Supporting Global Software Development with Web 2.0 Technologies - Insights from and Empirical Study

Alexander Benlian
Ludwig-Maximilians-University, benlian@ise.tu-darmstadt.de

Thomas Hess
Ludwig-Maximilians-University, thess@bwl.lmu.de

Supporting global software development with Web2.0 technologies – Insights from an empirical study

Alexander Benlian
Ludwig-Maximilians-University
Munich School of Management
Institute for Information Systems and New Media
Ludwigstrasse 28, 80539 Munich, Germany
benlian@bwl.lmu.de

Thomas Hess
Ludwig-Maximilians-University
Munich School of Management
Institute for Information Systems and New Media
Ludwigstrasse 28, 80539 Munich, Germany
thess@bwl.lmu.de

ABSTRACT

The paper at hand presents findings of a survey- and case study based investigation of using innovative Web2.0 technologies in the software development process of global companies. While the empirical study shows that adoption rates are already quite high, the satisfaction with using Web2.0 technologies and the resulting success highly depend on the adoption speed as well as on the mastery of special operating capabilities within the local IT and product development departments. In addition to these quantitative and survey-based findings, 3 case examples shed light on specific application scenarios of Web2.0 applications in different industries. Besides the exemplary illustration of where these innovative web technologies can support global software development processes, economic and strategic benefits of introducing them are highlighted.

Keywords

Web2.0 technologies, global software development, innovation management

INTRODUCTION

The thrust to global markets requires international companies to increasingly develop (software) products for a diverse set of local markets. As a consequence, software product development teams are often spread over the entire globe in different time zones and are dependent on information and communication technologies for regular synchronization. Execution issues coming up in such decentralized operating modes are manifold (Nidumolu 1995; Karolak 1999; Herbsleb, Mockus, Finholt and Grinter 2001): In the planning stage, requirements management among decentralized teams is often complicated by a lack of communication and standardized processes leading to requirement changes later in the software cycle and higher rework rates. In the design phase, a lack of structure and standardized versioning frequently complicates the specification process among remotely working developers. Needless to say that time differences and an inability to leverage existing information (e.g., prior team websites) hamper smooth collaboration. In the build phase, code reuse in a decentralized multi-site setting, for instance, depends on a network of developers willing to share and understand code, which has often not been facilitated by adequate tool support so far. In the deployment stage, various standards in the form of different topologies across environments complicate the roll-out. To make matters worse, feedback from pilots/test groups in local regions are often not incorporated before deployment.

Underlying root causes of these issues in decentralized development settings can be tracked down to inefficiencies in providing coordination and transparency for a smooth and frictionless communication (Nidumolu 1995). This is where Web2.0-technologies may come into play, as they are promised to deliver on some of the early promises of the Internet, such as open standards and collaborative communication (Vossen and Hagemann 2007) and thus may lend themselves to play an important role in the management of global software development teams. In particular, Web2.0-technologies are said to facilitate collaborative generation of information, reuse of code and applications in the form of services, allowing providers to develop foundational applications, and fostering device independence and richer customer experiences (Musser 2006).

The paper at hand presents the findings of an empirical study on the adoption and satisfaction using Web2.0-technologies in global software development based on a survey of more than 100 companies operating in global markets. In the next section, a literature review and the methodology of this research study are presented. Then, major results are illustrated, before the findings of this research are discussed which are embedded into the current research context. The last section concludes with pointing out research shortcomings and future research steps.

LITERATURE REVIEW AND RESEARCH METHODOLOGY

Technology adoption is a topic in IS research which has been examined in many facets and with manifold research models. Drawing on classical psychological and diffusion theories (e.g., Rogers 1962), the Technology-Acceptance Model (Davis 1989) laid the conceptual foundation for the exploration of user acceptance in the IS field. Derivatives attempted to extend the explanatory power of the TAM by including additional antecedents (e.g., Burton-Jones and Hubona 2006) or by integrating it with additional dependent variables (e.g., Wixom and Todd 2005; Yi et al. 2006). One main line of reasoning in all of these studies suggests that higher user satisfaction strengthens the acceptance and adoption of existing and/or innovative technology. Web2.0 technologies, as relatively new kind of so-called social software, have already been analyzed in several adoption context scenarios (e.g., Hsu and Lin 2008; Stöckl et al. 2005). However, the fruitfulness of their use in global software development processes as management mechanisms (i.e., coordination and communication tools) has been neglected so far in academic research. Addressing this research gap in IS adoption literature, this study aims at exploring the following three research questions:

1. To what extent are Web2.0 technologies already adopted in the software development context?
2. How satisfied are companies with the return on investment for Web2.0 technologies in software development and how is this interlinked with adoption behavior?
3. In which development stages are Web2.0 technologies primarily used?

