Context Analysis of Technical Papers ICSE 2020

Table of Contents

[B535 - Context-aware In-process Crowdworker Recommendation 2](#_Toc51254033)

[A087 - Causal Testing: Understanding Defects' Root Causes 3](#_Toc51254034)

[A654 – A Tale from the Trenches: Cognitive Biases and Software Development 4](#_Toc51254035)

[A691 – Here We Go Again: Why Is It Difficult for Developers to Learn Another Programming Language? 4](#_Toc51254036)

[B073 – Big Code != Big Vocabulary: Open-Vocabulary Models for Source codeType: XXX 5](#_Toc51254037)

[A322 – Unblind Your Apps: Predicting Natural-Language Labels for Mobile GUI Components by Deep Learning 5](#_Toc51254038)

[A309 – Translating Video Recordings of Mobile App Usages into Replayable Scenarios 5](#_Toc51254039)

[A481 – Time-travel Testing of Android Apps 6](#_Toc51254040)

[B171 – Primers or Reminders? The Effects of Existing Review Comments on Code Review 6](#_Toc51254041)

[B435 – Towards the Use of the Readily Available Tests from the Release Pipeline as Performance Tests. Are We There Yet? 7](#_Toc51254042)

[A949 – White-box Fairness Testing through Adversarial Sampling 7](#_Toc51254043)

[B110 – Taxonomy of Real Faults in Deep Learning Systems 7](#_Toc51254044)

[B159 – An Empirical Study on Program Failures of Deep Learning Jobs 8](#_Toc51254045)

[AB000 – Title 8](#_Toc51254046)

[AB000 – Title 8](#_Toc51254047)

[AB000 – Title 9](#_Toc51254048)

[AB000 – Title 9](#_Toc51254049)

[AB000 – Title 9](#_Toc51254050)

[AB000 – Title 10](#_Toc51254051)

[AB000 – Title 10](#_Toc51254052)

### AWARDED PAPERS A close up of ware Description automatically generated

# B535 - Context-aware In-process Crowdworker Recommendation

Type: pilot study, proposal of an approach + evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| * Crowdsourced Software Development * Crowdworking * Online, Distributed, open software * Testing | Type of software dev process  Working Env  Software Dev Process | Thema | Titel, Abstract, Introduction |
| Open Source Software, On-Demand Web Storage Software | Software Dev Type, Sourcing  Software Type | Abgrenzung | Introduction |
| Bug detection | Software Dev Process, SE Task | Detail | Background, non-yielding windows in crowdtesting processes |
| Mobile application | Software Type | Detail | Experiment Design 4.2 Dataset |
| Talent identification |  | Detail | Discussion |
| Functional, usability, security, performance tests | Test types | Verallgemeinbarkeit | Threats of Validity |
| Bug triage, code review, developing, consulting, documentation, analysis | SE Task | Abgrenzung | Related Work |

„Fourth, for the generalizability of our approach, a recent systematic review [66] has shown current crowdtesting services are dominated by functional, usability, and security test of mobile ap- plications. “

„We believe that the proposed approach is generally applicable to supporting other testing types such as security and performance testing, “

# A087 - Causal Testing: Understanding Defects' Root Causes

Type: benchmarking, controlled experiment, prototype

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Causal Testing, Root-cause Analysis / identification | Debugging approach | Thema | Title, Abstract |
| Delta debugging | Debugging approach | Abgrenzung | Introduction |
| Java  Eclipse | Technology  IDE | Detail | Introduction |
| Fault localization (automated debugging) | Debugging approach | Abgrenzung | Introduction |
| Automated test generation | Bug detection approach | Abgrezung, Detail | Introduction |
| Background and experience of study participants |  | Verallgemeinbarkeit | Threats of Validity |
| Mutation testing  In-house testing  Fuzz testing (automated test input generation)  Traditional testing  Reproducing field failures |  | Abgrenzung | Related Work |

„Our use of this well-known and widely-used benchmark of real-world defects aims to ensure our results generalize“

„The user study used 37 participants, which is within range of higher data confidence and is above average for similar user studies. Our study also relied on participants with different backgrounds and experience.“

# A654 – A Tale from the Trenches: Cognitive Biases and Software Development

Type: 2part observational field study, interview

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Company culture  Education  Experience |  | Abgrenzung | Background |
| Representation of work environment  Vs lab-based studies, hackathons |  | Details  Abgrenzung | Introduction, Background |
| Start up, type of companies  Experience of developers  Role/ job position |  | Details | Methodology |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„While desirable, generalizability was not the main focus of this study “

Nicht wirklich Kontextfaktoren, weil qualitative study?

