

Please quote as: Böhm, F.; Fähring, J.; Huber, M.; Leimeister, J. M. & Krcmar, H. (2011): Scenario planning for innovation development: an overview of different innovation domains. In: R&D Management Conference, Norrköping, Sweden.

Scenario planning for innovation development: an overview of different innovation domains

Felix Böhm

Technische Universität München
boehmf@in.tum.de

Jens Fähling

Technische Universität München
faehling@in.tum.de

Michael J. Huber

Technische Universität München
hubermic@in.tum.de

Jan Marco Leimeister

Universität Kassel
leimeister@uni-kassel.de

Helmut Krcmar

Technische Universität München
krcmar@in.tum.de

Abstract

Innovations expressed by customers are often unstructured, unclear, and ambiguous, making it difficult to interpret customers' contributions, integrate them into a company's R&D department and transfer them into comprehensible requirements for the development of product, service or software innovations. In order to close this gap, the paper presents scenarios as a technique to support the communication between customers and developers along the R&D process and to reduce feedback cycles. A basic understanding of scenarios and their application in innovation management is delivered by outlining scenario definitions, identifying dimensions for the classification of scenarios, and providing an overview of scenario representation techniques for different innovation domains.

Keywords: scenario planning, innovation management, customer integration

1 Introduction

Methods for customer integration such as idea competitions, lead user workshops and internet-based innovation communities are successful ways for companies to understand their customers' needs as well as their ideas for new and innovative products (Bretschneider et al., 2009). Most methods for customer integration, especially during the early stages of the inno-

vation process, allow for a creative but also unstructured, unclear, and ambiguous elaboration of ideas (Reichwald & Piller, 2006). This, however, makes it difficult to understand and interpret customers' ideas and to transfer them into comprehensible requirements for the development of concrete product, service or software innovations (Ulwick, 2005).

The development of a clear understanding of customers' needs is often challenging and expensive. According to Reichwald & Piller (2006), two kinds of information are required for innovating and developing products: On the one hand, customers and users express need information in terms of requirements, needs, and wishes; on the other hand, solution information which naturally resides at the manufacturer describes ideas, possible solutions, and their production process. It is quite challenging to clearly and completely adapt need information expressed by customers as solution information to a company's research and development department (von Hippel & Katz, 2002; Thomke & Hippel, 2002). This is largely due to the fact that customers are often unable to describe their requirements and needs to the suppliers in an accurate and complete manner. Either they do know what they want but cannot convey it in detail, or they do not know the "right" solution until conducting trial-and-error analyses (von Hippel, 1998; von Hippel, 2005). As a consequence, customers can hardly be expected to deliver readily utilizable solutions.

Apart from the lack of required knowledge and expertise to fulfill tasks within innovation processes, customers are not familiar with the kind and structure of information needed by companies for innovating new valuable products. Solutions, design specifications, needs, and benefits are potential types of information expressed by customers. However, all of these input types hardly help companies with implementing innovations when the information does not fulfill certain criteria. Customers' expressions of needs as well as benefit statements must be concise, actionable, unambiguous and measurable, in order to avoid confusion about the need or desired benefit that was captured (Ulwick, 2002; Ulwick, 2005).

Another challenge in the field of a business- and customer-overarching innovation process is the right balance between creativity and structure. Many of the above mentioned problems of transferring customer ideas to comprehensible requirements are derived from the high degree of design freedom in conventional ideation processes. Scenarios are a generally accepted and appropriate technique to overcome this dilemma of openness versus compliance (Carroll, 2000; Szulanski & Amin, 2001; Drew, 2006; Bergmann et al., 2009).

As seen at the information translation problem between customer and technical expert, their shared innovation processes in open innovation require structured outputs. In order to establish this interaction competence and minimize the communication problems related to customer-developer interaction, scenarios are an appropriate and helpful technique. They support developing and utilizing customer innovations in a goal-oriented and structured way by enabling the innovators to 'share and reassemble personal knowledge to create a common understanding between the internal and external environment in an organization' (Bergmann et al., 2009). Scenario planning is an established approach for the communication between customers and developers, and allows for describing alternative options of solutions. Hence, this approach is recommended for settings where a decision could result in a large spectrum of possible future outcomes (Drew, 2006). Practical examples state that 'experience has taught [...] that the scenario technique is much more conducive to forcing people to think about the future than the forecasting techniques [...] formerly used' (Benard, 1980). Generating ideas for innovations as well as testing their suitability on daily use, is supported by the application of scenarios. This way innovation processes can be managed in each phase, and acceptance of an innovation can be checked previously (Wilms, 2006).

The aim of this paper is to provide an overview of scenarios as a technique for the communication between customers and developers along the R&D process. In a first step, the paper introduces different definitions of scenarios. Second, different types of scenarios are summarized and distinguished by dimensions. In a third step, different ways of scenario representations are described and distinguished by design parameters. The paper also illustrates some examples of scenarios from software, product and service engineering. Finally, an overview of data elements that are generated by the scenario techniques is presented and supports the selection of an appropriate scenario technique for specific purposes.

2 Foundations of Scenarios

The diversity of scenario characteristics and the variety of domains where scenarios can be applied have led to a rather ambiguous opinion in the interpretation and the globally accepted definition of scenarios (Filippidou, 1998). This section gives an overview of the defining properties and characteristics of scenarios (see Table 1). However, some core properties can be unified among the majority of the definitions.

Gausemeier et al. (1996) claim that scenarios are based more on projections and forecasts rather than on prognoses. The complexity arises from the underlying development potentialities of multiple cross-linked influencing variables. Scenarios should describe prospective situations so that product planners can visualize this situation in a future market, for which they are developing products now. The description of the development makes scenarios more credible and they can be understood easier. A scenario's description of possible futures also reflects diverse viewpoints of the past, the present, and the future (van Notten et al., 2003; Gausemeier et al., 1996). Scenarios are instruments for describing the design vocabulary and exchanging ideas and thoughts effectively, focusing on episodic cases. From a user's point of view, they can be utilized for describing requirements and designs of innovative artifacts and their intentional way of use, i.e., the functionalities and organizational processes are supported and available (Filippidou, 1998).

