

# Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

*“Our greatest glory is not in never falling,  
but in rising every time we fall.”*

Confucius (551–479 BC)

Teacher, editor, politician, and philosopher

Living lab methodologies need to enhance reactivity to changing requirements as these appear in a project. Agile methods allow for quick reactivity, but have been critiqued for not sufficiently taking into account the end-user perspective. In this article, we describe how to blend living lab methodologies with agile methods and, to this end, we present a Framework for Agile Living Lab projects (FALL). To make the framework actionable, we propose a number of actor roles. With concrete examples from living lab practice and a discussion of the theoretical basis, this article is relevant to both academics and practitioners.

## Introduction

Agile development and living labs have separately received much attention from innovation-driven practitioners and academics over the last decade (Almirall et al., 2008; Dybå & Dingsøyr, 2008; Følstad, 2008). Despite different backgrounds and foci, both concepts share some commonalities. As major goals common to both approaches, we identify shared ambitions to: i) increase cost efficiency, ii) augment stakeholder collaboration, and iii) cooperate with users. However, both approaches are characterized by some weaknesses. Although living labs champion end-user involvement in both design and development, results from user co-creation often fail to become incorporated in ongoing technological development cycles (Sauer, 2013). Given that innovation frequently has unintended outcomes (Sveiby et al., 2009) – such as unforeseen shifts in requirements – living labs react more rapidly to such shifts in scope. Agile methodologies, on the other hand, have been developed specifically to tackle shifting requirements, yet they lack a structured focus on users and do not target collaboration with them (Cajander, 2013; Singh, 2008).

The question that we aim to answer here is:

*How can we integrate agile methodologies, with their advantages for structuring flexible work processes, with living lab methodologies, which are known to be user-driven?*

Addressing this question should yield novel insights into how to conduct living lab projects, both from a theoretical and practical perspective, by means of a framework, which we have named the Framework for Agile Living Labs (FALL).

The article is structured as follows. First, we introduce FALL and its component phases. Next, we describe FALL's various actor roles and their associated tasks. Then, we illustrate how SCRUM acts as a backbone for this agile framework. Finally, we highlight the contribution of the study and its limitations, and we offer conclusions.

## FALL: Framework for Agile Living Lab projects

In this section, we present the essence of the FALL model, which is the result of an inductive (from practice to model) and deductive (from existing theory to model) process. The inductive part was done by participatory action research through hands-on experience in a variety of Flemish living lab projects run by imec.livinglabs ([www.iminds.be/livinglabs/](http://www.iminds.be/livinglabs/)) and the “VRT Proeftuin” ([deproeftuin.vrt.be](http://deproeftuin.vrt.be)). The theoretical foundations for the deductive part are rooted in design science research, a scientific discipline that aims to contribute to the scientific body of knowledge by building information systems. The main phases of FALL are derived from the

# Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

action design research (ADR) method (Sein et al., 2011). However, in order to make a research-oriented framework more practical and applicable in daily living lab practice, we implemented FALL using agile techniques.

Figure 1 illustrates FALL, which focuses on living lab projects running from the early stages of a project idea to the real-world evaluation of a working software prototype. This framework was created because living labs need a robust methodology to structure and value user feedback. Because innovation frequently has unintended or unexpected effects (Sein et al., 2011), living labs must adjust rapidly to user feedback. This need for rapid adjustment is further underlined by variability in the timelines of project objectives and the diversity of control points, as commonly experienced in living labs (Leminen & Westerlund, 2011). Each of FALL’s component phases is described in the subsections that follow.

### Phase 1: Problem Formulation

The first FALL phase, Problem Formulation, aims to produce a concise statement to scope the effort of the team, which should reflect the perspectives of relevant end users and stakeholders. There are different ways of gathering information on what the problem statement should contain. The living lab and user research literature has much to say on how to leverage insights and domain knowledge from representative end users and stakeholders, for example through co-creation techniques and focus groups. Another way is to create on-line crowdsourced ideation campaigns on general

topics that representative end users can participate in. Outputs of such efforts (e.g., story boards, user scenarios, Lego Serious Play models) in the problem formulation phase are by definition abstract, guiding the overall design and development efforts in the subsequent phases, where the outputs of co-creation become more concrete.

