Supplemental Material: Laboratory Packages for Human–Oriented Experiments in Software Engineering: A Structure Proposal

CATHY GUEVARA-VEGA, BEATRIZ BERNÁRDEZ, MARGARITA CRUZ, AMADOR DURÁN, ANTONIO RUIZ-CORTÉS, MARTÍN SOLARI

1 PRIMARY STUDIES

EMPIRICAL SOFTWARE ENGINEERING (EMSE)

[1] J. Maldonado, J. Carver, F. Shull, S. Fabbri, E. Doria, L. Martimiano, M. Mendonça, and V. Basili. 2006. Perspective-based reading: a replicated experiment focused on individual reviewer effectiveness. *Empirical Software Engineering*, 11, 1, 119–142. DOI: 10.1007/s10664-006-5967-6.

[2] J. Cruz-Lemus, M. Genero, M. Manso, S. Morasca, and M. Piattini. 2009. Assessing the understandability of uml statechart diagrams with composite states—a family of empirical studies. English. *Empirical Software Engineering*, 14, 6, 685–719. DOI: 10.1007/s10664-009-9106-z.

[3] J. Feigenspan, C. Kästner, S. Apel, J. Liebig, M. Schulze, R. Dachselt, M. Papendieck, T. Leich, and G. Saake. 2013. Do background colors improve program comprehension in the #ifdef hell? *Empirical Softw. Engg.*, 18, 4, (August 2013), 699–745. ISSN: 1382-3256. DOI: 10.1007/s10664-012-9208-x.

[4] I. Reinhartz-Berger and A. Sturm. 2014. Comprehensibility of uml-based software product line specifications. *Empirical Softw. Engg.*, 19, 3, (June 2014), 678–713. ISSN: 1382-3256. DOI: 10.1007/s10664-012-9234-8.

[5] C. Apa, O. Dieste, E. Espinosa, and E. Fonseca. 2014. Effectiveness for detecting faults within and outside the scope of testing techniques: an independent replication. *Empirical Software Engineering*, 19, 378–417. DOI: 10.1007/s10664-013-9267-7.

[6] J. Siegmund, C. Kästner, J. Liebig, S. Apel, and S. Hanenberg. 2014. Measuring and modeling programming experience. *Empirical Software Engineering*, 19, 5, 1299–1334. DOI: 10.1007/s10664-013-9286-4.

[7] W. Wu, A. Serveaux, Y. Guéhéneuc, and G. Antoniol. 2015. The impact of imperfect change rules on framework api evolution identification: an empirical study. *Empirical Softw. Engg.*, 20, 4, (August 2015), 1126–1158. ISSN: 1382-3256. DOI: 10.1007/s10664-014-9317-9.

[8] Y. Tu, E. Tempero, and C. Thomborson. 2016. An experiment on the impact of transparency on the effectiveness of requirements documents. *Empirical Software Engineering*, 21, (February 2016). DOI: 10.1007/s10664-015-9374-8.

[9] A. Fernández-Sáez, M. Genero, D. Caivano, and M. Chaudron. 2016. Does the level of detail of uml diagrams affect the maintainability of source code?: a family of experiments. *Empirical software engineering*, 21, 1, 212–259. DOI: 10.1007/s10664-014-9354-4.

[10] J. Grigera, A. Garrido, J. Panach, D. Distante, and G. Rossi. 2016. Assessing refactorings for usability in e-commerce applications. *Empirical Softw. Engg.*, 21, 3, (June 2016), 1224–1271. ISSN: 1382-3256. DOI: 10.1007/s10664-015-9384-6.

[11] J. King, J. Stallings, M. Riaz, and L. Williams. 2017. To log, or not to log: using heuristics to identify mandatory log events-a controlled experiment. *Empirical Softw. Engg.*, 22, 5, (October 2017), 2684–2717. ISSN: 1382-3256. DOI: 10.1007/s10664-016-9449-1.

[12] M. Riaz, J. King, J. Slankas, L. Williams, F. Massacci, C. Quesada-López, and M. Jenkins. 2017. Identifying the implied: findings from three differentiated replications on the use of security requirements templates. *Empirical Softw. Engg.*, 22, 4, (August 2017), 2127–2178. ISSN: 1382-3256. DOI: 10.1007/s10664-016-9481-1.