Definition of a scenario	Source
Description of a complex future situation, whose incidence cannot be predicted definitely and the representation of a development that could lead from the present to this situation.	Gausemeier et al. (1996)
A scenario can be defined as a description of a possible set of events that might reasonably take place. The main purpose of developing scenarios is to stimulate thinking about possible occurrences, assumptions relating these occurrences, possible opportunities and risks, and courses of action.	Jarke et al. (1998)
Scenarios are descriptions of possible futures that reflect different perspectives on the past, the present and the future.	van Notten & Rotmans (2001)
Scenarios are a disciplined method for imagining possible futures that companies have applied to a great range of issues. Each scenario tells a story of how various elements might interact under certain conditions. They explore the joint impact of various uncertainties, which stand side by side as equals.	Schoemaker (1995)
Scenarios produce forecasts of future business environments and identify conditions leading to major changes in these environments.	Huss & Honton (1987)

Table 1: Definitions of scenarios in innovation processes (Gausemeier et al. 1996; Jarke et al., 1998; van Notten & Rotmans, 2001; Schoemaker, 1995; Huss & Honton, 1987)

Each definition contains the aspect of scenarios as a description of a possible future. However, different authors emphasize different additional aspects. In order to provide a useful definition of scenarios as a technique for the communication between customers and developers, we propose the following definition:

‘A scenario tells the story of a possible future or of a possible set of events that might reasonably take place.’

This definition emphasizes that scenarios can be used as a technique for the communication between customers and developers by telling stories of a possible futures or possible set of events in the future. In addition to raw requirements, stories also contain useful information about how possible specific demands can be fulfilled. Communication is improved through a common understanding of future situations, enabled by the use of scenarios. Especially in the field of innovation, a clear and understandable expression of possible futures or prospective set of events it is crucial for a purposeful and prospective development of an idea.

The aspect of ways and developments, i.e. the process leading toward a future situation is omitted in this paper. A reasoning of the derivation is not primarily needed in innovation management, but the application of scenarios as a communication technique between customer and developer. This is why the main focus is on telling stories.

2.1 Scenario classification

Before initiating a scenario planning process, a set of questions should be answered by the scenario creators in order to give an orientation context (see Table 2). This orientation context gives an insight into various forms and key aspects of the development and utilization of scenarios. Different scenario intentions can be compared, and single scenarios can be classified in terms of their character and field of use. The classification also creates a common basis where all participants assume the same premises (Gausemeier et al., 1996; van Notten et al., 2003).

Two authors dealing with scenarios and their classification were considered for the creation of this paper’s list of defining dimensions. Gausemeier et al. (1996) and van Notten et al. (2003) present numerous dimensions according to which scenarios can be classified and described. Table 2 shows the selected dimensions for classifying the scenarios.

Dimension	Defining question	Characteristic		Source
Problem statement	Should concrete decision problems be solved by applying the scenarios?	Yes	Decision scenarios	Gausemeier et al. (1996)
		No	Orientation scenarios	
Controllability	Apart from external conditions, are influenceable factors integrated into the scenario creation?	Yes	Systems scenarios	
		No	External scenarios	
Form of organization	Are the scenarios used by persons/groups that have also created them?	Yes	Internal projects	
		No	External projects	
Temporal condition	In addition to future situations, are the ways from the present to this future depicted as well?	Yes	Process-related scen.	Gausemeier et al. (1996); van Notten et al. (2003)
		No	Situational scenarios	
Point of origin	Reason from a specific future situation? Explore paths that need to be taken to desirable future situations.	Anticipative scenarios (Back-casting scenarios)		
	The present as starting point? Multiple future situations to explore.	Explorative scenarios (Forecasting scenarios)		

Goal-orientation	Describe probable or preferable futures? Prospective, strategy, or intervention scenarios, depending on one's interpretation	Normative scenarios		
	Explore possible futures? Baseline, reference, non-intervention scenarios?	Descriptive scenarios		
Probabilities of occurrence	Are probabilities of occurrence assigned to the pictures of the future?	Yes	Prediction	Gausemeier et al. (1996)
		No	Projection	
Nature of dynamics	Are extreme pictures of the future developed?	Yes	Extreme scenarios	Gausemeier et al. (1996); van Notten et al. (2003)
		No	Trend scenarios	
Time scale	What planning interval is called for the pictures of the future?	< 2y.	Short-range scenarios	Gausemeier et al. (1996)
		> 5y.	Long-range scenarios	

Table 2: Orientation context for scenario classification
(own table based on Gausemeier et al. 1996; van Notten et al., 2003)

This collection of 18 different types of scenarios in 9 dimensions is supposed to give the scenario creator an idea of *what* he wants to develop and represent. It defines the 'content' aspect of scenarios. Each scenario type has specific characteristics and is used under certain preconditions and for certain purposes. According to the identified scenario type and its characteristics, configurations on the scenario planning process as well as on the scenario representation and technique (the 'form' aspect) can be made in order to adjust them to the individual purpose.

2.2 Scenario representation

After classifying the intended scenario, this chapter describes different forms for the representation of scenarios. As a preparation for the specific configuration of a particular scenario technique, this section provides a list of possible design elements of scenarios.

Filippidou (1998) and Rolland et al. (1998) present numerous design parameters and defining features that have been evaluated in terms of their suitability for the subsequent configuration of the scenario technique. Based on these encountered attributes, scenario representations can be classified and described. Table 3 shows the selected dimensions for classifying the scenarios. The authors define the solution space for developing scenarios and provide a basis for coordinating the objectives and goals of all participants.

Design parameter	Scope	Characteristic
Level of formality	Modeling language (UML), tables, scenario scripts	Formal scenarios
	(Prosaic) Usage/Stories descriptions	Informal scenarios
Concreteness/ Abstraction level	General ideas, rough narration of stories	'Vague'
	Coherent solution, carefully grounded in the details, evaluating/validating target system	'Detailed'
Open-endedness	'Big' picture of situations, greater context where requirements need to function	'Scenarios-in-the-large'
	Describing specific needs, concerns or events established within 'larger' scenarios	'Scenarios-in-the-small'

Table 3: Design parameters for scenario representation (Filippidou, 1998; Rolland et al., 1998)

3 Examples of techniques for designing innovation scenarios

This section presents a selection of scenario techniques from the three innovation domains: *software engineering*, *product engineering*, and *service engineering*. The focus of this review is on the techniques' outcome and representation, which includes textual descriptions, graphical illustrations, as well as diagrams, and thus covers a broad range of forms of expression. Each technique will be briefly described with the aid of an example. Furthermore, all data elements that are generated with the techniques are summarized in order to provide an overview of the results that can be expected from each technique. These rather basic but established, tool-detached approaches are described in place of a wide variety of scenario techniques of the innovation domains.