The empirical study at hand is based on a survey with quantitative and qualitative questions on the adoption and satisfaction of using Web2.0 technologies. 400 IT managers and executives of globally operating companies of the banking, insurance, telecommunication, and media industry (100 companies each) were randomly drawn from publicly available industry databases and asked by mail to respond to their past IT investments into Web2.0 technologies and their satisfaction level in the context of software (product) development. Approximately three weeks after the initial mailing, follow-up calls and electronic mailings were initiated not only to increase response rate, but also to have a discussion on specific and, most of all, successful application scenarios of Web2.0 technologies in the software development process. Three of the most impact- and insightful application scenarios have been elaborated as case examples and are outlined in the second part of the results section. The survey design is motivated by research papers on user satisfaction (DeLone and McLean 1992; Wixom and Todd 2005) which suggests that beliefs and attitudes about system usage influence the technology acceptance and adoption. Based on DeLone and McLean's and Rogers pre-works (DeLone and McLean 1992; Rogers 1962), survey items were developed to reflect the adoption (i.e., adoption typology of early adopters, early followers, and late followers) and (perceived) satisfaction of Web2.0 technologies (see survey items in the Results chapter). For a comprehensive selection of Web2.0 technology categories, this study draws on the pre-works of O'Reilly who describes 9 major technologies in the context of Web-based collaboration (O'Reilly 2005): Web services, Collective intelligence[1], Peer-to-peer networking, Social networking, RSS[2], Podcasts, Wikis, Blogs, and Mash-ups[3].

With a response rate of ~27% (n=108), a significant amount of responses could be gathered to gain insight into adoption behavior and performance effectiveness of Web2.0 technologies. Table 1 summarizes key characteristics of the contacted sample and the final survey response of this study.

[1] Collective intelligence refers to any system that attempts to tap the expertise of a group rather than an individual to make decisions. Technologies that contribute to collective intelligence include collaborative publishing and common databases for sharing knowledge.

[2] RSS (Really Simple Syndication) allows people to subscribe to online distributions of news, blogs, podcasts, or other information.

[3] Mash-ups are aggregations of content from different online sources to create a new service. An example would be a program that pulls apartment listings (e.g., www.craigslist.org) from one site and displays them on a Google map to show where the apartments are located.

Industry type	Contacted sample		Survey response	
	n	%	n	%
Banking	100	25%	28	28%
Insurance	100	25%	27	25%
Telecom-munication	100	25%	26	24%
Media	100	25%	27	25%
Total	400	100%	108	100%

Table 1. Characteristics of sample and survey response

RESULTS

In the following section, the results on the diffusion of and satisfaction with employing Web2.0 technologies are illustrated. While quantitative results from the survey are presented in the first part, in the second part qualitative details are shown in form of three case studies on the application and effectiveness of Web2.0 technologies in software development processes.

Adoption behavior of and satisfaction with Web2.0 technologies

Among the executives familiar with the nine Web2.0 technologies cited in the survey, more than 60 percent say that their companies are already investing in one or more of these trends for purposes of supporting software development processes. The most frequently cited investment is in Web Services, being used or considered to be used in the near future by 80 percent of the respondents familiar with the tools.

