

# A FRAMEWORK—INFORMED DISCUSSION ON SOCIAL SOFTWARE

## *Why Some Social Software Fail and Others do Not?*

Roberto Pereira, M. Cecilia C. Baranauskas

*Institute of Computing, State University of Campinas, Av. Albert Einstein N1251, Campinas, SP, Brazil*

Sergio Roberto P. da Silva

*Department of Informatics, State University of Maringá, Av. Colombo N5790, Maringá, PR, Brazil*

Keywords: Social Software, Collaborative Systems, Social Software Design.

Abstract: The possibility of developing more interactive and innovative applications led to an explosion in the amount of systems available on the web in which users interact with each other and have a primary role as producers of content—the so-called social software. Despite the popularity of such systems, few of them keep an effective participation of its users, promoting a continuous and productive interaction. This paper aims at starting a discussion about the factors that contribute for the success of certain systems in keeping their users attention while others fail. To achieve this goal, we present a discussion informed by a conceptual framework. To situate the discussion in a practical context, we illustrate with an analysis of a collaborative system for usability evaluation on the web.

## 1 INTRODUCTION

The Web 2.0 advent incited the development of new applications characterized by mass collaboration, communication and interactivity. The emergence of the “new web” encouraged the creation of technologies such as social networks, social search, social categorization (folksonomies), collaborative editing, publishing and sharing, among others (Bryant, 2006). These technologies, developed for supporting a “social web” are called social software. We can exemplify with the YouTube, the Flickr, the Orkut, the Second Life, the Delicious, the Twitter, among others, in which millions of users interact, communicate, create, share and organize information. These systems show the “power of the collective”, the opportunities and knowledge that can be generated through the collaborative work and through the mass interaction. According to Webb (2004), the goal of social software is to deal with groups, with the interaction among people. And in this context, the interaction will occur in an unprecedented scale and intensity, leading to a situation in which issues related to human-computer

interaction are extended to issues related to human-computer-human interaction.

Despite the popularity and the growing in the number of users of the social software cited above, just a small fraction of systems is really successful. Therefore, Zengestrom (2005) raises a discussion on **why some social software work and others do not**. That is, what are the factors responsible for the success of some [social] systems and for the failure or abandonment of others? To understand these factors and, then, trying to give an answer, it is necessary to consider that designing systems for the social web is a complex challenge in which several points need to be addressed.

The main particularity of social software is in the design process, because human factors and group dynamics introduce design difficulties that are not obvious without considering the human psychology and nature (Webb, 2004). Moreover, due to the recent emergence and popularization of social software it is still necessary to understand what are the impacts that this new range of applications could cause (or cause) both in the social and the technological aspects.

Despite the lack of formal metrics to determine whether a social software succeed or not, the number of users and their level of activities offer significant evidences. **If there are no users there will be nor information neither other kind of knowledge to be analyzed.** Thus, being completely dependent on their users, the success of social software depends heavily on how users feel when using them, on their interface features and on their interaction mechanisms (Pereira and Silva, 2008). Users need to feel confident, guided, rewarded and motivated to use the application because, otherwise, there is no reason for using such systems to produce or organize information or to interact with each other.

Although the concept of social software is relatively new, discussions around the design of collaborative systems are receiving attention from academy since more than two decades. In Winograd and Flores (1987), the authors discuss about the impact of computer systems on the social relations of their users, emphasizing that this impact must be taken into account when designing a system. Ackerman (2000) says that when designing a collaborative system the biggest challenge is social instead of technological. The author emphasizes that systems do not fully meet the requirements of sharing information, the social policy of groups, responsibilities, among others, because we do not have knowledge on how to develop systems that fully support the social world. Neris *et al.* (2008) point the challenge created by the users' diversity of skills, in general, saying that to address this challenge it is necessary to know users in their skills, formalizing interaction requirements and studying solutions of interface\interaction for the diversity. Systems should reflect understandings about how people actually live and work in their organizations, communities, groups and other forms of collective life. Otherwise, as Ackerman (2000) argues, produced systems will be useless, inefficiently automating and distorting the collaboration, and other social activities.

