

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE GRADUAÇÃO EM SISTEMAS DE
INFORMAÇÃO
CENTRO TECNOLÓGICO - DEPARTAMENTO DE
INFORMÁTICA E ESTATÍSTICA.**

Eduardo Bellani

**Um novo modelo de produção para a competição em
um mundo cada vez mais integrado**

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

José Eduardo De Lucca - Mestre em Ciências da Computação - UFSC

Florianópolis, Maio de 2008

Um novo modelo de produção para a competição em um mundo cada vez mais integrado

Eduardo Bellani

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Sistemas de Informação, área de concentração Sistemas de Informação e aprovado em sua forma final pelo Programa de Graduação em Sistemas de Informação.

Prof. Maria Marta Leite

Coordenadora do Curso

Banca Examinadora

José Eduardo De Lucca - Mestre em Ciências da Computação - UFSC

Achilles Colombo Prudêncio

Djali Avelino Valois

Não se preocupe se as pessoas vão roubar suas idéias. Se elas forem realmente boas você terá que fazer elas serem engolidas.

Ofereço este trabalho a todos que me desejam bem, e ao
momento no qual sempre viveremos ...

Agradecimentos

As pessoas que me mostraram o mundo e me ajudaram a achar a força que existe em todos nós, meus pais. Ao meu irmão por limitar minha arrogância a níveis toleráveis(ou quase). A minha namorada, por ser uma pessoa que sempre está disposta a escutar. Ao meus amigos e mentores, pelo conhecimento e apoio. A todas as oportunidades que tive até aqui, e a imensidão que chamamos de vida.

Sumário

Sumário	vi
Lista de Figuras	viii
Glossário	ix
Resumo	x
Abstract	xi
1 Introdução	1
1.1 Objetivos	2
1.1.1 Geral	2
1.1.2 Específicos	2
2 Revisão conceitual e contextualização	4
2.1 Modelo exemplo: <i>getting real</i>	4
2.1.1 Metodologias Ágeis	6
2.1.1.1 <i>Extreme Programming</i>	8
2.1.1.2 <i>Scrum</i>	8
2.1.2 O princípio da simplicidade	10
2.1.3 Código Aberto	11

2.1.4	Desenvolvimento Global de Software	12
2.1.5	<i>Telecommuting</i>	13
2.2	O modelo Tradicional	15
2.3	O mundo (quase) plano	16
2.4	Um novo continente: O Cyber Espaço	17
3	Proposta de trabalho	19
3.1	Por que ser ágil?	19
3.2	Como ser ágil e inovador	21
3.2.1	Redução de custos de operação	21
3.2.2	A disputa por talento de qualidade	23
3.2.3	Institucionalização da inovação	24
3.3	Comparações	26
3.4	Caso CNXS-ELC-GiveZooks	28
3.4.1	Introdução	28
3.4.2	Descrição	29
3.4.3	Comparativos	32
4	Conclusão	34
4.1	Revisão	34
4.2	Dificuldades encontradas	35
4.3	Pensamentos finais	36
	Referências Bibliográficas	37
A	Contrato	40

Lista de Figuras

2.1	Ilustração da metodologia Xp	9
2.2	Ilustração da metodologia Scrum	9
2.3	Esquema do formato de trabalho <i>telecommuting</i>	14
2.4	Ilustração do modelo <i>waterfall</i>	16
3.1	Demonstração da exponencialidade da tecnologia	20
3.2	Ilustração sobre o equilíbrio entre projeto e refatoração	28
3.3	Ilustração do caso CNXS-ELC-Givezooks	30
A.1	Contrato de trabalho, página 1	41
A.2	Contrato de trabalho, página 2	42
A.3	Contrato de trabalho, página 3	43
A.4	Contrato de trabalho, página 4	44
A.5	Contrato de trabalho, página 5	45
A.6	Contrato de trabalho, página 6	46
A.7	Contrato de trabalho, página 7	47
A.8	Contrato de trabalho, página 8	48

Glossario

bens intangíveis - O termo é usado aqui para definir toda produção cuja existência é apenas digital e/ou abstrata. Software é um exemplo, assim como música, livros, poemas, análises, jornalismo, etc.

Getting real - Metodologia de desenvolvimento e filosofia de administração baseada nas experiências da empresa 37signals

open source - Filosofia que prega a abertura dos projetos e designs envolvidos com determinado produto para a comunidade, de forma a alavanca-la para o desenvolvimento do mesmo.

waterfall - Método de análise tradicional rígido e sequencial, com etapas claras e bem definidas.

GSD - Global Software Development, ou seja, desenvolvimento global de software .

telecommuting - É um tipo de trabalho no qual os empregados dispõem de uma flexibilidade em local e hora de expediente.

startup - Empresa em seus primeiros estágios, geralmente do ramo tecnológico e afins.

turnover - Frequência de troca de pessoal, ou seja, o grau de ocorrência de troca de mão de obra, por causa de demissão, usualmente voluntária.

Resumo

O presente trabalho trata da análise de um modelo representado pela filosofia *Getting Real* para a construção e administração de negócios que se baseiem primariamente na produção, pesquisa e desenvolvimento de bens intangíveis. Expõe-se também uma revisão da atual situação social, tecnológica e econômica do ambiente em que tais empreendimentos se encontram e como o modelo mencionado acima pode ser uma alternativa para a sobrevivência nestas condições.

Palavras Chave: getting real, ágil, empreendedorismo, ciberespaço, inovação, mercados globais.

Abstract

The present work is about an analysis of a model represented by the Getting Real philosophy for the construction and management of business that are based primarily on the production, research and development of intangible goods. It's also exposed a review of the current social, technological and economic situation of the environment in which those entrepreneurships are located and how the model mentioned above could be an alternative for the survival on these conditions.

Key words: getting real, agile, entrepreneurship, cyberspace, global markets, innovation.

Capítulo 1

Introdução

A visão aqui exposta não engloba uma determinada metodologia de desenvolvimento em detrimento de outra apenas pelo exercício acadêmico em si, e sim é um estudo sobre como determinadas práticas e filosofias de desenvolvimento afetam a sobrevivência de empreendimentos, e como um empreendedor pode aumentar drasticamente suas chances de sucesso com elas.

Vários aspectos culturais e econômicos contrastam com as necessidades inerentes de ser empreendedor, especialmente em um negócio de alto risco e imprevisibilidade como o do ramo tecnológico, onde há uma constante mudança nas regras do jogo e nas expectativas dos consumidores e desenvolvedores. Ainda há a impressão que neste contexto se faz necessário regras e processos bem definidos e engessados. Realmente, existem casos nos quais estes são necessários(YENICERI, 2007) porém no caso de empresas onde se desenvolve e utiliza-se bens intangíveis em alto grau, eles podem significar a morte da empresa.

A utilização de uma filosofia moderna de desenvolvimento e relacionamento, adaptada as novas realidades e ao mundo planificado pela crescente utilização de meios cada vez mais sofisticados e baratos de comunicação, para o desenvolvimento de software(e outros bens intangíveis), com o objetivo de burlar as dificuldades inerentes ao mercado interno e local de cada indivíduo, e ter acesso a mercados novos de uma maneira eficaz, onde a existência de inúmeras realidades e potenciais pode ser alavancado. É disto que trata-se fundamentalmente este trabalho.

Apesar das idéias nas quais este trabalho se baseia estarem sendo implementadas em vários empreendimentos e terem adquirido certa fama e reputação nos últimos tempos, devemos ter em mente que alguns conceitos que tratamos aqui não tem mais de uma década de existência. Vale aqui o conselho dado no clássico *The Psychology of Computer Programming*(WEINBERG, 1985), de Gerald M. Weinberg:

Por causa que este é um campo novo, não foi sempre possível apoiar certas idéias com evidências "científicas". De fato, muitas das visões neste trabalho são meramente as opiniões do autor, frequentemente opiniões fortes, mas baseadas em nada que não observações feitas ao longo dos anos. Sem dúvida, várias destas opiniões estão erradas, bem como estão muitas das idéias suportadas por mais evidências. Mas há um mundo de diferença entre uma idéia errada e uma estéril. ¹

1.1 Objetivos

1.1.1 Geral

Com este trabalho espera-se elucidar os benefícios que metodologias e filosofias de processo ágeis e flexíveis(exemplificadas neste caso pela *getting real*) podem trazer a empreendimentos e a produção de software e de produção intelectual em geral e, por contraste, as dificuldades trazidas por um pensamento tradicional e engessado num mundo cada vez mais dinâmico e integrado.

1.1.2 Específicos

- Analisar a filosofia *getting real* e o que ela acarreta, mostrando sua credibilidade para a produção de software competitivo.

¹*Because this is a new field-a new way of looking at familiar things- it has not always been possible to support certain ideas with "scientific"evidence. Indeed, many of the views in the book are merely the author's opinions-often strong opinions, but not based on anything better than persona observation over many years. No doubt, many of these opinions are just plain wrong, as are many of the ideas supported by more evidence. But there is a world of difference between a wrong idea and a sterile one. If any reader takes issue with something expressed here, my fondest hope is that he will set up an experiment and prove me wrong.*

- Criar uma análise do ambiente atual para as empresas, principalmente aquelas que lidam e produzem diretamente bens intangíveis.
- Descrever se há a necessidade de um modelo como o *getting real* para as empresas no ambiente descrito no objetivo acima, e em caso afirmativo, e como implantá-lo.

Capítulo 2

Revisão conceitual e contextualização

2.1 Modelo exemplo: *getting real*

Getting real em si é uma combinação de conceitos de software e de produção e gerenciamento deste como pode ser visto no artigo de revisão *Getting Really Real* de Boyd(BOYD, 2006):

Getting real é ao mesmo tempo um manual de operação sobre como construir ótimo software, e uma condenação compreensiva dos preconceitos estúpidos que permeiam a maior parte do desenvolvimento de software. ¹

A obra é uma visão inovadora baseada em conceitos ágeis e em uma percepção que o tamanho de uma equipe pode atrapalhar, e muito a conclusão de um projeto. Em suma, ela contém muito mais do que um modelo de negócios, mas também dicas, conselhos, e caminhos para o desenvolvimento e manutenção de software(ou quaisquer outros tipos de bens intangíveis que possam seguir o mesmo modelo de negócios), seguindo um novo paradigma. Na mesma revisão citada acima(BOYD, 2006) podemos encontrar um trecho que revela mais sobre as características da obra:

Getting real é uma quebra tão radical quanto possível da sabedoria convencional sobre construir software. Considere o impacto de contêineres de

¹*Getting real is at once an operating manual on building great software, and a comprehensive condemnation of the startlingly dumb preconceptions that animate most software development.*

carga e despachos comerciais: ao invés de centenas de estivadores carregando caixas de carga, e manualmente empilhando-as nos navios, os containers de hoje são erguidos de caminhões ou trens por guindastes e empilhados em navios como blocos de lego enormes. Um punhado de operadores de guindaste pode literalmente fazer o trabalho antes feito por centenas. E o navio fica nas docas por um centésimo do tempo que demoraria anteriormente.

Observe que o aumento de produtividade não é devido a indivíduos serem mais inteligentes ou melhor treinados: é o resultado de uma abordagem completamente diferente para o problema, e a aplicação de blocos melhores no processo.