# A691 – Here We Go Again: Why Is It Difficult for Developers to Learn Another Programming Language?

Type: empirical study / inspection of Stack Overflow questions, semi-structured interview

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Programming experience |  | Detail | Study Design |
| Programming languages  C, C++, C#, Visual Basic, Objective-C, Swift, Python, Java, Scala, PHP, R, Kotlin, Matlab, Node, Ruby, Perl, Lua, Clojure |  | Detail | Results |
| Programming experience  Novices |  | Abregrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„Because the sampling approach is non-probabilistic, it does not allow for sample-to-population, or statistical generalization. Rather, our approach targets diversity (rather than representativeness) in order to identify evidence of interference across many different programming languages. “

„We believe our results provide further insight as to why plans may not generalize across languages. „

# B073 – Big Code != Big Vocabulary: Open-Vocabulary Models for Source codeType: XXX

Type: empirical study, repository mining

Result: Neural language model (NLM) and it’s evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Java, C Python |  | Detail | Abstract |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„Finally, the technique of using subword units is not limited to language modeling, but can easily be incorporated into any neural model of code, such as models to suggest readable names [3], summarizing source code [5, 49], predicting bugs [70], detecting code clones [87], comment generation [47], and variable de-obfuscation [10]. “

# A322 – Unblind Your Apps: Predicting Natural-Language Labels for Mobile GUI Components by Deep Learning

Type: analysis of apps (mining study), presentation of model and experiments

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Mobile application  Android and ios |  | Thema  Details | Titel, Abstract |
| Accessibility | Quality issues | Thema, Abgrenzung | Abstract, Related Work |
| Compatibility, performance, energy-efficiency, GUI design, GUI animation linting, localization, privacy security | Quality issues | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Nicht wirklich was zur Verallgemeinerung gesagt, weil sehr spezifisches Thema? Und es mehr proposal des Ansatz/Model ist

Es wurde erwähnt, was die future work sind um es zu verbessern

# A309 – Translating Video Recordings of Mobile App Usages into Replayable Scenarios

Type: presentation of an approach, case study

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Mobile Applications  Android |  | Thema | Title, Abstract |
| Crowdsourced App Feedback |  | Thema | Abstract |
| Industrial partners |  | Detail | Abstract |
| Open and closed sourced app |  | Detail |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„Our approach has various limitations that serve as motivation for future work “

„can be trained once and used for any device with a given screen size “

„Threats to the generalizability of our conclu- sions are mainly related to: (i) the number and diversity apps used in our evaluation; (ii) the representativeness of usage scenarios depicted in our experimental videos; and (iii) the generalizability of the responses given by the interviewed developers. “

„Finally, we do not claim that the feedback we received from developers generalizes broadly across industrial teams. However, the positive feedback and suggestions for future work we received in our interviews illustrate the potential practical usefulness of V2S. “

# A481 – Time-travel Testing of Android Apps

Type: presentation of approach, experiment for evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Android apps  Mobile application |  | Thema | Title, Abstract |
| Open and closed apps |  | Detail | Abstract |
| Testing | Software Dev Process | Thema | Title |
| Large Industrial apps |  | Detail | Conclusion |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Nicht wirklich was zur Verallgemeinerung gesagt, weil presentation of approach?

# B171 – Primers or Reminders? The Effects of Existing Review Comments on Code Review

Type: controlled online experiment, psychological experiment

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Code Review | Software Dev Process | Thema | Title, Abstract |
| Experience |  | Detail | Introduction |
| Android project |  | Detail | Background and related work |
| Popular open source software systems |  | Detail | Background and related work |
| Requirement elicitation, pair porgramming, software reuse, project management, effort estimation |  | Abgrenzung | Background and related work |
| Correctness issues  Maintainability, design issues |  | Detail  Abgrenzung | Experiment Design and Structure |
| Java |  | Detail | Experiment Design and Structure |

„To have a diverse sample of subjects (representative of the overall population of software developers who employ contemporary code review), we invited developers from several countries, organizations, education levels, and background. “

# B435 – Towards the Use of the Readily Available Tests from the Release Pipeline as Performance Tests. Are We There Yet?

