3 Social Software in the Enterprise

Thanks to the efforts of Andrew McAfee in leading the charge, the application of social software in the enterprise is now widely referred to as Enterprise 2.0. As with all new terms, there have been many attempts to define it and many disagreements along the way (most notably a public spat with Wikipedia editors who refused to accept that the term warranted its own entry in the online encyclopaedia). Individual commentators have understandably looked at the issue from their own perspectives, preferring to explain it within the context of their expertise.

M.R. Rangaswami (2006) of investment and management firm Sand Hill Group sums up the wider themes:

*Enterprise 2.0 is more than just Web 2.0 for business. Enterprise computing is far more complex than personal computing. It includes legacy environments, innumerable vendors, mismatched data sources, stringent regulations and far flung users. While Web 2.0 can deliver genuine advantages for both business users and consumers, the real ‘Enterprise 2.0’ will encompass a far broader and more complex vision.*

*Enterprise 2.0 is the synergy of a new set of technologies, development models and delivery methods that are used to develop business software and deliver it to users.*

He identifies the key characteristics of Enterprise 2.0 software as being flexibility, simplicity and lightweight, created using a combination of technologies and development and delivery models, quite literally painting the big picture (see Figure 3.1).

In this model, Rangaswami suggests that there are three essential elements that sit between the developers of enterprise social software (whether vendors, internal IT departments, line-of-business units or service providers) and the end users: technology; development models; and delivery methods. With multiple options within each, there are also multiple routes that can be chosen to connect creator to consumer.
More than just Web 2.0, Enterprise 2.0 demands lightweight software that is easy to adopt, use and integrate and can be created and delivered using a variety of technologies and models.

In addition to the debates over defining Enterprise 2.0, there have also been attempts to create classification models for it. The two most widely known are the SLATES and FLATNESSES mnemonics, proposed by McAfee and Dion Hinchcliffe respectively, both briefly summarized here.

**SLATES**

SLATES (search, links, authoring, tags, extensions and signals) is the acronym created by Harvard Business School associate professor Andrew McAfee to provide a framework around the use of social software in the business context. According to McAfee, each of the six components of the SLATES acronym provides an essential component of Enterprise 2.0.

**Search**

Knowledge workers must be able to find what they are looking for, not only through the use of page layout and navigation but more importantly by using keywords. In a survey by Forrester Research (Morris 2005), only 44 per cent of respondents said that it was easy to find what they were looking for on their intranet.

**Links**

Links provide guidance to knowledge workers about what is valuable and are one of the key indicators that search engines use to assess the importance of content in order to deliver accurate and relevant results. On the Internet this works well, but on most intranets linking is often restricted to a relatively
small group of knowledge managers or editors. By giving the entire company the ability to create links, the value of search increases.

**Authoring**

People have a desire to author, whether it be original thought, experience, a comment, a link and so on. When knowledge workers are given the tools to author information, the intranet changes from being created by a few to become a living body of linked, collective knowledge.

**Tags**

The Forrester Research survey (Morris 2005) showed that what knowledge workers wanted most, after better search results, was better categorization of content. Information architects often painstakingly create these *taxonomies* in an attempt to organize information by meaning; but by allowing employees to attach *tags* (one-word keywords) to the information they create and find valuable, *folksonomies* emerge based on actual practice.

**Extensions (or extrapolation)**

The use of tags – combined with authoring and linking – allows knowledge systems to identify patterns and use these as extensions to information and relationships. For example, if 50 employees assign the same tag to different pieces of information, not only can that tag serve as a method of linking, but also as a method of valuing those contributions. These tags can extrapolate meaning and imply relationships across different departments and time zones, even when those people have never worked together before.

**Signals**

In the age of information overload, you may be thinking that by allowing knowledge workers to create even more information, links and tags, the problem of information overload will quickly be exacerbated. That’s where signals come in, alerting users when new information of interest is created. Whilst these can be email alerts, technologies such as really simple syndication (RSS) allow employees to consume information in a much more efficient and controlled manner.

**FLATNESSES**

In October 2007, Dion Hinchcliffe augmented the SLATES components using the longer acronym of FLATNESSES (freeform, links, authorship, tagging, network-oriented, extensions, search, social, emergence and signals), having identified seven lessons learned from watching Enterprise 2.0 in action:

1. Enterprise 2.0 is going to happen in your organization whether you like it or not;
2. effective Enterprise 2.0 seems to involve more than just blogs and wikis;

3. Enterprise 2.0 is more a state of mind than a product you can purchase;

4. most businesses still need to educate their workers on the techniques and best practices of Enterprise 2.0 and social media;

5. the benefits of Enterprise 2.0 can be dramatic, but only build steadily over time;

6. Enterprise 2.0 doesn’t seem to put older IT systems out of business;

7. your organization will begin to change in new ways because of Enterprise 2.0. Be ready.

He argues that McAfee’s SLATES acronym fails to capture the essential social, emergent and freeform aspects of Enterprise 2.0, and presents FLATNESSES as a more refined conception of Enterprise 2.0 (see Figure 3.2 above).

Both of these mnemonics are useful in understanding some of the key characteristics of Enterprise 2.0, but for the uninitiated they can be somewhat daunting. They focus on the theoretical as opposed to the practical aspects of introducing social software into the enterprise. So, while they are essential in understanding the context, I feel a more pragmatic approach is required.

**INTRODUCING THE 4CS APPROACH**

For the purposes of categorizing the social software tools that will be covered in this book, I propose a more simple four-category classification model,
focused on the action involved rather than components or characteristics, which will be hereon referred to as the 4Cs approach:

1. communication;
2. cooperation;
3. collaboration;
4. connection.

*Communication:* communication platforms are those that allow people to converse with others, either by text, image, voice or video, or a combination of these.

*Cooperation:* sharing software enables people to share content with others in structured and unstructured ways.

*Collaboration:* collaboration tools encourage people to collaborate with each other on particular problems, directly and indirectly in both central and distributed ways.

*Connection:* networking technologies make it possible for people to make connections with and between both content and other people.

There is clearly some overlap between these categories, most notably in the case of cooperation and collaboration (Figure 2.2 in Chapter 2, Schopieray's Venn diagram, goes some way towards explaining this distinction). One can surmise that cooperation focuses on helping individuals work towards a common product where the knowledge gained from the process is not the goal, whereas collaboration is focused on the knowledge gained from the process of constructing something. Even so, both share the objective of enabling a group of individuals to produce something better than that which they could have produced alone.

In the context of social software, collaboration and connection require more formality than communications and cooperation, mainly because they depend on people to do things in a relatively structured manner. Likewise, collaboration and cooperation often require a higher level of interaction than connection and communication, because of the inherent focus on groups rather than individuals. These relationships can be visualized easily (see Figure 3.3) and should be considered within the context of the appropriate corporate culture when prioritizing the introduction of different forms of social software into an organization.

For example, a company with predominantly formal organizational structures and a culture of group interaction will benefit most from social software that enables collaboration. Conversely, an organization with an informal structure and a culture that rewards individual effort may prefer to invest in social
software to support communication. This framework can help any company decide where to focus their time and effort for most benefit, rather than being led by vendors trying to sell their blog/wiki/social networking solution without any understanding of the organizational structure or culture into which it will be introduced.

This approach can also be used to support organizational change. For example, if a company is trying to encourage a shift from individual effort to group problem solving, but within the confines of a relatively informal culture, then it should focus on cooperative social software that requires more interaction. Using this approach, it is possible to identify the preferred social software footprint for any organization. The examples below show the social software footprints for three different organizations (see Figure 3.4):

1. very informal, collaborative culture;
2. very formal, highly collaborative culture;
3. informal and formal, more focus on individual effort but some group problem solving.

The next step is to overlay some of the specific tools and technologies currently available – and any more that might emerge in the future – onto this matrix in order to map them directly to organizational culture (see Figure 3.5).
Figure 3.4 The social software footprints for three types of organization

Figure 3.5 The 4Cs social software technology framework
In the next section, I present some specific case studies that demonstrate how companies are using these tools to help derive specific business benefit in each of the 4Cs categories.