[13] A. Przybyłek. 2018. An empirical study on the impact of aspectj on software evolvability. *Empirical Softw. Engg.*, 23, 4, (August 2018), 2018–2050. ISSN: 1382-3256. DOI: 10.1007/s10664-017-9580-7.

[14] F. Ricca, M. Torchiano, M. Leotta, A. Tiso, G. Guerrini, and G. Reggio. 2018. On the impact of state-based model-driven development on maintainability: a family of experiments using unimod. *Empirical Softw. Engg.*, 23, 3, (June 2018), 1743–1790. ISSN: 1382-3256. DOI: 10.1007/s10664-017-9563-8.

[15] I. McChesney and R. Bond. 2019. Eye tracking analysis of computer program comprehension in programmers with dyslexia. *Empirical Softw. Engg.*, 24, 3, (June 2019), 1109–1154. ISSN: 1382-3256. DOI: 10.1007/s10664-018-9649-y.

[16] I. Salman, B. Turhan, and S. Vegas. 2019. A controlled experiment on time pressure and confirmation bias in functional software testing. *Empirical Softw. Engg.*, 24, 4, (August 2019), 1727–1761. ISSN: 1382-3256. DOI: 10.1007/s10664-018-9668-8.

[17] A. Viticchié, L. Regano, C. Basile, M. Torchiano, M. Ceccato, and P. Tonella. 2020. Empirical assessment of the effort needed to attack programs protected with client/server code splitting. *Empirical Software Engineering*, 25, (January 2020). DOI: 10.1007/s10664-019-09738-1.

[18] A. Santos, S. Vegas, O. Dieste, F. Uyaguari, A. Tosun, D. Fucci, B. Turhan, G. Scanniello, S. Romano, I. Karac, M. Kuhrmann, V. MandiÄ‡, R. RamaÄ, D. Pfahl, C. Engblom, J. Kyykka, K. Rungi, C. Palomeque, J. Spisak, M. Oivo, and N. Juristo. 2021. A family of experiments on test-driven development. *Empirical Software Engineering*, 1573–7616. DOI: 10.1007/s10664-020-09895-8.

[19] J. Olsson, E. Risfelt, T. Besker, A. Martini, and R. Torkar. 2021. Measuring affective states from technical debt: a psychoempirical software engineering experiment. *Empirical Software Engineering*, 26, (September 2021). DOI: 10.1007/s10664-021-09998-w.

[20] J. da Costa, R. Gheyi, M. Ribeiro, S. Apel, V. Alves, B. Fonseca, F. Medeiros, and A. Garcia. 2021. Evaluating refactorings for disciplining #ifdef annotations: an eye tracking study with novices. *Empirical Software Engineering*, 26, (September 2021). DOI: 10.1007/s10664-021-10002-8.

[21] D. Wu, X. Jing, H. Zhang, B. Li, Y. Xie, and B. Xu. 2021. Generating api tags for tutorial fragments from stack overflow. *Empirical Software Engineering*, 26, (July 2021). DOI: 10.1007/s10664-021-09962-8.

[22] C. Gralha, M. Goulão, and J. Araújo. 2020. Are there gender differences when interacting with social goal models? *Empirical Software Engineering*, 25, (November 2020), 1–38. DOI: 10.1007/s10664-020-09883-y.

[23] R. Silva, C. Roy, M. Rahman, K. Schneider, K. Paixão, C. Dantas, and M. Maia. 2020. Crokage: effective solution recommendation for programming tasks by leveraging crowd knowledge. *Empirical Software Engineering*, 25, (November 2020), 1–52. DOI: 10.1007/s10664-020-09863-2.

[24] R. Jolak, M. Savary-Leblanc, M. Dalibor, A. Wortmann, R. Hebig, J. Vincur, I. Polášek, X. Pallec, S. Gérard, and M. Chaudron. 2020. Software engineering whispers: the effect of textual vs. graphical software design descriptions on software design communication. *Empirical Software Engineering*, 25, (November 2020). DOI: 10.1007/s10664-020-09835-6.

[25] E. Aktaş and C. Yilmaz. 2020. Automated issue assignment: results and insights from an industrial case. *Empirical Software Engineering*, (July 2020). DOI: 10.1007/s10664-020-09846-3.

[26] T. LaToza, M. Arab, D. Loksa, and A. Ko. 2020. Explicit programming strategies. *Empirical Software Engineering*, 25, (July 2020). DOI: 10.1007/s10664-020-09810-1.

