0.694	0.532	0.496	0.509
Con.	217	274	169	147	99	127	70	70	0.536	0.586	0.677	0.456	0.554	0.513

Wrong Link Detection: Combination of Improvements

Code Con.	*Freq.*	*Code Struct*	GS links	Link Cand.		Correct		Wrong		Not found	Precision		Recall		$F_{0.5}$	F_1
				All	Edit	All	Edit	All	Edit		All	Edit	All	Edit		
True	*20*	*0*	217	124	123	82	82	42	41	135	0.661	0.667	0.378	0.378	0.578	0.482
True	*20*	*4*	217	151	148	101	101	50	47	116	0.669	0.682	0.465	0.465	0.624	0.553
True	*100*	*0*	217	71	71	47	47	24	24	170	0.662	0.662	0.217	0.217	0.469	0.326
True	*100*	*4*	217	87	87	58	58	29	29	159	0.667	0.667	0.267	0.267	0.513	0.382