In addition to the co-creation of problem formulations, and as is customary in design science research, it is important to map the existing state of the art on the type of system being created. This step is often forgotten or not accounted for in living lab literature, yet creates an important baseline against which to gauge the innovation potential of the project.

From the problem definition, a first solution can be devised in the form of a set of assumptions to be tested by building minimum viable products (MVPs). However, these assumptions are often uncertain statements that should be verified. As in lean UX (Gothelf & Seiden 2013), selecting what assumption to test first can be done by prioritizing them in terms of high risk and low maturity. Risk refers to the consequences of the assumption being false while the project still holds the assumption to be true. Maturity is the amount of knowledge that the project team has regarding the assumption. Testing high-priority (i.e., high-risk and low-maturity) assumptions should be the focus of MVP 0 (see Example 1). The functionality of the MVP can be described using a SCRUM backlog that bundles and de-

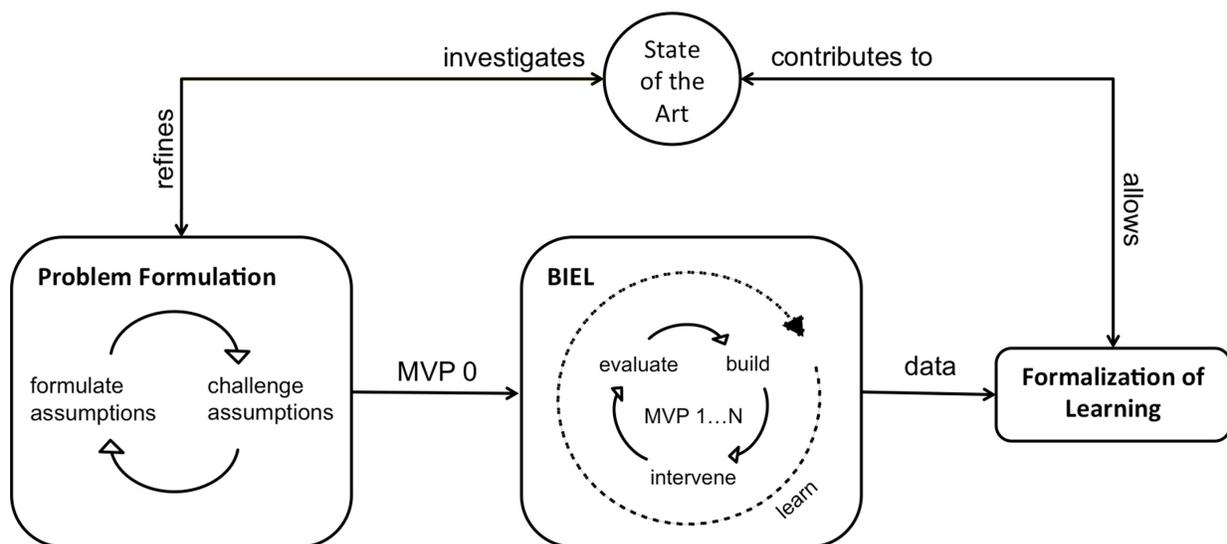


Figure 1. Framework for Agile Living Lab projects (FALL)

## Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

### Example 1. Testing high-priority assumptions with the first MVPs

In the ZWERM ([www.zwermgent.be](http://www.zwermgent.be)) project, we started from a very general problem statement: to engage smart citizens with the city through mobile applications. However, this problem statement was too broad to be actionable. By gathering feedback from different stakeholders in the project (including a great number of citizen inhabitants, which were the prospective end users of the project), we arrived at the following problem statement: “How can we build a system that allows neighbourhood citizens to play a game through which they increase social cohesion and use this social cohesion to take actions that are important to the neighbourhood?” A conceptual solution was imagined to answer this question, which could be expressed as a number of assumptions:

1. The question can be answered through Internet-of-Things-enabled public space furniture.
2. The question can be answered by building a game in which a “check-in” (a person swiping an RFID card on a card reader) will be the central game mechanism to engage players with the system and to get them to know each other.
3. The question can be answered by creating a number of missions that can be played with the user’s own device and inciting the user to take positive action.

Of these three assumptions, the second and third one were identified as being the highest risk and least mature. Therefore, they became the object of the first MVPs.

describes the functionality of the MVP as user stories. A user story has the form “as a <role> I can <functionality>”. Once MVP 0 has been defined in a backlog, it needs to be built, evaluated and tested, which happens in the next phase.