[27] R. Morales, F. Khomh, and G. Antoniol. 2020. Repor: mimicking humans on refactoring tasks. are we there yet? *Empirical Software Engineering*, 25, (July 2020). DOI: 10.1007/s10664-020-09826-7.

[28] S. Vegas, P. Riofrío, E. Marcos, and N. Juristo. 2020. On (mis)perceptions of testing effectiveness: an empirical study. *Empirical Software Engineering*, 25, (July 2020). DOI: 10.1007/s10664-020-09805-y.

[29] S. Fakhoury, D. Roy, Y. Ma, V. Arnaoudova, and O. Adesope. 2020. Measuring the impact of lexical and structural inconsistencies on developers' cognitive load during bug localization. *Empirical Software Engineering*, 25, (May 2020). DOI: 10.1007/s10664-019-09751-4.

[30] C. Vassallo, S. Proksch, T. Zemp, and H. Gall. 2020. Every build you break: developer-oriented assistance for build failure resolution. *Empirical Software Engineering*, 25, (May 2020). DOI: 10.1007/s10664-019-09765-y.

[31] L. Allodi, M. Cremonini, F. Massacci, and W. Shim. 2020. Measuring the accuracy of software vulnerability assessments: experiments with students and professionals. *Empirical Software Engineering*, 25, (March 2020). DOI: 10.1007/s10664-019-09797-4.

[32] R. Abdalkareem, V. Oda, S. Mujahid, and E. Shihab. 2020. On the impact of using trivial packages: an empirical case study on npm and pypi. *Empirical Software Engineering*, 25, (March 2020). DOI: 10.1007/s10664-019-09792-9.

[33] M. Sayagh, N. Kerzazi, F. Petrillo, K. Bennani, and B. Adams. 2020. What should your run-time configuration framework do to help developers? *Empirical Software Engineering*, 25, 2, 1259–1293. DOI: 10.1007/s10664-019-09790-x.

[34] C. Vassallo, S. Panichella, F. Palomba, S. Proksch, H. Gall, and A. Zaidman. 2020. How developers engage with static analysis tools in different contexts. *Empirical Software Engineering*, 25, 2, 1419–1457. DOI: 10.1007/s10664-019-09750-5.

[35] W. Mahmood, D. Strüber, A. Anjorin, and T. Berger. 2022. Effects of variability in models: a family of experiments. *Empirical Software Engineering*, 27, 3, 1–38. DOI: 10.1007/s10664-021-10112-3.

[36] B. Penzenstadler, R. Torkar, and C. Martinez. 2022. Take a deep breath: benefits of neuroplasticity practices for software developers and computer workers in a family of experiments. *Empirical Software Engineering*, 27, 4, 1–64. DOI: 10.1007/s10664-022-10148-z.

[37] J. Pereira, F. Brito, and G. Carneiro. 2022. Crowdsmelling: a preliminary study on using collective knowledge in code smells detection. *Empirical Software Engineering*, 27, 3, 1–35. DOI: 10.1007/s10664-021-10110-5.

2 PRIMARY STUDIES

INFORMATION AND SOFTWARE TECHNOLOGY (IST)

[1] H. Eichelberger and K. Schmid. 2009. Guidelines on the aesthetic quality of uml class diagrams. *Information and Software Technology*, 51, (December 2009), 1686–1698. DOI: 10.1016/j.infsof.2009.04.008.

[2] J. Cruz-Lemus, M. Genero, D. Caivano, S. Abrahão, E. Insfrán, and J. Carsí. 2011. Assessing the influence of stereotypes on the comprehension of UML sequence diagrams: A family of experiments. *Information and Software Technology*, 53, 12, 1391–1403. ISSN: 09505849. DOI: 10.1016/J.INFSOF.2011.07.002.

[3] A. Fernández-Sáez, M. Genero, M. Chaudron, D. Caivano, and I. Ramos. 2014. Are forward designed or reverse-engineered uml diagrams more helpful for code maintenance?: a family of experiments. *Information and Software Technology*, 57, 644–663. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2014.05.014.

[4] F. Häser, M. Felderer, and R. Breu. 2016. Is business domain language support beneficial for creating test case specifications: a controlled experiment. *Information and software technology*, 79, 52–62.

[5] M. Otero and J. Dolado. 2004. Evaluation of the comprehension of the dynamic modeling in uml. *Information and Software Technology*, 46, 1, 35–53. ISSN: 0950-5849. DOI: 10.1016/S0950-5849(03)00108-3.

