Relationship of requirements engineering to innovation prototyping

Mervi Ranta, Kasperi Kaija, Henrik Asplund and Seppo Sahi
PM&RG – Product Modelling and Realization Group
Department of Computer Science and Engineering
Helsinki University of Technology
P.O.B 5400, FIN-02015 HUT, Finland
fax +358-9-451 3293
tel. +358-9-451 4807, Mervi.Ranta@hut.fi
Kasperi.Kaija@hut.fi
tel. +358-9-4515197, Henrik.Asplund@hut.fi
tel. +358-9-451 3371, Seppo.Sahi@hut.fi

Abstract

This papers explorers the relationship of requirements engineering to innovation prototyping. The former is part of product development whereas the later is committed to pre-product development. Thus, they have fundamental differences in their objectives. Requirements engineering is solution oriented and its core is validation of user needs. Innovation prototyping is problem oriented and its core is experimentation settings based on prototypes. In spite of the different objectives the approaches have common characteristics such as making the trace of the development accessible to all stakeholders and allowing them to share the design documentation in comprehensive form. Actually requirements engineering is embedded in innovation prototyping and innovation prototyping is input for requirements engineering. Finally the paper proposes service idea generation, that applies innovation prototyping methodology, as a playground for product development, of which requirement engineering is an inherent part.

Introduction

Innovation prototyping is an approach and methodology for service idea generation, service design and development. It is developed by the Product Modelling and Realization Group at the Helsinki University of Technology. The objective of innovation prototyping is to generate new ideas for future services that provide a basis for the product development. This is done by cumulating a pool of ideas and service prototypes that include background information that gives a sound basis for the following product development. The foundation of innovation prototyping is information management that makes the development process transparent to all stakeholders.

Particular attention is put on experimentation settings allow discovery of features and testing of hypothesis. The built innovation prototypes specified according to the requirements of the experimentations and focused on the features that are crucial for the experimentation setting. The aim is to allow early evaluation of service ideas, profit opportunities, user aspects, and technological feasibility of service ideas.

Innovation prototyping is applied in the development of mobile and ubicomp (ubiquitous computing) services. In this area the fast development and rough competition requires a particular methodology for service idea generation in order to create service ideas that can be forwarded to productification and product development.

An innovation prototype differs from conventional product prototypes, since it is a not intended to match or present the final product, but rather be a tool that allows efficient demonstration and testing of service ideas and services. Thus it is often different from any product and focuses on the most essential features of the issue that is to be demonstrated or experimented.

This paper is written to explore the relationship of innovation prototyping, which is a methodology for service idea generation, and requirements engineering, which is a methodology for ensuring that right requirements are fulfilled in the product development. First, requirements engineering is introduced briefly according to existing research. Then, the objectives of the approaches are compared in terms of problem orientation versus solution orientation, and experimentation setting versus validation. Then innovation prototyping and requirements engineering are discussed as methods for service idea generation and product development. This leads to noticing that requirements engineering is embedded in innovation prototyping and that innovation prototyping is input for requirements engineering. This is followed with a description of service idea generation as a playground for product development. Finally some conclusions are given.

Background

Requirements engineering is a 30-year-old term, designed to describe the actual process of answering the most important question of any software project. Namely what to do. [7 (p. 2)]. Developing software is traditionally divided into four phases; requirements, design, code and testing. [7 (p.1)] Having answered poorly during the first phase, that of the requirements, can result to unsatisfactory software and therefore cause huge extra costs, needed to make changes into already completed software. The purpose of requirements engineering is to make sure one does end up with the right and correctly formatted answers in the form of requirements. [5 (p.14)]

Requirements engineering is a process of four interlocking parts. These are requirements elicitation, requirements analysis and negotiation, requirements documentation and requirements validation. In reality requirements engineering process consists of constant interaction between consecutive parts and iteration after iteration until satisfactory requirements are reached. [5 (p.14)]