3.1 *Software engineering*

Scenarios in the area of software development can have various manifestations, depending on the respective forms of scenario generation and its usage. They range from detailed descriptions of usage contexts or small scale examples in the Human Computer Interaction (HCI) community to use cases in Software Engineering to scenario scripts as test data in Requirements Engineering (Rolland et al., 1998). Scenario approaches have gained increased interest in requirements engineering research and practice (Weidenhaupt et al., 1998). Through the use of scenarios, requirements elicitation is enhanced by providing a technique that is understandable to users and clients. These techniques, e.g., mock-ups, often help to reveal drawbacks in the specification and to draw a more concrete and precise picture of the system. The emphasis for developers during actor identification and scenario identification is to understand the application domain. This results in a shared understanding of the scope of the system and of users' work processes to be supported. Once developers have identified and described actors and scenarios, they formalize scenarios into use cases (Bruegge & Dutoit, 2009).

Use case approach

The use case approach has initially been described by Jacobson (1995) in the field of object-oriented software engineering (OOSE) (Jarke et al., 1998; Bustard & Wilkie, 2000; Carroll, 2000; Bodker, 2000). It is used for modeling the behavior of a system, a subsystem or a class and to explain the interaction of a system with its environment (Bruegge & Dutoit, 2009; Jacobson & Bylund, 2000; Booch et al., 2006). Use cases describe functions of a system from an actor's point of view, whereas they can involve more than one actor. Furthermore, they are specified by a set of events which generate visible results for the actors. All use cases in combination represent the functionality of an entire system. Use cases can be identified in different ways, e.g. by collecting user needs and wishes or through analyzing textual presentations of problems. They also comprise descriptions about how actors use a system and how a system satisfies the actor's needs (Bruegge & Dutoit, 2009; Hitz et al., 2005; Bittner & Spence, 2003). A use case is per se a generalization of a scenario and usually comprises a whole set of interrelated scenarios. Accordingly, we speak of scenarios as instances of use cases (Larman, 2002). Use cases consist of the following elements:

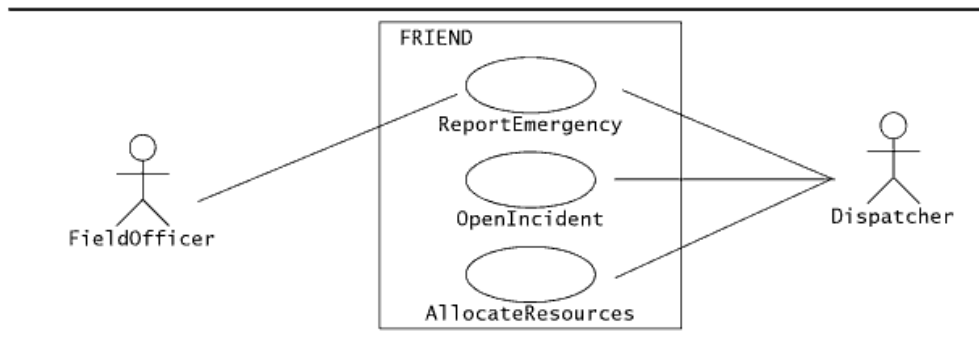


Figure 1: Example for a use case diagram: accident management system (Bruegge and Dutoit, 2009)

Actors are objects located outside the system boundary and interacting with the system by calling one or more functionalities in terms of an actual use case. Actors are identified by a unique name and can be either human users (e.g., system administrator or a bank customer) or also other systems (e.g., a central database or a fabrication line). Actors are visually represented by stick-figures labeled with their identifier (Bruegge & Dutoit 2009; Hitz et al., 2005; Bittner & Spence, 2003).

The *system boundary* separates use cases inside the system from actors outside of the system and differentiates between the tasks accomplished by the system and the tasks accomplished by its environment. The data elements within the system boundary are responsible for the execution of the behavior demanded by the data elements outside the boundary.

Communication relationships (between actors and use cases) express the information exchange between actors and use cases and can be applied to denote an actor's access to functionality.

Relationships (between use cases) support the reuse of existing functionality as well as the stepwise specification of new functions. This way redundancy is avoided and system can be depicted in a layered manner.

Descriptions of the use case provide the full specification of what happens in the use case. Every use case in the use case model relates to a document describing how the various data elements and entities collaborate to fulfill the goal represented by the use case. The textual description of a use case is written in natural language and is composed of six fields: name, participating actors, flow of events, entry condition, exit condition, and quality requirements (Bruegge & Dutoit, 2009).

Use cases, actors, system boundary, and relationships are all graphical elements, whereas the use case description is a textual element. Examples for the graphical use case diagram and for the textual representation of a use case can be seen in Figure 1 and in Figure 2, respectively.

<i>Use case name</i>	ReportEmergency
<i>Participating actors</i>	Initiated by FieldOfficer Communicates with Dispatcher
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. The FieldOfficer activates the "Report Emergency" function of her terminal. 2. FRIEND responds by presenting a form to the FieldOfficer. 3. The FieldOfficer fills out the form by selecting the emergency level, type, location, and brief description of the situation. The FieldOfficer also describes possible responses to the emergency situation. Once the form is completed, the FieldOfficer submits the form. 4. FRIEND receives the form and notifies the Dispatcher. 5. The Dispatcher reviews the submitted information and creates an Incident in the database by invoking the OpenIncident use case. The Dispatcher selects a response and acknowledges the report. 6. FRIEND displays the acknowledgment and the selected response to the FieldOfficer.
<i>Entry condition</i>	<ul style="list-style-type: none"> • The FieldOfficer is logged into FRIEND.
<i>Exit condition</i>	<ul style="list-style-type: none"> • The FieldOfficer has received an acknowledgment and the selected response from the Dispatcher, OR • The FieldOfficer has received an explanation indicating why the transaction could not be processed.
<i>Quality requirements</i>	<ul style="list-style-type: none"> • The FieldOfficer's report is acknowledged within 30 seconds. • The selected response arrives no later than 30 seconds after it is sent by the Dispatcher.