[6] F. Siqueira. 2018. Comparing the comprehensibility of requirements models: an experiment replication. *Information and Software Technology*, 96, (November 2018). DOI: 10.1016/j.infsof.2017.11.002.

[7] E. Souza, A. Moreira, J. Araújo, S. Abrahão, E. Insfran, and D. S. Da Silveira. 2018. Comparing business value modeling methods: a family of experiments. *Information and Software Technology*, 104, 179–193. DOI: 10.1016/j.infsof.2018.08.001.

[8] M. Campusano, J. Fabry, and A. Bergel. 2019. Live programming in practice: a controlled experiment on state machines for robotic behaviors. *Information and Software Technology*, 108, 99–114. DOI: 10.1016/j.infsof.2018.12.008.

[9] J. Zubcoff, I. Garrigós, S. Casteleyn, J.N. Mazón, J.A. Aguilar, and F. Gomariz-Castillo. 2019. Evaluating different i*-based approaches for selecting functional requirements while balancing and optimizing non-functional requirements: a controlled experiment. *Information and Software Technology*, 106, 68–84.

[10] F. Pérez, J. Echeverría, R. Lapeña, and C. Cetina. 2020. Comparing manual and automated feature location in conceptual models: a controlled experiment. *Information and Software Technology*, 125, 106337. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2020.106337.

[11] F. Dalpiaz, P. Gieske, and A. Sturm. 2021. On deriving conceptual models from user requirements: an empirical study. *Information and Software Technology*, 131, 106484. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2020.106484.

[12] Á. Domingo, J. Echeverría, Ó. Pastor, and C. Cetina. 2021. Evaluating the influence of scope on feature location. *Information and Software Technology*, 140, 106674. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106674.

[13] A. Lavalle, A. Maté, J. Trujillo, M. Teruel, and S. Rizzi. 2021. A methodology to automatically translate user requirements into visualizations: experimental validation. *Information and Software Technology*, 136, 106592. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106592.

[14] J. de la Vara, B. Marín, C. Ayora, and G. Giachetti. 2020. An empirical evaluation of the use of models to improve the understanding of safety compliance needs. *Information and Software Technology*, 126, 106351. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2020.106351.

[15] D. Firmenich, S. Firmenich, G. Rossi, M. Wimmer, I. Garrigós, and C. González-Mora. 2022. Engineering web augmentation software: a development method for enabling end-user maintenance. *Information and Software Technology*, 141, 106735. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106735.

[16] F. Kifetew, A. Perini, A. Susi, A. Siena, D. Muñante, and I. Morales-Ramirez. 2021. Automating user-feedback driven requirements prioritization. *Information and Software Technology*, 138, 106635. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106635.

[17] L. Ardito, R. Coppola, G. Malnati, and M. Torchiano. 2020. Effectiveness of kotlin vs. java in android app development tasks. *Information and Software Technology*, 127, 106374. DOI: 10.1016/j.infsof.2020.106374.

[18] M. Urbieta, L. Antonelli, G. Rossi, and J. do Prado Leite. 2020. The impact of using a domain language for an agile requirements management. *Information and Software Technology*, 127, 106375. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2020.106375.

[19] M. Gil, M. Albert, J. Fons, and V. Pelechano. 2020. Engineering human-in-the-loop interactions in cyber-physical systems. *Information and Software Technology*, 126, 106349. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2020.106349.

[20] A. Alazba and H. Aljamaan. 2021. Code smell detection using feature selection and stacking ensemble: an empirical investigation. *Information and Software Technology*, 138, 106648. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106648.

[21] P. Valderas, V. Torres, and V. Pelechano. 2020. A microservice composition approach based on the choreography of bpmn fragments. *Information and Software Technology*, 127, 106370. DOI: 10.1016/j.infsof.2020.106370.

[22] M. de Freitas Farias, M. de Mendonça Neto, M. Kalinowski, and R. Oliveira. 2020. Identifying self-admitted technical debt through code comment analysis with a contextualized vocabulary. *Information and Software Technology*, 121, 106270. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2020.106270.

[23] S. Kuttal, X. Chen, Z. Wang, S. Balali, and A. Sarma. 2021. Visual resume: exploring developers' online contributions for hiring. *Information and Software Technology*, 138, 106633. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106633.

