



Mass customization

Process enabled application development

White Paper

By Jon Pyke,
Chief Strategy Officer,
Cordys

Oh - how things have changed!

As we have stated before, the entire field of computing is fast becoming a "Cloud - a collection of disembodied services that are accessible from anywhere and detached from the underlying hardware.

Some say that the advent of Cloud computing could have a greater impact on business use of IT than the PC revolution did in the 1980s. The flexibility and potential cost savings of using applications in the Cloud accessed via the web is likely to fuel adoption across the board. There are practical examples of scenarios where this is already happening. We all know that the average user is using web-based applications such as Salesforce.com, independently of the corporate IT department. This is putting increased pressure on IT management to look at such technologies and determine how best they can be used within the enterprise. This means that it is becoming increasingly difficult to maintain the fortress approach of keeping all IT within the confines of the corporate network. Corporate IT cannot continue to resist the wave of change that is so dramatically affecting the way we live.

Consider for a moment the application development lifecycle that we have followed, endured even, over the past 40 years.

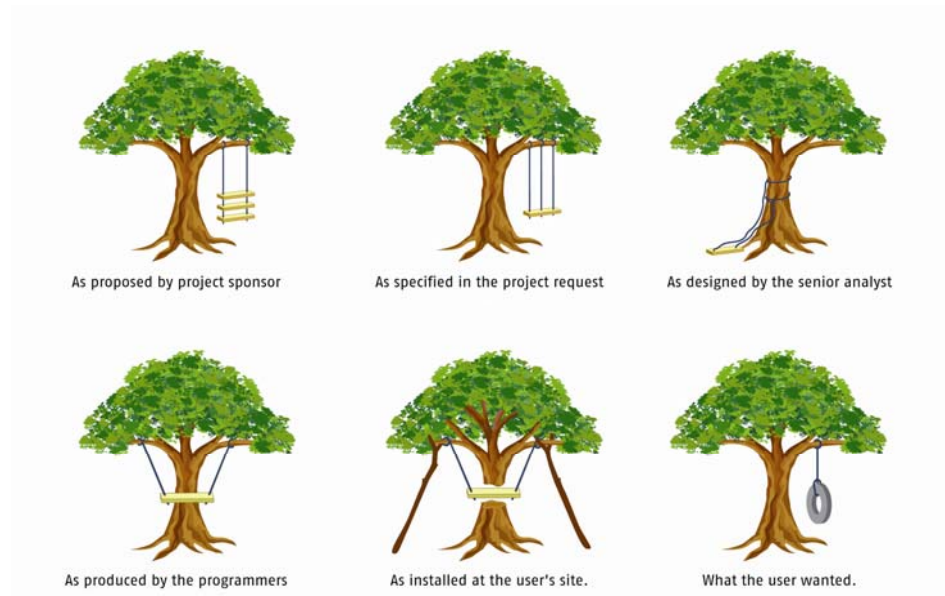


Diagram 1: Illustrating the communication gap between business and IT

The above illustration first appeared some 35 plus years ago in a long gone publication called *Computer Executive*¹. It was submitted by an unknown reader to highlight the communications block between, what was then called the EDP group, and the business user. The worrying thing about this illustration is that in many environments it still holds true.

The problem regarding the communication issue, highlighted with the above illustration, has not got any easier to resolve. In fact, there is an argument to say that it has actually got worse. Back in 1972 what went on in the data processing department was viewed as something of a black art, practiced by people with beards and sandals. The business user wasn't expected (and probably didn't care) to know what was going on. Business representatives tried to articulate a need in terms that could be understood by computer experts, and several months later a specification of the problem would appear (in "computer speak") on the user's desk – along with the question "is this what you meant?" which started the process depicted above.

Application development – a changing world

Having said that, the way we buy and build applications is finally changing and it is changing beyond all recognition – and it's happening very quickly.

Furthermore, the way applications are deployed is also changing. Consider Apple and the iPhone. The success of the App Store for the iPhone is nothing short of phenomenal. At the time of writing Apple had just reported more than 1 billion downloads in less than a year's operation – there are over 35,000 applications ready and waiting for iPhone users to access. Is this the way forward for applications?