O espírito no *getting real* é o mesmo: jogue fora os preconceitos de tentar fazer com que centenas de passos sejam mais rápidos, e use um grupo menor que vê o processo de forma diferente. No caso do *getting real*, o processo se baseia em ferramentas melhores como *Ruby on Rails* e também em uma certeza que menos é mais, menor é o novo maior, e que qualquer coisa que fuja do ato de entregar software de qualidade ao usuário deveria ser descartado.²

Mais especificamente, *getting real* é uma fusão do modelo de metodologias ágeis com uma abordagem de código aberto para lidar com seus clientes e com seus desenvolvedores, ao mesmo tempo que se abraça a inovação como um estilo de vida gerencial.

²*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 stevedores 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.*

2.1.1 Metodologias Ágeis

Freqüentemente a atividade de criação de software é caótica e pautada pelo *code and fix*, ou seja, grandes períodos de desenvolvimento seguido por períodos de manutenção e correção de erros. Esse caminho tende a levar, em projetos que não minúsculos, ao atraso cada vez maior e a perda conseqüente de controle sobre o processo de criação. A solução para este problema está na adoção de metodologias capazes de lidar com a complexidade e mudança de uma maneira mais prática.(FOWLER, 2005)

A inspiração tradicional para a maioria das metodologias de produção são disciplinas como a engenharia civil ou mecânica. Nestas ocasiões os engenheiros elaboram planos meticulosos, onde os requisitos e outros detalhes são descritos em detalhes, e depois os entregam a trabalhadores para executar este plano. Tais trabalhadores não precisam ter treinamento e habilidades intelectuais avançadas(apesar de terem capacidades físicas e manuais bastante desenvolvidas). Esse processo assume algumas características que não se traduzem ao processo de desenvolvimento de software muito bem, como descrito por Fowler em seu famoso artigo *The New Methodology*(FOWLER, 2005):

Quando se constrói uma ponte, o custo do esforço de design é aproximadamente 10% do total, com o resto sendo construção. Em software, a quantidade de tempo gasta em codificação é muito, muito menor. ... para um projeto de larga escala, apenas 15% do projeto é codificação e testes unitários - uma inversão quase perfeita da proporção em construção de pontes.

Mesmo que você considere toda a etapa de testes como sendo construção, o design ainda representa apenas 50% do trabalho. Isso levanta uma questão importante sobre a natureza do design em software comparado com seu papel em outros ramos da engenharia. ³

Estas metodologias existem a um bom tempo, e se provaram pesadas, ao ponto de diminuir o ritmo de produção com o seu foco em processos burocráticos.

³When you build a bridge, the cost of the design effort is about 10% of the job, with the rest being construction. In software the amount of time spent in coding is much, much less ... for a large project, only 15% of the project is code and unit test, an almost perfect reversal of the bridge building ratios. Even if you lump in all testing as part of construction, then design is still 50% of the work. This raises an important question about the nature of design in software compared to its role in other branches of engineering.

As metodologias ágeis foram a resposta a esta condição, e elas se encontram na metade do caminho entre uma metodologia carregada e burocrática e o caos do *code and fix*. As diferenças entre as metodologias tradicionais e as ágeis podem ser sumarizadas em:(FOWLER, 2005)

- Metodologias ágeis são adaptativas ao invés de predeterminantes. Metodologias de engenharia tendem a tentar planejar uma grande parte do processo de desenvolvimento detalhadamente por um longo período de tempo. Isso funciona bem até as coisas mudarem. Então a natureza de tais métodos é a de resistir à mudança. Para os métodos ágeis, entretanto, mudanças são bem-vindas. Eles tentam ser processos que se adaptam e se fortalecem com as mudanças, até mesmo ao ponto de se auto-modificarem.
- Métodos ágeis são orientados a pessoas ao invés de serem orientados a processos. O objetivo dos métodos de engenharia é de definir um processo que irá funcionar bem, independentemente de quem os estiverem utilizando. Métodos ágeis afirmam que nenhum processo jamais será equivalente a habilidade da equipe de desenvolvimento. Portanto, o papel do processo é dar suporte a equipe de desenvolvimento e seu trabalho.

A primeira reação das pessoas aos processos e técnicas ágeis é uma presunção que eles podem funcionar para projetos de pequena escala, porém eles nunca funcionariam para projetos ditos reais. Isso simplesmente não é verdade, como Ambler cita em *Introduction to Test Driven Design (TDD)*

... relata ter trabalhado em um sistema em Smalltalk que adotou uma abordagem completamente voltada a testes que durou 4 anos e 40 anos-pessoa de esforço, resultando em 250,000 linhas de código funcional e 250,000 linhas de código de teste. Apesar de existirem projetos maiores, está claro que desenvolvimento baseado em testes funciona para projetos de bom tamanho.

4

⁴... reports working on a Smalltalk system taking a completely test-driven approach which took 4 years and 40 person years of effort, resulting in 250,000 lines of functional code and 250,000 lines of test code. ... Although there are larger systems out there ... it is clear that TDD works for good-sized systems.

Para aumentar o grau de explanação sobre metodologias ágeis, a seguir serão apresentados dois exemplos clássicos e significativos destas metodologias, uma de desenvolvimento de software, a chamada *Extreme Programming*, e outra de gerenciamento, conhecida como *Scrum*:

2.1.1.1 *Extreme Programming*

Extreme Programming, também chamado de Xp, é uma metodologia que enfatiza diversos aspectos do processo evolutivo e técnicas para aprimora-lo. Na figura 2.1 encontra-se demonstrado os 3 grandes laços iterativos e incrementais do Xp:

No laço exterior em azul é o laço de disponibilizações, onde se interage com o mundo externo e deixa entrar-se requisitos em uma maneira controlada ao mesmo tempo que software funcional é enviado para fora.

No laço do meio em vermelho temos as interações do time, onde se realiza as atividades diárias.

No laço interno em verde temos as atividades de codificação, onde dois programadores, trabalhando em pares produzem código testado unitariamente.

2.1.1.2 *Scrum*

Scrum é uma maneira de organizar a equipe de produção através de laços cíclicos e reuniões focadas, como a figura 2.2 pode melhor demonstrar.

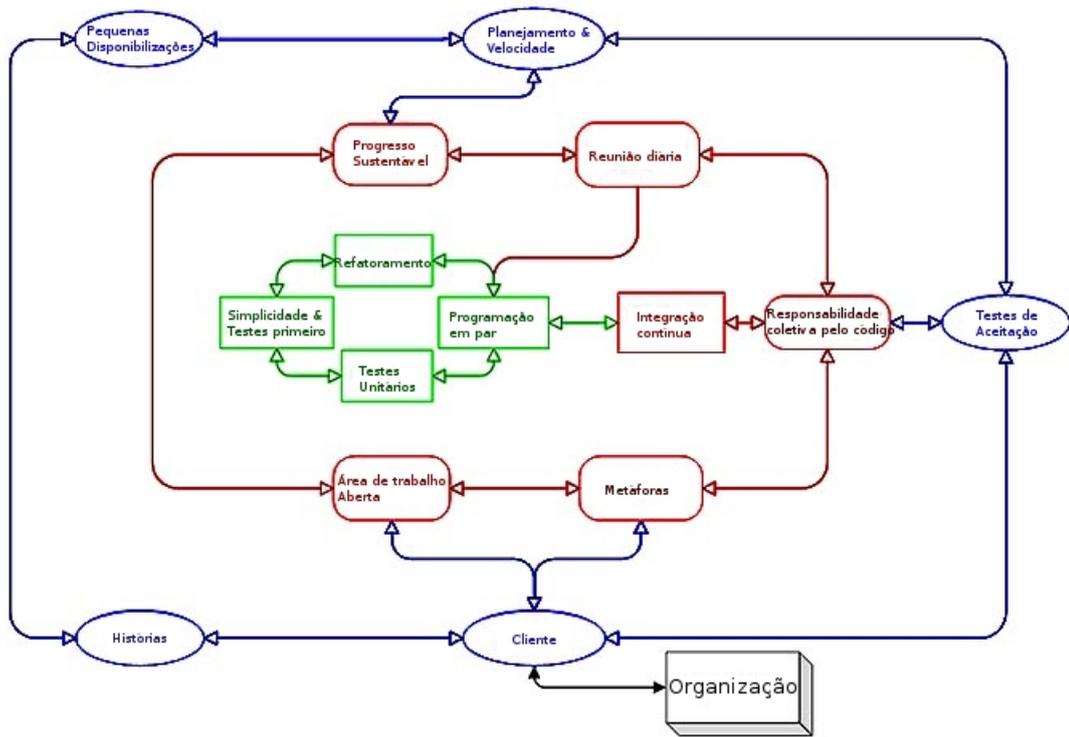


Figura 2.1: Ilustração da metodologia Xp

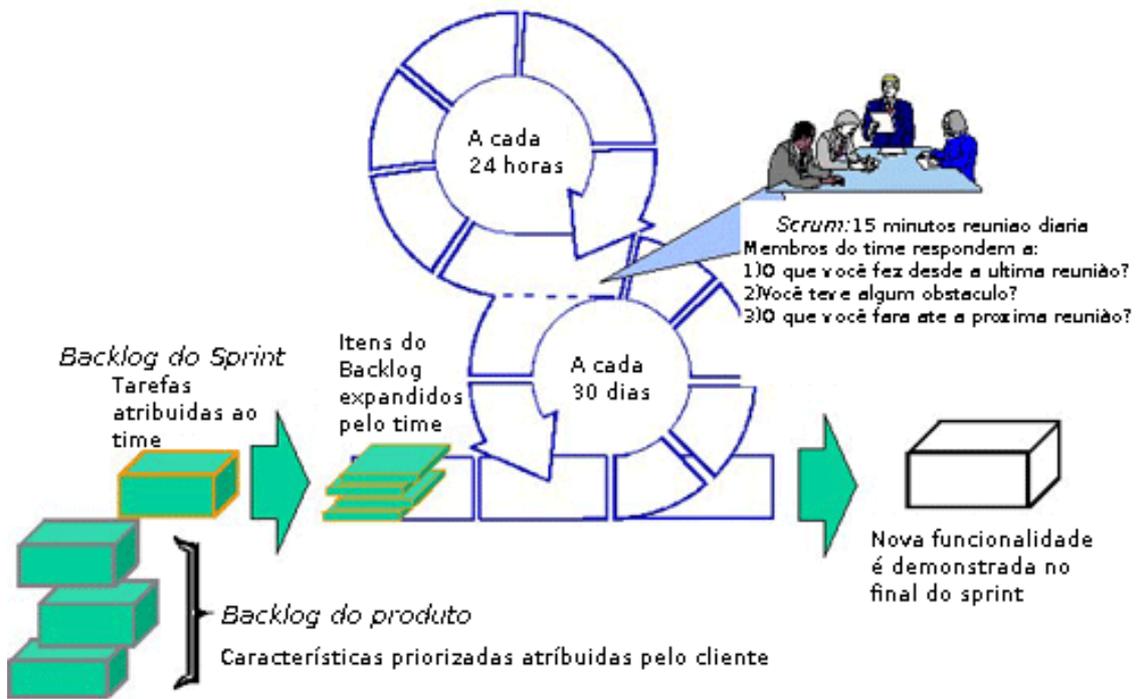


Figura 2.2: Ilustração da metodologia Scrum

2.1.2 O princípio da simplicidade

A simplicidade como valor de produção é um fator essencial da filosofia *getting real* e também é um importante aspecto das metodologias ágeis, tanto que dois slogans destas metodologias, "Faça a coisa mais simples com que você possa trabalhar" e "Você não irá precisar disto" são simplesmente formas diferentes de traduzir o conceito de simplicidade. (FOWLER, 2001)

Existem diversas justificativas para a proposta do princípio da simplicidade, e elas acabam variando do fato que cada vez que se duplica a base de software (ou qualquer outro capital puramente intelectual) o fator complexidade não acompanha linearmente este aumento, mas sim exponencialmente. (37SIGNALS, 2006) Outras razões, ainda de acordo com (37SIGNALS, 2006) são:

- Menos software é mais fácil de administrar.
- Menos software reduz sua base de código e isso significa menos manutenção (e mais tempo para o desenvolvimento)
- Menos software reduz seu custo de mudança, facilitando a rápida adaptação.
- Menos software significa menos erros e menos suporte ao usuário.