[24] D. dos Santos, E. de Almeida, and I. Ahmed. 2022. Investigating replication challenges through multiple replications of an experiment. *Information and Software Technology*, 147, 106870. DOI: 10.1016/j.infsof.2022.106870.

[25] V. Dashuber and M. Philippsen. 2022. Trace visualization within the software city metaphor: controlled experiments on program comprehension. *Information and Software Technology*, 150, 106989. DOI: 10.1016/j.infsof.2022.106989.

[26] E. M. Grua, M. De Sanctis, I. Malavolta, M. Hoogendoorn, and P. Lago. 2022. An evaluation of the effectiveness of personalization and self-adaptation for e-health apps. *Information and Software Technology*, 146, 106841. DOI: 10.1016/j.infsof.2022.106841.

[27] T. Ferreira, S. Vergilio, and M. Kessentini. 2022. Variability testing of software product line: a preference-based dimensionality reduction approach. *Information and Software Technology*, 152, 107031. DOI: 10.1016/j.infsof.2022.107031.

3 PRIMARY STUDIES

JOURNAL OF SYSTEMS AND SOFTWARE (JSS)

[1] M. Otero and J. Dolado. 2005. An empirical comparison of the dynamic modeling in oml and uml. *Journal of Systems and Software*, 77, 2, 91–102. ISSN: 0164-1212. DOI: 10.1016/j.jss.2004.11.022.

[2] D. Caivano, M. Fernández-Ropero, R. Pérez-Castillo, M. Piattini, and M. Scalera. 2018. Artifact-based vs. human-perceived understandability and modifiability of refactored business processes: an experiment. *Journal of Systems and Software*, 144, 143–164. ISSN: 0164-1212. DOI: 10.1016/j.jss.2018.06.026.

[3] P. Paulweber, G. Simhandl, and U. Zdun. 2021. On the understandability of language constructs to structure the state and behavior in abstract state machine specifications: a controlled experiment. *Journal of Systems and Software*, 178, 110987. ISSN: 0164-1212. DOI: 10.1016/j.jss.2021.110987.

[4] M. Nasir, N. Ikram, and Z. Jalil. 2022. Usability inspection: novice crowd inspectors versus expert. *Journal of Systems and Software*, 183, 111122. ISSN: 0164-1212. DOI: 10.1016/j.jss.2021.111122.

[5] A. Mukhtar, B. Hofer, D. Jannach, and F. Wotawa. 2022. Spreadsheet debugging: the perils of tool over-reliance. *Journal of Systems and Software*, 184, 111119. ISSN: 0164-1212. DOI: 10.1016/j.jss.2021.111119.

[6] K. Nafi, B. Roy, C. Roy, and K. Schneider. 2020. A universal cross language software similarity detector for open source software categorization. *Journal of Systems and Software*, 162, 110491. DOI: 10.1016/j.jss.2019.110491.

[7] F. Corradini, A. Morichetta, A. Polini, B. Re, L. Rossi, and F. Tiezzi. 2020. Correctness checking for bpmn collaborations with sub-processes. *Journal of Systems and Software*, 166, 110594. DOI: 10.1016/j.jss.2020.110594.

[8] M. Pan, Y. Lu, Y. Pei, T. Zhang, J. Zhai, and X. Li. 2020. Effective testing of android apps using extended ifml models. *Journal of Systems and Software*, 159, 110433. DOI: 10.1016/j.jss.2019.110433.

[9] K. Bernsmed, D. Soares Cruzes, M. Gilje Jaatun, and M. Iovan. 2022. Adopting threat modelling in agile software development projects. *Journal of Systems and Software*, 183, 111090. ISSN: 0164-1212. DOI: 10.1016/j.jss.2021.111090.

[10] S. Teixeira, B. Agrizzi, J. Pereira, S. Rossetto, I. Pereira, P. Costa, A. Branco, and R. Martinelli. 2020. Laura architecture: towards a simpler way of building situation-aware and business-aware iot applications. *Journal of Systems and Software*, 161, (December 2020), 110494. DOI: 10.1016/j.jss.2019.110494.

[11] Q. Shen, S. Wu, Y. Zou, Z. Zhu, and B. Xie. 2020. From api to nli: a new interface for library reuse. *Journal of Systems and Software*, 169, 110728. ISSN: 0164-1212. DOI: 10.1016/j.jss.2020.110728.