Figure 2: Example for a use case description: report emergency (Bruegge & Dutoit, 2009)

Scenario approach by Bruegge/Dutoit

The scenario approach is an easy and understandable method of ascertaining requirements for improving the interaction between developers and users. Bruegge & Dutoit (2009) define a scenario in the domain of OOSE as "a concrete, focused, informal description of a single feature of the system from the viewpoint of a single actor." It represents an instance of a use case and provides a description of concrete events and sets of actions, meaning a single scenario does not describe all potential situations of a certain event and contains no decision-related steps. Common cases are depicted with the focus on usability of the system.

The use of scenarios during software engineering can be roughly distinguished into four types. *As-is scenarios* are applied for the description of current situations, whereas *visionary scenarios* describe a future system. *Evaluation scenarios* support the evaluation of a system in terms of their contribution for fulfilling user tasks. *Training scenarios* are used for the introduction of new users to a system. According to their purpose these scenarios types can be applied for the description of specific situations and events.

Like the description of use cases, the textual representation of a scenario contains a few obligatory fields: The unambiguous *scenario name*, the *participating actor instances*, and the step-by-step descriptive *flow of events*.

Some questions can help designers, developers, users, and other stakeholders of a system to identify scenarios: What tasks should be performed by the system? What is the content and origin of the data accessed by the actor? Is it modifiable or removable? What kind, frequency, and time of external changes does the actor need to inform the system about? Vice versa, about what events does the system need to inform the user?

In Figure 3 an example of a scenario description is illustrated. The scenario is based on a use case called "ReportEmergency" and describes a flow of events after a warehouse is on fire.

<i>Scenario name</i>	<u>warehouseOnFire</u>
<i>Participating actor instances</i>	<u>bob, alice:FieldOfficer</u> <u>john:Dispatcher</u>
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. Bob, driving down main street in his patrol car, notices smoke coming out of a warehouse. His partner, Alice, activates the “Report Emergency” function from her FRIEND laptop. 2. Alice enters the address of the building, a brief description of its location (i.e., northwest corner), and an emergency level. In addition to a fire unit, she requests several paramedic units on the scene given that area appears to be relatively busy. She confirms her input and waits for an acknowledgment. 3. John, the Dispatcher, is alerted to the emergency by a beep of his workstation. He reviews the information submitted by Alice and acknowledges the report. He allocates a fire unit and two paramedic units to the Incident site and sends their estimated arrival time (ETA) to Alice. 4. Alice receives the acknowledgment and the ETA.

Figure 3: The warehouseOnFire scenario for the ReportEmergency use case (Bruegge & Dutoit, 2009)

Wireframes and Mockups

Wireframing is a method used during specification where objects are placed strategically on a framework for a web page or software application frontend. A wireframe represents a skeletal plane that consists of simple page representations, showing the structure and navigation of the site or the website or application as well as the location of content. It also expresses the functionality in terms of the relationship with other objects, sets integrity constraints, and includes data type definitions. The wireframe depicts the object’s expected behavior and exceptional behavior in certain situations. Executable code or design elements (colors, typography, or pictures) are not included in the wireframes, which distinguishes them from prototypes and mockups, respectively. Thus, wireframes can be assembled and tested quickly, e.g. by users, to determine whether the site or application structure is sensible (Becker & Berkemeyer, 2002; Murray, 1999) The elements of a wireframe are threefold: *information design* describes the presentation of information in form of information elements that are arranged on the web-page or the application, *navigation design* provides a set of screens that are related to each other and communicate through links, *interface design* determines the selection and arrangement of interface elements in order to provide the application’s or website’s functionality to the user (Garrett, 2010). An example for a wireframe is shown in Figure 4.

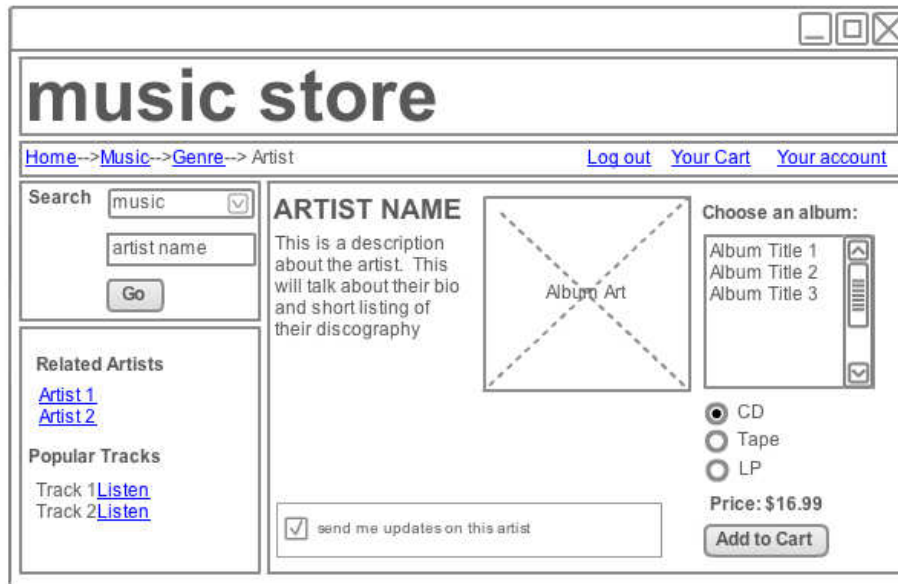


Figure 4: Example for a mockup (Gliffy, 2011)

Mockups promote the communication among users and developers by increasing the understanding of functional requirements. Thus, they represent a useful enhancement to the textual and formal scenario approach of use cases with no significant impact on effort. This is also due to the fact that the cognitive theory of multimedia learning includes textual as well as graphical cognitive channels in comprehension (Ricca et al., 2010a; Ricca et al., 2010b).

3.2 Product engineering

Possible developments of products and their environment can be analyzed and determined through product scenarios. They are used to identify potential prospective fields of application and requirements toward technologies resulting from that. The outcome of the product scenarios is a robust product which meets the current requirements and can be adjusted to possible developments in the future. Product scenarios can either be specific or general, whereas the specific type is used for designing a definite for a concrete supplier and the general type is applied when general designs or courses of action for a product are demanded (Paul, 1996; Gausemeier et al., 1996).

