

Uma Abordagem de Apoio à Avaliação e Melhoria da Produtividade de Desenvolvedores de Software

Edgy Paiva¹, Adriano Albuquerque¹

¹Fundação Edson Queiroz - Universidade de Fortaleza
Mestrado em Informática Aplicada - Bloco J, Sala 30
Av. Washington Soares, 1321 - Bairro Edson Queiroz - CEP 60811-905
Fortaleza – Ceará – Brasil

edgy.paiva@ivia.com.br, adrianoba@unifor.br

Abstract. *From a competitive perspective, software development companies need to improve the productivity of their development teams and, to do so, it is very important to know which factors influence the team's productivity. However, it is observed that companies often do not know what these factors are and, moreover, they cannot measure and mitigate their influence on the development productivity. This work aims to identify the main factors that influence the productivity of software developers, and for each of these factors, propose "best practices" to mitigate their influence on the developer's productivity. Furthermore, it was defined an approach and a process to support the evaluation and improvement of the productivity of the software developers. To evaluate the proposed approach, a supporting tool was developed and a case study was implemented in a software development company.*

Resumo. *Por uma questão de competitividade, as empresas precisam, cada vez mais, melhorar a produtividade da sua equipe de desenvolvedores e, para isso, é muito importante conhecer os fatores que influenciam essa produtividade. No entanto, observa-se que normalmente as empresas não sabem que fatores são esses e, além disso, não conseguem medir e mitigar a influência dos mesmos. Este trabalho tem por objetivo identificar os principais fatores que influenciam na produtividade dos desenvolvedores de software e, para cada um desses fatores, propor "melhores práticas" para mitigar a influência dos mesmos na produtividade dos desenvolvedores. Além disso, foi definida uma abordagem e um processo para apoiar a avaliação e melhoria da produtividade de desenvolvedores de software. Para avaliar a abordagem proposta, foi desenvolvida uma ferramenta de apoio e foi realizado um estudo de caso em uma empresa de desenvolvimento de software.*

1. Introdução

Alguns dos principais problemas enfrentados pela indústria de engenharia de software estão relacionados à produtividade do desenvolvedor (programador).

Segundo Peck e Callahan (2002), ao longo dos últimos anos, muitos estudos têm demonstrado uma grande disparidade na produtividade de programadores que têm um nível de experiência similar. Se uma empresa é capaz de identificar os programadores mais produtivos e eliminar, ou ainda, treinar os menos produtivos, o aumento da produtividade resultante deverá ser uma vantagem competitiva.

Com o desenvolvimento do software sendo visto mais como uma arte do que uma ciência tem sido difícil estudar fatores que afetam a produtividade de projetos, bem como prever com exatidão os custos relacionados. A falta de conhecimento a respeito da produtividade de desenvolvimento de software e de métodos de medição dessa produtividade tem causado confusão na estimativa dos custos do software. A fim de melhorar a qualidade de um produto de software, bem como aumentar a produtividade do processo de desenvolvimento, medições precisas dos custos de desenvolvimento e dos fatores de produtividade são necessárias (YU, 1990).

Na indústria de software, produtividade e qualidade são fundamentais e devem ser consideradas em conjunto, pois o fato de um profissional conseguir executar sua tarefa com velocidade, mas sem qualidade, não significa que ele é muito produtivo. O mais importante, para a questão de produtividade, é executar a tarefa com qualidade e no menor tempo possível.

Segundo Yu, Smith e Huang (1990), basicamente há três grandes etapas no processo de melhoria da produtividade: medição, análise e melhoria da produtividade do projeto de software.

A principal motivação deste trabalho é desenvolver uma abordagem de apoio à avaliação e melhoria da produtividade de desenvolvedores de software de uma empresa.

Outras questões também deverão ser respondidas e servirão de apoio à motivação principal deste trabalho:

- Que fatores influenciam na produtividade de uma equipe de desenvolvedores (programadores) de software?
- Quais os fatores que mais influenciam a produtividade dos desenvolvedores de um determinado projeto?
- Quais são as “melhores práticas” para minimizar a influência negativa dos fatores que mais influenciam na produtividade dos desenvolvedores?
- A implementação dessas “melhores práticas” pode aumentar a produtividade do desenvolvedor?

Normalmente, num ciclo de vida de desenvolvimento de software, o maior esforço acontece durante a fase de programação (desenvolvimento), e as respostas das perguntas acima, utilizando técnicas e/ou ferramentas, trarão um grande benefício às empresas de desenvolvimento de software.

O artigo está organizado da seguinte forma. A segunda seção consiste em realizar uma pesquisa bibliográfica de estudos sobre produtividade na área de desenvolvimento de software. A terceira seção consiste em obter uma lista de fatores que influenciam na produtividade do desenvolvedor de software e realizar uma pesquisa para obter o ranking desses fatores. A quarta seção envolve a definição de um processo para definição e acompanhamento de fatores que influenciam a produtividade de desenvolvedores num projeto e apresentação de uma ferramenta de apoio para automatizar o processo. A quinta seção consiste na apresentação de um estudo de caso realizado em uma empresa de desenvolvimento de software, incluindo uma pesquisa qualitativa para identificar “melhores práticas”. A última seção consiste em explicitar as conclusões deste trabalho e indicar trabalhos futuros.

2. Estudos sobre Produtividade na Indústria de Software

Segundo Bernolak (1997), produtividade significa quanto e quão bem nós produzimos com os recursos utilizados. Se produzirmos mais ou melhores bens com os mesmos recursos, aumentamos a produtividade. Se produzirmos os mesmos bens com menos recursos, também aumentamos a produtividade. Recursos significam todos os recursos humanos e físicos envolvidos, isto é, as pessoas que produzem os produtos ou fornecem os serviços e os ativos com os quais as pessoas podem produzir os produtos ou fornecer os serviços.

