ECE444: Software Engineering

Software Engineering Research

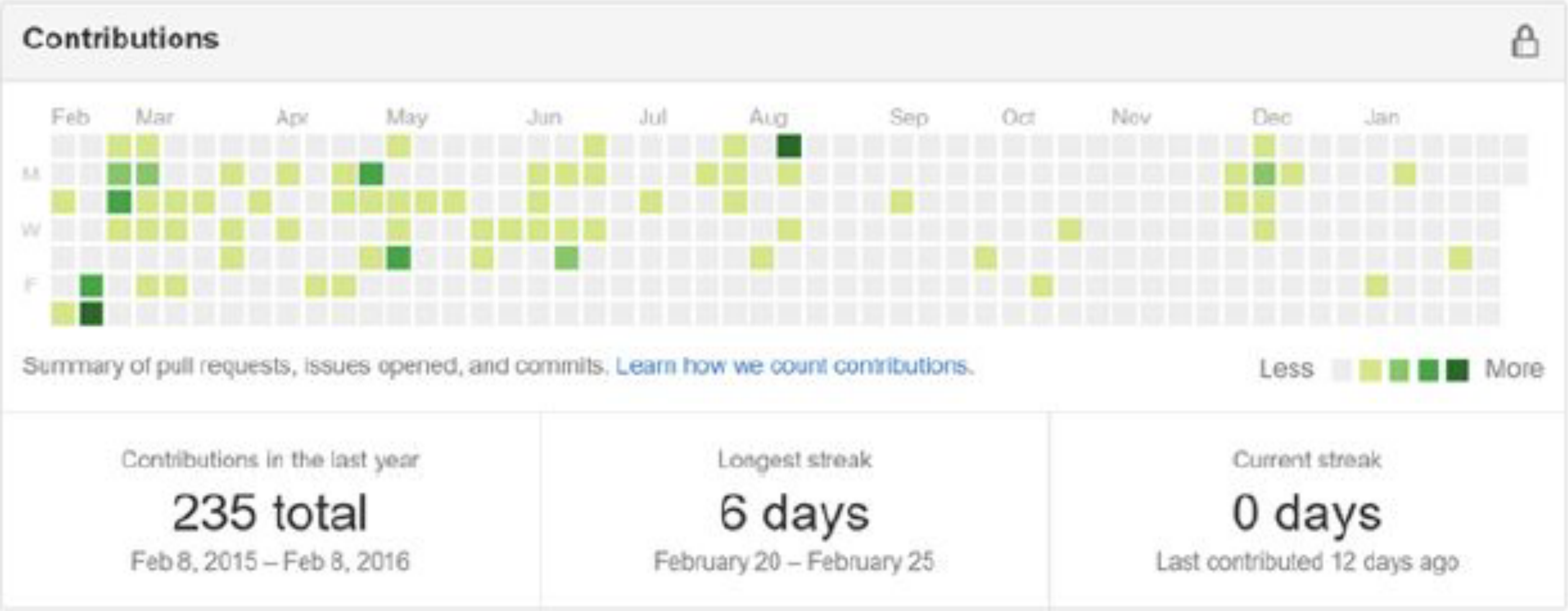
Shurui Zhou

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

(Competing) concerns in SE…

- **Code**: faster, cheaper, more features, more reliable/secure
- **Developers**: more productive, more skilled, happier, better connected
- **Organizations/communities**: attract/retain contributors, encourage a participatory culture, increase value

> “Measuring programming progress by lines of code is like measuring aircraft building progress by weight.”



Contributions

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Contributions in the last year

235 total

Feb 8, 2015 – Feb 8, 2016

Longest streak

6 days

February 20 – February 25

Current streak

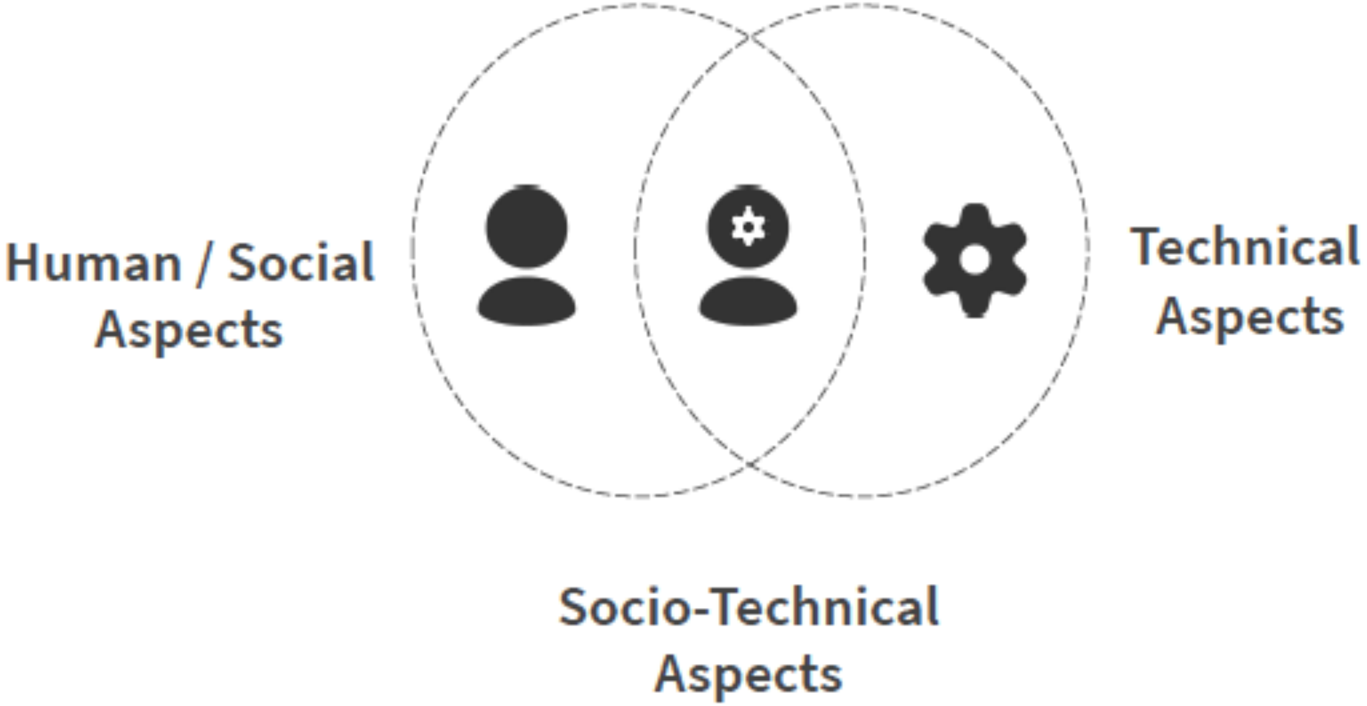
0 days

Last contributed 12 days ago

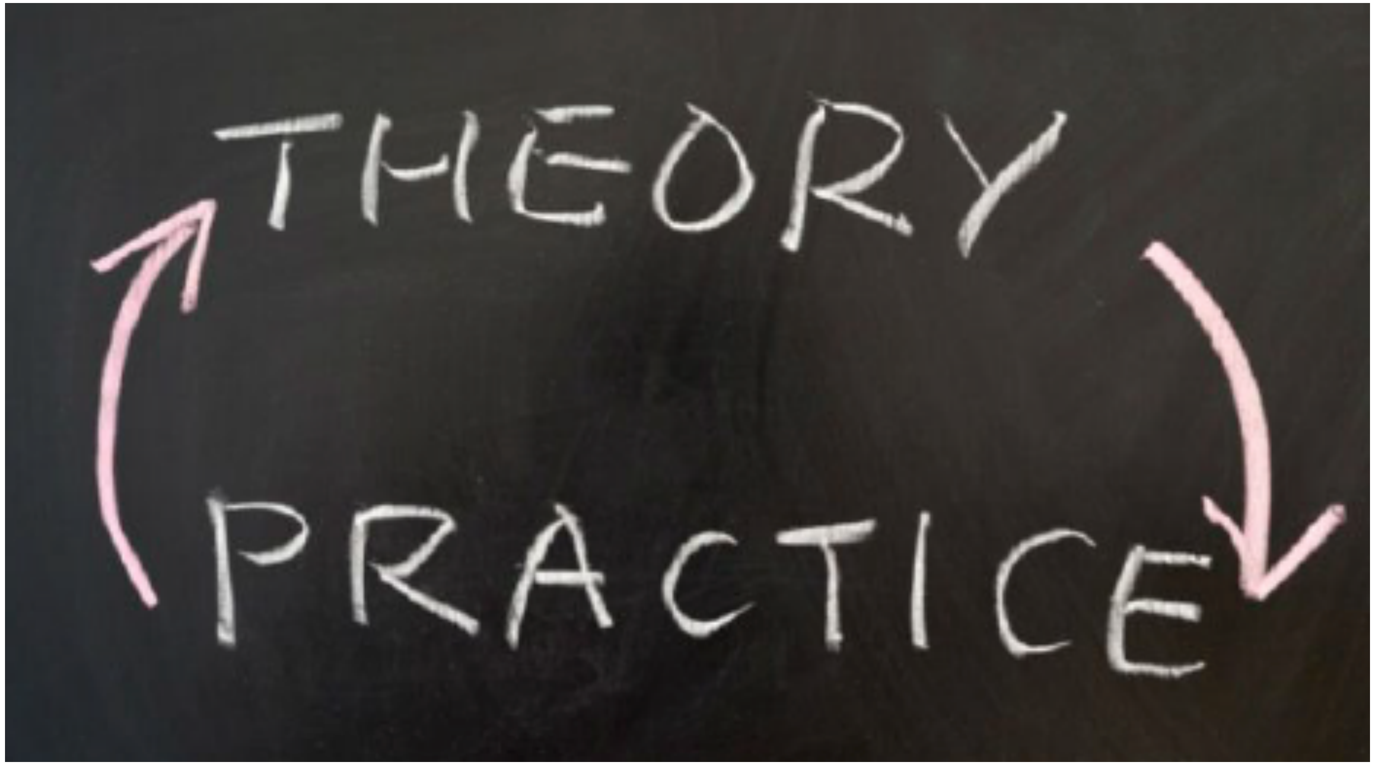
Contributing graphs considered harmful (Hanselman)

https://www.hanselman.com/

Software Engineering **Design** Space



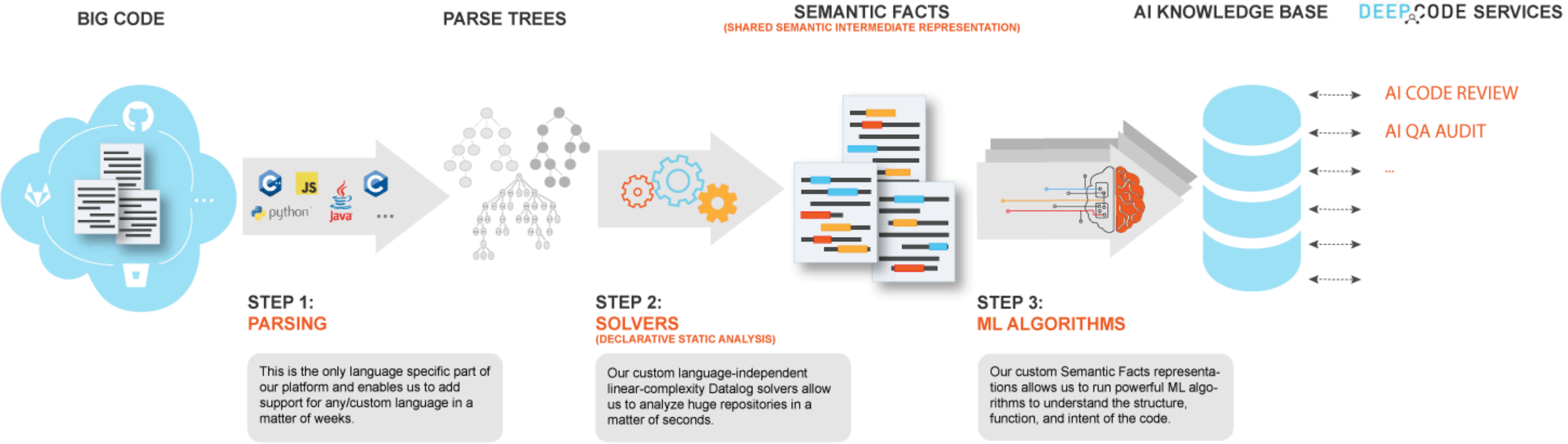
Research success?


THEORY
PRACTICE

Success practice transfer stories from research

- Automated testing (Facebook)
- Code review tools (Microsoft)
- Software Analytics (Hassan et al.)

DeepCode from ETH



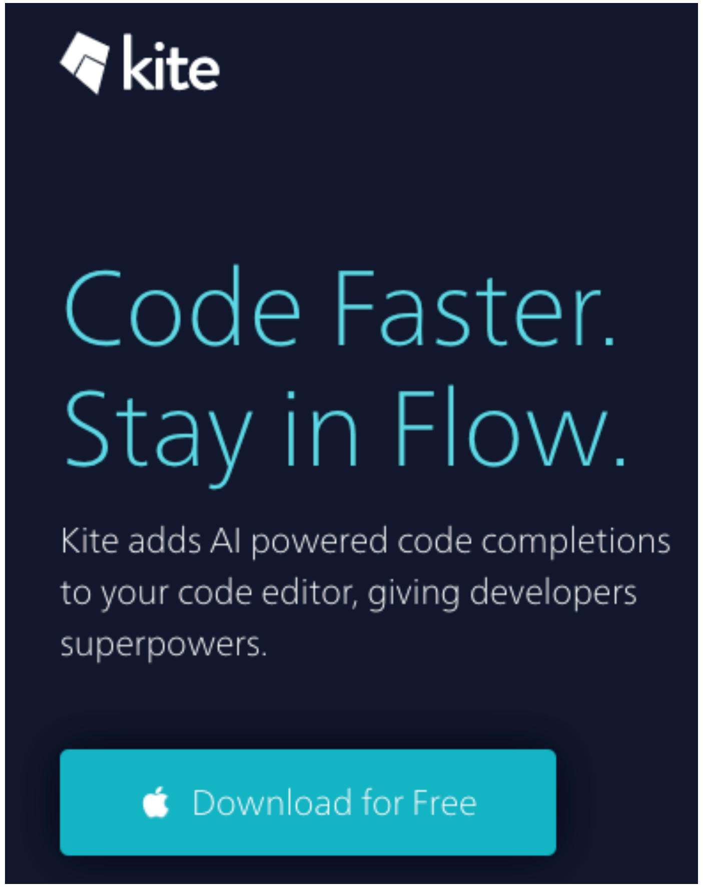
This is the only language specific part of our platform and enables us to add support for any/custom language in a matter of weeks.