[12] H. Vrzakova, A. Begel, L. Mehtätalo, and R. Bednarik. 2020. Affect recognition in code review: an in-situ biometric study of reviewer's affect. *Journal of Systems and Software*, 159, 110434. DOI: 10.1016/j.jss.2019.110434.

[13] O. Karras, K. Schneider, and S. A Fricker. 2020. Representing software project vision by means of video: a quality model for vision videos. *journal of Systems and Software*, 162, 110479. DOI: 10.1016/j.jss.2019.110479.

[14] M. Baldassarre, D. Caivano, D. Fucci, N. Juristo, S. Romano, G. Scanniello, and B. Turhan. 2021. Studying test-driven development and its retainment over a six–month time span. *Journal of Systems and Software*, 176, 110937. ISSN: 0164-1212. DOI: 10.1016/j.jss.2021.110937.

[15] J. Blanco and D. Lucrédio. 2021. A holistic approach for cross-platform software development. *Journal of Systems and Software*, 179, (September 2021), 110985. DOI: 10.1016/j.jss.2021.110985.

[16] R. Pereira, T. Carção, M. Couto, C. Jácome, J. Fernandes, and J. Saraiva. 2020. Spelling out energy leaks: aiding developers locate energy inefficient code. *Journal of Systems and Software*, 161, 110463. ISSN: 0164-1212. DOI: 10.1016/j.jss.2019.110463.

[17] A. Di Sorbo, S. Laudanna, A. Vacca, C. A Visaggio, and G. Canfora. 2022. Profiling gas consumption in solidity smart contracts. *Journal of Systems and Software*, 186, 111193. DOI: 10.1016/j.jss.2021.111193.

[18] M. Rahimi and M. Vierhauser. 2022. Visualization of aggregated information to support class-level software evolution. *Journal of Systems and Software*, 192, 111421. DOI: 10.1016/j.jss.2022.111421.

4 PRIMARY STUDIES

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING (TSE)

[1] M. Perepletchikov and C. Ryan. 2011. A controlled experiment for evaluating the impact of coupling on the maintainability of service-oriented software. *IEEE Transactions on Software Engineering*, 37, 4, 449–465. DOI: 10.1109/TSE.2010.61.

[2] L. Prechelt. 2011. Plat_forms: a web development platform comparison by an exploratory experiment searching for emergent platform properties. *IEEE Transactions on Software Engineering*, 37, 1, 95–108. DOI: 10.1109/TSE.2010.22.

[3] J. Wilkerson, J. Nunamaker, and R. Mercer. 2012. Comparing the defect reduction benefits of code inspection and test-driven development. *IEEE Transactions on Software Engineering*, 38, 3, 547–560. DOI: 10.1109/TSE.2011.46.

[4] J. Krein, L. Prechelt, N. Juristo, A. Nanthaamornphong, J. Carver, S. Vegas, C. Knutson, K. Seppi, and D. Eggett. 2016. A multi-site joint replication of a design patterns experiment using moderator variables to generalize across contexts. *IEEE Transactions on Software Engineering*, 42, 4, 302–321. DOI: 10.1109/TSE.2015.2488625.

[5] A. Aranda, O. Dieste, and N. Juristo. 2016. Effect of domain knowledge on elicitation effectiveness: an internally replicated controlled experiment. *IEEE Transactions on Software Engineering*, 42, 5, 427–451. DOI: 10.1109/TSE.2015.2494588.

[6] J. Panach, Ó. Dieste, B. Marín, S. España, S. Vegas, Ó. Pastor, and N. Juristo. 2018. Evaluating model-driven development claims with respect to quality: a family of experiments. *IEEE Transactions on Software Engineering*, 47, 1, 130–145. DOI: 10.1109/TSE.2018.2884706.

[7] E. Karac, B. Turhan, and N. Juristo. 2019. A controlled experiment with novice developers on the impact of task description granularity on software quality in test-driven development. *IEEE Transactions on Software Engineering*, 47, 1315–1330. DOI: 10.1109/TSE.2019.2920377.

[8] A. Tosun, O. Dieste, S. Vegas, D. Pfahl, K. Rungi, and N. Juristo. 2019. Investigating the impact of development task on external quality in test-driven development: an industry experiment. *IEEE Transactions on Software Engineering*, 47, 11, 2438–2456. DOI: 10.1109/TSE.2019.2949811.