This section presents established scenario approaches in the domain of product development. These approaches are rather sophisticated and process-driven, compared to the solution-oriented scenario approaches in software engineering presented before in this research.

Scenario approach by Gausemeier

A scenario approach for the development of products is described by Gausemeier et al. (1996; 1998). According to this approach, the development of a product is always associated to a market for which product goals can be determined by scenario management. Furthermore, future-robust product strategies can be developed if the goal is already defined or known. Scenario projects follow the five stages: scenario preparation, scenario field analysis, scenario prognostics, scenario development, and scenario transfer. Each of these stages requires or delivers data elements that are important or essential for the creation of a product scenario. Since this section focuses on the form of expression of scenario approaches, the scenario development is the relevant phase with the scenario description being the point of interest.

The scenario description is divided into two parts, the *list of characteristics* and the *prosaic formulation of the scenarios*. When developing the lists of characteristics, the relevant projec-

tions for the description of a scenario are filtered from the pre-scenario catalogue. These relevant projections of a scenario, called characteristics (similar to key factors), are collected in lists of characteristics, which are complemented by noncritical characteristics. Two types of characteristics can be distinguished: A *distinct characteristic* of a scenario uses only one projection of a key factor for describing a scenario. An *alternative characteristic*, however, requires multiple projections of a key factor. These distinct and alternative characteristics are then documented in the list of characteristics.

The list of characteristics offers the scenario creator a framework for the scenario description. Scenarios can be described in a way that anybody who was not involved in the scenario creation or does not know the different future projections is able to understand it. The scenario descriptions are written down in natural language. It is important to consider the type of scenario that is desired (situational scenarios, process scenarios, etc.) and its attributes when describing it. Also, the coherences between the particular characteristics of a scenario should be mentioned. The description can be extended by the identification of disruptive factors or events, robustness, and sensitivity analyses, and includes when the description is of large scale summaries (up to 40 pages) and cross references might be appropriate.

The third component of a scenario description is the headline. It is an important element serving as a summary and arouses interest for the scenario. An example of a scenario description, according to Gausemeier (1996; 1998), is shown in Figure 5.

Scenario I: Human-centered age

List of characteristics

1	A:	Event shopping	Distinct characteristic
4	B:	External quality assurance	Distinct characteristic
7	A:	New fields of application for BZV	Single alternative
	B:	Stagnation and decline	Single alternative
12	B:	Extensive specialization	Distinct characteristic
18	A:	Price increase	Neglected
	B:	Decreasing or constant prices	Neglected
19	B:	Human banking	Distinct characteristic

...

Description of the scenario

The people in a human-centered age are facing a worldwide decline of mass factors: a break-up of mass markets, mass culture, and mass communication comes up:

- The mass markets of the industrial age will decompose to smaller segments, which are very directly influenced by the people of the post-industrial society [34A]. Through a consequent enforcement of the "Sustainable Development" concept [37A] environment and growth are harmonized.

...

Figure 5: Example for scenario description according to Gausemeier (1996)

Scenario approach by Paul

Paul (1996) embeds the scenario approach for product development into the context of business and technology scenarios, where each scenario type interacts with the other types. For

example, product scenarios demand robust technology, developed by a technology scenario, and vice versa. Since this section focuses on product scenarios, a detailed description of the other business scenario type is disregarded at this point.

After a business scenario with the request for a future-robust product has been developed, the product with the highest strategic relevance has to be filtered out and selected. This can be done with a technology-market-portfolio or by identifying the level of innovation, differentiation, etc. A company-specific delimitation of the product, i.e., how wide or narrow the problem statement is comprehended, is a first important step of the product scenario development. A question which has to be considered here would be: “How similar or related are the components, and can they be represented by one scenario or is a multiple scenario depiction better in terms of information quality?” The results of the product scenarios are various development possibilities of the product’s environment, e.g., market, customer, and technology developments. These development possibilities then set the demands towards the product, of which a robust product is generated. The robust product, in turn, can determine demands to the business strategy and the technology.

The features of a product are realized through the technologies used for it. Technology scenarios within the scenario-driven product development are needed to consider all the technologies related to the product and their coherences. In order to develop these promising technologies the functional structure has to be created. By drawing a functional structure tree, the main purpose and main function is decomposed into sub-purposes and subtasks. This structuring is continued until concrete solution approaches and elements can directly be assigned to the single functions, where possible (see Figure 6).

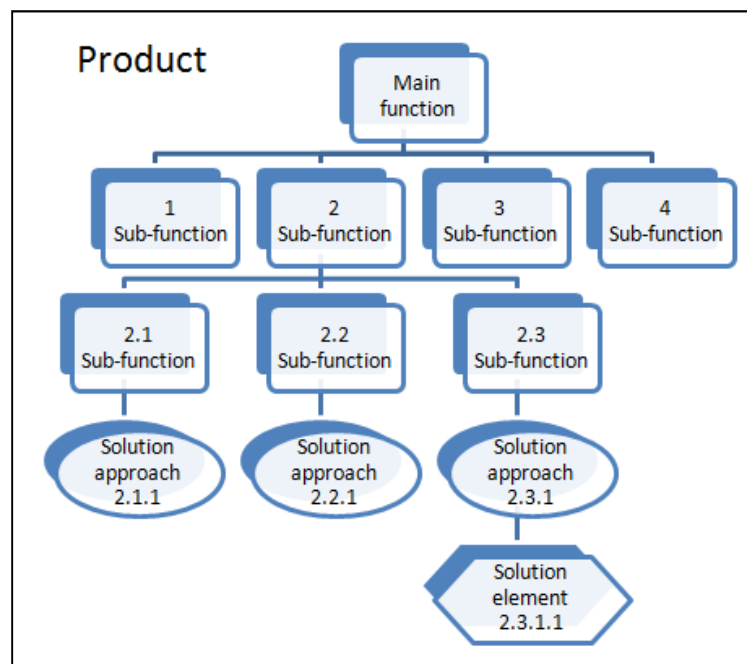


Figure 6: Example for a functional structure (Paul, 1996)

The solution approaches, i.e., the technologies are combined with each other in a way that the superior functions are fulfilled. This combination is continued successively until the overall function is achieved.

