5-12-2012

APERTO: A Framework for Selection, Introduction, and Optimization of Corporate Social Software

Alexander Richter
Universitat der Bundeswehr München, a.richter@unibw.de

Sebastian Behrendt
Universitat der Bundeswehr München, s.behrendt@unibw.de

Michael Koch
Universitat der Bundeswehr München, michael.koch@unibw.de

Follow this and additional works at: http://aisel.aisnet.org/sprouts_all

APERTO: A Framework for Selection, Introduction, and Optimization of Corporate Social Software

Alexander Richter
Universitat der Bundeswehr München, Germany

Sebastian Behrendt
Universitat der Bundeswehr München, Germany

Michael Koch
Universitat der Bundeswehr München, Germany

Abstract
The deployment of social software in enterprises to support collaborative work has become increasingly important in the past few years. At the same time, the characteristics of social software--most importantly the so-called Nutzungsoffenheit--require a change of mindset. Corporate social software differs strikingly from traditional business software, which has clearly defined common usage scenarios for its functions. Classic approaches concerning the requirements analysis, change management and success measurement of business software can be applied only partly or not at all. In this report, the APERTO framework, consisting of the APERTO five-level model, the CUP-Matrix, as well as the tools developed therefrom, is introduced. It enables a complete and consistent categorization and classification of the usage potentials of corporate social software, and thus supports its selection, implementation, and optimization. The approach described in this report was applied successfully multiple times in the past few months in projects to select and implement solutions in various German enterprises.

Keywords: enterprise 2.0, corporate social software, APERTO, requirements analysis, success measurement

PERSPECTIVES ON CORPORATE SOCIAL SOFTWARE

Core characteristic: Nutzungsoffenheit

Social software has become increasingly important in many corporations in the past number of years. The use of wikis, weblogs, and social networking services can cause a fundamental change of communication within a company (Cook, 2008; Buhse & Stamer, 2008; McAfee, 2009). This change is not only evident in the use of the tools, but in fundamental changes in communication which are enabled by these tools: information consumers become information producers, who create contents voluntarily and self-organized (e.g. Stocker et al., 2012). In this context, social software is, compared to previous information systems, characterized by a significantly stronger orientation towards the requirements of the user (often referred to as "me-centricity", e.g. Back & Koch, 2011).

Another very important characteristic of social software is the so-called Nutzungsoffenheit. Nutzungsoffenheit means that the true nature and potential of such technologies does only manifest when people make sense of and incorporate them in their day-to-day work routines. In essence, the technology and its set of features do not precipitate its forms of usage (Riemer et al., 2009, p. 186).

This adoption process lasts until the users have discovered the sense and purpose for themselves and incorporated the platform into their daily work processes[1] . In this aspect, corporate social software differs strikingly from traditional business software such as ERP-, CRM-, or PPC-systems, which have clear structures, processes and pre-defined use cases underlying already during their development (Al-Mudimigh et al., 2001). Consequently, while introducing social software, a company faces the challenge of deducting its potential in its own corporate context.

Process model for introduction

Before the implications of this characteristic of social software are described any further, it is helpful to discuss the process of a system implementation, to highlight the challenges during this process in an enterprise. Various frameworks or process models exist for the introduction of software (e.g. Shin & Lee, 1996). Based on the fundamentally similar structure of these models and on our own experience and observations regarding the introduction of social software in a number of companies, the process for introduction which is shown in Figure 1 is proposed.

In a strategy-development phase, the project team defines a concrete project goal[2] and develops a vision that describes the long-term goals of the implementation project. Only if such goals are defined it is possible later on to define the requirements or to measure or monitor the success of the platform.

- In the selection phase, the usage of currently existing platforms is analyzed (actual) and the requirements of the future platform (target) are determined. The platform evaluation goes beyond the selection phase, as the platform is still being evaluated during its introduction, for example by pilot users.

---

[1] This is a dynamic process, which does not have to be completed at a certain moment, even though the user has already realized and perceived an additional value. An overview of the phases to pass through during this adoption process shown in the SNEP-model (Riemer et al. 2012).

[2] The process and the necessity of the definition of a project goal do not exclude an explorative approach (Richter & Stocker 2011). In this case, the goal of the usage is the identification of use cases.

- Depending on the size of the group to which the platform is introduced, there might be distinct pilot and roll-out phases. However, most important in terms of change management is the guidance of the users based on a benefit-oriented documentation.
- In the ongoing usage, the success of the platform should be measured in order to ensure optimized usage in the course of a continuous adaptation.
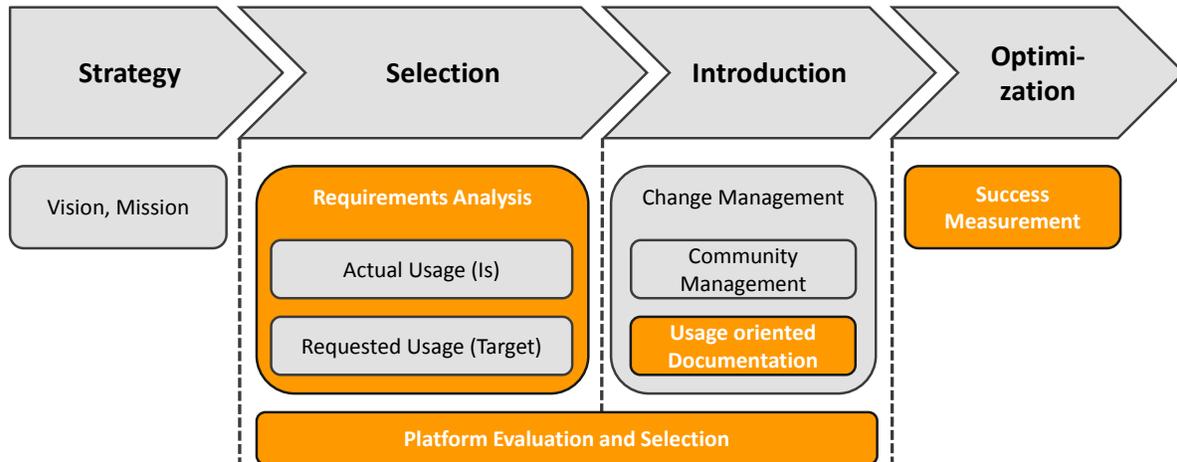

Figure 1: Process model for introduction of corporate social software

Each of the highlighted steps (requirements analysis, platform evaluation and selection, benefit-oriented documentation, success measurement) will be explained in more detail in this report.