[9] C. Czepa and U. Zdun. 2020. On the understandability of temporal properties formalized in linear temporal logic, property specification patterns and event processing language. *IEEE Transactions on Software Engineering*, 46, 1, 100–112. DOI: 10.1109/TSE.2018.2859926.

[10] M. El-Attar. 2021. Empirically evaluating the effect of the physics of notations on model construction. *IEEE Transactions on Software Engineering*, 1–1. DOI: 10.1109/TSE.2021.3060344.

[11] R. Mohanani, B. Turhan, and P. Ralph. 2021. Requirements framing affects design creativity. *IEEE Transactions on Software Engineering*, 47, 5, 936–947. DOI: 10.1109/TSE.2019.2909033.

[12] D. Feitelson, A. Mizrahi, N. Noy, A. Ben, O. Eliyahu, and R. Sheffer. 2020. How developers choose names. *IEEE Transactions on Software Engineering*, 1–1. DOI: 10.1109/TSE.2020.2976920.

[13] Z. Eberhart, A. Bansal, and C. Mcmillan. 2020. A wizard of oz study simulating api usage dialogues with a virtual assistant. *IEEE Transactions on Software Engineering*, 1–1. DOI: 10.1109/TSE.2020.3040935.

[14] G. Scoccia, I. Malavolta, M. Autili, A. Di Salle, and P. Inverardi. 2021. Enhancing trustability of android applications via user-centric flexible permissions. *IEEE Transactions on Software Engineering*, 47, 10, 2032–2051. DOI: 10.1109/TSE.2019.2941936.

[15] K. Papis B, K. Grochowski, K. Subzda, and K Sijko. 2020. Experimental evaluation of test-driven development with interns working on a real industrial project. *IEEE Transactions on Software Engineering*. DOI: 10.1109/TSE.2020.3027522.

[16] D. Liu, Y. Feng, X. Zhang, J. Jones, and Z. Chen. 2020. Clustering crowdsourced test reports of mobile applications using image understanding. *IEEE Transactions on Software Engineering*, 1–1. DOI: 10.1109/TSE.2020.3017514.

[17] A. Blouin and J. Jezequel. 2021. Interacto: a modern user interaction processing model. *IEEE Transactions on Software Engineering*, 1–1. DOI: 10.1109/TSE.2021.3083321.

[18] J. Zhang, H. Jiang, Z. Ren, T. Zhang, and Z. Huang. 2021. Enriching api documentation with code samples and usage scenarios from crowd knowledge. *IEEE Transactions on Software Engineering*, 47, 6, 1299–1314. DOI: 10.1109/TSE.2019.2919304.

[19] P. Mäder, T. Kuschke, and M. Janke. 2021. Reactive auto-completion of modeling activities. *IEEE Transactions on Software Engineering*, 47, 7, 1431–1451. DOI: 10.1109/TSE.2019.2924886.

[20] M. Soltani, A. Panichella, and A. van Deursen. 2020. Search-based crash reproduction and its impact on debugging. *IEEE Transactions on Software Engineering*, 46, 12, 1294–1317. DOI: 10.1109/TSE.2018.2877664.

[21] S. Romano, C. Vendome, G. Scanniello, and D. Poshyvanyk. 2020. A multi-study investigation into dead code. *IEEE Transactions on Software Engineering*, 46, 1, 71–99. DOI: 10.1109/TSE.2018.2842781.

[22] Y. Zhou, C. Wang, X. Yan, T. Chen, S. Panichella, and H. Gall. 2020. Automatic detection and repair recommendation of directive defects in java api documentation. *IEEE Transactions on Software Engineering*, 46, 9, 1004–1023. DOI: 10.1109/TSE.2018.2872971.

[23] B. Kitchenham, L. Madeyski, G. Scanniello, and C. Gravino. 2021. The importance of the correlation in crossover experiments. *IEEE Transactions on Software Engineering*, 48, 8, 2802–2813. DOI: 10.1109/TSE.2021.3070480.

[24] J. Melegati, H. Edison, and X. Wang. 2020. Xpro: a model to explain the limited adoption and implementation of experimentation in software startups. *IEEE Transactions on Software Engineering*. DOI: 10.1109/TSE.2020.3042610.

5 PRIMARY STUDIES

INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE'2021)

[1] L. Braz, E. Fregnan, G. Çalikli, and A. Bacchelli. 2021. Data and materials for: why don't developers detect improper input validation?'; drop table papers; –. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 203–204. DOI: 10.1109/ICSE-Companion52605.2021.00090.