Subsequently, the product scenarios and their possibility funnel have to be adjusted if changes in the development of the product occurred due to the possible product characteristic. Only those generated possibilities of product realization are chosen which also deliver robust products under the changed circumstances. This means products that lead to success in all

given product scenarios. The rudimentary process and the outcome of developing product scenarios are shown in Figure 7.

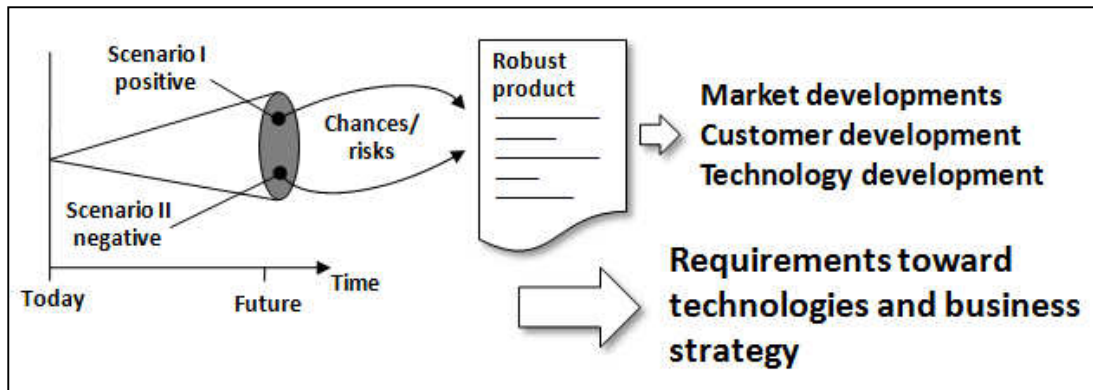


Figure 7: Result of the product scenarios (Paul, 1996)

3.3 Service engineering

Scenario approaches are particularly helpful in the field of service development processes. Their broad and divergent nature of alternative views helps identifying uncertainties and their driving forces, supports managing them, and enables managers to make informed decisions. Scenarios are also incorporated for managing risks by providing decision makers a broader view for evaluating impending decisions (Ahn & Skudlark, 2001). Since formal planning and forecasting methods have proven dissatisfactory, the use of scenario approaches became a widespread and popular planning method in the service domain (Phelps et al., 2001).

The following section presents the Service Blueprint, a scenario approach in the service domain.

Service Blueprint

Service Blueprinting (SB) is an approach to visualize service processes in an objective and distinct structure of service systems. Thereby, all different stakeholders of the service are able to understand and handle it impartially, regardless of their roles or their individual perspective (Zeithaml et al., 2005; Shostack, 1982). Apart from revealing weak spots in the service process, the SB can also be used to support the development of new market potentials and to test the quality of a service (Meis et al., 2010). According to our definition, a service blueprint allows for the description of a service scenario as a possible set of events that might reasonably take place.

The Service Blueprint is divided into a horizontal axis to show the chronological sequence of events conducted by the service customer and the service provider, and a vertical axis to distinguish between the different fields of action (Fließ & Kleinaltenkamp, 2004). The fields of action are illustrated by different swim lanes which separate each field into two service-relevant areas.

The *'line of interaction'* separates the customer processes from the supplier processes, and represents the direct interaction between those roles. The *'line of visibility'* separates the visible front end processes from the invisible back end supplier processes. The *'line of internal interaction'* distinguishes between the supportive processes of providing the employees the required functions to deliver the service and the invisible activities of the back office. Since support activities are conducted by employees with no customer contact, they require an *'internal interaction.'* The *'line of order penetration'* separates the integrative disposed, directly customer-induced activities, the service creation process, from the autonomously disposed, customer-independent activities, the service potential process. The *'line of implementation'*

separates preparation activities from facility activities. Preparation activities are conducted autonomously by the supplier and serve to plan, manage, and control a specific service process or multiple related services. Facility activities are logically and chronologically in front of the preparation activities and deal with the sourcing of potential and consumption factors (Fließ & Kleinaltenkamp, 2004).

Apart from these lines combined with the time dimension, all main functions of the service must be identified. Input and output factors must be shown and errors, bottlenecks, etc. must be handled. Finally, key figures like standard execution time and standard deviation are indicated in the blueprint in order to define the degree of variation where the consumer's perception of the overall quality and timeliness is not yet affected (Shostack, 1982).

A blueprint's form of expression can be done in various notations. The 'standard notation' (Fließ & Kleinaltenkamp, 2004), follows the notation of flow diagrams, the modeling of coherences and sequences of single activities (an example see Figure 8). Other alternative notations are Gantt charts or event-driven process chains (EPC) (Meis et al., 2010).

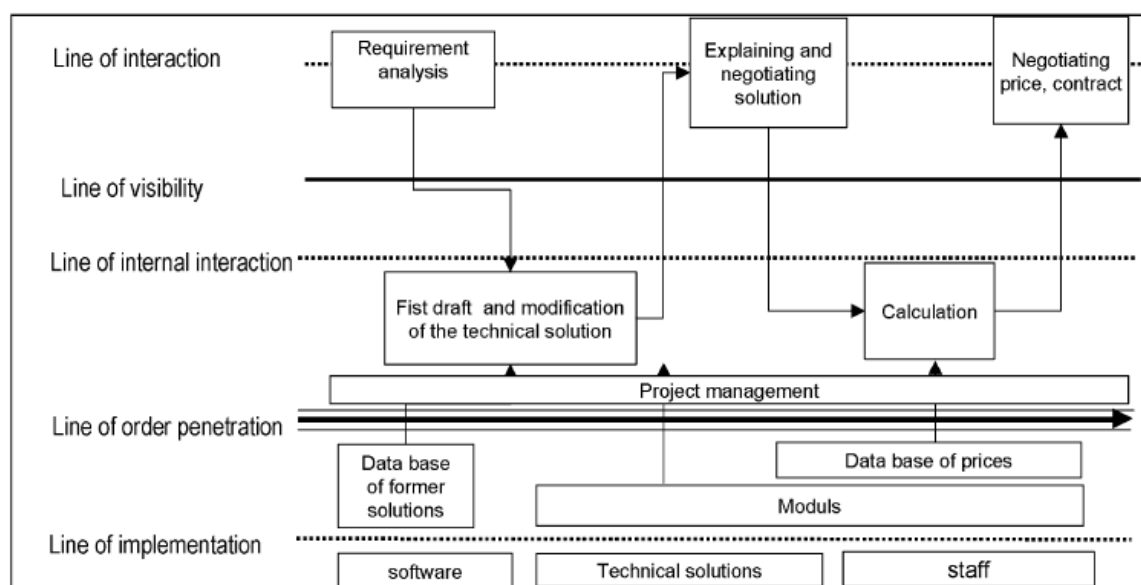


Figure 8: Service scenario of a simplified acquisition process (Fließ & Kleinaltenkamp, 2004)