No ambiente de desenvolvimento de software, se conseguirmos desenvolver o mesmo software com menos recursos, principalmente recursos humanos, temos um ganho de produtividade e competitividade. Grossman (1993) indica que as empresas precisam perceber que os ganhos de produtividade são uma das principais armas para atingir vantagens de custo e qualidade sobre seus concorrentes.

Melhorar a produtividade no desenvolvimento de software é um grande desafio para as companhias. Segundo Drucker (1991), o maior desafio enfrentado pelos gerentes, nos países desenvolvidos do mundo, é aumentar a produtividade dos trabalhadores do conhecimento e de serviços.

A relação entre produtividade e qualidade é fundamental, pois é extremamente importante que o produto de software desenvolvido tenha um bom nível de qualidade. Dessa forma, ao medir a produtividade, temos que avaliar também se o que foi produzido está com o nível de qualidade exigido.

Segundo Boehm (2006 apud Nguyen et al, 2011), ao longo de décadas, processos de software se transformaram de *code-and-fix* e *waterfall-like* para espiral, iterativo e abordagens ágeis de desenvolvimento de software. Da mesma forma, ferramentas de desenvolvimento mais avançadas, métodos e tecnologias surgiram. Tais avanços progressivos mudaram as práticas de desenvolvimento de software, levando a custos mais baixos de desenvolvimento de software, redução do tempo de desenvolvimento e aumento da produtividade e qualidade. Caracterizar os fatores que influenciam a produtividade de software e como esses fatores mudaram devido à evolução das práticas de software é importante para pesquisadores e profissionais na definição de “melhores práticas” no desenvolvimento de software.

Nguyen et al. (2011) destacaram que existem muitas dificuldades envolvidas na caracterização e compreensão da produtividade de software. Uma razão, segundo Scacchi (1995 apud Nguyen et al., 2011), é que um grande número de fatores afeta a produtividade dos projetos de desenvolvimento de software. Outra razão, de acordo com Trendowicz e Munch (2009 apud Nguyen et al., 2011), é que o impacto relativo de cada fator difere de um ambiente para outro. Além disso, os fatores e seus impactos têm evoluído ao longo dos anos, refletindo mudanças nos processos, ferramentas de desenvolvimento e tecnologias. Por exemplo, as capacidades de armazenamento e processamento que já foram os principais obstáculos para sistemas de 40 anos atrás, hoje não são mais.

No trabalho de Nguyen et al. (2011), é colocado que alguns autores, Boehm (1981), Jones (1998), Kauffman e Kumar (1993) enfatizaram que muitos estudos têm sido focados na identificação de fatores e seu impacto sobre a produtividade e, conseqüentemente, nos custos de software. Mas que, embora estudos anteriores tenham

contribuído significativamente para o entendimento da produtividade de software, há uma falta de estudos experimentais sobre a forma como o impacto desses fatores evolui com o tempo para refletir as mudanças nas práticas de desenvolvimento de software. Essa deficiência resulta em baixa precisão na estimativa de futuros projetos e gera estratégias insatisfatórias para melhorar a medição e gerenciamento da produtividade de software.

3. Pesquisa sobre os Fatores que Influenciam a Produtividade do Desenvolvedor de Software

Medir e melhorar a produtividade dos desenvolvedores de software é um dos grandes desafios enfrentados pelas empresas desenvolvedoras de software. Sendo assim, objetivando auxiliar essas empresas a identificar possíveis causas que estejam interferindo na produtividade das suas equipes, apresenta-se na Tabela 1 uma lista de 32 (trinta e dois) fatores, retirados da bibliografia existente a partir da utilização de práticas de revisão sistemática (KITCHENHAM, 2004), que influenciam a produtividade dos desenvolvedores (PAIVAa et al, 2010; PAIVAb et al, 2010).

Tabela 1. Lista de Fatores que Influenciam a Produtividade do Desenvolvedor

Fator	Descrição
Acesso à Internet	Possuir acesso livre à Internet, inclusive chats, youtube, redes sociais, etc.
Ambiente de Trabalho	O ambiente de trabalho ser agradável, com bom espaço, sem barulho e com conforto.
Arquitetura	Possuir uma arquitetura de software clara e adequada.
Atraso Tecnológico	Estar utilizando ferramentas / versões de software defasados. Exemplo: Fortran.
Benefícios	A empresa oferecer benefícios adicionais, como plano de saúde, vale alimentação e participação nos lucros.
Comprometimento	Ser dedicado e comprometido com o seu trabalho.
Comunicação	Existir uma comunicação adequada no projeto, reuniões frequentes de status, entre outros.
Documentação	Existir documentação atualizada em todas as fases do ciclo de vida de desenvolvimento de um software.
Domínio de Aplicação	Possuir conhecimento do domínio de aplicação do que está sendo desenvolvido. Exemplo: Sistemas Médicos.
Escritório em Casa	Trabalhar no formato de "home-office" (em casa).
Estação de Trabalho	Possuir uma estação de trabalho adequada (infraestrutura física) para executar suas atividades.
Experiência	Ter conhecimento e experiência nos processos, metodologia, tecnologia e ferramentas utilizadas no seu trabalho.
Ferramenta Desenvolvimento	Utilizar-se de uma ferramenta adequada de desenvolvimento. Exemplo: Visual Studio, Eclipse, etc.
Gerência de Projeto	Possuir uma gerência de projeto adequada.

