

Software Development Trends in 2009

A Whitepaper developed by [Keystone Technology Consultants](#).

Introduction

Software development remains a combination of art and discipline. The individual writing a line of code can write it many ways and arrive at the goal of a function within a business process. But the overall effort is a discipline with a moving technology. The organizations and process being modeled in the user interface, business rules and data structures are constantly changing in terms of capabilities demanded and available, social and political changes, and certainly today the financial aspect.

This white paper is provided as a guide in 2009 to how the discipline aspect within the changing industry can be leveraged to provide better long term solutions.

We will first discuss the impact of change looking back ten years, and then discuss five key technologies looking forward. These are:

1. The Economy's Impact
2. Cloud Computing
3. Mobile Computing
4. Social Networking
5. Rich User Interfaces

We then conclude by summarizing them into an ideal project.

The change has been occurring, and will occur, so it must be understood and harnessed

Software development technologies and methodologies are fluid. Being technology based, neither remains the same over even a few years. People in this career field are well aware of the challenge of keeping up with the constant migration. While most industries face cyclical changes on a regular basis, software development must also deal with structural changes. Since the dawn of "Y2K" in 1999, software development has changed significantly. Then most applications were thick clients intended for installation on a desktop PC connecting to a back end database such as SQL Server. The application model was at the time considered complex in the sense there were multiple tiers and communication between each to marshal data quickly and securely over a local area network, and occasionally over HTTP or other protocol via the internet.

Since then the changes have been enormous; with data now routinely available from web services, RSS feeds, selected internet ports combined with some exchange interface, handshake method, GPS devices, images, email stores, SharePoint objects, etc. The coding tools have become more sophisticated to handle more events, data types, deployment targets, and debugging methods across systems, among others. Additionally, they have been merged with project management, source code control, and testing tools. Coding tools such as Visual Studio.Net and Eclipse now offer the developer with a complete world they can be productive in. The databases have changed to support embedded objects, XML, and web services. Everything scales to new heights both up (more tiers and ability to harness local resources, e.g. server memory) and out (more clustered machines).

The web has delivered numerous new opportunities and challenges. Firms no longer have to create every module; instead they can obtain services, data processing, and information from others who expose their functionality via web services. Nobody would write a FedEx shipping calculator, FedEx offers one you can incorporate into your application.

Open source has opened up new offerings, with hugely successful projects such as Linux, Firefox, Apache, MySQL and others showing us that not only can teams operate across the world, they can create great software and support it as well as a commercial vendor, though under a different model.

Additionally, new methodologies have sprung up such as Agile and SCRUM, and they bring with them new means to plan, execute, and measure a software project. Projects that used to have an acceptable delivery of years and ROI of a decade are now expected in some part in weeks or months and an ROI of months.

Many of these changes are driven by new technologies that enable new capabilities to support competitive advantages in a business model; including capabilities in the product or service, and also the ability to harness the best talent globally in a distributed team that can collaborate on a project wherever, and whenever, they are.

All of these changes and many more, happened in a ten year period. Even high technology fields such as medicine, space exploration, and energy have not seen this type of change.

The need to understand what is available in terms of technology and methodologies, and how they are shaping and being shaped by global and social forces is essential for any information systems executive. This white paper is written with that in mind, and explores five key technologies that should be evaluated and included in any project plans. And to the extent that your organization relies on information systems as developers, part of your business plan.

#1: The Economy's Impact

As we embark upon 2009, the most important factor is the economy. That may seem strange for a white paper on software development, but it is central to the business objectives of any organization, and will shape the type and size of any software projects you select. Although projects have gotten markedly shorter in duration, they need to be even shorter now. A project that is much longer than four months in this unsure economy may be cancelled before it is ever delivered and put to use. Risks and

Rewards must be identified and accounted for in any plans, and Return on Investment (ROI) must be measurable in months, perhaps less than one year. Larger projects may be phased, but should be done in some way that will deliver value incrementally. It may even be advisable to integrate new modules with existing legacy systems where it makes sense, to add new functions that create additional revenue or reduce costs while at the same time keeping what you have up and running. This may create additional expenditures in the long term; but in the short term will help manage risks in an uncertain environment. When investment funds are in short supply; projects must be evaluated carefully.

