

RESEARCH REPORT

ReSoTo Lemma

Anna Kadurina, Denitsa Goranova, Sobhi Kadi,
Mex Steens, Pedro Da Costa Ribeiro Ferreira Da
Silva

Table of Contents

| | |
|---|---|
| Introduction | 2 |
| Research Questions | 3 |
| Main Question | 3 |
| Sub-Questions | 3 |
| Research Methods | 3 |
| What are the primary objectives of creating the ReSoTo platform, and how do they align with the needs of software researchers in research projects? | 5 |
| 1. Goal: Technical Setup Automation | 5 |
| 2. Goal: Increasing the Productivity of Developers | 5 |
| 3. Goal: Putting User-Centered Design into Practice Need | 6 |
| 4. Goal: Making the Development and Administration of Research Platforms Easier ... | 6 |
| Interview | 7 |
| What are the specific pain points of the ReSoTo platform end-users when creating research platforms? | 7 |
| References | 9 |
| Interviews | 9 |

Introduction

In order to meet the unique requirements of software developers involved in information and communication technology (ICT) research projects, this study examines how to improve the ReSoTo (Researchers Solution Toolkit) platform. ReSoTo is a cutting-edge microservices-based solution intended to make research platform development easier and more efficient. The toolkit automates technical settings using XML and LEMMA scripts, freeing developers to focus on domain-specific tasks rather than infrastructure issues. The foundation of this analysis is the idea that software developers' productivity and

engagement may be significantly increased by making research platforms more effective and user-friendly.

The ultimate objective of the project is to reduce the amount of time spent on repeated setup processes by improving ReSoTo through automation and user-centered design. In order to do this, the study will examine the main issues that developers encounter, evaluate current frameworks and technologies, and recommend design changes that are in line with developers' requirements. This study intends to provide useful improvements that will make ReSoTo a more effective, efficient, and user-friendly platform for the ICT research community by addressing five important areas.

Research Questions

Main Question

How can the ReSoTo platform be enhanced using user-centered design principles to automate the generation of research platforms for software developers?

Sub-Questions

1. What are the primary objectives of creating the ReSoTo platform, and how do they align with the needs of software researchers in research projects?
2. What are the specific challenges of the ReSoTo platform end-users when creating research platforms?
3. What existing technologies, frameworks, and best practices can be integrated into ReSoTo to enhance its effectiveness and efficiency?
4. How can we design and implement ReSoTo's key components—such as the LEMMA Domain Script web form, admin panel, and storage logic—to meet user needs within their work contexts?
5. How do current users interact with ReSoTo, and what usability improvements can be made to enhance their experience?
6. What storage logic is required to manage multiple research projects efficiently within ReSoTo?

Research Methods

| Question | Methods | Explanation |
|----------|------------------------------|--|
| No. 1 | Literature Study, Interview. | We make use of literature studies because we review written papers and |

| | | |
|--------------|---|--|
| | | documents about the ReSoto platform and the vision behind it. We make use of interview by interviewing the lead stakeholder to get more insights and ensure the objectives we gather from literature study are valid. |
| No. 2 | Interview, Survey. | By interviewing the end-users, we get an idea of the main pain points that need to be addressed. |
| No. 3 | Literature Study, Available Product Analysis. | |
| No. 4 | IT Architecture sketching, Prototyping. | IT Architecture sketching is crucial to ensure we are tackling the key components. We use Prototyping to apply the initial concept from the architecture to see if the solution is feasible. System testing comes into place once we start to get into production ready environment and we still need to test the overall readiness of the system, |
| No. 5 | Observation, Usability testing. | We make use of observation to get a feeling on how users interact with the system to see if it is clear. After observation we use usability testing to detect possible problems and inconsistencies that might exist. |
| No. 6 | Literature study, Community research. | We use literature study to learn about the possible solution that are out there, and we combine that community research to see |

| | | |
|--|--|---|
| | | in which way they were used by others in similar situations |
|--|--|---|

Results

Research Question 1:

What are the primary objectives of creating the ReSoTo platform, and how do they align with the needs of software researchers in research projects?

Methods: Literature Study, Interview

The ReSoTo platform's main goals stem from the necessity of giving software developers working on research a more efficient, user-focused tool. A number of issues commonly seen in research settings, especially those pertaining to technical setup and resource management, were intended to be resolved by this platform. A summary of ReSoTo's primary goals and how they directly meet the demands of software researchers is provided below.

1. Goal: Technical Setup Automation

Need: Setting up technical settings for new research projects usually takes a lot of time for software developers working on research projects. This configuration entails setting up servers, databases, and other infrastructure, frequently using time consuming and intensive and vulnerable to error manual procedures.

The Method Used by ReSoTo: ReSoTo automates a large portion of the technical setup needed for ICT research initiatives by utilizing model-driven engineering (MDE) methodologies and tools like LEMMA (Language Ecosystem for Modeling Microservice Architecture). By offering a framework for modeling domain-specific elements, LEMMA makes it possible for XML scripts to automatically generate backend setups and microservices. **Alignment with Developer Needs:** By eliminating the need for repetitive settings, this automation frees up developers to concentrate on infrastructure rather than essential research features. ReSoTo guarantees correctness and consistency across installations by producing code and configuration files from high-level models, increasing productivity and cutting down on setup times.

2. Goal: Increasing the Productivity of Developers

Need: The primary research focus is sometimes reduced by the complexity of overseeing several research initiatives. Developers must spend a lot of time managing various

technological needs, keeping duplicate settings, and doing the same operations repeatedly across projects in the absence of a centralized and efficient solution.

The Method Used by ReSoTo: By integrating project management into a single toolset, ReSoTo seeks to increase efficiency by enabling users to oversee and implement several research projects from a single interface. The platform reduces duplicate activities through automated, flexible settings, freeing up developers' time to focus on the particulars of each research project.

Acceptance to Developer Requirements: The issue of project fragmentation is addressed by this centralization. ReSoTo increases productivity by eliminating duplication and combining resources, enabling researchers to do more with less work.

