A Virtual Lab Maturity Model for guiding the co-development of advanced Virtual Research Environments

Zhiming Zhao[1,2,*] W. Daniel Kissling[1] Geerten M. Hengeveld[3] Ioannis Athanasiadis[4]
Karline Soetaert[5] Andries Hof[6]
[1]University of Amsterdam, Science Park 904, Amsterdam, the Netherlands
[2]LifeWatch ERIC Virtual Lab Innovation Center, Science Park 904, Amsterdam, the Netherlands
[3]Netherlands Institute of Ecology, NIOO-KNAW, Droevendaalsesteeg 10, Wageningen, the Netherlands
[4] Wageningen University & Research, Wageningen, the Netherlands
[5] Royal Netherlands Institute for Sea Research, the Netherlands
[6] Rijksinstituut voor Volksgezondheid en Milieu, the Netherlands
z.zhao@uva.nl
*corresponding author

Abstract—Advanced data science and AI technologies, such as Digital Twins, offer new avenues for innovation in addressing complex scientific problems. However, developing an effective Virtual Research Environment (VRE) to realize these innovation potentials is technically challenging. Incorporating the emerging technologies into the research lifecycle as a new problem-solving paradigm requires assembling diverse software and technological components, which are often not yet fully production-ready. Additionally, successful implementations are often limited to demonstrator cases that are difficult to generalize, leading to significant development risks when creating mature services within a VRE intended for large scientific communities. While agile practices can support development by promoting close engagement with end users throughout the process, their iterative nature does not inherently guarantee the evolution of a technically sustainable solution, potentially increasing risks in project delay or even failure. This paper introduces a Virtual Lab Maturity Framework designed to facilitate the co-development process between end users and development teams, ensuring a more coordinated and sustainable approach to building effective VREs.

Keywords–Digital Twin, Virtual Research Environment, Virtual labs and Research Infrastructure

1. Introduction

Emerging computing and data science technologies, such as digital twins and foundational AI models, frequently attract early adopters who test these innovations and explore new paradigms for studying scientific problems. A majority of the scientific communities, often experience technical barriers in integrating these innovations in their research practice. Facilitating the use of new data science technologies through research infrastructure services and supportive environments can lower these technical barriers, allowing a larger community to conduct research on remote and distributed platforms. This includes automating scientific workflows, integrating diverse data types, managing research data, and fostering collaboration with partners. During the past decades, research infrastructures have been developed for communities in specific domains, such as those in the earth, environmental and life sciences, e.g., LifeWatch[1], eLTER[2], Euro-Argo[3], ICOS[4], and EPOS[5]. Those infrastructures provide research assets, e.g., research data, software tools, and infrastructure capacities, via various services like catalogues, data repositories, workflow automation, and user support.

A Virtual Research Environment (VRE)[2] aims to integrate all essential tools required by researchers for working in a digital environment and to provide a user-centered, customizable platform for conducting data-intensive, computing, and AI-driven activities over remote infrastructures. VREs are increasingly recognized as key components of modern research infrastructures. They are sometimes referred to by different names; for example, in the United States, the term “Science Gateway” [1] is commonly used.

A successful Virtual Research Environment (VRE) often results from years of experience and lessons learned from various case studies. However, developing a VRE to support emerging technologies like Digital Twins and AI workflows is often challenging.

For instance, enabling researchers to incorporate Digital Twins into their work requires making extensive research assets —such as data, models, and software— available through the VRE. This process can be time-consuming and labor-intensive. Additionally, advanced VRE features, such as integrating software components like physical system models, data processing pipelines, and visualization tools for Digital Twin outputs, are typically specific to particular applications. Gathering requirements and implementing these features as functional components necessitates multidisciplinary team collaboration. Dedicated support for a single small-scale Digital Twin may fulfill needs of a specific use case but does rarely lead to

[1]https://www.lifewatch.eu/
[2]https://elter-ri.eu/
[3]https://www.euro-argo.eu/
[4]https://www.icos-cp.eu/
[5]https://www.epos-eu.org/

2642-6587/25/$31.00 ©2025 IEEE
DOI 10.1109/SOSE67019.2025.00007

software that is maintainable and stable enough to serve as general and portable infrastructure services. Incrementally developing VRE features with continuous input from end-users—an approach often recognized as essential—can improve outcomes. However, traditional agile practices do not always result in sustainable, scalable solutions suitable for a large user community.

