A new production model for the competition in a progressively more integrated world.

Eduardo Bellani[1]

[1] Departamento de informática e estatística – Centro Tecnológico –
Universidade Federal de Santa Catarina (UFSC)
CEP 88040-900 – Campus Universitário – Caixa Postal 476 – Florianópolis – SC – Brazil

ebellani@gmail.com

Abstract. *The present work is about an analysis of the Getting Real philosophy as a model for the construction and management of businesses that are based primarily on the production, research and development of intangible goods. It's also a review of the modern social, technological and economic situation of the business environment and how the model mentioned above could be an alternative for the economical survival under the conditions derived from this current situation.*

1. Introduction

The following paper is not about the analysis of a particular methodology in comparison with another as an academic exercise per se, but a study on how some practices and philosophies affect the survival chances of business, and how an entrepreneur could increase her chances with them.

Several cultural, economic and technological aspects contrast with the inherent needs of the entrepreneur, specially those that are involved in the high risk of the technological fronts. There's still the impression that in this context a rigid and well defined process is necessary. Sure there are cases where this necessity is real and well based[Yeniceri 2007], but in the cases where the business is mainly about intangible goods like software, that kind of thought could mean the death of the enterprise, due mainly to the lack of capacity to adapt to new and constant changing in the environment.

The challenge then is to adopt a philosophy that itself embraces the change in the environment as a constant through the life cycle of the productive process, and by doing that be more prepared to a world that is getting increasingly complex and integrated because of the convergence of several technological advances and revolutions. In the next chapters there will be a description of such philosophy, known as Getting Real, how to implement it and a brief description of why to do it.

2. Development

The world in it's most diverse aspects is going through a process of flattening motivated by profound political, technological, social and economical changes[Friedman 2005]. Nobody really knows for sure if those tendencies are going to keep on going through the future, but what can be said with certainty is that they already developed an environment that is very integrated and connected, and where the change of the business rules is becoming the norm.

2.1. Why

Despite the imprevisibility of the future, there is a growing body of evidence that we are at the beginning of a very accelerated curve of revolutions and advances that will keep on feeding the already accelerated pace of change in how humans interact with each other and their environment. Kurzweil in [Kurzweil 2005] compiled several well grounded points, and we need to cite only three to illustrate the point:

- The rate of paradigm shift (technical innovation) is accelerating, right now doubling every decade.
- The power (price-performance, speed, capacity, and bandwidth) of information technologies is growing exponentially at an even faster pace, now doubling about every year. This principle applies to a wide range of measures, including the amount of human knowledge.
- For information technologies, there is a second level of exponential growth: that is, exponential growth in the rate of exponential growth (the exponent). The reason: as a technology becomes more cost effective, more resources are deployed toward its advancement, so the rate of exponential growth increases over time. For example, the computer industry in the 1940s consisted of a handful of now historically important projects. Today total revenue in the computer industry is more than one trillion dollars, so research and development budgets are comparably higher.

The consequences of ignoring the new world trend could be daunting to the success of businesses and startups alike, as Grantham et al. makes clear on Corporate Agility: A Revolutionary New Model for Competing in a Flat World[Grantham et al. 2007]:

> Reducing fixed operating costs in order to increase corporate agility is the central business challenge of the twenty-first century. In an ever-more-dynamic global economy, with production routinely outsourced into the least costly labor markets, and innovation and design no longer the exclusive province of North America, Europe, and the Pacific Rim, businesses must move from a fixed-cost to a variable-cost business model in order to remain competitive and to increase their ability to react quickly to changing conditions. Human resources will soon become as scarce as petroleum, and so any company with plans for growth had better get started developing programs to attract and retain qualified, engaged employees. *Without institutionalizing innovation, industry leaders today will find themselves laggards tomorrow.*

2.2. What

To better deal with this changing environment, the author suggests a new approach that has been built on the experience of developing web based software systems, the Getting Real approach. While there is nothing really innovative on what it advocates, the fusion of several strategies and a no nonsense way of communicating them are certainly a refresh approach on the crowded world of development methodologies. More specifically, it is an amalgam of agile methodologies with an open source management style to deal with clients and developers, plus the embracing of change as a business life constant.