Our custom language-independent linear-complexity Datalog solvers allow us to analyze huge repositories in a matter of seconds.

Our custom Semantic Facts representations allows us to run powerful ML algorithms to understand the structure, function, and intent of the code.

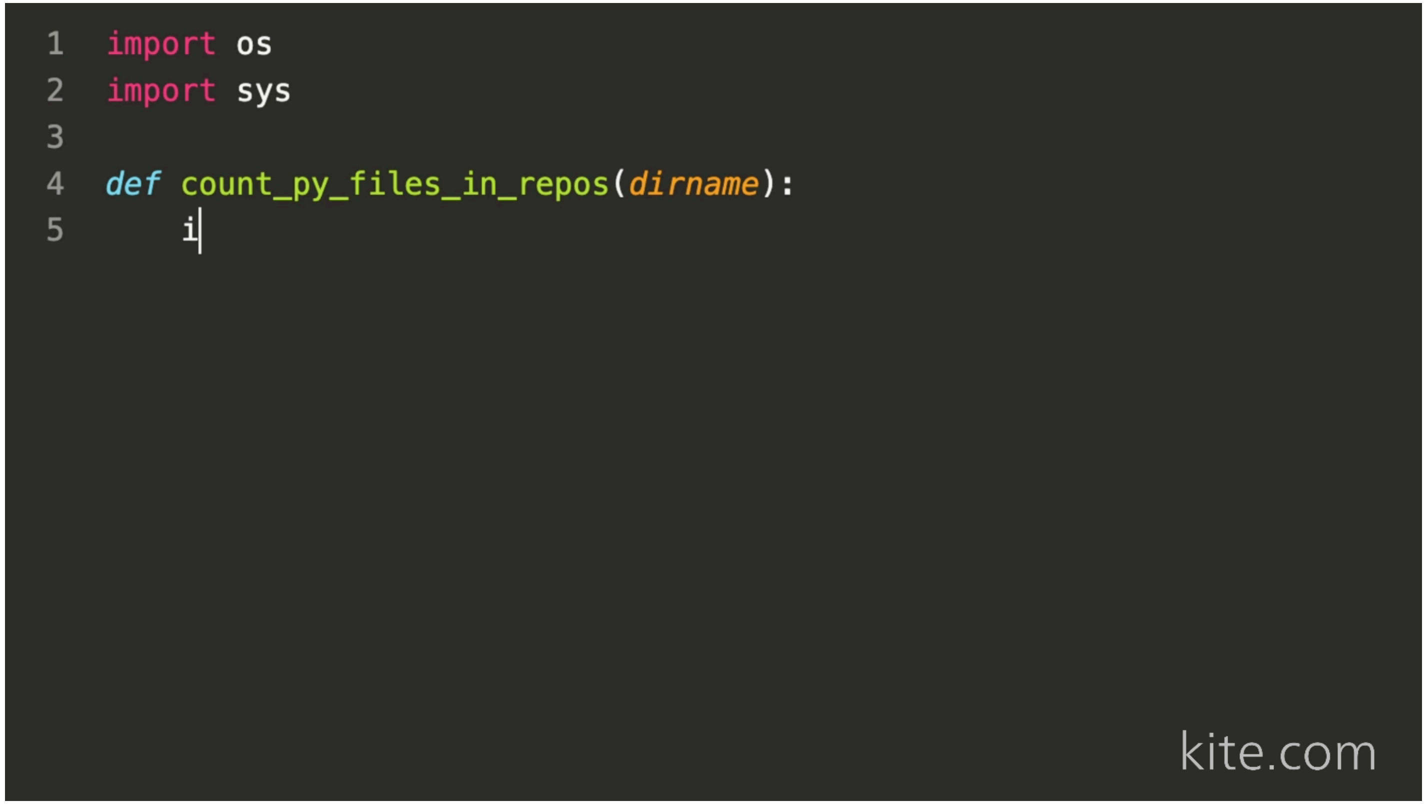
https://www.kite.com/

kite from MIT&Stanford



Code Faster.
Stay in Flow.

Kite adds AI powered code completions to your code editor, giving developers superpowers.



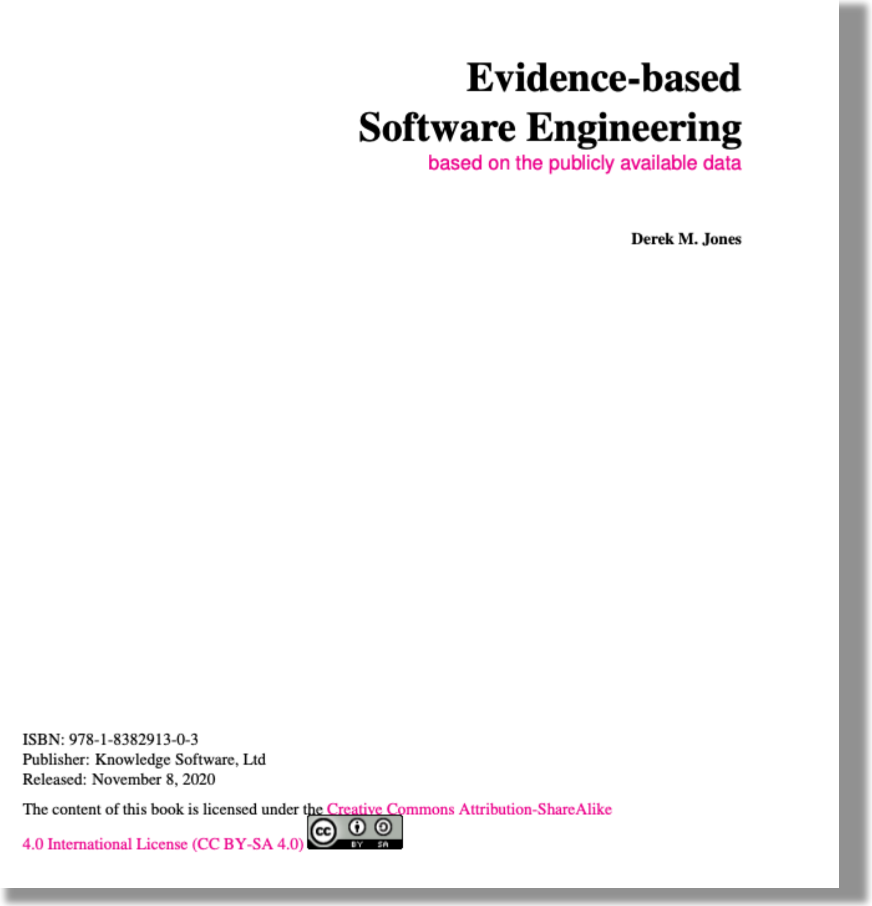
```python
import os
import sys

def count_py_files_in_repos(dirname):
    i
```

1968 NATO Conference on Software Engineering

- international experts on computer software who agreed on defining best practices for software grounded in the application of engineering.

"Academic software engineering research has been a backwater primarily staffed by those interested in theory, with a tenuous connection to practical software development."

- Lack of **industrial relevance** (doesn't scale or solve industry problems) [Briand]
- **Poor replication** of software engineering studies [Menzies et al.]
- **Poor actionability** (practitioners know which modules are buggy…)
- **Perils of mining** software repositories [Kaliamvakou, German et al.]
- Lack of focus on **human/social aspects** [Storey et al.]



Evidence-based
Software Engineering
based on the publicly available data

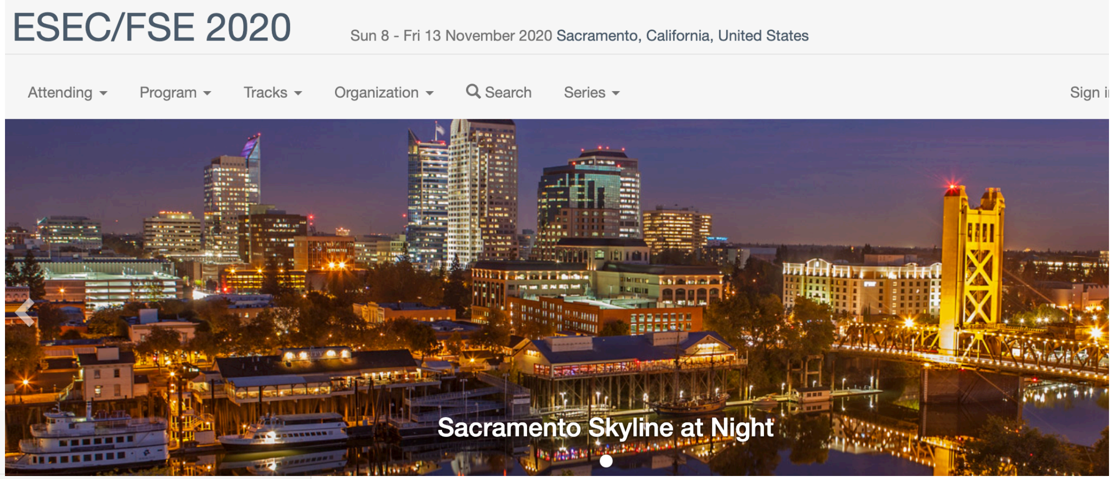
Derek M. Jones

ISBN: 978-1-8382913-0-3
Publisher: Knowledge Software, Ltd
Released: November 8, 2020

ICSE21

43rd INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING

MAY 23-29, 2021 MADRID, SPAIN

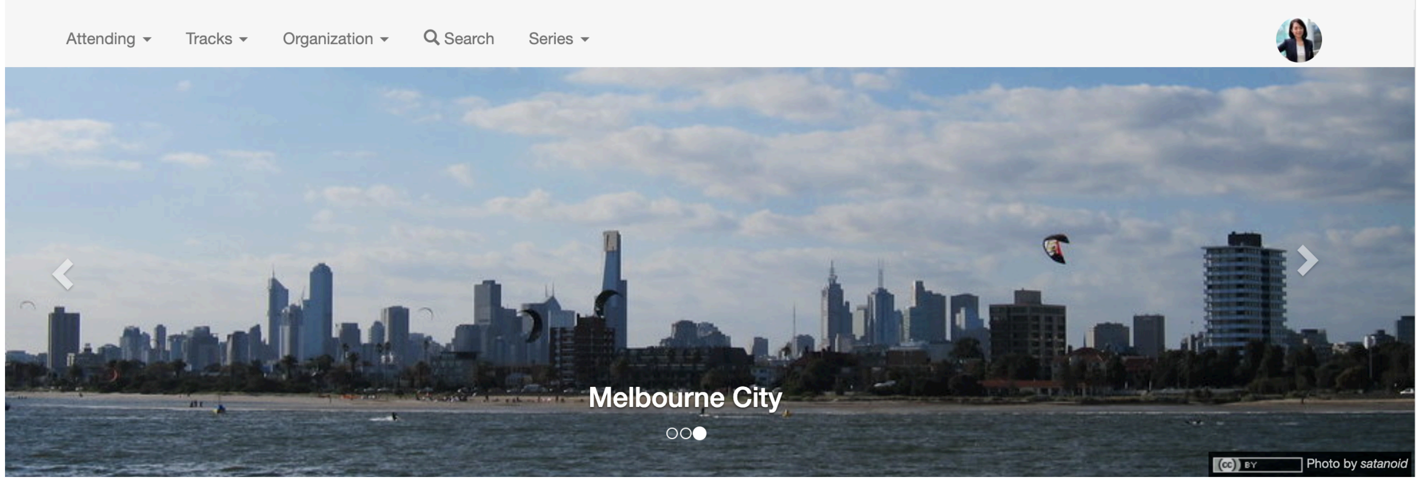


ESEC/FSE 2020

Sun 8 - Fri 13 November 2020 Sacramento, California, United States

ASE 2020

Mon 21 - Fri 25 September 2020 Melbourne, Australia

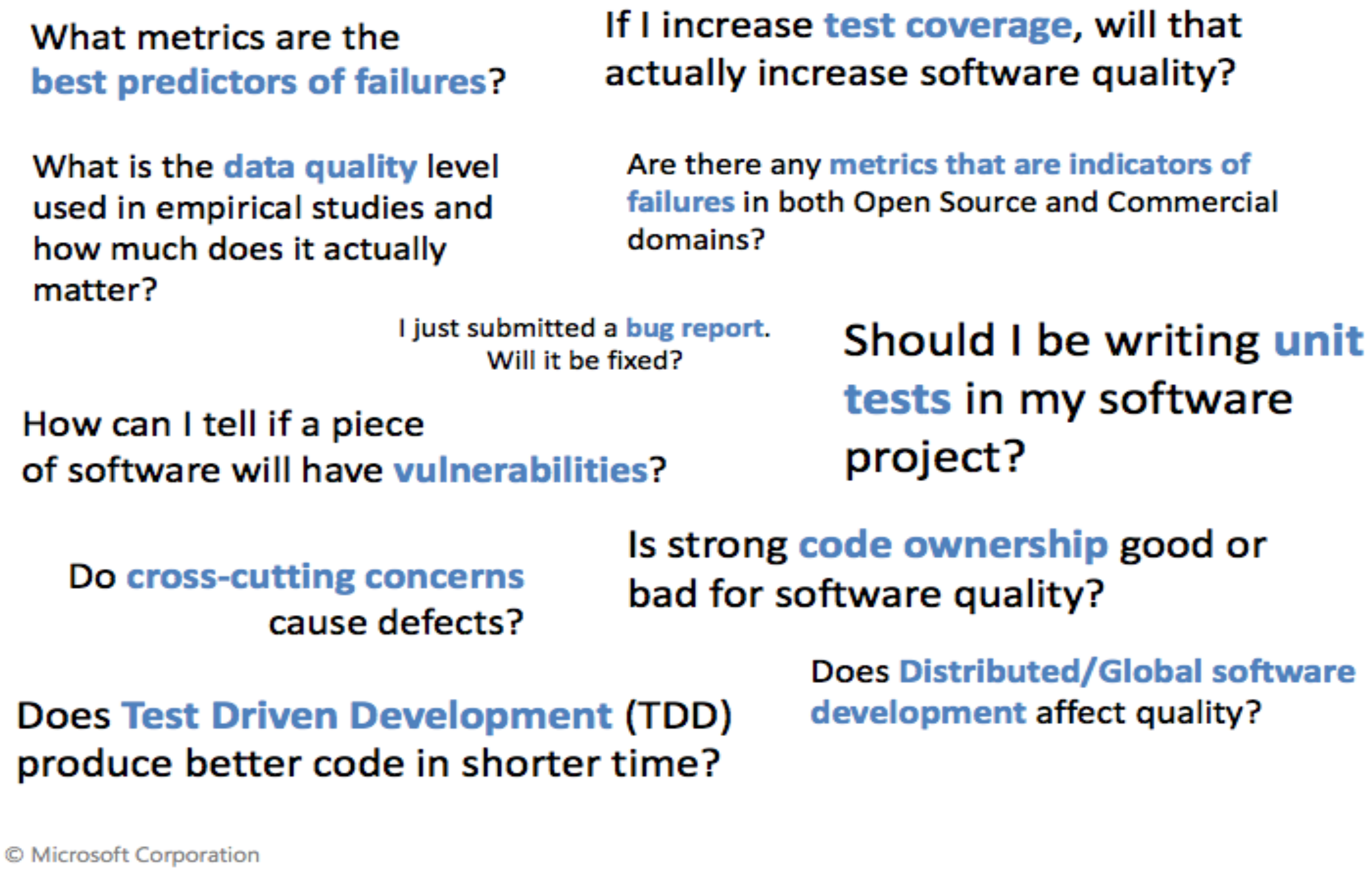


29th IEEE International Requirements Engineering Conference

Notre Dame, South Bend, USA
September 20-24, 2021

ISSTA 2020 Sat 18 - Wed 22 July 2020

MSR 2020 Mon 29 - Tue 30 June 2020



What metrics are the **best predictors of failures**?

