

Conducting Runtime Quality Analyses using Domain Storytelling

Extended Abstract

Sebastian Frank¹, Julian Brott¹, Levin Kerschberger², Eduard Schander²,
Ariane Kraus², Jens Plüddemann², Georgia König², Paul Möhring²,
Matthias Haeussler², Georg Rekas², André van Hoorn¹

¹ University of Hamburg, Germany

² Novatec Consulting GmbH, Germany

1 Context and Problem

The runtime qualities of application systems — e.g., performance, reliability, and resilience — directly influence companies’ business success. Over the last few years, corresponding analysis measures such as load tests, resilience tests, and monitoring have become widespread in practice, and mature commercial and open-source tools have been developed. However, these measures are all at the technical level and not interpreted at the (business) domain level. At the same time, software architecture and software development approaches such as Domain-driven Design (DDD) [1], which are becoming increasingly widespread, essentially do not consider runtime quality concerns despite their criticality.

2 Objective

With the *dqualizer* approach [9], we envision closing the gap between the domain-specificity of application systems and the (technical) measures and findings of quality assurance utilizing a domain-centric approach. Domain experts’ questions are mapped to technical, executable tests. Vice versa, *dqualizer* translates the analysis results into a representation understandable for domain experts. The approach integrates means to model and monitor runtime quality metrics into DDD-based techniques, e.g., Domain Storytelling [5], that enable domain experts to describe domain-centric runtime quality concerns. Subsequently, the *dqualizer* approach intends to gather the relevant metrics with open-source tools and aggregate the analysis results on the domain level.

3 Method

For this purpose, we integrate means to model and monitor runtime quality metrics into DDD-based techniques that enable domain experts to describe domain-centric runtime quality concerns. As a first step, we extended an existing editor [13] for Domain Stories [5] to support the annotation of domain elements to specify Runtime Quality Analysis (RQA) tests. We base the format of the required data on

a well-known quality scenario format [4] and use terminology and explanations suitable for non-technical stakeholders. We derive the RQA test inputs from the established load testing tool JMeter [3] and the resilience engineering tool Chaos Toolkit [7]. The editor prototype presents RQA results in a text-based report similar to the work by Okanović et al. [2].

We conducted a qualitative user study in which four domain experts from industry had to solve four tasks on a modified cinema application example [5]. The participants’ performance in solving the tasks and feedback provided in a questionnaire indicate that domain experts are able to specify RQA tests using our approach. However, the usability of the editor still needs to be improved.

Based on the experiences and lessons learned, we implemented a new modeler specifically for specifying RQAs. Further, we implemented early versions of the core components of the *dqualizer* approach. These components comprise (1) a modeler for defining mappings between domain story elements and OpenAPI specifications [12], (2) a transformation service that translates the domain-level RQAs into technical RQAs, (3) an engine for executing these technical RQAs as k6 [11] load tests, and (4) a Grafana [10] dashboard for presenting the results. Thus, in the current state of the approach, we are able to specify and execute simple load tests using Domain Stories.

4 Result

Our preliminary results comprise the prototypical extension of the Domain Story modeler Egon.io [13] for specifying load and resilience tests and reporting test results. We gathered feedback from domain experts and lessons learned in a qualitative user study. Despite the editor’s limitations regarding functionality and usability, the feedback indicated that domain experts are able to model RQAs. Finally, our results also comprise the prototype of the *dqualizer* approach as a whole, focusing on conducting load tests.

5 Talk Outline and Additional Resources

We already presented our vision of the dqualizer approach at SSP '22 [6]. In this presentation, we will focus on the concrete realization of the approach for load testing using Domain Storytelling. In particular, we will present our preliminary results and lessons learned from developing the extended domain story modeler and conducting the qualitative user study as already presented in another publication [9]. In addition, we will showcase the current state of the dqualizer tooling for conducting load tests using domain stories. The dqualizer approach is publicly available [8].

6 Acknowledgments

This research was supported by the Federal Ministry of Education and Research (BMBF) [grant number 01IS22007A/B].

References

- [1] V. Vernon. *Domain-driven design distilled*. Addison-Wesley, 2016.
- [2] D. Okanović et al. “Concern-driven reporting of software performance analysis results”. In: *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*. ACM, 2019, pp. 1–4.
- [3] The Apache Software Foundation. *JMeter*. <https://jmeter.apache.org>. 2019.
- [4] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. 4th ed. USA: Addison-Wesley Longman Publishing Co., Inc., 2021.
- [5] S. Hofer and H. Schwentner. *Domain Storytelling: A Collaborative, Visual, and Agile Way to Build Domain-Driven Software*. Addison-Wesley, 2021.
- [6] H. Holz et al. “dqualizer: Domain-Centric Runtime Quality Analysis of Business-Critical Application Systems”. In: *SSP 2022 (2022)*. https://dqualizer.github.io/files/SSP22_Dqualizer_ExtendedAbstract.pdf.
- [7] Chaos Toolkit Team. *ChaosToolkit*. <https://chaostoolkit.org>. 2023.
- [8] dqualizer Team. *dqualizer Website*. <https://dqualizer.github.io/>. 2023.
- [9] S. Frank et al. “dqualizer: Domain-Centric Runtime Quality Analysis of Business-Critical Application Systems”. In: *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*. 2023, pp. 397–404.
- [10] Grafana Labs. *Grafana*. <https://grafana.com/>. 2023.
- [11] Grafana Labs. *k6 Load Testing Tool*. <https://k6.io/>. 2023.
- [12] OpenAPI Initiative. *OpenAPI*. <https://www.openapis.org/>. 2023.
- [13] *WPS Domain Story Modeler*. <https://egon.io/>. 2023.