A Importância Dos Desenvolvedores De Software Sob A Perspectiva Dos Supervisores

Guilherme Costantin Tângari

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2015

Guilherme Costantin Tângari

A Importância Dos Desenvolvedores De Software Sob A Perspectiva Dos Supervisores

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Marcelo de Almeida Maia

Uberlândia
2015

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada "**A importância dos desenvolvedores de software sob a perspectiva dos supervisores**" por **Guilherme Costantin Tângari** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação.**

Uberlândia, 13 de fevereiro de 2015

Orientador: _____
Prof. Dr. Marcelo de Almeida Maia
Universidade Federal de Uberlândia

Banca Examinadora:

_____
Prof. Dr. Autran Macêdo
Universidade Federal de Uberlândia

_____
Prof. Dr. Rogério Eduardo Garcia
Universidade Estadual Paulista

À minha família e à minha futura esposa.

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais, Gilberto e Valéria, que fizeram um enorme esforço para que eu pudesse ter acesso à educação que tive. Saibam que serei eternamente grato por todos os sacrifícios que fizeram e nenhuma herança no mundo supera a base educacional que vocês me proporcionaram. Sem dúvida nenhuma, vocês são os grandes responsáveis por eu ter chegado até onde cheguei e divido todos os méritos com vocês. Agradeço também à minha irmã Larissa por todo o apoio nesses últimos anos, também fundamental para que eu pudesse seguir em frente. De maneira especial, também gostaria de agradecer à minha noiva, que me deu todo o suporte, mental e estrutural, e me deu forças quando elas faltaram para seguir em frente. Liana, com certeza sem você esse caminho teria sido muito mais árduo, talvez impossível de ser traçado. Com seu apoio e sua motivação, consegui chegar ao fim dessa jornada, e também divido os méritos com você. Também gostaria de deixar um abraço especial para toda sua família, que sempre me deram apoio incondicional. Gostaria de agradecer também à empresa na qual trabalho atualmente, o MahaGestão, que apostou em mim como profissional e forneceu a flexibilidade necessária no decorrer do curso viabilizando assim esse mestrado. Não menos importante, gostaria de agradecer a todos os meus amigos, aos de longa data, que me acompanham desde o ensino básico e que continuam firme ao meu lado onde sempre estiveram sempre que precisei, às amizades feitas ao longo da minha graduação, que também já se tornaram amizades de longa data, e que compartilham de uma realidade de estudos e trabalhos semelhantes, sempre apoiando uns aos outros, e também à todos os outros grupos de amigos, do Maha, dos casais (amiguinhos), da Charanga, da minha noiva que eu adotei como meus (chicas), todos vocês foram muito importantes nessa jornada. Um salve especial ao Lucas (Lucão) pela sua colaboração com meu trabalho. E por último, gostaria de agradecer pela orientação do professor Marcelo Maia, seu apoio e compreensão foi fundamental para a conclusão desse trabalho e do mestrado como um todo.

"Precisar de dominar os outros é precisar dos outros. O chefe é um dependente."
(Fernando Pessoa)

Resumo

Várias empresas de tecnologia usam a quantidade de entregas como métrica de avaliação de performance de desenvolvedores de software. Esse é o conceito clássico de produtividade, e ainda é amplamente usado pelas empresas hoje em dia. Também é bastante comum misturar o conceito de importância com produtividade. Porém, a importância de um desenvolvedor para a empresa e, mais especificamente, o time em que trabalha não está apenas relacionado com a quantidade de linhas de código produzidos. Existe uma variedade de fatores que contribuem para a relevância de um desenvolvedor dentro de uma organização. Esse trabalho visa mapear alguns desses fatores, medir quais deles possuem maior influência e propor um modelo de avaliação da importância dos desenvolvedores que considere mais do que apenas as entregas. Levantamos, baseados em estudos passados, dezesseis fatores que mais tendem a participar da avaliação de importância dos desenvolvedores. Descobrimos que, dentre esses fatores, alguns são mais importantes que os outros, bem como uma variação nos fatores mais importantes quando olhamos sob a ótica de uma determinada empresa ou time. Nós também construímos um classificador de alta acurácia que pode indicar a importância do desenvolvedor baseado em uma série de atributos.

Palavras-chave: Importância do desenvolvedor. reconhecimento de padrões. produtividade. fatores humanos..

Abstract

Several technology companies use the amount of deliveries as evaluation metric for the software developer's performance. This is the classical concept of productivity, and still is widely used by the companies nowadays. It is also quite common to confuse the concepts of importance and productivity. But developer's importance for the company, and more specifically, for the respective team, is not related only with the amount of line of codes produced. There is a variety of factors that contribute to the relevance of a developer inside an organization. This work aims to map those factors, measure which ones has greater influence in today's companies and propose an evaluation model of the developer's importance that considers more than just deliveries. We raised, based on past studies, sixteen factors that are more likely to be used in the developer's importance evaluation. We figured out that, between those factors, some are more important than others, and that there is a variation in the most important factors when we analyze under the perspective from different companies or teams. We also built a high accuracy classifier that can identify the developer's importance based on a series of factors.

Keywords: Developer's importance. pattern recognition. productivity. human factors..

Lista de ilustrações

Lista de tabelas

Lista de siglas

FP *Function Points*

GQM *Goal-Question-Metric*

LOC *Lines of code*

WEKA *Waikato Environment for Knowledge Analysis*

Sumário

Capítulo 1

Introdução

Atualmente, as empresas em geral estão investindo em técnicas para aumento de produtividade, com o intuito de aumentar a competitividade no mercado, e isso não é diferente para a indústria de software, que permanece investindo em métodos, ferramentas e melhores práticas para ter um ganho na produção de seu software (de Barros Sampaio et al., 2010). Porém, diferentemente do hardware, que tem ganhos em ordens de magnitude de preço e performance por década, a produção de software parece ter dificuldade em evoluir (BOEHM, 1987). As taxas de produtividade atuais são similares às taxas de décadas atrás (uma ou duas linhas de código por homem-hora)(BOEHM, 1987). Brooks et al. (Brooks Jr, 1987) afirma ainda que não há nenhuma técnica, seja de tecnologia ou de gerenciamento, que por si só prometa um aumento de uma ordem de magnitude na produtividade, simplicidade e confiabilidade do software.

Os métodos tradicionais para se medir produtividade em desenvolvimento de software são baseados em linhas de código (LOC) e pontos de função (FP)(WAGNER; RUHE, 2008), por exemplo, a quantidade de LOC ou FP produzidos por um desenvolvedor em uma hora. Uma definição um pouco mais abstrata coloca produtividade como sendo os outputs entregues pelos inputs consumidos, podendo os outputs ser LOC, FP ou alguma outra saída considerada relevante, e inputs como recursos utilizados para produzir aquela saída (tempo, pessoal, etc), como mostrado na equação abaixo (BOEHM, 1987; WALSTON; FELIX, 1977; YU; SMITH; HUANG, 1991).

$$\text{Produtividade} = \frac{\text{outputs produzidos pelo processo}}{\text{inputs consumidos pelo processo}} \quad (1)$$

1.1 Problema e Motivação

Dentro das empresas de TI, geralmente é encontrado um departamento de recursos humanos ou desenvolvimento pessoal. Esses departamentos são responsáveis pela devida remuneração e determinação de cargos e posições dentro das empresas, e geralmente o fazem com base em avaliações de desempenho vinda dos supervisores responsáveis de cada

área. Geralmente, são encontrados um ou mais supervisores ou gerentes responsáveis na área de desenvolvimento, comumente conhecida pelos nomes de indústria ou fábrica de software.