Necessity for a multiple-level observation

As described above, for traditional business software (such as ERP, CRM, or PPC systems), it is usually possible to clearly associate use cases with functions or function bundles. For instance, the stock-management form in an ERP system is used only for managing the stock. Usually, it is even possible to associate one function (e.g. manual incoming inspection) with a specific business process (e.g. acquisition). Due to the Nutzungsoffenheit this does not apply to social software. Thus, it is not expedient during the requirements analysis of social software to orientate oneself by the functions of the platform, since usually multiple functions exist to support the same work practice and the same function can be applied to support different work practices. For this reason, it is particularly important to differentiate between functions and their possible usage as part of a work practice. For this complex problem, it is necessary to be clear about the fact that various observation levels exist, and that they have to be described in a structured and understandable manner.

It has been proposed variously to inspect IT systems on different levels. Krcmar (2005) uses a five-level model for information management that, beginning with the basic functionalities, stretches over basic technology to technology bundles and then to function-oriented technology bundles, which finally are applied in usage contexts. Also, multiple levels of observation or abstraction are usually foreseen when modeling business processes. The architecture of integrated information systems (ARIS) proposes five distinct process levels: function or transaction at the bottom level, followed by activity or process step, process, main process or (business) scenario, and finally process field (Scheer et al., 2006).

3

A multiple-level approach seems useful for corporate social software as well. Briggs et al. (2009) have already proposed a seven-level model for collaboration, in which they structure right from the individual functions (scripts) all the way to the desired goals. Unfortunately, the authors of this model do not take the step from theory into practice, as this model is not applicable in implementation processes in its abstract form. The development of levels in the form of applicable models in cooperative systems was, until now, limited to the level of the functions. In this context, Büchner, Matthes, & Neubert (2009) and Williams & Schubert (2011) devised frameworks which describe the categorization of different functions of social software. Even though these frameworks enable an orientation on the level of functions, they lack consideration of the levels that build on top of that. This can lead to misunderstandings or meaningless analysis results. Measures may be undertaken which only lead to useful results at first sight, because of the missing knowledge about the differences of the observation levels.

The aperto five-level model for corporate social software

The following five-level model in Figure 2 intends to bridge this gap and to connect existing business processes with the functions included in corporate social software platforms. Additionally, it enables the alignment of one's own observations and thus ensures comparison. The five-level model is based on observations in different practice projects (see Chapter 2.1) and is part of the aperto framework[3].
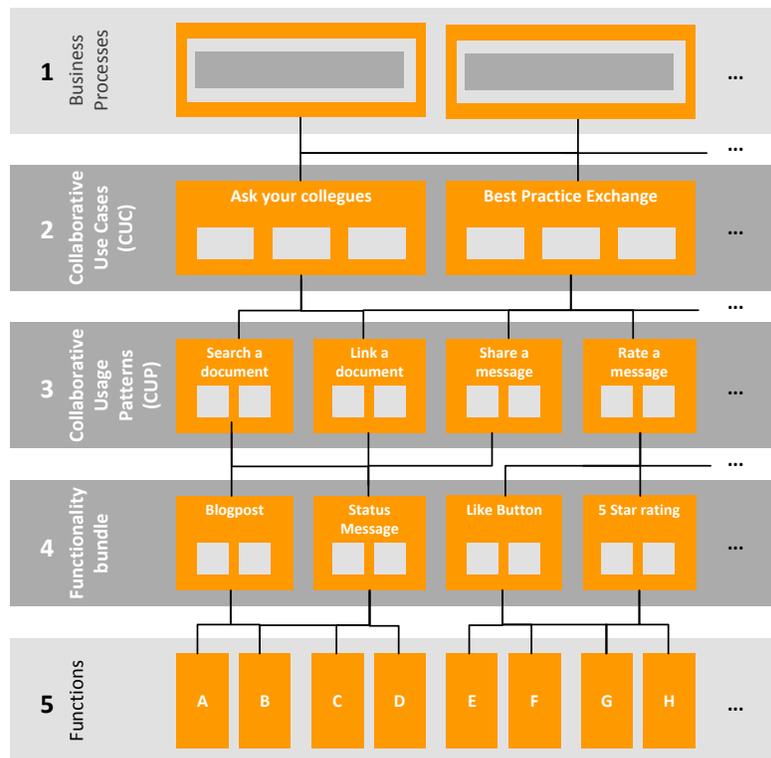

Figure 2: The aperto five-level model

---

[3] The aperto framework (aperto is Italian for open) is the sum of the five-level model, the CUP matrix located on the third level, and the tools for selection, introduction, and optimization of corporate social software developed from that.

The individual levels can be described as follows:

1. A business process describes a series of multiple activities that are executed sequentially in order to achieve a business goal (based on Staud, 2006, p. 9). Business processes can be encapsulated into one another and can be interdependent. This level will not be discussed in more detail, since processes may vary strongly between corporations.

2. Within a business process, multiple collaborative processes take place. This means (IT-supported) interaction takes place between multiple persons within a common activity and with the common goal to advance the business process (for instance asking a colleague for advice). It is possible to deduce a variety of such collaborative processes from existing experience reports of successfully introduced platforms. Within this report the term [Collaborative] Use Case (CUC) is used as common in practice. Because a CUC can appear in multiple business processes, there is an n:n relationship between Level 1 and Level 2.

3. On the third level, Collaborative Usage Patterns (CUPs) are found that can be subordinated to one or many CUCs. They describe in an abstract manner the collaborative options for use that can be covered by corporate social software. The level of the CUP is the direct transition from the verbalized requirements (CUC) to the technical realization and hence plays an important role in the selection, implementation and optimization process.

4. A CUP can be depicted through different function bundles. Due to the Nutzungsoffenheit, a function bundle can be applied to multiple CUPs, which again results in an n:n relation between these levels.

5. A function bundle consists of multiple individual functions, which also can be found in multiple function bundles (n:n relation).

