Adapted User Story Template

The adapted user story template is based on today’s de facto standard template by Connextra (Davies, n.d.), where the "so that" clause is optional:

“{…Who} As a <user or stakeholder type>

{What} I want <some software feature>

{Why} So that <some business value>”

By analyzing the results of our experiment on the identification of the difficulty of classifying user requirements (Maier & Berry, 2017), we found that we might increase the goodness of the classification when we put attention to the benefit-part of the user story, i.e., to the “so that” clause of the Connextra user story template. The benefit-part of a user story is the rationale of a user story. If a rationale is missing, classifying it becomes almost impossible. So, we have to ensure that each user story provides exactly one rationale of a requirement. If there are several rationales, add additional requirements.

The benefit at the beginning of the user story can denote a pragmatic quality or a hedonic quality the user expects the interactive software product to provide as the overall purpose of the user story. The stakeholder type denotes the perspective from which the user story is relevant. The new system feature denotes the concrete need of the user who provided the information included in the user story. The implementation of this need is necessary to fulfill the given purpose.

What the user story is still missing is information on the user’s intended interaction with the interactive software product, given that the need has been already implemented. Another missing information is the information on the source of the user requirement. As each user constructs his or her reality based on his or her experiences (Rupp, 2009), user requirements always stem from experiences that are related to the interactive software product for which the user formulates a requirement (Rupp, 2009; Ebert, 2012).

So, our adapted user story template focuses on the granularity levels of the goal of the user requirement being described as a user story: The purpose provides the most general goal of the user requirement, the need describes the functionality that the user needs to achieve the purpose, and the intention describes the concrete action the user wants to perform by using the functionality.

{Purpose} In order to -<Action>- -<Goal/Desire>- [<concretization of Goal/Desire>]

{Product Quality} by a better -<Product Quality>-,

{Perspective} As <Role>,

{Need} I want to be able to <Need>,

{Intention} to <Intention>.

For example, I have experienced

{Product Quality} a better <<Product Quality>>

When I interacted

{Related Software Product} with <Software Product>

{Functionality} that provided the possibility to <Functionality>

{Implementation} implemented by <Implementation>.

This user story template is separated into two parts: an essential user requirement part that comprises the first five lines from ‘Purpose’ to ‘Intention’ and an underlying experience part, which provides information on the source of the user requirement.

In the template, the curly brackets denote the labels of the respective lines, i.e., the information that is provided in the respective line. This way, an analyst finds the required information without having to read the whole user requirement.

Angle brackets with hyphens, “-<” and “>-”, denote information that is selected from a list.
Goal/Desire is selected from the elements of pragmatic quality or hedonic quality from the UX Quality Model, Action is selected from a list of possible modifications of the selected element of pragmatic quality or hedonic quality: A selected element can be increased, decreased, adapted, or enabled. Eventually, Product Quality is selected from the elements of the system quality from the UX Quality Model.

Single angle brackets, “<” and “>”, denote information that the writer of a user story has to provide as free text.

Double angle brackets, “<<” and “>>”, denote information that is exactly copied from an information provided in a previous line of the user story. In the user story template, the only information that is copied is the Product Quality. The repeat of the product quality in the related experience part of the user story creates a stronger link between the related experience part and the essential user requirement part of the user story.

Square brackets denote optional information. The only optional information within the user story template is the information on the specific context of the modification of a particular pragmatic quality or hedonic quality, which might be necessary in some cases.

Related Software Product refers to an interactive software product that the originator of the requirement used in the past. It should specify a concrete system that is named and linked to if possible, so that a developer who has to work with the requirement is able to visit the system that caused the experiences that the originator of the requirement describes.

Examples:

An example of a pragmatic user requirement in the adapted user story template is:
{Purpose} In order to increase the completeness with which specific goals are achieved
{Product Quality} by a better functional appropriateness,
{Perspective} as a participant in a delivery (e.g., recipient, supplier),
{Need} I want to be able to communicate with the other participant,
{Intention} to be able to coordinate the delivery.
For example, I have experienced
{Product Quality} a better functional appropriateness
when I interacted
{Related Software Product} with a logistics company's website
{Functionality} that provided the possibility to communicate with the other participant,
{Implementation} implemented by a messenger.