Look also at the Salesforce.com approach. Their strategy was to target the needs of specific audiences. In the early days of Salesforce.com many salespeople of large organizations were using it on their laptops without their IT departments being aware of it. Once they had captured these key users, they addressed the needs of developers and small companies to build add-ons and generate additional revenue through the AppExchange. Salesforce.com subscribers grew from 30,000 in 2000 to just under 1 million in 2007.



Diagram 2: The Salesforce.com revenue growth by quarter

Who will ever write a windows application from scratch again?

What an outrageous thing to say! How could anyone possibly think that? It would almost spell the end of Microsoft as we know it!

But what is the likely impact on the business of this phenomenon? And why is this different from what's gone before? Is it just more IT hyperbole?

It is hard to imagine the extent of change we have seen in the past 20 years. There is almost nothing we now do in our daily business lives that does not involve accessing and using the internet. This ubiquitous tool is now so tightly bound into our very existence that you have to stop and wonder how we used to do things before.

But how have these technology advances made things different or have they actually impaired a organization's ability to change the way they develop and deploy applications?

A short but relevant history lesson

To gain an understanding of what might happen in the future we sometimes need to take a look into the past.

Over the past 40 years the IT industry has evolved at an incredible pace and that pace of change has accelerated over the past ten.

A brief history of the IT evolution looks something like:

- 1960's and 70's Mainframe – COBOL, payroll and accounts
- Late 70's early 80's – Minis, Fortran/Pascal, dbases and applications
- Late 80's – PC, DOS, Lotus 1-2-3 and Macros
- 90's – Client/Server, Windows 3.1, Office and VB
- Late 90's – Internet, HTML, basic pages
- 2000 – to date – Internet, JAVA Script, JAVA, ecommerce
- Next phase is Cloud

A perennial problem throughout this evolution has been software development and the constant quest to empower business users with the required skills to build their own productivity tools.

As long ago as 1982 James Martin (ex-IBM) published a book called *Application Development Without Programmers* – that suggested, and indeed promoted, the notion that we were about to see the birth of an applications development approach that would, in some way, eliminate the need for programmers or coding. His approach went even further and suggested that these tools could be used by non-programmers – by business users. This was at the time when most computer applications were built on mainframe systems, were batch driven, and still required (in many cases) punched cards for data input.

But his wasn't the only approach nor was he the only advocate of simplifying application development and making it more accessible and relevant to the business user. Ted Codd, also an IBM employee, came up with the Relational Database (winning the Turing award in 1981 for relational database development), and on the back of his invention came a raft of tools such as procedural "fourth generation languages", nonprocedural 4GLs.

As the hardware technology changed, got smaller, faster, cheaper, and ubiquitous through the advent of the "standard" IBM Personal Computer, more "business user" tools became easily accessible and readily available. That brought with it the idea of "the killer" application – the spreadsheet being the most notable.

What was missing though was good, reliable development tools – ways of simplifying the task of building applications and putting them into the hands of the business – enter Visual Basic.

It's questionable whether Microsoft had any real idea as to the impact of this simple, almost elegant, way of developing applications for the emerging Windows platform that we now take for granted – up until now Windows is the way we interact with personal computers. The decision made in the early 90's had a major impact on the success of the Windows operating system.

Consider what was happening at the time. For example, if you were involved in developing software products and applications for the emerging "client/server" market one key decision had to be – will I use OS/2 or Motif or Windows (or stay with a green screen VT220)?

The answer was determined by the advent of Visual Basic (VB). Almost overnight it became simple to build Windows applications – less experienced and (in some cases) less skilled developers were in a position to build simple applications that met a simple business requirement, on demand. The business didn't have to wait for IT – they could build simple applications in weeks instead of months or years.

The skill set required to use VB emerged from the simple programs that tech-savvy users were building applications on their TRS-80s and the like. These applications didn't need to be particularly scalable or, importantly, multi-user. They just needed to fulfil an immediate business need and therefore were simple applications to solve simple requirements.

This gave Windows a clear advantage over its "graphical" competitors by targeting a set of developers who, until that time, had not used been in a position to build robust business applications but now had the basic skills to deliver what was needed – and importantly, were consistent with the environment and the user experience expectations offered by Windows.