In this paper, we aim to address these challenges by proposing how a reference model and a quality assessment framework can facilitate co-development among various stakeholders in advanced VREs. The rest of the paper is organized as follows. First, we will discuss the background and related work regarding VRE development. Next, we will introduce the proposed reference model and quality framework, which guide the co-development approach. Finally, we will illustrate how this approach is applied using Digital Twin as an example within the ongoing Dutch project, LTER-LIFE[6].

2. Related work

When describing support systems for research activities, terms such as Research Infrastructure, Virtual Research Environments (VREs), and Virtual Labs are often used interchangeably in various contexts. Based on [6] and [13], we distinguish these concepts as follows: *Virtual Research Environment* is often operated as a *software platform* in a *Research infrastructure* as a *service*. Via this *Platform service*, users can create a customized *Virtual Lab* to conduct research for a specific scientific purpose. Services on *Research infrastructures* often require capacity from *e-Infrastructure* to deploy and operate.

In this section, we will review the state of the art regarding the development of virtual research environments that support emerging technologies in scientific research.

2.1 Jupyter notebooks in VRE

One of the key challenges in developing Virtual Research Environments (VREs) is ensuring a high-quality user experience during research activities. Researchers typically prefer client software that aligns with their daily workflows and the common practices within their community. For example, data scientists across various domains frequently use Jupyter Notebooks and Python, while the ecological modeling community often relies on RStudio and R.

Computational notebook-based environments, such as Jupyter Notebooks and R Markdown, combine narrative explanations with code fragments, providing an interactive interface for experimentation. These environments have become essential tools for many researchers' daily work. However, native notebook environments primarily support individual users and often depend on external tools to manage the full research lifecycle—such as search engines for discovering data, software, and notebooks from remote sources; workflow engines or orchestrators to run notebooks on remote infrastructure when local resources are insufficient; and version control systems like Git or social media platforms for collaboration and sharing results.

[6]https://lter-life.nl/

There are various approaches to supplement these notebook environments with additional features. One method is integrating environments like Jupyter into existing VRE platforms; for instance, the D4Science VRE provides a customized Jupyter Hub for its users. Another approach is extending Jupyter with missing functionalities—such as discovery, cloud automation, and workflow composition—using extensions; an example of this is the NaaVRE system [13]. As the needs of scientific users evolve—such as handling new data types or managing workflows for novel experimental scenarios—developing effective research support systems will require collaborative design and development efforts between domain researchers and research software engineers.

2.2 Research software

Virtual Research Environments (VREs) are essential software tools designed to support scientific research. These are often referred to as research software. Over the past decades, the significance of research software has been increasingly recognized by the research community within the context of open science. Broadly, all software involved in the research lifecycle—including tools that facilitate research activities and software produced by researchers—can be considered research software. However, in a more specific sense, research software refers to software developed explicitly for research purposes, requiring domain-specific knowledge to develop and use.

Research software are often classified into three tiers based on their prevalence and reach[7]:

1) **Analysis code (tier one):** Created by scientists during their research to support specific activities such as data generation, transformation, or visualization. For example, notebooks developed by researchers using Python or R are considered analysis code.
2) **Tool prototypes (tier two):** Developed to explore new ideas or standardize processes, such as tools for retrieving data from specific repositories or transforming data formats.
3) **Software infrastructures (tier three):** Designed to support a broader user base in conducting research based on well-established or widely adopted concepts. These infrastructures are sometimes developed by professional software engineers. Virtual Research Environments (VREs) can be viewed as research software infrastructures. Through VREs, users—such as researchers can create customized instances (often called Virtual Labs) to facilitate domain-specific research.