Type: exploratory study

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Testing | Software Dev Process | Thema | Titel |
| Performance | Software quality | Thema | Abstract |
| Open source systems  <-> commercial closed source systems |  | Detail |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„Our findings might not be generalizable to other systems. Future studies can apply our approach on other systems, such as commercial closed source systems.“

# A949 – White-box Fairness Testing through Adversarial Sampling

Type: presentation of an approach, qualitative evaluation by comparing it with state-of-the-art approaches, evaluation by benchmarking

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| DeepLearning, DNN |  | Thema | Abstract |
| Dependability  fairness | DL Quality issues | Detail  Thema | Abstract |
| Lightweight procedures   * Gradient computation * Clustering   Guided Search  Input Specific |  | Eingrenzung  Differenzierung, Abgrenzung | Abstract  Methodology, Qualitative Evaluation |
| fraud detection  facial recognition  self-driving  medical diagnosis  banking (domain of datasets) | DL application | Abgrenzung  Detail | Introduction  Experiment |
| Machine Learning models |  | Abgrenzung | Introduction |
|  |  |  |  |
|  |  |  |  |

„our approach only employs lightweight procedures like gradient computation and clustering, “

# B110 – Taxonomy of Real Faults in Deep Learning Systems

Type: manual artifact analysis, semi-structured interview, survey

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| TensorFlow, Keras and PyTorch | DL frameworks | Detail | Abstract |
| Deep Learning |  | Thema | Titel |
| Translating texts  Converting voice tot ext  Banking  Medicine  Self-driving | DL applications | Detail | Introduction |
|  |  |  |  |
|  |  |  |  |

„The main threat to the external validity is general- isation beyond the three considered frameworks, the dataset of artefacts used and the interviews conducted. “

# B159 – An Empirical Study on Program Failures of Deep Learning Jobs

Type: manual analysis of artefacts, interviews

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Root cause, bugfixing |  | Thema | Abstract |
| Deep Learning |  | Thema | Title |
| Debugging and testing |  | Thema | Abstract |
| Speech and Image Recognition, NLP,  reading comprehension, object detection, gaming, advertisement | DL applications | Details | Introduction  Background |
| Microsoft Azure  Amazon SageMaker  Google Cloud AI | DL training, testing plattforms | Details | Introduction |
| TensorFlow, PyTorch, MXNet, CNTK, Philly | DL frameworks | Details | Introduction |
| Industrial jobs <-> Github issues, Stackoverflow questions |  | Abgrenzung | Introduction |
| Code/script defects  Hardware or system failures |  | Eingrenzung  Abgrenzung | Introduction |
| Microsoft | Partiicpated company | Verallgemeinbarkeit | Discussion |
| Python | Technology | Veralgemeinbarkeit | Discussion |

„we believe that our failure categories are prevalent and most of our results can be generalized to other DL platforms. “

### AWARDED PAPERS A picture containing gear, ware, wheel Description automatically generated

# B273 – HyDiff: Hybrid Differential Software Analysis

Type: Presenting an approach, introducing differential metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Deep Learning  Bug Detection |  | Thema | Abstract |
| Java | Technologie | Detail | Implementation Details |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„We cannot claim, the results for each case study will generalize for other kinds of programs (or classifiers) written in other languages or from other domains. In order to mitigate the impact of randomness on the results, we repeated each experiment 30 times and report 95%-confidence intervals. „

# A335 – SLEMI: Equivalence Modulo Input (EMI) Based Mutation of CPS Models for Finding Compiler Bugs in Simulink

Type: experiments, introduction of a method

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Bug detection |  | Thema | Abstract |
| Model based design tools |  | Thema | Introduction |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„just represent a particular implementation of the underlying bug-finding techniques for one particular CPS language, evaluated with models produced by a single seed model generator “