Utilizar apenas o método tradicional acima para avaliar o desempenho de um desenvolvedor pode ser muito negativo para a empresa. Esse método de avaliar apenas LOC por exemplo é considerado primitivo e produz resultados que não estão de acordo com a realidade (SYMONS, 2010). Quantidade de linhas de código não leva em consideração o esforço necessário para escrevê-las, nem o conhecimento que serviu de base para o mesmo acontecer. Problemas complexos, por exemplo, geralmente necessitam de um desenvolvedor mais experiente para serem resolvidos, e muitas vezes, não são necessárias muitas linhas de código para o fazer.

Existem outras noções de produtividade que também não são levadas em consideração ao avaliar apenas linhas de código, por exemplo, um desenvolvedor com conhecimento em uma ferramenta muito específica, ou o supervisor, podem ser consultados frequentemente por outros desenvolvedores, agilizando e melhorando o processo de desenvolvimento de outros desenvolvedores, logo, esse desenvolvedor/supervisor possui uma produtividade indireta, que não é mensurada no método tradicional pois quando se está ajudando e ensinando, não se está escrevendo linhas de código.

Características comportamentais, encontradas no perfil do desenvolvedor, também podem influenciar na avaliação do desenvolvedor, aumentando sua relevância para a empresa. Pró-atividade, criatividade e liderança são alguns exemplos de características que geralmente fazem diferença na carreira de um funcionário dentro de uma organização. O método clássico também falha em não considerar esses quesitos, pois a avaliação de linhas de código não demonstra inovação, alinhamento com o negócio, etc.

Vários estudos se dedicam a levantar fatores que influenciam na produtividade, tanto no desenvolvimento quanto na manutenção de software, (de Barros Sampaio et al., 2010; WAGNER; RUHE, 2008; CALOW, 1991; VOSBURGH et al., 1984), dentre outros. Entender esses fatores e ter um mecanismo de avaliação de produtividade justo é muito importante para as empresas que desenvolvem qualquer tipo de software.

Retenção de talentos (BOEHM et al., 2000; CHATZOGLOU; MACAULAY, 1997; DEMARCO; TIMOTHY, 1987; GUZZO, 1988; SCUDDER; KUCIC, 1991; WOHLIN; AHLGREN, 1995; WOHLIN; ANDREWS, 2001) e motivação da equipe (BOEHM, 1987; DEMARCO; TIMOTHY, 1987; BOEHM, 1984; JONES, 2000; SHARP et al., 2009; BOEHM et al., 1982; BOEHM; PAPACCIO, 1988; HANTOS; GISBERT, 2000), por exemplo, são duas questões de extrema relevância para qualquer time. Motivação é tido como um dos fatores chave para o sucesso do software (SHARP et al., 2009), e software é feito por pessoas, e pessoas, quando tem seu trabalho reconhecido e valorizado tendem a produzir mais e melhor. Uma avaliação de desempenho que considere apenas um aspecto, como a quantidade de entregas, e não leva em conta a dificuldade e a finalidade do código,

e o relacionamento do dia a dia com os colegas e com a empresa, não é uma avaliação justa, que faz com que ocorra a desmotivação individual ou geral da equipe. Rotatividade de funcionários é um problema comum à empresas de software(ABDEL-HAMID; MADNICK, 1991; WALLACE; KEIL; RAI, 2004), e uma alta taxa de rotatividade pode levar à uma consequente diminuição de produtividade devida a uma perda de conhecimento (MELO et al., 2011; CORAM; BOHNER, 2005), além do aumento de custo com contratação e treinamento e o mais importante, a perda de talentos que vão em busca de reconhecimento em outro empresa (até mesmo no concorrente).

Várias empresas estão começando a ganhar consciência dessas questões e estão motivadas a melhorar a forma que são feitas as avaliações de desempenho dos desenvolvedores. Esse trabalho visa investigar como os supervisores entendem a noção de importância, indicando quais fatores são mais relevantes em suas avaliações sobre os desenvolvedores. Estamos procurando responder essas perguntas:

1. Quais são os critérios mais importantes usados pelos supervisores em sua classificação sobre os desenvolvedores?
2. É possível construir um classificador de desenvolvedores de alta acurácia, usando os critérios propostos? Essa pergunta pode ser refinada em duas novas perguntas:
   a) É possível se obter um classificador genérico, i.e., independente de empresa?
   b) É mais adequado construir classificadores customizados para cada empresa?

1.2 Objetivos e Contribuição

Como mencionado anteriormente, o desempenho de um desenvolvedor é muito relacionado com sua produtividade, que possui um conceito clássico de quantidade de entregas. Porém, os supervisores e gerentes que participam ativamente da sua área de desenvolvimento e convivem com os desenvolvedores que ali trabalham, possuem percepções diferentes sobre cada membro de seu time. Isso nada mais é que uma avaliação subjetiva de desempenho, que não considera apenas a produção individual resultante, mas diversas outras características que fazem que o desenvolvedor possua uma determinada importância na visão do seu supervisor.

Entende-se que a métrica mais comum hoje é a "produtividade" no sentido de quantidade de entregas. Porém, muito mais é levado em consideração ao avaliar a importância de um determinado desenvolvedor, o problema é que essa avaliação geralmente acontece de uma forma subjetiva por parte do supervisor, isto é, cada supervisor, de acordo com a vivência, identifica quais os desenvolvedores mais importantes devido à sua percepção, sem a orientação de métricas bem estabelecidas.

Foi feito um levantamento, baseado em estudos passados, de várias métricas que podem influenciar na avaliação de cada desenvolvedor. Será realizado uma pesquisa com

sujeitos humanos (supervisores de áreas de desenvolvimento representando empresas ou times), onde cada um desses sujeitos humanos irá avaliar individualmente os desenvolvedores seguindo as métricas levantadas. Será então analisado o conjunto de dados resultante da aplicação da pesquisa com intuito de identificar um padrão nas avaliações dos desenvolvedores. Dessa forma, pretende-se apontar quais as métricas que mais influenciam na avaliação dos supervisores sobre os desenvolvedores, e também obter um classificador de importância dos desenvolvedores de alta acurácia, utilizando essas métricas mais relevantes. Os supervisores que preencherem uma quantidade significativa de dados (10 ou mais questionários) também terão sua empresa/time analisados individualmente, tanto na descoberta das métricas mais relevantes como na construção do classificador de importância do desenvolvedor.