The remaining report is structured as follows: in Chapter 2, the second and third level of the aperto five-level model and the Collaborative Use Cases (CUC) and Collaborative Usage Patterns (CUP) located on those levels are explained. Afterwards it is described how to apply the aperto framework to solve various challenges in the selection (Chapter 3), implementation (Chapter 4), and optimization (Chapter 5) of corporate social software.

COLLABORATIVE USE CASES AND COLLABORATIVE USAGE PATTERNS

In order to explain the second (Collaborative Use Cases) and third (Collaborative Usage Patterns) level of the aperto five-level model, the exemplary CUC "Ask your colleagues" is used. Based on this, the development of a model for Level 3 of the previously introduced aperto five-level model is explained.

Collaborative Use Cases

In practice-driven discussions, the second level of the aperto five-level model often serves to reveal critical points in using corporate social software or to make the benefits tangible for the users[4]. As part of the framework, a larger number[5] of Collaborative Use Cases (CUCs)

---

[4] E.g. http://socialsoftwarematrix.org/category or http://enterprise20blog.com/2009/10/15/classification-of-enterprise-20-use-cases for an overview

have been collected in a catalog based on the above-mentioned observations from introduction projects. For a company that has only little or no experience with social software, the CUC catalog (or similar lists) provides a possibility for orientation based on examples from successful corporate social software implementation projects. Collaborative Use Cases can — depending on the company structure and the business process — turn out to be very distinct. Therefore, a mutually exclusive and collectively exhaustive overview in term of the goals is not possible. An example from the CUC catalog is shown in Table 1.

CUC	Goals	Description
Ask your colleagues	• Fast and efficient problem solving	• Employee asks a question to (all other) employees
	• High-class answers	• Question gets answered by other employees
	• Reduction of redundant questions	• Author names the "most helpful answer" or the "correct answer"
		• Frequently Asked Questions can possibly be highlighted in a prominent location
…	…	…

Table 1: Exemplary excerpt from an individually customizable catalog with CUCs

Approach for developing the Collaborative Usage Patterns

As part of a long-term research project, the usage of social software in companies was analyzed systematically through qualitative analysis of a large dataset. Mainly interviews with platform stakeholder where used and the content exchanged on these platforms was analyzed as well. The dataset originates from more than 20 organizations[6] that worked with us in the past five years.

While developing the present results, the method of the permanent comparison[7] was applied. Using this qualitative approach, the data was initially "coded openly" individually per case with the goal of identifying distinct usage practices[8]. As a result of this coding step, more than 80 usage practices throughout all cases were identified. Next, all relations and contexts were

---

[5] Currently the catalog contains approximately 70 such Collaborative Use Cases, but it is growing continuously. However, in our opinion, the amount of Collaborative Use Cases is of secondary importance. More important is that the requirements of the employees are known in order to highlight the central Collaborative Use Cases to them.

[6] Among these organizations are: Accenture, Allianz, BMW, Bosch, different departments in the German Armed Forces, Capgemini, Communardo, Deutscher Skiverband, EADS, ESG, Fraport, IBM, Microsoft, Pentos, Rheinmetall, SAP, Siemens and T-Systems.

[7] This method is assigned to the socio-economic approach of the so-called "Grounded Theory" (Glaser & Strauss, 1967).

[8] While coding the content, the so-called method of genre-analysis was applied. Different results of these usage analyses were published in renowned conferences (e.g. Richter, 2010; Richter & Stocker, 2011; Richter, Mörl, Trier, & Koch, 2011; Riemer & Richter, 2010; Riemer et al., 2011)

identified and abstract classes of the use practices were created. This approach is also called theoretical coding (Böhm, 2004). Our analysis showed that all usage possibilities of corporate social software can be described by the two dimensions action and object. Figure 3 visualizes the entire approach.
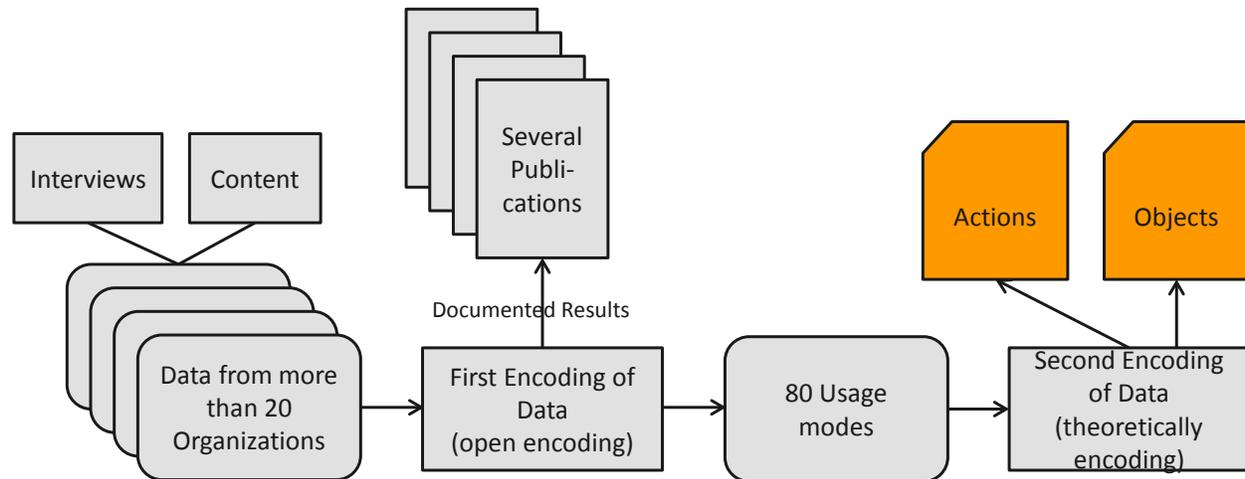

Figure 3: Approach used to develop the aperto framework

The abstract result of our analysis — the two dimensions action and object — will be explained shortly in the following.

Composition of Collaborative Usage Patterns

An action describes ***what*** is done within the platform, but not ***how*** the support of the action is realized through the platform[9]. Although the interaction with the system always follows a collaborative goal, and supports a collaborative process (as described in the five-level model), the intention of an action can be coordination, cooperation, or collaboration. An action can consist of multiple steps, whereby each step follows or requires another. The interactions within an action can happen synchronously or asynchronously, meaning that the timespan of the collaboration can stretch over multiple steps and does not necessarily happen at the time of one individual interaction. Furthermore, an action can be either directed or undirected regarding the recipients of an individual interaction.

