Final Scientific Report SNF Project no. 200020-181973
"Agile Software Assistance"

May 11, 2022

1 Summary of results

As software systems evolve, developers struggle to track and understand the vast amount of software information related to the software source code itself, the application domain, its quality concerns, changes to the underlying infrastructure, and the software ecosystem at large. In this project we have pursued three research tracks that explore ways in which to assist software developers in modeling, analyzing and ensuring quality of the software systems they develop.[1]

Speculative software analysis

In this track we explored the speculative analysis of software information to better support software developer tasks. Given the importance of comments in source code to understand and maintain software systems, we decided to focus on analyzing the content and quality of class comments in various programming languages.

Given the unstructured nature of comments and different standards, developers get confused about which conventions to follow while writing comments. Therefore, they post questions related to comment convention on Q&A platforms such as Stack Overflow, or Quora. We investigated these sources to gain insight into the challenges developers face. We formed a taxonomy of challenges they face and also collected the recommendations experts provided to resolve the challenges. The taxonomy can help tool developers and researchers to identify the gaps in the current tools, and help them design future tools to satisfy developer information needs. [Ran21] [RPL$^+$21b] [RBP$^+$21]

Knowing that developers become confused about comment conventions, we dived deeper into developer commenting practices regarding comment conventions. We first explored the adherence of Smalltalk comments to the commenting conventions. Then we extended the study to Java and Python. In particular, we analyzed numerous coding style guidelines and open-source projects to see the extent to which developers follow these guidelines in writing code comments. As a result, we highlighted the aspects of comments that are not well covered by the style guidelines. [BRPN21] [RAS$^+$21] [RPL$^+$21a]

Pooja Rani contributed to the track on Speculative software analysis. She successfully defended her dissertation in January 2022 [Ran22], and is currently engaged in post-doctoral research at the newly-formed Software Engineering Group at the University of Bern.

Executable domain models

In this track we explored how executable domain models can support agile development methods.

To facilitate non-technical stakeholder participation in software engineering (SE) activities, such as requirements engineering (RE) and software modeling, an appropriate methodology and corresponding tools must be developed. A traditional round-trip engineering approach based on model transformations does not scale well to modern agile development environments where numerous artifacts are produced using a range of heterogeneous tools and technologies. To boost artifact connectivity and maintain their

[1] NB: a fourth track on API client migration was cut due to the reduction in the requested funding.

consistency, we proposed to create and manage software-related artifacts as first-class entities directly in an integrated development environment (IDE). We exemplified our approach in the Glamorous Toolkit IDE, and discussed the results of a semi-structured pilot survey we conducted with practitioners and researchers to evaluate its usefulness in practice. [PCSN21] [PCSN22]

With behavior-driven development (BDD), domain experts describe system behavior and desired outcomes through natural language-like sentences, *e.g.*, using the Gherkin language. There is no empirical evidence about the actual usage of these Gherkin features. To fill this gap, we analyzed the content of 1,572 spec files extracted from 23 open-source projects. Our results shed some light on the discrepancies between the recommendations for defining Gherkin specifications and their actual adoption in practice. [CPSN22]

Nitish Patkar contributed to the track on Executable domain models. He successfully defended his dissertation in March 2022 [Pat22], and is now working as a research associate at the FHNW.

Domain-specific software quality

In this track we explored how domain-specific quality concerns and their corresponding corrective actions be effectively specified and monitored. We focused in particular on security concerns in mobile apps. In recent years the market for apps for mobile devices has exploded. These apps are not only complex but highly popular. As the average app developer is not a security expert, there is a considerable risk of security holes in apps.

We carried out a series of empirical studies to determine the security flaws commonly arising in mobile apps, and we identified numerous "security smells" in software that serve as reliable indicators for security risks. [GGFN19] [GTNG21] This work led not only to new software analyses to warn developers of security risks in their software, but also to various techniques to mitigate and counter identified security risks. [GGNG21] [GGTN20] [GNG21]

In parallel, we also investigated in depth a particular class of security risks related to the use of cryptography in software. Software developed today is increasingly connected to the internet, or is exposed to internet access. As a consequence, cryptography concerns play an increasingly important role in the development of modern software systems. Unfortunately, few developers have a deep knowledge of cryptography, and mistakes are frequently made in the usage of crypto APIs. We carried out empirical studies to assess (i) the state of the art in crypto usage, (ii) known crypto vulnerabilities in the wild, (iii) shortcomings of crypto APIs as used in practice, (iv) the state of developer crypto knowledge and experience, and (v) root causes and remedies of crypto abuses. [HAEN21] [HGK+19] [HGN20a] [HGN20b] [HGN20c] [HGN21a] [HG21] [HGN21b] [HNG21] [HNSG21] [EHNG22]

Mohammadreza Hazhirpasand contributed to the track on Domain-specific software quality. He is now working as a software security expert in the banking sector in Switzerland, and he is scheduled to defend his dissertation May 11, 2022 [Haz22].