1.3 Estrutura da Dissertação

Além do presente capítulo introdutório, esta pesquisa apresenta-se desenvolvida e documentada dentro da seguinte estrutura organizacional:

Capítulo 2: Referencial Teórico

Nesse capítulo serão apresentados os conceitos importantes que serão abordados ao longo desse trabalho, bem como outros estudos que atuaram em cima desse mesmo tópico e que serviram de base para produção desse estudo.

Capítulo 3: Metodologia

Nesse capítulo será apresentado a metodologia usada para conduzir a pesquisa com sujeito humanos, mostrando a criação do conjunto de critérios para os supervisores avaliarem os desenvolvedores e a aplicação da pesquisa. Será mostrado também a estratégia de seleção de características utilizada para identificar quais critérios possuem maior relevância no momento da avaliação do supervisor, e a estratégia para a construção do classificador de importância dos desenvolvedores.

Capítulo 4: Resultados Gerais (todas empresas)

Nesse capítulo será apresentado o resultado da aplicação da pesquisa, mostrando o número de supervisores e empresas envolvidos na pesquisa e a quantidade de avaliações de desenvolvedores obtidas. Será mostrado também o resultado da aplicação dos algoritmos de seleção de características e de classificação para todas as empresas.

Capítulo 5: Resultados individuais (por empresa)

Nesse capítulo será apresentado o resultado da aplicação da pesquisa, aplicação dos algoritmos de seleção de características e de classificação para as empresas que atingirem o número mínimo de avaliações de desenvolvedores.

Capítulo 6: Discussão

Aqui serão discutidos os resultados obtidos da aplicação dos algoritmos de seleção de características e classificação, apresentando também algumas ameaças à validade desse estudo.

Capítulo 7: Conclusões

Por fim, baseada nos resultados obtidos, será apresentada a conclusão do trabalho e serão levantadas algumas possibilidades de trabalhos futuros que possam ter como base esse estudo.

Capítulo 2

Referencial Teórico

Este estudo gira em torno do conceito de importância dos desenvolvedores, sob as perspectivas dos seus supervisores. Porém, notamos que esse conceito se mistura com conceitos de produtividade, performance ou eficiência. Vários estudos, que serviram de base para levantar as métricas para os supervisores avaliarem os desenvolvedores, eram na verdade estudos sobre produtividade e controle de custos e qualidade de software.

Ao longo do tempo, vários estudos se dedicaram a encontrar e classificar os chamados fatores de produtividade (VOSBURGH et al., 1984; WALSTON; FELIX, 1977; BROOKS, 1981; HANSON; ROSINSKI, 1985; JONES, 1986; BOEHM; PAPACCIO, 1988; JONES, 1997; SCUDDER; KUCIC, 1991; BANKER; DATAR; KEMERER, 1991; BOEHM, 1984; BANKER; DATAR; KEMERER, 1987; SCACCHI; HURLEY, 1995; BRIAND; EMAM; BOMARIUS, 1998; JONES, 2000; LOKAN, 2001; CLINCY, 2003; WAGNER; RUHE, 2008; de Barros Sampaio et al., 2010). Em COCOMO (BOEHM et al., 2000) (Constructive Cost Model) considerado um dos estudos mais proeminentes na área, Boehm classifica os fatores de produtividade diversas categorias (produto, projeto, etc.). Abstraindo ainda mais as categorias, ele classifica os fatores em fatores técnicos e fatores do tipo "soft" (WAGNER; RUHE, 2008), que são os fatores não-técnicos que influenciam na produtividade.

De Marco e Lister (DEMARCO; TIMOTHY, 1987) apontaram que "talvez os maiores problemas de trabalhar com sistemas não são tanto tecnológicos quanto sociológicos". Em seu estudo, por exemplo, eles afirmam que um dos fatores que mais influencia na produtividade é a rotatividade de funcionários (como ressaltado na Seção 1.1). Em suma, eles proporcionaram o primeiro e mais compreensível estudo sobre os fatores "soft" influenciando na produtividade dos desenvolvedores de software (WAGNER; RUHE, 2008).

Como este trabalho se dedica a medir a importância do desenvolvedor, seu foco estará mais voltado para os fatores do tipo "soft" do que para os fatores do tipo técnico (que envolvem fatores relativos à produto e projeto, como tamanho do software, levantamentos de requisitos , etc (de Barros Sampaio et al., 2010)).

2.1 Levantamento de Características

Nessa seção serão mostrados quais os fatores do tipo "soft" serão utilizados para traduzir do abstrato para o concreto a avaliação dos supervisores sobre seus desenvolvedores. Serão mostrados também estudos onde esses fatores foram referenciados, sob o contexto de avaliação de fatores que influenciam na produtividade dos desenvolvedores de software. Esses fatores, como mostrado na Seção 3.1.1, serão utilizados para compor as métricas do modelo Goal-Question-Metric (GQM), que auxiliará na condução da pesquisa com as empresas.

Primeiramente, é importante citar que foram agrupados os fatores levantados por semelhança de conceito de forma a deixar o levantamento mais conciso. Esses grupos ajudarão na elaboração das perguntas do modelo GQM. São eles:

Características técnicas

Aqui estão agrupados os fatores relativos às habilidades do desenvolvedor. Dentre os fatores mais citados na literatura, foram selecionados quatro, que contemplam a experiência do desenvolvedor em um determinado método ou ferramenta, se ele possui conhecimento especializado em uma determinada tecnologia, ou mesmo se possui uma diversidade de conhecimentos em variadas tecnologias, e sua capacidade em resolver problemas complexos.

Características comportamentais (quando em equipe)

Foram encontrados diversos estudos que mostram como a coesão e a comunicação de um time de desenvolvimento influencia na produtividade dos desenvolvedores pertencentes a esse time. As métricas foram divididas sobre o comportamento do desenvolvedor quando encontra um problema (se ele é do tipo introspectivo que tenta resolver sozinho ou comunicativo que logo consulta alguém do time), quando algum colega solicita ajuda e de um modo geral como é a sua comunicação com seus colegas de time.

Características Individuais (encontradas no perfil do desenvolvedor)

Aqui é necessário entrar um pouco na ciência de Gestão de Pessoas. Para usar como fatores de importância do desenvolvedor, foram mapeadas algumas das competências citadas como mais importantes para as organizações hoje em dia. Competência constitui um repertório de comportamentos capazes de integrar, mobilizar, transferir conhecimentos, habilidades, julgamentos e atitudes que agregam valor econômico à organização (BOYATZIS, 1982; BOYATZIS, 2008; SHIRAZI; MORTAZAVI, 2009). As competências selecionadas são mostradas na Tabela 1 e os estudos que as referenciam geralmente estão ligados a estudos de gestão por competências.

Compromisso com o Time/Empresa

Amplamente encontrado em literatura relativa às metodologias de desenvolvimento

ágil, por serem fatores que baseiam a filosofia ágil que visa melhorar a produtividade de um time, estão os fatores mapeados à essa categoria. Foco nos clientes, nos resultados e organização são alguns deles. Outro fator também se encaixa bem à essa categoria, que é o tempo de trabalho do desenvolvedor na empresa, que representa a fidelização do funcionário.