Compared to classic information processing (Mertens & Hofmann, 1986) with the three process steps "input", "processing", and "output" it is not possible to determine a strict order of the actions due to the distinctness of the CUPs. Therefore, a simple grouping scheme was applied: it is possible to differentiate between different kinds of ***reception*** (search), ***enhancement*** (rate, tag, clarify, and edit), and ***delivery*** (share and notify)[10]. The following Table 2 briefly describes these in order to prevent misunderstandings.

---

[9] Depending on the software, this can be realized through different means. This will be explained in more detail later in this report.

[10] A similar structure related to social software is proposed by Ehms (2010, p. 49) within the three dimensions selection, annotation and publication. Our process differs in the way that the differentiations are more granular and occur detached from functionalities, which is necessary due to the initially described Nutzungsoffenheit.

7

	Action	Description
Reception	Search	Search for specific content using different criteria.
Enhancement	Edit	Direct changes and modifications of content in order to create a final, up-to-date version of the content.
	Rate	Rate the content in terms of quality or suitability for the specific purpose.
	Tag	Mark the content in order to allocate content to a certain subject or matter to increase retrievability when searching and collecting content.
	Clarify	Exchange different perspectives, interpretations, or opinions regarding a content.
Delivery	Notify	Notify others about relevant content which already exists.
	Share	Provide content which does not yet exist on the platform, in order to make it available to others.

Table 2: Actions of the CUPs

Besides the actions, a variety of different objects have been identified, e.g. table, photo, audio, video, person, skill, contact data, tasks, goal, appointment, deadline, decision, responsibility, poll, option, room, project, resource, service, discussion, experience, news, problem, idea, suggestion, link, opinion, status, location.

The abstraction of the objects[11] leads to three distinct object classes into which all objects can be classified:

- Message: Messages serve to deliver content, documents, persons, or reference and consist of multiple physical or logical units.
- Documents: Documents are collections of content, which are encapsulated or stored as one physical or logical unit.
- Person: Persons (profiles) are digital representations of real people.

Additionally, reference objects — connections to external content — emerged as a fourth class. As references do not contain content themselves — they link to content outside the platform — whether or not the content has to be taken into account depends on the business process. They will not be discussed in more detail.

The CUP-Matrix

The cross-combination of the seven identified actions with the three identified object classes leads to 21 abstract Collaborative Usage Patterns (CUPs) of social software in corporations, which are visualized in Figure 4 in the form of a matrix.

---

[11] We would like to point out that in business informatics a variety of abstraction classes for objects exist. As an example, we mention Trier (2005), who differentiates between activities, documents, employees, and topics. However, we cannot link any of the existing classifications to the results of our analysis.

	Message	Document	Person
Search	Search for advice	Search for "projectplan.doc"	Search for an expert
Edit	Refinde an idea	Edit a document within the platform	Add skills to a person profile
Rate	Rate an idea	Rate an image	Rate the project team
Label	Assign a suggestion to a event	Assign a document to a project	Mark a person as an expert
Clarify	Discuss an idea	Discuss the content of a document	Comment on John Does profile
Share	Communicate a task	Forward a document	Introduce a new colleague
Link	Point to an idea	Link to „guidelines.doc"	Send a link to a expert profile

Figure 4: CUP matrix

It is important to note that these 21 occurrences that result from the combination of an object class and an action (e.g. Search – message – "search for a hint") are not a Collaborative Usage Pattern but a concrete example. Hence, a CUP is the combination of an object class and an action that can be represented through multiple examples.

In the following chapters it is shown in detail how the CUPs and CUCs can support the aforementioned challenges of the implementation process, and have already successfully supported real projects.

SELECTION OF CORPORATE SOCIAL SOFTWARE WITH THE aperto FRAMEWORK

As explained briefly in Chapter 1, the functional requirements analysis[12] of corporate social software can only be done poorly through direct deviation of suitable functionalities. Thus, in the following sections, the possible application of the aperto framework — more specifically the CUP-Matrix — in the selection process for corporate social software is described (Figure 5). It is possible to identify three different steps: (1) functional requirements analysis (see Chapter

---

[12] Additional to the functional requirements analyses, non-functional requirements like integration options with other platforms or the usability have to be considered as well during the selection of corporate social software. In this point social software acts like other business applications. For this reason the already mentioned and other non-functional requirements are not considered in the described approach and have to be evaluated otherwise.

