

2DV505 - Current Topics within Computer Science - S2 Work Plan

Student 1: Mutasem Salloum (ms227fm)

Student 2: Rashed Qazizada (rq222ah)

1 Introduction



For this assignment, we have chosen option 2 (Research Question Proposal).

- **Research Question:** What are the current methodologies and approaches used to achieve reusability in self-adaptive software systems, and what are the benefits and challenges of implementing reusable components in these systems?

- **Bibliographic Details in IEEE Format:**

1. N. Abbas, J. Andersson, and D. Weyns, “ASPLe: A methodology to develop self-adaptive software systems with systematic reuse,” *Journal of Systems and Software*, vol. 167, p. 110626, 2020. DOI: <https://doi.org/10.1016/j.jss.2020.110626>.
2. S. Korra, V. Biksham, K. Vinaykumar, and T. Bhaskar, “Code-level self-adaptive approach for building reusable software components,” in *Intelligent Computing and Applications*, B. N. K. Rao, R. Balasubramanian, S. Wang, and R. Nayak, Eds., pp. 49–57. Springer Nature Singapore, 2023. DOI: https://doi.org/10.1007/978-981-19-41627_6.
3. I. Pekaric, R. Groner, T. Witte, J. G. Adigun, A. Raschke, M. Felderer, and M. Tichy, “A systematic review on security and safety of self-adaptive systems,” *Journal of Systems and Software*, vol. 203, p. 111716, 2023. DOI: <https://doi.org/10.1016/j.jss.2023.111716>.

- **Supporting Sections and Paragraphs:**

- **ASPLe: A methodology to develop self-adaptive software systems with systematic reuse** by N. Abbas et al.:

- * **Section 5.2 to 5.5:** Discusses different methods used to make software reusable, focusing on how the ASPLe process works in self-adaptive systems. This supports the different approaches to achieving reusability.

- * **Section 6.7 (Discussion):** Talks about the problems faced when trying to make components reusable, especially issues related to security and managing changes, which directly supports the question about challenges.

- **A systematic review on security and safety of self-adaptive systems** by I. Pekaric et al.:

- * **Section 2.3.3 and 2.3.4:** Provides information about potential risks and safety factors, which are important for understanding the challenges in making reusable components safe.

- * **Section 4.4 (Classification and analysis):** Explains how the authors checked the reliability of the information about security and safety, helping to ensure the findings are accurate.
- **Intelligent Computing and Applications - Code-level self-adaptive approach for building reusable software components** by S. Korra et al.:
 - * **Section 6.2:** Explains the benefits of using reusable components in adaptive systems compared to regular software systems. This supports understanding how these methods work at the code level.

Overview of the Topic

In modern software engineering, self-adaptive systems are a crucial topic, enabling different applications to autonomously adjust to changing environments or conditions. This adaptability is extremely important in dynamic environments like cloud computing and IoT, where responsiveness is key.

- **Self-Adaptation:** A system with the ability to monitor and adjust its behavior without human intervention, ensuring optimal performance despite external or internal changes.
- **Reusability:** Breaking the adaptive system into smaller decoupled components that can be treated and reused individually, which reduces development time and costs. This is crucial for creating flexible and efficient self-adaptive systems.
- **Methodologies for Reusability:** Approaches such as ASPLe provide structured methods to develop reusable components, promoting modular design.
- **Implementation Challenges:** While reusability offers benefits, it also presents challenges, particularly in security and safety, as highlighted in reviews on self-adaptive systems.

2 Objective and Scope

Objective

This work aims to investigate the current methodologies and approaches that are being used to achieve reusability in self-adaptive systems. Additionally, this research will address the associated benefits and challenges of implementing reusable components.

Scope

This study will focus on the following aspects:

- **Methodologies for Reusability:** Examining existing frameworks and methodologies that facilitate the development of reusable components in self-adaptive systems.
- **Benefits of Reusable Components:** Analyzing how reusability in self-adaptive systems contributes to efficiency, flexibility, and reduced development costs.
- **Challenges and Risks:** Investigating security, safety, and integration issues related to reusing components, as these are critical for maintaining system integrity and performance.

Limitations

The study will be limited to peer-reviewed articles published in tier 2 journals between 2020 and 2024 to ensure that the findings are based on the most relevant and recent research.