Tabela 1 – Levantamento dos fatores de importância por grupo de semelhança

Agrupamentos	**Fatores de importância**	**Referências (Tabela 2)**
Características técnicas	Experiência relevante	(1)
	Conhecimento especializado	
	Diversidade de habilidades	
	Capacidade de resolução de problemas complexos	
Características comportamentais	Principal comportamento do desenvolvedor ao encontrar um problema	(2)
	Disposição para ajudar colegas quando solicitado	
	Comunicação com os colegas	
Características individuais	Liderança	(3)
	Pró-atividade	
	Criatividade	
	Empreendedorismo	
Compromisso com o Time/Empresa	Foco no cliente	(4)
	Foco nos resultados	
	Organização e planejamento	
	Tempo de trabalho	

Tabela 2 – Revisão da literatura correlata aos fatores de importância

Grupos de fatores	**Estudos onde os fatores são citados**
1	(CHATZOGLOU; MACAULAY, 1997; COLE, 1995; JONES, 1986; MAXWELL; FORSELIUS, 2000; BANKER; DATAR; KEMERER, 1991; BOEHM et al., 2000; BROOKS, 1981; FINNIE; WITTIG; PETKOV, 1993; JONES, 2000; LAKHANPAL, 1993; SCUDDER; KUCIC, 1991; TURCOTTE; RENNISON, 2004; VOSBURGH et al., 1984; WALSTON; FELIX, 1977; WOHLIN; AHLGREN, 1995; WOHLIN; ANDREWS, 2001)
2	(ALPER; TJOSVOLD; LAW, 2000; BOEHM et al., 2000; CHATZOGLOU; MACAULAY, 1997; LAKHANPAL, 1993; RASCH, 1991; SCUDDER; KUCIC, 1991; VOSBURGH et al., 1984; WALSTON; FELIX, 1977; WOHLIN; AHLGREN, 1995; LALSING; KISHNAH; PUDARUTH, 2012)
3	(BOYATZIS, 1982; BOYATZIS, 2008; SHIRAZI; MORTAZAVI, 2009; Faria Sueli, 2005; DUTRA, 2004; FLEURY; FLEURY, 2001)
4	(LALSING; KISHNAH; PUDARUTH, 2012; MELO et al., 2011; Faria Sueli, 2005; SCHWABER, 2004; CORAM; BOHNER, 2005)

2.2 Data Mining e Classificação

Segundo Witten (HOLMES; DONKIN; WITTEN,), Mineração de dados se trata de coletar os dados brutos e transformá-los em algo mais útil, como informação ou predições de algo que pode vir a acontecer, que podem ser úteis no mundo real.

Para realizar as análises deste estudo, será utilizado uma ferramenta chama WEKA (Waikato Environment for Knowledge Analysis)(HOLMES; DONKIN; WITTEN,). Weka é um software open-source que contém um grande número de algoritmos para classificação, pré-processamento de dados, seleção de características, clusterização, regras de associação, etc. Será feito uso desses algoritmos de aprendizado de máquina, para realizar a mineração de dados no conjunto de dados resultante da pesquisa.

O uso de algoritmos de aprendizado de máquinas não é novidade no estudo sobre a produtividade dos desenvolvedores. Sharpe et al. (SHARPE; CANGUSSU, 2005) utilizou algoritmos de reconhecimento de padrões, mais especificamente, algoritmos de clusterização, para classificar os desenvolvedores de acordo com sua produtividade. Neste caso, como a avaliação dos desenvolvedores pelos supervisores será obtida em cima de um con-

junto pré-definido de classes, serão utilizados algoritmos de seleção de características para selecionar apenas as características mais relevantes nessa avaliação e algoritmos de classificação que utilizarão essa avaliação dos supervisores como base (classe) para tentar obter a maior acurácia possível perante essa classificação prévia feita pelos supervisores.

Capítulo 3

Metodologia

Para investigar a prática atual de avaliação de desenvolvedores, é preciso olhar mais profundamente em como as empresas o fazem. No propósito de conduzir esse estudo, foi decidido executar uma pesquisa com sujeitos humanos para extrair a informação desejada e analisá-la. Neste estudo foi pedido para os respondentes primeiro classificar o desenvolvedor (em níveis de importância), e depois preencher o resto do formulário com as características do desenvolvedor.

Para analisar os dados obtidos, serão utilizados métodos de classificação para se obter uma visão clara em como as características afetam a classificação dos supervisores, e como produto, dependendo dos dados encontrados, obter um classificador que possa ser utilizado para ajudar os supervisores a ganhar *insights* de como melhorar a importância geral do time.

Esta seção se dedica a explicar o desenho desse experimento, mostrando a criação e a aplicação da pesquisa com os supervisores, e mostrar também como foi conduzida a análise dos dados obtidos dessa pesquisa, incluindo a análise de critérios e a construção do classificador.

3.1 Pesquisa

Nas próximas subseções serão tratadas a abordagem utilizada para compor a pesquisa, a aplicação da pesquisa em si nos supervisores das empresas participantes e a caracterização dos mesmos.

3.1.1 Goal-Question-Metric

Foi utilizada uma abordagem chamada Goal-Question-Metric(GQM) (BASILI; CALDIERA; ROMBACH, 1994) que ajudou a compor a pesquisa. GQM é uma abordagem do tipo top-down, que é baseada no pressuposto que primeiro, para se medir algo, é necessário especificar objetivos (goal), dos quais é possível derivar perguntas (question), e

então, especificar as métricas (metric) que precisam ser coletadas para responder essas perguntas.

Para atingir o propósito de um objetivo, é necessário determinar 3 coordenadas:

a) **Questão *(issue)*:** Qual questão/assunto está sendo lidado;

b) **Objeto/Processo *(object/process)*:** Qual é o objeto central da análise;

c) **Ponto de vista *(ViewPoint)*:** Sob a perspectiva de quem a análise está sendo feita.

A Tabela 3 mostra o modelo GQM, com o propósito, as perguntas derivadas do mesmo e as métricas levantadas para responder essas perguntas.

3.1.2 Aplicação da Pesquisa

A pesquisa foi aplicada de maneira remota, para dar a liberdade necessária para os respondentes participarem da pesquisa. Para isso, foi utilizada a ferramenta de formulários do Google (Google Form).

Para preservar a privacidade das empresas participantes, não foi solicitado nenhum tipo de identificação, tanto do respondente como do desenvolvedor sendo analisado. A única identificação solicitada foi o nome da empresa de onde vinham as avaliações. Foi necessário solicitar esse dado para uma investigação mais profunda nos casos em que as empresas atingiam o mínimo de 10 desenvolvedores avaliados.

3.1.3 Caracterização dos respondentes

A pesquisa foi projetada para ser aplicada em empresas que possuem uma área de desenvolvimento de software, com uma estrutura hierárquica mínima onde existisse o cargo de supervisor, ou gerente, ou engenheiros-chefe, etc (para futuras referências, essa pessoa será referenciada como supervisor).

Os respondentes da pesquisa são então supervisores de times de desenvolvimento. Entende-se que eles são as pessoas certas para o fazer, porque diferentemente do dono da empresa eles estão perto o suficiente do dia a dia de trabalho, e conseguem julgar quais os desenvolvedores mais importantes, mesmo que eles não usem um método formalizado para tal. Eles devem responder a avaliação para cada desenvolvedor.