Fator	Descrição
Gestão do Conhecimento	Possuir um ambiente e política de gestão do conhecimento.
Linguagem de Programação	A linguagem de programação utilizada ser de última geração.
Localização Física	Ter fácil acesso ao local de trabalho (sem trânsito, com disponibilidade de meios de transporte).
Metodologia	A empresa não possuir metodologia formal de desenvolvimento de software.
Metodologia tipo Ágil	Utilizar uma metodologia com uso de processos e/ou práticas ágeis (XP, SCRUM).
Modernidade	Utilizar sempre as novas tecnologias lançadas pelo mercado.
Motivação	Estar motivado com o trabalho.
Nível de Maturidade	A empresa possuir um nível de maturidade de processo (Ex. MPSBR, ISO ou CMMI).
Relacionamento Interpessoal	Manter um bom relacionamento com os colegas de trabalho e clientes.
Remuneração	Ter um salário de acordo com o mercado.
Requisitos em conformidade	Os requisitos fornecidos pelo analista estarem claros, completos e estáveis.
Reuso de código	Existir na empresa uma biblioteca de código para reuso e uma política de reuso.
Tamanho da Equipe	O projeto possuir um grande número de pessoas na equipe (mais de 20).
Tamanho do Projeto	O tamanho do projeto (esforço) ser bem elevado.
Teste	O projeto possuir uma política de teste (o teste é executado antes de o código ir para o cliente).
Tipo de Projeto	Ser um projeto de desenvolvimento ou manutenção.
Treinamento	Ter sido realizado treinamento adequado para execução da atividade.
Uso de Prototipação	Ter sido utilizada prototipação no projeto.

Para obter o *ranking* desses fatores, foi aplicado um questionário com gerentes de projeto e desenvolvedores. O tamanho da amostragem foi de 20 (vinte) gerentes de projeto e 77 (setenta e sete) desenvolvedores, totalizando 97 (noventa e sete) profissionais.

O *ranking* dos fatores com alta influência na produtividade está descrito na Tabela 2 e será apresentada a seguir uma análise dos 5 (cinco) primeiros fatores.

Tabela 2. Ranking dos Fatores com Alta Influência na Produtividade

POSICÃO	FATOR	QTDE	%
1	Comprometimento	86	88,7%
2	Motivação	86	88,7%
3	Requisitos em conformidade	83	85,6%
4	Ambiente de Trabalho	82	84,5%
5	Estação de Trabalho	79	81,4%
6	Ferramenta Desenvolvimento	78	80,4%
7	Remuneração	77	79,4%
8	Relacionamento Interpessoal	73	75,3%
9	Experiência	72	74,2%
10	Gerência de Projeto	72	74,2%
11	Teste	71	73,2%
12	Arquitetura	68	70,1%
13	Treinamento	68	70,1%
14	Comunicação	60	61,9%
15	Benefícios	51	52,6%
16	Domínio de Aplicação	49	50,5%
17	Reuso de código	48	49,5%
18	Documentação	46	47,4%
19	Nível de Maturidade	40	41,2%
20	Metodologia tipo Ágil	31	32,0%
21	Linguagem de Programação	30	30,9%
22	Gestão do Conhecimento	29	29,9%
23	Uso de Prototipação	29	29,9%
24	Acesso à Internet	28	28,9%
25	Localização Física	28	28,9%
26	Modernidade	19	19,6%
27	Escritório em Casa	10	10,3%
28	Tamanho do Projeto	10	10,3%
29	Tipo de Projeto	9	9,3%
30	Atraso Tecnológico	4	4,1%
31	Tamanho da Equipe	4	4,1%
32	Metodologia	3	3,1%

Os fatores que têm mais alta influência positiva na produtividade são: “Comprometimento” (1º) e “Motivação” (2º). Entende-se que esse resultado está relacionado com o fato de esses fatores serem a base para qualquer profissional desempenhar produtivamente as suas atividades.

Para os desenvolvedores, o fato dos requisitos estarem em conformidade (3º) aumenta a produtividade, pois o não entendimento do requisito deixa dúvidas e, conseqüentemente, reduz a produtividade.

O “ambiente de trabalho” (4º) e uma boa “estação de trabalho” (5º) também foram tidos como importantes. Ações simples, como substituir o equipamento do desenvolvedor ou colocar mais memória e, quem sabe, até uma mudança para uma cadeira mais confortável pode influenciar positivamente na produtividade.

Para que uma empresa possa se tornar mais competitiva, é necessário medir, analisar e melhorar os principais fatores que influenciam na produtividade do desenvolvedor. Por exemplo, em uma empresa onde é identificado que a estação de trabalho está influenciando bastante na produtividade do desenvolvedor, o investimento em novas estações (custo baixo em relação à remuneração do desenvolvedor), poderia trazer um grande ganho de produtividade para a empresa.

4. Processo para Definição e Acompanhamento de Fatores que Influenciam a Produtividade de Desenvolvedores em um Projeto de Software

Como, na prática, os desenvolvedores trabalham efetivamente em projetos de software, percebe-se a necessidade de se ter um *ranking* de fatores por projeto, pois cada projeto tem suas particularidades, e um fator pode ter mais relevância do que outros em um determinado projeto. Além disso, a partir desse *ranking*, é importante implementar o uso de “melhores práticas” para mitigar a influência dos fatores na produtividade, bem como avaliar a eficácia dessa implementação.

Dessa forma, foi definido um processo (Figura 1) com as seguintes macro-atividades:

- Definir o Projeto / Equipe de Trabalho.
- Realizar a pesquisa com a equipe do projeto que servirá de subsídio para elaboração do ranking de influência dos fatores para o projeto.
- Definir as “melhores práticas” a serem utilizadas no projeto com base nos fatores que mais influenciam a produtividade.
- Avaliar a eficácia da implementação da “melhor prática” naquele fator.