A tradução deste princípio para a prática de produção se dá pela não adição de qualquer coisa a base de software que não seja imediatamente necessário, mesmo que no futuro seja. Construir flexibilidade, componentes e arcabouços que apenas resolvem necessidades projetadas significa pagar um custo inicial na expectativa que este se pagará no futuro. Esta idéia é vista como uma parte chave no projeto tradicional de software. (FOWLER, 2001)

Numa metodologia ágil, o ponto é deixar estas estruturas crescerem naturalmente a partir da necessidade da produção, pois do contrário passa-se a gastar esforço que poderia ser gasto em algo necessário hoje em algo que será necessário amanhã (e num ambiente dinâmico a certeza da necessidade nem sempre está certa). Além disso adiciona-se complexidade (mais código), que, ao menos temporariamente, será inútil. (FOWLER, 2001)

2.1.3 Código Aberto

O maior impacto, tanto no mundo em geral como a abrangência da importância do código aberto nas metodologias ágeis e no *getting real* é melhor compreendido no trecho de *What Business can learn from Open Source* de Graham (GRAHAM, 2005):

Mas a maior coisa que os negócios tem a aprender com o código aberto não é sobre o Linux ou o Firefox, mas sobre as forças que os produziram ⁵

O termo código aberto não é usado aqui como uma descrição do tipo de licença que o código fonte vai usar, mas sim o tipo de organização entre os desenvolvedores, o código e os usuários. No *getting real*, usa-se um ciclo de distribuições que se assemelha ao padrão imortalizado pelo movimento de código aberto, mais especificamente o SO Linux: distribuições frequentes são uma parte crítica do modelo de desenvolvimento do Linux. A maioria dos desenvolvedores acreditavam que era uma má política para projetos não triviais, por causa que versões antecipadas por definição são defeituosas e você não quer gastar a paciência dos seus usuários. (RAYMOND, 1999)

Essa política funciona bem por causa que quanto mais usuários, maior a quantidade de stress que seu sistema vai receber, e conseqüentemente os erros tendem a aparecer mais. Este fato pode ser melhor exemplificado como a "Lei de Linus", da obra *The Cathedral and the Bazaar* de Raymond (RAYMOND, 1999):

Dado uma base grande o suficiente de testadores e co-desenvolvedores, quase todos os problemas serão caracterizados rapidamente e a solução óbvia para alguém. Dado um número suficiente de olhos, todos os erros são superficiais.

6

Outra relação interessante é entre os próprios desenvolvedores, e suas relações de poder. Este modelo prega maior confiança mútua entre os programadores

⁵*But the biggest thing business has to learn from open source is not about Linux or Firefox, but about the forces that produced them.*

⁶*Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone. Given enough eyeballs, all bugs are shallow.*

e a responsabilidade individual de cada um com o código e com os outros. Este fenômeno reflete ainda mais nas comunidades que são formadas em torno de certos códigos, comunidades estas que possuem não apenas programadores, mas usuários, entusiastas e artistas, todos ligados pelo interesse em comum no sucesso do produto.

2.1.4 Desenvolvimento Global de Software

Uma consequência da adaptação do modo código aberto de desenvolvimento é o potencial para o desenvolvimento global de software (GSD, na sigla original). Desenvolvimento global de software, também conhecido como distribuído, significa que o desenvolvimento de software é espalhado por diversos locais que podem estar localizados em diversos países ou continentes. (SOUZA,)

Este processo é explicitado na filosofia da própria obra *Getting Real* (37SIGNALS, 2006), pois o modelo comporta claramente esta situação:

Quando a *37signals* estava construindo o Basecamp, tínhamos várias limitações: ... uma diferença de 7 horas (David estava programando na Dinamarca, enquanto o resto de nós estávamos nos Estados Unidos) ⁷

Código aberto é fator importantíssimo em desenvolvimento global de software, não apenas por sua maneira de organizar a equipe, mas para aqueles que querem contratar pessoal técnico. Com ele, você pode observar o trabalho de alguém e suas contribuições, boas ou más, através de um intervalo extenso de tempo. Segundo (HANSSON, 2007), o significado do currículo ou de antecedentes em determinadas faculdades dão muito menos segurança em relação ao nível de informação que se pode obter em um projeto de código aberto no qual o indivíduo participou. O livro *Getting Real* (37SIGNALS, 2006) também faz um argumento similar:

O típico método de contratação - baseado em títulos, currículos, etc - é bobo em vários sentidos. Realmente importa de onde o diploma de alguém é ou a sua nota de vestibular? Você pode realmente confiar num currículo? ⁸

⁷When *37signals* was building *Basecamp*, we had plenty of limitations. We had: ... A 7-hour time difference (David was doing the programming in Denmark, the rest of us were in the States)

⁸The typical method of hiring for technical positions - based on degrees, resums, etc. - is silly in a lot

Dentre as motivações que estimulam a adoção destes processos, de acordo com (SOUZA,), a participação em alguns mercados, requisição de regulações governamentais, pode fazer sentido de um ponto de vista de mercado mover parte da equipe para perto de determinado pólo tecnológico e, em particular importância no contexto globalizado, a busca por uma equipe técnica competitiva e competente.

O foco no na competência técnica, desenvoltura social e motivação, que pode ser facilmente observado através de uma análise do código aberto, acima de vários outros fatores, levam a adoção de práticas específicas não apenas na contratação, mas no desenvolvimento de software de uma maneira na qual times que muitas vezes se distribuem geograficamente ou estão distantes do seu mercado alvo possam prosperar.

2.1.5 *Telecommuting*

Como consequência do GSD, desenvolve-se um novo tipo de relacionamento trabalhista conhecido como "*telecommuting*". Uma definição acessível pode ser encontrada no artigo *The Good, the Bad, and the Unknown About Telecommuting: Meta-Analysis of Psychological Mediators and Individual Consequences* de Gajendran, Ravi e Harrison.(GAJENDRAN; HARRISON, 2007)

Telecommuting é um tipo alternativo de acordo trabalhista onde empregados fazem suas tarefas em locais que não o escritório central, ao menos por uma parte de seu turno, usando meios eletrônicos para interagir com outros dentro e fora da organização.⁹

Tal acerto tende a trazer uma série de consequências, tanto positivas quanto negativas, para uma organização e para os empregados da mesma. O resultado geral tende a ser positivo, como mostra um recente estudo sobre o assunto: *Telecommuting* tem um claro fator positivo: pequenos porém favoráveis efeitos na autonomia percebida, intenção de turnover e stress, além disso também não possui nenhum claro efeito

of ways. Does it really matter where someone's degree is from or their gpa? Can you really trust a resumé or a reference?

⁹*Telecommuting is an alternative work arrangement in which employees perform tasks elsewhere that are normally done in a primary or central workplace, for at least some portion of their work schedule, using electronic media to interact with others inside and outside the organization.*

danoso na qualidade das relações no trabalho ou nas oportunidades de carreira percebidas(GAJENDRAN; HARRISON, 2007)

Ou seja, é um acordo vantajoso para ambas as partes, onde o desenvolvedor pode se sentir mais independentemente e produtivo, impactando positivamente em várias características, e em uma relação patrão empregado, pode aumentar a confiança entre as partes, tal afirmação encontra respaldo no mesmo artigo citado acima(GAJENDRAN; HARRISON, 2007):

... organizações que permitem aos empregados a flexibilidade de trabalhar de casa estão dando um sinal positivo, visivelmente demonstrando sua confiança e suporte aos empregados. Este sinal por sua vez deveria gerar uma aumento do comprometimento psicológico e uma tendência menor de demissão ¹⁰

Pode-se observar um resumo destas interações na figura 2.3.

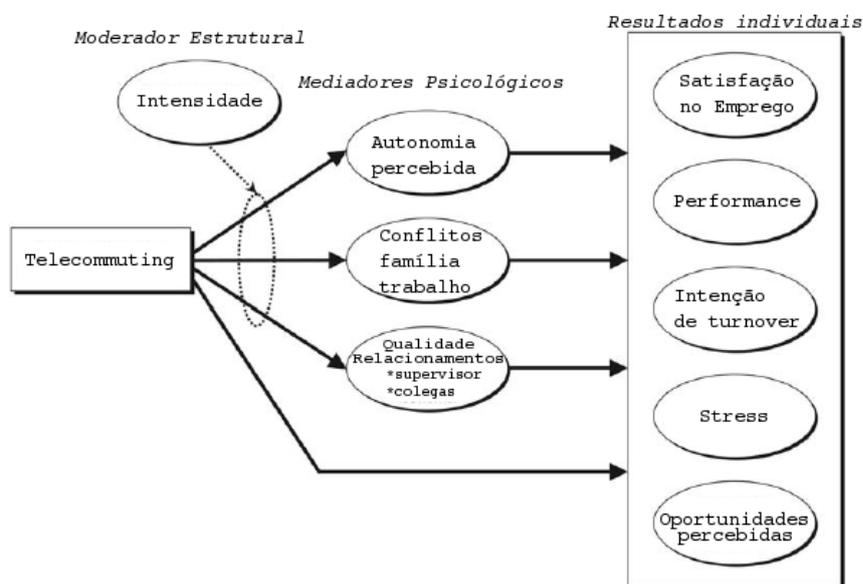


Figura 2.3: Esquema do formato de trabalho *telecommuting*

¹⁰... organizations providing employees with the flexibility to work from home are providing a positive signal, visibly demonstrating their trust and support for employees' well-being.

2.2 O modelo Tradicional

Segue-se, para efeito de contraponto, a análise do framework RUP em sua implementação mais comum adotada em projetos que falham com ele, que agrega e incita o modelo de desenvolvimento *waterfall* tradicional. (BITTNER, 2001)

De fato, o RUP em si pode ser uma miríade de possibilidades, como deixa claro Fowler no artigo *The New Methodology*(FOWLER, 2005):

De fato, o RUP em si pode ser uma miríade de possibilidades, como deixa claro Martin Fowler: "Minha experiência com RUP é que se problema é sua infinita variabilidade. Eu já vi descrições de usagem do RUP que variam desde *waterfall* rígido com "análise de iterações"até uma metodologia perfeitamente ágil. ... isso levou a um resultado onde as pessoas podem fazer praticamente qualquer coisa e chamar isso de RUP, resultado este que faz com que o termo perca seu significado. ¹¹

Como se pode observar na figura 2.4 o modelo *waterfall* passa 6 estágios definidos e seqüenciais:análise de requisitos, design, implementação, testes e validação, integração e manutenção. Esse pensamento data dos anos 60, quando a construção de software era algo novo e caótico e algum modelo tinha que ser adotado, então os cientistas e programadores escolheram o que lhes era mais familiar, de diversos outros ramos da engenharia(FOWLER, 2005).

Para seguir este modelo, os passos deve ser seguidos de maneira puramente linear. Uma fase só pode ser iniciada quando a subsequente for completamente concluída e aperfeiçoada.