If I increase **test coverage**, will that actually increase software quality?

What is the **data quality** level used in empirical studies and how much does it actually matter?

Are there any **metrics that are indicators of failures** in both Open Source and Commercial domains?

I just submitted a **bug report**. Will it be fixed?

Should I be writing **unit tests** in my software project?

How can I tell if a piece of software will have **vulnerabilities**?

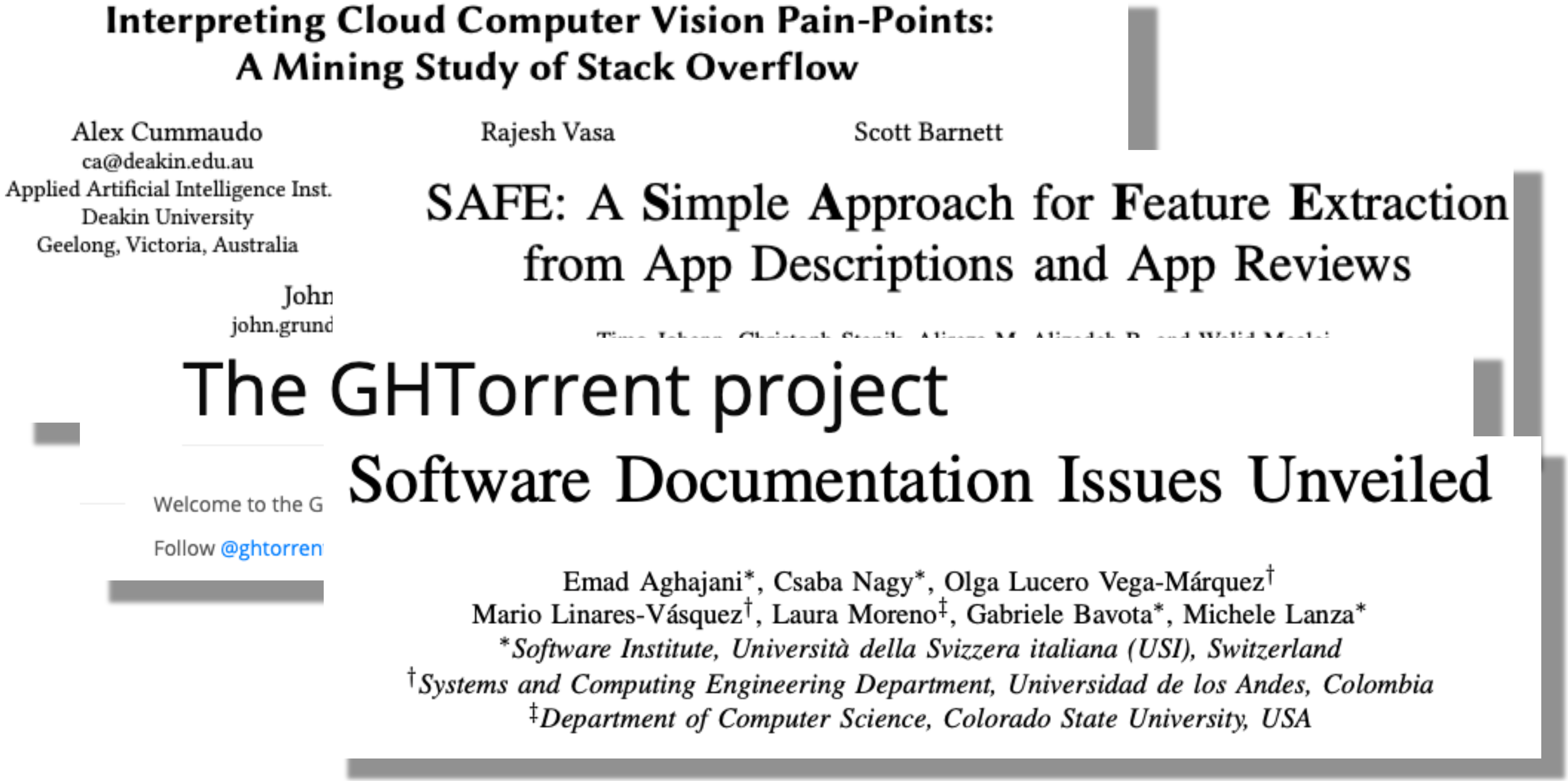
Do **cross-cutting concerns** cause defects?

Is strong **code ownership** good or bad for software quality?

Does **Distributed/Global software development** affect quality?

Does **Test Driven Development** (TDD) produce better code in shorter time?

Mining Software Repository



Interpreting Cloud Computer Vision Pain-Points: A Mining Study of Stack Overflow

Alex Cummaudo
ca@deakin.edu.au
Applied Artificial Intelligence Inst.
Deakin University
Geelong, Victoria, Australia

Rajesh Vasa

Scott Barnett

John
john.grund

SAFE: A Simple Approach for Feature Extraction from App Descriptions and App Reviews

Timo Johann, Christoph Stanik, Alireza M. Alizadeh B. and Walid Maalej

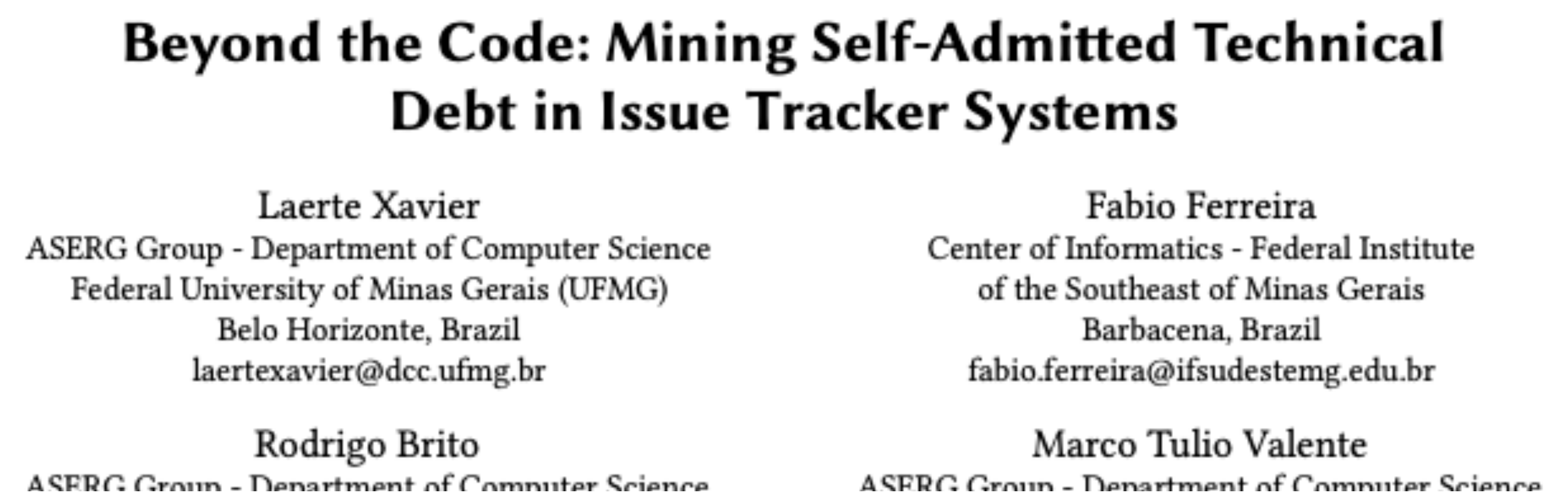
The GHTorrent project

Welcome to the G

Follow @ghtorren

Software Documentation Issues Unveiled

Emad Aghajani*, Csaba Nagy*, Olga Lucero Vega-Márquez[†]
Mario Linares-Vásquez[†], Laura Moreno[‡], Gabriele Bavota*, Michele Lanza*
*Software Institute, Università della Svizzera italiana (USI), Switzerland
[†]Systems and Computing Engineering Department, Universidad de los Andes, Colombia
[‡]Department of Computer Science, Colorado State University, USA



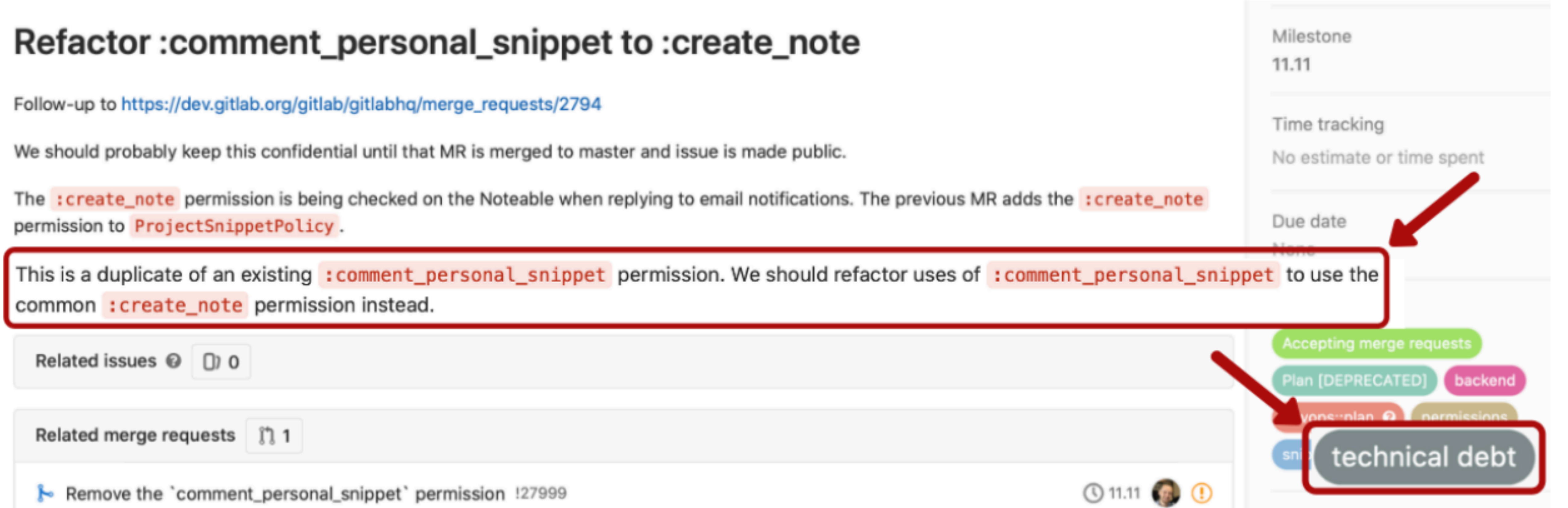
Beyond the Code: Mining Self-Admitted Technical Debt in Issue Tracker Systems

Laerte Xavier
ASERG Group - Department of Computer Science
Federal University of Minas Gerais (UFMG)
Belo Horizonte, Brazil
laertexavier@dcc.ufmg.br

Fabio Ferreira
Center of Informatics - Federal Institute
of the Southeast of Minas Gerais
Barbacena, Brazil
fabio.ferreira@ifsudestemg.edu.br

Rodrigo Brito
ASERG Group - Department of Computer Science

Marco Tulio Valente
ASERG Group - Department of Computer Science



Refactor :comment_personal_snippet to :create_note

Follow-up to https://dev.gitlab.org/gitlab/gitlabhq/merge_requests/2794

We should probably keep this confidential until that MR is merged to master and issue is made public.

The `:create_note` permission is being checked on the Noteable when replying to email notifications. The previous MR adds the `:create_note` permission to `ProjectSnippetPolicy`.

This is a duplicate of an existing `:comment_personal_snippet` permission. We should refactor uses of `:comment_personal_snippet` to use the common `:create_note` permission instead.