This also creates a need to manage a project portfolio. New tools for Project Portfolio Management (PPM) can help a team analyze projects across a spectrum of criteria that are evaluated against corporate goals. Some of them, such as Microsoft Project Portfolio Server, will collect those goals and information on each project, to allow measurement and rankings. We do not recommend taking the results directly as they are presented, but do recommend the tool such as this for inclusion in your process. They provide a team based way to communicate and evaluate initiative in a less subjective way.

Organizations should also look to their vendors for assistance with financing and planning projects for a lower initial cost. Many companies, such as Microsoft offer financing programs with very attractive terms, even in this capital thin market. They can be used to finance equipment, licenses, and services. Additionally, vendors should be asked what they can do to change the cost structure of a software project, so that the impact is lessened on the organization. This should be done in a partnership where ideas are freely shared such that the project better meets the functional, quality, and budgetary needs of the organization. This is the mode that Wal-Mart and auto makers have long used to improve the products or services suppliers provide.

#2 Cloud Computing

The phrase “cloud computing” has become generalized to the extent that not many can agree on its real meaning. For our purposes here, we describe it as “a set of services that are hosted elsewhere on the internet that we can use in our process”. By this we mean that someone has a set of code that we can use for some function, and we do only a minimal install, if anything, to employ it. The following examples would be included:

- Disk space at RapidShare or Microsoft Live Mesh
- P2P file sharing services such as BitTorrent
- Utility computing by the hour or month from vendors such as Amazon’s EC2 service
- Hosted applications such as Salesforce.com or ZoHo
- Narrow focus applications with a rich user interface like MindMeister and Gmail
- Complete office environments via a Citrix or Terminal Server
- Information feeds such as RSS, web services, etc.

All of these have different uses, but they exist somewhere else and you can access them as needed to create and store new content, generate mash-ups, and run an entire operation with no significant upfront investment.

In particular for software development in the cloud is the ascension of Service Oriented Architecture (SOA). This approach to software modeling and development is focused on the idea that business processes need to work together in a loosely coupled format.

In the past software was tightly held and when linked to a separate business entity (which could be anything from a trading partner in a foreign country to an application installed on the same machine) was tightly coupled. The interfaces to exchange information were largely custom developed, closely defined to the underlying data structures and means to physically transfer, and were only effective within that relationship. If a partner in this relationship changed things, or could no longer supply the needed data, the entire interface often had to be redeveloped, often with changes in the underlying system.

But new technologies change the terms of integration and in the end the entire model of software development. These technologies are known as Service Oriented Architecture (SOA), and are made possible by using the internet as a transport layer between systems that are made available to provide services (known as “Web Services”) to each other. This helps us in many areas, and any business process provided by application or even system software can be reviewed for outsourcing. This capability was not available before technologies like the web and web services. SOA is a model that ties disparate systems together using communication interfaces that allow them to talk to each other in industry standard terms. You might consider it an evolution of long established Electronic Data Interface (EDI). But EDI, while still providing value in many functions, has two inherent problems. First of all, it is focused on the format of the messages, not the means of actually transporting them. Because of this, so called Value Added Networks grew to provide connection services and clearing houses between two systems that could not natively talk to each other. This increases costs, and reduces flexibility. Secondly they only cover specific transactions that a standards board has set. So you can transport a Purchase Order and an invoice, but not a flight reservation, request for properly formatted +4 zip codes, or product specification; some things that would interest various organizations. So EDI has a standard, but it is limited to specific functions and specific data elements within those functions. This in turn limits an organization from responding to their own needs.

What SOA provides is the ability of two organizations to agree to outsource a function or even an entire business process, such as loan approval, appraisals, and inventory analysis.