A idéia central deste modelo é o *big design up front*, ou seja, a idéia de que quanto mais tempo gasto inicialmente antes de começarem o desenvolvimento do serviço em si irá economizar um tempo exponencialmente Também há a ênfase na produção de meta-informação e documentação sobre as diferentes partes do sistema antes da construção dos mesmos.

¹¹*My experience with RUP is that it's problem is its infinite variability. I've seen descriptions of RUP usage that range from rigid waterfall with "analysis iterations"to picture perfect agile. ... led to a result where people can do just about anything and call it RUP - resulting in RUP being a meaningless phrase.*

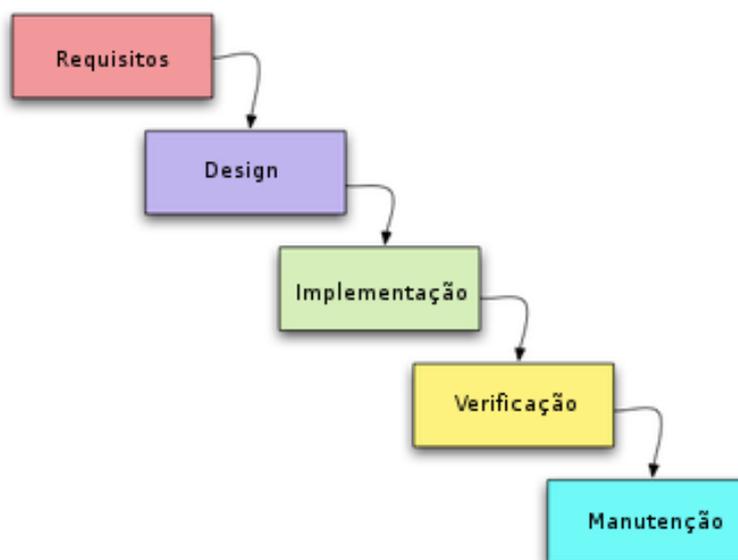


Figura 2.4: Ilustração do modelo *waterfall*

Seguindo este mesmo raciocínio, uma equipe tradicional seria extensa, com várias camadas de gerenciamento e módulos de produção, de atendimento, de controle de qualidade, documentação, testes, etc. Pode-se observar que o tamanho dos recursos humanos para se obter esses resultados começa a crescer junto com os requisitos não apenas da idéia inicial, mas também da metodologia adotada.

Pode-se observar que são criadas várias camadas entre o conceito inicial e o resultado final, e a cada etapa mais e mais informações sobre as etapas passadas vão sendo perdidas, e mais engessado e imóvel o processo vai se tornando. Apesar de amplamente criticado, esse tipo de atitude ainda permeia o pensamento de muitos desenvolvedores de software, gerentes de projeto e empreendedores nos dias de hoje. (LAPLANTE, 2004)

2.3 O mundo (quase) plano

O mundo em seus mais diversos aspectos está passando por um processo de planificação motivado por grandes e profundas mudanças políticas, científicas, tecnológicas e econômicas. Nominalmente, estas mudanças de acordo com (FRIEDMAN, 2005) são:

1. As mudanças sócio-econômicas que se seguiram ao colapso do muro de Berlim
2. A crescente acessibilidade da internet
3. O crescimento de IA limitada, e sua capacidade de fazer com que o software opere sem intervenção humana
4. Código aberto
5. Sub contratação de serviços
6. Realocação de produção e negócios de um país ou mesmo continente para outro
7. Um aumento no processo de logística e suprimento
8. Controle sobre a produção crítica
9. Informação globalizada
10. Conectividade cada vez mais barata

Ninguém pode adivinhar com certeza como este mundo cada vez mais globalizado e planejado vai ser no futuro longínquo, ou mesmo no próximo, mas um fator pode-se afirmar, ele será diferente do atual. E mesmo que as dez tendências listadas anteriormente não tenham planejado completamente o campo de competição, este está mais plano do que há uma década, com uma forte tendência de aceleração neste processo, e isto trará consigo severas dificuldades, competições e oportunidades (GRANTHAM; WARE; WILLIAMSON, 2007)

2.4 Um novo continente: O Cyber Espaço

O ciberespaço pode ser diferenciado da Internet, no sentido do ciberespaço ser um subconjunto desta. Mais especificamente, a internet como um todo pode ser considerada uma facilitadora da vida das pessoas, como exemplo pode-se citar o pagamento de contas sem sair de casa, ou obter notícias em tempo real sobre os mais diversos assuntos. Porém isto se restringe apenas a isto, fazer o existente ficar mais fácil

e acessível. O cotidiano se torna mais fácil e informado, mas não fundamentalmente diferente.(LESSIG, 2006).

O termo ciberespaço significa um uso social, econômico, político e tecnológico específico da grande rede. Por essa definição chegamos a conclusão que o ciberespaço nada mais é que o uso compreensível da internet e de meios de comunicação digitais de forma a revolucionar o meio com que nos comunicamos e interagimos. Outra maneira mais simples de ver esta distinção é feita pela comparação entre a definição dos termos em si. Enquanto a Internet é o nome da rede mundial de computadores, onde estes trocam informações de maneira geograficamente dispersa, o ciberespaço é um determinado conjunto de ligações humanas que se passa dentro deste domínio.

Lessig resume estas definições em sua obra *Code v2*(LESSIG, 2006):

Ciberespaço, por contraste, não é apenas fazer o cotidiano ser mais fácil. É fazê-lo ser diferente, ou talvez melhor. ... ele evoca meios de interação que não seriam possíveis antes. ¹²

¹²*Cyberspace, by contrast, is not just about making life easier. It is about making life different, or perhaps better. It is about making a different (or second) life. It evokes, or calls to life, ways of interacting that were not possible before.*

Capítulo 3

Proposta de trabalho

A seguir segue-se o centro do trabalho, uma tentativa de explicar como o mundo está passando por profundas e aceleradas mudanças, e como aplicar o modelo explicado no capítulo anterior para aumentar as chances de sucesso nestas condições.

Busca-se esclarecer estas conexões com uma análise final de um caso real onde este modelo é aplicado parcialmente, e uma análise comparativa da situação junto com a filosofia *Getting Real* para demonstrar pontualmente as áreas de intersecção.

Nestes tópicos encontra-se o pensamento fundamental do autor ao construir o trabalho, porém ainda usando-se referências para respaldar melhor estas opiniões.

3.1 Por que ser ágil?

Ninguém pode realmente prever com certeza suficiente o futuro das organizações e mercados, especialmente dado nossos tempos conturbados e dinâmicos. Porém uma certeza existe, ele será fundamentalmente diferente do que é hoje. Diversos fatores contribuem para tal, e pode-se citar alguns como Kurzweil em (KURZWEIL, 2005), e o mesmo torna mais fácil esta compreensão na figura 3.1.

- O aumento da frequência de troca de paradigmas na inovação técnica está acelerando, dobrando a cada década
- O poder da tecnologia da informação esta crescendo exponencialmente em um fator

ainda maior, dobrando a cada dois anos.

- Para este tipo de tecnologia, há ainda outro tipo de crescimento exponencial, a aceleração do ritmo de crescimento exponencial.

Considerando que estes são apenas os fatores mais proximamente conectados a atividade de TI dentre vários outros, pode-se ter idéia da profundidade da mudança que nosso mundo está passando, mudança esta que continuará a se aprofundar nas próximas décadas, deixando as pessoas cada vez mais conectadas e o ambiente cada vez mais globalizado e achatado, tanto em termos de informação quanto culturais e econômicos.

Todas essas conseqüências claramente não foram e provavelmente não serão compreendidas em curto prazo. O que podemos deduzir é que a mudança começa a se desenhar como um grande fator constante do jogo econômico-científico, e as metodologias do passado, por se prenderem e limitarem-se a meios de produção e adaptação que levava em conta um ambiente pouco dinâmico estão fadadas, senão a levarem seus adeptos ao fracasso, a pelo menos atrasá-los o suficiente para que sejam insignificantes em pouco tempo.

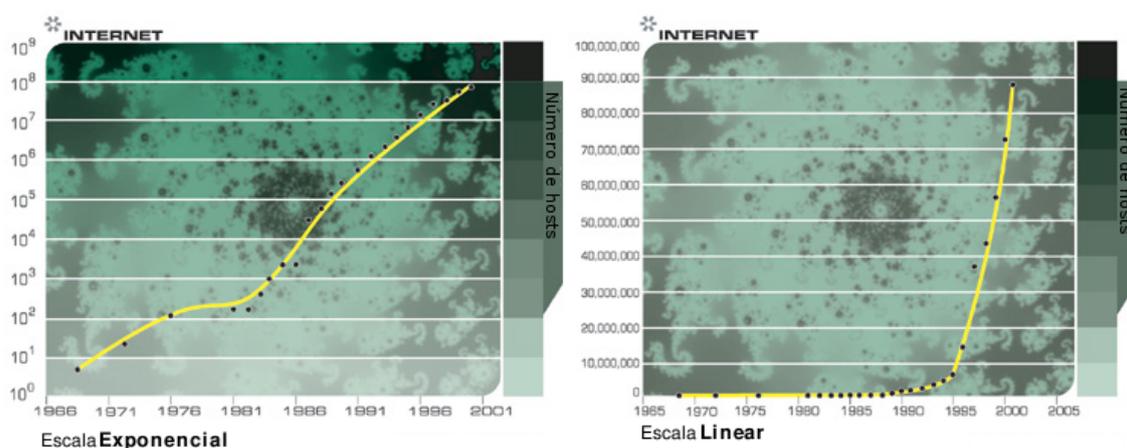


Figura 3.1: Demonstração da exponencialidade da tecnologia

Esta situação pode acarretar graves conseqüências para aqueles que não estão preparados para ela, como dito no livro *Corporate Agility: A Revolutionary New Model for Competing in a Flat World* de Grantham et al. (GRANTHAM; WARE; WILLIAMSON, 2007)

A redução de custos fixos de produção e operação para o incremento de agilidade no processo produtivo é o grande desafio do século vinte e um. Em um mercado global cada vez mais dinâmico ... empresas precisam mudar seus processos e modelos de negócio de um custo fixo para um modelo ágil variável de acordo com o ambiente para se manter competitivos e aumentar sua habilidade de reagir rapidamente a mudanças Sem institucionalizar a inovação, líderes de hoje se encontrarão entre os retardatários do amanhã. ¹

3.2 Como ser ágil e inovador

Segundo (GRANTHAM; WARE; WILLIAMSON, 2007) existem três grandes desafios para os empreendimentos nos dias de hoje, e se enfrentados corretamente podem estabelecer um arcabouço para a adaptação ao futuro.

- Redução de custos de operação
- A disputa por talento de qualidade
- Institucionalização da inovação

Todos os três são importantes objetivos a serem atingidos, pois mesmo individualmente eles contribuem para a criação de uma cultura capaz de lidar com mudanças e imprevistos. Ponto a ponto, podemos trabalhar melhor como atingí-los.

3.2.1 Redução de custos de operação

O corte de custos é o objetivo que mais se traduz diretamente para um pensamento tradicional. Esse pensamento deve ser distribuído em todos os aspectos, no sentido de se construir o mínimo possível(mas nunca menos que o elegante),

¹*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, ... 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.*

mesmo impondo barreiras no próprio crescimento de maneira a não se tornar vulnerável(37SIGNALS, 2006) . Ao perder e deixar de ganhar peso o empreendimento torna-se mais flexível, e em épocas de necessidade tem-se a capacidade de realocar recursos com uma facilidade maior do que com vários laços ligando-o a diversos investimentos.