This paper sheds light on the discussion about why some systems work and others do not. Given the inherent complexity in any attempt of finding a synthesized answer, the discussion will be generated around an analysis of a collaborative system for usability evaluation on the web. This analysis considers a functional framework proposed by Smith (2007)—the **social software honeycomb**, to explain how social software works and, thus, to determine which elements should be considered when designing them. The paper is organized as follows: section 2 presents the social software framework

elements; section 3 describes the *TesteUsabilidade* system and presents an analysis about its resources and the participation of its users, discussing the elements considered by the system; section 4 revisits the framework. Finally, section 5 presents our conclusions and directions for future research.

## 2 THE SOCIAL SOFTWARE BUILDING BLOCKS

Smith (2007) proposed a framework to illustrate a list of seven elements that give a functional definition for social software (see Figure 1). These “social software building blocks” are: identity, presence, relationships, conversations, groups, reputation and sharing—an overview about the discussions that led to the honeycomb framework can be found in (Pereira, Baranauskas and Silva, 2010). Each element can be basically understood as follows:



Figure 1: Social software honeycomb (Smith, 2007).

**Identity:** a unique identifier of a user within the system. Something that represents his/her “me”. **Presence:** resources that allow knowing whether certain identity is online, sharing the same space at the same time. **Relationship:** a way to determine how users of the system can relate\are related to others. **Reputation:** a way of knowing the status of a user in the system. **Groups:** the possibility to form communities of users who share common interests, ideas or opinions. **Conversation:** resources for communication (synchronous and/or asynchronous). **Sharing:** refers to the possibility of users sharing objects that are significant, important to them.

The identity appears at the centre of the framework because, according to Smith (2007), it is the most basic requirement of any social system. One may understand from this structure that not all software has all of these elements. Actually, systems usually have three or more of such elements, but have a main focus on only one or two of them. To illustrate, consider Figure 2 which presents the elements implemented by the systems: *Youtube*,

*Delicious* and *Orkut*, which are examples of systems with a great number of active users. The dark gray hexagons correspond to the core element in which the system is focused. The light gray hexagons correspond to the other elements that are implemented by the system and which work as a complement to the core element. Those that are not explicitly considered in each system appear in blank.

Considering the *YouTube* system (Figure 2 (a)), it is possible to see that it focuses on the “sharing” element: the main purpose of users in the system is to share videos—posting and watching videos. Additionally, the system implements the elements of “identity”: each user has his/her profile with favourite videos and added videos; “conversation”: users comment and respond to comments about the videos; “groups”: the system provides resources for the formation of groups and channels in which users can join and participate; and “reputation”: the system implements a collaborative scheme of reputation over the comments posted in videos in order to identify and avoid spam.

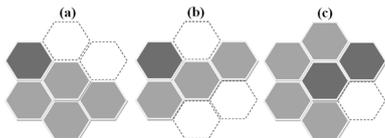


Figure 2: (a) *Youtube*, (b) *Delicious*, (c) *Orkut*.

As we mentioned earlier, the elements presented in this section are far from being exhaustive and complete. However, they are good starting points in defining a conceptual framework to assist in the understanding of social software. The framework makes it possible to examine these systems and to understand how people use them to meet their personal and social goals. Following, we describe the *TesteUsabilidade* (2007) system and present an analysis based on the social software framework.

### 3 A HONEYCOMB ANALYSIS OF THE TESTEUSABILIDADE

*TesteUsabilidade* (2007) is a collaborative system to create a space for evaluating the usability of any application or page available on the web. According to its creators, it is a collaborative system that aims at offering a free, simple, online and fast resource for the Hallway Testing (Olson, 1996), in which people are randomly allocated to test a product. Basically, the system allows its users to register their websites to be evaluated by other users, receiving

feedback, responding to comments, evaluating these comments and, consequently, also acting as evaluator into the system. Thus, the main intention is the provision of a social environment in which users help each other to improve the quality of their products, and enabling the exchange of knowledge related to the design and evaluation of websites. There are no methods or pre-defined rules to guide the evaluations. The system provides an internal page with some tips about how to evaluate; however, evaluations of websites are basically a message containing users’ perceptions, their views and considerations in unstructured natural language.

Despite being an interesting initiative, the *TesteUsabilidade* system did not succeed. In the next subsection, we present data showing the system stagnation. Like every social software which depends on users’ participation and collaboration for achieving success, something is missing to encourage the participation of users and to improve the quality of the evaluations they accomplish.