Business Process Engineering and SOA

Businesses have spent a lot of time thinking about the process they used, and they have defined and redefined it using Business Process Reengineering, Six Sigma, Total Quality Management, Continuous Improvement, etc. ad nauseam. Many improvements were made, and that unique business process, designed for the organization that employs it, has been encapsulated in the organization that uses it. It is hidden in business rules and databases, web pages and email clients, ready to be forced on users at the appropriate time to make sure the business process is followed. This gives rise to ERP systems, and large Line of Business applications (LOB) that perform functions across an enterprise, following a predefined business process to ensure uniformity, predictability, and cohesion among business functions. Take for example the simple Zip Code + 4 function: an organization wants to ensure that every mailing is successfully delivered to the right location, and so as part of the process they run the address against a computer application that returns the correct nine digit zip code. In this way, costs are reduced, and the correct targets receive the content in the mailing. But sometimes the rules or even underlying data change in this specific function. New addresses are added, street names changes, zip codes redefined, etc. This may necessitate a change deep within the LOB or ERP, and your organization,

with its very organizational specific business process built into the system, is the only one capable of making the change using internal employees familiar with the code and structure. You bear the full cost of the change, and can only amortize it across your organization. But the Zip Code + 4 function is NOT core to the business. It needs to be done, but it is not how you actually generate revenue. The content of the mailing should do that, but the commonly used Zip Code + 4 does not.

The architecture organizations have traditionally used over this re-engineering period is monolithic. They built single system, or tightly integrated systems, that they owned or highly customized to the point of not even looking like the application they procured. These were often hidden behind a firewall, focused on the broad and internal needs of the organization's process. What they did not invite was the ability to simply integrate one little external function into the business process. So the Zip Code + 4 function remains locked in the LOB or ERP. This limits flexibility to reuse and share the function, expand it to new acquisitions; in short to generally focus on the core process.

But what if you found another organization, that focuses only on this functional need, and which would allow you to connect your business process directly to theirs? You send them the address, and they send you back the properly formatted address with the correct Zip Code + 4? That is what SOA and web services can provide. You can pick from various vendors who will provide the service, selecting the one that best meets your quality and budgetary needs.

In summary, software design phases need to examine all needed functions and review the market place for suitable SOA services that may provide for the need. Furthermore, software projects that do write lines of code should be developed so that interesting and unique functions can be exposed as services for others to call. This will allow your catalog of products or services to be more accessible in the market place, not only driving new revenue, but forging stronger relationships with those that subscribe to your published services.

#3 Mobile Computing

This trend must be understood from both a technology and work force perspective.

From a technology perspective, new devices and means to connect to offices and remote locations have allowed people to go even further towards a virtual office. Long seen as a trend, this is not revolutionary for 2009, but has made some major strides of late that should be taken into account. The most prevalent and impactful trend is in connectivity options. 802.11x wireless is almost ubiquitous now, so users can be connected from any coffee shop or home office. But even when that is not available, the wireless carriers such as AT&T and Sprint have stepped in to provide high speed ("3G") connections over digital cellular networks. These can be accessed via a plug in card on a laptop computer, a tethered smart phone, or increasingly a carrier's SIM chip installed directly in the laptop. These networks provide better security than open 802.11x wireless access points, and can also be faster. Additionally they work in more places, subject to a carrier's capabilities within a geographical region.

Additional technical developments include more capable small form devices, such as the iPhone, Windows Mobile 6x, and newer Blackberries. These devices can serve as an entire office productivity platform in many situations, with the ability to email, access websites, edit Microsoft Office documents, and read PDFs. Finally, the entry of "Netbooks", small highly portable notebooks that are intended for only light local processing, and focus more on Internet access, will continue the push to cloud based

applications. These low cost devices are perfect for accessing a CRM service such as Salesforce.com, but attempting to run a local high end CRM client will be frustrating and unproductive. They simply lack the horsepower to fully support the software needs of these fat client applications.