Claro que o corte para uma corporação já em existência que não tenha implementado essa estratégia preventiva é mais difícil(essa é a principal vantagem de uma *startup* frente a corporações(GRAHAM, 2007)), mas uma mudança gradual para técnicas que valorizam um produto ou serviço de menor escala mas de diferenciação maior, empregados e sócios trabalhando de casa, uma liberdade maior na escolha de softwares e plataformas pode ser cultivada e implementada.

O custo de ativos pode ser bastante diminuído com a adoção de escritórios virtuais, ou seja, a internet como plataforma para a maior parte dos negócios e encontros, e apenas o essencial ser realizado no mundo físico. O uso desta mesma plataforma para pagamentos, reuniões, projeções e controle de pessoal acarreta o mesmo benefício. É o pensamento gerencial de código aberto junto ao desenvolvimento global de software.

A especialização e os limites auto impostos para o produto ou serviço, além da óbvia redução de custo e a concentração de esforços maior, também tem a vantagem de poder abrir nichos de mercado. Pode-se considerar o que Anderson diz em seu já famoso livro *The long tail : how endless choice is creating unlimited demand*(ANDERSON, 2007):

Os grandes sucessos agora competem com uma infinidade de nichos de mercado, de qualquer tamanho. E os consumidores estão cada vez mais favorecendo aquele com mais escolhas. A era de um tamanho serve em todos está acabando, e em seu lugar está algo novo, um mercado de multitudes.²

²*The hits now compete with an infinite number of niche markets, of any size. And consumers are increasingly favoring the one with the most choice. The era of one-size-fits-all is ending, and in its place is something new, a market of multitudes.*

3.2.2 A disputa por talento de qualidade

Este desafio começa a ser sentido entre as empresas que produzem primariamente bens intangíveis, por dois fatores principais. O primeiro deles é o envelhecimento da população e sua saída do mercado de trabalho, especialmente trabalhadores especializados e de alto valor agregado. A segunda é o próprio crescimento exponencial do mercado tecnológico e a estabilização da quantidade de mão de obra qualificada proveniente provavelmente de um sistema de educação mundial que não está preparado para os novos paradigmas que surgiram, onde os indivíduo tem de assumir grande parcela da responsabilidade por seu desenvolvimento.(NEGROPONTE, 2000)

O gerenciamento do talento e do conhecimento por detrás deste seria uma solução para o envelhecimento e aposentadoria de pessoal, bem como o turnover, como pode ser observado na seguinte análise de Weinberg no já citado clássico *The Psychology of Computer Programming*(WEINBERG, 1985):

... a habilidade de sobreviver aos seus membros é um dos pontos fortes da organização de um time, pois o trabalho a ser feito não é abandonado se alguma pessoa é removida. Essa habilidade provém, é claro, da interação entre os membros do time, por cause que esta permite que os objetivos e conhecimentos de um time sejam transmitidos para novos membros quando um antigo deixa o time. Nesse sentido, um projeto ou equipe de programação é como um rio que continua o mesmo rio ainda que suas aguas estejam em constante mudança. ³

Claro que nem todas as empresas tem o problema com aposentadorias, especialmente no caso de *startups*, onde a própria média de idade de dos membros já é a segurança contra suas aposentadorias(forá o fato que em geral todos os membros e empregados de uma *startups* são sócios ou tem alguma participação acionária)(GRAHAM, 2007).

³... *the ability to survive its members can be one of the strong points of a team organization, for the work to be done is not set aside if one person happens to be removed from the scene. This ability comes, of course, from the interaction among team members, because this interaction permits the goals and accomplishments of the team to be transmitted to new members and so remain when old members depart. In a sense, a programming project or team is like a river which remains the same river even though its water is undergoing constant change.*

O problema da educação e preparação é maior que o escopo deste trabalho, além de ser algo que deva ser abordado em conjunto com vários segmentos da sociedade. Há porém, algumas maneiras de contorná-lo. Se uma empresa for grande o suficiente ela pode tentar treinar pessoal até que a qualificação necessária seja obtida. O quanto que isso é possível é debatível de qualquer maneira.

Um meio eficiente é a contratação e formação de parcerias em um nível global, usando o ciberespaço como meio de atração, relacionamento e construção de parcerias. Desta forma uma empresa pode ter acesso não apenas a novos mercados, porém a novos talentos e conhecimentos, buscando no globo por indivíduos que, apesar da dificuldade do sistema de ensino conseguiram construir seu talento de maneira positiva.

A construção de um ambiente que propicie a atração de talento e a manutenção e permanência deste (que é tão importante quanto (GRANTHAM; WARE; WILLIAMSON, 2007)), é análogo a filosofia adotada pelo empreendimento, e pode-se analisar a semelhança com a metodologia ágil do *getting real* nos pontos a seguir, que segundo (HARDING, 2000) constituem a essência deste tipo de ambiente:

- Ver lealdade em contribuições e não em disposição de estabilidade
- Baixa tolerância com incompetência
- Recompensas baseadas no nível de performance ao invés da duração do emprego
- Esperar que os empregados tenham capacidade de construir seu próprio conhecimento e carreira

3.2.3 Institucionalização da inovação

Apesar deste objetivo não ser mais difícil de quantificar em relação aos anteriores, ela é percebida por sua notável falta em negócios que estão estagnados em suas áreas. Inovar significa fazer o novo, e essas companhias não apenas falham em construir novos serviços e produtos, mas também em adotar processos que promoveriam a inovação e agilidade.(GRANTHAM; WARE; WILLIAMSON, 2007)

Como exemplo, poderíamos citar a situação da indústria automobilística nos Estados Unidos, onde a decadência de lucros pode ser diretamente atribuída a falha na

percepção de mudanças profundas no mercado, onde consumidores atentos a mudanças climáticas e preços de combustíveis em rápida ascensão acabaram minando a demanda por carros grandes e com alto consumo. E foi exatamente o que elas mais produziram enquanto seus lucros despencavam.(GRANTHAM; WARE; WILLIAMSON, 2007)

A Microsoft é outra empresa em que se pode desenhar essa comparação. Inicialmente uma startup altamente ágil e inovadora, com o poder e o tamanho sua capacidade de criação e de influência foram se extinguindo. O golpe mais forte ainda está sendo sentido, pois a empresa encontra-se em uma situação onde seu modelo de negócio baseia-se em um paradigma(desktop) que está sendo rapidamente substituído por outro(web), e seus próprios esforços para mudar encontram resistência da própria empresa que se recusa a deixar seu mais lucrativo nicho diminuir.(GRAHAM, 2007) Nesse caso é fácil observar uma característica importante a se evitar, como Graham explicita tão claramente em *What Business can learn from Open Source*(GRAHAM, 2005):

Viva pelo canal, morra pelo canal: se você depende de um oligopólio, hábitos se enraízam e estes são difíceis de perder quando se encontra competição ⁴

Inovação, assim como qualquer forma de mudança, é fácil e aceitável apenas como um pensamento abstrato. Concretamente as pessoas tem uma resistência natural a mudanças, pois por natureza a inovação não se encaixa com o que a maioria pensa. Uma máxima foi extraída por Livingston diretamente das experiências profissionais de fundadores de empresas em *Founders at Work: Stories of Startups' Early Days*(LIVINGSTON, 2007):

Não se preocupe com se as pessoas vão roubar suas idéias. Se elas forem realmente boas você terá que fazer elas serem engolidas. ⁵

Aparentemente o melhor caminho para uma inovação parte do ato de escutar. Escutar a você mesmo e aos outros, suas necessidades conscientes e, se possível, inconscientes. Novamente no contexto de criadores de empreendimentos, Graham cita uma maneira de fazer isto no artigo *Ideas for a startup*(GRAHAM, 2005):

⁴*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.*

⁵*Don't worry about people stealing your ideas. If your ideas are any good, you'll have to ram them down people's throats.*

Ao invés de olhar para uma empresa de sucesso e traçar o caminho até os problemas que elas resolveram, olhe para um problema e imagine uma empresa para resolvê-lo. ⁶

Outra métrica útil nesta empreitada é trabalhar nos problemas mais difíceis e desafiadores positivamente que encontrar. Se você não está com um certo receio do quão desafiador vai ser o desafio, significa que não é o suficiente, pois a barreira não está alta o suficiente. A motivação para o aprendizado é útil, e uma barreira de dificuldade ajuda a ter menos concorrência. Se essa métrica é útil positivamente, ela também o é em sua direção negativa, pois os sinais da falta de necessidade de performance pessoal são observáveis na tentativa de parecer eficiente, e não ser eficiente. Ternos, enfeites e artefatos de luxo no escritório, etc. Este tipo de esforço não apenas não ajuda, mas atrapalha, ao criar situações artificiais para manter esta aparência.(LIVINGSTON, 2007).

E quanto as restrições e dificuldades, a obra *Getting Real*(37SIGNALS, 2006) faz uma síntese sobre o assunto:

Nunca há o suficiente. Nunca há tempo. Nunca há dinheiro. Nunca há pessoal. Isso é uma boa coisa. Ao invés de apavorar-se sobre estas limitações, abrace-as. Elas motivam inovação e forçam o foco. Ao invés de tentar removê-las, use-as em sua vantagem. ⁷

3.3 Comparações

Modelos de produção lineares não são de todo ruins, porém em um ambiente que requer mudanças constantes e dinamismo, como é o caso das *startups* via de regra, eles não são adequados. Por exemplo, em projetos onde os requisitos não tem a tendência de mudar com o tempo eles funcionam perfeitamente(projetos de engenharia civil são ótimos exemplos, softwares massificados ou para situações pouco complexas são outros). Fowler no artigo *The New Methodology*(FOWLER, 2005) deixa isto bem claro:

⁶*Instead of starting from companies and working back to the problems they solved, look for problems and imagine the company that might solve them.*

⁷*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.*

Existem alguns casos de desenvolvimento de software onde a previsibilidade é possível. Organizações como o grupo de software do ônibus espacial da NASA são um exemplo típico de como o desenvolvimento de software pode ser previsível. É necessário muita cerimônia, um time grande e requisitos estáveis. Existem projetos que são ônibus espaciais. Entretanto, eu não acredito que muitos sistemas se enquadrem nesta categoria. Para estes, você precisa de um tipo diferente de processo. ⁸

Apesar de sua utilidade dentro destes moldes, sua rigidez mostra-se falha em sua utilização dentro das condições encontradas em uma empresa, mais especificamente *startups* e empresas que dependem de bens intangíveis como matéria prima e como resultado de sua cadeia produtiva, onde o ambiente é caótico e as definições de produtividade raramente são encontradas em um sistema linear.(GRAHAM, 2007).

Num ambiente como este, é um erro presumir que é possível conseguir planejar todos os aspectos de um software antes de sua construção. (BITTNER, 2001)

Como vimos anteriormente, todas estas etapas(figura 2.4) estão mescladas no modelo *getting real*. Não há separação clara nem sequencial, e nem é possível construir uma. Usa-se os testes, os resultados destes e de informações ambientais e o bom senso para levantar mais requisitos, e realizar manutenção, e muito mais.

Além disso a rigidez imposta pelo modelo *waterfall* tende a se espalhar pelo pensamento da equipe, levando a conseqüências desastrosas, especialmente num mundo cada vez mais dinâmico. Enquanto o *getting Real* prega a utilização de relatórios somente quando estritamente necessários, o modelo tradicional obriga o desenvolvedor a fazê-los em várias ocasiões. Começa-se a confundir o mapa com o caminho.