3.2 Seleção de Características

No intuito de conduzir a análise dos critérios para determinar quais fatores são os mais relevantes e possuem maior influência na avaliação do supervisor, foi utilizada a ferramenta WEKA(HOLMES; DONKIN; WITTEN,).

Tabela 3 – Goal-Question-Metric

Objetivo	**Propósito**	Medir
	Questão	a importância
	Objeto	do desenvolvedor
	Ponto de vista	sob a perspectiva do supervisor
Pergunta	Qual a produtividade desse desenvolvedor?	
Métricas	Quantidade de entregas por mês	
	Avaliação subjetiva da produtividade	
Pergunta	Qual o nível de habilidade técnica que o desenvolvedor em questão possui?	
Métricas	Experiência relevante	
	Conhecimento especializado	
	Diversidade de habilidades	
	Resolução de problemas complexos	
Pergunta	Qual o nível de habilidade interpessoal que o desenvolvedor em questão possui?	
Métricas	Comunicação com colegas	
	Disposição para ajudar colegas quando solicitado	
Pergunta	Quando se depara com um problema, qual o principal comportamento do desenvolvedor?	
Métricas	(Introspectivo) Tenta resolver sozinho	
	(Introspectivo) Busca na documentação e livros	
	(Comunicativo) Procura ou pergunta em sites de perguntas e respostas	
	(Comunicativo) Pede ajuda para os colegas ou supervisores	
Pergunta	Qual o nível dessas características no perfil comportamental do desenvolvedor?	
Métricas	Liderança	
	Criatividade	
	Empreendedorismo	
	Pró-atividade	
Pergunta	Qual o compromisso do desenvolvedor em relação à empresa?	
Métricas	Tempo de trabalho	
	Organização e planejamento	
	Foco no cliente	
	Foco nos resultados	

Vários problemas reais possuem diversas características envolvidas (mostradas na Tabela 3), e apenas algumas dentre elas são relevantes para o objetivo final (KIRA; RENDELL, 1992), neste caso, classificar a importância de um desenvolvedor. Para resolver esse problema, será utilizada uma estratégia chamada seleção de características (do inglês, Feature Selection), onde será selecionado um subconjunto de características para

se ter em foco, e ignorar o restante para acelerar o processo de aprendizado, aprimorar a qualidade do classificador e atingir a melhor acurácia do algoritmo de aprendizado de máquinas (KIRA; RENDELL, 1992; KOHAVI; JOHN, 1997).

O algoritmo que será utilizado é chamado GainRatioAttributeEval, que avalia os atributos um a um independentemente e os ordena de acordo com sua influência sobre a classe. A seleção de características escolhe baseada nessa ordenação, e isso permite eliminar atributos irrelevantes. Esse método, como não avalia um subconjunto de atributos, não elimina atributos redundantes (apenas os irrelevantes), mas isso não é um problema, pois todos os atributos são conhecidos e esse tipo de avaliador não necessita de um método de busca, o que o torna muito rápido em sua execução.

3.3 Classificação

Como resultado dessa pesquisa, será gerado um conjunto com várias avaliações de supervisores sobre os desenvolvedores. Nesse conjunto serão aplicados algoritmos de aprendizado de máquinas, no intuito de gerar um classificador de importância. Os algoritmos de aprendizado de máquinas precisam de dois conjuntos de dados: um para realizar o treinamento e um para testar o resultado, para verificar a acurácia do classificador. A Figura 1 mostra o processo que melhor representa esse cenário.

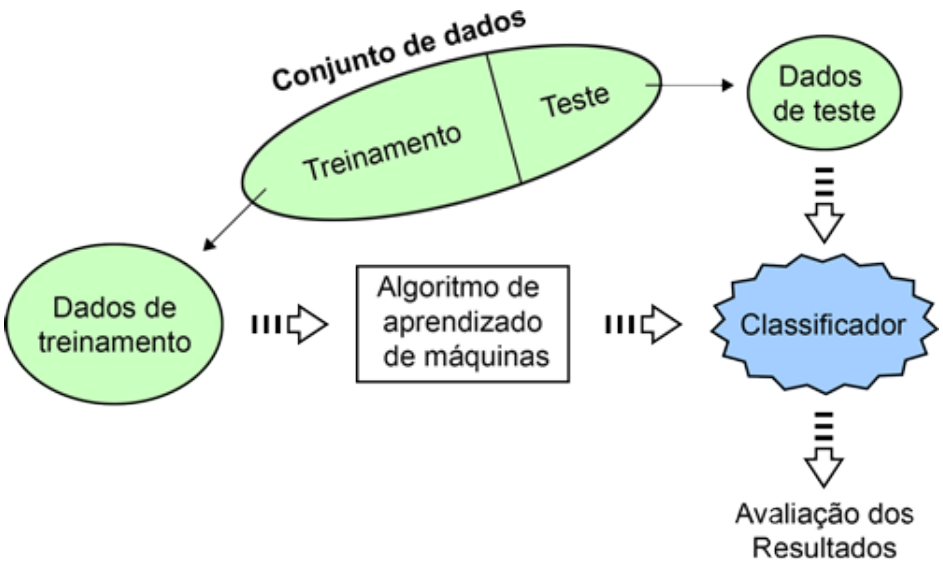

Figura 1 – Processo dos algoritmos de aprendizado de máquinas

Existe uma variedade de métodos para avaliar a performance de um classificador. É possível, por exemplo, selecionar uma porcentagem de um conjunto de dados para aplicar os algoritmos de aprendizado de máquinas e usar o resto para teste. O método de avaliação padrão e mais confiável é chamado cross-validation. Neste caso, será utilizado o 10-fold cross-validation que divide o conjunto em 10 partes iguais (as chamadas folds), usa 9 partes dessa divisão para treinamento e guarda uma parte para teste, e depois, faz isso

mais 9 vezes, sempre alternando o pedaço utilizado para teste, assim, uma determinada parte é usada 9 vezes para treinamento e uma para teste. O resultado será a média das 10 vezes que o algoritmo executou.

Os algoritmos de aprendizado de máquina que serão utilizados são o J48, um classificador em árvore, e o NaïveBayes, um classificador bayesiano.

J48 é uma variação de um famoso sistema chamado C4.5, que foi descrito por Quinlan (QUINLAN, 1993) que usa árvores de decisão para construir um classificador (WEKA inclusive fornece uma visão da árvore gerada com todos os seus pesos.

NaïveBayes é um método probabilístico que possui duas assunções: que os atributos são igualmente importantes e estatisticamente independentes (essa assunção de independência nunca é 100% correta, mas os métodos baseados nela geralmente funcionam bem na prática). Observando as métricas que serão coletar e o conjunto de dados que será gerado após a aplicação da pesquisa, esses dois métodos parecem servir bem aos propósitos deste estudo.