Requirements elicitation aims to the understanding and describing the needs of the users. Here users are those persons that the resulting software is being built for. Requirements elicitation strives not to document ready requirements, but to find user needs using methods of observation, interview and studying, preferably executed in the users' own environment and in the target domain for the resulting software. In practice requirements engineering process is finding out user needs and turning them into requirements. User need is an informal data oriented in the context of use and a requirement is a property to be met in order to satisfy the user's need. [5 (p.14-17)]

As the elicitation process produces requirements, they are to be analyzed and then require to be negotiated whether to be included in the requirements documentation. The goal is to filter out all trivial requirements and prepare the rest for more formal representation. [5 (p.16)]

Requirements documentation requires that the analyzed requirements be documented in a formal manner. It would be beneficial that this manner would follow a representation understandable to users. This would help the validation of the requirements that follows next, aimed detect problems in the requirements documentation. Requirements documentation is always an interpretation of all significant user needs to be responded to and therefore require user feedback. A poor representation of requirements can foil the whole requirements engineering process. [5 (p. 16-17)]

The resulting requirements documentation should emerge from the requirements specification process as a specification and therefore be able to function as a basis for designing software. Placing design requirements into the requirements specification should be avoided. [3 (p.4)] Requirements engineering has been successful if the resulting document correctly documents all the necessary requirements for the particular system in a manner that leaves no change to misinterpretation. There also cannot be any inconsistencies between the resulting document and any higher-level document or inconsistencies between individual requirements. If some aspect of the system cannot be described, then this must be explained in the documentation. All requirements are to be ranked in a manner that helps to understand how a particular requirement contributes to the overall system. All requirements must also be able to be verified from the resulting software product as requirements met. All requirements must be able to be traced to their origin and must have an unambiguous name that can be used as reference. The resulting document must also be modifiable in a manner that retains its integrity. [3 (p. 4-8)]

Requirements engineering is often considered difficult. There is a high and well-documented risk that software projects fail mainly because of causes that are requirements engineering related. Either requirements engineering is done poorly or just badly. Formal method is often necessary to assure proper use and execution of the requirement engineering process. Successful requirement engineering creates a basis for a successful fulfillment of user needs. [7 (p. 1-2), 4]

Objectives of innovation prototyping and requirements engineering

The difference of innovation prototyping and requirement engineering is that the former one is devoted to service idea generation and the latter to product development. They have many common objectives such as giving proper emphasis to user needs and attempting to trace the sources and development of development data. However, since requirements engineering methods are for product development and innovation prototyping is committed to the pre-product development, they have distinct objectives. In the following we compare two differences that are crucial for the methodologies. The first one is problem orientation versus solution orientation and the second one is experimentation setting versus validation.

Problem orientation vs. solution orientation

Innovation prototyping utilizes the concepts and structuring of scenarios, services, use cases and realization as shown in Figure 1 and explained in [1]. Requirements engineering usually uses the concept product or software in referring to the result of product development. Usually they do not distinguish the scenario and use case concepts. However, requirement engineering uses the similar and resembling concepts in sufficiently analogous way, so these concepts can be here used in explaining the orientations in the approaches.

The left hand side of Figure 1 shows how innovation prototyping aims at revealing common denominators that appear in scenarios and has the ultimate goal of discovering service features. Prototypes are built to establish experimentation settings for scenarios and services, thus they act as tools and means and are not at all the sole and main end result of the development. Use cases offer a tool for expressing the content of the scenarios from viewpoint of the software and hardware system. Furthermore, use cases may be used to express new or expected technological possibilities as inspiration for scenarios and services.

The right hand side of Figure 1 shows how the ultimate goal of requirement engineering is the realization of the right product. Scenarios and use cases are used for validating the user needs to ensure that the realized product matches the user needs. Requirement engineering is crucial for ensuring that the product is what users really need and not what they might think their need or, what too often seems to be the case, what software engineers think users need.