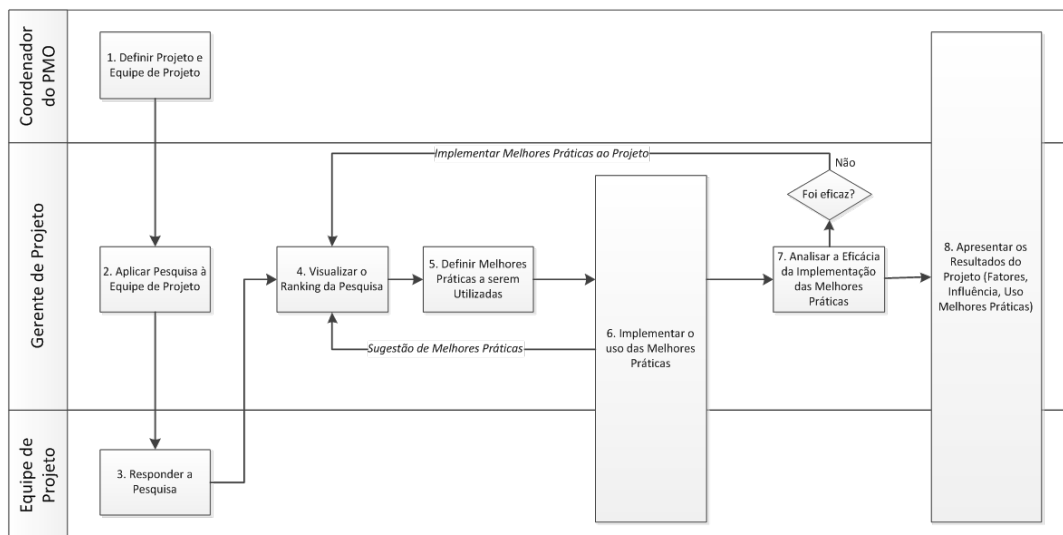


Figura 1 - Processo para definição e acompanhamento de fatores que influenciam a produtividade de desenvolvedores num projeto

Foi desenvolvida uma ferramenta chamada de MPDS (Melhoria da Produtividade do Desenvolvedor de Software) que tem por objetivo automatizar o processo descrito na Figura 1, bem como permitir gerar uma base de conhecimento sobre fatores que influenciam na produtividade dos desenvolvedores e “melhores práticas” para mitigar esses fatores em múltiplas organizações.

Para exemplificar o uso da ferramenta, é apresentado na Figura 2, o *ranking* que é gerado automaticamente após todos os colaboradores do projeto terem respondido a pesquisa diretamente na ferramenta.

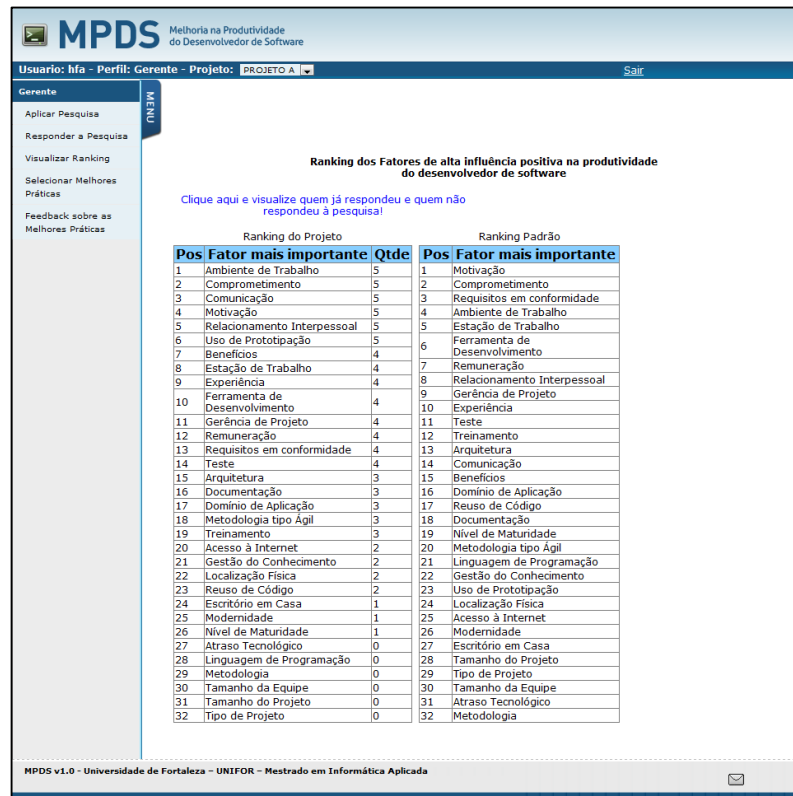


Figura 2 – Exemplo de uso da ferramenta MPDS (*Ranking* de Fatores)

5. Estudo de Caso em uma Empresa de Desenvolvimento de Software

Para este estudo de caso, utilizou-se a metodologia proposta por Yin (2003). De acordo com o autor, estudo de caso é um método preferido quando:

- Questões de “como” e “por que” são necessárias.
- O investigador tem pouco controle sobre os eventos
- O foco é sobre um fenômeno contemporâneo dentro de um contexto da vida real

Neste estudo, percebeu-se que a metodologia de Yin (2003) é mais adequada, pois atende a todos os pontos citados anteriormente. O processo do estudo de caso definido por Yin (2003) é iterativo e as principais fases do processo são:

- Planejar – Fazer todo o planejamento do estudo de caso
- Projetar – Preparar o projeto do estudo de caso (definir as hipóteses, dados, coleta e análise)
- Preparar – Preparar tudo que for necessário para execução do estudo de caso
- Coletar – Fazer a coleta dos dados do estudo de caso
- Analisar – Analisar os dados coletados
- Compartilhar – Divulgar o resultado da análise

As principais hipóteses para este trabalho foram:

- H1 – A identificação dos principais fatores que influenciam na produtividade do desenvolvedor em um projeto possui uma alta relevância para que se possa melhorar a produtividade do desenvolvedor.