Related issues 0

Related merge requests 1

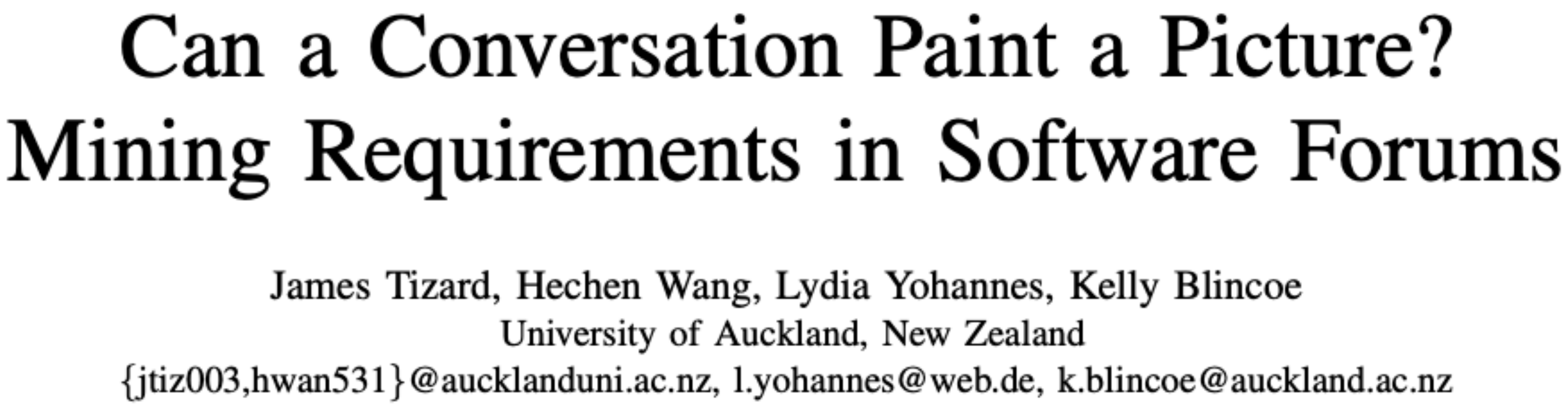
Remove the `comment_personal_snippet` permission !27999 11.11

Milestone
11.11

Time tracking
No estimate or time spent

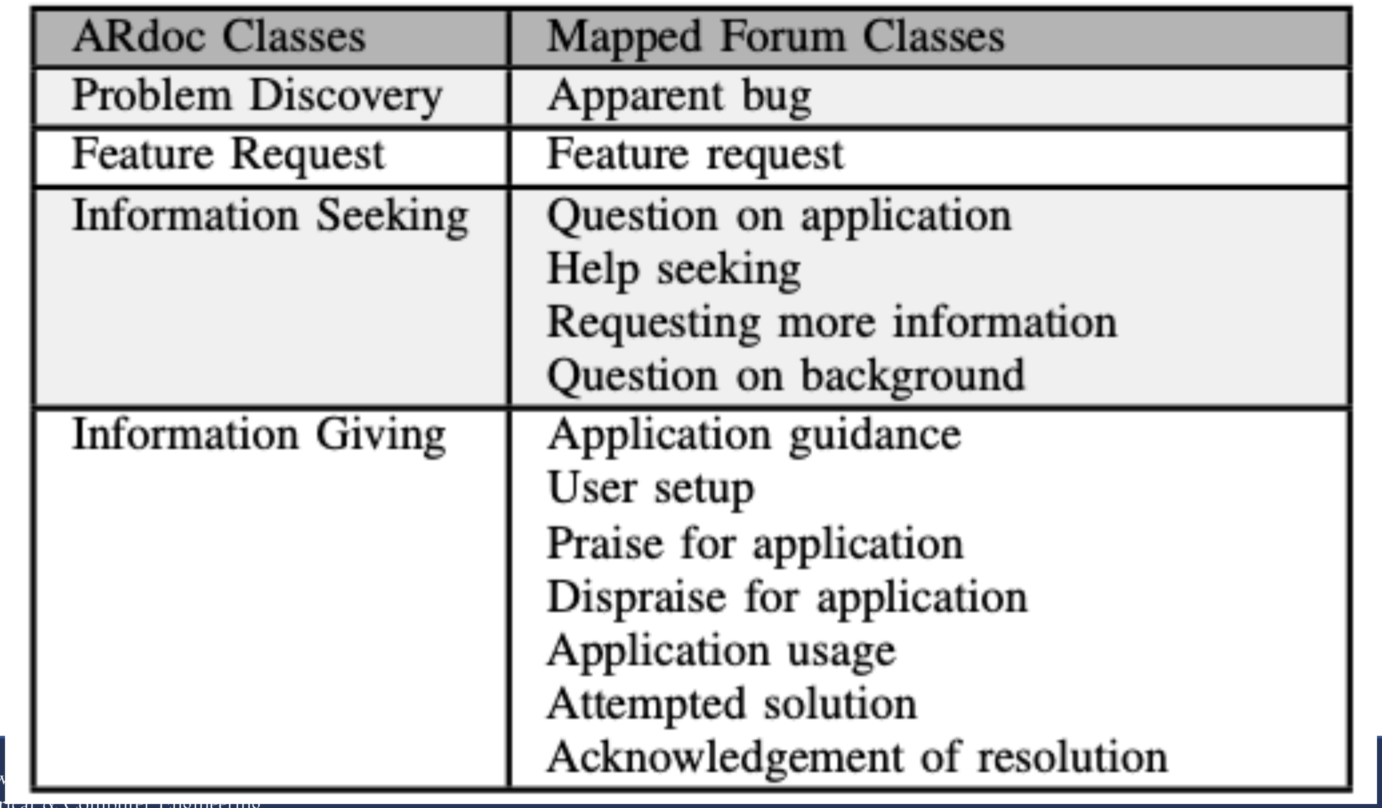
Due date
None

Requirement



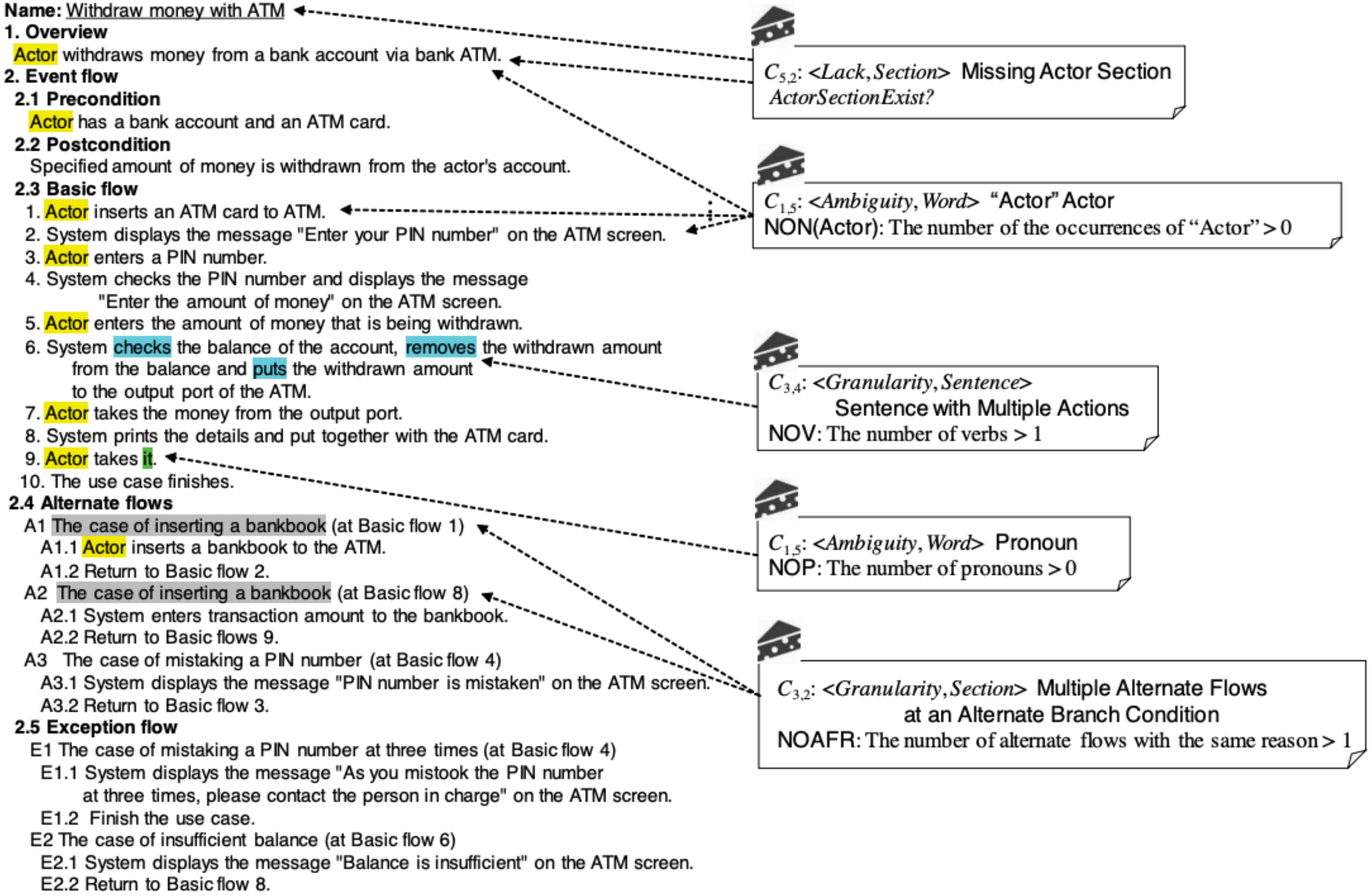
Can a Conversation Paint a Picture? Mining Requirements in Software Forums

James Tizard, Hechen Wang, Lydia Yohannes, Kelly Blincoe
University of Auckland, New Zealand
{jtiz003,hwan531}@aucklanduni.ac.nz, l.yohannes@web.de, k.blincoe@auckland.ac.nz



ARdoc Classes	Mapped Forum Classes
Problem Discovery	Apparent bug
Feature Request	Feature request
Information Seeking	Question on application Help seeking Requesting more information Question on background
Information Giving	Application guidance User setup Praise for application Dispraise for application Application usage Attempted solution Acknowledgement of resolution

Detecting Bad Smells in Use Case Descriptions



Name: Withdraw money with ATM
1. Overview
Actor withdraws money from a bank account via bank ATM.
2. Event flow
2.1 Precondition
Actor has a bank account and an ATM card.
2.2 Postcondition
Specified amount of money is withdrawn from the actor's account.
2.3 Basic flow
1. Actor inserts an ATM card to ATM.
2. System displays the message "Enter your PIN number" on the ATM screen.
3. Actor enters a PIN number.
4. System checks the PIN number and displays the message "Enter the amount of money" on the ATM screen.
5. Actor enters the amount of money that is being withdrawn.
6. System checks the balance of the account, removes the withdrawn amount from the balance and puts the withdrawn amount to the output port of the ATM.
7. Actor takes the money from the output port.
8. System prints the details and put together with the ATM card.
9. Actor takes it.
10. The use case finishes.

2.4 Alternate flows
A1 The case of inserting a bankbook (at Basic flow 1)
A1.1 Actor inserts a bankbook to the ATM.
A1.2 Return to Basic flow 2.
A2 The case of inserting a bankbook (at Basic flow 8)
A2.1 System enters transaction amount to the bankbook.
A2.2 Return to Basic flows 9.
A3 The case of mistaking a PIN number (at Basic flow 4)
A3.1 System displays the message "PIN number is mistaken" on the ATM screen.
A3.2 Return to Basic flow 3.
2.5 Exception flow
E1 The case of mistaking a PIN number at three times (at Basic flow 4)
E1.1 System displays the message "As you mistook the PIN number at three times, please contact the person in charge" on the ATM screen.
E1.2 Finish the use case.
E2 The case of insufficient balance (at Basic flow 6)
E2.1 System displays the message "Balance is insufficient" on the ATM screen.
E2.2 Return to Basic flow 8.

$C_{5,2}$: <*Lack*, *Section*> Missing Actor Section
ActorSectionExist?

$C_{1,5}$: <*Ambiguity*, *Word*> "Actor" Actor
NON(Actor): The number of the occurrences of "Actor" > 0

$C_{3,4}$: <*Granularity*, *Sentence*> Sentence with Multiple Actions
NOV: The number of verbs > 1

$C_{1,5}$: <*Ambiguity*, *Word*> Pronoun
NOP: The number of pronouns > 0

$C_{3,2}$: <*Granularity*, *Section*> Multiple Alternate Flows at an Alternate Branch Condition
NOAFR: The number of alternate flows with the same reason > 1

Fig. 6. Example result of the smell detection.

NERO: A Text-based Tool for Content Annotation and Detection of Smells in Feature Requests

Fangwen Mu[1,2], Lin Shi[1,2]*, Wei Zhou[1], Yuanzhong Zhang[1], Huixia Zhao[1]

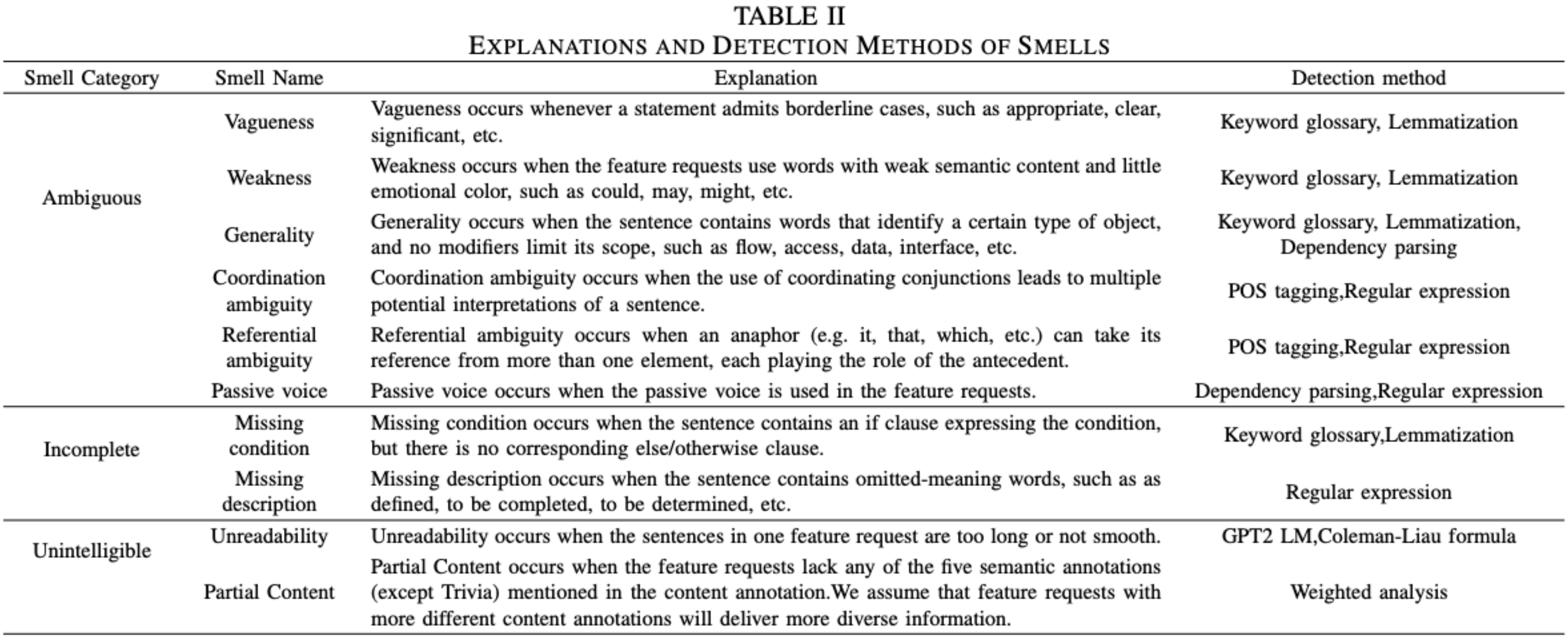
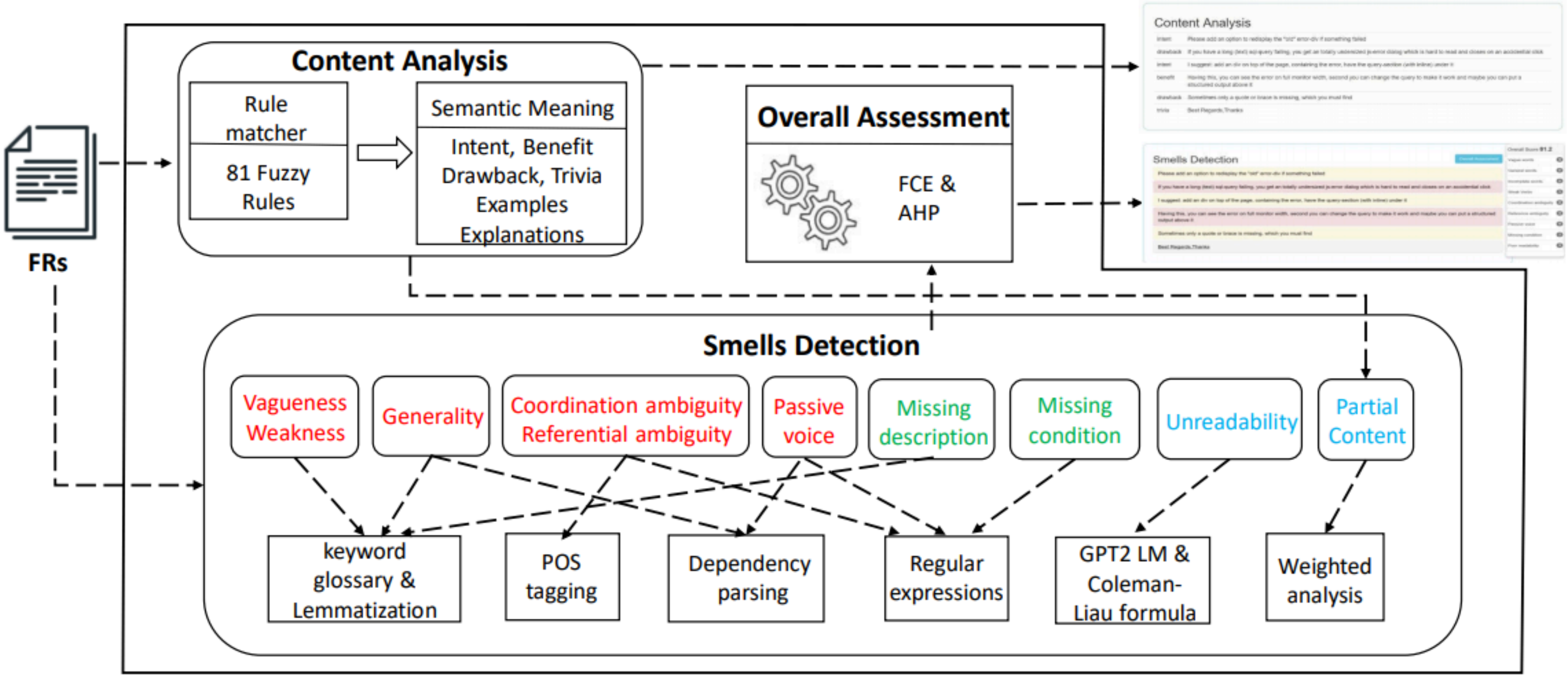