On *Getting Really Real* from Boyd[Boyd 2006] we can see a more detailed description of the Getting Real work:

> Getting real is as radical a break with conventional wisdom about building software as is imaginable. As an analogy, consider the impact of cargo containers on commercial shipping: instead of hundreds of stevadores carrying boxes of cargo, and manually stowing them in the ships' cargo holds, today pre-loaded containers are lifted from trucks or trains, by cranes, and stacked on the ships like enormous lego blocks. A handful of trained crane operators can literally do the work formerly accomplished by hundreds. And the ship stays at the dock for a hundredth of the time it would have under the former approach to loading and unloading.
>
> Note that this productivity boost is not due to the specific individuals involved being smarter, or better trained: it is the outcome of a completely different approach to the problem, and the application of better building blocks in the processes involved. getting real is the same: throw out the preconceptions of striving to get hundreds to take small steps faster, and instead use a smaller group who see the process differently. In the case of getting real, however, the process insights rely on better programming tools – like Ruby on Rails – as well as a pervasive and zen-like certainty that less is more, small is the new big, and that anything that detracts from the act of delivering quality software to the user should be deep-sixed.

2.2.1. Agile methodologies

The agile methodologies are a response for the more bureaucratic and rigid ways of thinking of the past of software development, which were based on engineering methods, they represent a middle way between a too rigid process and a too lax one. Fowler makes a comparison between the old way of building software and the agile way on his article The New Methodology[Fowler 2001]:

- *Agile methods are adaptive rather than predictive.* Engineering methods tend to try to plan out a large part of the software process in great detail for a long span of time, this works well until things change. So their nature is to resist change. The agile methods, however, welcome change. They try to be processes that adapt and thrive on change, even to the point of changing themselves.
- *Agile methods are people-oriented rather than process-oriented.* The goal of engineering methods is to define a process that will work well whoever happens to be using it. Agile methods assert that no process will ever make up the skill of the development team, so the role of a process is to support the development team in their work.

2.2.2. Open source management

The greatest impact the open source world does on the environment for businesses is made clear on Graham's article *What Business can learn from Open Source*[Graham 2005]:

> But the biggest thing business has to learn from open source is not about Linux or Firefox, but about the forces that produced them.

Still according to the same reference[Graham 2005], there are 3 major points in open source that the business world should pay attention to:

- **The power of the amateur**
  People work harder and better in what they like, and even better at something they love. The emotional commitment of human beings to the work they produce is not reflected in the current traditional business structure. Business tends to focus on professionalism, with the main goal of work being money and all the connotations of formality and detachment from it. Using a famous example:

  > It's not that Microsoft isn't trying. They know controlling the browser is one of the keys to retaining their monopoly. The problem is the same they face in operating systems: they can't pay people enough to build something better than a group of inspired hackers will build for free.
- **Workplaces**

  > Another thing blogs and open source software have in common is that they're often made by people working at home. That may not seem surprising. But it should be. It's the architectural equivalent of a home-made aircraft shooting down an F-18. Companies spend millions to build office buildings for a single purpose: to be a place to work. And yet people working in their own homes, which aren't even designed to be workplaces, end up being more productive.

  What that means essentially is that the professionalism that the traditional model focuses so much on is also present on the workplaces, transforming them into places that hinder productivity.
- **Bottom-up**
  On the open source way of doing things, the ideas themselves prove their values, so they tend to bubble from everywhere, mainly from the bottom, mainly because there is more bottom than up.

  > ...people make what they want, and the best stuff prevails.

  The above way is remarkably similar to the market economy, whereas the corporate world behaves internally more like a communist state, with orders from the top that get distributed from a chain of command, and there is very little input in the inverse direction.

2.2.3. The simplicity factor

The simplicity as a value of production is one of the most essential aspects of the Getting Real philosophy, and that is another aspect of the philosophy that is extracted from the agile methodologies, fact that can be seen in the 2 Xp slogans ”Do the Simplest Thing that Could Possibly Work” and ”You Aren’t Going to Need It” (known as YAGNI)[Fowler 2001], which are just different ways of translating the concept of simplicity.

There are several bases to justify the adoption of the simplicity as a production principle, but in the root they all have the fact that in duplicating the software base (or any other intangible goods), the complexity factor does not follow this increase linearly, but exponentially.[37signals 2006] Other reasons, still according to [37signals 2006] are:

> Less software is easier to manage.
> Less software reduces your codebase and that means
> Less maintenance busywork (and a happier staff).
> Less software lowers your cost of change so you can adapt quickly.
> You can change your mind without having to change boatloads of code.
> Less software results in fewer bugs.
> Less software means less support.

2.3. How

According to Grantham et al. in [Grantham et al. 2007] there are 3 great challenges facing businesses today:

1. Reducing fixed operating costs
2. Confronting the coming talent shortage
3. Institutionalizing innovation

Given the complexity and variety of possible businesses and startups, it is not possible to come up with a detailed plan for each and every situation. But by dealing with each of these challenges using the Getting Real approach perhaps it is possible to get some pointers on how to implement the philosophy in a great number of locations.

2.3.1. Costs

The cut in costs is the most straightforward challenge, and the one most easily visible for people across the spectrum of business. The Getting Real way would be to make this pervarsive on the production, and that would be done through the simplicity principle.