- H2 – O uso de “melhores práticas” para mitigar os fatores que têm mais influência na produtividade do desenvolvedor no projeto pode aumentar a produtividade do mesmo.
- H3 – O uso da abordagem proposta promove uma maior satisfação da equipe de desenvolvedores e, conseqüentemente, uma maior produtividade dos mesmos.
- H4 – Após a execução do processo, haverá um aumento real de produtividade da equipe, levando em consideração a produtividade anterior média da empresa ou do projeto.

Durante a fase de planejamento do estudo de caso, foi percebida a necessidade de se realizar uma pesquisa qualitativa para identificar as “melhores práticas” para mitigar a influência dos fatores na produtividade do desenvolvedor. Estas “melhores práticas” foram cadastradas na ferramenta MPDS e foram utilizadas pelos gerentes de projeto durante o estudo de caso.

5.1. Pesquisa Qualitativa para Identificação das “Melhores Práticas”

A pesquisa qualitativa foi realizada com profissionais experientes e que tinham o perfil de gerente de projetos, de forma a se obter, para cada fator que influencia na produtividade do desenvolvedor, uma ou mais “melhores práticas” capazes de mitigar essa influência.

Os profissionais que participaram da pesquisa foram divididos em 2 (dois) grupos, e para cada grupo foi realizada uma reunião de, aproximadamente, 90 (noventa) minutos, sendo executadas as seguintes atividades:

- Para cada fator, o gerente de projeto contribuiu, na sua visão, com “melhores práticas” para mitigar a influência do fator na produtividade.
- Para cada 5 (cinco) fatores, foi realizada uma discussão entre o grupo, para que fosse obtido o “consenso” em relação à melhor prática.

Essas atividades foram realizadas nos 2 (dois) grupos e, ao final, foram consolidadas todas as “melhores práticas”. Como esta pesquisa obteve, na média, mais de 2 (duas) melhores práticas para cada fator, apresentar-se-á na Tabela 3 somente as “melhores práticas” de alguns fatores (ao todo foram mais de 50 “melhores práticas”).

Tabela 3. Exemplo de “Melhores Práticas” para mitigar os fatores que influenciam na produtividade do desenvolvedor

Fator	Melhores Práticas	Descrição
Acesso à Internet	Definir horário para acesso livre	Definir, dentro da política de acesso à Internet na empresa, horários em que o uso da Internet é permitido como, por exemplo, no horário do almoço ou após o horário do expediente.
	Definir uma política de acesso	Definir uma política de acesso, incluindo todas as regras para que o uso indiscriminado da Internet não interfira na produtividade do profissional.
	Liberar acesso com monitoramento	O acesso à Internet deve ser liberado para todos os colaboradores, porém deve ser deixado claro que não devem fazer uso indevido da Internet, bem como que todo acesso será monitorado para evitar desvios de conduta.

Fator	Melhores Práticas	Descrição
Requisitos em Conformidade	Realizar revisão dos artefatos por outro analista (pares)	Utilizar a técnica de revisão em pares durante a fase de especificação de requisitos
	Utilizar prototipação	Utilizar prototipação para auxiliar a fase de elicitação de requisitos, auxiliando a validação do usuário e o entendimento do desenvolvedor.
	Validar de forma clara com o cliente	Definir um processo de validação de requisitos com o cliente, de forma a não serem necessárias mudanças frequentes nos requisitos durante a fase de desenvolvimento.
Tamanho do Projeto	Criar indicadores de medição e acompanhamento	Definir indicadores de acompanhamento do projeto e acompanhá-los com mais frequência.
	Fazer iterações com entregas parciais	Dividir o projeto em iterações menores de forma a termos "n" projetos pequenos.

5.2. O Estudo de Caso

O estudo de caso foi realizado numa empresa de desenvolvimento de software do Estado do Ceará. Fundada em 1996, a empresa é certificada e auditada nas melhores práticas: CMMI nível 2, MPS.BR nível E e ISO 9001:2008, possui escritórios na região Nordeste do Brasil e atuação em todo território brasileiro, nos Estados Unidos, Reino Unido e Portugal.

Foram selecionados, para o estudo de caso, 3 (três) projetos de desenvolvimento em clientes diferentes da empresa, conforme descrito na Tabela 4. A ferramenta MPDS foi utilizada e automatizou todo o processo de geração do *ranking* dos fatores nos projetos, bem como a seleção das “melhores práticas” e análise da eficácia do uso das mesmas.

Tabela 4. - Caracterização dos projetos do estudo de caso

CARACTERÍSTICAS	PROJETO		
	A	B	C
Tipo de Projeto: Desenvolvimento	✓	✓	✓
Tipo de Projeto: Manutenção			✓
Tecnologia: Web	✓	✓	✓
Tecnologia: Desktop			✓
Metodologia: CMMI/MPSBR		✓	✓
Metodologia: Ágil	✓		
Linguagem: Java			✓
Linguagem: .NET C#	✓	✓	
Linguagem: Outras Linguagens			✓
Tamanho do Projeto	Grande (Acima de 12 meses)	Médio (Acima de 3 meses até 12 meses)	Pequeno (Até 3 meses)
Integração com Sistemas Existentes	✓	✓	✓
Uso de Tecnologia Inovadora	✓		
Nível de Experiência do Gerente de Projeto	Alto (Acima de 5 anos)	Alto (Acima de 5 anos)	Alto (Acima de 5 anos)
Nível de Experiência Médio dos Desenvolvedores	Alto (Acima de 5 anos)	Médio (Acima de 1 ano até 5 anos)	Médio (Acima de 1 ano até 5 anos)

Todos os 3 (três) projetos (A, B e C) tiveram melhoria na produtividade e será apresentado a seguir o detalhamento do estudo de caso dos projetos A e B.