We do not intend to discuss here the advantages or disadvantages about the method applied by the system. We will focus at two main questions: How to encourage users to evaluate more? And how can websites have more chances of being evaluated? The next subsection gives us a picture of users’ participation in the system and the next one applies Smith’s framework to it.

#### 3.1 Users’ Participation

The system was available on the web in the middle of 2008. Data collected on May, 17, 2008, presented a total of 195 registered users, 153 comments and 147 websites registered to be evaluated. At October, 14, 2009, it presented 516 registered users, 386 comments and 324 websites. By that time, just 54% of the websites were commented and just 28% of the comments were replied. When comparing the data obtained from the two periods, it indicates that the data roughly doubled. However, a small portion of users is responsible for the most part of the evaluations, and a small portion of websites receives the most part of the evaluations—the long tail phenomenon (Anderson, 2006). Despite the finding of low users activity over a period of about three semesters, what allows us to say that it does not maintain a constant participation of its users, the most ten active users remained practically the same in this interval.

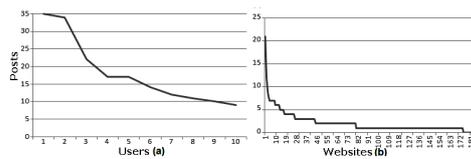


Figure 3: Long tail: (a) users; (b) websites.

The graph in Figure 3 (a) shows the curve of the number of posts created by the ten most active users in the system. We noticed that only 2% of users were responsible for exactly 47% of the evaluations. The user with the highest number of reviews conducted 36 evaluations while the user in the 10<sup>th</sup> position conducted 9; in 2008, the values were 25 and 3 evaluations, respectively—there was no change in the first seven positions of the ranking. On the other hand, the graph of Figure 3 (b) shows the concentration of evaluations in a small portion of websites, while more than a half had never been evaluated. In the graph, the “X” axis is the amount of websites, while the “Y” axis is the number of evaluations each website received. The difference in the distribution of comments by the websites may be explained by what Barabási (2003) calls “the rich get richer”: the tendency of nodes that have a high number of connections in receiving more connections. However, a forgotten website tends to stay there, until someone decides to be the first to comment on it. Nonetheless, the time waiting for an evaluation is relative and, sometimes, the website’s owner may wait months for a single feedback. When (if) this feedback comes, it can be no longer required or expected and, in those cases, a late help leads to an evaluation being wasted. Currently, the best chance that a website has for receiving its first evaluation is when it remains among the users’ most recent added websites. After that, its chances tend to decrease and it starts to count on the “lucky” to get its first evaluation.

Klamma *et al.* (2007) discuss about incentive mechanisms inspired by Social Exchange Theory. Grounded in their experiments, a feature that can help in soften the problems shown above is to allow users, when registering a website for evaluation, explicitly requesting an evaluation to some members of the community. Ex: user “A” asks user “B” to evaluate his website. Thus, there are more chances of “A” having his website evaluated and, besides, there is the possibility of “A” expressing his interest in receiving a feedback from user “B”, whether “A” considers this user as an expert or just because “A” think s/he may assist and attend to his request. Furthermore, user “B” will have reasons that encourage s/he to, at least, give an opinion about the

website. The participation of users is being driven by the request of another, which may be a factor for them to feel recognized, excited to participate more, and mainly, to make evaluations of better quality—even more if there is a ranking of the users most requested for evaluating websites. This strategy can help in two issues: encouraging the participation of users and conducting useful evaluations. We would see the elements of “conversation”, “relationships” and “reputation” present in this strategy.

### 3.2 The System Building Blocks

Perhaps, one of the most interesting (and important) contributions of *TesteUsabilidade* is allowing users to read the usability evaluations conducted by other users, generating an exchange of experience and even of knowledge. Furthermore, it is possible that the user who posts a website to be evaluated reply the comments received keeping an interaction with the evaluators. This opened mode of displaying evaluations makes it possible to see some posts in which: i) suggestions are taken into account; ii) there is an effective communication; and iii) websites are evaluated again after being restructured.

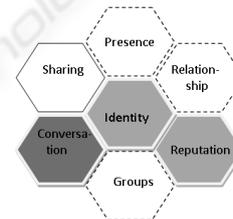


Figure 4: *TesteUsabilidade*'s elements.