In addition, Pascal Gadient contributed to this track, though he was not employed by project funds. He is scheduled to defend his dissertation on May 10, 2022 [Gad22].

Changes to the research plan and important Events

There were no major deviations from the research plan.

2 Research output

All reported publications are available electronically from the project's home page:

`http://scg.unibe.ch/asa3`

Journal papers

[BSGN21] Arianna Blasi, Nataliia Stulova, Alessandra Gorla, and Oscar Nierstrasz. RepliComment: identifying clones in code comments. *Journal of Systems & Software*, page 111069, 2021. URL: `http://scg.unibe.ch/archive/papers/Blas21a-RepliComment.pdf`, `doi:10.1016/j.jss.2021.111069`.

[GGFN19] Pascal Gadient, Mohammad Ghafari, Patrick Frischknecht, and Oscar Nierstrasz. Security code smells in Android ICC. *Empirical Software Engineering*, 24:3046–3076, 2019. URL: `http://scg.unibe.ch/archive/papers/Gadi18a.pdf`, `doi:10.1007/s10664-018-9673-y`.

[RPL+21a] Pooja Rani, Sebastiano Panichella, Manuel Leuenberger, Andrea Di Sorbo, and Oscar Nierstrasz. How to identify class comment types? A multi-language approach for class comment classification. *Journal of Systems and Software*, 181:111047, 2021. URL: `http://scg.unibe.ch/archive/papers/Rani21d.pdf`, `arXiv:2107.04521`, `doi:https://doi.org/10.1016/j.jss.2021.111047`.

[RPL+21b] Pooja Rani, Sebastiano Panichella, Manuel Leuenberger, Mohammad Ghafari, and Oscar Nierstrasz. What do class comments tell us? An investigation of comment evolution and practices in Pharo Smalltalk. *Empirical Software Engineering*, 26(6):1–49, 2021. URL: `http://scg.unibe.ch/archive/papers/Rani21b.pdf`, `arXiv:2005.11583`, `doi:10.1007/s10664-021-09981-5`.

Conference papers

[BRPN21] Mathias Birrer, Pooja Rani, Sebastiano Panichella, and Oscar Nierstrasz. Makar: A framework for multi-source studies based on unstructured data. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 577–581, 2021. URL: `http://scg.unibe.ch/archive/papers/Rani21c.pdf`, `doi:10.1109/SANER50967.2021.00069`.

[EHNG22] Arash Ale Ebrahim, Mohammadreza Hazhirpasand, Oscar Nierstrasz, and Mohammad Ghafari. FuzzingDriver: the missing dictionary to increase code coverage in fuzzers. In *29th edition of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, March 2022. URL: `http://scg.unibe.ch/archive/papers/Ebra22a.pdf`.

[GEN19] M. Ghafari, M. Eggiman, and O. Nierstrasz. Testability first! In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–6, sep 2019. URL: `http://scg.unibe.ch/archive/papers/Ghaf19a-TestabilityFirst.pdf`, `doi:10.1109/ESEM.2019.8870170`.

[GGNG21] Pascal Gadient, Pascal Gerig, Oscar Nierstrasz, and Mohammad Ghafari. Phish what you wish. In *21st IEEE International Conference on Software Quality, Reliability, and Security (QRS)*, December 2021. URL: `http://scg.unibe.ch/archive/papers/Gadi21b.pdf`, `doi:10.1109/QRS54544.2021.00113`.

[GGTN20] Pascal Gadient, Mohammad Ghafari, Marc-Andrea Tarnutzer, and Oscar Nierstrasz. Web APIs in Android through the lens of security. In *27th edition of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, March 2020. URL: `http://scg.unibe.ch/archive/papers/Gadi20a.pdf`, `doi:10.1109/SANER48275.2020.9054850`.

[GNG21] Pascal Gadient, Oscar Nierstrasz, and Mohammad Ghafari. Security header fields in HTTP clients. In *21st IEEE International Conference on Software Quality, Reliability, and Security (QRS)*, December 2021. URL: `http://scg.unibe.ch/archive/papers/Gadi21c.pdf`, `doi:10.1109/QRS54544.2021.00020`.

[GTNG21] Pascal Gadient, Marc-Andrea Tarnutzer, Oscar Nierstrasz, and Mohammad Ghafari. Security smells pervade mobile app servers. In *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, October 2021. URL: `http://scg.unibe.ch/archive/papers/Gadi21a.pdf`, `doi:10.1145/3475716.3475780`.