3. Goal: Putting User-Centered Design into Practice Need

Research tools frequently face the problem of not being adapted to developers' real-world requirements, which results in low acceptance rates and poor usability. Platforms that lack user-centered design may overlook important usability and functionality elements, which would make them more harmful than beneficial.

The Method Used by ReSoTo: A user-centered design strategy is being used in the development of the ReSoTo platform to make sure that the functionality, interface, and workflows match the real methods used by researchers. This contains responsive feedback loops, task-specific features, and easy-to-use navigation that improve usability and guarantee developers can use the product effectively.

Conformity to Developer Requirements: ReSoTo's user-centered design reduces the inconveniences that come with tools that aren't intuitive. ReSoTo addresses usability throughout the whole design process, which makes

4. Goal: Making the Development and Administration of Research Platforms Easier

Need: Creating and maintaining research platforms requires setting up a number of complex and distributed databases, systems, and components. Researchers would greatly benefit from a uniform toolset that streamlines these procedures by offering greater control and consistency.

The Method Used by ReSoTo: ReSoTo provides a set of tools made especially to streamline and consolidate the development and administration of various research platforms. Developers may manage resources, monitor project progress, and reuse components across projects with this method's centralized repository.

Conformity to Developer Requirements: ReSoTo lowers the overhead and learning curve associated with maintaining multiple platforms by streamlining these difficult operations.

In the end, this grouping increases workflow effectiveness and frees up more time for researchers to focus on experimental tasks

Interview

Our research into the main goals of the ReSoTo platform generated a number of important findings that shed light on the requirements and difficulties faced by software developers working in research settings. The key conclusion is that developers spend significant amounts of time from essential research activities by working on recurring technical setup tasks including setting up network rules, maintaining APIs, and configuring infrastructure. ReSoTo may greatly increase productivity by automating these setup procedures using model-driven engineering and tools like LEMMA, freeing up developers' time for experimentation and creativity. Additionally, the requirement for simplified oversight across numerous projects is addressed by centralizing project management in a single, user-friendly platform, which eliminates the redundancy and fragmentation that now inhibit productivity. The significance of a user-centered design approach is another important result; ReSoTo can increase adoption and engagement by incorporating quick-access features and straightforward navigation to boost usability. When taken as a whole, these goals meet the practical requirements of developers and provide a more effective, user-friendly, and useful toolbox that tackles the organizational and technological difficulties of managing research projects.

Research Question 2:

What are the specific pain points of the ReSoTo platform end-users when creating research platforms?

Methods: Survey, Interview

Several significant pain issues that limit productivity and general platform satisfaction have been found based on survey results and in-depth interviews with end users of the ReSoTo platform. The main problems that users have experienced fall into three categories: difficult manual configuration procedures, tool separation, and interface design usability concerns. A summary of each pain area is provided below, along with citations to survey and interview participant input.

Users have identified the extensive manual setup needed for every new research activity as a major source of discomfort. According to survey data, more than 70% of engineers report feeling "moderate to high levels of frustration" since early configuration chores including setting up servers, databases, and APIs are recurrent. Interviewees shared this

perspective, stating that 66% of a project's early setup time is frequently spent manually configuring infrastructure.

The lack of automated setup procedures, according to many users, leaves room for mistakes, which can throw off project schedules and lower the caliber of results. According to those who surveyed, this problem would be significantly mitigated by features like pre-configured configurations or reusable templates. Model-driven engineering (MDE) tools like LEMMA, which might automatically build and configure required components, cutting setup time and improving consistency across projects, are clearly needed for ReSoTo, according to this input.

The discomfort of dealing with fragmented tools was often highlighted in both survey and interview comments. 50% of developers said they were "frequently disrupted" by having to transition between platforms for duties like version control, deployment, and monitoring, according to the study results. User interviews revealed that this fragmentation causes recurrent data entry and synchronization across several platforms, as well as complicating processes. "It is very hard and boring to switch between tools," one candidate said. The centralization of these responsibilities via ReSoTo would save us a great deal of trouble and minimize mistakes.

Another major problem was usability; according to 60% of poll participants, the present interfaces of apps like ReSoTO are "difficult to navigate. or at least some of them " According to interviews, users usually have trouble accessing regularly used functionalities because of hidden features and complex menu layouts. "Everything we need is buried under layers of options—the interface feels like it was designed for a power user," one developer observed. This comment emphasizes the need for a more user-centered, intuitive design that puts an emphasis on task-oriented processes and ease of access.

75% chose "None" for usability difficulties, suggesting that most people may not see usability to be a significant concern. Nevertheless, one user complained that the UI was crowded, while another said that it was hard to find important functions.

Additionally, survey respondents said that by reducing the cognitive load needed to use the system, customizable dashboards and shortcuts to frequently used features would increase productivity. Platforms like Spring Initializr had low ratings for general usability, with the majority receiving scores between 1 and 3 out of 5, suggesting that ReSoTo may encounter similar issues if left unchecked.

Key findings:

The results of the survey and interviews point to a number of crucial areas where ReSoTo has to be improved, with a particular emphasis on tool integration, user-centered interface design, and automation of technical settings. ReSoTo can let developers concentrate on research activities instead of infrastructure administration by resolving these issues, which will increase user happiness and productivity.

Research Question 3:

What existing technologies, frameworks, and best practices can be integrated into ReSoTo to enhance its effectiveness and efficiency?

Methods: Literature Study, Available Product Analysis

To identify suitable technologies, frameworks, and best practices for ReSoTo, a literature review of model-driven engineering (MDE) and microservices-based platforms in research projects was conducted, along with analysis of existing products. This approach helped pinpoint tools and methods that align well with ReSoTo's objectives.

Findings:

1. Model-Driven Engineering (MDE) Tools

According to MDPI (28 Jan, 2023), MDE is a general engineering approach that focuses on using models as the primary artifact for software development. In MDE, models can be used to capture the structure and behavior of a system, and these models can be transformed into the final implementation of the system using model transformations and code generators.