The impact was significant, as Windows became more accepted in the work place, so other tools became available that enabled the business to "do their own thing". Business users began to write macros for their office applications, to build complex and extensive Excel Spreadsheets – embedding Visual Basic into these products was a masterstroke). Entirely new industry sectors grew up spurred on by the advent of products such as Lotus Notes (the precursor to the internet), PowerBuilder, DBASE and products of that nature.

Situational Applications

We were making good progress towards the goal but it all came to a shuddering halt around 1998 when the world caught Y2K and the emphasis shifted to replacing everything with CRM and ERP solutions – catering for the end user seemed to drop off of the agenda – IT knew best and was at the height of its power – dictating what the business needed and being too busy to respond to "trivial" requests.

But the requirement didn't go away. The business still needed to have available applications that could be used to meet specific business needs – they were and are crying out for what in today's parlance are called, Situational Applications (SAs). But what do we mean by the term and why is it important today?

According to Wikipedia, Situational Application is software created for a small group of users with specific needs. The application typically has a short life span, and is often created within the group where it is used, sometimes by the users themselves. As the requirements of a small team using the application change, the application often also continues to evolve to accommodate these changes. Significant changes in requirements may lead to an abandonment of the application altogether – in some cases it is just easier to develop a new one than to evolve the one in use.

According to IBM² Situational Applications is a term used to describe applications built to address a particular situation, problem or challenge. The development life cycle of these types of applications is quite different from the traditional IT-developed. SAs are usually built by power users using short, iterative development life cycles that are measured in days or weeks, not months or years. As the requirements change, the SA often continues to evolve to accommodate these changes. Significant changes in requirements may lead to an abandonment of the used application altogether. In some cases it is simply easier to develop a new one than to update the one in use.

Why now?

So as we have seen, the idea of end-user computing in the enterprise is not new. However, things have changed – both from a technology basis and business environment. Consider also the dramatic change in tech savvy-ness of anyone coming out of school (the Millennials) these days! Their knowledge and familiarity with web technologies is going to be a major factor in driving the take up of “end user application development” and pushing the concept into the mainstream more than ever before.

So, the combination of (1) the Cloud (no deployment issues), (2) the tools (better than before), (3) the standardized integration methods (much easier to mashup), (4) pricing models which are based around “pay as you go” operational expenditure and (5) the Millennials (better understanding of system building) – taken together, enables us to reach the tipping point where everyday business users will want, and demand, access to technology that will enable them to assemble “applications” on demand. This will undoubtedly have a dramatic impact on the way applications are built and deployed.

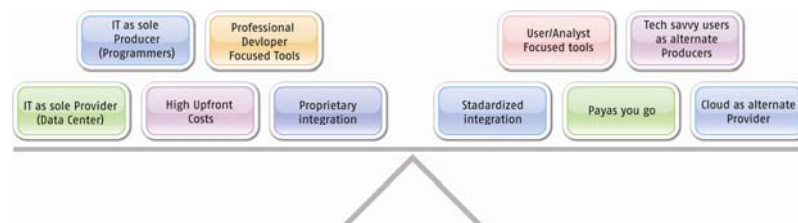


Diagram 3: The application on demand tipping point³

One of the main, if not the most important, virtues of taking this approach is the speed at which the applications can be developed and

the immediate business pay back they deliver when compared with conventional IT development and deployment cycles. The result is an improved ability to respond to or anticipate changing business demands. Also, the organization saves money whenever it changes computerized working methods – usually an expensive and protracted rigmarole. As a bonus, the organization becomes better fitted to exploit future business and computing opportunities, including Business Process Outsourcing (BPO) and Web services.

The downside is that the development is, more often than not, performed in isolation of the corporate needs and may run counter to corporate governance, standards and compliance issues. It therefore can be of limited value in the longer term.

There is one other aspect that needs to be considered in this discussion and that is the needs of the smaller organization – what is commonly known as the SME market but also extends beyond this to include distributed organizations and departments.