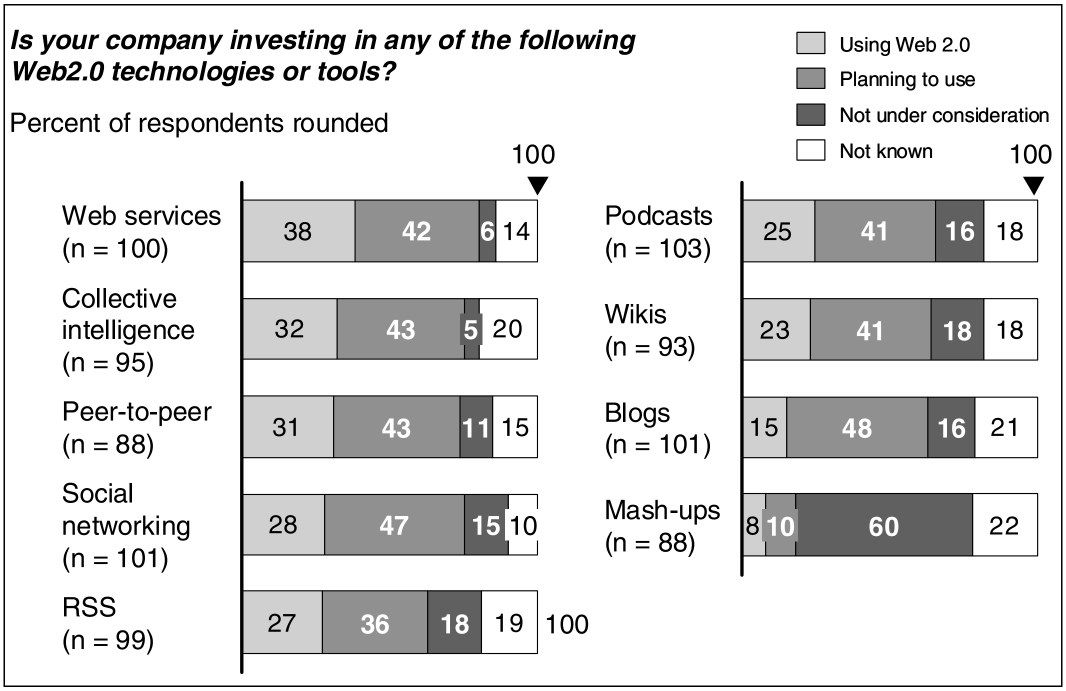


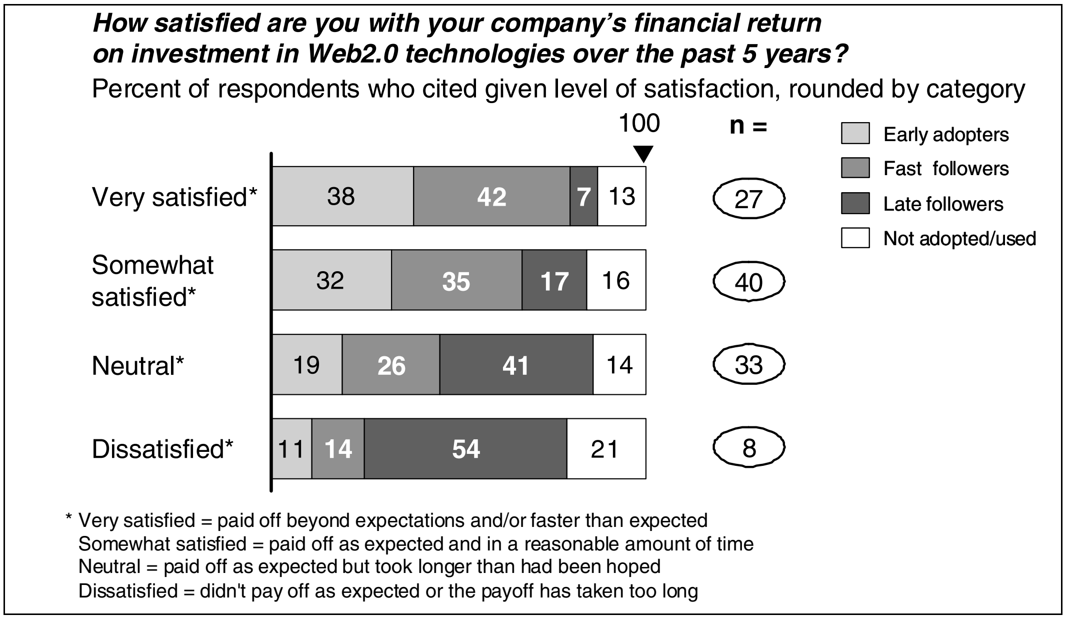
Figure 1. Adoption level of Web2.0 technologies for software product development processes