As Florian Rademacher (2022) explained, LEMMA is an MDE solution that aims to facilitate the design, development, and deployment of microservice architectures using:

- Modeling languages dedicated to the concise definition of microservices including their domain information and deployment specification;
- Model transformations allowing automatic generation of domain-specific code, microservices code, and deployment configuration for heterogeneous generation targets;
- Model analysis to support in detecting design smells microservice architectures prior to their development and/or deployment

Since ReSoTo aims to reduce manual setup through automation, integrating LEMMA is an effective solution. It helps generate infrastructure configurations, microservices, and necessary integrations, significantly reducing the time and effort developers spend on initial setups.

Further research discovered Eclipse Modeling Framework (EMF) as also beneficial for ReSoTo. According to the Eclipse Foundation, Eclipse EMF is a modeling framework and code generation facility for building tools and other applications based on a structured data model. From a model specification described in XML, EMF provided tools and runtime support to produce a set of Java classes for the model, a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor.

2. Backend Frameworks and Microservice Architecture

According to both Medium and GeeksForGeeks, Spring Boot leads the Top 10 frameworks for microservices based architecture. It is widely adopted for building microservices and has extensive support for RESTful services, which are essential for ReSoTo's modular, service-oriented architecture. Additionally, the framework supports Docker and Kubernetes for easy deployment, scalability, and reliability in production environments.

Alongside the backend frameworks, another aspect to think about is the deployment, portability, scalability, and management of services. Docker containers simplify the development and deployment of microservices, whereas Kubernetes provides orchestration for these containers, enabling automatic scaling and management of services. This aligns with ReSoTo's goal of efficient resource management.

3. Continuous Integration/Continuous Deployment (CI/CD)

As Paulo Gardini Miguel explains, there are multiple options for CI/CD tools. Based on a review, Argo CD, CicleCI, Jenkins, Github Actions, and Azure DevOps made the Top 5 list. All of these have pros and cons and are all suitable options for ReSoTo. However, the platform is going to be used and managed by the Fontys Innovation Lab, therefore Gitlab is the most suitable option.

4. User-Centered Design and Interface Tools

According to Luis Minvielle, the "big three" frameworks in frontend development are: React, Angular, and Vue. React stands out as the go-to frontend framework. React is beginner-friendly, has a big community, maturity, and reliability. Another suitable option is the supported by Google and a solid digital community – Angular. It is a widely adopted framework. It particularly sticks out in enterprise-level setups, especially when paired with TypeScript.

5. Best practices and Standards

- Assign each microservice a single responsibility
- Use separate data storage for each microservice
- Keep code at a similar level of maturity

- Have a separate Build for each microservice
- Deploy into Containers
- Adopt Domain-Driven Design (DDD) that focuses on defining a common language and model around the business domain, which guides the design of microservices.
- Implement API Gateways
- Utilize Monitoring and Logging
- Embrace CI/CD

According to Wensen Ma, these are some of the essential Microservice Best Practices. These practices are applicable to ReSoTo and will enhance the code quality, stability, security, and scalability.

To refine ReSoTo and ensure it meets the needs of research project management effectively, a detailed analysis of comparable platforms can provide valuable insights. Platforms like GitHub, JupyterHub, and Camunda offers features that streamline workflows and enhance automation. For instance, GitLab's continuous integration/continuous deployment (CI/CD) pipelines could inspire further automation in ReSoTo's project setup process, while Camunda's intuitive workflow design offers a model for developing a user-friendly interface. Comparing these products based on usability and best practices highlights simplicity and modularity as critical factors for adoption. The findings from this analysis can inform actionable improvements for ReSoTo, such as simplifying the interface for ease of use, and integrating flexible project management tools to streamline research setup and management.

Research Question 4:

How can we design and implement ReSoTo's key components—such as the LEMMA Domain Script web form, admin panel, and storage logic—to meet user needs within their work contexts?

Methods: IT Architecture Sketching, Prototyping

1. Architecture

To design properly the components and how they are going to interact with each other, we relied on the C4 diagrams. On Figure 1 below, you can see the C2 which is on service level. We are utilizing microservices architecture with API Gateway to route the requests. Each service has its own responsibility, and they are based on the goal of the project. The Researchers and Developers interact with the frontend to complete their tasks. The

frontend calls the gateway. The API Gateway not only routes the requests but also authenticates the users with KeyCloak. Each service then contains its own logic for the relative field that it is in. To distribute data properly, we also have a separate database per microservice.