An example of a hedonic user requirement in the adapted user story template is:

{Purpose} In order to increase the enablement of my personal development
{Product Quality} by a better accountability,
{Perspective} as a citizen,
{Need}I want to be able to be informed about my partners in the project
{Intention} to learn who the organizations involved are and what they do in the context of the project.

For example, I have experienced
{Product Quality} a better accountability
when I interacted
{Related Software Product} with an online learning platform
{Functionality} that provided the possibility to be informed about the project partners,
{Implementation} implemented by a detailed partner description page with corresponding legal information and all information about the partners providing the content, together with links to websites and other information regarding the partners.

User Requirement Classification Trainer

The analysis of the results of the experiment on the difficulty of the classification of user requirements revealed some reasons for the difficulty and corresponding improvement suggestions for reducing these difficulties.

Subjects of the first experiment indicated issues with the separation of HQ and product quality or functionality. When a subject found indications of a product quality or functionality in a user requirement, the subject classified this user requirement as pragmatic, even if included keywords made this requirement hedonic. So, one of the improvement suggestions is a clarification that each hedonic quality requires an appropriate functionality to show up, i.e., functionality and hedonic quality do not contradict each other. A related improvement suggestion is the provision of a stronger distinction between product quality, e.g., functionality, a task, a process, and the experiences a user makes when using a product, i.e., PQ and HQ. Finally, some subjects mentioned that they were missing examples of pragmatic user requirements and hedonic user requirements. So, the definitions should be improved by the provision of examples that clarify the definitions.

Apart from the provision of general information on the definitions of PQ and HQ in detail as the essence of the systematic literature research on UX emergence and on UX Quality Models, the User Requirement Classification Trainer will take into account these improvements.

With the User Requirement Classification Trainer, a user will learn how to classify user requirements into PQ-related user requirements and HQ-related ones. The trainer was developed as a browser tool in Java 8 and JavaScript. The information provided by the trainer was taken from literature and from the UX Quality Model. Apart from one exception, sources of the respective information given in the trainer will not be provided in this section. Instead, the text will be given exactly as it is provided in a browser. In a browser, however, for each example the trainer shows the path from the keyword in the example to PQ or HQ superimposed on a tree diagram from the UX Quality Model. Figure 1 shows the decision tree with the elements of PQ and HQ.

