# A Strategic View on Software Reengineering

**By *Andrea Tomasini <me@andreat.de>***
http://www.andreat.de

## *Introduction*

The purpose of this paperwork is to give to the reader a very high level perception of what the problems are inside the software industry: what were the mistakes made in the past, and what are the available innovations that could transform a "potential business" into a "successful process centric business".

## *Executive Summary*

Today's way of developing software products is clearly not optimised and controllable, due to the lack of technology stability and some other technical factors the whole software industry is continuously experiencing changes as well as failures. Are we really sure that the fact that a project fails depends on the "technology" we are using? Indeed technology is important but the reasons for project failures – still more than 66% of projects fail, more than 82% end over schedule, and less than 52% with the promised feature set - has nothing to do with technology: bad requirements management, scarce or absent documentation, bad change management, bad project management… (see "Standish Group CHAOS Report 2004" for more detailed information).

The solution for a better project governance is to focus on the *process* as the combination of *people*, *procedures* and *infrastructure* (*tools)* that should become the way through which the business regains control and starts to grow in a structured way. Michael Hammer[i] said two very important things on how business is going nowadays: *"doing business is easier for the customer but so much more complex to the supplier. It requires getting it right first time"* and *"As work becomes more demanding and more complex, process becomes absolutely essential."*

## *Preamble: The Software Revolution*

The software industry is experiencing a sort of redemption: the introduction of software into the manufacturing industry, some years ago, was clearly perceived as an accident by the automatic control and the electronic equipment engineering personnel.

Nowadays even if many people are unaware, all sort of appliances such as washing machines, telephones, TVs, and watches, have their analog and mechanical parts replaced by CPUs and software. The long term effect of these changes is hard to predict, but it appears of paramount importance to provide high quality, secure and robust software.

If Software Engineering is compared to other disciplines such as Chemical or Civil Engineering, it appears to be less mature and more unstable. It could simply be a matter of timing. The shift from the "Creative Art" phase to the "Engineering" phase of Software

Engineering took only 50 years[1], compared to the slow evolutionary development observed in both Chemical and Civil Engineering. The dynamics of the Software Industry and the heavy demands placed upon it to find immediate solutions in today's fast moving markets means that the solutions are often not constructed with sound engineered principles.

The revolution that started in the early '90's thanks to the Internet diffusion and the E-Business phenomenon drew a lot of attention to the Software Industry, and for the first time software has been perceived as the primary factor responsible for the delivery of value added services to the customer. It has become clear that software offers a more "effective" way to build flexible and easily configurable "machines".

## The Value Added discriminator

The sudden growth of the "Service" industry has moved to an even higher level the importance of the Software Business, this is due primarily to the fact that there is virtually no "physical" constraint and that much of the E-Business can be done over the internet. This has birthed niche internet service businesses which offer to their clients instantaneous response and customer satisfaction.

[1] According to literature the transitions that a discipline should follow to became "engineered" are: Art-Production-Commerce-Science-Engineering, software appeared in its first forms around 1950 (1954).

At the beginning, most of the E-Business initiatives were porting of the existing services over a new channel (the Internet), but after a while the business itself steered towards a complete new way of serving and acquiring customers, the new path was focused around the "Customer". Their needs now drive the business. They determine more and more what is possible for any enterprise to do. Such dynamic forces need the Service industry to be more process centric in order to meet the deluge of demands from new and growing customers.

*"The customer does not see or care about the company's organisational structure or its management philosophies, or its detailed tasks. The customer sees only the company's products and services, the end result of its processes. A process perspective requires that we start with customers and what they want from us, and work backward from there."* – (Michael Hammer)

Nowadays the key to success is to be able to deliver end-to-end solutions to the final customer, therefore the "Customer Centric" approach causes the Software industry to face a completely new order of complexity and a necessity to market within timelines that were inconceivable in the past years. The problem of delivering complex, multilayered, distributed and integrated solutions to the market has become the everyday job of Software Engineers who engage their talent and energy without having the tools, infrastructure and methods required to achieve that task.

On one side, the service industry has never experienced the need for standardisation or for defined processes in order to be able to effectively deliver solutions to their customers. In "traditional" industries such as the Mechanical, Chemical, Civil manufacturing industries there is a deep culture of processes and defined standards, and there is a high level of attention to the quality and safety of the products. The software processes in these more mature industries are generally coupled with real production processes, and therefore the flexibility and the freedom that the software development processes may have is very constrained by the production requirements[2]. On the other side, this fast evolution has stretched the software development industry to face problems and complexities that often exceed its skill levels.