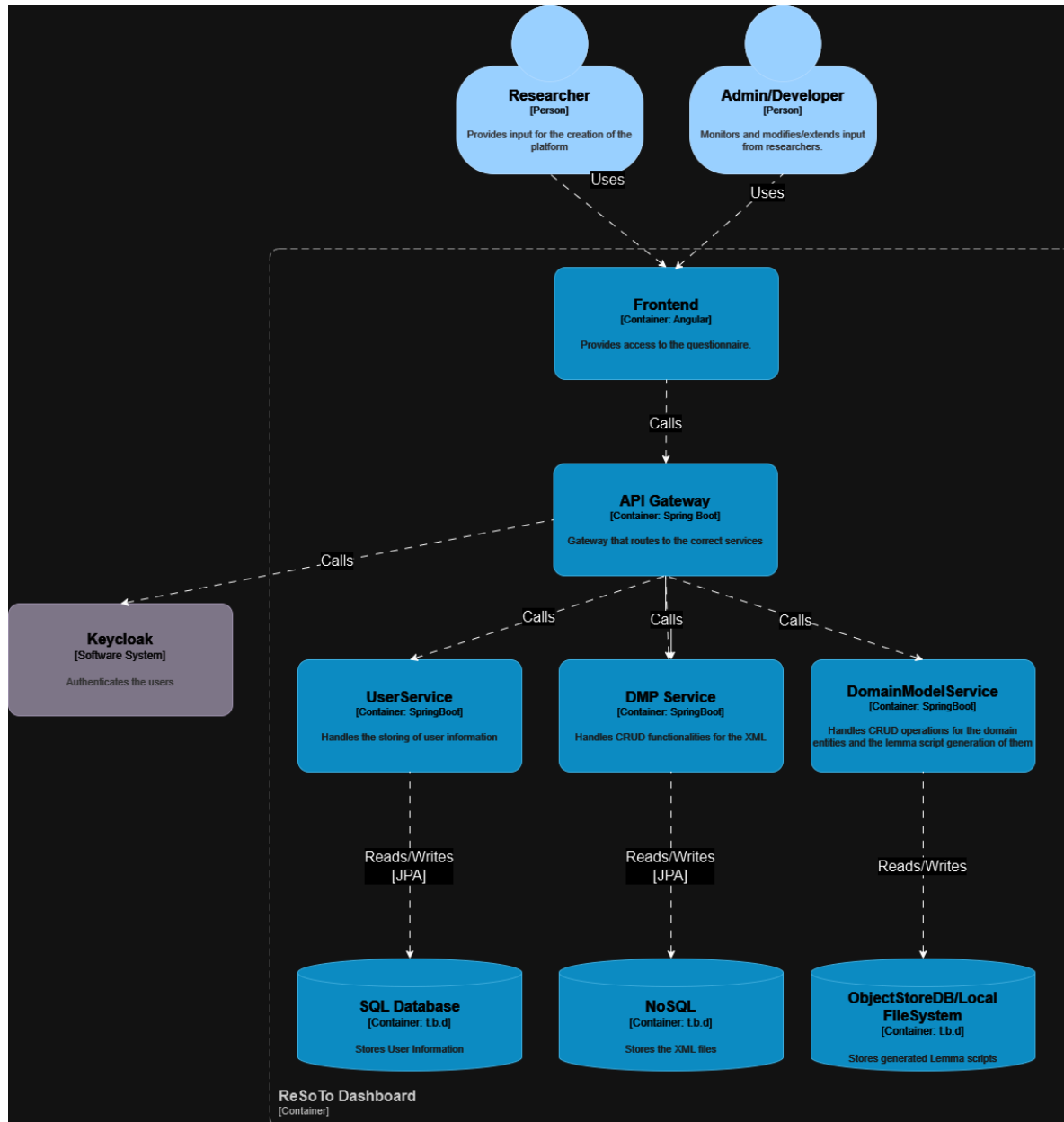
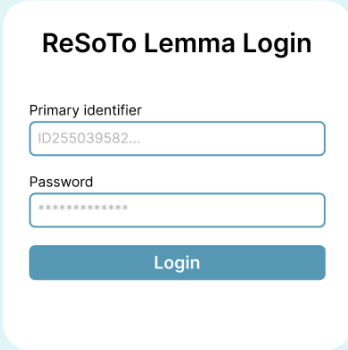


Figure 1 – C2 diagram of ReSoTo

2. Prototype

To create a prototype, we decided on using Figma. With its many capabilities it is the perfect tool to tackle this task. We have continuously sought feedback from our stakeholder and improved the prototype based on it. We have decided on using blue colors for this project, without any additional distractions whatsoever. We aimed at an easy-to-use interface that is logical to navigate. In the figures below, you can see the prototype for the ReSoTo project.



The image shows a login form titled "ReSoTo Lemma Login" centered on a light blue background. The form is a white rounded rectangle containing two input fields and a button. The first input field is labeled "Primary identifier" and contains the text "ID255039582...". The second input field is labeled "Password" and contains ten asterisks "*****". Below the password field is a blue button with the text "Login" in white.