„the EMI approaches may not generalize to other CPS models. “

# B261 – Efficient Generation of Error-Inducing Floating-Point Inputs via Symbolic Execution

Type: presentation of a tool, experimental evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Floating-point programs, numerical bugs |  | Thema | Introduction, Abstract |
| C | Technology | Detail | Experimental Evaluation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„This observation on the inaccuracy patterns can be generalized to other floating- point code. It is highly likely that an input that conforms to our inaccuracy checks will produce a high numerical error in the result “

# B024 – Efficient Generation of Error-Inducing Floating-Point Inputs via Symbolic Execution

Type: presentation of a tool, experiments

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Finding bugs |  | Thema | Abstract |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Keine Verallgemeinerung, Threats of Validity

„We designed and implemented Ankou, the first grey-box fuzzer that operates with a high dimensionality representation of the program state space. “

# A615 – On the Efficiency of Test Suite based Program Repair: A Systematic Assessment of 16 Automated Repair Systems for Java Programs

Type: Systematic Assessment

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Java | Technology | Eingrenzung | Title |
| Testing |  | Thema | Title, Abstract |
| Opensource tools |  | Details | Abstract |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„Our study considers only the Defects4J benchmark and only java repair tools. All findings might thus be valid only for this configuration. Nevertheless, this threat is mitigated by the fact that we use a large set of repair tools and a renowned defect benchmark to study a performance criterion that was largely ignored in the literature. “

# B410 - Quickly Generating Diverse Valid Test Inputs with Reinforcement Learning

Type: proposal of an approach, evaluation, benchmarking

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Property based Testing |  | Thema | Title, Abstract |
| Reinforcement Learning |  | Thema | Title |
| Fuzz Testing  Search-based software testing |  | Abgrezung | Related Work |
|  |  |  |  |
|  |  |  |  |

# A026 – Practical Fault Detection in Puppet Programs

Type: proposal of an approach, evaluation, performance benchmarking

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Infrastructure as Code (IaC) |  | Thema | Introduction |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: X

„FStrace is a generic model that can be applied to other domains with partially ordered constructs “

**^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^**

**noch mal iterieren – Future Work, Applications (versteckt), evtl Target group, auch mehr achten auf related Work**

# A542 – Gap between Theory and Practice : An Empirical Study of Security Patches in Solidity

Type: empirical study, manual investigation of Solidity contracts

Target group: Solidity developers, security experts

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Solidity  Blockchain, Ethereum | Technology | Thema | Abstract, Title  Abstract |
| Security |  | Thema | Title |
| security-critical applications |  | Detail | Introduction |
| Cripto currency, financial services, games | Domain, Application | Detail | Introduction |
| Vulnerability caused by compilers patches vs caused by solidity language features |  | Abgrenzung | Related Work |
| Real world developers |  | Abgrenzung | Related Work |
| Deepness/detailness of study |  | Abgrenzung | Related Work |

Threats of Validity: Ja aber nur Beschreibung von internal Validity, nicht external oder construct

# A075 - Debugging Inputs

Type: presentation of an algorithm, experiments

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Debugging  Automated debugging and repair, delta debuggin |  | Thema | Title  Thema |
| Data corruption | Application | Detail | Abstract |
| Data recovery, Data repair |  | Thema | Abstract |
| open source C, C++ or Java programs |  | Detail | Evaluation Setup |
| XXX relevant? |  | Abgrenzung | Related Work |
|  |  |  |  |

Threats of Validity: V

„There is a threat that ddmax does not generalize to other applications and gram- mars. However, we have mitigated this threat by evaluating ddmax using mature subject programs with varying input sizes.“

„we have presented the first generic technique for au- tomatically repairing failure-inducing inputs “

Future Work: V

Synthesizing input structures, Learned grammars, From input repair to code repair, End-user debugging, Hybrid repair, Semantic Input Repair, Fuzzing

# A530 - Empirical Review of Automated Analysis Tools on 47,587 Ethereum Smart Contracts

Type: Empirical Review of Automated Analysis Tools, Presentation of a framework, experiment

„first systematic comparison of recently proposed techniques to better understand their real capabilities “

Target group: Blockchain Developers

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Ethereum  Blockchain  Solidity, Etherscan | Technology | Thema | Title  Abstract |
| auto- mated analysis, testing, and debugging |  | Thema | Abstract |
| Distributed applications |  | Detail | Introduction |
| Dataset enablinh reproducibility  more systematic and comprehensive findings |  | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V