### Phase 2: BIEL (*Build - Intervene - Evaluate - Learn*)

The creation of the successive MVP prototypes takes place in the “build - intervene - evaluate - learn” (BIEL)

phase, in which the MVPs are always created (built), presented for feedback (intervention), and evaluated. The BIEL phase is derived from Sein and colleagues’ (2011) action design research method. The reason why these four activities are bundled into one phase is because they take place concurrently and not necessarily in sequence. Indeed, it is often the case in a living lab that an intervention in a real-world environment takes place over a longer period of time. In such a situation, building goes on while the intervention is taking place, as for example, bug fixes and change requests are addressed and integrated in the live functional prototype. Similarly, evaluation can take place during the intervention, for instance using qualitative observations.

Learning has been added as a separate loop in Figure 1 to indicate that it is highly ingrained in all activities taking place in the BIEL phase. Indeed, living lab practitioners, like other people, do not learn as a separate activity, but learn by performing all the necessary activities in the BIEL phase. For example, the actual building of MVPs yields extensive learning on what will work and what will not.

Such MVPs, in the context of FALL, would better be termed minimum viable *prototypes*. Indeed, all MVPs submitted for feedback are intermediate prototypes on the road to the outcome of a living lab project. Examples of prototypes often used in living labs include:

- paper prototypes (Snyder, 2003), which are sketchy representations of the graphical user interface (GUI)
- graphical user interface (GUI) mock-ups or extended paper prototypes with graphic style added (see Example 2)
- clickable prototypes that allow for a certain degree of interaction
- functional prototypes that can be used on the device of the user allowing real-world intervention and testing

In living labs, all participating stakeholders – including end users – can build or participate in the creation of prototypes. As such, we perceive co-design as a possible process of the BIEL phase. A technique we often use for such co-design is the lean UX “design studio” technique, in which end users are asked to individually draw the GUI for a system, after which a GUI design is made to reflect the group consensus.

## Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

### Example 2. GUI mockups as MVPs

After identifying the most important assumptions and deciding to make them the subject of MVP 0, the partners of the ZWERM project started working on several MVPs. MVP 0 consisted of GUI mockups for the website used to play the game. Feedback was gathered from representative end users, and a new version of the GUI design was created (MVP 1). Next, a functional version of the website and the check-in system were built and deployed at our research facility during a three-week period (MVP 2). Data was gathered through observations, a survey, and an analysis of the system logs. This allowed us to formulate answers to assumptions 2 and 3 (see Example 1). The feedback on assumption 2 was definitely positive, while the feedback on assumption 3 was more nuanced, with some missions working well and other not working at all. Based on what we learned during the intervention with MVP 2, we created a fully functional prototype (MVP 3) that was tested in real-life environments over a four-week period. Again, data was gathered through observations, a survey, and an analysis of the system logs.

### Phase 3: Formalization of Learning

The Formalization of Learning phase is where all that has been learned is reflected upon and placed in some format that is fit for consumption by an academic, a business, or a public audience (see Example 3). In the case of the former, it is important to contextualize the formalization of learning in terms of the existing scientific knowledge base (i.e., the state of the art) and the problem formulation. For a business audience, the formalization of learning may be more oriented towards insights that are useful for market introduction of the concepts, underlying the system.

### FALL Roles

In order to use FALL, a number of roles are necessary, because certain tasks must be performed during a FALL project. Assigning these tasks to roles in the living lab team can ensure that all tasks are accounted for. Starting from the main actions to be completed in our experience of running living labs (inductive) and from the literature (deductive), we identify eight key roles, which are borrowed from the literature on living labs, user research, agile methods, and user experience design:

### Example 3. Formalizing learnings from the MVPs

After the real-life environment intervention with MVP 3 and the evaluation based on the collected data, we drafted a number of papers, which described the system and formulated guidelines for the future design of similar systems. In addition, the core findings of the entire project were formalized into a project description (vision, architecture, business plan) for a spin off that leveraged the main elements of ZWERM.