Collective intelligence applications, Peer-to-peer and social networking are also popular with around 75 percent saying they are using or considering it. Also technologies, which appeared to be less relevant in the context of software product development showed a relatively high adoption rate: Around two-thirds of the respondents answered that they use or consider using RSS, Podcasts, Wikis, and Blogs. The applicability of mash-up applications is considered rather low. Only around 8 percent said that they were using mash-ups in software development. 60 percent said that Mash-ups were not under consideration in terms of IT investments in the near future. Figure 1 summarizes the adoption level of the nine different Web2.0 technologies in the software development process.

More than half of the executives surveyed say they are pleased with the results of their investments in Internet technologies over the past 5 years, and nearly three-quarters say that their companies plan to maintain or increase investments in Web2.0 technologies in coming years.

A fourth of all those surveyed say they are very satisfied with their returns on investment, around 37 percent are somewhat satisfied, and 31 percent are rather neutral. A mere 12 percent say they are disappointed with previous investments. It is interesting to note that the respondent group that answered that they are very satisfied with the adoption of Web2.0 technologies have the highest share of early adopters (~46 percent) and those who are dissatisfied with the introduction of Web2.0 technologies have only a minor share of early adopters (~9 percent). Neutral and dissatisfied participants see themselves rather as late followers or non-adopters than on the forefront of technology adoption. Late followers make up around 40 percent of the total of neutral and even more than 50 percent of discontent respondents. As an insightful lesson learned from this analysis, companies that acted quickly in the previous wave of investment are more satisfied than late movers making the necessity to act quickly in Internet technology adoption even more compelling. With a rho=~0.334, a Spearman correlation analysis revealed a highly significant relationship between adoption speed and satisfaction (p=0.005, significant at the 0.01 level (2-tailed)). Figure 2 illustrates the main findings on the satisfaction level of companies with the adoption of Web2.0 technologies.

Asked what might have been done differently to make the previous investments in Internet technologies more effective, only 18 percent say they would not have acted differently. 42 percent say they would have strengthened their companies' internal capabilities to make the most of the market opportunity at hand. Among the 24 percent who say they would have moved faster, many describe their companies as fast followers or early adopters – a strategy consistent with the view that speed is of essence in technology investments. Only 10 percent of respondents say that they have invested at the right time but have overestimated the market potential. A mere 3 percent give the answer that they should have waited for technology to be developed further, that is, be less expensive and/or better able to meet company's needs.



How satisfied are you with your company's financial return on investment in Web2.0 technologies over the past 5 years?

Percent of respondents who cited given level of satisfaction, rounded by category

* Very satisfied = paid off beyond expectations and/or faster than expected
Somewhat satisfied = paid off as expected and in a reasonable amount of time
Neutral = paid off as expected but took longer than had been hoped
Dissatisfied = didn't pay off as expected or the payoff has taken too long

Figure 2. Level of satisfaction with Web2.0 technologies

Most Web2.0 tools are simple-to-use applications that are hosted offsite (e.g., Wikis, Blogs, and Social networking), which makes them easy to implement. Given that ease, it is not surprising that almost two thirds of the survey participants say that bottom-up grassroots efforts are the trigger to try out Web2.0 technologies in the development process as compared to formal pilot programs. One executive responded in the survey, "*These projects started at the grassroots level; however, the value was rapidly demonstrated. This led to projects being taken up by their 'natural' owners within the organization who continue to invest and develop the projects.*" Another one goes further, asserting that top-down management would have been a hindrance: "*The most effective efforts started as grassroots efforts. The role of senior management was to provide the support for this to continue and then get out of the way. Executives cannot mandate successful adoption of Web2.0 technologies. Their role is to supply permissions and resources – and set the boundaries – and then let intelligent and motivated teams run with this.*" The ease of exploring these technologies is cited as factor in helping advocates avoid typical barriers to implementation (or perhaps just inertia) by quickly pulling together prototypes. As one IT managers says, "*The ease of implementation can be used to overcome the usual resistance to trying things and taking risks. A simple working prototype and some data from the sandbox with real users is very compelling when compared to the traditional business approvals process.*"