## The Industrial Approach

At the beginning of the decade, the attempt to transform the Software Engineering discipline into a real structured and regulated one has been finalised by trying to apply to the software development process the same constraints that have been typically designed for the supply and manufacturing chain. The major problem in the software industry - beside the lack of consolidated standards and the relative immaturity of most of the technology (which is continuously evolving) – is that software activities are performed by creative knowledge workers, and therefore the responsibility for the quality of the product is delegated to the people inside the development project. The success of projects is still mostly governed by the individuals skills and capabilities of the team, and is therefore often unpredictable[3].

Due to the lack of standardised interfaces, there is a loss of information when the development is carried on to successive steps, because too often different people with different skill sets and different roles are in charge of the various phases. Moreover in a "chain" approach it can be very hard to keep track of the "big-picture" of a project and of the fundamental business objectives. The project is broken-down into small parts, and responsibility for the final objectives is split into the *relations*[4] between the parts, and therefore for the most part it

[2] Refers to production environment in which real tangible goods are produced, like for examples: appliances, medicines, mechanical parts, electro domestics. The way in which processes are tied to the "production" needs is that there are some fixed integration and testing phases that have to be scheduled according to the state of the production of the physical parts of the product.

[3] Standish Group (http://www.standishgroup.com) in its famous chronicle report CHAOS on the project failure and success - published every two years since 1994 – in 2004 has reported that even if there are significant improvements since 1994, still more than 66% of projects fail and time overruns have significantly increased to 82% from a low of 63% in the year 2000. In addition, this year's research shows only 52% of required features and functions make it to the released product. This compares with 67% in the year 2000.

[4] From a formal perspective, if a problem is broken down into small problems to be more easy to solve, the complexity of the singles problems decreases compared to the whole, but the complexity of the relations between the problems increases. A good approach is always to consider a break-down in which the complexity between relations and problems is equal.

is lost. One of the major causes for project failure is that final products do not meet the market or customer expectations, it happens not because of poor quality in the product itself, but more tragically because of poor requirements management, lack of communication, bad change management, and scarce or absent documentation to name but a few. (see "Standish Group CHAOS Report 2004" for more detailed information).

Different software engineering practices have attempted to separate information into contiguous slots, but this often has led to a complete loss of transparency and visibility over the whole project objectives and the real customers needs. Moreover, to file people into folders and assign them with specific *tasks,* forcing them to follow procedures and rules, has still not led to an improvement in productivity.

Due to the nature of software engineering, it will be always more closely aligned to a scientific form of art rather than to a rigid and consolidated industrial discipline. The standardisation process for software has to follow a different path from other engineering disciplines. This is the main reason why the concept of modelling the software development process into a set of well defined and disciplined activities, assigned to specific characterised roles, is failing time and time again. There is a need for more flexibility and tailoring capabilities in order to succeed, as evidenced by the continuously growing attention to the Agile movement and some Open Source models: the SEI[5] (Software Engineering Institute) itself has an entirely new CMMI approach - "… I prefer the CMMI to the Capability Maturity Model for Software (SW-CMM) due to its flexibility, particularly in the measurement and analysis and project monitoring and control areas…"[6], an expert interviewed in a piloting phase - is moving toward this direction.

## *The New Era, a Paradigm Shift*

It is quite clear that nowadays the tendency is to reduce these procedures and rules to the minimum needed to have a good balance between control and performance. The main reason is that all the actors that are involved in the development of a project want to know what is happening around them and what the customer needs are, they want to know how they can bring their contribution to the project development. Above all else they want and need motivation.

"*A disregard for processes had been built into the structure and culture of traditional organisations whose founding premise, Adam Smith's ideas of specialisation of labour, was a rejection of process. This premise argued that success was based on fragmenting processes into simple tasks and then resolutely focusing on them. Whereas attention to process created stresses that could not be papered*

[5] The Software Engineering Institute at Carnegie & Mellon University is the formal founder and promoter of the CMM – Capability Maturity Model and all its flavours http://www.sei.cmu.edu .

[6] Extracted from the SEI Annual Report 2003 – Software Engineering Process Management.