The development of research software is deeply integrated into the research lifecycle. Modern software engineering practices—such as software-oriented engineering, Agile methodologies, and DevOps—are often employed in research software development to improve quality, collaboration, and deployment.

[7]https://everse.software/RSQKit/three_tier_view

2.3 Software service engineering: agile and DevOps

Over the past decades, we have witnessed a revolutionary shift driven by services in software engineering—not only in modeling software architecture and integrating complex applications, but also in the online delivery of business value enabled by digitalization.

Large software systems are often composed of multiple components spread across distributed locations on the internet. These components are modeled as services and interconnected through standardized protocols, such as Remote Procedure Call (RPC) over HTTP, and data formats like XML or JSON, which allow for storing and exchanging data efficiently between systems, especially in web services and APIs. These services are typically deployed on cloud infrastructures to ensure high runtime quality, utilizing enhancements like load balancers and auto-scaling features. Maintaining the continuous operation of high-quality services is critical for delivering the intended business value. Achieving this requires DevOps practices, which facilitate continuous monitoring, runtime adaptation, and frequent updates of service implementations to meet evolving requirements and introduce new features.

In research software engineering, DevOps practices are commonly adopted for Tier 2 and Tier 3 software, as discussed earlier. The testing, integration, and deployment of Virtual Research Environment (VRE) platforms are automated through CI/CD (Continuous Integration/Continuous Deployment) pipelines, triggered whenever source code changes occur. Docker containers are used to encapsulate software services, enabling deployment across distributed infrastructures. DevOps fosters closer collaboration between development and operations teams, with development activities often managed using Agile practices—resulting in shorter, incremental delivery cycles compared to traditional waterfall models[12]. Industry-developed practices such as LEAN[5] optimization—aimed at minimizing waste—and IT Service Management—focused on enhancing the sustainability of IT assets—are integrated into DevOps workflows. As a result, DevOps practices encompass a collection of methods designed to optimize the entire development and operational lifecycle.

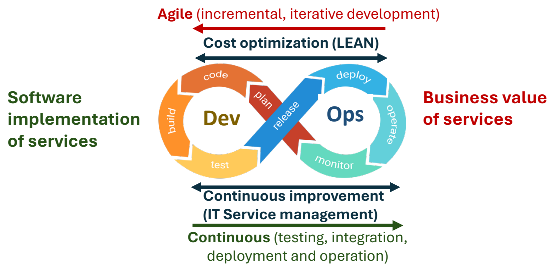
Agile (incremental, iterative development)
Cost optimization (LEAN)
Software implementation of services
code
plan
deploy
Dev
release
Ops
build
operate
test
monitor
Business value of services
Continuous improvement (IT Service management)
Continuous (testing, integration, deployment and operation)

Figure 1. Software services, DevOps and Agile.

The development lifecycle of VRE is often closely related to research activities, where research activities, software prototype, development, and service operations often follow practices in their lifecycles. The lifecycle research software across those different cycles.

2.4 Reference model and reference architecture

The development of Virtual Research Environments (VREs) involves multiple stakeholders, including domain researchers, modellers, research software engineers, training and support staff, and infrastructure specialists. Effective communication among these groups, particularly for requirement gathering and planning, can be challenging due to the lack of shared vocabulary. Establishing a common set of vocabularies and clearly defining the relationships among them is therefore essential. Approaches based on multiple viewpoints, such as Open Distributed Processing (ODP) [7], have been adopted in various projects, including ENVRI and ENVRIplus, to identify the key concepts needed to model the system effectively.