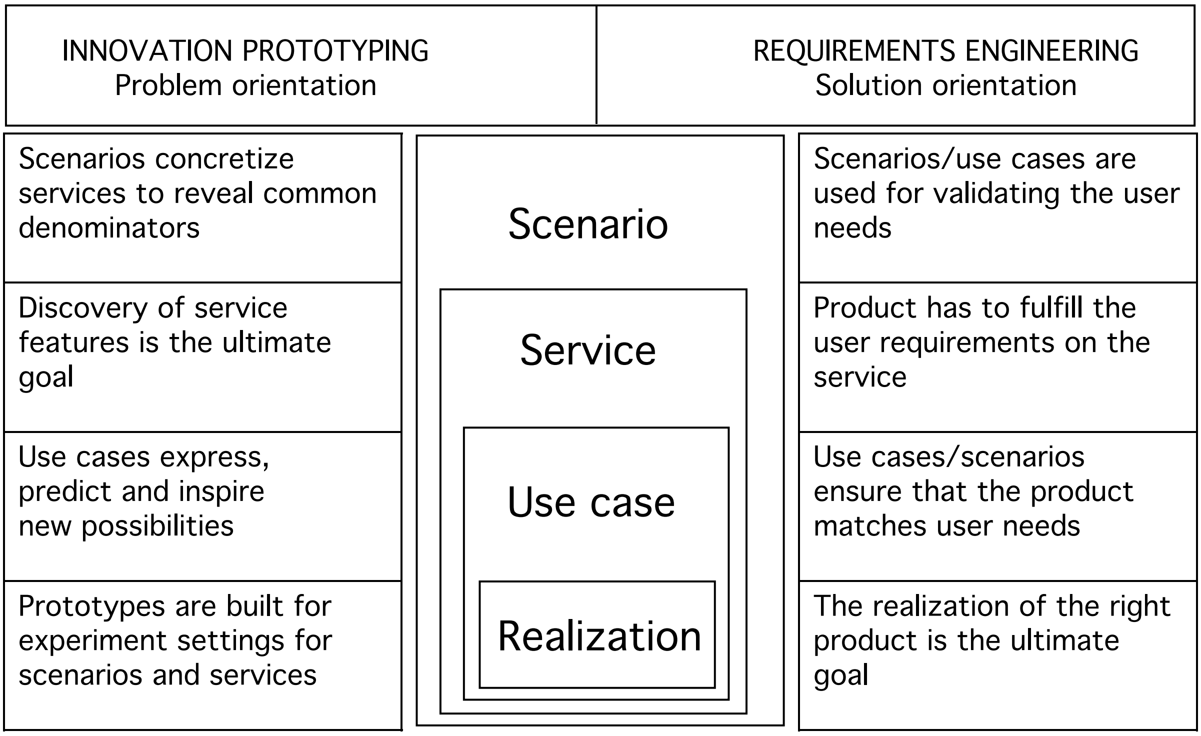

INNOVATION PROTOTYPING
Problem orientation
REQUIREMENTS ENGINEERING
Solution orientation
Scenarios concretize services to reveal common denominators
Discovery of service features is the ultimate goal
Use cases express, predict and inspire new possibilities
Prototypes are built for experiment settings for scenarios and services
Scenario
Service
Use case
Realization
Scenarios/use cases are used for validating the user needs
Product has to fulfill the user requirements on the service
Use cases/scenarios ensure that the product matches user needs
The realization of the right product is the ultimate goal

Figure 1. Problem orientation vs. solution orientation

Thus, innovation prototyping and requirements engineering approach the task from opposite directions and are actually carrying out a different task. On the one hand requirement engineering takes the challenge to describe the required solution completely beforehand and validate it with the intended users in order to ensure focusing on right requirements of the right product. On the other hand innovation prototyping strives to describe what kind of solutions with what kind of features are needed, i.e., what is the problem to be solved. The constructive work with experimentation prototypes is a means for discovering the common denominators and thus it is a way to understand the problem [8]. Thus, the requirement engineering aims to capture a particular need and innovation prototyping aims to explore the conceivable future needs.