Figure 2 – Login page design

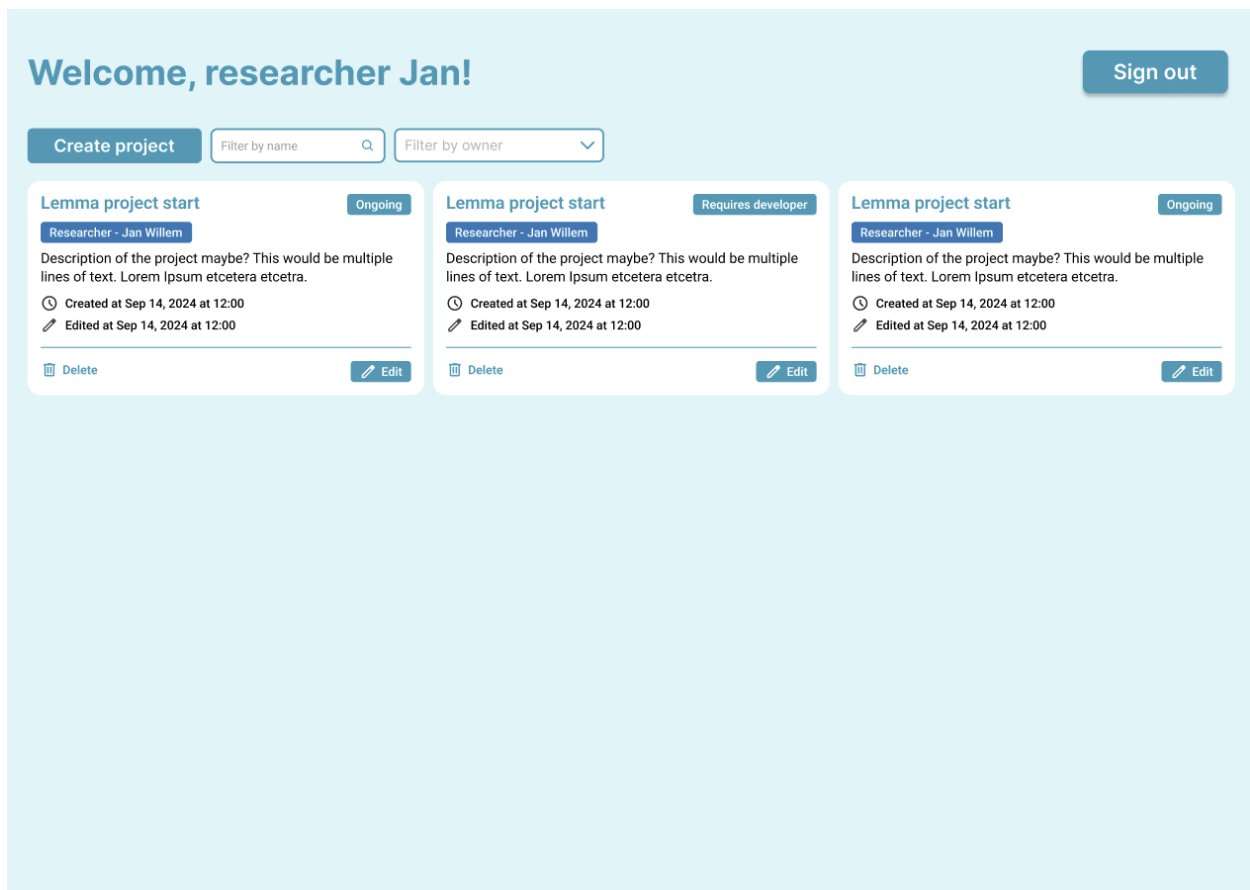


Figure 3 – Dashboard design

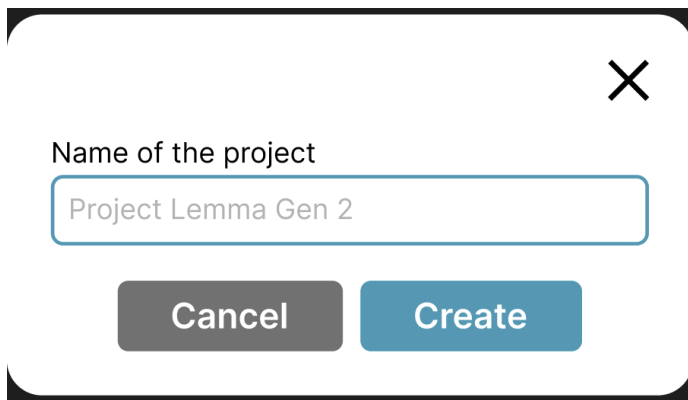


Figure 4 – Pop-up "Create Project" design

ReSoTo Lemma Generation

What data will you collect?

☐ Time series data

☐ Vector data

☐ Geo data

☒ For Developer

How much data will be collected?

25 GB

☒ For Developer

Where and how will you store your data?

☐ Azure

☐ Fontys Netlab

☒ For Developer

What kind of database system would you like to use?

☐ MySQL

☐ MongoDB

☐ PostgreSQL

☒ For Developer

Are there any specific requirements?

Select requirement

☒ For Developer

Figure 5 – Questionnaire design

ReSoTo Domain modeling tool

ChargingStationManagement

ParkingArea

1.

Select variable type

Variable name

☐ Immutable

Select relationship

☐ Identifier ☐ Unique

+ Add field

ParkingMeter

1.

Select variable type

Variable name

☐ Immutable

Select relationship

☐ Identifier ☐ Unique

+ Add field

Delete class

Add another class

Enter classes domain here...

Enter class name here...

1.

Select variable type

Variable name

☐ Immutable

Select relationship

☐ Identifier ☐ Unique

+ Add field

Delete class

Add another class

Delete classes domain

Add another classes domain

Cancel

Save

Figure 6 – Domain Modeling Tool design

Research Question 5:

How do current users interact with ReSoTo, and what usability improvements can be made to enhance their experience?

Methods: Observation, Usability testing.

The objectives of this question are:

- Understand how users interact with the newly developed solution (ReSoTo) to identify natural usage patterns, obstacles, and areas of confusion.
- Validate and uncover specific problems in ReSoTo's usability through structured testing.
- Validate that the application meets the business requirements.

Confirm that the application provides a user-friendly experience.

Usability testing:

Test prerequisites and assumptions:

The testing process is based on the following assumptions:

- The tester performing the test has sufficient knowledge on how to setup a software project.
- The tester performing the test has a clear understanding of the application's functionality and the expected outcome of each test case.
- The tester performing the tests must have access to the ReSoTo test environment.

Before conducting the tests, the following prerequisites must be met:

- The application must be deployed in the testing environment with all/most planned features and functionalities implemented.
- Test accounts for Researcher and Developer roles must be created and provided to the testing team.
- The testing environment must replicate the production environment as closely as possible to ensure accurate test results.

Login information:

Username: Johannes

password: test123

Testing data:

Project name: Sustainable Materials in Modern Construction: A Lifecycle Analysis

Project Description: An in-depth analysis of eco-friendly construction materials and their environmental impact over a 50-year lifecycle. The study highlights bamboo and recycled steel as key alternatives to traditional materials.

Test scenarios:

| Test Information | | | | | Test Execution | |
|------------------|---|---|--|---|----------------|------------------------------------|
| Case Nr. | Test Name | Test Description | Data | Expected Result | Actual Result | Acceptance Criteria met: Pass/Fail |
| TC - 01 | Login | Upon navigating to the application, the log in page will appear, the user enters the credentials and clicks on login. | Login credentials | Upon successful login. The tool opens the dashboard page. | | |
| TC - 02 | Initialize a new project. | The user clicks on "Create Project" button on the dashboard, a dialog will appear asking the user to enter a project name. and to click on "Create" button. | Use test data, project name and description. | Upon filling a project name and clicking on "Create" button, the tool will redirect the user to the questionnaire page. | | |
| TC - 03 | Answering the questionnaire and save the project. | The user fills in as much as possible of the questions or mark the question to be answered by a different user. User clicks on "Save" button to save the project. | | Upon answering the questions and clicking on "Save" the questionnaire should be save and a message confirming that should appear on the window. And a dialog to ask the user whether they | | |

| | | | | | | |
|---------|--------------------|---|--|--|--|--|
| | | | | want to continue to domain modelling or back to dashboard will appear. | | |
| TC - 04 | Delete DMP project | On the dashboard page the user clicks on the delete button of the project they want to delete | | A message will appear with either confirming or the error message. | | |

Results

Tester: Boendermaker, Wouter W.J.