1. *Process manager*: as in agile methods, the aim of FALL is to increase the amount of self-organization of the team. However, someone is needed to guide the team with the methodology or process of working with FALL. This is the responsibility of the process manager.
2. *User researcher*: takes the lead in getting input from users at different stages of the project. In addition, the user researcher has the responsibility of keeping the story backlog up to date from the perspective of the end users. They are also responsible for making sure the problem formulation is grounded in the state of the art from a non-technical perspective.
3. *Researcher*: active in an academic domain that is relevant to the design problem in the social or natural sciences, researchers contribute the insights that are needed to create innovations by leveraging knowledge from various research fields and applying them to the design problem at hand.
4. *Architect*: their role is to create the systems architecture and to update and prioritize the backlog in terms of the stories that are not facing towards the user, such as “the server backend should be able to automatically backup the user data that is stored in the database”. The architect is also responsible for making sure the problem formulation is grounded in the state of the art from a technical perspective.
5. *UX Designer*: responsible for creating MVPs that represent the GUI of the system. These can be wireframes, clickable prototypes, or GUI mock-ups. It is crucial to note that, although the UX designer holds the skillset to build these artifacts, creating them should never be done solely from only the perspective of the UX designer. Core to the philosophy of

## Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

FALL is that the UX designer should work with the feedback that was gathered from the project actors (other team members, representative end users, etc.).

6. *Developer*: responsible for translating the story backlog into functional MVPs.
7. *User*: involved in the project to bring domain-oriented knowledge to the team through co-design and usability testing processes, as guided by the user researcher. It is important that the users involved in the FALL project be as representative as possible of the user group that will eventually use the outcome of the living lab project.
8. *Stakeholder*: like users, stakeholders are also involved in contributing domain-oriented knowledge, but are not necessarily representative of the eventual user population. Stakeholders often hold higher-level interest than users and operate from a policy, commercial, or academic point of view.

### SCRUM as a Backbone

The Problem Formulation, BIEL, and Formalization of Learning phases require work to be done by a multitude of people. Within FALL, we propose to facilitate this work through SCRUM, the most widely adopted agile methodology. As a result, work in the living lab will be organized according to sprints, which are time-boxed iterations. At the beginning of each sprint, the objectives and the end-time of that particular sprint are defined. These objectives differ according to the phase of FALL in which the project is situated. In the Problem Formulation phase, the objectives will be focused on scrutinizing the body of knowledge and creating a problem formulation with related assumptions on how to address the defined problem. In the BIEL phase, the objectives will be on creating MVPs, testing them and defining new MVPs based on insights from building and testing previous versions. In the Formalization of Learning phase, the aim will be to contribute to the body of knowledge based on what was learned during the living lab's execution.

At the end of the sprint, a demonstration of the work is given and the whole process is evaluated in a sprint retrospective. Also, the backlog is updated according to the tasks that present themselves in the future milestones of the FALL project. The story backlog therefore constitutes a crucial FALL project management tool, as it keeps an overview of tasks in progress or to be com-

pleted. Such a backlog can be established at the start of each new BEIL cycle in which a new MVP is to be developed.

Generating the user stories in the backlog can be done by the project group (including users), for example by allowing group members to generate user stories on sticky notes. Subsequently, user stories can be prioritized using the categories of the MOSCOW method: “must have, should have, could have, won't have” (Figure 2). Most of the time, the former two categories will form the basis for BIEL activity in the upcoming cycle, yet user stories in the latter two should be kept for future use.

Conducting FALL as a SCRUM project provides the agility that is needed in living labs, where requirements are unstable due to ongoing end-user feedback. Having time-boxed iterations, at the start of which the premises of the project are questioned, helps in integrating new insights.



**Figure 2.** User stories being prioritized using the MOSCOW method  
(Photograph courtesy of the D-Pac project: d-pac.be)

# Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

## Conclusion

With agile methods, it is challenging to take the user perspective into account in a structural manner, whereas living labs often fail to incorporate emergent user feedback into running design and development processes. We proposed to address these issues by creating a framework that allows living labs to be executed in an agile way. FALL – the Framework for Agile Living Labs – draws on lean UX and SCRUM as agile methods. In addition, it takes design science research as a theoretical basis and structures the process along the lines of action design research. Through this article’s concrete examples, practical guidelines, and theoretical foundations, we have attempted to address both theoretical and practical implications. FALL can be used as a basis on which to align the research, design, development and evaluation activities that are core to many living lab projects, providing actionable guidelines to researchers and practitioners alike.