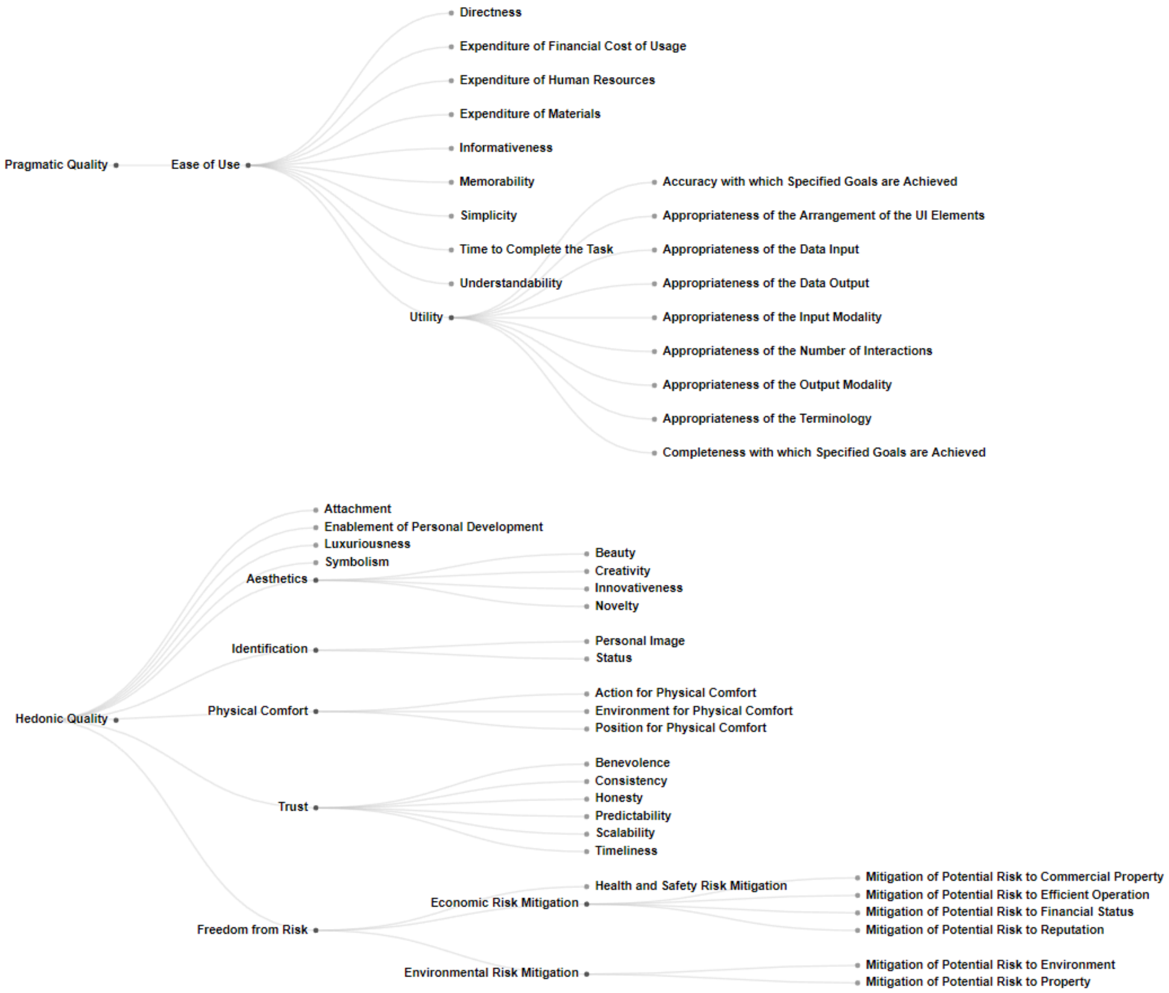
Pragmatic Quality
Ease of Use
Directness
Expenditure of Financial Cost of Usage
Expenditure of Human Resources
Expenditure of Materials
Informativeness
Memorability
Simplicity
Time to Complete the Task
Understandability
Utility
Accuracy with which Specified Goals are Achieved
Appropriateness of the Arrangement of the UI Elements
Appropriateness of the Data Input
Appropriateness of the Data Output
Appropriateness of the Input Modality
Appropriateness of the Number of Interactions
Appropriateness of the Output Modality
Appropriateness of the Terminology
Completeness with which Specified Goals are Achieved
Hedonic Quality
Attachment
Enablement of Personal Development
Luxuriousness
Symbolism
Aesthetics
Beauty
Creativity
Innovativeness
Novelty
Identification
Personal Image
Status
Physical Comfort
Action for Physical Comfort
Environment for Physical Comfort
Position for Physical Comfort
Trust
Benevolence
Consistency
Honesty
Predictability
Scalability
Timeliness
Freedom from Risk
Economic Risk Mitigation
Health and Safety Risk Mitigation
Mitigation of Potential Risk to Commercial Property
Mitigation of Potential Risk to Efficient Operation
Mitigation of Potential Risk to Financial Status
Mitigation of Potential Risk to Reputation
Environmental Risk Mitigation
Mitigation of Potential Risk to Environment
Mitigation of Potential Risk to Property

Figure 1: Decision Tree with the Elements of PQ and HQ

The trainer comprises two parts, a training part and an exercise part, where the user will be asked to apply his newly gained skills.

Training Part

The training part of the trainer consists of eight sections that describe:

- the emergence of user experience,
- the connection of user requirements and user experience,
    - what are ambiguous user requirements,
- how to classify user requirements,
- the identification of PQ-related user requirements,
- examples of PQ-related user requirements,
- the identification of HQ-related user requirements, and
- examples of HQ-related user requirements.

In this experiment package, only those parts of the trainer are shown that are most important for understanding what the subjects of the experiment learned to correctly classify user requirements. Furthermore, for each of a PQ-related user requirement and an HQ-related user requirement, only one example is shown in this experiment package. The paper that describes the experiment includes additional examples. The most important parts for the purpose of understanding what the subjects learned by using the trainer are:

- how to classify user requirements,
- the identification of PQ-related user requirements,
- example of a PQ-related user requirement,
- the identification of HQ-related user requirements, and
- example of an HQ-related user requirement.