Figure 4 illustrates the elements considered by the system. The focus is in the “conversation” that is implemented using the resources of comments and responses to the evaluation of websites. The element “identity” is considered as the user’s profile that exposes personal information relating to the user participation. The element of “reputation” is represented in a scheme by assigning a grade to evaluations accomplished by users and which are publicly available at their profiles. The “sharing” element must be addressed separately. The system does not allow an effective share of objects: users do not have resources to share photos, documents, etc. One can interpret that the websites, or their evaluations, are the objects being shared. However, considering the semantics of the word “sharing” (*To give part of what one has to somebody else to use or consume; To have in common*), it becomes clear that this element is not implemented by the system,

because the object remains exclusively of its owner. The other elements of “presence”, “relationship” and “groups” are not considered by the system.

Once identified the elements in the system, it is also necessary to question whether these elements were actually chosen and implemented properly to meet the goals of the system. As shown by usage data, the system has no sufficient elements for supporting users’ participation. The interaction starts around the object with focus on the element of “conversation”, but it seems that users do not feel motivated to start new interactions after a few experiences. In this case, the addition of new elements (e.g., relationships), could improve the process of interaction. Currently, the “conversation” is maintained through messages/replies in a scheme similar to a forum. The system already considers the elements of “identity” and “reputation” and it facilitates the insertion of the “relationship” element, which tends to keep users more engaged with the system and focus on the activities of other users.

The “conversation” element also should be more explored. Users do not have a resource for direct interaction with other users being restricted only to the posting of evaluation messages. If in this process users have doubts, need help or want to keep an effective and direct contact with other users, they need to look for resources external to the system. The initiative of implementing the element of “reputation”, even relevant, loses much of its impact due to the absence of other elements such as “relationships” and even “groups”: a reputation has importance as a status, and a status has value when it can be displayed, admired, used as a distinction form. But in the way it is implemented, it becomes only one indicator related to the history of ratings of each user (this example shows how the elements influence each other).

#### 4 REVISITING THE HONEYCOMB FRAMEWORK

After seeing examples of social software which got success, as well as a system that, despite considering some of the social software framework elements, not kept the participation of its users, some questions remain: What is the main difference among these systems, taking both as social software? What have they as singular characteristics that are crucial for their acceptance and for keeping the attention of their users? First of all, the framework elements are far from exhaustive and complete. Zangestrom

(2005) asserts that an important element missing in Smith’s original framework is the **Object**—the social object being built/modified is determining which elements should be considered and how they should be considered (e.g., in *Youtube* the object is videos, in *Flickr* it is photos). In Smith’s framework the “object” is not made explicit; it is behind the scene, as the thing people “share” in the social software.

It is necessary to understand what maintains the collaboration, the participation and the effective interaction among users. Knorr-Cetina (1997) addresses the individual and the object as central elements in a process of social interaction (an object-centred sociality), discussing objects around which the discussions occur, the focus is maintained, among other social interactions. In this context, we can say that systems should have a well defined object of interaction, since these objects are generally complex, open and questionable. They are in constant process of being materially defined, constantly acquiring new properties and modifying the existing ones. Besides, it should be a symmetric relation in the sharing of the object regarding who gets the benefits of the task. In the *TesteUsabilidade* this relation is asymmetric as the site owner is the person who gets the most benefit of the participation (not the person doing the evaluation task). With these assumptions a new question arises: how the elements of the framework are implemented in a system so that the object of interaction is put into focus, generating discussions, acquiring new properties, being challenging and motivating?

When designing social software, depending on the combination and the focus given to each element, the environment can be quite completely different impacting, consequently, in how it will be understood by its users. The question here is to choose the right elements and the right way for implementing them. To combine and implement the framework elements we need a socio-technical approach, or what Baranauskas (2009) defines as Socially Aware Computing: “*the theory, artifacts and methods we need to articulate to actually make the design socially responsible, participatory and universal as process and product*”.

In summary, to design a system that can be accessed by everyone, keeps its users attention and produces useful contents, we need a new Science of Design, aligning system development with social practices with the end user. Further work in revisiting the honeycomb framework is now being conducted inspired by Organizational Semiotics (Baranauskas and Bonacin, 2008).

## 5 CONCLUSIONS

The process of designing social software is highly complex because we must consider human factors, group dynamics, social and psychological aspects to understand how to design a system that effectively satisfy the needs of their users and that really meet the demands imposed by the “social”. In this paper we aimed at putting focus on possible factors that influence the success (or failure) of social software systems which depend on the effective participation of users. The paper discussed elements necessary for the functioning of social software and briefly suggested theories that can guide and give basis for understanding how to design successful systems. We observed that the object of interaction and the symmetry of interests in the shared object are decisive factors in determining how a system should be designed, which elements should be considered and how these elements should be implemented to provide an effective, productive and continuous interaction. Moreover, the process of designing social software needs to address the view of a Socially Aware Computing, otherwise, it seems to be impossible the development of systems that completely satisfies the users requirements, needs, and expectations to fulfil social demands.

In the same way as the discussion on the shared object need to be expanded, the discussion presented in this paper is a just starting point in defining a conceptual framework to help in the understanding and design of social software. With this framework, it is possible to analyze these systems and to infer why people use (or do not use) them to meet their personal and social goals. The next steps of our approach involve expanding the framework proposed by Smith (2007) where some elements should be added and other must be reconsidered and strongly discussed (Object, Awareness, Emotional and Affective aspects, Personalisation to name a few others). Additionally, discussion around the theories we mentioned in this paper (Socially Aware Computing, Organizational Semiotics and Object-Centred Sociality) are being considered as theoretical referential.

## ACKNOWLEDGEMENTS

This work is partially funded by Microsoft Research – FAPESP Institute for IT Research and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

## REFERENCES

- Ackerman, M., 2000. The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Human-Computer Interaction*, v.15, n.2.
- Anderson C., 2006. *The Long tail*. New York. Hyperion Books, 1<sup>st</sup> Ed.
- Barabási, A., 2003. *Linked: How Everything Is Connected to Everything Else*. Plume. 1<sup>st</sup> Ed.
- Baranauskas, M.C.C., Socially Aware Computing. In *ICECE 2009 VI International Conference on Engineering and Computer Education*. pp.1-5. 2009.
- Baranauskas, M. C. C. ; Bonacin, R., 2008. Design-Indicating Through Signs. *Design Issues*,v.24,p.30-45.
- Bryant, T., 2006. *Social Software in Academia*. Boulder, Educause Quarterly, Vol. II.
- Klamma, R., Chatti, M., Duval, E., Hummel, H., Thora, E. Social Software for Life-long Learning. *Educational Technology & Society*. Vol 10. pp. 72-83. 2007.
- Knorr-Cetina, K., 1997. Sociality with Objects: Social Relations in Postsocial Knowledge Societies. *Theory, Culture & Society*. Vol 14(4) pp. 01-30.
- Liu, K. 2000. *Semiotics in Information Systems Engineering*. Cambridge University Press.
- Neris, V.P.A., Martins, M.C., Prado, M.E.B., Hayashi, E.C.S. and Baranauskas, M.C.C. Design de Interfaces para Todos - Demandas da Diversidade Cultural e Social. In *XXXV Semish/XXVIII Congresso da Sociedade Brasileira de Computação (CSBC)*. pp.76-90. 2008.
- Olson, J.S., Moran, T.P., 1996. Mapping the method muddle: Guidance in using methods for user interface design. In *HCI Design: Success Stories, Emerging Methods and Real-World Context*.
- Pereira, R. and Silva, S.R.P. 2008. Aspectos da Interação Humano-Computador na Web Social. in *VIII Simpósio Brasileiro de Fatores Humanos Sistemas Computacionais (IHC'08)*. ACM Vol.378. pp.350-351.
- Pereira, R., Baranauskas, M.C.C. and Silva, S.R.P. 2010. Social Software Building Blocks: Revisiting the Honeycomb Framework. In *i-Society 2010. (to be published at IEEEExplore)*.
- Smith, G., 2007. Social Software Building Blocks. <http://inform.ca/publications/social-software-building-block>. Access Jan,10,10..
- TesteUsabilidade. 2007. [www.testeusabilidade.com.br](http://www.testeusabilidade.com.br).
- Webb, M., 2004. On social software. Available in: [http://interconnected.org/home/2004/04/28/on\\_social\\_software](http://interconnected.org/home/2004/04/28/on_social_software). Access Jan,10,10.
- Winograd, T., Flores, F., 1987. *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley Longman Publishing Co., Boston.
- Zengestrom, J., 2005. Why some social network services work and others don't. [http://zengestrom.com/blog/2005/04/why\\_some\\_social.html](http://zengestrom.com/blog/2005/04/why_some_social.html). Access Jan,10,10.