Tester background: Researcher at Fontys labs, with research experience.

Usability testing results:

| Test Information | | | | | Test Execution | |
|------------------|---------------------------|---|--|---|----------------|------------------------------------|
| Case Nr. | Test Name | Test Description | Data | Expected Result | Actual Result | Acceptance Criteria met: Pass/Fail |
| TC - 01 | Login | Upon navigating to the application, the log in page will appear, the user enters the credentials and clicks on login. | Login credentials | Upon successful login. The tool opens the dashboard page. | OK | Pass |
| TC - 02 | Initialize a new project. | The user clicks on "Create Project" button on the dashboard, a dialog will appear asking the user to enter a project name. and to | Use test data, project name and description. | Upon filling a project name and clicking on "Create" button, the tool will redirect the user to the questionnaire page. | OK | Pass |

| | | | | | | |
|---------|---|---|--|--|----|------|
| | | click on “Create” button. | | | | |
| TC - 03 | Answering the questionnaire and save the project. | The user fills in as much as possible of the questions or mark the question to be answered by a different user. User clicks on “Save” button to save the project. | | Upon answering the questions and clicking on “Save” the questionnaire should be save and a message confirming that should appear on the window. And a dialog to ask the user whether they want to continue to domain modelling or back to dashboard will appear. | OK | Pass |
| TC - 04 | Delete DMP project | On the dashboard page the user clicks on the delete button of the project they want to delete | | A message will appear with either confirming or the error message. | OK | Pass |

As we can see from the results a success rate of 100 percent was achieved user were able to complete the tasks with no issues.

Observation test

Areas of improvement:

Project Tiles:

- Users found it unclear that the displayed tiles represented projects.
- The placement of the "Delete" option on the project tiles was problematic, increasing the risk of accidental deletion.

Questionnaire and Domain Modeling Tool:

- Users needed better preparation before entering the questionnaire or domain modeling tool. A **hint or notification** should be added to inform users about what to expect.
- **Tooltips** were suggested for questions to explain their purpose and expected answers.

Clarity in Questionnaire:

- Ambiguity in questions led to user confusion:
 - **"How will you ensure backup?"** needs clarification and more descriptive guidance.
 - **"What data will be archived?"** should be rephrased as **"How will data be archived?"** to improve clarity.
- Questions like **"What data types will be collected?"** should allow multiple selections to account for varied research needs.
- Some questions, such as **"Who owns the data?"**, should have limited answer options (e.g., a predefined list).

Questionnaire Purpose and Metadata:

- Users struggled to distinguish which questions contributed to metadata versus those used in script generation.
- **Grouping related questions** and clarifying their purpose was recommended.

Developer Option Confusion:

- The "Developer" option was mistaken as an answer instead of an indicator for another user. This requires redesign to eliminate ambiguity.

Adjusting Saved Choices:

- Users were uncertain if they could adjust choices after saving a project. This feature should be made explicitly clear.

Questionnaire Templates:

- Users suggested offering **predefined templates** to reduce the time required to complete questionnaires.

Audience-Specific Targeting:

- Questions should be reshaped to align with specific research fields or audience needs.

Auto-Save Functionality:

- Adding **auto-saving** to prevent data loss during the questionnaire process was identified as a priority.

Positive feedback:

1. Users appreciated the tool's visual appeal and its proper functioning.
2. Feedback indicated that the platform's interface was generally well-received, with no significant technical glitches encountered during usage.

Key insights

The usability testing and observation sessions revealed that while users found ReSoTo visually appealing and technically functional, they encountered several challenges that hindered seamless interaction with the platform. Users struggled with ambiguous features and unclear labels, such as the purpose of project tiles and the placement of the "Delete" button, which posed a risk of accidental actions. The questionnaire section, a critical component, lacked sufficient guidance and clarity, leading to confusion about the intent and purpose of certain questions. For instance, questions like "How will you ensure backup?" or "What data will be archived?" were deemed vague, and users often misunderstood the "Developer" option as an answer rather than a marker for delegation. Furthermore, users were unclear about which parts of the questionnaire influenced metadata versus those driving script generation, causing unnecessary hesitation and errors. This indicates a pressing need for better grouping of questions and more transparent labeling to convey their purpose effectively.

The testing also highlighted a significant opportunity to simplify user workflows by introducing predefined templates tailored to different research contexts, reducing redundancy and saving time. Users expressed concerns about losing progress when navigating the questionnaire and emphasized the need for an auto-save mechanism to preserve input data being deleted. Moreover, allowing users to adjust their saved choices was identified as essential for fostering flexibility. Despite these challenges, users appreciated the foundational structure and functionality of ReSoTo but felt the experience could be greatly enhanced with clearer instructions, intuitive design, and features that cater specifically to their diverse research needs. Addressing these issues would not only streamline the user experience but also ensure that ReSoTo aligns more closely with its goal of being a user-centered, efficient research tool.

Research Question 6:

What storage logic is required to manage multiple research projects efficiently within ReSoTo?

Methods: Literature study, Community research.

A thorough and scalable storage logic that facilitates organization, accessibility, scalability, and integration with questionnaire insights is necessary to handle several research projects inside ReSoTo, with a focus on storing XML project configuration files. A hierarchical file organization model forms the basis of this system, in which projects are methodically grouped in directories according to their category, such as community research or literature studies. Every project is given its own subfolder, which is named according to conventional practices and reflects a project ID or explanatory identification. This permits easy access to certain projects and guarantees logical separation.

The XML files in each project folder are named according to standard standards, for example, config_main.xml for the main configuration, and metadata.xml or supplementary_data.xml for any project-related files. A relational database or a document-based system such as PostgreSQL or MongoDB is used to construct a strong metadata indexing system in order to improve this structure. All project metadata, such as unique IDs, project names, study emphasis, tags or keywords, file locations, creation and modification dates, contributors, and linkages between projects, such references to related or derivative works, are centrally stored in this database. Users may do exact searches and queries using a variety of parameters, including keywords, project IDs, or certain tags, thanks to the metadata's sophisticated indexing and retrieval capabilities.