Experimentation setting vs. validation

Experimentation settings are crucial part of the innovation prototyping. Experimentation setting refers to facilitating the necessary environment for formal testing methods. In innovation prototyping the various viewpoints each use their own experimentation methods and require different gadgets, tools, recording facilities etc. to ensure feasible and reliable tests [9]. Examples of the experimentations include usability tests and user studies, software performance tests, various network tests and comparisons of different technologies. The following are concrete examples of typical experimentation settings in innovation prototyping:

- Usability tests and user studies require zero prototypes, justified group of target users as well as logging and recording facilities. [8, 11, 12]

- Context-based user interface adaptation was studied based on the fuzzy context information sensed via on-board sensors of lightning, movement, volume etc. The performance of the application adaptation was examined by collecting context data during real usage. [10]
- The mobile music player Ämppäri was originally realized as our first mobile prototype to experiment the features of the software and hardware systems. Currently it is being reused in a new project as a test application for on the one hand demonstrating the operation of system allowing vertical handover between different networks (e.g. WLAN, GPRS) and on the other hand for carrying out usability tests. [1]

The left hand side of Figure 2 summarizes the characteristics of experimentation in innovation prototyping. The objective is to establish experimentation settings that ensure controlled conditions for formal methods. The focus of the experimentation determines the requirements for the prototype that is built. The obtained output is compared with the assumptions and an analysis is carried out for reliable interpretation of the results. Before any actual experimentation takes place, the experimentation itself has been classified and it's reliability and performance is defined. The objective is to obtain reusable results and ultimately to discover universally applicable principles, features and laws.

	Experimentation of service ideas in innovation prototyping	Product requirements validation in requirements engineering
Task	Experimentation of features of future services and environments	Validating that the product is well founded or justifiable
Objective	Experimentation settings under controlled conditions to discover laws or to test hypothesis	Validation of a sound authoritative basis and confirming the acceptance of the user
Means	Innovation prototypes are built to fit the reliability criteria of the chosen experimentation method	Requirements elicitation and analysis strives on user needs. Formal documentation is a basis for validation
Reliability	Test plan with assumptions, expectations and focused scope is a basis for reliable analysis of the results	Proper documentation covers all necessary requirements and leaves no chance for misinterpretation

Figure 2. Characteristics of experimentation and validation

The right hand side of Figure 2 gives rough characteristics of validation in requirements engineering. Validation, as it is know in the requirement engineering, is determining the value of individual requirements to the stakeholders. The definition of a stakeholder here is any person that is a user to the target system or a person that is directly involved in introduction of such system not including the engineers controlling the requirements engineering process, unless the engineers are also users. Validation ensures a sound basis for the development by confirming the acceptance of the user. User needs are collected during requirements elicitation and formal documentation provides basis for their validation.

The requirements and their validation are bound to the actual environment, the actual users, their real tasks and even the actual product as far as possible of the product. If a set of requirements is not based on a study of an actual environment, but on speculation, the resulting requirements can turn out to be and probably will be to generic.

The necessity of concrete settings appears to be common for innovation prototyping and requirements engineering. However, the objective and usage of the settings differs. In requirements engineering the specified user, environment and product are used to validate justification and acceptance of the specific users in the specific environment. In innovation prototyping the concrete prototype and usage environment are not all embracing, but focused and carefully limited to obtain specific results that apply to all cases that share the focused setting. For example the above-mentioned mobile usage, handling of fuzzy context and music player were aimed for discovering common features of mobile users, context interpretation and mobile applications.

Innovation prototyping is inherently experimentation oriented. The objective is to construct something concrete enough to allow formal testing. Requirement engineering is validation oriented. The objective is to validate whether the requirements and interpretations of user observations and interviews match with the stakeholders' context.

Service idea generation and product development methods

Innovation prototyping proceeds through iterations of planning test settings, realizing needed prototypes and carrying out the experimentation. Examples of such constructive rounds are