Beyond technical concerns and developments, the workforce continues to change. People still employed are being asked to work more hours in a tightened economy, and they increasingly do this from home or the sidelines of a local soccer field. This means they either need to take work with them (notebooks, etc.) or access it remotely (web based applications, remote desktop, etc.). Furthermore, a global economy means that people may work on systems across an ocean, and in close collaboration with people they have never met. They need the same experience in terms of performance and capability, along with the experience of collaboration, as those working in the corporate office.

These technology and workforce changes mean that the software being developed must be designed with this trend in mind. The term “Smart Client” has been coined to describe these applications; ones that must work as well when disconnected as they do connected, and navigate between each seamlessly. The impact on the software that may be developed in this environment is as follows:

- Security – Because we do not have physical barriers to access, the software must ensure secure connections by authenticated users. This means that technologies such as Secure Socket Layer (SSL) encryption, and Secure ID keys should be used to minimize the risk of malicious data loss.
- Cached Data - In a sometimes connected environment it cannot be assumed that the user now connected will be connected as they take their next action within the software. In fact, it may be weeks before they reconnect again. Obviously some software requires a regular and sustained connection, but many applications could go significant periods without the need for connectivity. If you are using wireless connected devices in a remote environment to inventory livestock, the basic application should be up to date, and any business rules or lookup data be up to date as needed to do local processing. But the application should be engineered to do three important tasks:
 - Update itself – upon connection, if a newer version is available, it should be applied without user intervention (other than a delay to process other immediate tasks). It should also report the version in use to central management server. Included in updates may also be some domain specific data, such as lookup lists, inventory, pricing, etc.
 - Store data locally for long periods of time, and navigate between an online and offline mode seamlessly. In this case, the application should work whether connected or not. The user should not be concerned unless a business rule forces it (like “rush orders must be submitted in 4 hours”). The local storage should be robust and secure. The application should monitor for a reconnect, and replicate data both ways as needed.
 - Handle data transfers from older versions – the transmitted data should be versioned alongside the application, so that for example orders originally created in the viewed on version 1, can still be received and processed at the central database now on version 1.1. Obviously there are limits to this, but small updates should never prohibit processing older transactions or require rekeying, etc.
- Remote Kill – Because these devices can be lost or stolen, and are presumably still connected at some point, the ability to send a remote kill, in which all data and important applications are removed, will be important to account for. Third party applications exist to provide this service and should be reviewed for inclusion into the solution.
- The need to include social networking capabilities. Because of the disconnected nature of such users, the ability to collaborate should be considered. This is further described below.

#4 Social Networking

Some have called 2009 the year of FaceBook. The popular website, which could be considered MySpace for adults, creates an environment in which people can track other's lives through text, pictures, videos, and chat sessions. Numerous plug-ins exist to add functionality, and whole communities have sprung up around companies, schools, social causes, and personal or professional interests. It allows a level of interaction not provided by email or other electronic exchanges, and does so much more efficiently. Other sites of note offering similar capabilities include LinkedIn and JigSaw. An additional site, Second Life, offers a virtual reality type interface, and includes the concept of its own currency, real estate, and shopping venues.

But beyond the technology, the social aspect cannot be misunderstood. Far from goofing off, people use these mediums, along with Instant Messaging, texting, and other tools to interact. Any good business goes where its customers are or where its prospective employees congregate, and communicates on their terms. The youngest generation in the marketplace and work force has spent years with these technologies, and expects to be able to use them in their activities for work or personal life. They see no barrier and just use them the way an older generation used a phone. We recently heard the story of a high profile brokerage firm recruiting a new employee, who asked if she could use IM and texting. They told her that was not allowed, and she turned down their very generous offer. Whether they looked old and stodgy, or she viewed that as being too limited to her success in a contact driven industry, she did not want to be a part of that.

These tools must be viewed as that, tools to do the same thing phones, documents, email, and other forms of communication provide. As collaboration tools, they can actually be much more efficient than older forms of communication.