TABLE II
EXPLANATIONS AND DETECTION METHODS OF SMELLS

Smell Category	Smell Name	Explanation	Detection method
Ambiguous	Vagueness	Vagueness occurs whenever a statement admits borderline cases, such as appropriate, clear, significant, etc.	Keyword glossary, Lemmatization
	Weakness	Weakness occurs when the feature requests use words with weak semantic content and little emotional color, such as could, may, might, etc.	Keyword glossary, Lemmatization
	Generality	Generality occurs when the sentence contains words that identify a certain type of object, and no modifiers limit its scope, such as flow, access, data, interface, etc.	Keyword glossary, Lemmatization, Dependency parsing
	Coordination ambiguity	Coordination ambiguity occurs when the use of coordinating conjunctions leads to multiple potential interpretations of a sentence.	POS tagging,Regular expression
	Referential ambiguity	Referential ambiguity occurs when an anaphor (e.g. it, that, which, etc.) can take its reference from more than one element, each playing the role of the antecedent.	POS tagging,Regular expression
	Passive voice	Passive voice occurs when the passive voice is used in the feature requests.	Dependency parsing,Regular expression
Incomplete	Missing condition	Missing condition occurs when the sentence contains an if clause expressing the condition, but there is no corresponding else/otherwise clause.	Keyword glossary,Lemmatization
	Missing description	Missing description occurs when the sentence contains omitted-meaning words, such as as defined, to be completed, to be determined, etc.	Regular expression
Unintelligible	Unreadability	Unreadability occurs when the sentences in one feature request are too long or not smooth.	GPT2 LM,Coleman-Liau formula
	Partial Content	Partial Content occurs when the feature requests lack any of the five semantic annotations (except Trivia) mentioned in the content annotation.We assume that feature requests with more different content annotations will deliver more diverse information.	Weighted analysis

NERO: A Text-based Tool for Content Annotation and Detection of Smells in Feature Requests

Fangwen Mu[1,2], Lin Shi[1,2*], Wei Zhou[1], Yuanzhong Zhang[1], Huixia Zhao[1]
[1]Laboratory for Internet Software Technologies, Institute of Software Chinese Academy of Sciences, Beijing, China.
[2]University of Chinese Academy of Sciences, Beijing, China.
{fangwen, zhouwei, yuanzhong, huixia}@itechs.iscas.ac.cn, {shilin}@iscas.ac.cn



Content Analysis

intent	Please add an option to redisplay the "old" error-div if something failed
drawback	If you have a long (text) sql-query failing, you get an totally undersized js-error dialog which is hard to read and closes on an accidental click
intent	I suggest: add an div on top of the page, containing the error, have the query-section (with inline) under it
benefit	Having this, you can see the error on full monitor width, second you can change the query to make it work and maybe you can put a structured output above it
drawback	Sometimes only a quote or brace is missing, which you must find
trivia	Best Regards,Thanks

Smells Detection

Please add an option to redisplay the "old" error-div if something failed

If you have a long (text) sql-query failing, you get an totally undersized js-error dialog which is hard to read and closes on an accidental click

I suggest: add an div on top of the page, containing the error, have the query-section (with inline) under it

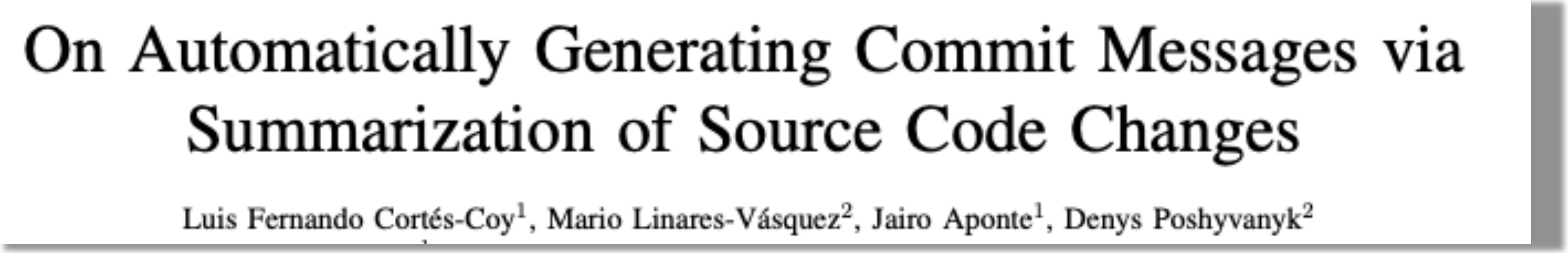
Having this, you can see the error on full monitor width, second you can change the query to make it work and maybe you can put a structured output above it

Sometimes only a quote or brace is missing, which you must find

Best Regards,Thanks

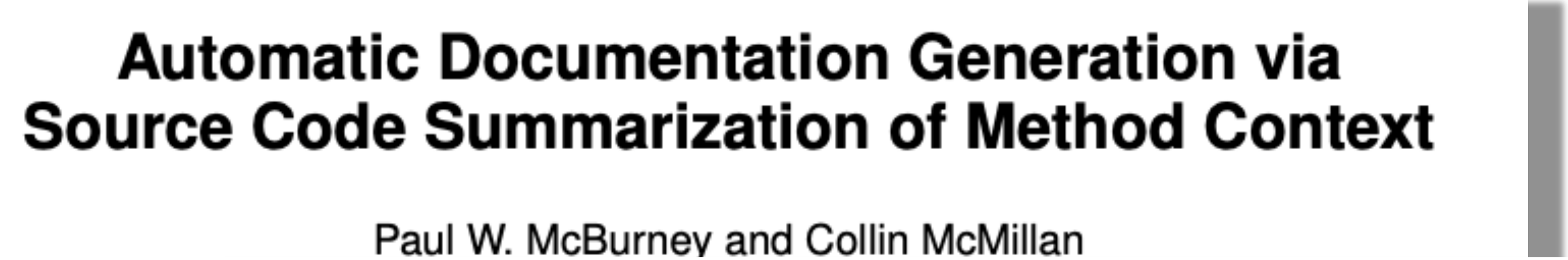
Fig. 1. The Overview of *NERO*

Documentation



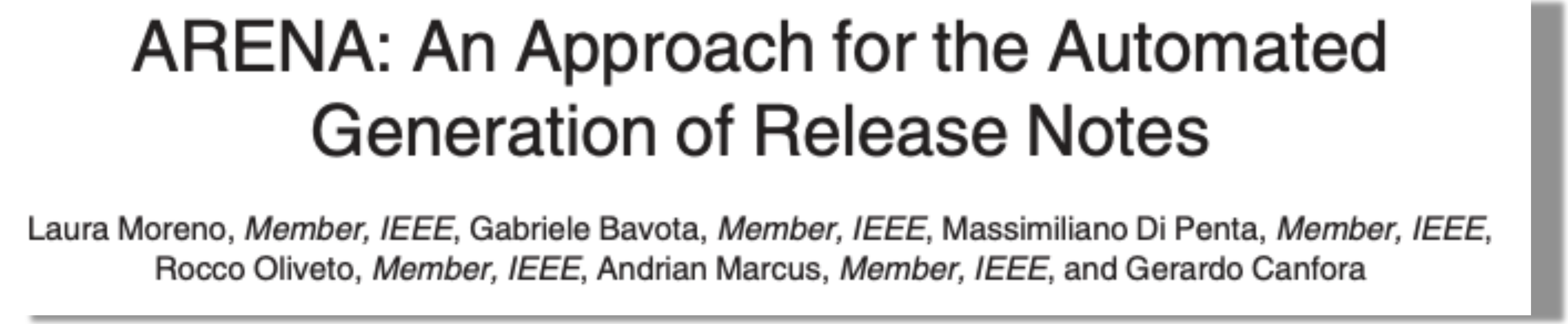
On Automatically Generating Commit Messages via Summarization of Source Code Changes

Luis Fernando Cortés-Coy[1], Mario Linares-Vásquez[2], Jairo Aponte[1], Denys Poshyvanyk[2]



Automatic Documentation Generation via Source Code Summarization of Method Context

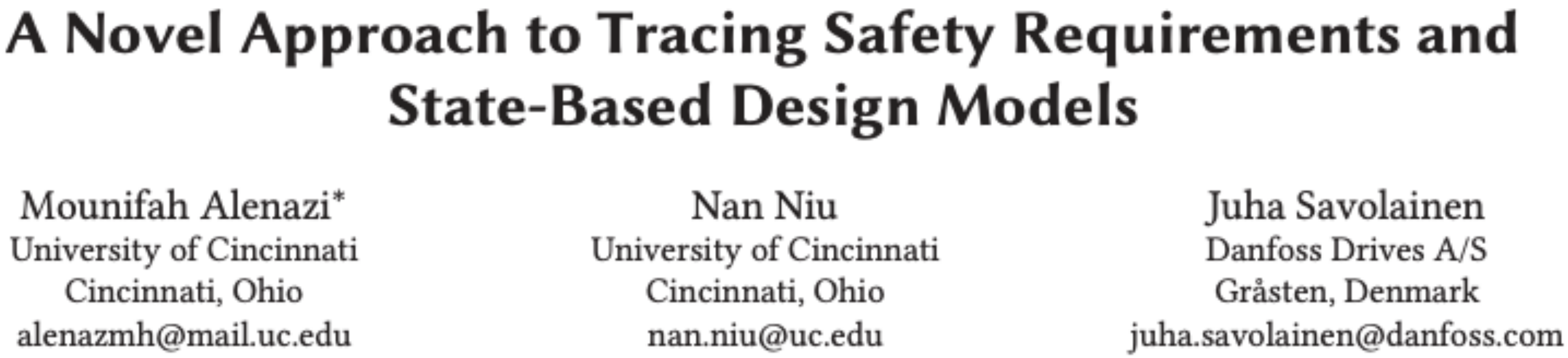
Paul W. McBurney and Collin McMillan



ARENA: An Approach for the Automated Generation of Release Notes

Laura Moreno, *Member, IEEE*, Gabriele Bavota, *Member, IEEE*, Massimiliano Di Penta, *Member, IEEE*, Rocco Oliveto, *Member, IEEE*, Andrian Marcus, *Member, IEEE*, and Gerardo Canfora