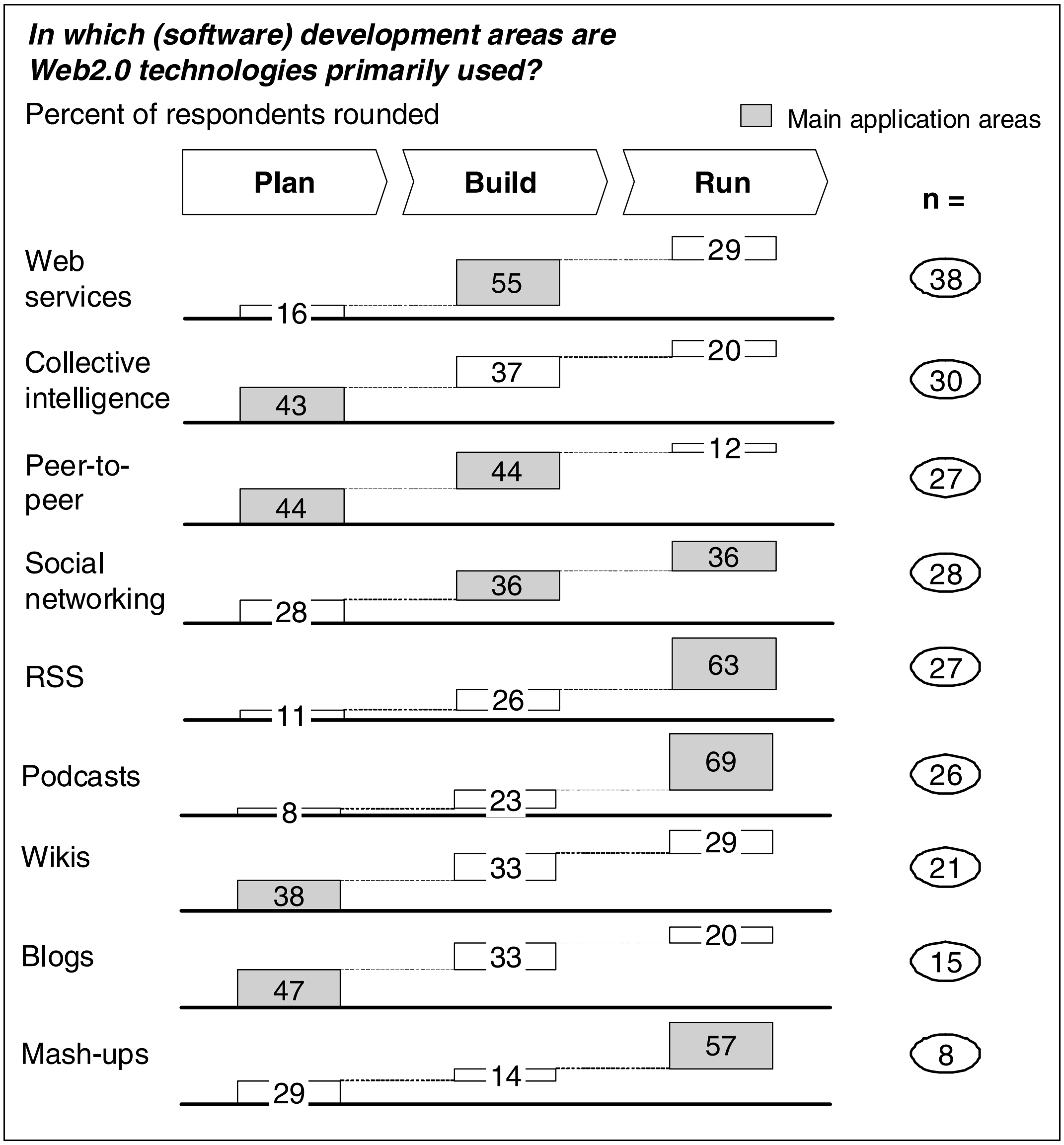


Figure 3. Main application areas of Web2.0 technologies in the software development process