Será utilizado também um terceiro algoritmo chamado AttributeSelectClassifier, que na verdade usa um método de seleção de características (nesse caso, o GainRatio) e um algoritmo para executar a classificação (nesse caso, J48 ou NaïveBayes). Esta é uma maneira mais apropriada para aplicar seleção de características porque assim o algoritmo seleciona as características no conjunto de treinamento, tornando os resultados mais confiáveis.

Por fim, para conduzir todas essas análises será utilizado um modo do programa WEKA chamado EXPERIMENTER, que permite rodar o mesmo experimento mais de uma vez e determinar a média e o desvio padrão, para evitar falsos resultados otimistas ou pessimistas (alta ou baixa acurácia) devido à seleção de atributos nos conjuntos de treinamento e teste. Esse modo permite comparar os resultados de diferentes algoritmos.

Capítulo 4

Resultados Gerais (Todas Empresas)

Seguindo os passos apresentados no capítulo anterior, serão mostrados os resultados da aplicação da pesquisa, a seleção de características executada no conjunto de dados gerados por essa pesquisa e a aplicação dos classificadores e suas respectivas acurácias na classificação dos desenvolvedores.

4.1 Pesquisa

Primeiramente serão apresentados os dados coletados com a pesquisa. Onze respondentes (supervisores) forneceram 61 respostas (avaliações únicas de desenvolvedores). Em alguns casos, alguns supervisores trabalham na mesma empresa, porém eles supervisionam diferente times. No total, oito empresas foram envolvidas na coleta de dados. A tabela 4 mostra detalhes sobre a participação de cada empresa. Assim como mostrado na Seção 3.1.3, as empresas participantes são empresas que possuem uma área de desenvolvimento de software com ao menos um supervisor. Todas as empresas estão situadas na cidade de Uberlândia, e não foi utilizado nenhum fator para discriminar a participação das mesmas na pesquisa, como porte da empresa, número de funcionários ou infra-estrutura por exemplo. De fato, participaram grandes empresas, em faturamento e plantel de funcionários, que fazem parte de grupos ainda maiores, como empresas que possuem um pequeno número de funcionários (até 10), formando assim um grupo bastante heterogêneo de empresas participantes.

Foi pedido para os supervisores classificarem os desenvolvedores em 5 graus de importância. Esses graus de importância e a distribuição dos desenvolvedores avaliados neles é mostrado na Figura 2.

Analisando os resultados da pesquisa, chegou-se à conclusão que os supervisores foram, em certo nível, conservadores em classificar seus desenvolvedores nas classificações mais baixas de importância. Uma possível explicação para esse comportamento é o supervisor, invés de analisar de uma forma comparativa os desenvolvedores ao realizar a classificação por importância, ter avaliado cada desenvolvedor individualmente, o que pode ter criado

Tabela 4 – Relação dos participantes na pesquisa

Empresa	**Quantidade de supervisores/times**	**Quantidade de desenvolvedores avaliados**
A	2	20
B	1	10
C	3	20
D	1	4
E	1	3
F	1	2
G	1	1
H	1	1
total	11	61

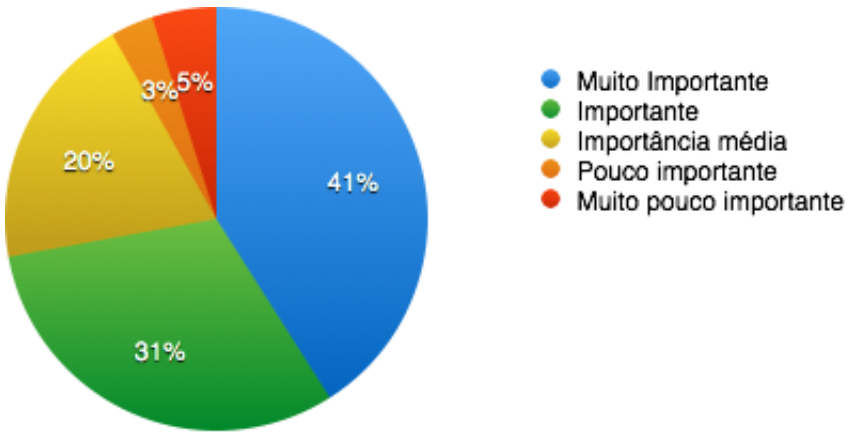

Figura 2 – Distribuição dos desenvolvedores avaliados pelas classes de importância

um viés ao seguir a linha de raciocínio de que se o desenvolvedor possui baixa importância, ele não estaria na equipe.

Esse viés na classificação dos desenvolvedores impacta diretamente a acurácia dos classificadores, visto que eles usam a classe como base para tentar descobrir um padrão nos dados. Caso a classificação não tenha ficado clara para todos os respondentes, os algoritmos podem se confundir e apresentar erros leves, classificando um determinado indivíduo em classes similares, mas não à classe que ele realmente pertence. Esses erros, mesmo que leves, em grande quantidade interferem e possuem um impacto negativo significativo na acurácia do classificador que é proposto nesse estudo.

A partir dessa análise, juntamente com a análise da matriz de confusão gerada pela aplicação dos algoritmos descritos na Seção 3.3, decidiu-se agrupar os desenvolvedores em apenas duas classes, baseadas nas cinco classes originais de importância, como mostrado na Tabela 5, com o intuito de obter uma análise mais realística.

Tabela 5 – Novo conjunto de classes de desenvolvedores

Novo conjunto de classes	**Conjunto de classes originais**
Alta importância	Muito importante
	Importante
Baixa importância	Importância média
	Pouco importante
	Muito pouco importante

Utilizando esse novo conjunto de classes para agrupar os desenvolvedores, obteve-se a distribuição mostrada na Figura 3.

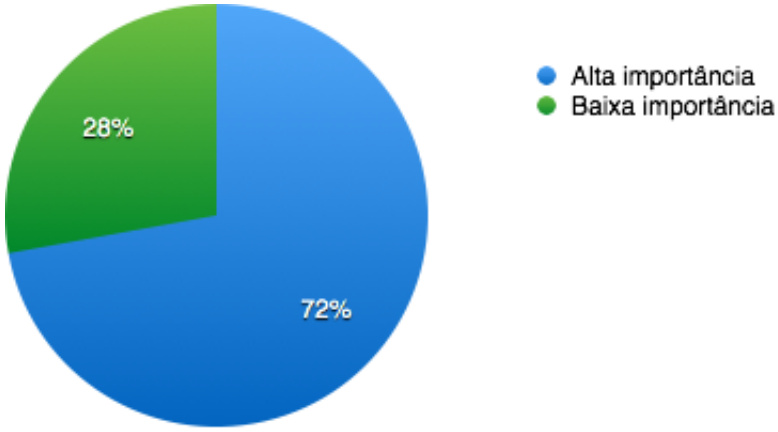

Figura 3 – Distribuição dos desenvolvedores avaliados pela nova classe de importância

4.2 Seleção de Características

Como explicado na Seção 3.2 , foi utilizado o algoritmo *GainRatio* para ordenar os atributos propostos, para prosseguir com a seleção de características. A Tabela 6 mostra as características ordenadas pelo mérito médio (grau de influência do atributo na classificação, calculado pelo algoritmo) resultante da aplicação do algoritmo usando as classes originais e a Tabela 7 mostra a mesma visão, utilizando agora o novo conjunto de 2 classes.