How to Classify User Requirements

The classification of a user requirement into a PQ-related one or a HQ-related one comprises three steps:

1. Identify the focus of the requirement, based on corresponding key words in the user requirement. The most important part of a user requirement is (a) the one that begins with 'in order to' or something similar and describes the rationale of the requirement from a user's point of view, or (b) a PQ assessment that describes the overall usefulness of the interactive software product or, as HQ assessment, the description of the product's overall appeal. Regarding PQ, an overall useful product is described by assessments like, amongst similar ones, clear, supporting, useful, structured, practical, predictable, simple, and controllable. Regarding HQ, a product's overall appeal is described by assessments like, amongst similar ones, captivating, stylish, beautiful, novel, premium, creative, original, innovative, presentable, integrating, exciting, amusing, thrilling, fun, enjoyable, interesting, pleasant, impressive, motivating, inventive, playful, competitive, exploratory, and reliable. If a user requirement contains such an assessment, the user requirement does not have to include a rationale in addition, since the assessment provides sufficient information to classify the user requirement correctly. Hence, proceed with Step 3.

2. If a user requirement lacks a rationale and an assessment of its overall usefulness and its overall appeal, respectively, stick to the explicit information provided by the requirement and do not try to over-interpret the requirement by assigning implicit meaning to it. A user requirement that lacks a rationale and an assessment of its overall usefulness and its overall appeal, respectively, is either an ambiguous user requirement for which you have to elicit additional information in order to decide if the user requirement is pragmatic or hedonic, or the user requirement addresses an objective product quality. In both cases, skip the remaining user requirement classification process and restart it at Step 1 when you have elicited additional information.

3. Identify which criterion of PQ or HQ the identified focus of the user requirement refers to. For the identification, use the decision trees for PQ and HQ that you see below. Remember that an HQ is built on a PQ, and that, in turn, each PQ is built on an objective product quality. That is, when a user requirement includes any key word that refers to HQ, the requirement is HQ-related, even if the same requirement includes key words that refer to PQ or to an objective product quality. When a user requirement includes a keyword that indicates PQ, the requirement is classified as pragmatic, even if the requirement also includes a product quality or if the requirement can be interpreted as hedonic, although a corresponding keyword is not explicitly given.

4. Classify the user requirement as PQ-related or as HQ-related, depending on the source of the criterion you identified in Step 2 or on the assessment of the product's overall usefulness and its overall appeal, respectively.