*over such as who would have control over the newly recognised processes consisting of diverse tasks crossing organisational boundaries which imperilled the protected domains of functional managers."* – (Michael Hammer)

People are still the key to success in all software projects. The processes are good and have increased and improved the "average" level of quality throughout an organisation and throughout the whole market. However those actors involved in the development of a project who are true knowledge workers need to be stimulated to do repeatable and often mundane tasks.

"*We do not provide unsatisfactory service because our employees are hostile to customers, but because no employee has the information and the perspective needed to explain to customers the status of the process whose results they await. We suffer from high costs not because our individual tasks are expensive, but because we employ people to ensure that the results of tasks are combined into a form that can be delivered to customers."* – (Michael Hammer)

Software engineering is rapidly evolving from a craftsman "structured" form of art into a well managed discipline, but care must be taken that the things that have to be changed are not machinery, like in a production chain, but human beings. The biggest challenge for all the companies is to face *Change*, to be able to make the shift from the present reality to the new paradigm of a process centric organisation more focused on people getting it right first time.

## The Process Centric Organisation

Bringing people into the centre of an organisation and letting them be responsible for the success and being involved in the whole process of completing an objective is the key to the concept of "Process Centric Organisation".

"Process" means: *"An organised group of related activities that together create a result of value to customers"* – (Michael Hammer)

Companies exist to create value, profit, and achieve agreed results, everybody in a company should know and therefore contribute to the company success. Customers are most interested in themselves, they want solutions not products, in these days *"doing business is easier for the customer but so much more complex to the supplier. It requires getting it right first time"* – (Michael Hammer)

Due to the increased level of complexity also at the business level, and to the higher risk of losing a customer through project errors, today it is more important than ever to establish a very well controlled and managed project environment. *"As work becomes more demanding and more complex, process becomes absolutely essential."* – (Michael Hammer).

Software Development is a highly brain intensive activity, it is not a repeatable mechanical discipline that can be bound to a very strict and formalised process. Therefore it is fundamental to

keep the minds of those involved in the process active and focused on the global objectives to be reached. The key to success is to use the contribution of as many minds as possible to find the most suitable solution to a problem, human beings are not machines. It is of key importance to compare and share ideas with others to have confirmation and a wider view on all the aspects of a specific problem.

## Emphasise the Human Factor

Software is all about "projecting" and managing the build process. The manufacturing industrial chain for the most part is automated. In the software industry the primary cost is to build the first and only "piece" of a product. One of its characteristics is that the end result is dependent on heavy upfront Research & Development costs and less on production and distribution costs.

Information in this critical phase is therefore vital to the success of the project. The focus is totally unbalanced toward the creative and design phase, it is therefore of vital importance that the *key individuals are able to access all available data and they should be encouraged to think "out of the box".* Knowledge workers should not be constrained, but should be able to see the big picture and yet be capable of detailing the tasks required to complete the overall goal. Humans are used to analyse reality by patterns that they have learned during their life experience. They first get the big picture, then compare it to a previously registered experience, and then define the pattern with which to scale down to details. This is repeated until the level of detail is good enough to be faced with direct capabilities.

In software development the real machinery is "brain". Brains should be connected together in order to share as much information as possible to achieve results in a shorter time, and they should be left free to choose the most natural and intuitive way to solve problems, and not be forced to follow specific paths.

## Key Concepts: Transparency, Traceability, Responsibility and Commitment

"*People who are aligned around a common goal but lack the discipline of a well defined process will go nowhere, albeit together. Likewise, the best designed process has no chance of survival when people aren't aligned around the process and it's goals.*" - (Michael Hammer)

In order to maximise brain power to find the most effective way to solve problems, the work environment should be designed for total *Transparency, Traceability* together with *Accountability,* and the company should be able to allocate to people clear *Responsibilities* and demand total *Commitment*. It is in such an Open Environment – where transparency exists – that the involved people will offer their cooperation and commitment to reach the final goal together. In terms of productivity, this kind of approach maximises the potential of all the team members and ensures that the work will be shared and divided to serve and to help everyone to reach the goal faster and with better results.

## *The Business Perspective*

"*Discovering innovative ways to improve business processes is now recognised as the path to business agility and competitive advantage.*" – (Howard Smith [ii])

The business impact of such a radical mind-shift depends mainly on the level of acceptance of the involved people. The responsibility for the day by day leading of activities will be in the hands of the team, and will be supervised by the company management who will have a new role, to ease the work of the team by preparing the right environment to let the team succeed and complete the projects within time and budget.