„We also aim to reduce threats to external validity and ensure the reproducibility of our evaluation by providing the source of our instrumentation tool, the scripts used to run the evaluation, and all data gathered. “

Future Work: „We identify four core challenges: increasing and ensuring the quality of the analysis, extending the scope of problems addressed by these tools, integrating the analysis into the development process, and extending the current taxonomy. “ XX other applications

# B459 - Automatically Testing String Solvers Type: XXX

Type: presentation of an automatic approach for generating test cases, experimental evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| program verification, program synthesis, and test case generation | Application | Detail | Abstract |
| Satisfiability modulo theories solvers  String solvers  Automatabased solvers |  | Thema  Eingrenzung  Abgrenzung | Abstract  Title  Abstract |
| Automatic testing |  | Thema | Title |
| Differential Testing  Fuzzing  Formal verification  Validation and proof checking |  | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

„Our technique is not limited to SMT solvers; it can also be applied for testing other types of solvers, “

Threats of Validity: V, aber keine external validity

Future Work: other classes of solvers and other components, MAX-SMT solvers and the quantifier instantiation mechanism

# A765 - MemLock: Memory Usage Guided Fuzzing

Type: proposing of a new greybox fuzzing technique, evaluation, experiments, benchmarking

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Fuzzing  Finding Bugs |  | Thema | Title |
| Critical security issues and bugs, due to uncontrolled memory consumption  Code coverage  Memory corruptoin |  | Thema  Abgrenzung  Abgrenzung | Abstract  Abstract  Introduction |
| 14 widely-used real-world programs |  |  |  |
| Coverage-based Grey-box Fuzzing  Static Analysis  Dynamic Analysis |  | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: sehr kurz, nur construct validity

Future Work: nicht explizit, aber angedeutet in threats of validity

# B248 - Symbolic Verification of Message Passing Interface Programs

Type: presentation of a tool, evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Message passing |  | Thema | Title |
| High performance computing  MPI programs with non-blocking operations  Non-determinism operations | Application  Programming paradigm  Program feature | Detail  Abgrenzung | Abstract |
| real-world MPI verification tasks |  | Detail | Abstract |
| symbolic execution, model checking | techniques | Detail | Introduction |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: X

Future Work: 3steps in the conclusion

# A271 - SAVER: Scalable, Precise, and Safe Memory-Error Repair

Type: proposal of a method/ presentation of an algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Memory-error repair  Memory-error-detector |  | Thema | Title, Abstract  Introduction |
| Open-source C programs (-> unsafe language, in comparison to Java) | technology | Eingrenzung | Abstract |
| Automated program repair techniques |  | Thema | Abstract |
| Flow graph, static analysis |  | Details | Introduction |
| Finding and fixing safety errors |  | Thema | Introduction |
| Memory errors |  | Eingrenzung | Introduction |
| Compile time |  | Details | Introduction |
| Other special-purpose automated program repair techniques:   * Safety policy violation * Buffer/interger overflows * Null dereferences * Concurrency error * Energy bug   General-purpose techniques   * Test-based: program verfication, metamorphic testing, contracts, test cases * Generate-and-validate: search algorithms * Semantic-based: constraint-satisfaction problem, SMT solver   Compile-time Memory Management |  | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V, sehr knapp, nur internal validity

Future Work: X

# A678 - Neurological Divide: An fMRI Study of Prose and Code Writing

Type: human study, brain scan, controlled experiment

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Cognitiv processes during code writing |  | Thema | Abstract  Title |
| Prose writing |  | Abgrenzung, Vergleich | Abstract |
| Code comprehension  Code review  Data structure manipulation  Prose review |  | Abgrenzung | Introduction,  Related Work |
| u ndergraduate and graduate students with an average of 5.2 semesters of programming experience |  | Details | Threat of Validity |
| Observational investigation |  | Abgrenzung | Conclusion |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: nur internal validity

„Nevertheless, our re- sults may not generalize to all in-the-wild programming; we leave a more thorough investigation to future research. “

Future Work:

# A986 - TRADER: Trace Divergence Analysis and Embedding Regulation for Debugging Recurrent Neural Networks

Type: proposal of a new technique

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Model Debugging  Recurrent Neural Network for textual input |  | Thema | Title |
| Automatic diagnosis  Word embeddings |  | Thema | Abstract |
| Software related applications   * Semantic programm embeddings * Bug detection * Software maintenance * Processing textual software artifacts: commits, comments, reviews   Other applications   * Selfdriving cars * Online ads * User authentication and monitoring at wearables/mobile devices |  | Eingrenzung  Abgrenzung | Introduction |
| convolutional neural networks (CNNs ) for fixes size inputs as images |  | Abgrenzung | Introduction |
| Software debugging   * Trace analysis * Differential analysis |  | Abgrenzung | Related Work |
| Fully automated <-> human effort |  | Abgrenzung | Related Work |
|  |  |  |  |

Threats of Validity: Nur internal

„The proposed TRADER, however, is not application specific “

„Our experiments show that our technique can consistently and effectively improve accuracy for real world models and datasets by 5.37% on average. “

Future Work: X

# B122 - Testing DNN Image Classifier for Confusion & Bias Errors

Type: proposal of a technique, evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Deep Neural Networks (DNN)  Testing  Image classification   * Computer vision |  | Thema  Details | Title  Abstract |
| consumer and business applications to safety-critical domains   * Self driving cars * Medical diagnos * Google photos search, fb image tagging |  | Details | Abstract |
| Automatic detectection of class- based confusion and bias errors  Bug Detection |  | Thema | Abstract |
| DNN-driven image classification software | Application | Thema | Abstract |
| XXX |  | Abgrenzung | Related Work |
| Debugging, Repair |  | Abgrenzung, Future Work | Discussion & Threat of Validity |
|  |  |  |  |

Threats of Validity: nur internal validity

Future Work: „A natural follow-up of our work will be debugging and repair leveraging DeepInspect’s bug detection “

# B135 - Repairing Deep Neural Networks: Fix Patterns and Challenges

Type: artefact analysis: mining Stackoverflow and Github, benchmarking

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Deep Neutral Networks  Bug repair  Bug fix pattern |  | Thema | Title  Abstract |
| deep learning libraries Caffe, Keras, Tensorflow, Theano, and Torch | Technologie | Details | Abstract |
| Bug detection |  | Abgrenzung | Introduction |
| Safety critical data science applications of DNNs |  | Detail | Discussion |
| Programming bugs (Java) |  | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V

Future Work: X

# B509 - Lazy Product Discovery in Huge Configuration Spaces

Type: proposal of a method, formalism, evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Software system configurations |  | Thema | Abstract |
| Highly-configurable software systems | Application | Thema | Abstract |
| Automated Lazy product discovery  <-> standard product discovery, interactive product discovery |  | Abgrenzung | Introduction, Conclusion |
| interdependent feature models. |  | Eingrenzung | Introduction |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V

„We plan to investigate other product-discovery problems for Gentoo and for other domains to get more insights. In particular, it would be interesting to investigate how lazy product discovery performs when varying both the size and the amount of interdependencies of the feature models “

Future Work: V

# B098 - Managing data constraints in database-backed web applications

Type: study, XX (wird erst im Detail beschrieben), user study

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Database-backed web applications   * Large amount of persistent data * Data generated by users | Application | Details | Title |
| Managing data constrainst  requirements |  | Thema | Title  Introduction |
| Verifying data constraints  Verifying applications  Specifying security/privacy-related data constraints |  | Abgrenzung | Related Work |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V

Future Work: X

# B086 - Improving Data Scientist Efficiency with Provenance

Type: contextual inquiry, presentation of a tool, controlled experiment

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Data science  Data processing   * Preprocessing data inputs * Training models * Parameter tuning * Adding analysis functions * Propagating changes |  | Thema | Title  Introduction |
| Debugging, testing |  | Details | Abstract |
| Using Data scripts for data analysis | Application | Anwendung? | Introduction |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V

Future Work: V

# B049 - On the Recall of Static Call Graph Construction in Practice

Type: XXX 2.Iteration, wird nicht klar in der 1.iteration

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| call graph construction |  | Thema | Title |
| Static program analysis   * Bug detection   <-> hybrid, dynamic analysis |  | Thema  Abgrenzung | Introduction  Related Work |
| real-world Java programs |  | Details | Abstract |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity: V