Users in this category do not typically have to have in-house systems development capabilities and cannot afford to bring in contractors to help them develop applications to suit their needs. This end of the market either makes do with inappropriate commercial packages or spends too much time trying to get to grips with technology and develop applications themselves. In doing so take their focus is distracted from the more important aspects of running their businesses.

How do small businesses take advantage of low cost, on demand, computing power to drive their business growth, making them more efficient and better able to compete in a global market economy?

Application Provisioning

The answer is process enabled Situational Applications.

The idea behind process enablement of Situational Applications is based upon tried and tested technologies in the Workflow and BPM arena. Offering development and application generators as a Service (Software-as-a-Service) business people can create custom Process-based applications to automate manual tasks. We plan to do this by making it possible to develop applications that will be readily available over the Web and by leveraging Web 2.0 technologies to create a community. We will make use of rich-client interfaces and scalable, multi-tenant server architecture.

Core to this proposition is the concept of Application Provisioning.

Application Provisioning is basically a mechanism whereby a user can put together an "application" based around his normal working patterns, using readily available services.

This means that it is possible to handle any sort of business problem usually tackled by enterprise solutions. The business user will leverage the capability to associate virtually any number of web services within the context of an application. Application Provisioning is effectively an application generator within a process and is inherently more flexible

and more able to comply with corporate guidelines than non-process based approaches.

The Importance of Process

Why do we need process enablement to deliver this vision?

There are many reasons as to why we need process technology to govern the deployment of Situational Applications and chief among them is compliance.

Situational Applications can be very disruptive and lead to anarchy and a breakdown of corporate governance and compliance. Think of all those Excel spreadsheets – Situational Applications – that are used to run most businesses – no control no compliance no ownership. Process enablement of these types of applications will provide ownership, control and auditability – making them compliant with the corporate demands without stifling innovation and change.

Furthermore with the high availability of Cloud infrastructure at a low cost and innovative Cloud Services available to accomplish brave new world tasks, the business needs an assembly and orchestration layer in the Cloud to fully deliver useful business advantages. As a result, the business can become more like the technology:

- More adaptable
- More interwoven
- More specialized

This also sets the business *free* to *mix and match* existing premises-based processes run on applications like SAP, Oracle, IBM Websphere, MS .Net or even early legacies with processes designed entirely on the Cloud without the need to translate business wishes into a complex requirement specification document. This means that:

- The applications are available to use right away and on demand
- The business avoids capital expenditures almost entirely and ensures that any operational expenditures exists only as a result of a revenue stream

So using process based technology we are able to deliver simple to use service based applications, where, and when they are needed (Situational Applications); effectively deploying the technology as managed services along with all the other services that can be found in today's organizations.

So instead of buying expensive software licenses and the requisite supporting infrastructure the eventual end users of the applications will access the processes and services they need when they need them – ensuring cost effective deployment and efficient project roll-out. The ICT departments get what they need, the end users get what they need.

Process enabled Situational Applications quickly become the ideal solution for those departments and organizations that need to develop and provision applications quickly and effectively at the lowest cost possible.

What can I do with this?

Here are some examples of the types of things the business user can build and use.

- Simple purchase order negotiation and approval
Purchase Orders can be quickly created through, say, a Google gadget that automatically gets saved as a Google Document. As part of the process, the order could then be sent to the vendor for further documented negotiation and once agreed it can be sent for approval and recorded in, say, the Salesforce CRM system.. As part of the application you could produce reports and charts to see how the process performed, value of orders, etc.
- Weekly time card recording
Register hours spent on various projects can be recorded through a Google gadget and automatically generate weekly reports for various clients. These could then be sent to your clients for verification and approval
- Service invoice
Build an application to create Service Quote and send it for approval to your Service Manager. Once approval, the quote can be sharedt with the customer. When service activities are completed, it will automatically generate a service invoice from the Quote, be approved and sent to the billing system for invoicing

As you can see from the above example, these applications are the sort that are essential for making a business run smoother but they can take a long time for the IT department to develop. With the approach described in this paper they can be put together in hours and can be changed in minutes to meet changing requirements.