When asked which stage of the software product development process (i.e., plan, build, or run stages) can be supported best by the nine Web2.0 technologies, the respondents' answers show a mixed picture (see Figure 3 on the previous page). While, according to the companies surveyed, Web Services, Peer-to-peer, and Social networking can be mainly applied in the Build phase, collective intelligence applications, Peer-to-peer networks, Wikis, and Blogs are now rather used in the Planning phase. As application scenarios in the Planning phase, companies say that scope and resource management and especially requirements gathering and specification can be supported by web-based information systems and databases which allow real-time interactions by team members. In the Build phase, executives mention that Web2.0 technologies facilitate machine-to-machine as well as human-to-human interaction and connectivity, foster modularization and active code sharing. Social networking, RSS, Mash-ups and especially Podcasts are technologies primarily used in the Run stage of the development phase where deployment, maintenance, and documentation jobs can be supported. As an example, the IT department of a global bank deploys Wikis and RSS to help developers spot spec changes. While Wikis provide a standardized template for product managers writing and updating specs, RSS messages alert and notify developers of updates and changes in real-time.

3 case examples on effective Web2.0 application scenarios in software development

In the following, 3 case examples for the successful application of Web2.0 technologies are presented covering the phases of software product development: Plan, Build, and Run.

Facebook for staffing development teams in a telco company

A global telecommunication company's IT department usually kept a central database of its developer profiles with only restricted access rights for staffing personnel. As a staffing manager complained about growing obsolete and incomplete information in the database, IT developers suggested to introduce a '*Facebook for staffing development teams*'. This Web2.0 application is based on the Peer-to-peer principle where IT developers manage their own profile data using tools typical for user generated content applications. IT developers now not only update information on their availability, skill level and industry expertise, but also add information on their track record of past projects and on team members they have already worked with (which can be crucial information for staffing next projects). Staffing personnel involvement is reduced to the approval of user-generated data and to their core responsibility: matching developer supply and demand.

As a result, turnover time for the staffing process could be cut by 33 percent. The new transparency of information also led to a more open and stimulating atmosphere among developers which were motivated to update and populate their page for the reward of recognition. Last but not least, staffing managers and also developers were much more satisfied with the matching outcomes. As anecdotal evidence, in one situation a developer from the IT department in India could contribute his specialist knowledge in a European project and transfer best-practices where it was really useful. This example illustrates that Web2.0 technologies may even contribute to global sourcing application scenarios.

Flickr for code reuse at an insurance company

In the past, a culture of software code sharing and reusing was neither nurtured by the management nor actively lived by the developers at the IT department of an insurance company in central Europe. However, as an analysis from an IT consultancy revealed, great productivity increases (i.e., it was predicted that feature points per developer per man-day could be decreased by 15 percent) in the software development department could be gained through a higher degree of resource sharing. Since existing legacy application systems in different branches of the insurance company (i.e., life and P&C) had been developed independently from one another over the last decades, looking for ways to increase the level of code exchange and reuse between branch developers seemed a promising idea.

During the IT transformation effort one initiative to increase code reuse among applications for the different insurance branches was the introduction of a Flickr-type application which made code more easily searchable and re-usable for developers. Software code which was eligible for reuse could be submitted, tagged, and uploaded to a central and easy-to-use 'Source Depot'. The developer community of the insurance company had then the opportunity to effectively search for code snippets in the Source Depot, to reuse the code, or to add input and give feedback to the author who was notified by RSS messages in the case of any updates or comments.

By allowing developers to give feedback in a developer community, the code quality increased significantly already after 3 months of the introduction of the Web2.0 application[4]. Furthermore, the amount of time to develop a new solution in one branch of the business decreased enormously by enabling developers to leverage previous solutions from other branches. Last but not least, through sharing code between branch developers, discussions and idea-generation were facilitated leading to a more cooperative and open working environment.

Digg for lead users and developers at a global banking corporation

Originally, the software development process in the IT department of a globally operating bank could be characterized by a classical iterative approach, generating new versions of IT applications with adapted features in regular cycles. One major drawback of the process, as mentioned by an IT manager, was the lack of user involvement at the beginning of new product versions. The introduction and collection of new requirements and features was limited to regional product developers in the marketing department and, to a lesser degree, to IT developers.