And based on this movement, combined with SOA and a mobile workforce as described herein, social networking, or newer forms of collaboration, should be considered for inclusion in any software project. These may be just links to sites like FaceBook, or may be a project built in SharePoint, but the ability to allow these communications within the business process, and the information being evaluated, will be key. When combined with workflow tools, the power of capturing the interactions around transactions will provide unique insight into the means and reasons why decisions are made. This will open new opportunities for training and improvement.

#5 Rich User Interfaces

The applications that incorporate some of the other technologies listed here used to be a sparse wasteland of User Interface (UI) functionality. Early web mail was an exercise in frugality, with a lot of slow return trips to the server, a one size fits all layout, and poor presentation and navigation features. Email was presented in a table, with options to view an email by clicking a link and then seeing it in a new window, where it could be managed by deleting it, forwarding it (often with another window), etc. Users were happy just to see it.

But the web now offers rich user interfaces, in the form of "Rich Internet Applications", which rival desktop offerings. Applications like ZoHo, Jott, MindMeister, and SumoPaint feel like desktop applications, with drag and drop, right click context menus, and progressive search list boxes. Of particular note is Sumo Paint, which a casual user would mistake for Adobe Photoshop; provided for

free and using Adobe Flash as a platform, this software is amazingly close to a desktop quality tool. In any cases, as long as an organization is willing to sacrifice some “nice to haves” they can accomplish everything they need to in freely available online software.

One of the advantages of this technology, except where noted, is that they are cross platform. A user can walk up to a Windows PC at a trade show, connect to an internet site, and start using the application. They can then disconnect, and go their hotel room and reconnect from a Macintosh, and have the same experience within the application.

Some of the technologies for consideration include:

- Tools that require a local download of an engine, such as Adobe Flash, Microsoft Silverlight, and Adobe Air. These applications download code modules at run time, which then run under the locally installed engine. They offer the richest form of interfaces available, and the best performance. They are limited by the need to install something locally, which may require a download or circumvention of policies in some instances (but generally the needed module is already available, especially in the case of Flash).
- AJAX – this is an extended use of JavaScript. In a web page, users receive JavaScript which runs on a local interpretive engine, and can communicate directly to a server, without reloading the page. This improves performance by reducing data transferred and allows a higher level of interactivity. As an example, many sites will have user interface elements such as a text box for a search activity. When the user keys in a few characters, say 3, the local JavaScript takes that input and sends it to the server, requesting the most probable or popular words that may be entered at that point. It then displays the list to the user who can select from it, or they can enter the next letter repeating the process. The increased network speeds available today provide a platform that allows this to work well. AJAX also allows drag and drop interfaces, and graphs, overlaid tags in images, etc.
- Smart Clients - This was discussed previously, but of special note here is the ability to interact with local devices and a remote server simultaneously. If you have applications such as an inventory tracker that need a local barcode scanner, a Smart Client developed in Windows technology can support numerous local devices, provide a rich user interface, and connect to a remote data source. Note that these may not support the cross functional platforms the others do.

The Ideal Project

The ideal project is one that is consistent with your organization’s goals. It will allow you to merge the technology and the business so that with good communication, everybody is pointed in the same direction.

But taken collectively, the ideal project will incorporate some of the technologies and approaches shown in this document. It will be a project that:

- Is aligned with the organization’s business goals
- Has a known and measureable ROI and an excellent chance to reach that level
- Incorporates pieces that others who focus on the service they do best and provide in the cloud

- Is universally available to users without barriers other than appropriate authentication; they can access it anywhere
- Has an embedded reason to be used, especially in the sense that it creates a community around it that drives usage and new benefits.
- Has a rich user interface that does not impede users, and is intuitive and powerful to increase their productivity.
- Actually embodies these characteristics in a way that allows others to also benefit, so that data is exposed in web services, communities can be extended, and new iterations built to truly extend not just the application, but also the organization in way that makes them a better choice for their customers.