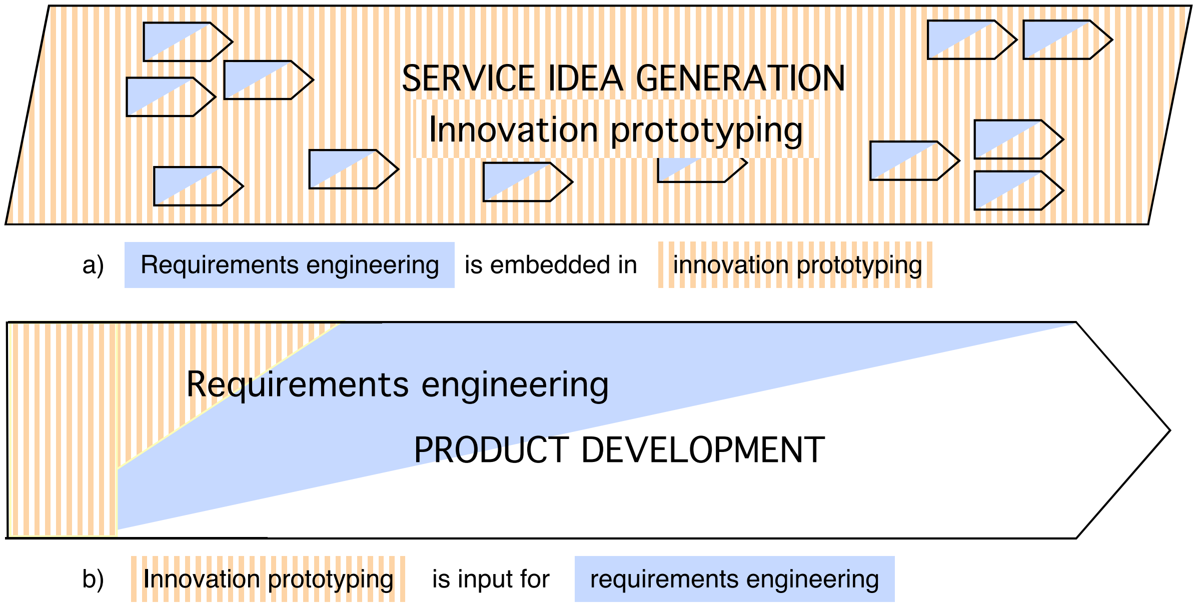
- The Ämppäri prototype that allows mobile users to listen music while being on the move. An embedded prototype was built to realize a service allowing the core features of providing a choice of huge number of music, allowing wide area of usage, facilitating test scenarios of sponsored regular quality music samples as well as high quality entertainment etc. The research is on going, since both the music playing software and the captured design decisions and construction practices turned out to be very reusable. [1]
- The usability and usage of mobile communication tool was studied by using a zero prototype that consisted of a PDA and a GSM phone for realizing mobile Internet usage. Seven students of a university of technology were chosen as a user group. Interviews, web diaries and essay writing were used to gather information. Additionally a proxy was applied for logging the visits to web pages. The research is on going. [2, 8, 11, 12]
- A little less typical innovation prototype is a software system for structuring user data. The system allows usability experts to interactively structure and build models of the highly non-structured data that has been collected from the users through interviews, diaries, role games etc. The system is not a typical innovation prototype, since its intended users are user study experts and product developers and not common end-users as in most of our cases. However, the system is being implemented according the innovation prototyping methods, since it is a tool for learning what kind of a tool would ultimately be needed, i.e., it could not be implemented as a regular software system, since nobody really knew what kind of models and functions it should provide and even for what tasks it was going to be used for. The work is on going. [6].

Obviously in each of the given three examples of innovation prototyping, it is sensible and important to take advantage of the requirements engineering methods in order to ensure proper quality of the implementation. Thus, requirements engineering is embedded in the service idea generation that we carry out with the innovation prototyping methodology. Figure 3a is an attempt to illustrate this relation. The service idea generation box is carried out by innovation prototyping methods, which is shown with the striped pattern. The small arrow shapes illustrate the individual innovation prototyping rounds. Notice that each arrow shape has at the top left an even colored area referring to the embedded requirements engineering.

The requirements engineering offers a fine set of principles and tools that are ready to be applied. However, it is necessary to emphasize two facts that have to be carefully remembered while carrying out requirements engineering for innovation prototypes

- The construction of an innovation prototype for service idea generation can be regarded as product development. However, it is the extremely important to remember that the word product refers now to the research system, i.e., innovation prototype, and not to the final end product or service.
- In requirements engineering it is extremely important to remember that the user whose needs are to be fulfilled is now the researcher and not the ultimate end user.