3.4 Data classification of scenario approaches

This section gives a tabular overview of the scenario approaches described above. As mentioned earlier, the focus of this research and its outcome are result-oriented rather than on the procedural aspects of different approaches for scenario planning. For this reason, all relevant key data elements of the respective approaches have been collected and noted in the table. Additionally, the form of representation has been assigned to the data elements.

The overview of different methods for scenario planning can be used as a kind of toolbox or framework for users and innovation managers. With this collection at hand, methods for customer integration, such as communities or workshops, can be planned and implemented in order to repeatedly produce unambiguous and complete scenarios which can seamlessly be transferred into companies' internal development processes. The collected representation of data elements in this section allows for a quick overview of required or expected outcomes of a scenario planning process (see Table 4).

Attributes/ Data elements	Software Development			Product Development		Service Dev.
	Use case approach	Scenario approach by Bruegge/Dutoit	Mockups/Wireframes	Scenario approach by Gausemeier	Product scenario approach by Paul	Service Blueprint
Actors	G	T	-	-	-	G
Functions	G	-	G	T	G, T	G
System/product boundary	G	-	G	T	T	G
Relationships	G	-	-	-	-	G
Scenario name / headline	T	T	-	T	-	T
Scenario description	T	T	-	T	T	-
Entry condition	T	-	-	-	-	-
Exit condition	T	-	-	-	-	-
Quality requirements	T	-	-	-	-	-
List of characteristics	-	-	-	TA	-	-
Type of characteristic	-	-	-	TA	-	-
Opportunities/Threats	-	-	-	-	-	T
Information design	-	-	G, T	-	-	-
Navigation design	-	-	G, D	-	-	-
Interface design	-	-	G	-	-	-
Hierarchical menus	-	-	G	-	-	-
Menu system	-	-	G, L	-	-	-
Dummy pages	-	-	G	-	-	-
Data items (color, size, etc.)	-	-	T	-	-	-
development possibilities	-	-	-	-	T	-
solution approaches/elements	-	-	-	-	G, T	-
technologies	-	-	-	-	G, T	-
adjusted possibility funnel	-	-	-	-	G	-
Logic and time flow	-	-	-	-	-	D, G
Input/Output functions	-	-	-	-	-	G
Errors, Bottlenecks, etc.	-	-	-	-	-	T, G
Standards and Tolerances	-	-	-	-	-	T, G
Activity type	-	-	-	-	-	T
Swimlanes	-	-	-	-	-	T, D

T: (Structured) text, D: Diagram, G: Graphic, TA: Table

Table 4: Data elements of described scenario techniques (own table)

Table 4 summarizes all data elements of the scenarios that developers can achieve from each scenario technique. This can be used as decision support for an appropriate scenario technique by providing an overview of the results. It serves as a basis for describing the underlying data elements of scenario techniques and allows comparing different techniques. The table and data elements can be used by other authors to classify and describe additional scenario techniques.

4 Summary and further research

This paper deals with the issue of integrating customers in innovation processes in a comprehensible, formalized and manageable way. Therefore, we presented scenarios as a technique for the communication between customers and developers along the R&D process. The purpose of this contribution was to provide a basic understanding of scenarios by outlining scenario definitions, classifications as well as representations. The presentation of a theoretical scenario foundation as well as the given overview on existing scenario techniques from three innovation domains - product, service and software development - serve for a better understanding of scenarios as a way for communication between customers and developers. This overview can be used as a kind of toolbox or framework for innovation managers. With this collection at hand, methods for customer integration such as virtual innovation communities or workshops can be planned and implemented in order to repeatedly produce scenarios as a description of a possible future or of possible set of events that might reasonably take place,

which can be transferred into companies' internal development processes. On the one hand scenarios foster customers' creativity and allow them to articulate needs as well as possible solutions. On the other hand scenarios make it easier for developers to understand the needs of their customers and what they expect.

We focused and also limited our contribution to a sample of six common scenario techniques. The presented classification structure however offers the opportunity to further research to extend this exemplary collection in order to provide a more extensive collection of scenario techniques. As a basis for further research in the innovation management's field of handling user input and integrating it into R&D processes, this paper can also be picked up to investigate for example practical implications of scenario types on the choice or development of customer integration methods and associated tools. Furthermore relations and dependencies between the given definitions, dimensions and design parameters could be explored in detail. Another perspective for further exploration could be the combination of different scenario approaches in order to create new possibilities of structuring and presenting outcome. This can for example be done based on those approaches presented in this research by integrating the respective data elements with each other or, more general, by joining textual, graphical, and diagram elements. Another conceivable option to refine this paper's design framework for scenarios is the comparison and evaluation of the different scenario approaches regarding their applicability for specific task.

Acknowledgement

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research centre 'Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products'.