É Possível notar algumas semelhanças e diferenças entre essas tabelas. Os três primeiros atributos da primeira tabela, por exemplo, ocupam também as três primeiras posições na segunda, apesar de estarem em ordem diferente, o que mostra que a mudança do conjunto de classes não teve um impacto muito significativo na relevância desses atributos.

Porém, após os três primeiros, é possível notar algumas diferenças mais expressivas no posicionamento dos atributos, mostrando então um maior impacto nos atributos menos relevantes causada pela troca do conjunto de classes.

Tabela 6 – Ordenação dos atributos (conjunto original de 5 classes)

Atributos	**Posição média**	**Mérito médio**
Capacidade de resolução de problemas complexos	1.4 +- 0.49	0.303 +- 0.03
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	1.6 +- 0.49	0.29 +- 0.022
Pró-atividade	3.6 +- 0.92	0.226 +- 0.01
Experiência relevante	4.8 +- 1.17	0.211 +- 0.014
Diversidade de habilidades	5.6 +- 1.36	0.202 +- 0.022
Conhecimento especializado	6.1 +- 2.51	0.2 +- 0.026
Tempo de trabalho (meses)	7.1 +- 1.37	0.184 +- 0.012
Criatividade	7.4 +- 1.62	0.18 +- 0.021
Foco nos resultados	8.7 +- 1.27	0.167 +- 0.017
Foco no cliente	10 +- 2.45	0.148 +- 0.023
Principal comportamento do desenvolvedor	11.1 +- 1.58	0.14 +- 0.021
Comunicação com os colegas	11.6 +- 1.02	0.137 +- 0.011
Organização e planejamento	12.3 +- 1.27	0.122 +- 0.015
Liderança	14.5 +- 1.2	0.097 +- 0.018
Empreendedorismo	15 +- 0.89	0.087 +- 0.012
Disposição para ajudar colegas quando solicitado	15.2 +- 0.6	0.084 +- 0.014

Tabela 7 – Ordenação dos atributos (novo conjunto de 2 classes)

Atributos	**Posição média**	**Mérito médio**
Pró-atividade	1.2 +- 0.4	0.168 +- 0.01
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	1.9 +- 0.54	0.156 +- 0.017
Capacidade de resolução de problemas complexos	3.2 +- 0.6	0.126 +- 0.013
Foco nos resultados	4.8 +- 0.98	0.112 +- 0.018
Experiência relevante	4.9 +- 1.3	0.107 +- 0.018
Criatividade	6.5 +- 1.36	0.095 +- 0.008
Organização e planejamento	6.9 +- 2.21	0.09 +- 0.014
Diversidade de habilidades	8.1 +- 1.14	0.08 +- 0.011
Conhecimento especializado	9.2 +- 1.78	0.071 +- 0.013
Foco no cliente	10.3 +- 1.95	0.063 +- 0.012
Tempo de trabalho (meses)	10.8 +- 1.08	0.063 +- 0.01
Disposição para ajudar colegas quando solicitado	11.6 +- 2.24	0.057 +- 0.019
Liderança	12 +- 0.89	0.052 +- 0.004
Comunicação com os colegas	13.6 +- 0.92	0.039 +- 0.009
Empreendedorismo	15.5 +- 0.5	0.015 +- 0.005
PRINCIPAL comportamento do desenvolvedor	15.5 +- 0.5	0.016 +- 0.007

4.3 Classificação

Considerando que os atributos foram ordenados, foi aplicado a técnica de seleção de características e usado apenas os atributos mais relevantes na classificação. Para determinar quantas características precisam ser selecionadas para se obter uma maior performance, foi conduzido um teste exaustivo (os classificadores rodaram com um crescente número de características selecionadas, de 2 a 16) e escolhido a configuração com melhor performance (8 atributos). A Tabela 10 e a Tabela 11 mostram então a aplicação exaustiva do algoritmo que associa a seleção de características com os algoritmos de classificação, para o J48 e para o NaïveBayes respectivamente.

A Tabela 8 mostra os resultados da aplicação do algoritmo J48.Neste caso, o J48 não apresentou uma boa performance, com acurácia perto de 60%, para o conjunto original de classes, porém já apresentou uma acurácia significativa ao ser aplicado sobre o novo conjunto de classes (80% de acertos). No caso do J48, para ambos os conjuntos de classe, o classificador obteve a melhor performance utilizando apenas 2 atributos, como mostrado na Tabela 10.

NaïveBayes, como mostrado na Tabela 9 obteve uma performance similar ao J48 quando aplicado no conjunto original de classes (acurácia perto de 61%). Já para o novo conjunto de classes, NaïveBayes atingiu uma acurácia expressiva de 85%. Ambos os melhores resultados do NaïveBayes foram atingidos utilizando 8 características, como mostra a Tabela 11.

Tabela 8 – Aplicação do J48 para os diferentes conjuntos de classe

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	60.64%
Conjunto com 2 classes	80.14%

Tabela 9 – Aplicação do NaïveBayes para os diferentes conjuntos de classe

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	61.12%
Conjunto com 2 classes	85.62%

Tabela 10 – Aplicação exaustiva do J48

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	60,64	80,14
3	55,38	78,33
4	58,55	77,67
5	57,14	77,5
6	56,31	77,02
7	53,45	76,52
8	51,88	75,88
9	51,07	75,24
10	50,6	74,74
11	49,29	74,74
12	48,64	74,74
13	48,81	74,74
14	48,14	74,74
15	48,48	74,6
16	48,48	74,6

Tabela 11 – Aplicação exaustiva do NaïveBayes

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	60,02	80,81
3	57,19	80,95
4	55,79	82,38
5	57,88	82,83
6	59,52	85,62
7	58,81	84,69
8	61,12	85,21
9	58,5	82,55
10	58,52	82,55
11	59,21	82,21
12	58,71	82,55
13	59,02	83,55
14	59,48	83,88
15	58,69	84,17
16	58,57	83,52

Capítulo 5

Resultados Individuais por Empresa

Na pesquisa realizada com as empresas, 3 delas atingiram o número mínimo de respostas que permitem uma análise individual da empresa (10 avaliações de desenvolvedores).Serão apresentados nesse capítulo os resultados obtidos ao analisar individualmente essas empresas. Para fins de privacidade, elas não serão identificadas. No decorrer do estudo, elas serão referenciadas como "Empresa A", "Empresa B" e "Empresa C", como apresentado na 4.

5.1 Análise das Empresas

Para cada uma das 3 empresas, será feita a análise utilizando os dois conjuntos de classes, apontando as diferenças nos resultados. Será mostrado então a distribuição dos desenvolvedores ao longo dos dois conjuntos de classes, a seleção de características executadas no conjunto de dados resultantes da pesquisa respondida pelos supervisores das respectivas empresas e o resultado da aplicação dos algoritmos de classificação (dados de diferentes times, porém pertencentes à mesma empresa foram mesclados para uma visão geral da empresa).