A reference architecture provides a standardized pattern outlining the core functional components of a system. For example, the Integrated Enhanced Virtual Research Environment (E-VRE) [9] serves as a reference architecture for classical data science, primarily supporting data discovery, processing, and workflow management. The primary goal of E-VRE is to offer an interoperability layer that facilitates workflows across various data infrastructures through semantic mapping, customizable authentication and authorization mechanisms, and a service-oriented architecture. However, this architecture lacks advanced description logic and integration capabilities for simulations, machine learning, and user interactions, which are essential components of Digital Twin applications.

2.5 Research Softwaer quality

Software quality is often evaluated using a quality model, which typically consists of a set of quality characteristics. Each characteristic can be further mapped onto one or more observable metrics or indicators. To manage the complexity of these characteristics, they are often decomposed into sub-characteristics or grouped into broader quality dimensions. For example, the ISO/IEC 25010 [11] standard defines nine quality characteristics, such as functional suitability, performance efficiency, compatibility, and safety. The research software community, including initiatives like EOSC and FAIR4RS, has developed customized software quality models for research software. In the EU EVERSE project[8], these existing models have been further harmonized into RSQKit (Research Software Quality Kit) [3], which proposes eleven quality dimensions—derived from the ISO standards [11] plus two additional dimensions: FAIRness and Sustainability, as shown in Fig.2.

Apart from quality characteristics, software quality can also be assessed based on its readiness level for operation. For example, the Technology Readiness Level (TRL) [9] is frequently used to evaluate project outputs, such as research software or Virtual Research Environments (VREs). Other models include the FAIRness maturity model [10], which assesses the level

[8]https://everse.software/

[9]https://euraxess.ec.europa.eu/career-development/researchers/manual-scientific-entrepreneurship/major-steps/trl

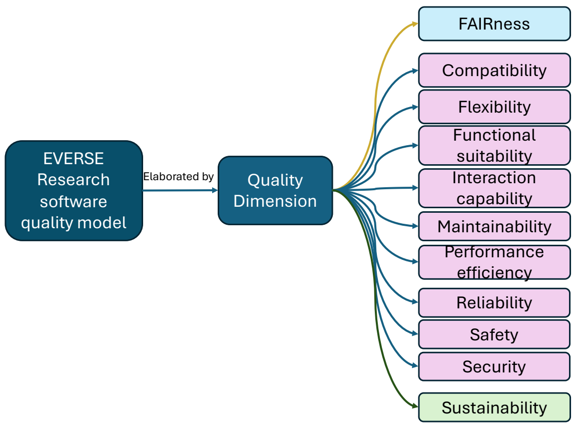

Figure 2. Research software quality dimensions recommended by the *EVERSE* RSQKit.

of FAIRness, and Digital Twin (DT) maturity models, which evaluate the development stage of digital twins [8].

2.6 Challenges and gap analysis

From the related work, we can draw the following conclusions:

1) Computational notebooks are widely used in data science and AI, and have been integrated into Virtual Research Environments (VREs) in various ways. However, supporting emerging technologies such as Digital Twins requires additional development and extensions to these environments.
2) VREs are typically operated as platforms enabling end users to set up domain-specific virtual labs for research activities. Nonetheless, a high technical readiness level of individual components within a VRE or virtual lab does not necessarily guarantee the maturity or robustness of the workflows or applications constructed on top of these components.
3) The ongoing demand for new features and the integration of cutting-edge technologies or research paradigms make VRE development inherently risky. This challenge lies in balancing the suitability and flexibility of virtual labs with the stability and maintainability of the underlying VRE infrastructure.
4) Data-centric scientific experiments often require substantial infrastructure capacity, especially when processing large datasets or running high-resolution models. If the underlying software is poorly tested or of low quality, this can lead to underutilized infrastructure and increased difficulties in debugging errors.

To bridge those gaps, we will present our proposed reference model guided quality maturity level based incremental approach for developing the advanced Virtual Research Environment.