Reducing the amount of code required for the software, and the software itself has a direct impact in programming hours, but also in maintenance, support, management and several things. As the Getting Real mentions [37signals 2006]

> The more massive an object, the more energy is required to change its direction. It’s as true in the business world as it is in the physical world.

That does not apply only to code, but to everything, as said before. So, instead of buying a complicated communication solution, why not just use VoIP and IM, and perhaps save for a weekly or monthly get together with the team to increase trust and communication, or perhaps replace those old servers?

2.3.2. Talent and people

The talent gap is a dangerous one for companies these days for several reasons. One is the aging and retiring of skilled workers, specially in developed countries. Another is the exponential growth of the needs in the global market and the stabilization of the supply of qualified workers, mainly because of a severe inadequate educational system that does not prepare the students to be able to adapt and be responsible for their own learning and development.[Negroponte 2000]

But the truly crucial aspect of it is that great talent is rare. And as Graham says so eloquently[Graham 2004]:

> The top 5% of programmers probably write 99% of the good software.

But how to attract talent? According to Getting Real[37signals 2006] the best way is to go beyond the traditional ways, through the open source world:

> The typical method of hiring for technical positions - based on degrees, resumés, etc. - is silly in a lot of ways. Does it really matter where someone's degree is from or their gpa? Can you really trust a resumé or a reference? ... By definition, involvement in open source requires at least some passion. Otherwise why would this person spend free time sitting in front of a screen? The amount of open source involvement often shows how much a candidate truly cares about programming.

This position is reflected almost perfectly by Graham [Graham 2004]

> It seems surprising to me that any employer would be reluctant to let hackers work on open-source projects. At Viaweb, we would have been reluctant to hire anyone who didn't. When we interviewed programmers, the main thing we cared about was what kind of software they wrote in their spare time. You can't do anything really well unless you love it, and if you love to hack you'll inevitably be working on projects of your own.

2.3.3. Innovation

This is perhaps the most crucial element in these new times where businesses are immersed. But the failure of most of the businesses world in producing innovative services and products, as well as adopting methodologies and practices that would stimulate innovation, is well documented.[Grantham et al. 2007]

The case of the Microsoft corporation is an emblematic one. Initially a very agile and innovative startup, with its growth and subsequent monopolization, its capacity of creation and innovation, as well as its influence is being slowly extinguished. One of the major problems is that Microsoft's business model is based around a paradigm (the desktop) that is being replaced fast by another (the web). And its own efforts to adapt are being self defeated because of the well grounded fear of seeing its cash cow dissipate[Graham 2007].

In the last case there's a very important lesson, that Graham explicits in [Graham 2005]:

> Live by the channel, die by the channel: if you depend on an oligopoly, you sink into bad habits that are hard to overcome when you suddenly get competition.

Here the simplicity factor from the Getting Real is the key. The constrains created by it are one of the major contributors to the promotion of creativity and flexibility.[37signals 2006]

> There's never enough to go around. Not enough time. Not enough money. Not enough people. That's a good thing. Instead of freaking out about these constraints, embrace them. Let them guide you. Constraints drive innovation and force focus. Instead of trying to remove them, use them to your advantage. ... Run on limited resources and you'll be forced to reckon with constraints earlier and more intensely. And that's a good thing. Constraints drive innovation.

3. Conclusion

The point in this work is to make it clear that any entrepreneur has ahead a new world paradigm, one rich in difficulties and opportunities. There are innumerable new frontiers and the access to new markets, peoples and ideas has never been so great in the history of humankind. On the other hand, the global competition and pressure for results, combined with the simultaneous revolutions in all fields on human endeavor make this environment very unstable and susceptible to profound and frequent changes.

The author hopes that the philosophy presented along this article can help to provide the tools that are necessary to make use of this new landscape, instead of being overwhelmed by it. Those tools, allied with a bit of courage may be the keys to entrepreneurs that have the taste and vision to try the new.

References

37signals (2006). *Getting Real*. `http://gettingreal.37signals.com`, 1 edition.

Boyd, S. (2006). Getting really real.

Fowler, M. (2001). Is design dead? pages 3–17.

Friedman, T. L. (2005). *The World is Flat*.

Graham, P. (2004). How to make wealth.

Graham, P. (2005). What business can learn from open source.

Graham, P. (2007). Microsoft is dead.

Grantham, C. E., Ware, J. P., and Williamson, C. (2007). *Corporate Agility A Revolutionary New Model for Competing in a Flat World*. AMACOM.

Kurzweil, R. (2005). *The singularity is near*. Phoenix.

Negroponte, N. (2000). From being digital to digital beings. *IBM Systems Journal*, 39(3&4):417–.

Yeniceri, G. (2007). Getting real or not.