ADR Manager: A Tool-Supported Approach for the Efficient Creation and Management of Architectural Decision Records

Research Project (Bachelor-Forschungsprojekt Informatik)

# Organization

|  |  |
| --- | --- |
| Examiner: | Prof. Dr. Stefan Wagner |
| Supervisors: | Dr. Justus Bogner |
| Students: |  |
| Timeframe: | 2020-10-16 – 2021-04-16 |

# Context & Motivation

While software architecture is often focused on structural elements of a system [1], there are also definitions that place *decisions* at the center of architecture [2]. In this context, an architectural decision is a software design choice that addresses a functional or non-functional requirement that is architecturally significant. Decisions are documented as architectural decisions records (ADRs) [3] and contain additional elements like the rationale or considered alternatives. To store them close to the source code, ADRs are usually created in structured text formats like Markdown [4].

However, due to the lack of convenient tool support, the efficient creation, browsing, or analysis of ADRs is currently not feasible. A graphical user interface (GUI) would add significant value for managing ADRs and likely increase industrial adoption of this technique.

# Objectives

The goal of this research project is therefore to analyze challenges and detailed requirements in the context of creating and managing ADRs. Based on this analysis, a tool supported approach with a GUI should be designed and implemented. The final tool support should then be evaluated with respect to functional suitability and usability [5] (similar to perceived usefulness and perceived ease of use in the technology acceptance model [6]). Lastly, it is also possible - albeit not mandatory - to submit the results to the ICSE SCORE competition [7].

# Methods

The elicitation of challenges and requirements should start with the supervisors but can also include literature or even broader interviews [8]. To design and develop the approach, some form of (rapid) prototyping could be feasible [9, 10], potentially even with an approach specific to user interfaces [11]. For the evaluation, a think-aloud study [12], dialog-based study [13], or some other form of field study data collection [14] could be used. An alternative may be a small-scale experiment [15]. The detailed study design should be developed by the students.

# References

[1] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. Westford, MA, USA: Addison-Wesley Professional, 2012.

[2] A. Jansen and J. Bosch, “Software Architecture as a Set of Architectural Design Decisions,” in 5th Working IEEE/IFIP Conference on Software Architecture (WICSA’05), 2005, pp. 109–120.

[3] <https://adr.github.io>

[4] O. Kopp, A. Armbruster, O. Zimmermann “Markdown Architectural Decision Records: Format and Tool Support,” ZEUS, 2018, p. 55-62

[5] International Organization For Standardization, “ISO/IEC 25010 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models,” *System and software quality models*, <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>, 2011.

[6] F. D. Davis, “Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology,” *MIS Q.*, vol. 13, no. 3, p. 319, Sep. 1989.

[7] <https://conf.researchr.org/home/icse-2021/score-2021#adr-manager-the-software-architects-favorite-tool>

[8] S. E. Hove and B. Anda, “Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research,” in 11th IEEE International Software Metrics Symposium (METRICS’05), 2005, no. Metrics, pp. 23–23.

[9] J. D. Naumann and A. M. Jenkins, “Prototyping: The New Paradigm for Systems Development,” MIS Q., vol. 6, no. 3, p. 29, Sep. 1982.

[10] N. M. Devadiga, “Tailoring architecture centric design method with rapid prototyping,” in 2017 2nd International Conference on Communication and Electronics Systems (ICCES), 2017, pp. 924–930.

[11] D. Baumer, W. Bischofberger, H. Lichter, and H. Zullighoven, “User interface prototyping-concepts, tools, and experience,” in Proceedings of IEEE 18th International Conference on Software Engineering, 1996, pp. 532–541.

[12] T. Boren and J. Ramey, “Thinking aloud: reconciling theory and practice,” IEEE Trans. Prof. Commun., vol. 43, no. 3, pp. 261–278, 2000.

[13] Shaochun Xu and V. Rajlich, “Dialog-based protocol: an empirical research method for cognitive activities in software engineering,” in 2005 International Symposium on Empirical Software Engineering, 2005, pp. 383–392.

[14] J. Singer, S. E. Sim, and T. C. Lethbridge, “Software Engineering Data Collection for Field Studies,” in Guide to Advanced Empirical Software Engineering, London: Springer London, 2008, pp. 9–34.

[15] A. J. Ko, T. D. LaToza, and M. M. Burnett, “A practical guide to controlled experiments of software engineering tools with human participants,” Empir. Softw. Eng., vol. 20, no. 1, pp. 110–141, Feb. 2015.