Reuniões são freqüentes nos estágios iniciais do modelo *waterfall*, para se definir os requisitos. Reuniões como estas produzem pouquíssima informação útil, e em geral as pessoas concordam com coisas diferentes sem perceber (37SIGNALS, 2006).

⁸*Organizations such as NASA's space shuttle software group are a prime example of where software development can be predictable. It requires a lot of ceremony, plenty of time, a large team, and stable requirements. There are projects out there that are space shuttles. However I don't think much business software fits into that category. For this you need a different kind of process.*

Quaisquer modelos que não abracem a mudança como uma constante no decorrer das atividades deve ser encarado com ressalva pelos fundadores de uma *startup*. O desenvolvimento de software, ainda mais nos dias de hoje, é uma atividade dinâmica e sujeita a reinterpretações constantes. Com isto vemos que ter como fundamento a simplicidade e a preparação para mudanças freqüentes é uma forte vantagem para um projeto. Ambas as características são base da metodologia *getting real*(37SIGNALS, 2006).

Resumindo, tentar linearizar e engessar uma situação fundamentalmente complexa e dinâmica como a que se encontra uma *startup* típica, onde cada membro da equipe precisa assumir vários papéis em diversas camadas de desenvolvimento e administração seria um gasto inadmissível de energia ao tentar impor o modelo. Este dispêndio de atenção, motivação e recursos poderia ser a diferença entre a vida e a morte para a *startup*. Especialmente se considerarmos o gráfico 3.2 sobre as mudanças que a era da internet trouxe para a produção de software apresentado a seguir.(HIGHSMITH, 2000)

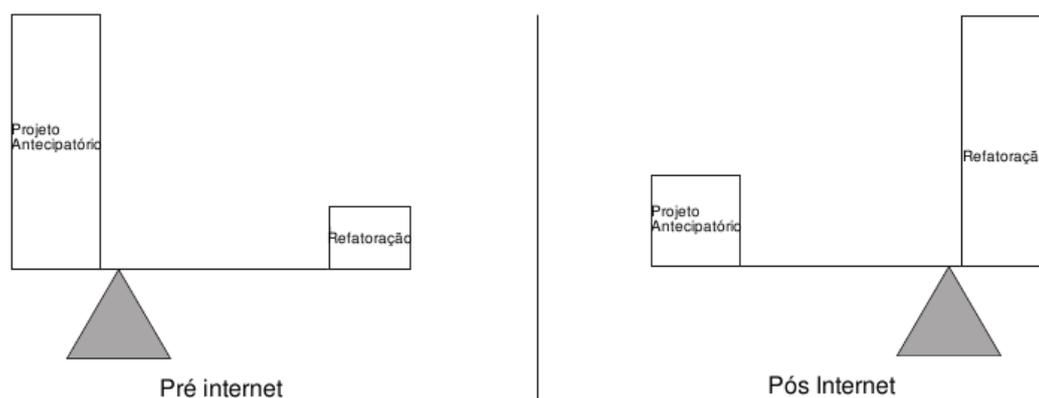


Figura 3.2: Ilustração sobre o equilíbrio entre projeto e refatoração

3.4 Caso CNXS-ELC-GiveZooks

3.4.1 Introdução

Para melhor ilustrar o ponto feito até agora, será descrito um caso no qual o autor deste trabalho está pessoalmente envolvido. O estudo remonta até o dia 15/12/2007, quando o processo começou, até o dia 27/03/2008, data arbitrariamente escolhida para o estudo.

Neste caso se buscará deixar claro como foi o processo como um todo, em um nível de granularidade mais elevado, para ser possível construir pontes entre o que foi discutido no trabalho até agora e um exemplo concreto de desenvolvimento de software profissional, para tentar tornar as idéias edificadas ao longo do trabalho mais acessíveis e mais tangíveis ao leitor.

3.4.2 Descrição

Foi primariamente criado uma presença virtual chamada CNXS - <http://www.cnxs.com.br>, onde o autor e 2 companheiros seus relatavam experiências e dicas relacionadas ao desenvolvimento de software e assuntos relacionados, especialmente na área de interesse particular, desenvolvimento web usando Ruby e o framework *Rails*. Os objetivos desta empreitada são, ordenadamente, os seguintes:

1. Obtenção de massa crítica de seus membros.
2. Aumento da reputação no ciberespaço dos membros individualmente e em equipe.
3. Renda o suficiente para a manutenção individual enquanto os objetivos acima estão em desenvolvimento.
4. Por final o desenvolvimento de um produto próprio, o qual terá mais chances de ser bem sucedido e financiado por causa dos fatores acima.

Para a realização dos três primeiros objetivos, a equipe precisava obter alguma maneira de adquirir experiência e conhecimento ao mesmo tempo que angariava contatos e provar-se ao mesmo tempo que conseguia lucro o suficiente para manter os membros de uma maneira aceitável aos mesmos. O caminho encontrado para isto foi a disponibilização dos serviços dos membros como empregados virtuais ao mercado de *Rails* no ciberespaço.

Na figura 3.3 mostra-se todo o esquema deste caso de uso, e nas subseqüentes legendas tenta-se levantar um paralelo entre a situação encontrada e as técnicas explicadas neste capítulo.

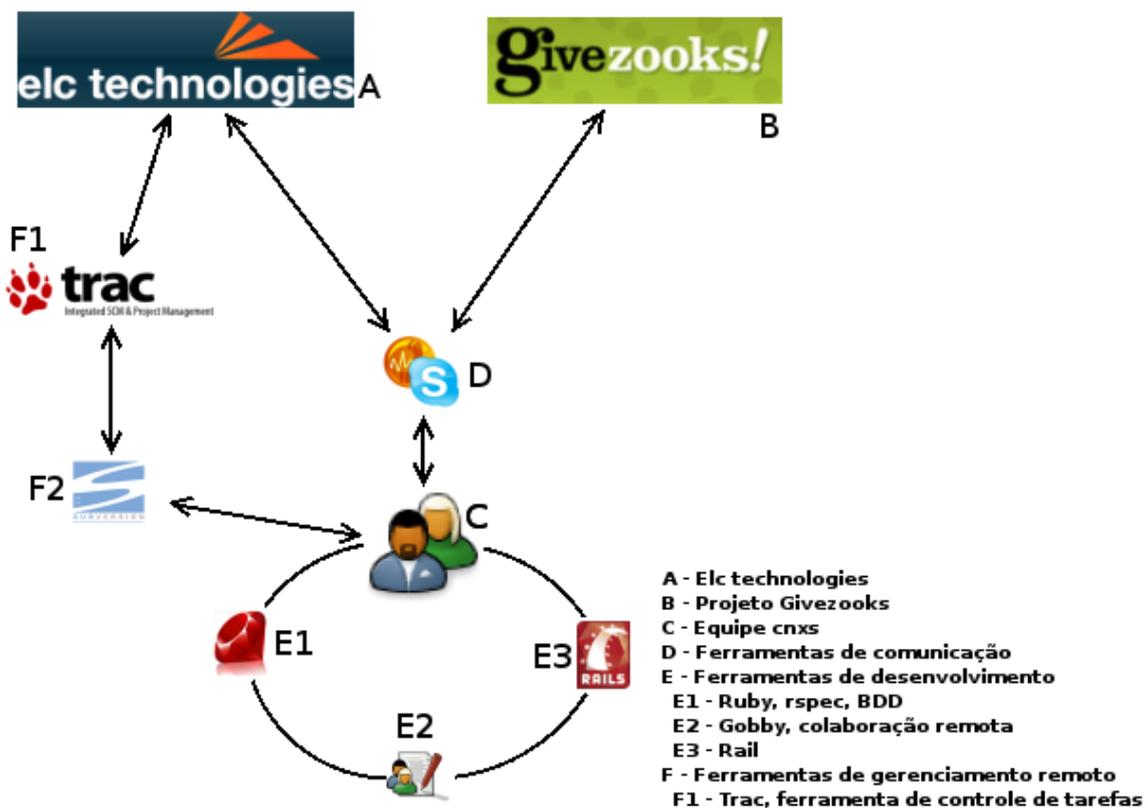


Figura 3.3: Ilustração do caso CNXS-ELC-Givezooks

- A: *ELC technologies* - <http://www.elctech.com/>, empresa baseada em Santa Barbara, Califórnia, especializada em desenvolvimento ágil web usando o framework *Rails*. Contratou parte da equipe de C como empregados via *telecommuting* através de um fórum de empregos virtuais no ciberespaço (<http://rubynow.com/>). Neste caso em particular houve a participação, além dos membros de C, de um gerente e arquiteto de projeto, dois outros programadores e diversos designers. Para a garantia da propriedade intelectual e segurança jurídica e formal houve uma assinatura de contrato padrão de de confidencialidade descrito no anexo A.1.
- B: *Givezooks* - <http://www.givezooks.com/>, outra empresa baseada na Califórnia, que contratou os serviços de A para o desenvolvimento de seu plano de negócios voltado a gerência de doações e organizações sem fins lucrativos, para tentar criar uma nova maneira de encarar a filantropia.
- C: Equipe CNXS: Uma proto empresa constituída por três membros que vendem serviços de desenvolvimento ágil web no ciberespaço. Dois de seus membros foram

contratados por A para ajudar na arquitetura e desenvolvimento do sistema de B.

- D: Estes 2 softwares, ekiga e skype, representam a utilização, bastante frequente no caso deste projeto, da comunicação através de voz e vídeo pela internet, criando um ambiente dinâmico para a troca de informação não somente entre a equipe e/ou gerente de projeto, mas também entre o cliente, para este poder retirar dúvidas e propor mudanças e relatar erros diretamente para os construtores do projeto. Neste caso há um achatamento entre as relações de poder, pois os programadores podem ter acesso direto a fonte mais confiável possível de informação, o cliente, e vice versa.
- E: O ciclo de desenvolvimento do software e colaboração entre desenvolvedores.

E1: A linguagem Ruby e o framework de BDD *rspec* são usados para garantir a qualidade durante o desenvolvimento ágil, como descrito nas técnicas de programação ágil no capítulo 2. Estes também são usados como uma forma de documentação, para suavizar a entrada, se necessário, de novos programadores, mantendo assim parte da inteligência do time dentro do projeto caso alguém tenha de ser substituído.

E2: Este ícone representa o software Gobby, que no projeto é usado como forma de integração e de colaboração dos programadores pelo ciberespaço. Com ele é possível dentro do projeto diversos programadores escrutinarem o mesmo arquivo em tempo real. Combinado com a3 pode-se ter uma experiência colaborativa de alto grau de informação.

E3: Este é o framework *Rails*. Mais que uma ferramenta para o desenvolvimento rápido e ágil de software web em si, é importante citá-lo como parte da filosofia de liberação e respeito aos gostos dos desenvolvedores. A fonte que podemos usar para fundamentar melhor essa afirmação vem do próprio livro *Getting Real*(37SIGNALS, 2006):

A razão número um para construir nossa aplicação usando *Ruby on Rails* é que é tão elegante, produtivo e lindamente construído. Ele tende a atrair o tipo de engenheiros que se importam com esses tipos de coisas... esse são exatamente o tipo de engenheiros que você quer em seu time

porque eles criam o tipo de software lindo, elegante e produtivo que você precisa para vencer no mercado.⁹

- F: A gerência da equipe, feita completamente no ciberespaço

F1: O software web de código aberto chamado TRAC, usado para o controle de tarefas e discussões sobre a mesmas pelo ciberespaço. Esse espaço é usado não somente para a identificação das tarefas, seus conteúdos e prazos, mas também como fórum para comentários e modificação do estado das mesmas. É possível também obter métricas para a análise de desempenho e progresso a partir dos dados deste software.