[HAEN21] Mohammadreza Hazhirpasand, Arash Ale Ebrahim, and Oscar Nierstrasz. Stopping DNS rebinding attacks in the browser. In *Proceedings of the 7th International Conference on Information Systems Security and Privacy - ICISSP*, 2021. URL: `http://scg.unibe.ch/archive/papers/Hazh21a.pdf`, `doi:10.5220/0010310705960603`.

[HG21] Mohammadreza Hazhirpasand and Mohammad Ghafari. Cryptography vulnerabilities on HackerOne. In *21st IEEE International Conference on Software Quality, Reliability, and Security (QRS)*, pages 18–27, December 2021. URL: `http://scg.unibe.ch/archive/papers/Hazh21f.pdf`, `doi:10.1109/QRS54544.2021.00013`.

[HGK+19] M. Hazhirpasand, M. Ghafari, S. Krüger, E. Bodden, and O. Nierstrasz. The impact of developer experience in using Java cryptography. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–6, sep 2019. URL: `http://scg.unibe.ch/archive/papers/Hazh19aJCA-Impact.pdf`, `doi:10.1109/ESEM.2019.8870184`.

[HGN20a] Mohammadreza Hazhirpasand, Mohammad Ghafari, and Oscar Nierstrasz. CryptoExplorer: An interactive web platform supporting secure use of cryptography APIs. In *27th edition of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 632–636, March 2020. URL: http://scg.unibe.ch/archive/papers/Hazh20a.pdf, doi:10.1109/SANER48275.2020.9054799.

[HGN20b] Mohammadreza Hazhirpasand, Mohammad Ghafari, and Oscar Nierstrasz. Java cryptography uses in the wild. In *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2020. URL: http://scg.unibe.ch/archive/papers/Hazh20c.pdf, doi:10.1145/3382494.3422166.

[HGN20c] Mohammadreza Hazhirpasand, Mohammad Ghafari, and Oscar Nierstrasz. Tricking Johnny into granting web permissions. In *Proceedings of the Evaluation and Assessment in Software Engineering*, EASE 2020, pages 276–281, New York, NY, USA, 2020. Association for Computing Machinery. URL: http://scg.unibe.ch/archive/papers/Hazh20b.pdf, doi:10.1145/3383219.3383248.

[HGN21a] Mohammadreza Hazhirpasand, Mohammad Ghafari, and Oscar Nierstrasz. Crypto experts advise what they adopt. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pages 179–184, 2021. URL: http://scg.unibe.ch/archive/papers/Hazh21e.pdf, doi:10.1109/ASEW52652.2021.00044.

[HGN21b] Mohammadreza Hazhirpasand, Mohammad Ghafari, and Oscar Nierstrasz. Worrisome patterns in developers: A survey in cryptography. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pages 185–190, 2021. URL: http://scg.unibe.ch/archive/papers/Hazh21d.pdf, doi:10.1109/ASEW52652.2021.00045.

[HNG21] Mohammadreza Hazhirpasand, Oscar Nierstrasz, and Mohammad Ghafari. Dazed and confused: What's wrong with crypto libraries? In *2021 18th International Conference on Privacy, Security and Trust (PST)*, pages 1–6, 2021. URL: http://scg.unibe.ch/archive/papers/Hazh21b.pdf, doi:10.1109/PST52912.2021.9647786.

[HNSG21] Mohammadreza Hazhirpasand, Oscar Nierstrasz, Mohammadhossein Shabani, and Mohammad Ghafari. Hurdles for developers in cryptography. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 659–663, 2021. URL: http://scg.unibe.ch/archive/papers/Hazh21c.pdf, doi:10.1109/ICSME52107.2021.00076.

[Leu19] Manuel Leuenberger. Exploring example-driven migration. In *Proceedings of the Conference Companion of the 3rd International Conference on Art, Science, and Engineering of Programming*, Programming '19, pages 29:1–29:3, New York, NY, USA, April 2019. ACM. URL: http://scg.unibe.ch/archive/papers/Leue19a.pdf, doi:10.1145/3328433.3328463.

[MHB+19] Leonel Merino, Mario Hess, Alexandre Bergel, Oscar Nierstrasz, and Daniel Weiskopf. PerfVis: Pervasive visualization in immersive augmented reality for performance awareness. In *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*, ICPE '19, pages 13–16, New York, NY, USA, 2019. ACM. URL: http://scg.unibe.ch/archive/papers/Meri19a-perfvis.pdf, doi:10.1145/3302541.3313104.

[MKNW19] Leonel Merino, Ekaterina Kozlova, Oscar Nierstrasz, and Daniel Weiskopf. VISON: An ontology-based approach for software visualization tool discoverability. In *VISSOFT'19: Proceedings of the 7th IEEE Working Conference on Software Visualization*. IEEE, 2019. URL: http://scg.unibe.ch/archive/papers/Meri19b-vison.pdf, doi:10.1109/VISSOFT.2019.00014.