These two facts mean that the requirements engineering must be focused on gathering the requirements for the experimentation setting and not of the ultimate usage situation. Thus the importance of the requirements depends on what is essential to ensure feasibility and reliability of the test method. If these extremely important facts are forgotten the innovation prototype loses its focus and the experimentation setting loses its power of reuse.



a) Requirements engineering is embedded in innovation prototyping

b) Innovation prototyping is input for requirements engineering

Figure 3. Service idea generation and product development methods

Requirement engineering emphasizes ability to trace requirements and producing documentations according to formal guidelines that ensure completeness and comprehension. Furthermore, requirements engineering benefits from getting access to all possible background material on usage environments, users, etc. Since innovation prototyping supports and provides methods fulfilling these needs, it can be can be regarded as input to requirements engineering. Figure 3b is an attempt to illustrate this. The striped pattern on the left refers to innovation prototyping that can be considered as pre-product development. The even color on the top shows the requirements engineering that continues throughout the product development process, however, it may require less resources during the later stages. Notice that the sizes of the striped and even color areas carry no meaning, as the picture is merely symbolic.

Innovation prototyping provides two types of input to requirements engineering

- The built innovation prototypes, such as the Ämppäri, the zero prototype on mobile Internet usage and the user data structuring tool, maybe used as such in the requirement engineering. Product development may be triggered by any innovation prototypes of any level, i.e., a scenario, service idea or realized prototype may be proven to be worth taking into the product development process. Notice that not only the prototypes as such, but more over their rationale and experimented features give requirements engineering a flying start.
- The usability, prototype building, experimentation and information management methods of innovation prototyping that have been developed for service idea generation mostly fit the goals of requirements engineering. Thus, they can be utilized in requirements engineering as individual methods even if innovation prototyping is not carried out as wholeness. Notice, that innovation prototyping is currently developed for the particular field of mobile and ubicomp services, so they are suitable for requirements engineering of software for this field.

Service idea generation as a playground for product development

The three examples of innovation prototypes were not were not an end in itself but each one was implemented with ultimate goals.

- The Ämppäri prototype has through its versions served as a tool for capturing and cumulating understanding on the criteria of mobility. At first, mobility simply appeared to mean something light that can be carried around. However, soon we recognized a bunch of other criteria, and moreover the fact that each expert understands mobility from own viewpoint. [1, 9]
- The usability research of mobile Internet users had from the very beginning other goals that merely studying the combined usage of a certain PDA device and a mobile phone. The big question of the on-going research is to understand and model the features of a mobile user. The initial zero prototype allowed setting a limited scope for the experimentation and collect data that was assumed to indicate differences of stationary and mobile usage of services. Thus, the zero prototype allowed discovery not just general observations on mobile users, but discovery of tentative principles that seem to apply under certain conditions. [8,11,12]
- The user data-structuring tool was constructed to make the work of the usability experts easier and to possibly discover new practices for them. The collected data is structured and modelled, so that following usability and user study rounds can utilize it and may avoid some time consuming stages. Thus, the aim is to be able to reuse and cumulate the user data for future user studies, in order to fully exploit the results of the laborious usability and user study materials. Furthermore, the user data-structuring tool had the ultimate goal of allowing the mobile service idea developers to play with the different views and links and experiment the usefulness of different models and structuring approaches. For example there is a need to find better ways to model mobility and this may require some changes to the usability test and user data methods in order to be able to capture more relevant data. [6]

The three examples show how each innovation prototyping round benefits the general understanding of future mobile and ubicomp services. Partly this happened due to the constant improvement of the methods. However, more importantly each innovation prototyping round looks for common denominators. The developers are able to follow the results and on-going work of others, which allows recognizing potential common denominators, such as various concepts, criteria and modeling possibilities of mobility in the examples.