Triggered by a bottom-up initiative, 3 IT architects set up a Digg-like[5] prototype to illustrate how lead users of different regions could be involved into the development process of a mortgage application IT system. Convinced by the economic benefit in terms of innovation, quality, and efficiency, IT managers soon transformed the prototype initiative into a formal project with the result of a tried and tested IT application. Lead users, who were identified by the regional marketing departments, were invited to the mortgage application system in order to post comments and further ideas on new or other functionality to an open blog. Based on these suggestions, lead users could then rate the content with positive or negative votes leading to a prioritized view of users. High-priority items now pass the first gate of the development process which then undergo another round of screening by project managers (for technical feasibility), product managers (market relevance and strategic fit) and software developers (for technical feasibility).

The introduction of this Web2.0 application system helped the firm to better integrate the local needs of customers into the development process. Although not being a vehicle to save development time, the Digg tool supported the local product and software development teams to increase the fit between what was needed in the market and what was offered by the bank. As a fruitful side effect, the pace of innovation increased in a positive way and even led to entirely user-triggered application systems for better convenience in the interaction between the bank and customers.

DISCUSSION ON THEORETICAL AND PRACTICAL IMPLICATIONS

Based on the findings of the quantitative and qualitative parts of the empirical study, theoretical as well as practical implications can be drawn.

This study's theoretical frame of reference was based on and motivated by literature on user satisfaction, which has first been introduced into IS research by DeLone and McLean (DeLone and McLean 1992). Its basic hypotheses suggest that higher user satisfaction strengthen the acceptance and adoption of technology. Despite of slight adaptations to the operationalization of user satisfaction for the context of Web2.0 technologies, this theoretical relationship could be supported by this empirical study. Furthermore, it was put forth and also supported that technology acceptance is fostered by increasing adoption speed, which not only has theoretical meaning, but also holds practical relevance for technology vendors and client companies.

A practitioner (i.e., IT executives, managers, and developers) is likely to find the results of this empirical study to be of value in many respects. First, the paper could show that Web2.0 applications seem to have permeated global companies already to a substantial degree with some variances between the application types. Higher adoption rates can, however, certainly be achieved. Second, the case studies could illustrate that an adequate integration of Web2.0 applications in the (global) software development process can entail significant economic benefits ranging from optimized throughput times to higher degrees of innovation and quality. Third, our empirical analysis could make the point that early adopters are more satisfied with using Web2.0 technology than laggards. Adoption speed thus seems to play an eminent role in the effectiveness of and satisfaction with using Web2.0 technologies. Given the fact, that almost half of the surveyed companies emphasized the necessity to build and strengthen their companies' internal capabilities before and in parallel to the introduction of Web2.0 technologies, also point to a major driver for implementation success. Last but not least, Web2.0 technologies obviously open

[4] These results were the outcome of a survey conducted by the hired IT consultancy.

[5] The Web2.0 application mainly incorporated similar functions as can be found at www.digg.com, including ideas, articles, and even video snippets on application scenarios.

up business processes in every step of the value chain for suppliers, business partners, employees, and, most of all, for customers. As one IT executive noted, "*We now see customers, particularly the professionals and customer experts, as having a much greater role in the development of new products*".

CONCLUSION

With regard to the empirical study at hand, first steps have been undertaken to shed light on the relationship between Web2.0 technology adoption and satisfaction. Although a complementary set of empirical tools have been employed ranging from quantitative surveys and qualitative case studies, some methodological and technical shortcomings had to be taken into account.

On the one hand, a higher number of participating companies from a more diverse set of industries could have yielded more representative results on adoption rates as well as on the relationship between perceived satisfaction and technology adoption. In addition, more accurate variables measuring performance and success criteria of introducing Web2.0 technologies (such as time, costs, and quality) – although difficult to obtain through survey-based questionnaires – would have generated more precise results on the real benefits of Web2.0 technologies. On the other hand, as indicated through the handful of case studies collected in the course of our empirical analysis, an action research oriented methodology (Baskerville 1999) where researchers immerse into the research units being investigated, would bring additional rich and in-depth insights into important process-related aspects revolving around technology adoption in global software development.

Despite and because of the shortcomings presented above, describing and explaining the implications of innovative web-based technologies on business and economy seems to be a promising and important field in IS research. A vast array of further areas of research can be thought of.