3. A REFERENCE MODEL GUIDED MATURITY LEVEL BASED INCREMENTAL APPROACH

As a type of Tier 3 research software, a Virtual Research Environment (VRE) is typically developed using service-

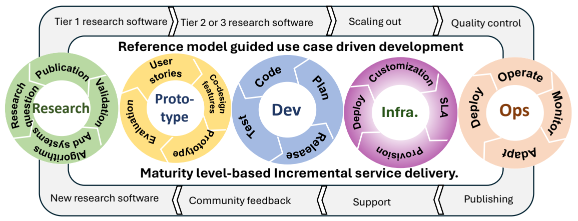

Figure 3. VRE as a research software in the lifecycle of research, software prototype and service DevOps.

oriented modeling, cloud computing, and DevOps practices. The development of a VRE often involves a combination of research activities, research software prototyping, and continuous integration and deployment processes, as illustrated in Fig. 3.

3.1 Reference model guided incremental development

The usefulness of a Virtual Research Environment (VRE) is crucial to its overall success; the features it offers must address the specific pain points faced by end users, such as scientists. Well-defined use cases can help the development team focus on key user stories and identify which features should be prioritized during implementation. To support this, developing a shared vocabulary for describing the various services within the VRE is essential. Incorporating new technologies, such as Digital Twins, for domain researchers requires additional efforts—such as conducting technology surveys and forming expert groups—to define relevant terms and concepts within the specific scientific context.

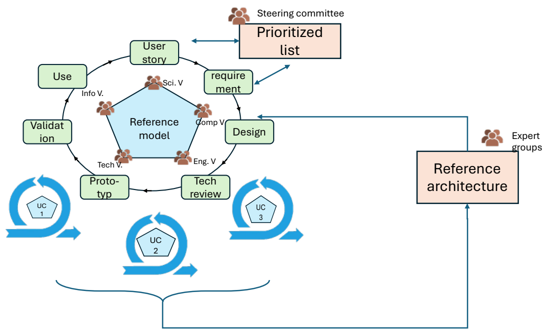

Figure 4. Reference model enhanced research software lifecycle

To guide the incremental co-development of the VRE, several key challenges must be addressed:

1) How to effectively facilitate communication among the diverse stakeholders involved in VRE development.
2) How to select use cases and user stories to prioritize features for development.
3) How to steer the evolution of the system so that incremental features converge into a final, robust, and maintainable solution.

Fig. 4 illustrates a reference-guided co-development approach for VRE development. A multi-viewpoint-based reference

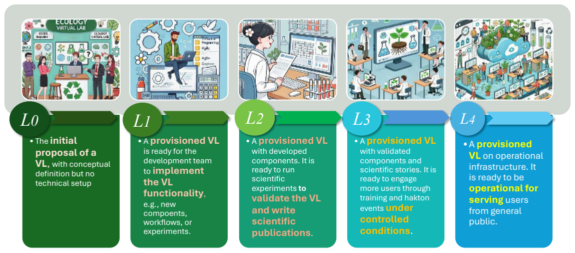
ECOLOGY VIRTUAL LAB
L0
• The initial proposal of a VL, with conceptual definition but no technical setup
L1
• A provisioned VL is ready for the development team to implement the VL functionality, e.g., new compoents, workflows, or experiments.
L2
• A provisioned VL with developed components. It is ready to run scientific experiments to validate the VL and write scientific publications.
L3
• A provisioned VL with validated components and scientific stories. It is ready to engage more users through training and hakton events under controlled conditions.
L4
• A provisioned VL on operational infrastructure. It is ready to be operational for serving users from general public.

Figure 5. The basic definition of the Virtual Lab maturity levels.

model provides an ontological framework that enables different stakeholders—such as domain researchers and DT experts (science viewpoint), data scientists (information viewpoint), modellers (computational viewpoint), research software engineers (engineering viewpoint), and infrastructure specialists (technology viewpoint)—to collaborate effectively. A steering committee is responsible for selecting science cases and guiding activities like requirement analysis, user story formulation, and analysis.

Selected science cases are developed within Virtual Labs following Agile practices. Multiple Agile teams can work concurrently on a use case, with experts periodically reviewing progress to assess architectural patterns and choices. These assessments help document and establish the reference architecture of the Virtual Research Environment.

