Quee Yew Hao Kelvin
Nanyang Technological Univesity
Speech for Speak Out on ICT 2006

Good afternoon ladies and gentlemen. I'm Kelvin Quee and today my presentation topic will be "Collaboration: A Symphony of Technology".

I'm going to start off by illustrating to you the pervading trends in the business environment today.

Advances in technology have shortened physical distances. It has also allowed a huge amount of information to the available at your fingertips. Together, they have made customers more discerning and also move in a huge amount of competitors.

Business problems have now become tougher nuts to crack then ever.

Business organizations have now realized that in order to thrive in the increasingly competitive and unpredictable environment, they have to "flatten their organizations". Businesses have also realized that in order to slay increasingly tougher problems they have to encourage cross-functional teams to form and function.

All these have encouraged Collaboration. Collaboration is increasingly needed today.

Yet, certain conditions are needed to collaborate effectively.

One - free communications. Team members must be able to communicate without obstacles. They must know that they are allowed to speak freely. It is only so that effective thought exchange can take place.

Two – Team members must be able to work on common documents and be allowed to make and track changes. This is almost a prerequisite for any collaborative software.

Three – Team members must be able to work anywhere with a Internet access. With today's office defined as "anywhere your laptop is", any collaboration software system must have this feature.

Let us examine the current dominating form of collaboration software. The current collaboration software is server-centric - Meaning that users or team members have to go through a server in order to communicate with other users. As you can see from the diagram, logically, as the number of users in a network increases, the network becomes more useful.

Metcalfe's Law says that the value of a network is directly proportional to the number of users squared.

Lotus Notes is one such implementation. Lotus Notes was developed by Ray Ozzie in 1989. Even at that point of time, when the Internet was in its infancy, Lotus Notes had implemented email and shared workspaces for collaboration. It has a document-centric database and already support public-key cryptography. It has a syncing feature known as "replication" that allow users to maintain seamlessly a copy of their documents both remotely, ie on the servers, and locally, ie on their own laptops or desktops.

Yet, a server-centric system has some obvious flaws. For one, the number of users at any one time on such a system is physically limited to the server capacity and the network infrastructure the clients reside on.

Another limitation is that, from the graphic, you can see that there is a single point of failure - meaning that if, for any reason the server goes down, all the clients are disconnected.

The server-centric system is also restrictive in the sense that everyone must be on the same physical network in order to work effectively.

To get around these limitations, geeks have developed an alternative to the server-client system. It's called the peer-to-peer network. In a peer-to-peer network, every peer is both a client and a server. In this sense, all physical resources and dynamically-scaled – the more peers are on the network, the more computing resources you have.

A peer-to-peer network is also inherently robust. There is no single point of failure. The only way to get everyone off the network is to have the multiple network failures, a near-impossible unless everyone is residing on the same network.

Ray Ozzie, who is also the creator of Lotus Notes, created one such implementation called Groove.

Groove, besides inheriting the obvious advantages of a p2p system also allows co-workers to form ad-hoc collaborative networks on their own. Team members can form temporary networks with their co-workers, or even with their suppliers, partners or customers, basically anyone, with an existing network connection.

While this might spell out unlimited convenience, it also raises security issues. You will never know whether your friendly peer on the network is friend or foe.

Also, as Metcalfe's Law suggests, the value of a network, when you have no peers is, almost, useless. At least, when you were in a server-client implementation, you had a server with a repository of documents to work with.

Given the advantages and disadvantages of both implementations, instinctively, we will want to combine both networks together to give a hybrid.

In a hybrid, we have the server handling the authentication and document storage. To keep the server load down, administrators will restrict document storage for backup purposes only.

Next, the hybrid will leverage off the dynamic resource scaling abilities of a p2p network, effectively transferring all load to the p2p network.

Groove already has such an implementation. Whether or not they will go further to exploit the flexibility of the model will depend on their current owner, Microsoft.

STOP: With such near complete features of collaborative question, we then beg to ask: Why aren't we seeing more collaboration?

There are 2 common misunderstandings that are inhibiting collaboration.

One - Collaboration… is… NOT… cooperation. Unlike common understanding, they are not the same. Dillenbourgh clarifies that although work is split in both cases, they are split differently. In cooperation, it's often hierarchical. In collaboration, each person might do parts of tasks in different levels, hierarchically. To picture that, we can imagine a multi-tiered wedding cake. In cooperation, we will split the cake by their different levels. I will eat the bottom tier, and since you are slimmer, you will eat the top tier. However, in collaboration, we will eat slices from different tiers.

The chief difference in consequences is this – In collaboration, I've an overall impression of how the project is like, while in cooperation, you will only have knowledge of how each task is like and that understanding is myopic and, often, wrong.

Oliver Zara, with understanding of that common misunderstanding, goes on to say that in collaboration, it's often very much like a marketplace. Information comes in from all sources and people selectively evaluate, take and contribute back. Over time, what remains of the marketplace is a shared, constantly developed and refined knowledge and understanding, or collective intelligence. And that is product of collaboration.

Two – There is an overemphasis of technology, or at least, information technology. Managers and Information Officers often overlook the fact the collaboration involves and what is just as important is People Technology. People need to be taught and encourages on how to collaborate.

A fundamental Cultural Change is needed to change the internal company culture from competitive to cooperative. Too often, people feel that if they share information with their co-workers, they maybe placing themselves at a disadvantage in terms of promotion. We need to change the conception.

With these 2 misunderstandings in mind, we will try to implement a Collaborative Work environment. And, often, it all boils down to Implementation.

Oliver Zara, in 2004, listed out the 3 steps to collaboration. “Want to Cooperate”, “Know how to cooperate” and “Have means to cooperate”. Zara says that, in his exact words, “people feel more ‘comfortable’ starting with step 3 and working backwards, in which case it is just a matter of time before the initiative will fail.”

But if we do Step 1 first, wouldn’t we have to first find out, if our People really “want to cooperate”?

That is why we need an Evaluation first.

For a complete evaluation, we will first need to see if our People are ready. Next, Organisation. The organization needs to be “flat”. Finally, we go on to “Nature of Work”. Not everyone needs the full suite of collaboration tools. Finally, we then we look at Technology. Some organizations may already have ready, off-the-shelf tools that they can use, like emails, which is a powerful tool, if used properly.

One might imagine that a project with over 6 million source lines of code might require highly sophisticated collaborative tools. Think again. The Linux kernel project is primarily coordinated using nothing more than plain emails and IRC.

Let’s do a conclusion.