9

3.1), (2) platform evaluation (see Chapter 3.2), and (3) the comparison of the requirements profile with the platform profiles based on the CUPs (see Chapter 3.3).
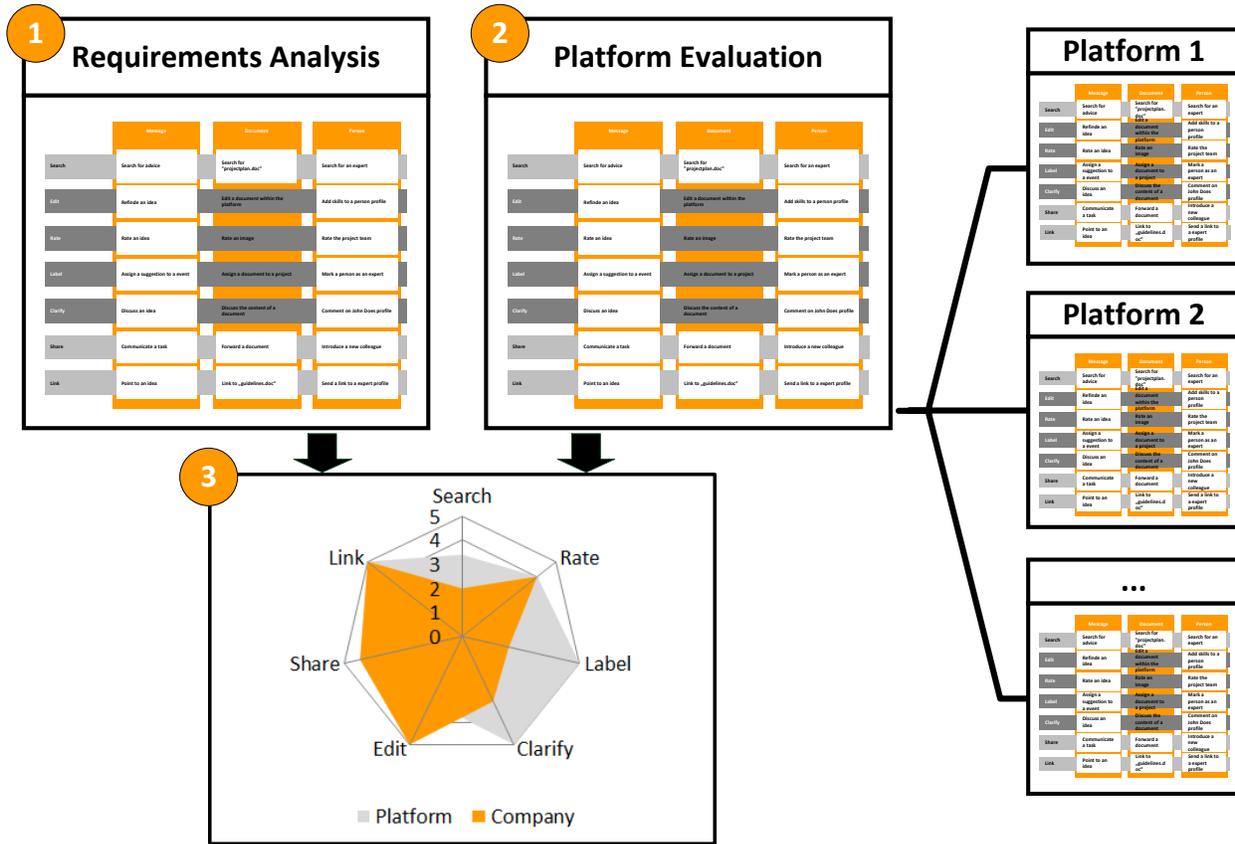

Figure 5: Selection process for corporate social software

Functional Requirements Analysis

The idea of the functional requirements analysis is to assess and rate the relevant CUPs in the usage context. The results can then be displayed as SpiderWebs, and can be directly compared to any possible platforms, which have been rated accordingly (see also chapters 3.2 and 3.3). In this way, a "step backwards" to the level of the functionalities can be avoided. The CUP-Matrix mainly helps here to focus and prioritize the company-relevant CUPs.

Different approaches are possible here:

- **Interviews**: On the one hand it is possible to conduct interviews with potential platform users to identify their requirements and draft them as CUCs. Multiple employees in the concerned field of business are questioned for their tasks and work practices. This survey is helpful to analyze in-depth the activities from the actors perspective. After the interviews, the identified activities are analyzed. The activities are grouped accordingly to the CUPs afterwards. If requirements arise, which only occur rarely, it is necessary to readjust in order to find out whether or not they are justified or why they have not been pointed out more often.

- **Usage analysis of existing platforms**: Many companies decide to start to explore the potential of the platforms for their business by using solutions that are easy to install (e.g. Wordpress) or freely available in the Internet (e.g. Yammer.com) (cf. Richter & Stocker, 2011). In this case it is possible to examine the messages exchanges on the platform by using a genre analysis (Riemer & Filius, 2009). Genre analysis can serve as an instrument to understand the communication practices of a social group, because "in identifying and labeling genres we try to capture the gestalt of the various components of the communicative act." (Kwasnik & Crowston, 2005, 80). Table 3 shows how in the case of the Siemens Technoweb 2.0 (Richter et al., 2011) every post was coded according to its purpose (e.g. search for a discussion partner).

Use Case	All[13]
News/ Status Updates	20%
Invitations	4%
Forwarding of links to articles, ...	11%
Pointers to experts	5%
Forwarding experiences	10%
Search for discussion partners	3%
Enquiry	4%
Opinion	20%
Enhancing an idea	1%
Search for support	9%
Preparation of decissions	2%
Search for reference	1%
Identification of contact person	3%
Marketing of central initiatives and internal services	6%
Meta communikation (e.g. comments on the platform)	7%

Table 3: Results of genre analysis in the case of Siemens

From the identified communication practices and from the requirements identified by the interviews the corresponding CUPs can be derived[14]. They are presented as SpiderWebs according to their frequency and relevance. The SpiderWebs visualize the core requirements of the corporation or its divisions concerning the CUPs in a simple manner. Figure 6 shows exemplarily the characteristics of the CUPs grouped by the three objects document, person and message. Prioritization is on a scale from 1 through 5. From a qualitative assessment, the following values can be deduced: the CUP has no importance (1), medium importance (3), or high importance (5). Values in between can arise from averaging the results from multiple interviewees. The SpiderWebs of the three objects document, person, and message in the

---

[13] This is the share of each genre. Each post can contain multiple genres.

[14] As mentioned in the first chapter the adoption happens step by step in a long-term process, which means that the communication practices can change over time. But this fact does not change the necessity to identify the requirements at a certain time and to make selection decision based upon it.

example below show the necessity to notify others on documents. The CUP share document may be ignored, however.
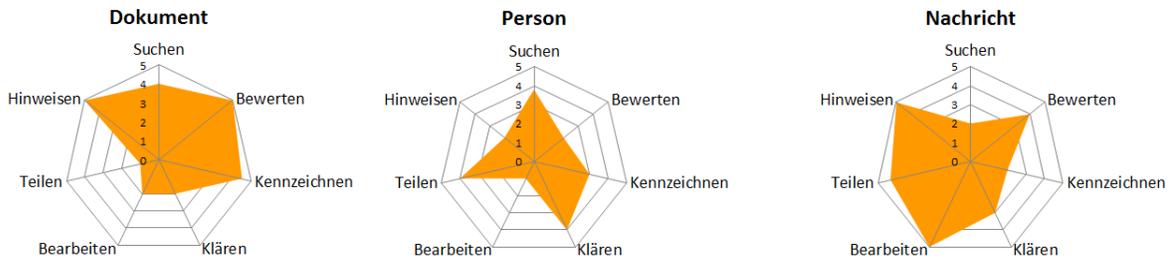