5 Bibliography

Ahn, J-H & Skudlark, A (2001): Managing risk in a new telecommunications service development process through a scenario planning approach, *Journal of Information Technology*, 17 (3), 103-118

Becker, S A & Berkemeyer, A (2002): Rapid Application Design and Testing of Web Usability, *IEEE MultiMedia*, 9, 38-46

Benard, A (1980): World oil and cold reality, *Harvard Business Review*, 58 (6), 91-108

Bergman, J & Jantunen, A & Saksa, J-M (2009): Enabling Open Innovation Process Through Interactive Methods: Scenarios And Group Decision Support Systems, *International Journal of Innovation Management*, 13 (1), 139-156

Bittner, K & Spence, I (2003): *Use case modeling*, Addison-Wesley

Booch, G & Rumbaugh, J & Jacobson, I (2006): *Das UML-Benutzerhandbuch*. Pearson Education

Bretschneider, U & Leimeister, J M & Krcmar, H (2009): Methoden der Kundenintegration in den Innovationsprozess: Eine Bestandsaufnahme, *Arbeitspapier des Lehrstuhl für Wirtschaftsinformatik, Technische Universität München*, 34

Bruegge, B & Dutoit, A H (2009): *Object-Oriented Software Engineering Using UML, Patterns, and Java*, 3rd edition, Prentice Hall

Carroll, J M (2000): Five reasons for scenario-based design, *Interacting with Computer* 13 (1), 43-60

Drew, S A W (2006): Building technology foresight: using scenarios to embrace innovation, *European Journal of Innovation Management*, 9 (3), 241-257

- Eversheim, W (2003): *Innovationsmanagement für technische Produkte*, 1st edition, Springer
- Filippidou, D (1998): Designing with scenarios: A critical review of current research and practice, *Requirements Engineering*, 3 (1), 1-22
- Fließ, S & Kleinaltenkamp, M (2004): Blueprinting the service company: Managing service processes efficiently, *Journal of Business Research*, 57 (4), 392-404
- Garrett, J J (2010): *The Elements of User Experience: User-Centered Design for the Web and Beyond*, New Riders Publishing
- Gausemeier, J & Fink, A & Schlake, O (1998): Scenario Management: An Approach to Develop Future Potentials, *Technological Forecasting and Social Change*, 59 (2), 111-130
- Gausemeier, J & Fink, A & Schlake, O (1996): *Szenario- Management. Planen und Führen mit Szenarien*, Hanser Fachbuchverlag
- Gliffy: http://www.gliffy.com/examples/wireframes/images/wireframe_example.png, accessed 08.04.2011
- Hitz, M & Kappel, G & Kapsammer, E (2005): *UML @ Work. Objektorientierte Modellierung mit UML 2*, 3rd edition, Dpunkt Verlag
- Huss, W R & Honton, E J (1987): Scenario planning—What style should you use?. *Long Range Planning*, 20 (4), 21–29
- Jacobson, I (1995): The use-case construct in object-oriented software engineering. In: Carroll J M (editors.): *Scenario-based design: envisioning work and technology in system development*. Hoboken: Wiley, 309–336.
- Jacobson, I & Bylund, S (2000): *The road to the unified software development process*, Cambridge University Press
- Jarke, M & Bui, X T & Carroll, J M (1998): Scenario Management: An Interdisciplinary Approach, *Requirements Engineering*, 3 (3-4), 155-173
- Larman, C (2002): *Applying UML and patterns: an introduction to object-oriented analysis and design and the unified process*, Prentice Hall PTR
- Meis, J & Menschner, P & Leimeister, J M (2010): Modellierung von Dienstleistungen mittels Business Service Blueprinting Modeling. In: Thomas, o & Nüttges M (editors.): *Dienstleistungsmodellierung 2010*, Heidelberg: Physica-Verlag HD, 39-64
- Murray, G & Costanzo, T (1999): Usability and the Web: An overview, *Network Notes*, 61
- Paul, M (1996): *Szenariobasiertes Konzipieren neuer Produkte des Maschinenbaus auf Grundlage möglicher zukünftiger Technologieentwicklung*, Universität Paderborn Heinz Nixdorf Inst.
- Phelps, R & Chan, C & Kapsalis, C (2001): Does scenario planning affect performance? Two exploratory studies, *Journal of Business Research*, 51 (3), 223-232
- Reichwald, R & Piller, F (2006): *Interaktive Wertschöpfung. Open Innovation, Individualisierung und neue Formen der Arbeitsteilung*, 1st edition, Gabler, Betriebswirt.-Vlg.
- Ricca, F & Scanniello, G & Torchiano, M & Reggio, G & Astesiano, E (2010a): On the effectiveness of screen mockups in requirements engineering: results from an internal replication, *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, New York: ACM, 17:1–17:10
- Ricca, F & Scanniello, G & Torchiano, M & Reggio, G & Astesiano, E (2010b): On the effort of augmenting use cases with screen mockups: results from a preliminary empirical study, *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, New York: ACM, 40:1–40:4
- Rolland, C & Achour, C B & Cauvet, C & Ralyté, J & Sutcliffe, A & Maiden, N & Jarke, M & Haumer, P & Pohl, K & Dubois, E & Heymans, P (1998): A proposal for a scenario classification framework, *Requirements Engineering*, 3, 23–47
- Schoemaker, P. J.H (1995): Scenario planning: a tool for strategic thinking. In: *Sloan Management Review*. 36 , 25–25.

- Shostack, G L (1982): How to Design a Service, *European Journal of Marketing*, 16 (1), 49-63
- Szulanski, G & Amin, K (2001): Learning to Make Strategy: Balancing Discipline and Imagination, *Long Range Planning*, 34 (5), 537-556
- Thomke, S & Von Hippel, E (2002): Customers as Innovators: A New Way to Create Value, *Harvard Business Review*, 80 (4), 74-81
- Ulwick, A W (2002): Turn Customer Input into Innovation, *Harvard Business Review*, 80 (1), 91-97
- Ulwick, A W (2005): *What customers want: using outcome-driven innovation to create breakthrough products and service*. McGraw-Hill Professional
- van Notten, P W F & Rotmans, J (2001): The future of scenarios, *Scenario and Strategy Planning*, 3 (1), 4-9
- van Notten, P W F & Rotmans, J & van Asselt, M B A & Rothman, D (2003): An updated scenario typology, *Futures*, 35 (5), 423-443
- von Hippel, E (1998): Economics of Product Development by Users: The Impact of "Sticky" Local Information, *Management Science*, 44 (5), 629-644
- von Hippel, E & Katz, R (2002): Shifting Innovation to Users via Toolkits, *Management Science*, 48 (7), 821-833
- von Hippel, E (2005): *Democratizing innovation*, MIT Press
- Weidenhaupt, K & Pohl, K & Jarke, M & Haumer, P (1998): Scenarios in system development: current practice, *Software, IEEE*, 15 (2), 34-45
- Wilms, F E P (2006): *Szenariotechnik: Vom Umgang mit der Zukunft*, 1st edition, Haupt Verlag
- Zeithaml, V A & Bitner, M J & Gremler, D D (2005): *Services marketing: integrating customer focus across the firm*, McGraw-Hill/Irwin