3 Planned Activities and Research Methods

1. **Literature Review:** Conduct a comprehensive review of peer-reviewed articles focusing on reusability in self-adaptive software systems. This will include analyzing recent studies published between 2020 and 2024 to identify key methodologies, benefits, and challenges.
2. **Case Study Analysis:** Identify and evaluate real-world examples of self-adaptive systems that successfully implement reusable components. This will involve examining documented case studies to understand practical applications and outcomes.
3. **Comparative Analysis:** Compare and contrast methodologies and findings across different studies. This will involve assessing how various approaches address the challenges of reusability and the effectiveness of implemented strategies.
4. **Synthesis of Findings:** Compile and synthesize findings from the literature review and case studies to create a comprehensive overview of the current state of reusability in self-adaptive software systems.

4 Expected Outcomes

For Option 2 (Research Question Proposal), the expected outcomes are to learn more about how to make software components reusable in self-adaptive systems, understand the benefits, and identify the challenges.

Expected Results

- Understand the methods used to make software reusable, including structured approaches like ASPLe and other code-level techniques.
- Learn about the advantages of reusing software, such as saving time, reducing costs, and making systems more flexible to changes.
- Identify the challenges, particularly in areas like security and safety, when trying to reuse components.

Supporting Evidence

- Evidence will come from the selected journal articles listed in Section 1: Introduction, under the subsection **Bibliographic Details in IEEE Format**.
- Analysis of real-life examples will also help demonstrate how these methods are used in practice.

Reporting Findings

- The findings will be organized into three main parts: methods for reusability, benefits of reusability, and challenges in adopting these methods.
- A summary will be made from both the literature and real-life examples to show a full picture of reusability in adaptive systems.

5 Potential Threats and Risks

Identify Risks

1. **Complexity of Research:** The methods used in different studies can be hard to understand because of the technical details, which could make the research challenging.
2. **Difficulty in Getting Clear Answers:** The topic of reusability in self-adaptive systems is still developing. Therefore, the available information may not fully answer all parts of the research question, especially regarding safety and security issues.
3. **Lack of Real-Life Examples:** There may not be enough real-world examples of using reusable components in these systems, which could make it hard to generalize the results.

Mitigation Strategies

1. **Dealing with Complexity:** We will seek help from our supervisors when facing difficult technical details and break down complex methods into smaller, manageable parts.
2. **Adapting to Unclear Answers:** We will expand our data collection efforts and document any unanswered areas, providing an explanation in our report.
3. **Handling Limited Data Availability:** If we face limited data, we will adjust our timeline to allow for more extensive literature research where necessary.

6 Timeline

Milestones

1. **Literature Review Summary Completion:** Completed – The research articles about reusability in adaptive systems have been read and summarized.
2. **Data Collection:** By November 1, 2024 – Collect detailed information from selected articles, focusing on key methodologies, benefits, and challenges. Ensure data is relevant for supporting the research question.
3. **Midterm Report Draft:** By November 15, 2024 – Submit a draft of the midterm report, focusing on analysis and findings, and address feedback from Iteration 1.
4. **Revisions and Feedback Integration:** By December 1, 2024 – Revise the draft based on supervisor feedback, ensuring all weaknesses are addressed.
5. **Final Report Preparation:** By December 20, 2024 – Finalize the report, ensuring it meets all requirements, including addressing all research questions and findings.

6. **Submission of Final Report:** By January 12, 2025 – Submit the final report.
7. **Presentation Preparation and Delivery:** January 13-17, 2025 – Prepare and deliver a presentation summarizing the key points of the final report.

7 Preliminary Structure of the Final Report

1. Introduction

- Introduce the topic of self-adaptive software systems and the significance of reusability.
- Present the research question and objectives of the study.
- Outline the importance of understanding methodologies, benefits, and challenges associated with reusability.

2. Methodology

- Describe the research design, including literature review techniques and case study analysis.
- Explain data collection methods, such as qualitative interviews with industry experts.
- Detail the systematic approach taken to analyze and synthesize the findings.

3. Results

- Present the key findings from the literature review, including identified methodologies and their effectiveness.
- Summarize data from case studies that illustrate the benefits and challenges of reusability.
- Include quantitative metrics and qualitative insights as evidence.

4. Discussion

- Analyze the implications of the findings, comparing and contrasting methodologies.
- Discuss the benefits and challenges in depth, using case study examples to provide context.
- Explore the relationship between reusability and overall system performance, flexibility, and security.

5. Conclusion and Future Directions

- Summarize the main conclusions drawn from the research.
- Highlight the practical implications for software engineering practices.
- Provide recommendations for future research, addressing identified gaps and challenges related to reusability in self-adaptive software systems.