Traceability

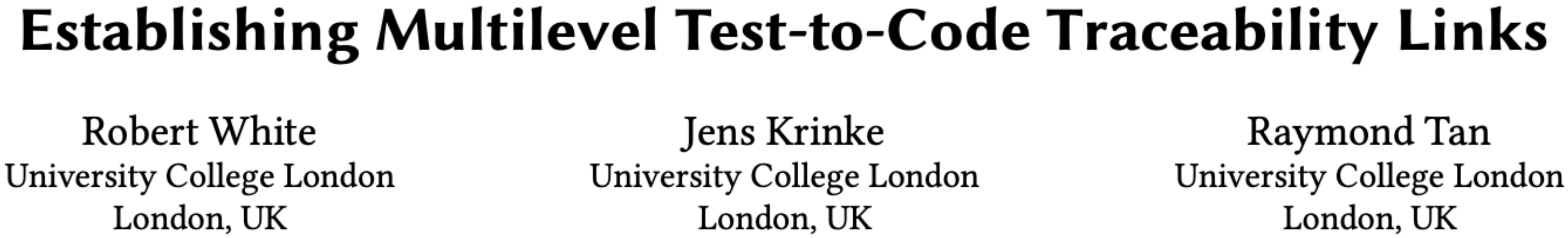


A Novel Approach to Tracing Safety Requirements and State-Based Design Models

Mounifah Alenazi*
University of Cincinnati
Cincinnati, Ohio
alenazmh@mail.uc.edu

Nan Niu
University of Cincinnati
Cincinnati, Ohio
nan.niu@uc.edu

Juha Savolainen
Danfoss Drives A/S
Gråsten, Denmark
juha.savolainen@danfoss.com

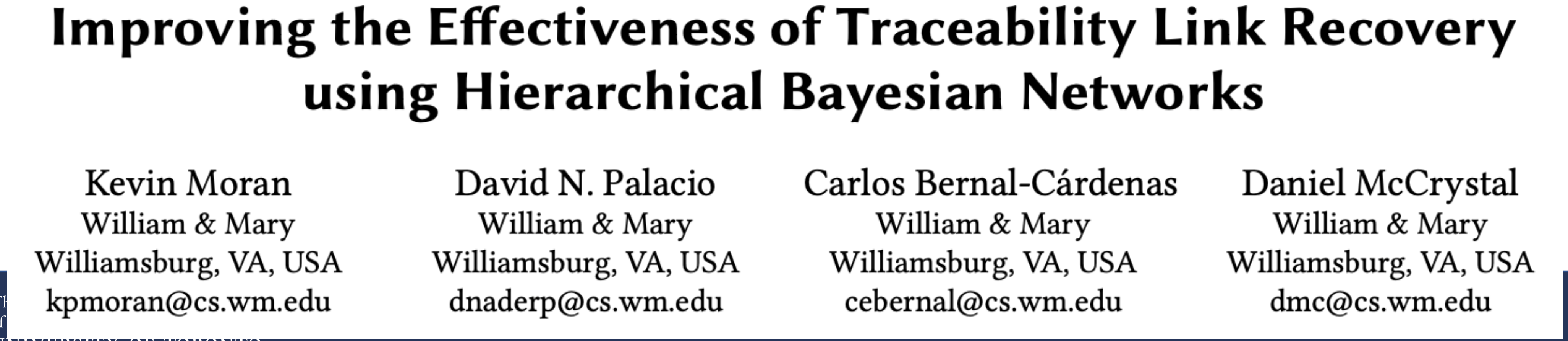


Establishing Multilevel Test-to-Code Traceability Links

Robert White
University College London
London, UK

Jens Krinke
University College London
London, UK

Raymond Tan
University College London
London, UK



Improving the Effectiveness of Traceability Link Recovery using Hierarchical Bayesian Networks

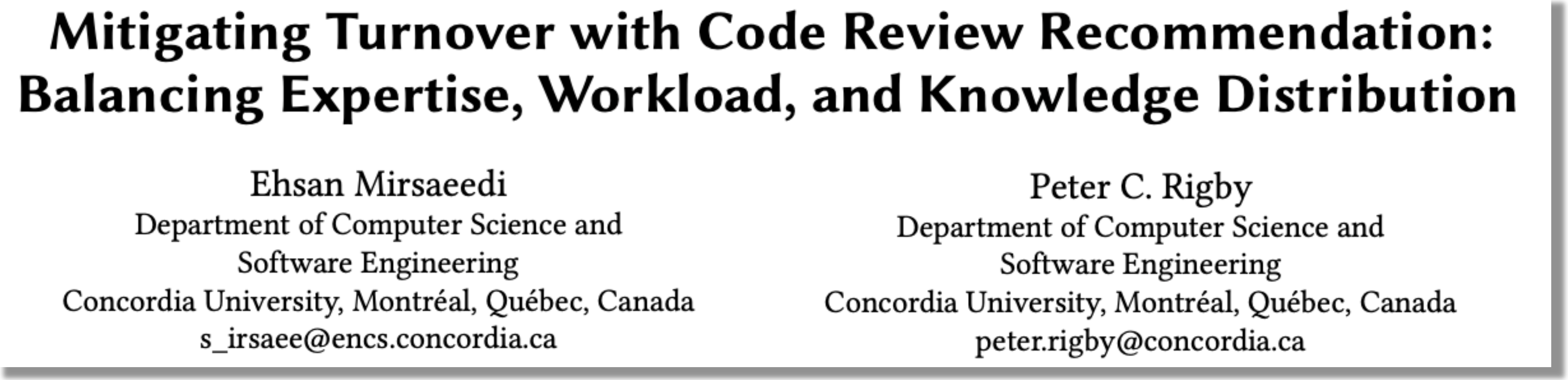
Kevin Moran
William & Mary
Williamsburg, VA, USA
kpmoran@cs.wm.edu

David N. Palacio
William & Mary
Williamsburg, VA, USA
dnaderp@cs.wm.edu

Carlos Bernal-Cárdenas
William & Mary
Williamsburg, VA, USA
cebernal@cs.wm.edu

Daniel McCrystal
William & Mary
Williamsburg, VA, USA
dmc@cs.wm.edu

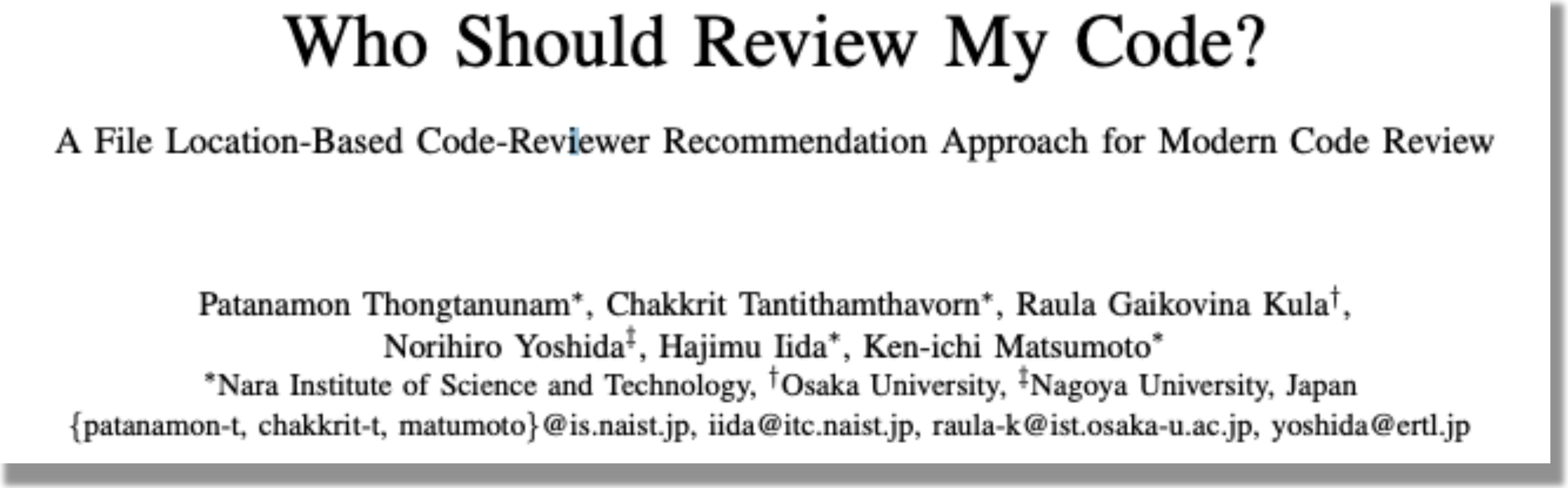
Code Review



Mitigating Turnover with Code Review Recommendation: Balancing Expertise, Workload, and Knowledge Distribution

Ehsan Mirsaeedi
Department of Computer Science and
Software Engineering
Concordia University, Montréal, Québec, Canada
s_irsaee@encs.concordia.ca

Peter C. Rigby
Department of Computer Science and
Software Engineering
Concordia University, Montréal, Québec, Canada
peter.rigby@concordia.ca

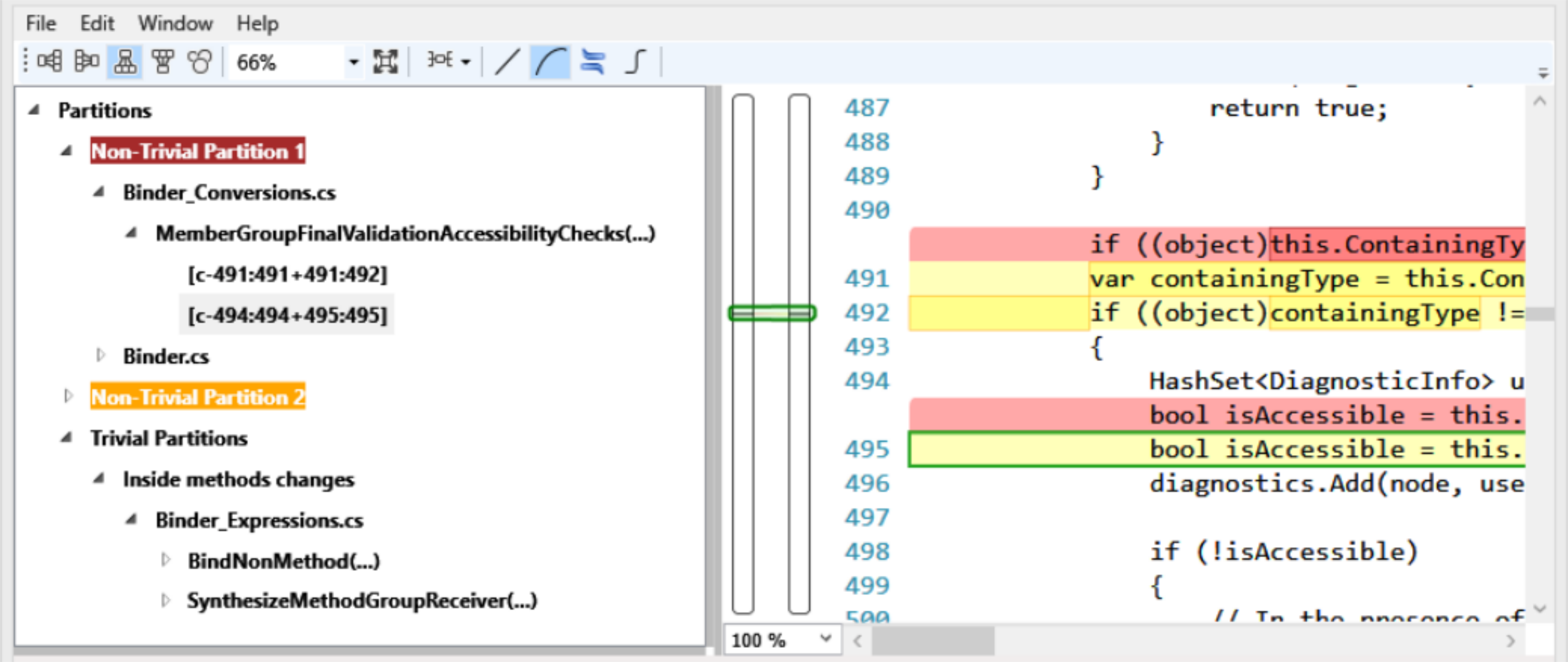


Who Should Review My Code?

A File Location-Based Code-Reviewer Recommendation Approach for Modern Code Review

Patanamon Thongtanunam[*], Chakkrit Tantithamthavorn[*], Raula Gaikovina Kula[†],
Norihiro Yoshida[‡], Hajimu Iida[*], Ken-ichi Matsumoto[*]
[*]Nara Institute of Science and Technology, [†]Osaka University, [‡]Nagoya University, Japan
{patanamon-t, chakkrit-t, matumoto}@is.naist.jp, iida@itc.naist.jp, raula-k@ist.osaka-u.ac.jp, yoshida@ertl.jp

Helping Developers Help Themselves: Automatic Decomposition of Code Review Changesets



- Partitions
  - Non-Trivial Partition 1
    - Binder_Conversions.cs
      - MemberGroupFinalValidationAccessibilityChecks(...)
        - [c-491:491+491:492]
        - [c-494:494+495:495]
    - Binder.cs
  - Non-Trivial Partition 2
  - Trivial Partitions
    - Inside methods changes
      - Binder_Expressions.cs
        - BindNonMethod(...)
        - SynthesizeMethodGroupReceiver(...)

```
                return true;
            }
        }

        if ((object)this.ContainingTy
        var containingType = this.Con
        if ((object)containingType !=
        {
            HashSet<DiagnosticInfo> u
            bool isAccessible = this.
            bool isAccessible = this.
            diagnostics.Add(node, use

            if (!isAccessible)
            {
                // In the presence of
```

Deployment

FastLane: Test Minimization for Rapidly Deployed Large-scale Online Services

Adithya Abraham Philip, Ranjita Bhagwan, Rahul Kumar, Chandra Sekhar Maddila and Nachiappan Nagappan
Microsoft Research

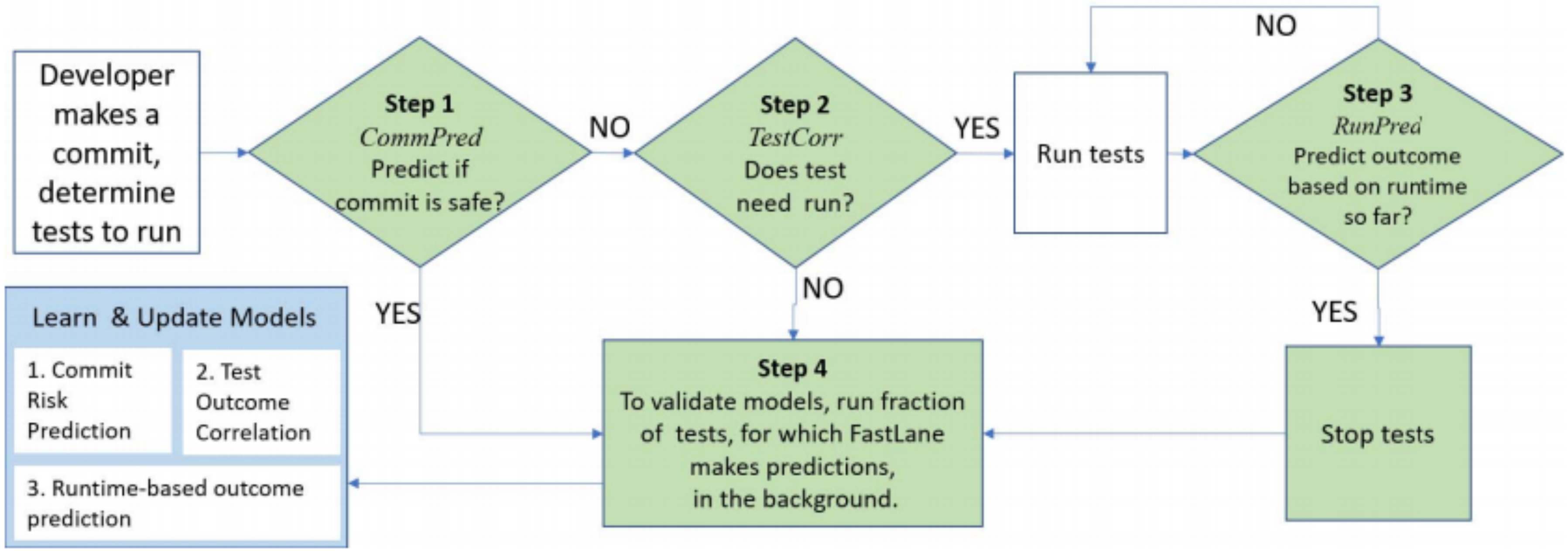

Developer makes a commit, determine tests to run
Step 1
CommPred
Predict if commit is safe?
NO
Step 2
TestCorr
Does test need run?
YES
Run tests
NO
Step 3
RunPred
Predict outcome based on runtime so far?
YES
Stop tests
NO
YES
Step 4
To validate models, run fraction of tests, for which FastLane makes predictions, in the background.
Learn & Update Models
1. Commit Risk Prediction
2. Test Outcome Correlation
3. Runtime-based outcome prediction

Fig. 1. The integrated algorithm and FastLane test prediction flow. This includes all three types of predictions FastLane performs (green boxes): 1. commit risk prediction, 2. test outcome-based correlation, and 3. runtime-based outcome prediction. The blue box captures the learning functionality used to continuously train FastLane.