The storage system makes use of cloud-based solutions like AWS S3, Azure Blob Storage, or Google Cloud Storage, which offer scalable, dependable, and secure storage environments, in order to guarantee scalability and handle an increasing number of projects and XML files. Network Attached Storage (NAS) systems can be used for on-premises deployments; to minimize data loss, they should be complemented with automatic backups and routine capacity monitoring. In order to maximize storage space and preserve access to historical data, maintenance routines are set up to archive older or dormant projects. To cut down on redundancy and increase storage efficiency, duplicate file identification and removal procedures are put in place. In order to protect against any system failures, automated backups are planned to guarantee the security and recoverability of all stored files and related information.

When you need to store structured data, such as metadata, and have relationships between different entities, such as projects, files, and users, relational databases are a good option.

PostgreSQL, MySQL, SQL Server, and SQLite are a few examples.

and we may utilize bigger file types. Large XML files and other binary data, such as ZIP files, are best kept in object storage, although relational databases may still hold the information.

AWS S3, Google Cloud Storage, and Microsoft Azure Blob Storage are a few examples.

Conclusion

The findings of this research highlight the critical role that automation and user-centered design play in enhancing the productivity and satisfaction of researchers using the ReSoTo platform. By addressing repetitive manual tasks through tools like LEMMA, the platform can significantly reduce the time spent on setup and increase focus on research-specific activities. Usability testing and interviews emphasize the importance of simplifying workflows and incorporating features such as customizable templates and auto-save functionality to cater to diverse user needs.

Furthermore, integrating best practices in microservices architecture and leveraging scalable storage solutions ensures that ReSoTo remains robust and efficient as it scales to handle multiple research projects. This research provides actionable insights and practical recommendations that pave the way for the successful implementation and adoption of the ReSoTo platform, ensuring it achieves its goals of efficiency, usability, and scalability.

References

Almeida, F., & Monteiro, J. (2021). Model-driven engineering for microservice architectures: Applications and challenges. *Journal of Systems and Software*, 183, 111078

Morrison, A., & Snipes, W. (2020). Leveraging automation in research software development: A practical approach. *Software Engineering Research*, **12(3)**, 207-217

Bettini, L., & Wimmer, M. (2019). Modeling languages and tools for software automation in ICT research. In *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE'19)*.

Brambilla, M., Cabot, J., & Wimmer, M. (2017). *Model-driven software engineering in practice*. Morgan & Claypool Publishers.

Schmidt, D. C. (2006). *Model-driven engineering*. *IEEE Computer*, 39(2), 25–31.

Zahra Mardani Korani, Armin Moin, Alberto Rodrigues da Silva, Joao Carlos Ferreira (28 Jan, 2023). Model – Driven Engineering Techniques and Tools for Machine Learning Enabled IoT Applications: A Scoping Review. Retrieved from:

<https://www.mdpi.com/1424-8220/23/3/1458>

Florian Rademacher (2022). LEMMA Documentation. Retrieved from:

<https://seelabfhdo.github.io/lemma-docs/>

Eclipse Foundation (N.A.). Eclipse EMF. Retrieved from:

<https://projects.eclipse.org/projects/modeling.emf.emf>

GeeksForGeeks (17 Sep, 2024). Top 10 Microservices Frameworks. Retrived from:

<https://www.geeksforgeeks.org/microservices-frameworks/>

Anil Kurmi (2023). Top 10 Microservices frameworks for 2023. Retrieved from:

<https://medium.com/microservices-architecture/top-10-microservices-framework-for-2020-eefb5e66d1a2>

Paulo Gardini Miguel (25 Oct, 2024). 20 Best CI/CD tools for 2024. Retrieved from:

<https://thectoclub.com/tools/best-ci-cd-tools/>

Luis Minvielle (17 Mar, 2024). The Best 7 Frontend Frameworks for Developers in 2024. Retrieved from:

<https://www.wearedevelopers.com/magazine/best-frontend-frameworks-for-developers>

Wensen Ma (Jan 25, 2024). 18 Essential Microservice Best Practices. Retrieved from:

<https://www.devskillbuilder.com/18-essential-microservice-best-practices-655fd4d20ee6>

Interviews

Interviewer: "Thank you for joining me today, Luuk. I really appreciate your time. To get started, can you briefly describe your experience with setting up infrastructure for research projects and any recurring challenges you encounter?"

Luuk Tuijtelaars: "Sure! I've been working on ICT research projects for about six years, mostly in a research-focused development role. Setting up infrastructure is one of those necessary but tedious tasks. We spend a lot of time configuring servers, databases, setting up APIs, and making sure everything communicates well. It's essential, but honestly, it's not where we want to spend the bulk of our time."

Interviewer: "That makes sense. So, would you say the technical setup phase takes up a significant amount of time at the beginning of each project?"

Luuk Tuijtelaars: "Absolutely. I'd estimate around 20–30% of our time at the start is spent purely on setup. And when we start new projects, even if we're reusing some of the same components, there are always small adjustments. It's often repetitive, and any manual process opens the door for mistakes."

Interviewer: "If you could automate more of these setup tasks, what specific processes would you want to see automated?"

Luuk Tuijtelaars: "For me, it'd be really helpful if the platform could automate initial configurations, like setting up servers and defining network rules. Also, being able to generate APIs automatically based on a model would be incredible—anything that could get us up and running faster without so much manual coding. And if it could integrate these setups into some sort of version control or centralized management, even better."

Interviewer: "That's very insightful. Speaking of centralization, do you think a toolkit like ReSoTo would benefit from having a single point of management for all project resources?"

Luuk Tuijtelaars: "Definitely. Right now, we're jumping between different tools for deployment, monitoring, and version control. Having a single interface or a dashboard in ReSoTo would make it much easier to track everything. It would also help reduce errors since we're not copying configurations from one place to another."

Interviewer: "Thank you, Luuk. Moving on, I'd love to hear more about any challenges you face with usability in current tools. Are there any common frustrations or things you feel could be improved?"