As mentioned before, innovation prototyping looks for common denominators in sets of needs trying to filter out future service models. Requirement engineering as a tool of product development is more oriented on product and has its goals in trying to filter out needs that originate from a certain and specific environment. Innovation prototyping might not begin its journey from any identifiable environment, but from a set of environments that are classified and named through controlled process of testing out bounded prototypes. These prototypes then have the opportunity to create value to an environment and more importantly enable understanding of such value to the person administrating the test. Innovation prototyping does not change the potential of an environment, but strives to see and understand changes that might occur in it and thereby discover its value in a new light.

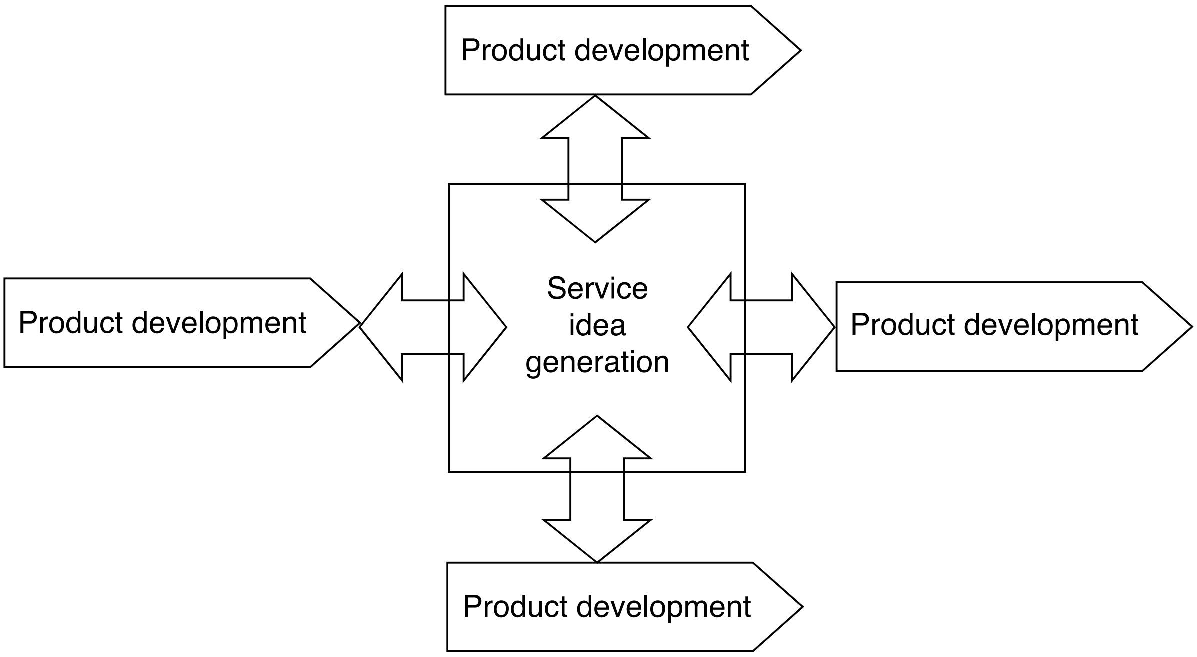


Figure 4. Service idea generation allows a playground for product development

Figure 4 illustrates how service idea generation allows a playground for product development, which includes the requirements engineering. It has many kinds of relationships to the product development projects. Product development may at any time pick from the playground's pool some scenarios, service ideas, use cases or prototypes that are found mature enough or critical enough to require urgent productification. A product development process may be triggered by the discoveries of the pool. Likewise product development may provide input for service idea generation and service idea generation may adopt results from product development.

Conclusions

The paper explored the relationship of innovation prototyping and requirements engineering and proposes that they can take advantage of each other's methods. On the one hand requirements engineering is embedded in the individual innovation prototyping rounds. On the other hand innovation prototyping is input for requirements engineering in the form of prototypes including their rationale and in the form of applicable methods specialized on mobile and ubicomp field.

In spite of all the common objectives and ability to share methods the objectives of the two differ clearly. In requirements engineering the scenarios and use cases have to match an exact user, usage environment, task and even product. The point lies in validating that the product is well founded and confirmed by the acceptance of the user. In innovation prototyping a scenario or service does not initially have to be correct. The main thing is that it provides feasible material or frame of reference for experimentation.