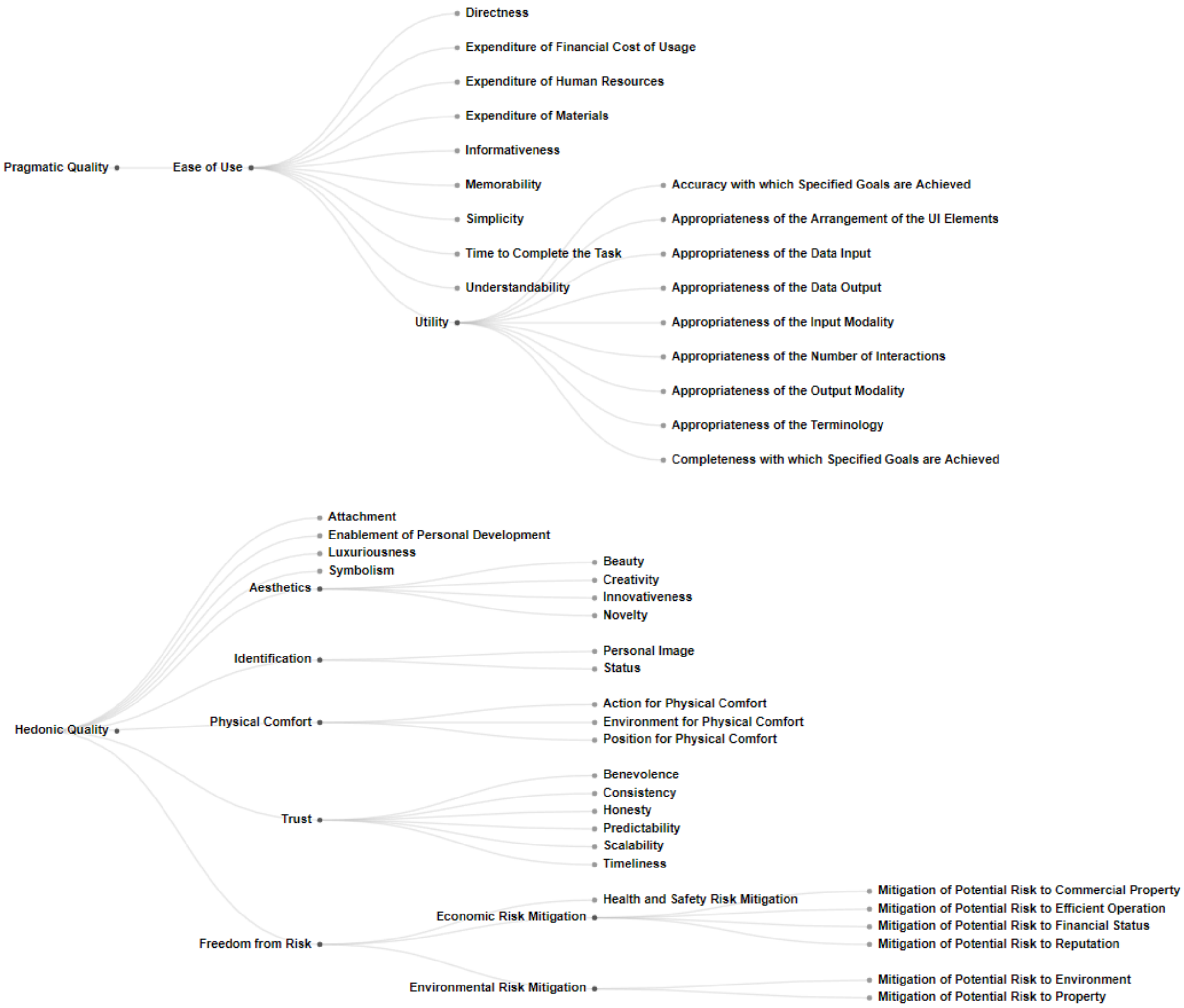
Pragmatic Quality
Ease of Use
Directness
Expenditure of Financial Cost of Usage
Expenditure of Human Resources
Expenditure of Materials
Informativeness
Memorability
Simplicity
Time to Complete the Task
Understandability
Utility
Accuracy with which Specified Goals are Achieved
Appropriateness of the Arrangement of the UI Elements
Appropriateness of the Data Input
Appropriateness of the Data Output
Appropriateness of the Input Modality
Appropriateness of the Number of Interactions
Appropriateness of the Output Modality
Appropriateness of the Terminology
Completeness with which Specified Goals are Achieved
Hedonic Quality
Attachment
Enablement of Personal Development
Luxuriousness
Symbolism
Aesthetics
Beauty
Creativity
Innovativeness
Novelty
Identification
Personal Image
Status
Physical Comfort
Action for Physical Comfort
Environment for Physical Comfort
Position for Physical Comfort
Trust
Benevolence
Consistency
Honesty
Predictability
Scalability
Timeliness
Freedom from Risk
Economic Risk Mitigation
Health and Safety Risk Mitigation
Mitigation of Potential Risk to Commercial Property
Mitigation of Potential Risk to Efficient Operation
Mitigation of Potential Risk to Financial Status
Mitigation of Potential Risk to Reputation
Environmental Risk Mitigation
Mitigation of Potential Risk to Environment
Mitigation of Potential Risk to Property

Figure 2: Decision Trees for Identifying PQ and HQ

The Identification of PQ-Related User Requirements

Pragmatic quality refers to a product's relevant functionality to effectively achieve a particular task, i.e., utility, and ways to access this functionality in a quick and easy way, i.e., ease of use.

Example of a PQ-Related User Requirement

Example: The product should provide a clear screen layout.'

Rationale: "clear" is an attribute that concerns the 'expenditure of human resources' that indicates 'ease of use', which is a criterion of pragmatic quality.

Although the provision of a clear screen layout might also be categorized as an aspect of utility or of 'aesthetics', the attribute 'clear' is in the focus of this requirement. The requirement does not refer to the product's fulfillment of a particular task, which would indicate its utility, nor does the requirement refer to the stimulation of the user's senses, which would indicate 'aesthetics'. A clear screen layout decreases the user's cognitive load and thus the expenditure of his resources.

The Identification of HQ-Related User Requirements

Hedonic quality covers all product attributes apart from utility and ease of use that emphasize an individuals' psychological well-being.