Figure 6: Instances of SpiderWebs

The following example should make this process more clear. Assume that within a business process a situation occurs repeatedly in which employees need the competency of colleagues to solve problems. The CUC "ask your colleagues" can be developed from this situation, as described in Table 4. Various CUPs can be derived from this situation. The process for "ask your colleagues" could take place as follows:

Exemplary process	CUPs
Employee asks a question to (all other) colleagues	• Sharing of a message
Colleagues answer the question	• Sharing of a message • Notify on a message • Sharing of a document • Notify on a document
The author of the question names the "most helpful answer" or the "correct answer"	• Rating of a message • Rating of a document
Frequently asked questions can possibly be highlighted in a prominent location	• Collecting of messages • Tagging messages

Table 4: CUPs for the CUC "Ask your colleagues"

Platform evaluation

The basic idea of the platform evaluation is to create a profile on the basis of the CUPs for each platform, thus avoiding having to inspect each individual function. For the classification of platforms already existing in companies or available on the market, an evaluation template has been developed, wherewith the type and extent of the 21 CUPs in the different platforms can be evaluated.

The basic hypothesis for the creation of the templates was, that the CUPs should be covered as broadly as possible within the platform. This means the more extensive the observed functions are regarding a specific CUP, the better. The template is aligned to applications available and established in the market. Platforms considered in the evaluation should be checked regularly for the support of the CUPs. Should the CUPs generally be supported very broadly in different platforms, the rating scheme has to be readjusted accordingly.

12

In the template, all distinct support or realization possibilities of a CUP through a platform are checked. These possibilities are then marked as realized (green), partly realized (yellow), or not realized (red), rated accordingly, and described qualitatively. This description can play a decisive role, in later discussions due to tight results. The template describes precisely which requirements are absolutely necessary for a green or a yellow rating. By using the template, the realization of the CUP is quantified and normed to a scale from 1 through 5. In Figure 7 the CUPs named earlier (Table 1) are rated.

CUPs	Realization	Evaluation Standard	Platform implementation
Search a document	Third Party support	Ask selected persons and receive answers: green: as attachment yellow: as a link	(green) Ask a question to a single person or a community. Get answer via comment (with attachment or links to already existing content)
	Explorative Browsing	Display of a complete document list green: pageing through the list yellow: seperate opening of each document	(yellow) Listing all available files within a community, no page through
Search a person	Name	Searching via a search form via a person name green: Searching via fragments yellow: searching only via full name	(green) Search field above the site, Fragment: Start of forename or surname
	Rating	... persons based on ratings. green: filter ... yellow: sort...	(red) Not possible
	Label	... persons based on labels. green: filter ... yellow: sort...	(green) Only within the member view of a community, filtering via tags in the right column
	Third Party support	Ask selected persons and receive answers: green: @-mention of a person yellow: link to the person profile or vcard	(yellow) Ask a question to a single person or a community. Get answer via comment (with attachment or links to already existing content)

Figure 7: Excerpt from a rating template

Comparison: Requirements and platform profiles

The comparison of requirements profiles and platform profiles shows how well each platform meets the priorities derived from the requirements analysis.

In Figure 8 an exemplary comparison of the requirements and a platform represented by an evaluation template shows directly the suitability of a platform. In this example, the platform is only partly suitable for supporting collaboration in this specific business context, because the (highly prioritized) sharing and editing of messages is only supported in a limited manner by the evaluated platform (comparison of the blue and the orange areas in the SpiderWeb "message").
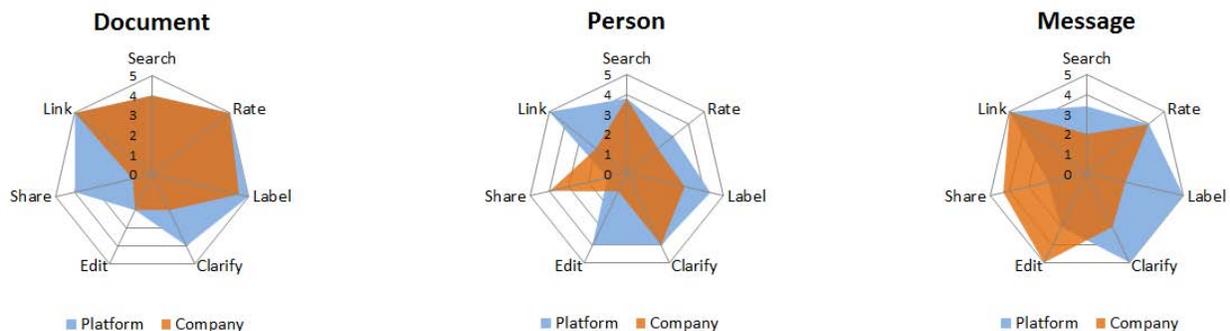

Figure 8: Matching between requirement profile and evaluation template

SUPPORTING THE IMPLEMENTATION WITH THE aperto FRAMEWORK

Due to the previously mentioned Nutzungsoffenheit, corporate social software does not determine usage cases for the users, but instead offers them lots of space to discover their own possibilities of use. While this provides the opportunity for each user to utilize the platform according to his own work practices, this freedom also bears the risk that the user might not seize the potential or the benefit of the platform[15].

For this reason, the adoption process has to be supported by pointing out practical possibilities of use, especially in the introductory phase (Figure 1). These practical possibilities of use should be understandable and helpful for the work of the addressed users. This can be an iterative process in which the users adopt the services in the course of their work practices, and management or employees identify yet unknown usage possibilities step by step[16]. Especially here it is relevant not to step back on the level of the functions, because the users rarely perceive the value of individual functions, but rather in the collaborative processes supported by the platform. Now the potential of the usage-oriented documentation comes into play, which should be elaborated in the course of the change management process. This helps the user to adopt the software by describing possible usage cases.

Different possibilities for a usage-oriented document are:
- Reports about usage cases of the platform (in on- or offline publications). Word of mouth advertising or viral marketing as well as the credibility and authenticity of the promoters and key users play an important role here.
- Collection of particular usage examples in the form of reports in which users describe their success stories with the platform, e.g. in articles in an employee newspaper, as part of the online documentation, in user blogs, or as a simple post of a user labeled with a hashtag (e.g. #bestpractice).

Both possibilities can be realized in text, as a podcast, or in the form of scenario posters/comics (see Figure 9).