Productivity

Characterizing Software Developers by Perceptions of Productivity

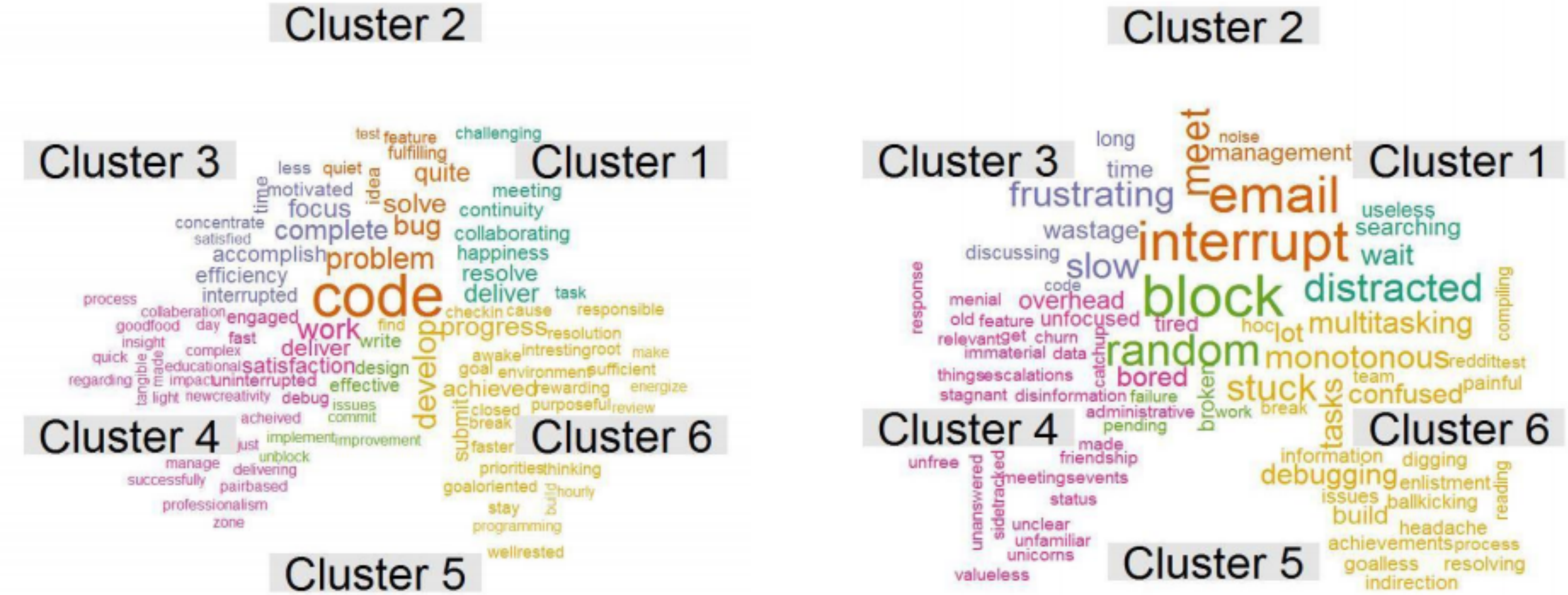
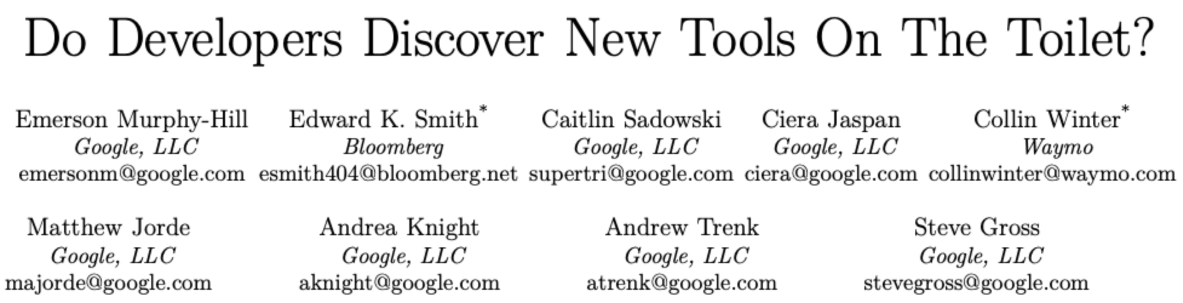


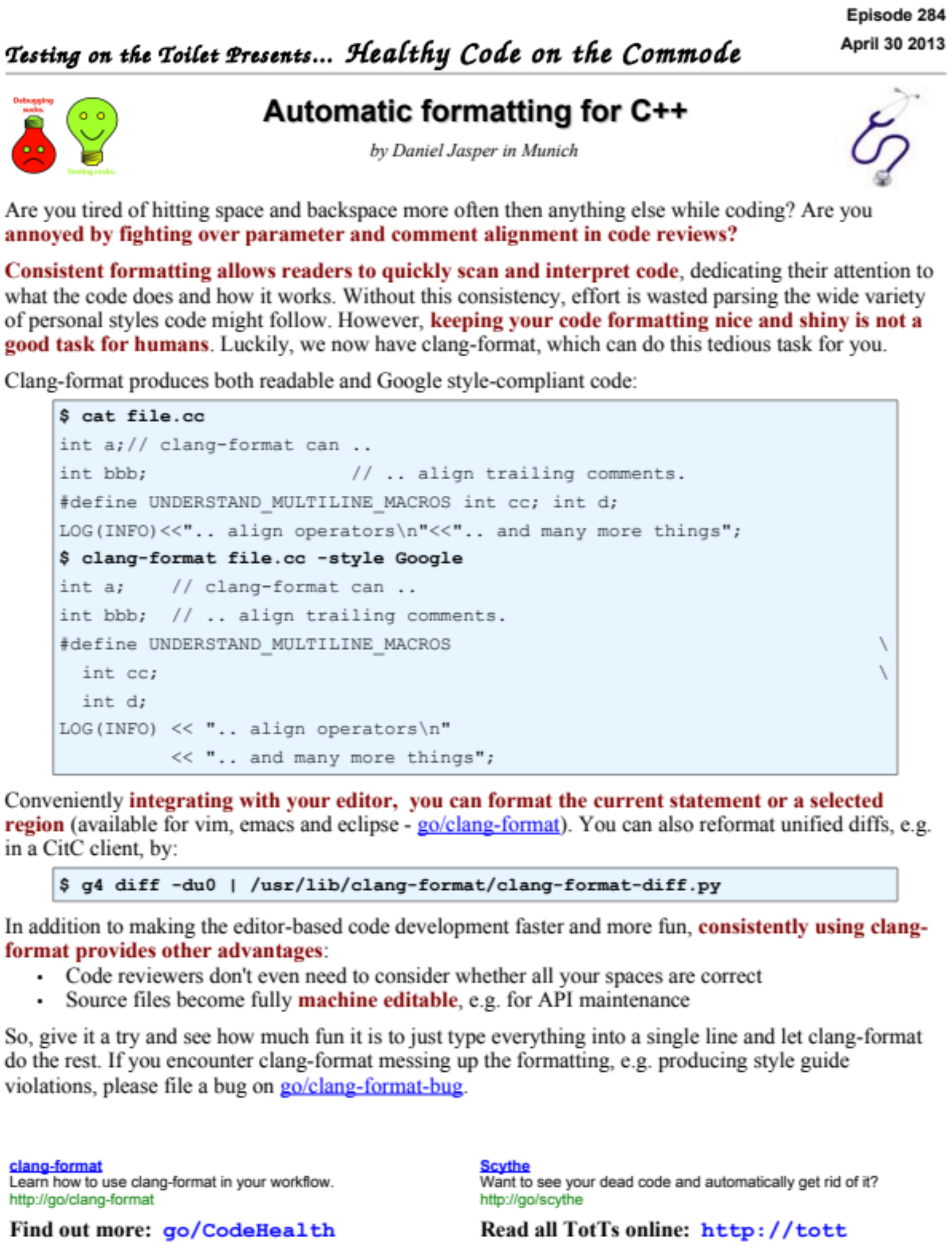
Fig. 1: Comparing the clusters with respect to words that developers associate with ***productive*** (left, Q1) and ***unproductive*** work days (right, Q2). Terms in **turquoise** are related to Cluster 1, **orange** to Cluster 2, **purple** to Cluster 3, **pink** to Cluster 4, **green** to Cluster 5, and **gold** to Cluster 6. The size of a term corresponds to the difference between the maximum relative frequency and the average relative frequency of the word across the six clusters.



Do Developers Discover New Tools On The Toilet?

Emerson Murphy-Hill, *Google, LLC*, emersonm@google.com
Edward K. Smith*, *Bloomberg*, esmith404@bloomberg.net
Caitlin Sadowski, *Google, LLC*, supertri@google.com
Ciera Jaspan, *Google, LLC*, ciera@google.com
Collin Winter*, *Waymo*, collinwinter@waymo.com
Matthew Jorde, *Google, LLC*, majorde@google.com
Andrea Knight, *Google, LLC*, aknight@google.com
Andrew Trenk, *Google, LLC*, atrenk@google.com
Steve Gross, *Google, LLC*, stevegross@google.com

- To increase awareness and adoption of software tools and practices, Google uses a technique called "Testing on the Toilet", or TotT for short
- Evaluation of the effectiveness of TotT
- **Hypothesis**: Testing on the Toilet increases usage of advertised developer tools.
- **Case Study**: CausalImpact, a Bayesian statistical technique that was developed to evaluate the impact of advertising on website traffic



Episode 284
April 30 2013

Testing on the Toilet Presents... Healthy Code on the Commode

Automatic formatting for C++

by Daniel Jasper in Munich

Are you tired of hitting space and backspace more often then anything else while coding? Are you **annoyed by fighting over parameter and comment alignment in code reviews?**

Consistent formatting allows readers to quickly scan and interpret code, dedicating their attention to what the code does and how it works. Without this consistency, effort is wasted parsing the wide variety of personal styles code might follow. However, **keeping your code formatting nice and shiny is not a good task for humans**. Luckily, we now have clang-format, which can do this tedious task for you.

Clang-format produces both readable and Google style-compliant code:

```
$ cat file.cc
int a;// clang-format can ..
int bbb;                    // .. align trailing comments.
#define UNDERSTAND_MULTILINE_MACROS int cc; int d;
LOG(INFO)<<".. align operators\n"<<".. and many more things";
$ clang-format file.cc -style Google
int a;    // clang-format can ..
int bbb;  // .. align trailing comments.
#define UNDERSTAND_MULTILINE_MACROS                                     \
  int cc;                                                               \
  int d;
LOG(INFO) << ".. align operators\n"
          << ".. and many more things";
```

Conveniently **integrating with your editor, you can format the current statement or a selected region** (available for vim, emacs and eclipse - go/clang-format). You can also reformat unified diffs, e.g. in a CitC client, by:

```
$ g4 diff -du0 | /usr/lib/clang-format/clang-format-diff.py
```

In addition to making the editor-based code development faster and more fun, **consistently using clang-format provides other advantages**:

- Code reviewers don't even need to consider whether all your spaces are correct
- Source files become fully **machine editable**, e.g. for API maintenance

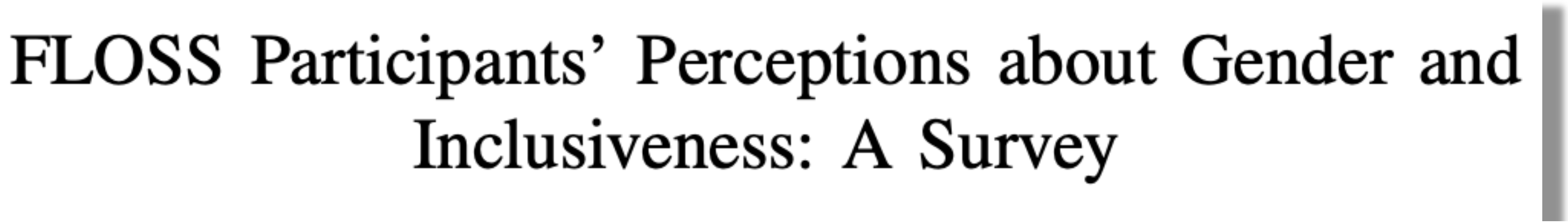
So, give it a try and see how much fun it is to just type everything into a single line and let clang-format do the rest. If you encounter clang-format messing up the formatting, e.g. producing style guide violations, please file a bug on go/clang-format-bug.

clang-format
Learn how to use clang-format in your workflow.
http://go/clang-format

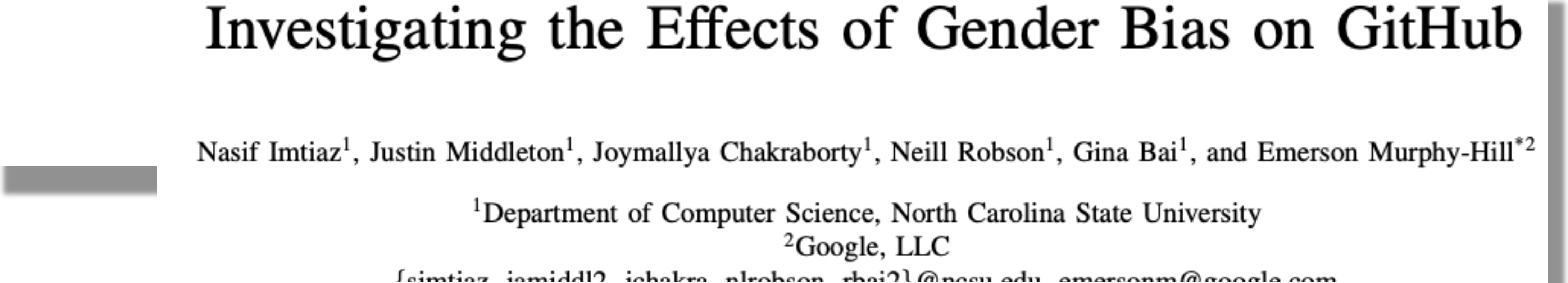
Scythe
Want to see your dead code and automatically get rid of it?
http://go/scythe