In particular, HQ comprises nine criteria:

- Enablement of personal development: The proliferation of knowledge and the development of skills by the achievement of some challenge; the provision of new impressions, opportunities, and insights.
- Identification: The provision of a self-expressive function; being socially recognized and exerting power over others.
- Symbolism: The provocation of memories of individually important past events, relationships, or thoughts.
- Attachment: Ability for the user to attach subjective value to a product/service; affection, fondness, or sympathy for someone or something.
- Aesthetics: A user interface enables pleasing and satisfying sensory interaction for the user.
- Luxuriousness: The degree to which a product/service is luxurious or looks expensive and superior in quality.
- Trust: The user's confidence that a system will behave as intended; a group of beliefs held by a person derived from his or her perceptions about certain attributes, especially with respect to honesty and benevolence perceived in the behavior of the other party (Flavian et al., 2006). Honesty is the belief that the system, service, person, or other entity will keep its, his, or her word, fulfill promises, is free of deceit, truthful and sincere (Oxford Dictionaries: https://en.oxforddictionaries.com/definition/honesty; https://en.oxforddictionaries.com/definition/honest). Benevolence is the belief that the system, service, person, or other entity is well meaning and kind, is interested in the wellbeing of another without intention of opportunistic behavior, motivated by a search for a mutually beneficial relationship (Oxford Dictionaries: https://en.oxforddictionaries.com/definition/benevolence).
- Physical comfort: The user is satisfied with physical comfort.
- Freedom from risk: A system mitigates the potential risk to economic status, human life, health, or the environment.

Example of an HQ-Related User Requirement

Example: My smart phone must not create a profile of my movements, which could allow a third party to know where I am and where I have been.
Rationale: This concern reflects a lack of honesty of the system relating to someone's possibility to misuse the user's movement profile against this user, and thus it indicates 'trust', which is a criterion of hedonic quality.

The Exercise Part of the User Requirements Classification Trainer

After the training part of the User Requirements Classification Trainer, a user of the trainer is asked to classify 41 user requirements as pragmatic or as hedonic. But before the exercise begins, the user can repeat the training part until he feels trained enough to start the exercise. The 41 user requirements are composed of 10 pragmatic user requirements (four for each of the two criteria of PQ one additional user requirement in the adapted user requirement, and one additional original user requirement from a research project) and 31 hedonic user requirements (3 for each of the nine criteria of HQ, additionally (1) two additional user requirements represented in the adapted user story template, and (2) two of the original user requirements from the research project). Five of the pragmatic user requirements and ten of the hedonic user requirements that are provided in the exercise part of the trainer were also used in the training part. So, the users of the trainer are assumed to correctly classify these repeated examples at the first time they are asked to classify those user requirements.
While the examples in the training part of the trainer were provided in a strict order, the examples in the exercise part of the trainer are provided in a random order that is changed with each run of the trainer.
Each example is repeated once when it was not classified
Correctly at the first time. So, an example is provided twice, at last, even if it is wrongly classified for the second time. When an example is shown, the user can select if the decision tree shall be shown or not. The selection is preserved for the next examples, until the user decides on hiding the decision tree

again. The correct path from a keyword within the example to the correct root (PQ or HQ) is not shown in the decision tree. The decision tree is just meant to prevent the user from keeping the UX Quality Model in mind. So, the classification of the user requirements is not influenced by the user's cognitive load and by wrong memories of the criteria and sub-criteria of PQ and HQ. As in the training part of the trainer, the correct classification of an example is shown with the arguments for the classification and the corresponding tree diagram before the next example is shown, even if the user requirement was classified correctly.

References

Evans, David, and Adzic, Gojko (2014). "'As a, I want, So that' Considered Harmful". http://blog.crisp.se/2014/09/25/david-evans/as-a-i-want-so-that-considered-harmful. Retrieved 2017-08-09.

Cohn, Mike. "User Story Template Advantages". https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template. Retrieved 2017-08-09.

Davies, Rachel: "Connextra User Story 2001: ConnextraStoryCard". http://agilecoach.typepad.com/photos/connextra_user_story_2001/connextrastorycard.html. Retrieved 2017-08-09.

Ebert, C. (2012). Systematisches Requirements Engineering. Vasa. Retrieved from http://medcontent.metapress.com/index/A65RM03P4874243N.pdf
Marcano, Antony (2011). "OLD FAVOURITE: FEATURE INJECTION USER STORIES ON A BUSINESS VALUE THEME". http://antonymarcano.com/blog/2011/03/fi_stories/. Retrieved 2017-08-09.

Rupp, C. (2009). Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis (5th ed.). Carl Hanser Verlag GmbH & Co. KG.