5.2.1. Estudo de Caso no Projeto A

O Projeto A, por estar utilizando um método ágil, não possuía uma métrica bem definida de medição de produtividade, como por exemplo, ponto de função. O trabalho era realizado em *sprints* mensais, onde a equipe avaliava, a partir de um conjunto de atividades (*backlog* do produto), que atividades poderiam ser concluídas no *sprint*.

Para que se pudesse chegar a uma métrica, foram identificadas, em todos os *sprints* do projeto, quantas funcionalidades por complexidade (baixa, média e alta) haviam sido desenvolvidas pela equipe. Definiu-se, então, um peso para cada complexidade, onde a baixa seria o peso 1 (um), a média seria o peso 3 (três) e a alta seria o peso 5 (cinco). Após multiplicar cada funcionalidade pelo seu peso, realizou-se o somatório e chegou-se ao número de unidades de software que foram desenvolvidas por *sprint*. Quanto mais unidades de software a equipe desenvolver dentro de um *sprint*, mais produtiva ela será.

Na Tabela 5, é apresentada a quantidade de funcionalidades por complexidade que foram desenvolvidas em cada Sprint, bem como é calculado o somatório das complexidades vezes o peso. Como a equipe do projeto A é composta de 5 (cinco) profissionais, estimamos, em média, 800 (oitocentas) horas de trabalho por mês de toda a equipe. A partir daí, ao considerar as horas da equipe e dividir pelas unidades de software (quantidade x peso) chegamos à produtividade. Esse valor corresponde ao tempo necessário para realizar cada atividade de complexidade Baixa. Nesse projeto, a melhoria da produtividade em relação à média histórica foi de 23,37%.

Tabela 5 - Produtividade da equipe do projeto A

Peso	1	3	5			
Sprint	Baixa	Média	Alta	Qtde x Peso	Horas da Equipe	Produtividade
set/10	2	6	7	55	800	14,55
out/10	2	9	5	54	800	14,81
nov/10	3	2	9	54	800	14,81
dez/10	12	7	5	58	800	13,79
jan/11	2	8	5	51	800	15,69
fev/11	0	2	8	46	800	17,39
mar/11	7	2	9	58	800	13,79
abr/11	5	4	10	67	800	11,94
mai/11	8	3	10	67	800	11,94
jun/11	2	5	9	62	800	12,90
jul/11	6	13	6	75	800	10,67
ago/11	3	7	10	74	800	10,81
set/11	3	4	11	70	800	11,43
out/11	4	3	13	78	800	10,26
Total	59	75	117	869	11200	12,89
nov/11	1	10	10	81	800	9,88
Aumento de Produtividade (Em relação a média histórica do projeto)						23,37%

5.2.2. Estudo de Caso no Projeto B

Nesse cliente foi utilizada a técnica de pontos de função para medição do tamanho do software. Pela base histórica, a produtividade média de horas por ponto de função dos projetos era 9,20. Isso significa que a equipe de projeto gastava 9,20 horas para desenvolver cada ponto de função. Portanto, quanto menor o esforço em horas para desenvolver cada ponto de função, maior será a produtividade da equipe.

Na Tabela 6, estão descritos os 4 (quatro) projetos executados nesse cliente, o mês em que foram concluídos e a produtividade de cada projeto. Percebe-se que a produtividade

tem melhorado a cada projeto desenvolvido nesse cliente e isso se deve ao fato de a equipe ter adquirido mais conhecimento do processo e do negócio.

Tabela 6 - Produtividade da Equipe em um dos Clientes do Estudo de Caso

Projeto / Data	Pontos de Função Produzidos	Esforço em Horas	Horas / PF
Projeto 1 - Jun/2010	484,25	4.670,40	9,64
Projeto 2 - Dez/2010	337,68	3.140,40	9,30
Projeto 3 - Jun/2011	291,56	2.428,14	8,33
Total	1.113,49	10.238,94	9,20
Projeto B do estudo de caso			
Nov/2011	50,50	385,00	7,62
Aumento de Produtividade (Em relação a média histórica no cliente)			17,09%

Após a execução da abordagem proposta nesse trabalho e aplicação das “melhores práticas”, houve um aumento de produtividade bem superior ao que se tinha anteriormente. Não se pode indicar que esse aumento deve-se exclusivamente à aplicação das “melhores práticas”, mas, com certeza, a implementação das mesmas influenciou muito positivamente na melhoria da produtividade.

Na Figura 3, é perceptível a melhoria da produtividade em relação à média histórica da produtividade no cliente. Os 2 (dois) primeiros projetos ficaram com uma produtividade inferior à média histórica (acima de 9,20 horas por ponto de função), já o 3º Projeto ficou com uma produtividade melhor (8,33 horas por ponto de função). O projeto do estudo de caso em questão teve uma produtividade ainda melhor com 7,62 horas por ponto de função, que corresponde a uma melhoria de 17,09% em relação à produtividade média.