Figure 9: Visualization of the Collaborative Use Case "ask your colleagues" in the form of a comic-strip

---

[15] Another challenge is seen in the selection of a suitable platform. This problem of the so-called media choice can at least be partly solved by communicating suitable Collaborative Use Cases.

[16] In the majority of the observed cases, usage scenarios occurred that have not been foreseen when introducing the tool.

14

In order to create the usage-oriented documentation it is helpful to be guided by other successful introduction projects. The CUC-catalog (or similar collections) mentioned in Chapter 2.4 can help users discover the potential of the platform and usage possibilities of interest to them.

OPTIMIZATION OF THE PLATFORM (USAGE) WITH THE aperto FRAMEWORK

Another possibility to use the aperto framework is to support optimizing the (usage of the) platform. The success measurement offers a company the possibility to become aware of the value of the deployed platform.

Different stakeholders of success measurement

The reasons to measure the success of a platform can be very different. Examples are:
1. The management or the sponsor of an investment is asking for a justification for the investment. The improvement in comparison to using the old system has to be assessed. In the sense of a benchmark, a comparison with metrics of other companies is possible.
2. A system stakeholder wants to improve (the usage of) the social software.
3. The added value of the social software to perform tasks should be shown to the users.

However, while many investors (see 1) are only satisfied with reliable numbers (how much do I save by using the platform?), system stakeholders and users are more interested in how the platform is used or can be used. In both situations, the step back to Level 5 on the aperto five-level model — measuring the frequency of the use of a function alone — is not sufficient.

It becomes obvious that the kind of success measurement and the success criteria depend on different factors. Furthermore, the previously described Nutzungsoffenheit poses a challenge for the measurement. How can the success of a platform be measured if it is unknown how it is used? The aperto framework offers an approach that enables the goal-oriented success measurement of social software in the context of the specific usage, which is based on the level of the CUC (in the following shown again by the example CUC "ask your colleagues"). Hereby, the success metrics are structured into three dimensions in accordance with (Cooper et al., 2010) that should be considered within every CUC: activity, personal benefit, business value. The consideration of the CUCs allows gathering metrics across companies which enables a benchmarking, even if the technological implementation of the CUCs is different.

Activities

In the first step it is useful to confirm the activities of the users through system-based data. Orientation is provided again by the CUP-Matrix. Build upon this, the results can be improved by a qualitative analysis (content analysis). In this way, the sharing of a message in the CUC "ask your colleagues" can be formulated and measured as a concrete number of asked questions. Such a metric always depends on the CUC and shows the extent of its usage.

Personal benefit

The next step is about assessing how the platform increases the performance of a user or yields a benefit for him in the medium term by using specific metrics (e.g. "due to the usage additional knowledge is acquired"). This can happen quantitatively if the frequencies of specific

activities are set in relation to each another. An example of such aggregated metric for the CUC "ask your colleagues" could be ratio of asked questions to (correctly) answered questions. Through this, the general problem-solving competency of the community is shown. The personal benefit, however, can be determined qualitatively as well by directly interviewing the users regarding their experiences and their observations. In the example "ask your colleagues", the perceived improved problem-solving competency (due to the fast access to helpful answers) could be such a metric. This provides insights into how useful the platform is perceived to be by the users. Furthermore, a qualitative content analysis can provide insights into how the platform is used.

Benefit for the corporation

In the third step the long-term added value of the social software for the business processes is determined regarding the defined aims of a CUC by interviewing the management or a combination of qualitative and quantitative analysis. The management performs a rating of the usage in the context of the company. The aim of this is to capture the added value of the corporate social software for the company in the form of a return on investment or through key performance or a key value indicator. Examples of metrics for the Use Case "ask your colleagues" would be the perceived time saving compared to the previous problem solving process (regarding the aim "fast problem solving"), as well as number of linked questions (regarding the aim "reducing redundant questions"). At this point it has to be noted that it is also possible to measure, for example, process runtimes or customer satisfaction, however it is not possible to affiliate those measures exclusively to the deployment of the social software (as a variety of other factors may have changed these measures in the meantime besides the software). The concept of success measurement in the course of Collaborative Use Cases is summarized in Table 5. The comparison shows that each level yields specific strengths and weaknesses. For this reason, a combination of all metrics as described above may be appropriate.

Dimensions	Activity	Personal Benefit	Benefit for the Company
Explanation	Measures that show an increase of the activity on the platform regarding a specific CUC	Measures that show that the user is supported through the platform	Measures that show the value of the usage for the business process
Example metrics	Quantitative • Number of shared messages • Number of positively rated messages Qualitative • Number of questions asked • Number of answers	Quantitative • Ratio of asked to answered questions • Number of persons that participated in the solution finding process Qualitative • Persons who had the same question and found an answer	Quantitative • Number of linked questions • Decrease of of second level support requests (resource saving) Qualitative Time elapsed for the solution finding (time saving)
Method	Rating of the usage through quantitative activity analysis	Rating through the user and qualitative content analysis and connection of multiple quantitative metrics	Rating through the management and combination of qualitative and quantitative analysis
Strengths	Clear measures	Partly clear measures	Information about the added value
Weaknesses	No information about benefits or added value	Data are based on the perception of the users (interviews) or are very complex to gather (content analysis)	Data are based on the perception of the management (interviews) or are very complex to gather (content analysis)

Table 5: Success measurement in the course of a Collaborative Use Case

CONCLUSION, PROOF OF CONCEPT, OUTLOOK

This report started by explaining why social software differs vastly from traditional business applications, which have a predetermined structure and predefined use cases from the early development stages.

With the aperto framework, consisting of the aperto five-level model, the aperto CUP-Matrix as well as the corresponding definitions and templates, a possibility was presented to completely seize the potential of corporate social software and thus make its potential benefit feasible. The aperto framework can support requirements analysis and platform evaluation and selection, as well as the introduction and the ongoing success measurement and optimization.

The described approach, was applied successfully in the past few months in multiple selection and introduction projects for social software in German companies. We are glad to see that a model developed on a scientific basis was effectively used in those projects. Meanwhile in all the companies a corporate social software solution is supporting the collaboration of employees, and we are curious to find out what results first success measurements will yield.