First of all, drawing on the given findings of this study, variances between industries and countries in the adoption and satisfaction with using Web2.0 technologies in global development processes could be explored. Industries and countries which are usually further down the technology adoption S-curve may also be the front-runners in the Web2.0 case (Rogers 1962). However, with India and China having high growth rates in Internet users due to growing tech-savvy middle-income strata, the usual adoption sequence could be reversed this time. Second, additional reasons (such as functional, social, and economic motives) for adopting Web2.0 technologies in software development should be investigated in order to complete the picture of why some companies are early adopters and others just jump on the bandwagon. Third, decision processes for adoption strategies should be analyzed in a more systematic way than our empirical study did. Although having provided first indications that most Web2.0 projects are triggered by bottom-up initiatives, further research should pay attention to a more varied range of innovation strategies than just bottom-up vs. top-down. Last but not least, researching the required operating capabilities needed to be better prepared for the introduction of Web2.0 technologies would contribute a great deal not only to the implementation success, but also to a sustainable readiness of the software development staff to integrate innovations on a regular basis. To highlight just a few critical operating capabilities, Web2.0 software applications will be in perpetual beta, so that services have to be continually updated or they will cease to perform. Real-time monitoring of user behavior to see which features are used, and how they are used, would be another required core competency. And finally, gathering collective intelligence based on new data management skills and a collaborative architecture (e.g., SOA) represents another integral part of Web 2.0 in order to be able to flexibly meet customers' needs by supporting the creative reuse of features.

Large software product development projects of globally operating companies require a large scale group effort of distributed teams which usually produces higher failure rates, less quality, and lower deployment speed. Root causes of these phenomena are most often a lack of coordination, transparency, and standardization. With the empirical study at hand we could show that Web2.0 technologies may be one remedy on the way to a more seamless global technology development and usage.

REFERENCES

1. Baskerville, R. L. (1999) Investigating information systems with action research, *Communications of the AIS*, 2, 19.
2. Burton-Jones, A., and Hubona, G. (2006), The Mediation of External Variables in the Technology Acceptance Model, *Information and Management*, 43, 6, 706-717.

3. Davis, F.D. (1989), Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, *MIS Quarterly*, 13, 3, 319-339.
4. DeLone, W. H. and McLean, E. R. (1992), Information systems success: The quest for the dependent variable, *Information Systems Research*, 3, 1, 60-95.
5. Herbsleb, J. D., Mockus, A., Finholt, T. A. and Grinter, R. E. (2001) An Empirical Study of Global Software Development: Distance and Speed, *Proceedings of the 23rd International Conference on Software Engineering* (ICSE'01), May 12-19, Toronto, Canada, 81-90.
6. Hsu, C.-L., and Lin, J.C.-C. (2008), Acceptance of Blog Usage: The Roles of Technology Acceptance, Social Influence and Knowledge Sharing Motivation, *Information and Management*, 45, 1, 65-74.
7. Karolak, D.W. (1999), Global Software Development: Managing Virtual Teams and Environments, *IEEE Computer Society Press*.
8. Musser, J. (2006), Web2.0 Principles and Best Practices, *O'Reilly Media Inc.*
9. Nidumolu, S.R. (1995), The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable, *Information Systems Research*, 6, 3, 191-219.
10. O'Reilly, T. (2005), What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software, *Web publication*: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (accessed 02/24/2008).
11. Rogers, E.M. (1962), Diffusion of Innovations, (1st Edition ed.). New York: *The Free Press*.
12. Stöckl, R., Kosyak, A., von Walter, B., and Hess, T. (2006), Success Factors of Communities for User Driven Content: The Case of Ciao.Com, *Proceedings of the 12th Americas Conference on Information Systems*, Acapulco, Mexico, 4444-4452.
13. Vossen, G. and Hagemann, S. (2007), Unleashing Web 2.0: From Concepts to Creativity. *Morgan Kaufmann*.
14. Wixom, B. H. and Todd, P. A. (2005), A theoretical integration of user satisfaction and technology acceptance, *Information Systems Research*, 16, 1, 85-102.
15. Yi, M., Jackson, J., Park, J., and Probst, J. (2006), Understanding Information Technology Acceptance by Individual Professionals: Toward an Integrative View, *Information and Management*, 43, 3, 350-363.