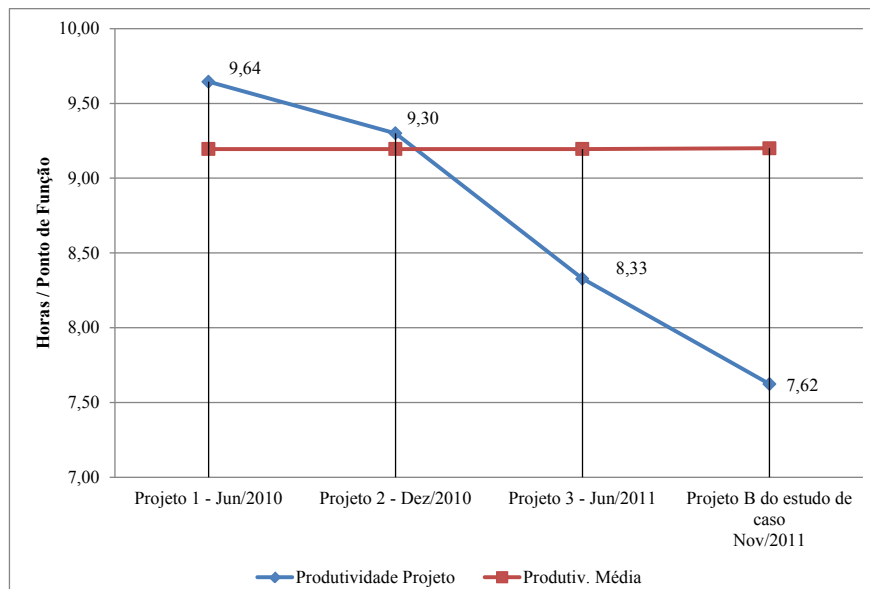


Figura 3 - Evolução da Produtividade em um dos Clientes do Estudo de Caso (Projeto B)

Após a execução do processo e análise dos dados, foram verificadas todas as hipóteses definidas no projeto desse estudo de caso.

Para cada projeto, foi realizada uma conversa com o gerente de projeto de forma a se obter o resultado da avaliação qualitativa das hipóteses H1, H2 e H3. Em seguida, foram analisados os resultados de melhoria da produtividade para a hipótese H4. A Tabela 7 apresenta com detalhes os resultados obtidos.

Como houve uma avaliação positiva de todas as hipóteses temos indícios de que a abordagem apresentada neste trabalho possui desempenho satisfatório e atende aos objetivos previstos, porém, como o estudo de caso está limitado a somente 3 (três) projetos e a uma única empresa, não é correto afirmar que essa abordagem será sempre satisfatória para todos os casos.

Tabela 7 - Verificação das hipóteses previstas do estudo de caso

HIPÓTESES				
PROJETO	H1 - A identificação dos principais fatores que influenciam na produtividade do desenvolvedor em um projeto possui uma alta relevância para que se possa melhorar a produtividade do desenvolvedor.	H2 - O uso de “melhores práticas” para mitigar os fatores que tem mais influência na produtividade do desenvolvedor no projeto pode aumentar a produtividade do mesmo.	H3 - O uso da abordagem proposta promove uma maior satisfação da equipe de desenvolvedores e consequentemente uma maior produtividade dos mesmos.	H4 - Após a execução do processo, haverá um aumento real de produtividade da equipe, levando em consideração a produtividade anterior média da empresa ou do projeto.
Projeto A	Sim. O Ranking é fundamental para que se possa atuar somente naqueles fatores que são relevantes na produtividade do desenvolvedor. No caso do Projeto A, o ranking do projeto ficou próximo do ranking padrão.	Sim. Quase 80% das práticas selecionadas do Projeto A foram aplicadas e apenas 7% (2 de 30 práticas) não foram eficazes	Sim. Os desenvolvedores se demonstraram satisfeitos em saber que a empresa está preocupada em melhorar a produtividade da equipe de desenvolvimento fazendo uso de melhores práticas.	Sim. Houve um aumento real de produtividade de 23,37% em relação à média histórica do projeto.
Projeto B	Sim. O Ranking é fundamental para que se possa atuar somente naqueles fatores que são relevantes na produtividade do desenvolvedor. No caso do Projeto B, apenas o fator "requisito em conformidade" ficou diferente do Ranking Padrão. Neste caso, esse fato não tem tanta relevância.	Sim. No caso do Projeto B, apesar de menos da metade das práticas selecionadas pelo gerente de projeto terem sido aplicadas, tivemos uma ótima avaliação de eficácia das práticas que foram aplicadas.	Sim. Os desenvolvedores se demonstraram satisfeitos em saber que a empresa está preocupada em melhorar a produtividade da equipe de desenvolvimento fazendo uso de melhores práticas.	Sim. Houve um aumento real de produtividade de 17,09% em relação à média histórica no cliente.
Projeto C	Sim. O Ranking é fundamental para que se possa atuar somente naqueles fatores que são relevantes na produtividade do desenvolvedor. No caso do Projeto C, entende-se que a imaturidade da equipe fez com que o Fator "Gerência de Projeto" ficasse bem diferente do Ranking padrão.	Sim. No caso do Projeto C, tivemos 2/3 das práticas selecionadas aplicada, e como o Projeto B, apenas 1 (uma) prática não foi eficaz.	Sim. Os desenvolvedores se demonstraram satisfeitos em saber que a empresa está preocupada em melhorar a produtividade da equipe de desenvolvimento fazendo uso de melhores práticas.	Sim. Houve um aumento real de produtividade de 57,42% em relação à média histórica no cliente.

6. Conclusão e Trabalhos Futuros

O tema “produtividade no desenvolvimento de software” tem bastante relevância no meio acadêmico e empresarial e cada vez mais, novas técnicas, metodologias e ferramentas são desenvolvidas para melhorar a produtividade no desenvolvimento de software.

No estudo de caso, verificou-se que a aplicação da abordagem na empresa de desenvolvimento de software gerou uma melhoria da produtividade e uma maior satisfação da equipe por contribuir diretamente nessa melhoria. Possivelmente, isso irá refletir em melhores resultados para a empresa.