F2: Software de controle de versão, representado neste projeto pelo *subversion* (SVN). Ele é usado neste projeto em suas mais diversas finalidades, como o controle e administração de responsabilidades por determinados códigos, segurança da base de código, revisão de erros, etc. Enfim, é um imenso e contextualizado repositório no ciberespaço, onde o time pode trabalhar em diferentes, ou até a mesmas, partes do projeto sem grandes problemas.

3.4.3 Comparativos

Alguns pontos importantes a serem destacados neste caso, e que ajudam a visualizar como este processo se alinhou com as metodologias ágeis e com a filosofia *Getting Real*:

A comunicação entre A, B e C eram bastante freqüentes, apesar de poder ter havido uma maior ênfase entre B e C, ela foi o suficiente para ambas as partes construírem uma relação de confiança e de modelamento de pensamentos, onde se observava exatamente o que o cliente necessitava.

A quantidade de gráficos e documentos e meta documentos gerados foi ínfima, e apenas numa base de necessidade. Isso assegurou uma constante entrega do que era realmente valioso: software.

⁹*The number one reason I wanted to build our app using Ruby on Rails is that it is so elegant, productive, and beautifully designed. It tends to attract the kind of engineers who care about just those sort of things...those are exactly the kinds of engineers you want on your team because they create the kind of beautiful, elegant and productive software you need to win the market.*

A equipe e o processo valorizam bastante seu tamanho pequeno e sua capacidade de agilidade, pois apesar de uma pequena resistência por parte dos responsáveis da ELC ter sido encontrada ao apresentar idéias de refatoramento de código (que se tornaram necessárias), a esmagadora maioria delas foram rapidamente consideradas e implementadas.

Há semanalmente (e as vezes mais vezes), consultas e demonstrações diretamente para o cliente, que é quem realmente importa ser consultado. Nestas diversas iterações e consultas foi-se construindo organicamente um produto próximo da visão inicial do cliente, o que deixou seu nível de satisfação bastante elevado.

Observando estas características e comparando com a própria auto definição do texto do *Getting Real*(37SIGNALS, 2006) vê-se claras semelhanças:

Getting Real é eliminar tudo que não representa valores reais e construir a coisa de verdade.

Getting Real é menos. Menos massa, menos software, menos recursos, menos papelada, menos tudo que não for essencial.

Getting Real é continuar a ser ágil e pequeno.

...

Getting Real é sobre iterações e diminuir o custo da mudança. *Getting Real* é sobre mexer, apresentar e constantemente melhorar seu produto.

Getting Real entrega o que o cliente precisa e elimina tudo aquilo que ele não necessita.¹⁰

¹⁰*Getting Real is about skipping all the stuff that represents real (charts, graphs, boxes, arrows, schematics, wireframes, etc.) and actually building the real thing. Getting real is less. Less mass, less software, less features, less paperwork, less of everything that's not essential. Getting Real is staying small and being agile. ... Getting Real is about iterations and lowering the cost of change. Getting Real is all about launching, tweaking, and constantly improving. Getting Real delivers just what customers need and eliminates anything they don't.*

Capítulo 4

Conclusão

4.1 Revisão

Neste trabalho foi analisado uma filosofia de produção e administração de bens intangíveis (mais especificamente software) conhecida como *getting real*. Para tal esta filosofia foi dividida em componentes que se assemelham, ou são idênticos a, outras obras mais clássicas, como foi o caso da metodologia ágil e de código aberto.

Houve também uma tentativa de análise básica das confluências tecnológicas que estão acontecendo durante nosso tempo, como uma forma de justificativa para a proposição de métodos paralelos a filosofia *getting real* para adaptar uma organização aos tempos conturbados e dinâmicos que espera-se terem sido propriamente apresentados.

Quanto aos objetivos iniciais deste trabalho, haviam 3 especificamente:

- Analisar a filosofia *getting real*.
- Criar uma síntese do ambiente atual do mercado de tecnologia da informação como um todo.
- Descrever se há a necessidade de um modelo como o *getting real* para as empresas no ambiente descrito no objetivo acima, e em caso afirmativo, e como implantar-lo.

A análise do *getting real* foi realizada (e está distribuída no capítulo dois, mais especificamente no subitem 2.1) em um bom grau de profundidade, oferecendo

uma visão resumida da obra e também mostrando paralelos de outros autores sobre temas relacionados. Ilustrou-se esta análise com apresentações rápidas de casos reais do uso desta metodologia por uma empresa reconhecida no mercado internacional de produção de software.

Baseado em obras de impacto, tanto no mundo acadêmico quanto no mercado, buscou-se deixar claro o quanto as condições do mercado de tecnologias da informação, e todos os outros que são afetados por este, estão sofrendo mudanças dramáticas que mostram claros sinais de aceleração.

Quanto a necessidade desta metodologia, ou de outras similares, viu-se, a partir de citações de obras e dados recentes, bem como casos demonstrativos como o da própria Microsoft, que ela é real e urgente, pois como observou-se no objetivo anterior, as profundas mudanças do mercado tendem a acelerar-se ainda mais.

4.2 Dificuldades encontradas

As principais dificuldades encontradas foram:

- A falta de referências acadêmicas disponíveis sobre o assunto abordado, pois o interesse neste campo ainda é bastante recente, e o autor viu-se na necessidade de abordar formas não tradicionais de pesquisa, como *blogs* de autores. Além disto uma derivação da filosofia *Getting Real* em metodologias que podiam ser mais facilmente referenciadas foi adotada para basear o trabalho mais consistentemente.
- Analisar o caso de uso, pois a falta de exemplos documentados em uma granularidade aceitável levou o autor a escolher um caso em que estava pessoalmente envolvido, e o distanciamento necessário para uma síntese provou-se difícil de se atingir. O autor espera ter conseguido realizar uma observação imparcial o suficiente, porém tem a consciência que impressões pessoais são difíceis de remover completamente em tais casos.

4.3 Pensamentos finais

Como foi tentado deixar claro até aqui, o empreendedor tem pela frente um novo paradigma mundial, rico tanto em oportunidades quanto em desafios. De um lado existem inúmeras novas fronteiras e povos cujo acesso nunca foi tão fácil na história da humanidade. De outro o nível da competição e da pressão por resultados e inovação também nunca foi tão intenso devido ao achatamento e integração do globo, bem como as inúmeras revoluções simultâneas ocorrendo nas mais diversas áreas do pensamento humano.

A filosofia descrita ao longo deste trabalho aponta para caminhos que permitem aproveitar esse achatamento, além de maneiras mais eficientes de lidar com as dificuldades encontradas neste admirável novo mundo, cujos territórios ainda estão a serem demarcados. E talvez com determinação e coragem, aliadas ao conhecimento e a tecnologia apresentados até aqui, parte deste território pode ser conquistado por empreendedores que tenham a visão de tentar o novo. O autor espera que esta seja sua contribuição.

Referências Bibliográficas

YENICERI, G. *Getting Real or Not*. April 2007. Disponível em: <<http://analystdeveloper.com/blogs/gurkaneng/archive/2007/04/28/3487.aspx>>.

WEINBERG, G. M. *The Psychology of Computer Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1985. ISBN 0442292643.

BOYD, S. *Getting Really Real*. apr 2006. Disponível em: <<http://www.stoweboyd.com/message/2006/04/getting.html>>.

FOWLER, M. The new methodology. dec 2005. Disponível em: <<http://martinfowler.com/articles/newMethodology.html>>.

FOWLER, M. Is design dead? Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, p. 3–17, 2001. Disponível em: <<http://martinfowler.com/articles/designDead.html>>.

37SIGNALS. *Getting Real*. 1. ed. <http://gettingreal.37signals.com>: [s.n.], 2006. 177 p.

GRAHAM, P. *What Business can learn from Open Source*. august 2005. Disponível em: <<http://www.paulgraham.com/opensource.html>>.

RAYMOND, E. S. *The Cathedral and the Bazaar*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 1999. ISBN 1565927249.

SOUZA, C. R. B. D. *Global Software Development: Challenges and Perspectives*. Disponível em: <<http://citeseer.ist.psu.edu/457465.html>>.

HANSSON, D. H. *Reduce the risk, hire from open source*. set. 2007. Disponível em: <<http://www.loudthinking.com/arc/000505.html>>.

GAJENDRAN, R. S.; HARRISON, D. A. The good, the bad, and the unknown about telecommuting: Meta-analysis of psychological mediators and individual consequences. *Journal of Applied Psychology*, v. 92, n. 6, p. 1524 –1541, 2007. Disponível em: <www.apa.org/journals/releases/apl9261524.pdf>.

BITTNER, C. L. P. K. K. How to fail with the rational unified process: Seven steps to pain and suffering. 2001. Disponível em: <www.agilealliance.org/system/article/file/941/file.pdf>.

LAPLANTE, P. C. N. The demise of the waterfall model is imminent and other urban myths. *ACM Queue 1*, February 2004. www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=110.

FRIEDMAN, T. L. *The World is Flat*. [S.l.: s.n.], 2005.

GRANTHAM, C. E.; WARE, J. P.; WILLIAMSON, C. *Corporate Agility A Revolutionary New Model for Competing in a Flat World*. [S.l.]: AMACOM, 2007. ISBN 978-0-8144-0911-4.

LESSIG, L. *Code v2*. [S.l.]: Basic Books, 2006. ISBN 978-0-465-03914-2.

KURZWEIL, R. *The singularity is near*. [S.l.]: Phoenix, 2005. ISBN 0442292643.

GRAHAM, P. *Learning from Founders*. january 2007. <http://www.paulgraham.com/foundersatwork.html>. Disponível em: <www.paulgraham.com/foundersatwork.html>.

ANDERSON, C. *The long tail : how endless choice is creating unlimited demand*. Random House Publ: [s.n.], 2007. London. ISBN 978-1-8441-3851-7.

NEGROPONTE, N. From being digital to digital beings. *IBM Systems Journal*, v. 39, n. 3&4, p. 417–, 2000. Disponível em: <<http://dblp.uni-trier.de/db/journals/ibmsj/ibmsj39.html#Negroponte00>>.

HARDING, K. *Understanding Emerging Workforce Trends*. jul. 2000. Disponível em: <<http://www.di.net/articles/archive/2086/>>.

GRAHAM, P. *Microsoft is dead*. abr. 2007. Disponível em: <<http://www.paulgraham.com/microsoft.html>>.

LIVINGSTON, J. *Founders at Work: Stories of Startups' Early Days*. Berkely, CA, USA: Apress, 2007. ISBN 1590597141.

GRAHAM, P. *Ideas for a startup*. out. 2005. Disponível em: <<http://www.paulgraham.com/ideas.html>>.

HIGHSMITH, J. *Extreme programming. e-Business Application Delivery*, 2000. Disponível em: <<http://www.cutter.com/content-and-analysis/resource-centers/agile-project-management/sample-our-research/ead0002.html>>.

Apêndice A

Contrato

Este contrato foi assinado entre a empresa ELC e os membros da equipe do CNXS. Basicamente ele garante o sigilo das informações confidenciais que porventura estes terão acesso, bem como a certas restrições quanto a possíveis futuros padrões por um determinado tempo.