[2] N. Cooper, C. Bernal-Cárdenas, O. Chaparro, K. Moran, and D. Poshyvanyk. 2021. A replication package for it takes two to tango: combining visual and textual information for detecting duplicate video-based bug reports. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 160–161. DOI: 10.1109/ICSE-Companion52605.2021.00067.

[3] N. Peitek, S. Apel, C. Parnin, A. Brechmann, and J. Siegmund. 2021. Program comprehension and code complexity metrics: an fmri study. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 524–536. DOI: 10.1109/ICSE43902.2021.00056.

6 PRIMARY STUDIES

THE ACM JOINT EUROPEAN SOFTWARE ENGINEERING CONFERENCE AND SYMPOSIUM ON THE FOUNDATIONS OF SOFTWARE ENGINEERING (ESEC/FSE'2021)

[1] M. Endres, M. Fansher, P. Shah, and W. Weimer. 2021. To read or to rotate? comparing the effects of technical reading training and spatial skills training on novice programming ability. In *To Read or to Rotate? Comparing the Effects of Technical Reading Training and Spatial Skills Training on Novice Programming Ability* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 754–766. ISBN: 9781450385626. DOI: 10.1145/3468264.3468583.

[2] Y. Jiang, H. Liu, Y. Zhang, N. Niu, Y. Zhao, and L. Zhang. 2021. Which abbreviations should be expanded? In *Which Abbreviations Should Be Expanded?* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 578–589. ISBN: 9781450385626. DOI: 10.1145/3468264.3468616.

[3] M. Liu, X. Peng, A. Marcus, C. Treude, X. Bai, G. Lyu, J. Xie, and X. Zhang. 2021. Learning-based extraction of first-order logic representations of api directives. In *Learning-Based Extraction of First-Order Logic Representations of API Directives* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 491–502. ISBN: 9781450385626. DOI: 10.1145/3468264.3468618.

[4] B. Shen, W. Zhang, C. Kästner, H. Zhao, Z. Wei, G. Liang, and Z. Jin. 2021. Smartcommit: a graph-based interactive assistant for activity-oriented commits. In *SmartCommit: A Graph-Based Interactive Assistant for Activity-Oriented Commits* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 379–390. ISBN: 9781450385626. DOI: 10.1145/3468264.3468551.

[5] L. Shi, X. Chen, Y. Yang, H. Jiang, Z. Jiang, N. Niu, and Q. Wang. 2021. A first look at developers' live chat on gitter. In *A First Look at Developers' Live Chat on Gitter* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 391–403. ISBN: 9781450385626. DOI: 10.1145/3468264.3468562.

[6] Z. Wang, H. Zhang, T. Chen, and S. Wang. 2021. Would you like a quick peek? providing logging support to monitor data processing in big data applications. In *Would You like a Quick Peek? Providing Logging Support to Monitor Data Processing in Big Data Applications* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 516–526. ISBN: 9781450385626. DOI: 10.1145/3468264.3468613.

7 PRIMARY STUDIES

IEEE INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE (RE'2021)

[1] M. Endres, M. Fansher, P. Shah, and W. Weimer. 2021. To read or to rotate? comparing the effects of technical reading training and spatial skills training on novice programming ability. In *To Read or to Rotate? Comparing the Effects of Technical Reading Training and Spatial Skills Training on Novice Programming Ability* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 754–766. ISBN: 9781450385626. DOI: 10.1145/3468264.3468583.

8 PRIMARY STUDIES

IEEE/ACM INTERNATIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING (ASE'2021)

[1] M. Endres, M. Fansher, P. Shah, and W. Weimer. 2021. To read or to rotate? comparing the effects of technical reading training and spatial skills training on novice programming ability. In *To Read or to Rotate? Comparing the Effects of Technical Reading Training and Spatial Skills Training on Novice Programming Ability* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 754–766. ISBN: 9781450385626. DOI: 10.1145/3468264.3468583.

[2] Y. Jiang, H. Liu, Y. Zhang, N. Niu, Y. Zhao, and L. Zhang. 2021. Which abbreviations should be expanded? In *Which Abbreviations Should Be Expanded?* (ESEC/FSE 2021). Association for Computing Machinery, Athens, Greece, 578–589. ISBN: 9781450385626. DOI: 10.1145/3468264.3468616.