Find out more: go/CodeHealth

Read all TotTs online: http://tott



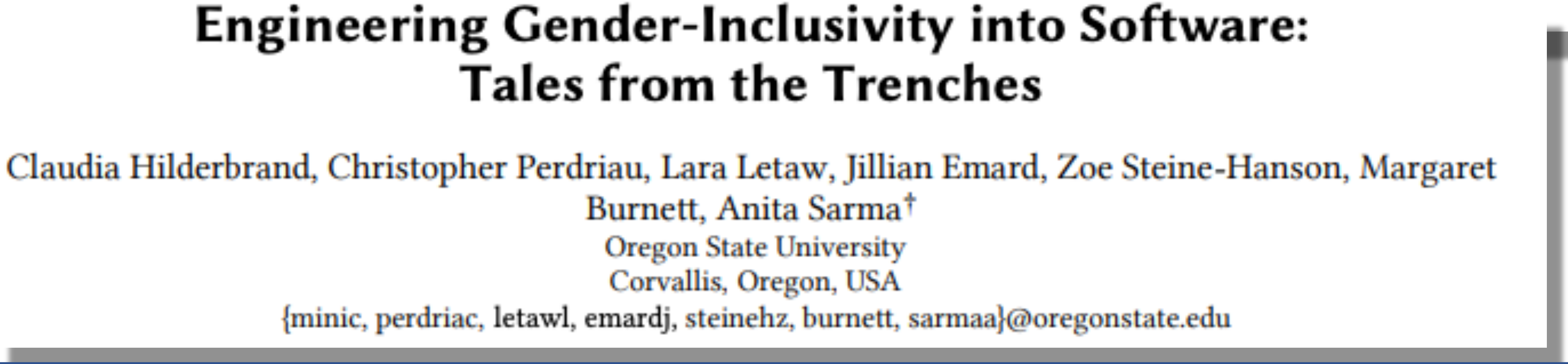
FLOSS Participants' Perceptions about Gender and Inclusiveness: A Survey



Investigating the Effects of Gender Bias on GitHub

Nasif Imtiaz[1], Justin Middleton[1], Joymallya Chakraborty[1], Neill Robson[1], Gina Bai[1], and Emerson Murphy-Hill[*2]

[1]Department of Computer Science, North Carolina State University
[2]Google, LLC
{simtiaz, jamiddl2, jchakra, nlrobson, rbai2}@ncsu.edu, emersonm@google.com



Engineering Gender-Inclusivity into Software: Tales from the Trenches

Claudia Hilderbrand, Christopher Perdriau, Lara Letaw, Jillian Emard, Zoe Steine-Hanson, Margaret Burnett, Anita Sarma[†]
Oregon State University
Corvallis, Oregon, USA
{minic, perdriac, letawl, emardj, steinehz, burnett, sarmaa}@oregonstate.edu

Clone Detection

Code Clone Categorization

- Type-1 clones – Identical code fragments but may have some variations in whitespace, layout, and comments
- Type-2 clones – Syntactically equivalent fragments with some variations in identifiers, literals, types, whitespace, layout and comments
- Type-3 clones – Syntactically similar code with inserted, deleted, or updated statements
- Type-4 clones – Semantically equivalent, but syntactically different code

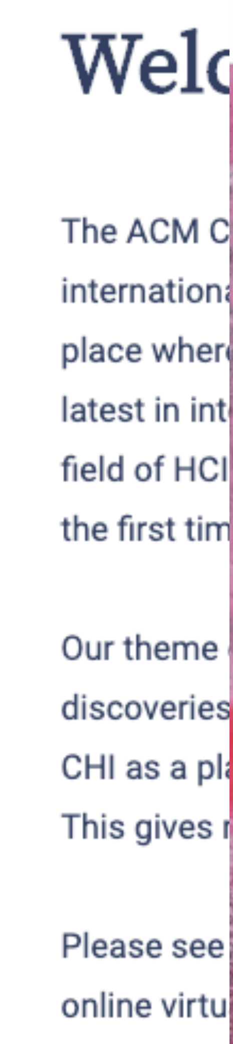
Key points of Code Clone

- Pros
  - Increase performance
    - Code inlining vs. function call
  - Increase program readability
- Cons
  - Increase maintenance cost
    - If one code fragment contains a bug and gets fixed, all its clone peers should be always fixed in similar ways.
  - Increase code size

Detecting Strategies

- Text matching
- Token sequence matching
- Graph matching

Collaboration

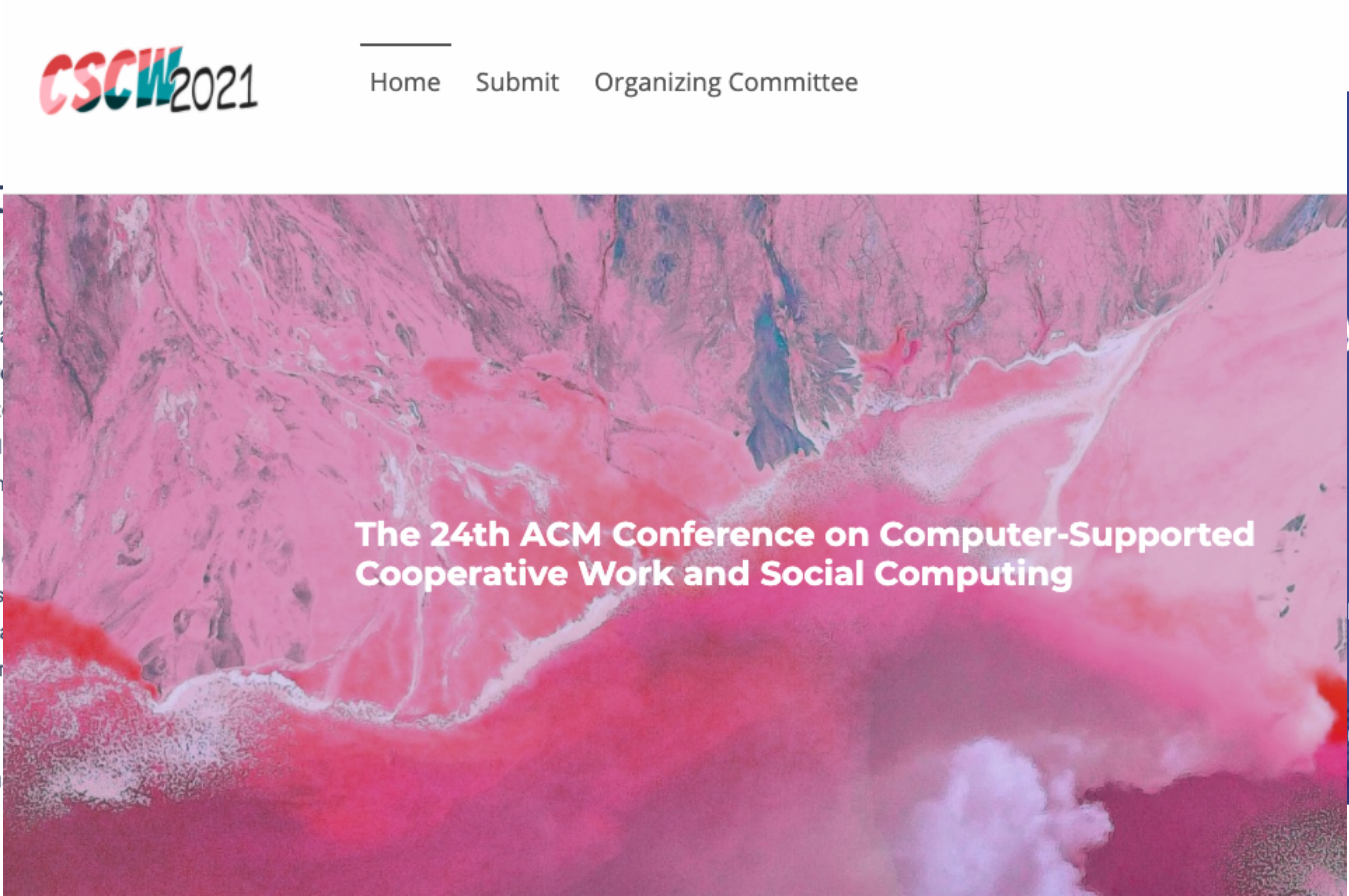


Welc

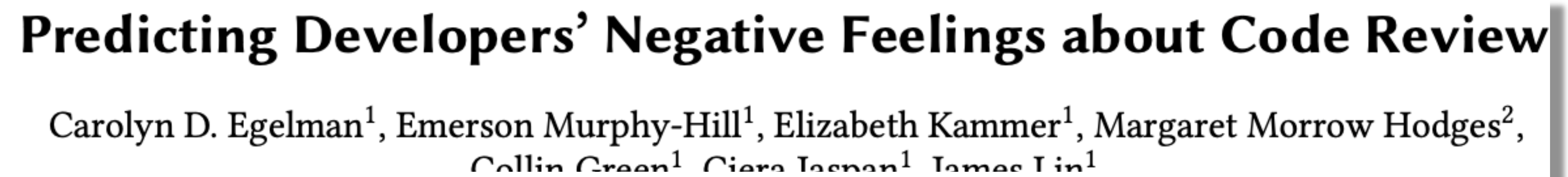
The ACM C
internationa
place where
latest in int
field of HCI
the first tim

Our theme
discoveries
CHI as a pl
This gives

Please see
online virtu

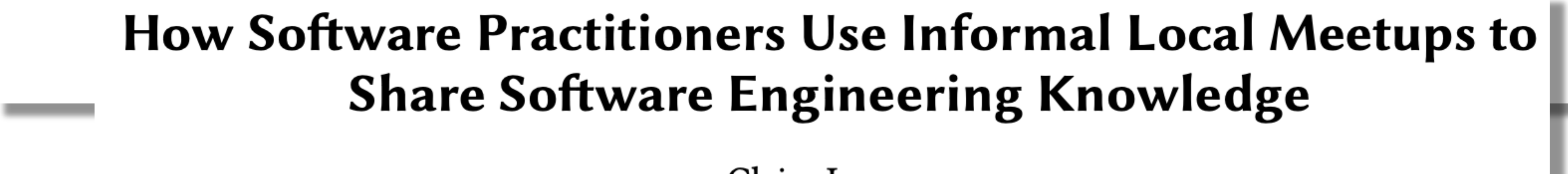


The 24th ACM Conference on Computer-Supported Cooperative Work and Social Computing

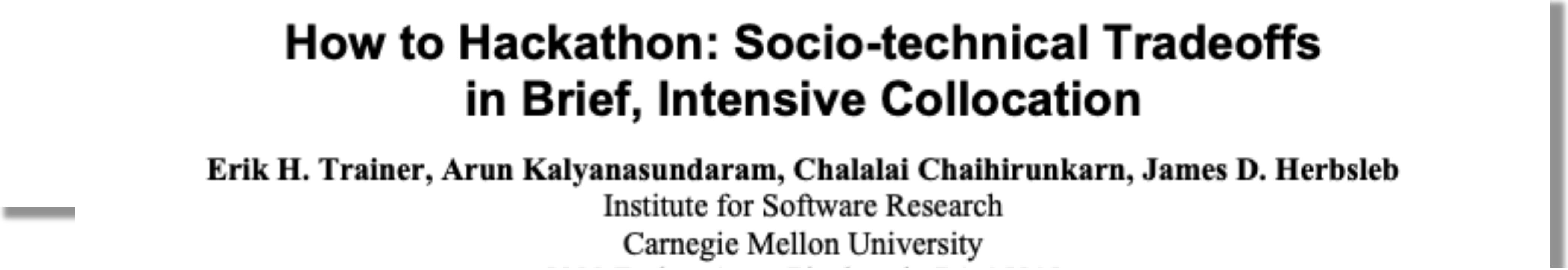


Predicting Developers’ Negative Feelings about Code Review

Carolyn D. Egelman[1], Emerson Murphy-Hill[1], Elizabeth Kammer[1], Margaret Morrow Hodges[2],
Collin Green[1], Ciera Jaspan[1], James Lin[1]

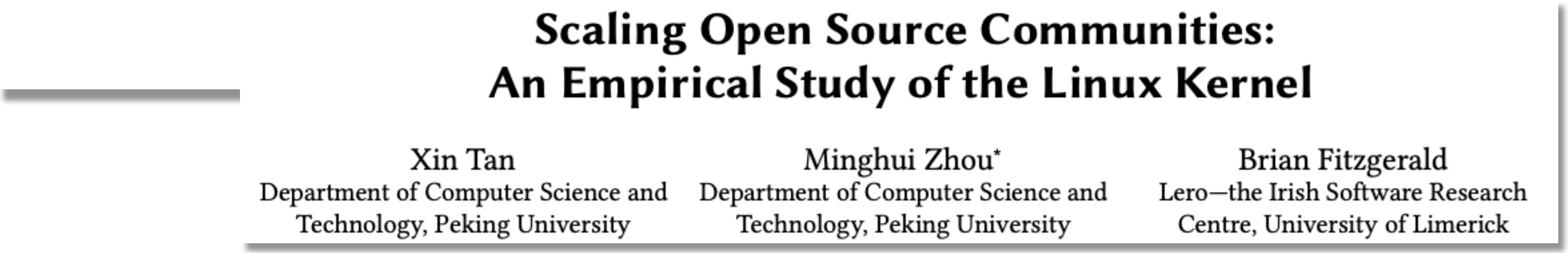


How Software Practitioners Use Informal Local Meetups to Share Software Engineering Knowledge



How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation

Erik H. Trainer, Arun Kalyanasundaram, Chalalai Chaihirunkarn, James D. Herbsleb
Institute for Software Research
Carnegie Mellon University



Scaling Open Source Communities: An Empirical Study of the Linux Kernel

Xin Tan
Department of Computer Science and Technology, Peking University

Minghui Zhou*
Department of Computer Science and Technology, Peking University

Brian Fitzgerald
Lero—the Irish Software Research Centre, University of Limerick