IT departments can play a championing role, educating business users on how to take advantage of these new capabilities without bogging IT down in traditional applications development. BPM veteran Peter Fingar writes in his recently released book, *Dot Cloud: The 21st Century Business Platform Built on Cloud Computing*⁴, "As companies adopt SOA to provide standardized services, the value of IT to the overall business mission grows exponentially. Reusable business process fragments (services) can be reused in many contexts and settings—the key is reusable business process segments, not just reusable software. Those reusable process segments can be tapped as companies design innovative business processes as 'situational' business processes across

multiple business channels. That is, they can be adapted to completely new business situations: new initiatives, campaigns and projects. So it is that software flexibility and reuse enables business process flexibility and reuse. That's the stuff of business agility in hyper competitive markets of the 21st century."

Conclusion

In summary, the key conclusions that we can draw are as follows:

- End user computing is key – more than ever before, business users have the wherewithal and the tools to build what they need on demand
- End user applications (either self developed or off the shelf) are where the money has always been spent (Lotus, Microsoft, Notes) and where most benefit has been gained – especially with personal productivity
- The communications issues and misunderstandings through the decades has meant that end user requirements have rarely been met
- Each new technology break out is created by end users – it's the business that drives innovation and cost cutting
- IT departments eventually take these break outs and make them corporate
- Internet and Cloud are the new break out – putting end users in control
- Internet and Cloud need a glue to bind these apps to make them useful

How?

Cordys provides the missing glue.

Welcome to the Cordys Process Factory.

Cordys Process Factory is a hosted development environment for building process based situational applications in the Cloud. Based on a Platform-as-a-Service, this process orchestration solution helps companies to rapidly build Situational Applications (MashApps®) to instantly meet specific business needs.

Cordys Process Factory operates on Cordys Business Operations Platform (BOP), which is a secure, high available and fully integrated SOA / BPM platform. Cordys Business Operations Platform delivers a complete Platform as a Service (PaaS) solution and is designed to facilitate SaaS delivery models. It has the concept of multi-tenancy at its core and provides robust capabilities to address specific security needs associated with SaaS.

The Cordys Process Factory

- Represents a new generation of business application development and deployment. It enables organizations to achieve sustainable savings and faster business innovation, without capital expenditure on IT investments or complex customizations of enterprise software
- Makes your processes compliant, auditable and controlled in a very efficient and cost-effective way, by automating and deploying them on the Cloud
- Makes it easy to create Mashup of different services and applications, both from on-premise and on-demand systems, and provides comprehensive reports and charts for quick analysis

Using the Cordys Process Factory, our customers are able to reduce IT cost, to increase IT effectiveness, to become more agile and to increase the speed of change and innovation.

References

1. Computer Executive- August/September 1972 edition
2. The rise of Web-based situational applications
<http://www.ibm.com/developerworks/webservices/library/ws-soa-situational1/>
3. Source: Jonathan Sapir – powerintheCloud.com
4. Dot Cloud: The 21st Century Business Platform Built on Cloud Computing, Meghan-Kiffer Press, 2009.
www.mkpress.com/cloud

For further information about Cordys, visit www.cordys.com

Cordys is a global provider of software for business process innovation and Enterprise Cloud Orchestration. The industry-leading Cordys Business Operations Platform (BOP) consists of a complete suite for next generation Business Process Management (BPM), Business Activity Monitoring (BAM) and innovative SaaS Deployment Frameworks (SDF), delivering a complete Platform as a Service (PaaS) solution. It includes an open, integrated set of tools & technologies including Composite Application Framework (CAF), Master Data Management (MDM) and aSOA Grid. The Cordys platform and its cutting-edge Cloud technology empowers customers to dramatically improve the speed of change, fundamentally altering the way they innovate their Business Operations to achieve a true customer-centric philosophy. Global 2000 companies worldwide have selected Cordys to achieve business performance improvements such as increased productivity, reduced time to market, higher security and faster response to ever-changing market demands.

Copyright © 2009 Cordys Software B.V. All rights reserved. The information in this document is the intellectual and proprietary property of Cordys Software B.V, and is protected by intellectual property laws and international intellectual property treaties.

Situational Apps_whitepaper_28052009

Rapid
solutions to
streamline your
business processes

CORDYS