### Control costs and maximise resource usage

To control costs and maximise resource usage is a key factor to succeed from a business perspective in any project based environment. The key to control in a discipline like Software Development is to give responsibility to people in terms of estimating activities' efforts and completing them with the expected level of quality.

"*So much of what we call Management today consists in making it difficult for people to work.*" – (Peter Drucker)

The key to success is to transform the people into managers of themselves, by applying Management by Objectives. People who are empowered, made responsible and are motivated, have the ability to define their path towards fixed objectives, and will work harder to reach the objectives in the required timeframe.

"*Management by objectives works if you first think through your objectives. Ninety percent of the time you haven't.*" – (Peter Drucker [iii])

One of the greatest benefits of a motivated and productive workforce is the willingness of the team's members to help each other to achieve a common result. The fact that people will see clearly the big picture of the whole project in which they are cooperating will empower their ability to participate in someone else's tasks.

### React in time and effectively to changes

The biggest problem in software projects stems from the *Change Management*. Most of the time these changes are handled outside or separate from the core development team. Surely the ones creating and developing the product are in the best position to understand how to address the changes, and the impact that those changes might have on the overall project. The answer is clear, but the problem is that it is difficult to establish a stable process in order to avoid loss of continuity in the development activities. In Open Source community people often work on more than one project at one time, may be they act as analyst in one project, they are developers in another one and they are fixing bugs and changes on a third one, and indeed they are productive and effective.

### Process control for better quality

It is proved that processes are of key importance in order to raise the

average level of quality of project management inside a company. It is also known that *Quality is Not for Free* or as Crosby pointed out in a famous joke: *"… Quality is for free: by mistake the cost of quality is measured in terms of costs of non compliancy, therefore is the cost to do things in the wrong way…"* - (Philip B. Crosby[iv]) therefore the right balance between process formalisation and its tailoring is the recipe for success. It is mandatory to have an environment in place that will let the people involved in the processes rapidly tailor the way they work without losing control and quality according to project needs. The flexibility of the environment is mandatory also to guarantee that the process will fulfil the project needs and not vice versa.

"*Quality in a product or service is not what the supplier puts in. It is what the customer gets out and is willing to pay for. A product is not quality because it is hard to make or costs a lot of money, as manufacturers typically believe. This is incompetence. Customers pay only for what is of use to them and gives them value. Nothing else constitutes quality."* – (Peter Drucker)

[i] Dr. Michael Hammer is the originator and leading exponent of the concept of reengineering. The author of the seminal Harvard Business Review article "Reengineering Work: Don't Automate, Obliterate," Dr. Hammer was named by Time to its first list of America's 25 most influential individuals.

[ii] Howard Smith is Chief Technology Officer (Europe) of Computer Sciences Corporation (CSC) and co-chair of the Business Process Management Initiative (BPMI.org). With more than 24 years in the IT industry, he is a sought after speaker and advisor. His work in predicting and shaping technology at the intersection with business led him to take an active role in the development and application of the third wave. He is currently researching the application of business process management to corporate sustainability, innovation and growth, for which he has global research and development responsibility at CSC.

[iii] Peter Drucker has written 35 books in all: 15 books deal with management, including the landmark books *The Practice of Management* and *The Effective Executive*; 16 cover society, economics, and politics; 2 are novels; and 1 is a collection of autobiographical essays. His most recent book, *Managing in the Next Society*, was published in fall 2002. Peter Drucker also served as a regular columnist for The Wall Street Journal from 1975 to 1995 and has contributed essays and articles to numerous publications, including the Harvard Business Review, The Atlantic Monthly, and The Economist. Throughout his career, he has consulted with dozens of organizations - ranging from the world's largest corporations to entrepreneurial start-ups and various government and non-profit agencies. Experts in the worlds of business and academia regard Peter Drucker as the founding father of the study of management (http://www.peter-drucker.com ).

[iv] Philip Bayard Crosby 1926 – 2001. Philip Bayard Crosby, a business philosopher, who had over 40 years of hands on management experience. As an author Mr. Crosby published fourteen books, all of which were best sellers. His first book, Quality is Free, has been credited with beginning the quality revolution in the U.S. and Europe. It has sold over 2.5 million copies and has been translated into 15 languages.