The development progress of the Virtual Labs will be evaluated based on their maturity in supporting research activities.

3.2 Virtual Lab Maturity model

The maturity level of a Virtual Lab is assessed based on key milestones in the scientific research lifecycle, such as formulating the research plan, preparing research software and workflows, conducting full-scale experiments to test hypotheses, training the community, and disseminating research results to a broad audience. A quality framework is proposed to model the Virtual Lab's maturity across five levels, as illustrated in Fig. 5.

1) A *Level 0 Virtual Lab (L0 VL)* refers to the conceptual design phase of a virtual lab, where a potential Lab Principal Investigator proposes a science case based on a research question which will require advanced data science or AI technologies provided by the virtual research environment, e.g., for a digital twin. A L0 VL does not yet have a concrete instance deployed on the VRE. Based on the availability of the data, software, development cost, and the scientific potential, a prioritized L0 VL will be selected to create
2) A *Level 1 Virtual Lab (L1 VL)* on a Virtual Research Environment. The VRE will make all the required services available by the L1 VL and create a customized view for the users of the L1 VL to develop new workflows or tools needed for the scientific applications. A L1 VL development team will be established to implement the missing components, preferably following Agile practices. This Agile team will have a close connection with the VRE supporting team to request special technical support. The aim of the L1 VL is to develop and test research software; thus the full-scale data is not always needed in the L1 VL. When all the software and components have been tested and integrated, it will trigger the creation of
3) A *Level 2 Virtual Lab (L2 VL)* for running experiments with full scale of data and models to tackle the research problem. An L2 VL often requires extra computing and storage resources to load full data sets or run heavy computations. The L2 VL aims to obtain high-quality scientific results for peer-reviewed publications, together with the involved data sets, research software, and workflows.
4) A *Level 3 Virtual Lab (L3 VL)*, based on the results of a successful L2 VL, will be deployed for tutorial or hackathon purposes. An L3 VL is customized for a specific training purpose and user groups to test computing workflows, replicate the scientific results, or collect feedback for further development. Multiple L3 VLs can be created for a specific L2 VL.
5) A *Level 4 Virtual Lab (L4 VL)* will be created when the content of a L3 VL becomes well tested and with good acceptance from the user community. The L4 VL aims to turn the Virtual Lab as a service that can be deployed in an operational research infrastructure to serve a broader user community.

3.3 Stakeholders in Virtual Lab co-development

Various stakeholders are involved in the co-development processes of Virtual Labs and the underlying Virtual Research Environment. Their roles can be categorized into the following four groups:

1) *Domain researchers* who utilize Virtual Labs to conduct scientific experiments based on specific hypotheses, such as ecologists involved in the LTER-LIFE project. Typically, domain researchers propose the initial L0 VLs and serve as scientific coordinators for the L1 VLs, as well as end users of the L2 VLs. Researchers with high scientific potential—referred to as *golden* users—receive prioritized support to develop both L1 and L2 VLs. For the L3 VL stage, other domain researchers, such as early-career scientists—often called *silver* users—are invited to use the virtual labs. *Ivory* users, or public users, are engaged primarily in operating the L4 VLs as part of the operational infrastructure.
2) *Public users* can be citizen scientists, students, and any users interested in using the VRE or the Virtual Labs provided by the domain researchers.
3) *Modelers* who develop the models required to address the scientific questions, using mechanistic, statistical, or machine learning approaches.
4) *Research Software Engineers* specializing in Virtual Labs develop the software components needed to integrate advanced technologies, such as AI workflows, models within digital twins, and visualization tools.