[Pat20] Nitish Patkar. Moldable requirements. In *Benevol'20*, 2020. URL: http://scg.unibe.ch/archive/papers/Patk20c.pdf.

[PCSN21] Nitish Patkar, Andrei Chis, Nataliia Stulova, and Oscar Nierstrasz. Interactive behavior-driven development: a low-code perspective. In *Proceedings of the 24rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. ACM, 2021. URL: http://scg.unibe.ch/archive/papers/Patk21a.pdf, doi:10.1109/MODELS-C53483.2021.00024.

[PCSN22] Nitish Patkar, Andrei Chis, Nataliia Stulova, and Oscar Nierstrasz. First-class artifacts as building blocks for live in-ide documentation. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022. URL: http://scg.unibe.ch/archive/papers/Patk22a.pdf.

[PGGN19] Nitish Patkar, Pascal Gadient, Mohammad Ghafari, and Oscar Nierstrasz. Towards a catalogue of mobile elicitation techniques. In *25th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, 2019. URL: http://scg.unibe.ch/archive/papers/Patk19a.pdf, doi:10.1007/978-3-030-15538-4_20.

[PGNH20] Nitish Patkar, Mohammad Ghafari, Oscar Nierstrasz, and Sofija Hotomski. Caveats in eliciting mobile app requirements. In *Proceedings of the Evaluation and Assessment in Software Engineering*, EASE 2020, pages 180–189, New York, NY, USA, 2020. Association for Computing Machinery. URL: http://scg.unibe.ch/archive/papers/Patk20.pdf, doi:10.1145/3383219.3383238.

[PMN20] Nitish Patkar, Leonel Merino, and Oscar Nierstrasz. Towards requirements engineering with immersive augmented reality. In *Proc. Programming'20 Companion*, pages 55–60. ACM, 2020. URL: http://scg.unibe.ch/archive/papers/Patk20b.pdf, doi:10.1145/3397537.3398472.

[Ran21] Pooja Rani. Speculative analysis for quality assessment of code comments. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 299–303, 2021. URL: http://scg.unibe.ch/archive/papers/Rani21a.pdf, arXiv:2102.09605, doi:10.1109/ICSE-Companion52605.2021.00132.

[RAS+21] Pooja Rani, Suada Abukar, Nataliia Stulova, Alexander Bergel, and Oscar Nierstrasz. Do comments follow commenting conventions? A case study in Java and Python. In *2021 IEEE 21st International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2021. URL: http://scg.unibe.ch/archive/papers/Rani21f.pdf, arXiv:2108.10766, doi:10.1109/SCAM52516.2021.00028.

[RBP+21] Pooja Rani, Mathias Birrer, Sebastiano Panichella, Mohammad Ghafari, and Oscar Nierstrasz. What do developers discuss about code comments? In *2021 IEEE 21st International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2021. URL: http://scg.unibe.ch/archive/papers/Rani21e.pdf, arXiv:2108.07648, doi:10.1109/SCAM52516.2021.00027.

[SBGN20] Nataliia Stulova, Arianna Blasi, Alessandra Gorla, and Oscar Nierstrasz. Towards detecting inconsistent comments in java source code automatically. In *2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 65–69. IEEE, 2020. URL: http://scg.unibe.ch/archive/papers/Stul20b-InconsistentComments.pdf, doi:10.1109/SCAM51674.2020.00012.

International Workshop papers

[CPSN22] Adwait Chandorkar, Nitish Patkar, Andrea Di Sorbo, and Oscar Nierstrasz. An exploratory study on the usage of gherkin features in open-source projects. In *5th Workshop on Validation, Analysis and Evolution of Software Tests (VST 2022, co-located with SANER 2022)*. IEEE, March 2022. URL: http://scg.unibe.ch/archive/papers/Patk22b.pdf.

PhD Theses

[Gad22] Pascal Gadient. *The Dilemma of Security Smells and How to Escape It*. PhD thesis, University of Bern, May 2022. URL: http://scg.unibe.ch/archive/phd/Gadi22a.pdf.

[Haz22] Mohammadreza Hazhirpasand. *The Bumpy Relationship of Developers and Cryptography*. PhD thesis, University of Bern, May 2022. URL: http://scg.unibe.ch/archive/phd/Hazh22a.pdf.

[Pat22] Nitish Patkar. *Supporting Multiple Stakeholders in Agile Development*. PhD thesis, University of Bern, March 2022. URL: http://scg.unibe.ch/archive/phd/patkar-phd.pdf.

[Ran22] Pooja Rani. *Assessing Comment Quality in Object-Oriented Languages*. PhD thesis, University of Bern, January 2022. URL: http://scg.unibe.ch/archive/phd/rani-phd.pdf.