ELC Technologies 

Confidential

Some month 01, 3000
 XXXX XXXX
xxxxxxx@gmail.com

Re: Employment Offer

Dear xxxxxxxx,

We are pleased to make you this trial offer for full-time **Developer** for ELC Technologies (ELC). The terms of our offer are as follows:

1. Your employment will begin on Month 01, 3000. You will report to xxxx xxxxx. Your initial assignment will be to sent to you by email. Additional assignments and milestones will be established quarterly in cooperation with your supervisor. You will be expected to work on projects assigned to you by ELC. You will remain an employee for so long as ELC, in its sole discretion, chooses to employ you, and you wish to remain employed.
2. You will be paid **\$xxx/month** for full-time work for the period beginning on your employment start date as provided above. Your performance will be reviewed after 3 months.
3. You will be required to execute the Employee Intellectual Property Agreement attached to this letter. If you have any questions concerning this letter or the attached Agreements, we recommend you consult legal counsel of your choice.
4. The employment terms and policies outlined above are subject to change at the discretion of ELC. We would appreciate receiving your acceptance or rejection of this offer as soon as possible. We will hold this offer open until xxxxxx

xxxx, we hope that you will accept this offer and look forward to a long and productive relationship with you. Please indicate your acceptance of the terms of this offer by signing and dating in the spaces provided below and returning the original letter to us. The copy of this letter is for your records.

Very truly yours,

ELC Technologies
 XXXX XXXXX
 Director of Engineering

Terms of Offer Accepted:

By: _____ Date _____
 XXXX XXXXX

Figura A.1: Contrato de trabalho, página 1

ELC Technologies, LLC.

EMPLOYEE INTELLECTUAL PROPERTY AGREEMENT (Confidentiality, Invention Assignment, Nonraiding and Noncompetition)

I, the undersigned employee, agree as follows for the benefit of ELECTRIC CONNECTION, LLC, dba, ELC Technologies, its parents, subsidiaries, and affiliates (collectively "ELC").

1. Confidentiality. I agree that information that is not generally known to the public to which I have been or will be exposed as a result of my being employed by ELC is confidential information that belongs to ELC. This includes information developed by me, alone or with others, or entrusted to ELC by its customers or others. I will hold ELC's confidential information in strict confidence, and not disclose or use it except as authorized by ELC and for ELC's benefit. If anyone tries to compel me to disclose any of ELC's confidential information, by subpoena or otherwise, I will immediately notify ELC so that ELC may take any actions it deems necessary to protect its interests. My agreements to protect ELC's confidential information apply both while I am employed by ELC and after my employment by ELC ends, regardless of the reason it ends.

ELC's confidential information includes, without limitation, information relating to ELC's trade secrets, research and development, product development plans, inventions, knowhow, software (including source code and object code), procedures, manufacturing, engineering, purchasing, accounting, marketing, sales, customers, suppliers, financial status or employees.

I understand that this agreement does not limit my right to use my own general knowledge and experience, whether or not gained while employed by ELC, or my right to use information that is or becomes generally known to the public through no fault of my own, but I have the burden in any dispute of showing that information is not ELC's confidential information.

I understand it is ELC's policy not to improperly obtain or use confidential, proprietary or trade secret information that belongs to third parties, including others who have employed or engaged me or who have entrusted confidential information to me. I will not use for ELC's benefit or disclose to ELC confidential, proprietary or trade secret information that belongs to others, unless I advise ELC that the information belongs to a third party and both ELC and the owners of the information consent to the disclosure and use.

2. Inventions, Copyrights and Patents. ELC owns all Inventions and Works I make, conceive, develop, discover, reduce to practice or fix in a tangible medium of expression, alone or with others, either (a) during my employment by ELC (including past employment, and whether or not during working hours), or (b) within one year after my employment ends if the Invention or

Figura A.2: Contrato de trabalho, página 2

Work results from any work I performed for ELC or involves the use or assistance of ELC's facilities, materials, personnel or confidential information. ELC also owns all Inventions and Works of mine that I bring to ELC that are used in the course of ELC's business or that are incorporated into any Inventions or Works that belong to ELC.

I will promptly disclose to ELC, will hold in trust for ELC's sole benefit, will assign to ELC and hereby do assign to ELC all Inventions and Works described in the prior paragraph, including all copyrights (including renewal rights), patent rights and trade secret rights, vested and contingent. I will waive and hereby do waive any moral rights I have or may have in the Inventions and Works described in the prior paragraph. I agree that all Works I produce within the scope of my employment (which shall include all Works I produce related to ELC's business, whether or not done during regular working hours) shall be considered "works made for hire" so that ELC will be considered the author of the Works under the federal copyright laws. At ELC's direction and expense I will execute all documents and take all actions necessary or convenient for ELC to document, obtain, maintain or assign its rights to these Inventions and Works. ELC shall have full control over all applications for patents or other legal protection of these Inventions and Works.

"Inventions" means discoveries, developments, concepts, ideas, improvements to existing technology, processes, procedures, machines, products, compositions of matter, formulas, algorithms, computer programs and techniques, and all other matters ordinarily intended by the word "invention," whether or not patentable or copyrightable. "Inventions" also includes all records and expressions of those matters. "Works" means original works of authorship, including interim work product, modifications and derivative works, and all similar matters, whether or not copyrightable.

I understand that this agreement does not apply to any Invention or Work of mine for which no equipment, supplies, facilities or trade secret information of ELC was used and which was developed entirely on my own time, unless (a) the Invention or Work relates (i) directly to ELC's business or (ii) to ELC's actual or demonstrably anticipated research or development, or (b) the Invention or Work results from any work I performed for ELC.

Figura A.3: Contrato de trabalho, página 3

3. ELC Materials. I will safeguard and return to ELC when my employment ends, or sooner if ELC requests, all documents and property in my care, custody or control relating to my employment or ELC's business, including without limitation any documents that contain ELC's confidential information.

4. Nonraiding of Employees. So long as I am employed by ELC and for twenty-four (24) months after my employment ends, regardless of the reason it ends, I will not directly or indirectly solicit any employee to leave his or her employment with ELC. This includes that I will not (a) disclose to any third party the names, backgrounds or qualifications of any ELC employees or otherwise identify them as potential candidates for employment; (b) personally or through any other person approach, recruit or otherwise solicit employees of ELC to work for any other employer; or (c) participate in any pre-employment interviews with any person who was employed by ELC while I was employed by ELC.

5. No Disparagement or Interference. I will not disparage ELC or its business or products and will not interfere with ELC's relationships with its customers, employees, vendors, bankers or others. This applies both while I am employed by ELC and after my employment by ELC ends, regardless of the reason it ends.

6. Other Employment While Employed By ELC. While I am employed by ELC I will not do work that competes with or relates to any of ELC's products or activities without first obtaining ELC's written permission. Any business opportunities related to ELC's business that I learn of or obtain while employed by ELC (whether or not during working hours) belong to ELC, and I will pursue them only for ELC's benefit.

7. Noncompetition After Employment by ELC Ends. For twenty-four (24) months after my employment by ELC ends, regardless of the reason it ends, I will not, directly or indirectly: (a) sell, market or propose to sell or market products that compete or will compete with ELC's then existing or reasonably anticipated products ("Competing Products") in any geographic area where ELC's products are then marketed, (b) design or develop Competing Products, or (c) work for or with, or provide services or information to, any person or entity that (i) sells, markets or proposes to sell or market Competing Products in any geographic area where ELC's products are then marketed, (ii) is designing or developing Competing Products, or (iii) is shown on the attached list of Competing Companies.

The persons and entities (other than myself) that are covered by this noncompetition provision are referred to as the "Competing Companies." ELC maintains a list (the "List") of examples of Competing Companies. A copy of the current List is attached as Exhibit A. The List contains examples only, and is not necessarily a complete list of all Competing Companies.

Where a Competing Company has multiple divisions, this noncompetition provision shall apply only to those divisions that are involved with Competing Products, provided that I and the

competitor provide written assurances satisfactory to ELC that the information and work product I provide to other divisions of the competitor will not be shared, directly or indirectly, or intentionally or unintentionally, with the division involved with Competing Products.

I understand that in cases where this noncompetition provision does not apply, I am still subject to all other obligations I have to ELC, including my obligations related to ELC's inventions, copyrights and confidential information.

8. Disclosure of Other Work. Before I undertake any work for myself or anyone else during my employment by ELC or within twenty-four (24) months after my employment ends that will involve subject matter related to ELC's activities, I will fully disclose the proposed work to ELC. Any current disclosures shall be made in Exhibit B.

9. Reasonableness of Terms. I acknowledge that the terms of this agreement are reasonably necessary to protect ELC's legitimate business interests. I acknowledge that if my employment with ELC ends my experience and capabilities are such that I can obtain employment that does not violate this agreement, and that an injunction to enforce this agreement will not prevent me from earning a reasonable livelihood.

10. Future Consulting or Employment for ELC. If my employment relationship with ELC ends but ELC employs me again or engages me as a consultant, then this agreement shall apply to my later employment(s) or engagement(s) unless they follow a period of a year or more during which I was neither employed nor engaged by ELC. If this agreement becomes applicable to a consulting relationship, the references in this agreement to my employment by ELC shall be treated, as appropriate, as referring to my consulting relationship with ELC.

11. No Guarantee of Employment. I understand this agreement is not a guarantee of continued employment. My employment is terminable at any time by ELC or me, with or without cause or prior notice, unless otherwise provided in a written employment agreement.

12. No Conflicting Agreements. I am not a party to any agreements, such as confidentiality or noncompetition agreements, that limit my ability to perform my duties for ELC.

Figura A.5: Contrato de trabalho, página 5

13. Miscellaneous. If I breach this agreement it will cause ELC irreparable harm. If I breach or threaten to breach this agreement, ELC will be entitled to injunctive or other equitable relief as well as money damages. If I breach this agreement, I will hold in trust for ELC all income I receive as a result of the violation. I consent to ELC notifying anyone to whom I may provide services of the existence and terms of this agreement. In any lawsuit arising out of or relating to this agreement or my employment, including without limitation arising from any alleged tort or statutory violation, the prevailing party shall recover their reasonable costs and attorneys fees, including on appeal. This agreement shall be governed by the internal laws of the state of California without giving effect to provisions thereof related to choice of laws or conflict of laws. Venue and jurisdiction of any lawsuit involving this agreement or my employment shall exist exclusively in state and federal courts in Santa Barbara, CA, unless injunctive relief is sought by ELC and, in ELC's judgment, may not be effective unless obtained in some other venue. If any part of this agreement is held to be unenforceable, it shall not affect any other part. If any part of this agreement is held to be unenforceable as written, it shall be enforced to the maximum extent allowed by applicable law. My obligations under this agreement supplement and do not limit other obligations I have to ELC, including without limitation under the law of trade secrets. This agreement shall be enforceable regardless of any claim I may have against ELC. This agreement shall survive the termination of my employment, however caused. The waiver of any breach of this agreement or failure to enforce any provision of this agreement shall not waive any later breach. This agreement is binding on me, my heirs, executors, personal representatives, successors and assigns, and benefits ELC and its successors and assigns. This agreement is the final and complete expression of my agreement on these subjects, and may be amended only in a writing signed by ELC and me.

DATED this xxxxxx

EMPLOYEE:

xxxxxx

ELC Technologies.

ACCEPTANCE:

By:xxxxxx

Figura A.6: Contrato de trabalho, página 6

List of Competing Companies (Exhibit A)

Terralien
Enomaly
ElasticLive
Yub.com
Shopping.com

Figura A.7: Contrato de trabalho, página 7

Employee Inventions Disclosure (Exhibit B)

I have stated all inventions for disclosure above or selected below to indicate that I have no inventions to disclose.

I have no inventions to disclose.

DATED this _____

EMPLOYEE:

XXXXX XXXXX

Figura A.8: Contrato de trabalho, página 8