At the same time, we are curious which course the application of corporate social software will take in the coming years. We will observe the further development of the platforms and the exploration of new usage possibilities with great interest. We presume that the aperto framework will be able to support corporations against the background of changing collaboration structures due to its high degree of abstraction.

ACKNOWLEDGMENTS

GLOSSARY

aperto	Framework for the selection, implementation and optimization of social software in corporations, consisting of the aperto five-level model, the aperto CUP-Matrix, as well as the aperto CUP catalog and different processes which rely on these artifacts. The name of the framework is derived from the Italian word for "open" – referring to the Nutzungsoffenheit of social software.
CUC	A Collaborative Use Case (CUC) is the concrete, exemplary variation of multiple CUPs.

CUP	Collaborative Usage Patterns (CUP) describe the usage possibilities that can be covered by corporate social software and through that the IT-supported interaction between multiple persons in an abstract manner.
CUP-Matrix	The CUP-Matrix displays all possible abstract collaborative usage possibilities in the form of a matrix with the seven actions and the three objects in the X and Y coordinates.
Function Bundles	A function bundle is the application of a CUP with the help of a specific number of functions of the Enterprise 2.0 platform at hand.
Collaborative Process	Interaction between multiple persons in the course of a common activity and with the common goal to proceed in the business process.

LITERATURE

Al-Mudimigh, A., Zairi, M. & Al-Mashari, M. (2001). ERP software implementation: an integrative framework. European Journal of Information Systems, vol. 10, pp.216-26.

Back, A. & Koch, M. (2011). Broadening Participation in Knowledge Management in Enterprise 2.0. it - Information Technology, vol. 53 no. 3, 135–141.

Briggs, R. O., Kolfschoten, G., de Vreede, G.-J., Albrecht, C., Dean, D. R., & Lukosch, S. (2009). A Seven-Layer Model of Collaboration: Separation of Concerns for Designers of Collaboration Systems. In Proceedings of the International Conference on Information Systems (ICIS), Phoenix.

Büchner, T., Matthes, F., & Neubert, C. (2009). A concept and service based analysis of commercial and open source enterprise 2.0 tools. In Proceedings of the International Conference on Knowledge Management and Information Sharing, Madeira.

Buhse, W. & Stamer, S. (2008). Enterprise 2.0 - The Art of Letting Go. New York: iUniverse.

Cook, N. (2008). Enterprise 2.0 – How social software will change the future of work. Gower: Surrey.

Cooper, C. N., Martin, M., & Kiernan, T. (2010). Measuring the value of social software. IBM White Paper. Retrieved from ftp://public.dhe.ibm.com/ftp/lotusweb/services/ibm_wp_measuring-social-software_june2010.pdf.

DeLone, W. & McLean, E. R. (1992). Information Systems Success: The Quest for the Dependent Variable. Information Systems Research, vol. 3 no. 1, pp. 60.

Ehms, K. (2010). Personal Weblogs in Organisations: Toy or tool for contemporary knowledge management (PhD Thesis). Universität Augsburg.

Glaser, B. G. & Strauss, A. L. (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine Publishing Company.

Krcmar, H. (2005): Informationsmanagement. Berlin: Springer.
Koch, M. & Richter, A. (2009). Enterprise 2.0: Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen. München: Oldenbourg Wissenschaftsverlag.

McAfee, A. (2009). Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges. Boston: McGraw-Hill Professional.

Richter, A. (2010). Der Einsatz von Social Networking Services in Unternehmen - Eine explorative Analyse möglicher soziotechnischer Gestaltungsparameter und ihrer Implikationen. München: Gabler Verlag.

Richter, A. & Riemer, K. (2009). Corporate Social Networking Sites–Modes of Use and Appropriation through Co-Evolution. In Proceedings of the Australasian Conference on Information Systems, Melbourne.

Richter, A. & Stocker, A. (2011). Exploration & Promotion: Einführungsstrategien von Corporate Social Software. In Proceedings of the International Conference Wirtschaftsinformatik 2011, Zürich.

Richter, A., Mörl, S., Trier, M., & Koch, M. (2011). Anwendungsszenarien als Werkzeug zur (V) Ermittlung des Nutzens von Corporate Social Software. In Proceedings of the International Conference Wirtschaftsinformatik 2011, Zürich.

Riemer, K., Altenhofen, A., & Richter, A. (2011). What are you doing? - Enterprise microblogging as context building. In Proceedings of the European Conference on Information Systems (ECIS), Helsinki.

Riemer, K., Diederich, S., Richter, A., & Scifleet, P. (2011). Short Message Discussions: On the Conversational Nature of Microblogging in a Large Consultancy Organisation. In Proceedings of the Pacific Asia Conference on Information Systems (PACIS), Brisbane.

Riemer K. & Filius S. (2009) Contextualizing Media Choice Using Genre Analysis, Business and Information Systems Engineering - the International Journal of WIRTSCHAFTSINFORMATIK, vol.1:2, pp. 164-76

Riemer, K., Froessler, F. and Klein, S. (2007). Real time communication – modes of use in distributed teams. Proceedings of 15th European Conference on Information Systems (ECIS), St. Gallen, Switzerland.

Riemer, K., Steinfield, C. and Vogel, D. (2009). eCollaboration: On the nature and emergence of communication and collaboration technologies. Electronic Markets, vol.19, no. 1, pp. 181-88.

Riemer, K. & Richter, A. (2010). Tweet Inside: Microblogging in a Corporate Context. In Proceedings of the Bled eConference, Bled.

Shin, H. & Lee, J. (1996). A process model of application software package acquisition and implementation. Journal of Systems and Software, vol. 32 no. 1, pp.57–64.

Stocker, A.; Richter, A.; Hoefler, P.; Tochtermann, K. (2012): Exploring Appropriation of Enterprise Wikis: A Multiple-Case Study. Computer Supported Cooperative Work, vol. 21(2-3), pp. 317-356.

Trier, M. (2005). IT-Supported Visualization of Knowledge Community Structures. In Proceedings of the Hawaii International Conference on System Sciences, Kauai.

Williams, S. P. & Schubert, P. (2011). An Empirical Study of Enterprise 2 .0 in Context Research Aims and Research Design. In Proceedings of the Bled eConference, Bled.

芽|Sprouts