O entendimento dos fatores que influenciam na produtividade dos desenvolvedores e a aplicação de “melhores práticas” para mitigar a influência desses fatores em um determinado projeto poderão influenciar positivamente na melhoria da produtividade, pois, muitas vezes, pequenas ações de melhoria em um determinado fator, que tem grande influência na produtividade, podem trazer altos ganhos.

A base de conhecimento gerada a partir do uso, por parte das empresas de software, da abordagem e da ferramenta MPDS propostos nesse trabalho, poderá ser bastante útil para a indústria de software, pois será possível determinar que fatores têm mais influência na produtividade, considerando múltiplas organizações e múltiplos projetos, bem como será possível saber que “melhores práticas” foram mais eficazes.

A abordagem apresentada poderia ser considerada como uma técnica de gerência de riscos em um projeto de software, pois o uso de “melhores práticas” para mitigar a influência dos fatores na produtividade poderá reduzir o risco do projeto.

Por último, ficou mais evidente que, pelo fato de o desenvolvimento de software ser uma atividade complexa e intensamente regida pelo conhecimento, é necessário um investimento contínuo no recurso que é mais importante para que as empresas tenham uma maior produtividade e uma maior qualidade: “as pessoas”.

Durante o desenvolvimento e a aplicação da abordagem proposta, algumas possibilidades de trabalhos futuros foram identificadas:

- Expansão do modelo para outros tipos de profissionais de desenvolvimento de software (gerente de projeto, analista de sistemas, analista de testes, dentre outros).
- Melhorias e evoluções na ferramenta MPDS, com a disponibilização da mesma em formato “*open source*” para uso da comunidade, permitindo que, no futuro, a ferramenta possa se integrar com outras ferramentas existentes.
- Aplicação do estudo de caso em um número maior de empresas e projetos de forma a obtermos uma base de conhecimento extensa sobre a influência dos fatores na produtividade dos projetos, bem como avaliação de eficácia do uso das “melhores práticas”.
- Avaliação, além da produtividade, de indicadores de qualidade, como densidade de defeitos, quantidade de erros por nível de severidade, dentre outros, de forma a obtermos um comparativo entre produtividade e qualidade.
- Realização de experimentos capazes de possibilitar a definição da relação de influência entre alguns fatores.

7. Referências

- Bernolak, I., "Effective measurement and successful elements of company productivity: the basis of competitiveness and world prosperity", *International Journal of Production Economics*, Vol. 52 No. 1-2, pp. 203-13, 1997.
- Joseph D. Blackburn, Gary D. Scudder, Luk N. Van Wassenhove, "Improving Speed and Productivity of Software Development: A Global Survey of Software Developers," *IEEE Transactions on Software Engineering*, vol. 22, no. 12, pp. 875-885, Dec. 1996.
- B. Boehm, "Software Engineering Economics", Prentice-Hall, Englewood Cliffs, NJ, 1981.
- CMMI-DEV, "CMMI for Development", V1.2 model, CMU/SEI-2006-TR-008. Software Engineering Institute, 2006.
- Drucker, PF, "The New Productivity Challenge," *Harvard Business Review* (69:6), pp. 69-79, November- December 1991.
- Grossman, E., "How to Measure Company Productivity: Handbook for Productivity Measurement and Improvement", Productivity Press, Cambridge, MA, 1993.
- Jones, C., "Program quality and programmer productivity". IBM Technical Report TR 02.764, pp. i, 42-78, January 1977.
- B. Kitchenham, "Procedures for Performing Systematic Reviews," joint technical report, Software Eng. Group, Keele Univ., and Empirical Software Eng., Nat'l ICT Australia, 2004.
- MPS.BR – Melhoria de Processo de Software Brasileiro, "Guia Geral (v. 1.0)", 2005.
- NBR ISO 9001 - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS.– "Sistema de Gestão da Qualidade – Requisitos", Rio de Janeiro, 2008.
- V. Nguyen, L. Huang and B. Boehm, "An analysis of trends in productivity and cost drivers over years", *ICSE International Conference on Software Engineering, Promise 2011 Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, ACM New York, NY, USA, 2011.
- Edgy Paiva, Danielly Barbosa, Roberto Lima and Adriano Albuquerque, "Factors that Influence the Productivity of Software Developers in a Developer View", *innovations in computing sciences and software engineering - 99-104*, doi: 10.1007/978-90-481-9112-3_17, 2010.
- Edgy Paiva, Danielly Barbosa, Roberto Lima and Adriano Albuquerque, "Fatores que influenciam a produtividade de desenvolvedores de software na visão dos gerentes de projeto" In: *IV International Conference on Research and Practical Issues of Enterprise Information Systems*, 2010, Natal.
- Chris Peck and Dale Callahan, "A Proposal for Measuring Software Productivity in a Working Environment," *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory* 2002, pp. 339-343, March 18-19, 2002.
- W. Scacchi, "Understanding Software Productivity", *Advances in Software Engineering and Knowledge Engineering*, Vol.4, Page 37-70, 1995.
- A. Trendowicz, J. Munch, "Factors Influencing Software Development Productivity— State-of-the-Art and Industrial Experiences", *Advances in Computers*, Vol. 77, pp.185-241, 2009.
- Robert K. Yin, "Case Study Research: Design and Methods", Third Edition, *Applied Social Research Methods Series*, Vol 5, 2003.
- Weider D. Yu, D. Paul Smith, and Steel T. Huang, "Software Productivity Measurements," *AT&T Technical Journal*, Volume 69, Number 3, pp. 110-120, May/June 1990.