Future Work: X

# A125 - Watchman: Monitoring Dependency Conflicts for Python Library Ecosystem

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity:

Future Work:

# A284 - Revealing Injection Vulnerabilities by Leveraging Existing Tests

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity:

Future Work:

B061 - Heaps'n Leaks: How Heap Snapshots Improve Android Taint Analysis

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity:

Future Work:

### AWARDED PAPERS A picture containing gear, ware, wheel Description automatically generated

# A

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity:

Future Work:

1. A641 Detection of Hidden Feature Requests from Massive Chat Messages via Deep Siamese Network
2. B348 Posit: Simultaneously Tagging Natural and Programming Languages
3. B359 CPC: Automatically Classifying and Propagating Natural Language Comments via Program Analysis
4. A409 A Study on the Prevalence of Human Values in Software Engineering Publications
5. B183 Mitigating Turnover with Code Review Recommendation: Balancing Expertise, Workload, and Knowledge Distribution
6. A258 Burn After Reading: A Shadow Stack with Microsecond-level Runtime Rerandomization for Protecting Return Addresses
7. Testing File System Implementations on Layered Models
8. How Does Misconfiguration of Analytic Services Compromise Mobile Privacy?
9. Understanding the Automated Parameter Optimization on Transfer Learning for Cross-Project Defect Prediction: An Empirical Study
10. Multiple-Entry Testing of Android Applications by Constructing Activity Launching Contexts
11. HeteroRefactor: Refactoring for Heterogeneous Computing with FPGA
12. Recognizing Developers' Emotions while Programming
13. Gang of Eight: A Defect Taxonomy for Infrastructure as Code Scripts
14. Tailoring Programs for Static Analysis via Program Transformation
15. Near-Duplicate Detection in Web App Model Inference
16. ComboDroid: Generating High-Quality Test Inputs for Android Apps via Use Case Combinations
17. Verifying Object Construction

### AWARDED PAPERS A picture containing gear, ware, wheel Description automatically generated

# A

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Threats of Validity:

Future Work:

1. Securing UnSafe Rust Programs with Xrust
2. How to Not Get Rich: An Empirical Study of Donations in Open Source
3. ReluDiff: Differential Verification of Deep Neural Networks
4. SLACC: Simion-based Language Agnostic Code Clones

### OTHER PAPERS ALREADY READ, BUT NOT HERE YET

A347 DeepBillboard: Systematic Physical-World Testing of Autonomous Driving Systems

A433 Engineering Gender-Inclusivity into Software: Ten Teams' Tales from the Trenches

A397 Studying the Use of Java Logging Utilities in the Wild

TODO

### NOT AWARDED PAPERS

TODO: all the other 66 papers

# A347 – DeepBillboard: Systematic Physical-World Testing of Autonomous Driving Systems Type: XXX

Type:

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# A433 – Engineering Gender-Inclusivity into Software: Ten Teams' Tales from the Trenches

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# A397 – Studying the Use of Java Logging Utilities in the Wild

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# B355 – Collaborative Bug Finding for Android Apps

Type: proposal of an approach, evaluation by university course

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
| Android , mobile application |  | Eingrenzung | Titel |
| Bug finding / detection |  | Thema | Titel |
| Software testing course, students, novice andoird app testers, seniors students (4y CS) | Experience | Detail | Abstract |
| Classroom setting |  | Detail | Introduction |
| GithubIssues |  | Eingrenzung |  |
| Search engine, recommendation |  |  |  |
| Opensource Android apps  Diverse in terms of category, size, popularity and number of issues |  |  |  |
| <-> pair programming  Collaborative testing  <-> crowdsourcing |  | Abgrenzung | Related Work |

„Our evaluation results may not generalize beyond An- droid apps, the educational setting and testing approaches that we evaluated. “

# AB000 – Title

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# AB000 – Title

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# AB000 – Title

Type: XXX

|  |  |  |  |
| --- | --- | --- | --- |
| Kontextfaktor | Faktorengruppe | Zweck | Vorkommen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |