

Enterprise 2.0

How Social Software Will
Change the Future of Work

NIALL COOK

GOWER

2 The Birth of Social Software

WHY CONSIDER SOCIAL SOFTWARE?

17

As I have begun to demonstrate, social software has covered our personal lives like a digital rash, fuelling huge growth in collaborative authoring platforms such as blogs, wikis and podcasts and massive expansion in social networking communities. The rise of new online businesses such as MySpace, YouTube, Digg, Delicious, Socialtext, Livejournal, Typepad and Friendster are testament to this. They embody what is being referred to as Web 2.0 and is attracting serious attention from consumers, the media, big business, and venture capitalists.

What they all have in common is the ability to facilitate interactions and conversations between people, resulting in the formation of networks made up of digital relationships. But there's even more to it than that. After all, we have been able to interact with other people online and create digital relationships for years. Now, however, we find ourselves in the midst of a 'perfect storm' where almost all the technological and economic barriers to entry have been removed, with a single focus on people. People connecting with each other; people sharing information in their own voices; people controlling what goes into the knowledge space. In other words, bottom-up rather than top-down. As consultant and blogger Stowe Boyd (2006) puts it:

The answer to nearly all 'why now?' questions is technology and money, and that is true here. The availability of low-cost, high bandwidth tools like blogs or systems like Ryze, when coupled with the critical mass of millions of self-motivated, gregarious and eager users of the Internet, means social software is certain to make it onto 'the next big thing' list. Investment groups are eager to find a successful business model in social software, and I am certain that there are many to be discovered in each of the three key areas that define social software.

Despite the wet blankets and the naysayers, we are witnessing the appearance of a new crop of inductive, bottom-up social software that lets individuals

network in what may appear to be crude approximations of meatworld [sic] social systems, but which actually are a better way to form groups and work them.

Perhaps just as interesting as the way that social software is transforming group interaction – across different time zones or in the same room – social software is destined to have a huge impact on how businesses get at their markets. So the essential elements of social software will be incorporated into more conventional software solutions, changing the way collaboration and communication is managed within and across businesses, and ultimately transforming how companies sell and interact with customers.

INTRODUCING SOCIAL SOFTWARE MEANS CEDING CONTROL

18

The Cluetrain Manifesto, introduced in the previous chapter, predicted the downfall of traditional one-to-many marketing techniques in the age of the Internet. Perhaps less well-known are the insights into the changes that might take place in the workplace:

- companies make a religion of security, but this is largely a red herring. Most are protecting less against competitors than against their own market and workforce;
- as with networked markets, people are also talking to each other directly inside the company – and not just about rules and regulations, boardroom directives or bottom lines;
- such conversations are taking place today on corporate intranets. But only when the conditions are right;
- companies typically install intranets top-down to distribute HR policies and other corporate information that workers are doing their best to ignore;
- intranets naturally tend to route around boredom. The best are built bottom-up by engaged individuals cooperating to construct something far more valuable: an intranetworked corporate conversation;
- a healthy intranet organizes workers in many meanings of the word. Its effect is more radical than the agenda of any union;
- while this scares companies witless they also depend heavily on open intranets to generate and share critical knowledge. They need to resist the urge to ‘improve’ or control these networked conversations;
- when corporate intranets are not constrained by fear and legalistic rules, the type of conversation they encourage sounds remarkably like the conversation of the networked marketplace.

In fact, *The Cluetrain Manifesto's* central thesis when it comes to the workplace is that when networked markets meet networked workers, a completely new conversation begins to take place between and among markets and employees. Not only that, but the authors argue that this conversation can make both parties smarter and enable them both to discover their human voices.

This may be a stretch for some. After all, every human being already has a human voice. People don't need a conversation on the Internet in order to discover theirs. In reality, it's actually more about *humanizing* the company. The Internet – and social media specifically – is enabling companies to show that they are made up primarily of people, each of whom have thoughts, views, opinions and expertise to share, and are able to engage in a dialogue just like any other person. The result is the 'humanizing' of companies previously seen as powerful monolithic entities interested only in profit.

In the same way that the Internet is shifting the balance of power from the publisher of information to the consumer, so too will employees come to demand and expect their employers to engage them in the decision-making process, rather than attempt to coerce them into implementing the decisions made by a select few – regardless of hierarchy. This is perfectly illustrated by change management consultant John Smythe (2007) in *The CEO: Chief Engagement Officer*.

Smythe argues that employee engagement is significantly driven by the degree to which people are usefully included in the decision-making process, both on a day-to-day basis and for highly strategic change, crisis and transformation. Employees do not want to work in command and control organizations, but want and expect well-governed inclusivity from their employers. Social software reinforces a culture that encourages them to participate appropriately and creatively in the decision making which affects their work.

Case study: Humanizing Microsoft

Microsoft is arguably one of the most powerful – and some say monopolistic – companies in the world. Constant criticism levelled at the organization provides it with a negative perception amongst many of its stakeholders. In May 2003, Microsoft hired Robert Scoble from NEC. He had been blogging for over two years already, often about Microsoft. Announcing the appointment on his blog (naturally), Scoble (2003) acknowledged the role that blogging had played in the move:

For one, it helped get me noticed. For two, it helped people inside Microsoft see how I thought without needing me to come up for an interview. For three, during the interviews, we were able to really get to the point of things, since they already knew my strengths and weaknesses.

Over the next few years, until his departure in June 2006, Scoble proceeded to put a human face on Microsoft. Some say he did more to humanize the company than any of its corporate marketing or communications campaigns, at least amongst the huge audience of software developers that Microsoft relies on so much.

In October of the same year, blogger and cartoonist Hugh MacLeod (2006) designed a poster 'for my buddies over at Microsoft', calling it the 'Blue Monster' (see Figure 2.1).

According to MacLeod, the Blue Monster was designed as a conversation starter, following a conversation of his own with Microsoft's Steve Clayton. They wanted to find a way to shift the balance of power of Microsoft's communication from external negativity (media, competitors and detractors) to internal positivity (Microsoft's own employees). The cartoon was intended to stir up conversations about Microsoft of a different kind to the norm, resulting in internal conversations from the outside in. That is, not as part of a formal employee engagement or marketing communications programme, but through what *eWeek's* Joe Wilcox called 'home-brewed, moonshine marketing', where a Microsoft supporter designs a poster and uses his blog to encourage any Microsoft employee to download and use it to start conversations that might help them tell their side of the story.

Its overall impact is debatable (one only sees a small handful of examples of usage, mainly via MacLeod and Clayton's own blogs), although I doubt mass domination was the primary objective. MacLeod's aim was to give



Figure 2.1 Hugh MacLeod's 'Blue Monster' poster

© Hugh MacLeod, <http://www.gapingvoid.com>

Microsoft employees what he calls a ‘social object’, a device designed to generate conversation, and this strategy appears to have worked. At the time of writing (just over a year after its launch), there are 57,500 results on Google mentioning Microsoft and the Blue Monster, over 1,200 friends of the Blue Monster group that MacLeod and Clayton set up on Facebook, and – after the Blue Monster was put on a bottle of Stormhoek wine – a half-page feature in the *Financial Times*.

WHAT SOCIAL SOFTWARE LOOKS LIKE

In Chapter 1 I suggested a classification for social software – the 4Cs – that encompasses four broad areas – communication, cooperation, collaboration and connecting. Within these categories fall tools such as blogs, wikis, social bookmarking, tagging and really simple syndication (RSS), each with their own unique history and features. In the consumer space they are distributed along the adoption curve but when it comes to their use in business they have hardly featured, even though they offer clear benefits. It is worth briefly describing here the features of the main technologies, as we will revisit them throughout the rest of the book:

Blogs: Blogs, or weblogs, are websites containing individual articles (posts) usually presented in reverse chronological order. Each post has its own URL (called a permalink) which makes it easy to find even after it has moved from the front page to an archive. In addition, many blogs encourage comments and trackbacks (comments in the form of a post made on the commenter’s blog, linking to the source post), and enable visitors to subscribe to updates using RSS (see below).

Wikis: A wiki is a website where the pages can be created, edited and linked collaboratively by anyone who has access to do so. In most cases a history of every change is retained so that all edits can be tracked and a previous version can be reinstated. The most popular publicly available wiki is Wikipedia (www.wikipedia.org) which, as of November 2007, contained over 9 million pages in 250-plus languages, edited by more than 75,000 active contributors. Again, many systems also allow commenting and allow visitors to subscribe via RSS to receive updates whenever pages are added or edited.

Social bookmarking: Social bookmarking is a way to store, organize, share and search bookmarks to web pages using a web service rather than the bookmarks (or favourites) functions of a web browser. Most services encourage people to ‘tag’ (see below) each bookmark with words that describe the meaning of the content, which then serve not only as an organizational structure but also as a way of connecting together bookmarks posted by different people on the same topic.

Tagging: Tagging is the process of assigning user-defined keywords to a piece of information, and can be done by the creator or the viewer of the content, depending on the system. These tags are often used to create aggregated informal classifications (or folksonomies), and as a navigation/discovery method. Tagging is widely used in blogs, wikis and social bookmarking, as well as other forms of social software.

RSS: RSS is a method of publishing frequently updated web content. Each RSS 'feed' is an XML-formatted document containing summaries or the full text of each item. When combined with an RSS aggregator or feed reading software, subscribers can automatically track a large number of websites without ever having to visit the sites again.

The benefits of these and other social software technologies are becoming increasingly clear to consumers as they move from the early adopters to the early and late majorities. Blogs make the process of publishing thought and opinion quick and inexpensive and can become highly effective income-generating media properties. Wikis allow groups and communities to collaborate in a distributed way, saving time and money. Social bookmarking lets people access links they have saved across the Internet, helping them discover related information and people more efficiently. Tagging removes the need for a single editor to work out what every piece of content means and categorize it accordingly, and in turn help consumers find the information they are looking for based on the experiences of others like them, reducing effort. Finally RSS drastically reduces the amount of time required to constantly check websites to see whether they have been updated, and reduces email overloads.

Given such benefits, it is a logical next step for each of these technologies – and many more – to be applied in an enterprise context. As we will discover, companies are using internal blogs to share knowledge and create conversation, wikis to collaboratively publish documents, enterprise bookmarking to augment existing ways of finding information and social networking tools to manage expertise and tacit knowledge. The technology enablers that accompany social software, such as tagging and RSS, deliver additional benefits to the organization and the individuals using them, connecting a disparate collection of tools together to form a coherent enterprise ecosystem.

CHALLENGES PRESENTED BY SOCIAL SOFTWARE

The physical and conceptual boundaries between the internal and external worlds are crumbling. Consumer software is finding its way into businesses, and employees are using it to talk to people inside and outside the organization, often without their employer's knowledge. That's pretty scary for many chief executives and chief information officers and could explain why the immediate reaction of most companies is to block it. But before you press the red button, let me try and convince you why that's not always the right approach.

Organizations need to accept that social software is a reality in today's always on, on demand, business and technology landscape. What's more, the desire of employees to use it can be harnessed to achieve business impact with very little investment. With it come some fundamental changes to how employees connect and communicate with each other. This is something that every company will need to understand and exploit in order to deliver the working environment that their staff expect. The biggest change will be to the enterprise itself, most evident in the way they select and purchase software, as well as who buys it. For example, if a cross-geography team wants a way to collaborate on a project there are multiple wiki services already available on the Internet – at costs that can be claimed on personal expenses. Businesses will eventually have no option but to purchase this same software centrally, if only to curb managers' spending.

It also means that non-IT staff will become increasingly interested in the software tools they are buying and using. 'Good enough' will do, if it means they can have it now and at minimum cost. This has implications for both the CIO and their internal customers, who now have the ability to implement software within their functions without any IT involvement whatsoever.

23

However, creating a social software infrastructure will be a challenge for most companies, with cultural and political issues to be addressed before any of these benefits can be realized. There are some organizations that simply won't gain from introducing such tools; others will need to resist the urge to deploy them in the same top-down way they know so well. Social software is also disruptive – it changes the role of knowledge management from 'command and control' to 'facilitate and aggregate'. This requires a very different way of thinking and possibly different internal team structures. Finally, consideration needs to be given as to how these tools integrate with existing formal information management systems, the owners of which will feel naturally threatened by cheaper, more usable software competing for employees' attention.

Before any of that can happen, the way in which organizations choose and purchase enterprise systems needs a rethink.

THE APPROACH TO ENTERPRISE SOFTWARE MUST CHANGE

Enterprise software has become ever more complex. Companies spend millions of dollars installing information and knowledge management systems, yet still struggle with the most basic challenge of persuading their employees to use them. For too long the focus has been on the buyer rather than the user, and this philosophy must shift if organizations want to see any of the benefits from an increased level of collaborative working amongst their employees.

First, the pricing must change. Ray Lane, once president of Oracle and lately a venture capitalist at Kleiner Perkins Caufield & Byers, says that software vendors must change their pricing models to compete with the large enterprise providers: 'the entire software industry made one huge mistake in the late 1990s – it focused on buyers and not users' (LaMonica 2006). It remains to be seen whether this will change (I very much doubt it), but this shift in the balance of power from buyer to user will be very much in evidence in new products and services, even those from the most established vendors.

So expect to see a renewed focus on user experience and ease of use. If it looks quick and simple, then one of the biggest barriers to adoption has already been overcome. Employees' familiarity with consumer services like Google Earth or photo-sharing site Flickr are resulting in higher business-user expectations, according to analysts and software executives. 'The design of business applications is more important than ever,' says Joe Kraus, CEO of JotSpot. 'If I'm a buyer at a manufacturing company and I'm using Google Earth to look at the plants of my competition, and the Siebel sales rep asks me to spend \$2 million on glorified database software, that causes a real disconnect.'

In the 1990s some enterprise software vendors were busy telling customers that even the simplest problems needed large, complex systems to solve them. Following the dot-com crash at the start of the millennium few of these vendors survived, usurped by cheap – if not free – alternatives. This trend continues unabated in the form of social software. As Peter Merholz (2005), president and founder of user experience firm Adaptive Path, puts it, 'enterprise software is being eaten away from below'.

The buyers must change too. The value (and therefore the return on investment) in enterprise software was traditionally calculated on the assumption that everyone in the organization would use it. When they didn't, the answer was to throw more resources at communicating, convincing and coercing usage. This supply-driven approach of forcing adoption simply doesn't work with social software, a theme that will be revisited in Part III of this book.

Finally, the methods of delivering enterprise software also need to change. The three main methods have been:

1. packaged software: a licence to install software on the company's servers;
2. Software as a Service (SaaS): remotely-hosted software usually delivered via a web browser, purchase on an on-demand basis;
3. software and professional services: a licence to install the software, but with additional professional services (e.g. customization, integration, etc.) provided by the vendor or a third party partner.

Software as a Service (SaaS) – the concept of renting externally hosted software – is becoming increasingly common as the method of delivering social software (perhaps driven by its online roots). This will undoubtedly contribute to the adoption of social software in smaller enterprises attracted by the prospect of outsourcing the effort of set up and maintenance to the vendor, only paying for what they need. No longer do such companies need to initiate large-scale implementation projects, with long lead times and massive enterprise infrastructure requirements. They can simply trial existing robust and secure services and purchase the ones that suit them best on a monthly, quarterly or annual basis. This means that smaller companies without the resources to deploy complex software can now access all the benefits previously available only to their larger competitors.

BRIEF HISTORY OF SOCIAL SOFTWARE

25

To some extent, social software is nothing new. The founding principles of the Internet were centred on connecting communities and giving them the tools to communicate. Ten years on, and we finally have the technology, desire and means to realize this dream. People are using the Internet to connect with others like never before, and they are doing so with a wide range of simple – yet perfectly effective – tools.

Social software has a history that some argue goes back to attempts to define collaboration-driven software as early as 1945 when Vannevar Bush described a hypertext-like device in ‘As We May Think’. Bush was Director of the US Office of Scientific Research and Development and had coordinated the activities of some 6,000 leading American scientists in the application of science to warfare. ‘As We May Think’ was his call for those same ‘men of science’ – now the fighting had ceased – to turn to the task of making human knowledge more accessible. In the process he predicted many kinds of technology that were subsequently invented, including hypertext. The article, a reworked version of his 1939 ‘Mechanisation and the Record’, described a system called memex:

Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, ‘memex’ will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

The article also describes some of the potential benefits of the memex machine to groups:

Wholly new forms of encyclopedias will appear, ready-made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified. The lawyer has at his touch the associated opinions

and decisions of his whole experience, and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest. The physician, puzzled by its patient's reactions, strikes the trail established in studying an earlier similar case, and runs rapidly through analogous case histories, with side references to the classics for the pertinent anatomy and histology. The chemist, struggling with the synthesis of an organic compound, has all the chemical literature before him in his laboratory, with trails following the analogies of compounds, and side trails to their physical and chemical behavior.

In 1958, the Advanced Research Projects Agency (ARPA) was formed in the US, developing the first successful satellite. By 1962, ARPA had changed its focus and began to offer research grants to universities, under the leadership of J.C.R. Licklider. His efforts led to the creation of ARPANET and ultimately the Internet itself.

26

Licklider was clearly a visionary, foreseeing the time when computers would be able to interact in real time with human beings, thus contributing to their ability to formulate insight and make decisions. In March 1960 his paper 'Man-Computer Symbiosis' set out this vision:

Man-computer symbiosis is an expected development in cooperative interaction between men and electronic computers. It will involve very close coupling between the human and the electronic members of the partnership. The main aims are 1) to let computers facilitate formulative thinking as they now facilitate the solution of formulated problems, and 2) to enable men and computers to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined programs.

In the anticipated symbiotic partnership, men will set the goals, formulate the hypotheses, determine the criteria, and perform the evaluations. Computing machines will do the routinizable work that must be done to prepare the way for insights and decisions in technical and scientific thinking.

Eight years later, with Robert Taylor, he wrote 'The Computer as a Communication Device' (Licklider and Taylor 1968), predicting 'in a few years, men will be able to communicate more effectively through a machine than face to face'. In so doing, he first introduced the concept of 'on-line interactive communities' and communities of interest:

In most fields they will consist of geographically separated members, sometimes grouped in small clusters and sometimes working individually. They will be communities not of common location, but of common interest. In each field, the overall community of interest will be large enough to support a comprehensive system of field-oriented programs and data.

In each geographical sector, the total number of users—summed over all the fields of interest—will be large enough to support extensive general-purpose information processing and storage facilities. All of these will be interconnected

by telecommunications channels. The whole will constitute a labile network of networks—ever-changing in both content and configuration.

What will go on inside? Eventually, every informational transaction of sufficient consequence to warrant the cost. Each secretary's typewriter, each data-gathering instrument, conceivably each dictation microphone, will feed into the network.

You will not send a letter or a telegram; you will simply identify the people whose files should be linked to yours and the parts to which they should be linked—and perhaps specify a coefficient of urgency. You will seldom make a telephone call; you will ask the network to link your consoles together.

It is worth remembering that this was written 40 years ago, yet could easily be describing many of the social networking and collaborative working communities that are just a few years old today.

27

In October 1962 Douglas Engelbart (who went on to invent the computer mouse five years later) published 'Augmenting Human Intellect: A Conceptual Framework':

In such a future working relationship between human problem-solver and computer 'clerk,' the capability of the computer for executing mathematical processes would be used whenever it was needed. However, the computer has many other capabilities for manipulating and displaying information that can be of significant benefit to the human in nonmathematical processes of planning, organizing, studying, etc. Every person who does his thinking with symbolized concepts (whether in the form of the English language, pictographs, formal logic, or mathematics) should be able to benefit significantly.

The concept of networked augmented intelligence is attributed to Engelbart based on this pioneering work. He believed that technologies were needed that would allow humans to manipulate information directly – and thus improve individual and group processes for knowledge work – rather than allow the state of current technologies to dictate man's ability to interact with data. It is this thinking that led to his invention of the mouse, a tool for such interaction that we all take for granted today.

In the 1970s the focus was on 'office automation', a term coined by IBM but that eventually led to the Electronic Information Exchange System (EIES), widely regarded as the first major example of collaborative software. In 1977, EIES was being used to network scientists worldwide using email, chat and discussion groups, long before the existence of the Internet as we know it today. After seeing EIES, in 1978 Peter and Trudy Johnson-Lenz coined the term 'groupware', defining it as 'intentional group processes plus software to support them'. Those companies that developed such software in the 1980s subsequently adopted the term.

However, the academic community preferred Computer-Supported Cooperative (or Collaborative) Work, or CSCW, a term still used for the annual conference bearing its name produced by the Association for Computing Machinery (ACM). The cooperative/collaborative debate between the human-computer interaction and information systems communities is very relevant to today's social software, summed up visually by Scott Schopieray (2003).

Schopieray's model identifies cooperation as being focused on the product and collaboration on the process, yet the objective of both being to generate a result that is better than what could have been produced alone (see Figure 2.2).

The term groupware continued to be used everywhere except academia, and was put on the map by Robert Johansen's book of the same name, published in 1988. Christopher Allen (2004) of Alacrity Ventures laments the role of marketing in corrupting the term further:

Unfortunately, it was this success that was also the downfall of the term 'groupware', for it got co-opted by marketing. Initially the co-opting was done by Lotus Notes, which I personally didn't feel deserved to be called groupware, as it was really more of a multi-user database that could be used to make groupware, but wasn't actually groupware. Then Microsoft further corrupted the term when they released Microsoft Exchange Server and Outlook with calendaring features to compete with Lotus Notes, and called that groupware as well.

Possibly the most important catalyst of all came in 1989 which a young British engineer working at CERN in Switzerland circulated a proposal for an

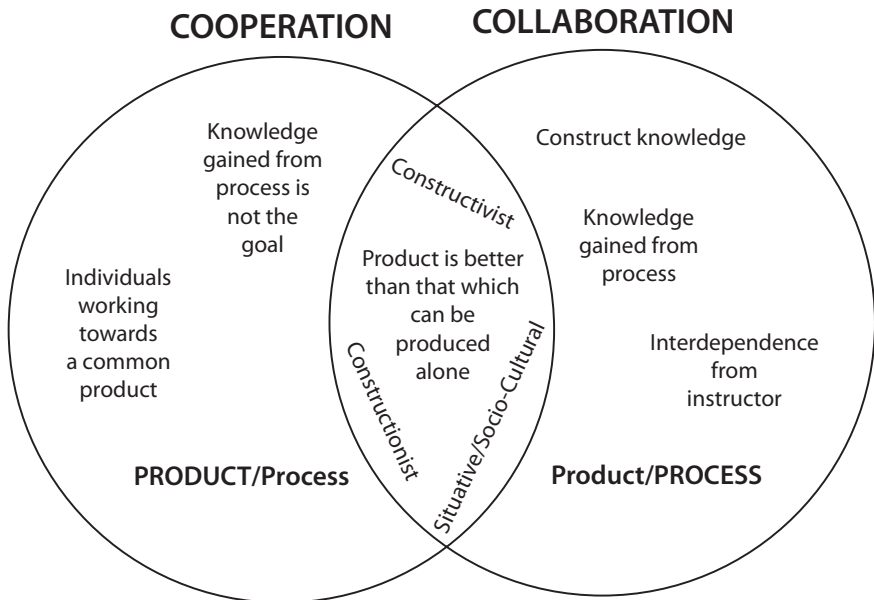


Figure 2.2 A comparison of cooperation and collaboration

© Scott Schopieray

in-house online document sharing system which he described as a 'web of notes with links'. His name was Tim Berners-Lee and he called his system the World Wide Web. The rest is now history.

As the use of the term groupware began to slow, and interest in the World Wide Web increased, 'social software' emerged as a way of describing functions that relied on user-selected filtering and evaluation. It was K. Eric Drexler, founder of the Foresight Institute, who is regarded as first using the term in this context in 'Hypertext Publishing and the Evolution of Knowledge', originally published at the Hypertext '87 conference (1991). In a reply to Christopher Allen (2004), Drexler explains why he used it:

I used the term 'social software' because I am concerned with communication and collaboration on all scales, including the whole of society. Thus, I see media at the scale of the World Wide Web as forms of social software.

29

However, it took 15 years for social software to enter into more common usage, when the first Social Software Summit was held in November 2002.

Whilst social software was not limited to the Internet, and even existed before it, the term Web 2.0 has garnered more enthusiasm in recent times. However, many commentators see it as nothing more than a buzzword trying to bring together a collection of ideas, with a limited shelf life (indeed, already 'Web 3.0' is being discussed). Others see it as completely distinct from social software, being nothing more than a descriptor for a second generation of web-based services facilitating collaboration between its participants.

The Internet in itself, however, did not spark the social software phenomenon that we are experiencing today. It is true that many of the characteristics of today's social software are based on older forms suggested and developed by Bush, Licklider, ARPA, Engelbart, Drexler and others, but it is without doubt the convergence of technological, cultural and societal factors that has brought about the desire and ability for people to use technology to communicate, connect and collaborate with each other on a scale never before experienced.

APPLYING SOCIAL SOFTWARE TO BUSINESS

It didn't take long for people to start thinking about the similarities between this social software and the new 'Web 2.0' concepts outlined by Tim O'Reilly and John Battelle, and how this might play out in the enterprise. The first to put a name to this thinking was Harvard Business School's Andrew McAfee when in Spring 2006 he referred to 'Enterprise 2.0' in his *MIT Sloan Management Review* article 'Enterprise 2.0: The Dawn of Emergent Collaboration' (McAfee 2006). McAfee's latest definition reads thus:

Enterprise 2.0 is the use of emergent social software platforms within companies, or between companies and their partners or customers.

McAfee suggests that Enterprise 2.0 has become a reality because of three broad and converging trends:

1. *Simple, free platforms for self-expression*: McAfee quotes American journalist A.J. Liebling who said that ‘freedom of the press is limited to those who own one’, adding that the birth of free publishing platforms in the form of blogs means that those limits now apply only to those with neither the ability nor propensity for self-expression.
2. *Emergent structures, rather than imposed ones*: instead of imposing their own ideas about how platforms should be constructed, McAfee points out that technologists started to build tools that let structure *emerge*.
3. *Order from chaos*: the third trend in McAfee’s argument is the ability for people to quickly and easily filter, sort and prioritize the flood of new online content.

30

Although he suggests otherwise, McAfee’s trends are still primarily technological, and it could be argued that alone they did not make Enterprise 2.0 a reality. In fact, it is hard to see why these trends are specific to the enterprise rather than the general Internet populace. As with all social movements, the social software movement represents the coming together of multiple trends out of which emerges a profound change in behaviour. Technology is just one element along with some of the seismic shifts taking place in both business and society.

He does, however, focus specifically on the failing practice of knowledge management. He believes that the information technologies that knowledge workers use for communication have historically been either *channels* (email, instant messaging, etc.) or *platforms* (e.g. intranets and portals), and they could not be more polarized. Channels allow information to be created by anyone, but consumed by very few (email threads are only visible to those who are included in them, for instance). Platforms, on the other hand, generally contain information that is available to the whole organization, yet generated by small groups.

Against this background, the focus of knowledge management in those organizations where the function exists has been to elicit information from people throughout the company – usually in a structured way – and put into a central system for everyone else to access. As anyone who has ever been on the receiving end of such requests can attest, this method of capturing information neither fits with the way most knowledge workers operate nor reflects what they produce. No wonder then that those trying to locate knowledge within an organization find it so difficult, and when they do find it more often than not it rarely provides the answer to their questions, which generally begin with: Who ...? How ...? or What ...?

McAfee heralds the arrival of new platforms that do not try to capture information in such an artificial way, but instead focus on the *practice*

(the activity involved in getting work done) and *output* (the product of that practice) of knowledge workers. Picking up on the example of the IT employees at Dresdner Kleinwort Wasserstein, he says:

The excerpts from the DrKW blogs, for example, record an interaction and its output, as well as the identities of three people involved. These blog entries are part of a platform that's readable by anyone in the company, and they're persistent. They make an episode of knowledge work widely and permanently visible.

Enterprise 2.0 was starting to gain understanding, supporters and case studies. But it still had a long way to go before even the early enterprise adopters could see its relevance.