Luuk Tuijtelaars: "Yes, usability is a big one. Many tools aren't tailored to developers' needs, especially for research environments. I've used platforms where the navigation isn't intuitive, or where a lot of features are hidden behind layers of menus. If ReSoTo can keep things simple and have commonly-used features easily accessible, I think that would make a huge difference."

Interviewer: "That's a great point. If you were to design ReSoTo, are there any specific UI elements or workflows that you'd want to see?"

Luuk Tuijelaars: "I'd say it should start with a project overview page that has the basics like project status, main configuration options, and a snapshot of resources in use. Having quick access to the main components without needing to search for them would be ideal. And maybe customizable shortcuts, so we can quickly access our most-used actions without clicking through too many tabs."

Interviewer: "Got it. Customizability sounds like a key aspect for you. One of our goals with ReSoTo is to focus on user-centered design. What specific functionality or features do you think would best support your daily workflows?"

Luuk Tuijelaars: "I'd say integration with version control would be a big one, especially for research projects where we're experimenting with different configurations. Being able to track changes and roll back to previous setups without having to manually reconfigure things would be invaluable. Also, any form of real-time feedback or error-checking could really streamline things."

Interviewer: "Interesting. Real-time feedback would be a valuable addition. Could you describe any other pain points related to managing multiple research projects simultaneously?"

Luuk Tuijelaars: "One of the big issues is consistency. When we're working on multiple projects, we sometimes duplicate setup processes because there isn't a clear template or reusable component in place. If ReSoTo could provide templates or allow us to reuse configurations across projects, that would really improve efficiency."

Interviewer: "That's great feedback. We're also looking at how best to handle storage and organization for ReSoTo. Do you have any thoughts on how storage logic could be structured to support multiple projects effectively?"

Luuk Tuijelaars: "Yes, storage logic is actually a big deal. Ideally, I'd like to see a project-based folder structure that could automatically organize files and configurations by project. Version control within each project's folder would also help a lot. And maybe a shared resource section for reusable code snippets or templates would save time too."

Interviewer: "Thanks, Alex. I have one last question. If you had a chance to use ReSoTo tomorrow with all these features, how do you think it would impact your work?"

Luuk Tuijelaars: "If it could automate setups, centralize management, and keep everything organized, it would honestly be a game-changer. We could focus on real research, experimenting with new features, and improving performance, instead of constantly revisiting setup tasks. Overall, I think it would speed up our workflows by at least 30–40%."

Interviewer: "I appreciate that, Luuk. We're also thinking on ways to help other initiatives in ReSoTo. Do you find it difficult to oversee many research projects at once?"

Luuk: "Yes, it may be challenging to oversee several projects, particularly when we're dealing with various setups for each one. Because there aren't any clear templates or reusable setups, we occasionally wind up repeating setup procedures, which makes consistency difficult. It would greatly increase project productivity if ReSoTo allowed us to develop and reuse templates."

Interviewer: "That's amazing feedback, Alex. Thank you again for your time and insights. Your feedback is invaluable as we work to refine ReSoTo into something that truly meets your needs."

Luuk Tuijelaars: "Happy to help! I'm excited to see how ReSoTo evolves. Good luck, and let me know if there's anything else I can contribute."

Survey questions:

- 1. What is your role in your organization?**
 - a. Software Developer**
 - b. Researcher**
 - c. Project Manager**
 - d. Other (please specify)**
- 2. How many years of experience do you have in research project development?**
 - a. 0–2 years**
 - b. 3–5 years**
 - c. 6–10 years**
 - d. More than 10 years**
- 3. How often do you use the configuration platform for your projects?**
 - a. Daily**
 - b. Weekly**
 - c. Monthly**
 - d. Rarely**
 - e. Not yet used**
- 4. How much time do you spend on initial setup and configuration for a typical research project using such platforms?**
 - a. Less than 10% of project time**
 - b. 10–20% of project time**

- c. 20–30% of project time
 - d. More than 30% of project time
- 5. How often do you find the setup process repetitive or redundant for each new project?
 - a. Always
 - b. Often
 - c. Sometimes
 - d. Rarely
 - e. Never
- 6. Which specific aspects of project setup do you find the most time-consuming? (Select all that apply)
 - a. Server setup and configurations
 - b. Database setup and connections
 - c. API deployment
 - d. Network rule configuration
 - e. Other (please specify)
- 7. How useful would automated setup tools or templates be in reducing your setup time?
 - a. Extremely useful
 - b. Very useful
 - c. Somewhat useful
 - d. Not useful
 - e. Not applicable
- 8. How many different tools do you typically use to manage a research project?
 - a. None
 - b. 1–2 additional tools
 - c. 3–4 additional tools
 - d. More than 4 additional tools
- 9. How disruptive is it to switch between different tools to complete tasks related to a single project?
 - a. Extremely disruptive
 - b. Very disruptive
 - c. Somewhat disruptive
 - d. Slightly disruptive
 - e. Not disruptive
- 10. If it was possible to integrate or centralize the following functions, which would be most beneficial? (Select all that apply)
 - a. Version control

- b. Deployment monitoring
- c. Resource management
- d. Configuration management
- e. Other (please specify)

11. How would you rate the overall usability of the such interfaces?

- a. Excellent
- b. Good
- c. Fair
- d. Poor
- e. Very poor

12. Which of the following usability issues, if any, do you encounter when using the configuration platforms? (Select all that apply)

- a. Difficulty navigating the menu structure
- b. Key features are hard to locate
- c. Interface feels cluttered or complex
- d. Essential functions require too many steps to access
- e. Other (please specify)

13. How helpful would the following improvements be in making the platforms that you use more user-friendly?

- a. Customizable dashboards
 - i. Extremely helpful, Very helpful, Somewhat helpful, Not helpful, Not applicable
- b. Quick-access shortcuts to frequently-used functions
 - i. Extremely helpful, Very helpful, Somewhat helpful, Not helpful, Not applicable
- c. Simplified navigation menus
 - i. Extremely helpful, Very helpful, Somewhat helpful, Not helpful, Not applicable