This article has made the following contributions. First, we introduced agile methodologies into the theory and practice of living labs. Second, we proposed an actionable, yet theoretically grounded set of constructs (MVP, BIEL, etc.) around which to conduct a living lab project in an agile way. We have placed this into a framework (FALL) and indicated how this framework can be supported methodologically through the different phases of FALL. Third, we proposed principles to be taken into account when performing living lab projects according to FALL: define project roles and use SCRUM as a backbone for project planning. As such, we contributed prescriptive knowledge to living lab theory and made a step towards overcoming practical hurdles that living lab projects can be confronted with.

A main topic in living lab research is how to gather insights from end users and stakeholders through participative techniques. Although we have hinted on ways to achieve this, we did not cover the participative processes in detail, given our focus on providing a general overview of the processes in FALL. As a key avenue for further research, greater elaboration is needed regarding the participative processes that are appropriate in different phases of FALL and the properties of their outputs.

## About the Authors

**Tanguy Coenen** is head of the software application prototyping team at imec and has a PhD in Business Administration from the Solvay Management School of the Vrije Universiteit Brussel in Belgium. During his career, he has been involved in many living lab projects, both at the national and the European levels. Tanguy fosters a variety of research interests but is focused on a “City of Things” project, which is applying living lab concepts to achieve the vision of a city as a living lab innovation platform.

**Sarie Robijt** currently works as a Service Design professional in retail business. As a service designer, she takes up several roles from user experience analyst to workshop facilitator and innovation coach. During her time at VRT Proeftuin, she worked on setting up and maintaining a living lab in the media innovation industry.

## References

- Almirall, E., & Wareham, J. 2008. Living Labs and Open Innovation: Roles and Applicability. *eJOV: The Electronic Journal for Virtual Organization & Networks*, 10: 21–46.
- Cajander, Å., Larusdottir, M., & Gulliksen, J. 2013. Existing but Not Explicit: The User Perspective in Scrum Projects in Practice. In *Human-Computer Interaction–INTERACT 2013*: 762–779. Berlin: Springer Berlin Heidelberg.
- Dybå, T., & Dingsøyr, T. 2008. Empirical Studies of Agile Software Development: A Systematic Review. *Information and Software Technology*, 50(9): 833–859. <http://dx.doi.org/10.1016/j.infsof.2008.01.006>
- Følstad, A. 2008. Living Labs for Innovation and Development of Information and Communication Technology: A Literature Review. *eJOV: The Electronic Journal for Virtual Organization & Networks*, 10: 99–131.
- Gothelf, J., & Seiden, J. 2013. *Lean UX: Applying Lean Principles to Improve User Experience*. Sebastopol, CA: O’Reilly Media Inc.
- Leminen, S., & Westerlund, M. 2011. *Open Innovation Paradoxes in Living Labs Networks*. Paper presented at the 13th EURAM Conference, Istanbul, Turkey, June 26–29.
- Ries, E. 2011. *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business.
- Sauer, S. C. 2013. *User Innovativeness in Living Laboratories: Everyday User Improvisations with ICTs as a Source of Innovation*. The Netherlands: University of Twente.

## Heading for a FALL: A Framework for Agile Living Lab Projects

Tanguy Coenen and Sarie Robijt

Sein, M., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. 2011. *Action Design Research. Management Information Science Quarterly*, 35(1):37–56.

Singh, M. 2008. U-SCRUM: An Agile Methodology for Promoting Usability. In *Proceedings of AGILE '08*: 555–560. Washington, DC: IEEE.  
<http://dx.doi.org/10.1109/Agile.2008.33>

Snyder, C. 2003. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Burlington, MA: Morgan Kaufmann.

Sveiby, K. E., Gripenberg, P., Segercrantz, B., Eriksson, A., & Aminoff, A. 2009. Unintended and Undesirable Consequences of Innovation. In *Proceedings of the XXth ISPIM Conference*, Vienna, Austria, June 21–24 June.

Westerlund, M., & Leminen, S. 2011. Managing the Challenges of Becoming an Open Innovation Company: Experiences from Living Labs. *Technology Innovation Management Review*, 1(1): 19–25.  
<http://timreview.ca/article/489>

**Citation:** Coenen, T., & Robijt, S. 2016. Heading for a FALL: A Framework for Agile Living Lab Projects. *Technology Innovation Management Review*, 7(1): 37–43. <http://timreview.ca/article/1048>



**Keywords:** living labs, design science research, agile, methodology, SCRUM, lean UX