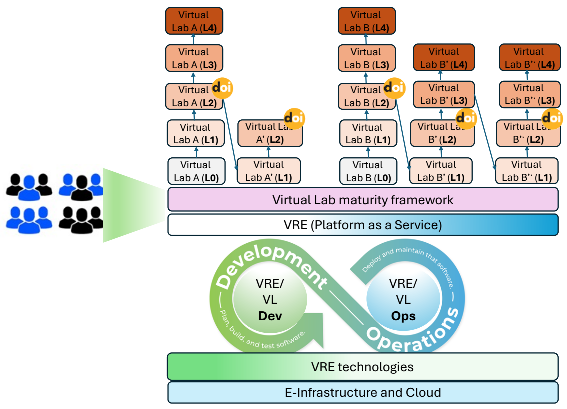

Figure 6. The evolution of different maturity levels of Virtual Labs.

5) *Training and support staff* responsible for organizing training activities and transferring knowledge to the user community.
6) *Infrastructure operation support staff* who provision the computing and storage resources required for different maturity levels of Virtual Labs, tailored to specific activities such as training or full-scale experiments.

Leveraging the maturity framework, a new role called the Virtual Lab Coordinator can be introduced to act as an intermediary between the VRE development team and potential domain researchers, facilitating communication and aligning development efforts with scientific needs.

3.4 The evolution of virtual lab maturity levels

Virtual labs do not always progress sequentially through the different maturity levels. For example, an L1 VL might focus on data product processing and serve as a building block for creating subsequent L2 VLs, but it does not necessarily need to produce a scientific paper to advance. Additionally, a new L0 or L1 lab can be established based on an existing virtual lab, regardless of its current maturity level. For instance, a researcher may be inspired by prior work and propose a different research idea. However, for an L2 virtual lab, all relevant assets—such as data, software, models, and workflows—must be made available together, especially if the resulting research is intended for open science publication.

Fig. 6 depicts the basic maturity level and their involved stakeholders.

The development teams of the Virtual Labs will be supported by the maturity management framework and will collect feature requests for the Virtual Research Environment (VRE) team. The maturity management framework manages the lifecycle of virtual labs, assesses the maturity levels of the labs, and stores the metadata of different virtual labs, including new features that virtual lab teams request the VRE to provide.

Figure 6 illustrates the potential collaboration among these teams.

4. Case study

LTER-LIFE is a project funded by the Dutch Research Council, aimed at developing a digital twin (DT)-oriented Virtual Research Environment for the ecological research community. The operational VRE platform and Virtual Labs will be deployed on national e-infrastructures such as SurfSara[10] and the European Infrastructure LifeWatch. NaaVRE[11] is utilized as the core technology for the DT VRE.

In LTER-LIFE, a reference model is established at the outset of the project, based on the ENVRI reference model [4]. This model reviews Digital Twin-related concepts through recent literature and community interviews. It serves as a basis for collecting user stories from scientific communities. The maturity framework guides the coordination between VRE developers and Virtual Lab teams, helping to allocate development and infrastructure resources effectively. For example, infrastructure capacity will be allocated to L2 Virtual Labs only when their software is ready for large-scale experiments. Additionally, targeted training activities can be organized more efficiently when specific VLs are required. The project is ongoing, and many actions remain on the agenda:

1) A management software is under development to automate the assessment and administration of Virtual Labs across different maturity levels;
2) Tools for (semi)automating metadata generation for virtual labs, as well as facilitating the enhancement of Findability, Accessibility, Interoperability, and Reusability (FAIR) of the virtual lab and its components—including data, models, and digital twins—will be essential in reducing the manual effort required from scientists;
3) Aligning the maturity framework with other existing quality models related to research assets—such as the data FAIRness maturity model, digital twin maturity framework, and research software quality model—is also an important area of focus.

5. Summary

The maturity framework was originally proposed for Digital Twin-oriented Virtual Research Environments and Virtual Labs. However, the concepts of maturity levels can be generalized for other non-DT cases. As an emerging problem-solving approach in scientific domains such as ecosystems, digital twins are still in their infancy regarding adoption by domain researchers for addressing real scientific challenges. Developing the maturity management framework faces challenges related to integrating automated testing, ensuring quality control of research software, and promoting the FAIR principles for digital assets.