The constructive work of building innovation prototypes that facilitate experimentation is a core of innovation prototyping. However, the prototypes are not an end in itself. Rather the purpose is idea generation and its preconditions are sound experiments and brokering of different viewpoints. Each innovation prototype and experimentation round is kept strictly focused and convergent to avoid vagueness and to ensure feasible and reliable conditions for discovering common denominators. In contrast requirements engineering is product oriented with the ultimate goal of realization of the right product.

The service idea generation, for which innovation prototyping offers methods, has its own objectives and ways of operation. Ideally it is a continuous process and not just a preceding stage of product development. The service idea generation continuously carries out several innovation prototyping rounds that proceed in their own pace maintaining and cumulating a pool of scenarios, service ideas, use cases and realized prototypes of various levels. Product development may be a direct spin-off that commercializes the results of innovation prototyping, it may utilize bits and pieces from the pool, or it may build on the common denominators or provide input to trigger innovation prototyping of potential areas.

It is crucial to facilitate the ability to discover and transfer principles and features across different rounds and innovation prototypes. This way a collection of common denominators is constantly cumulated and it becomes possible to reuse not just prototypes but also their rationales and features.

In the same way it is crucial facilitate the brokering-based balancing of viewpoints [9] to be able to discover the denominators and relations of developer viewpoints such as user study expertise, software engineering, hardware engineering, network expertise and various control factors such as privacy and security aspects.

The research on innovation prototyping is continuing as constructive innovation prototyping and as development of the methods and software tools to support the methods.

References

1. Asplund, H. and Ranta, M. 2003. How use cases are defined from scenarios in development of mobile services. In proceedings of COST269 Conference, Helsinki, Finland, 3-5 September 2003. COST.

2. Gaver, B., Dunne, T., Pacenti, E.: *Design: Cultural probes*; *interactions* 6, 1 (Jan. 1999), ss. 21 – 29

3. IEEE Std 830-1998 (1998). IEEE Recommended Practice for Software Requirements Specifications. The Institute of Electrical and Electronics Engineering, Inc., New York.

4. Kujala, S., Kauppinen, M., and Rekola, S. (2001). Bridging the Gap between User Needs and User Requirements. In Avouris, N. and Fakotakis, N. (Eds.) Advances in Human-Computer Interaction I (Proceedings of the Panhellenic Conference with International

Participation in Human-Computer Interaction PC-HCI 2001), Typorama Publications, pp. 45-50.

5. Kujala, S. (2002). User Studies: A Practical Approach to User Involvement for Gathering User Needs and Requirements. Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 116. Espoo: the Finnish Academies of Technology. ISBN 951-666-599-3.
6. Lagutin, D. and Ranta, M. 2003. Challenges of structuring data from usability and user studies. In proceedings of COST269 Conference, Helsinki, Finland, 3-5 September 2003. COST.
7. Nikula, U. (2002). BaRE - A Ready to Use Method for Requirements Engineering, in Department of Information Technology. Licentiate Thesis, 2002, Lappeenranta University of Technology: Lappeenranta.
8. Parkkinen, J. and Ranta, M. 2003. A zero prototype as an usability tool. In proceedings of COST269 Conference, Helsinki, Finland, 3-5 September 2003. COST.
9. Ranta, M. and Asplund, H. 2003. Brokering based balancing of viewpoints – experiences from pre-product development of mobile services. In proceedings of COST269 Conference, Helsinki, Finland, 3-5 September 2003. COST.
10. Himberg, J. and Mäntyjärvi, J., 2002. Context-based user interface adaptation. GO for better life seminar, T-106.850 Software Technology, Helsinki university of technology. May 2002.
11. Karvonen Kristiina, Parkkinen Jarmo and Poropudas, Liina: *"Investigating Mobile Use for Wearable Product Concept Design"*, in Conference Proceedings, June 22-27, 2003, Crete, Greece
12. Parkkinen, J. 2003. Mobiiliviestimen käytettävyyden arviointimenetelmiä. M.Sc. Thesis, Helsinki University of Technology, to be published 2003.