Acknowledgment

This research was made possible through funding from the Dutch Research Council (NWO) Large-Scale Research Infrastructures (LSRI) programme for the LTER-LIFE

[10]https://www.surf.nl/
[11]https://naavre.net/

(http://www.lter-life.nl) infrastructure (grant 184.036.014). The work is also partial funding by several European Union projects: ENVRI-Hub Next (101131141), EVERSE (101129744), BlueCloud-2026 (101094227), OSCARS (101129751).

References

[1] Prasad Calyam, Nancy Wilkins-Diehr, Mark Miller, Emre H. Brookes, Ritu Arora, Amit Chourasia, Douglas M. Jennewein, Viswanath Nandigam, M. Drew LaMar, and et al. Measuring success for a future vision: Defining impact in science gateways/virtual research environments. *Concurrency and Computation*, 33(19):e6099, October 2021.

[2] Leonardo Candela, Donatella Castelli, and Pasquale Pagano. Virtual Research Environments: An Overview and a Research Agenda. *Data Science Journal*, 12(0):GRDI75–GRDI81, 2013.

[3] Neil Chue Hong, Faruk Diblen, Daniel Garijo, Jason Maassen, Giacomo Peru, Aspasia Orfanou, Nikos Pechlivanis, and Fotis Psomopoulos. EVERSE RSQkit Reference Framework, November 2024.

[4] Abraham Nieva De La Hidalga, Alex Hardisty, Paul Martin, Barbara Magagna, and Zhiming Zhao. The ENVRI Reference Model. In Zhiming Zhao and Margareta Hellström, editors, *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences*, volume 12003, pages 61–81. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[5] Shaman Gupta and Sanjiv Kumar Jain. A literature review of lean manufacturing. *International Journal of Management Science and Engineering Management*, 8(4):241–249, November 2013.

[6] Keith Jeffery, Antti Pursula, and Zhiming Zhao. ICT Infrastructures for Environmental and Earth Sciences. In Zhiming Zhao and Margareta Hellström, editors, *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences*, volume 12003, pages 17–29. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[7] Peter F. Linington, Zoran Milosevic, Akira tanaka, and Antonio Vallecillo. *Building enterprise systems with ODP*. CRC press, Taylor & Francis group, 2012.

[8] Brett Metcalfe, Hendriek C. Boshuizen, Jandirk Bulens, and Jasper J. Koehorst. Digital twin maturity levels: a theoretical framework for defining capabilities and goals in the life and environmental sciences. *F1000Research*, 12:961, August 2023.

[9] Laurent Remy, Dragan Ivanović, Maria Theodoridou, Athina Kritsotaki, Paul Martin, Daniele Bailo, Manuela Sbarra, Zhiming Zhao, and Keith Jeffery. Building an integrated enhanced virtual research environment metadata catalogue. *The Electronic Library*, 37(6):929–951, November 2019.

[10] Research Data Alliance FAIR Data Maturity Model Working Group. FAIR Data Maturity Model: specification and guidelines. 2020. Publisher: Research Data Alliance Version Number: 1.

[11] Emma Sheppard. ISO/IEC 25010:2023 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model characteristics, sub- characteristics, definitions, and their proposed application to electric vehicle supply equipment (EVSE) software., March 2025.

[12] Kalpana Sureshchandra and Jagadish Shrinivasavadhani. Moving from Waterfall to Agile. In *Agile 2008 Conference*, pages 97–101, Toronto, ON, Canada, 2008. IEEE.

[13] Zhiming Zhao, Spiros Koulouzis, Riccardo Bianchi, Siamak Farshidi, Zeshun Shi, Ruyue Xin, Yuandou Wang, Na Li, Yifang Shi, Joris Timmermans, and W. Daniel Kissling. Notebook-as-a-VRE (NaaVRE): From private notebooks to a collaborative cloud virtual research environment. *Softw Pract Exp*, 52(9):1947–1966, September 2022.