Assim como foi feito com o conjunto de dados que representava todas as empresas, aqui foram aplicados os algoritmos de classificação J48 e NaïveBayes nos dados fornecidos pelas respectivas empresas, utilizando tanto o conjunto original de 5 classes quanto o conjunto de 2 classes.

Como explicado na Seção 3.3, para associar a seleção de características com os algoritmos de classificação da maneira correta, será utilizado o Algoritmo *AttributeSelectedClassifier*, usando o algoritmo *GainRatioAttributeEval* para selecionar os atributos mais relevantes. E da mesma maneira que foi realizado a seleção de características ao ser analisado o conjunto de dados de todas as empresas (apresentado na Seção 4.3), a quantidade de características ideal que maximiza a acurácia dos classificadores é escolhida através de um teste exaustivo, começando com apenas duas características e aumentando uma a uma até chegar nas dezesseis características apresentadas na Tabela 3 (o processo foi

iniciado com um número de características maior que 1 pois o objetivo não é encontrar correlação de alguma característica em específico com a classe, e sim encontrar um padrão na combinação de características que se correlacione com a importância dada pelo supervisor).

5.1.1 Análise da Empresa A

5.1.1.1 Pesquisa

A empresa A obteve avaliações de 2 supervisores. Como eram times com propósitos bem diferentes, foram consideradas apenas as avaliações de um dos times para representar a Empresa A, visto que o segundo não obteve avaliações suficientes para ser analisado individualmente. Logo, o time avaliado obteve 19 avaliações de desenvolvedores.

A Figura 4 e a Figura 5 mostram a distribuição dos desenvolvedores pelo conjunto de classes original e pelo conjunto de 2 classes, respectivamente. É possível notar que, ao comparar a distribuição dos desenvolvedores pelo conjunto original de 5 classes (Figura 4) desse empresa com o conjunto das empresas em geral (Figura 2), a Empresa A também apresentou um baixo uso das duas classes de importância mais baixas, porém, por ter utilizado mais a classe de importância média, obteve uma distribuição dos desenvolvedores mais balanceada no que se diz respeito ao novo conjunto de duas classes.

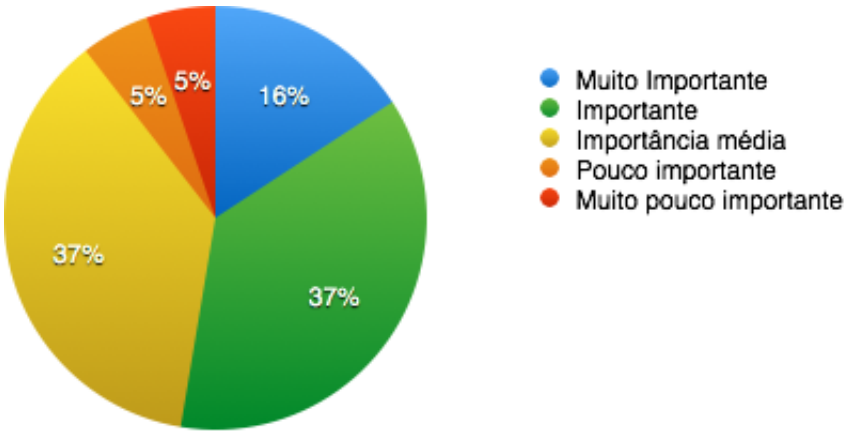

Figura 4 – Distribuição dos desenvolvedores pelo conjunto original de 5 classes (Empresa A)

5.1.1.2 Seleção de Características

O resultado da aplicação do algoritmo de seleção de características para o conjunto original de 5 classes e para o conjunto de 2 classes é apresentado na Tabela 12 e na Tabela 13 respectivamente.

Ao comparar o resultado da aplicação do algoritmos para ambos os conjuntos de classe, assim como aconteceu para os dados de todas as empresas, não houve grande variação entre os fatores melhores posicionados. Em ambas as tabelas, os atributos mais

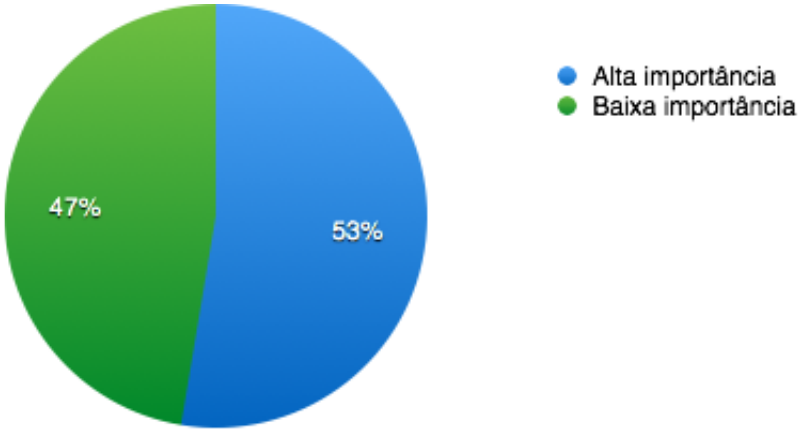

Figura 5 – Distribuição dos desenvolvedores pelo conjunto de 2 classes (Empresa A)

relevantes tendem mais para o lado das características individuais e para o compromisso do desenvolvedor para com a empresa, apesar de um fator técnico, de capacidade de resolução de problemas complexos, se fazer presente como atributo relevante nos dois cenários.

Ao fazer uma comparação entre os resultados da seleção de características da Empresa A com os dados de todas as empresas, pelo seu respectivo conjunto de classes, é possível encontrar algumas semelhanças, como o atributo mais relevante ser o mesmo tanto para o conjunto de 5 classes quanto para o de 2 classes, respectivamente, e também algumas diferenças significativas, como a distância de entre alguns fatores, como "Pró-atividade"para o conjunto de 5 classes e "Comunicação com os colegas"para o conjunto de 2 classes.

Tabela 12 – Ordenação dos atributos da Empresa A (conjunto original de 5 classes)

Atributos	**Posição média**	**Mérito médio**
Capacidade de resolução de problemas complexos	1.5 +- 0.81	0.582 +- 0.038
Foco nos resultados	2.6 +- 1.02	0.519 +- 0.047
Criatividade	3 +- 1	0.504 +- 0.035
Pró-atividade	5.1 +- 2.12	0.456 +- 0.048
Foco no cliente	6.4 +- 1.96	0.437 +- 0.04
Diversidade de habilidades	6.5 +- 1.91	0.435 +- 0.025
Experiência relevante	6.9 +- 2.39	0.427 +- 0.035
Conhecimento especializado	8.9 +- 3.05	0.396 +- 0.053
Principal comportamento do desenvolvedor	9.4 +- 2.91	0.384 +- 0.058
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	9.8 +- 2.44	0.383 +- 0.041
Comunicação com os colegas	10.5 +- 2.16	0.371 +- 0.051
Empreendedorismo	11.7 +- 1.79	0.351 +- 0.047
Disposição para ajudar colegas quando solicitado	12.1 +- 2.62	0.348 +- 0.05
Organização e planejamento	12.6 +- 2.24	0.336 +- 0.042
Tempo de trabalho (meses)	14.5 +- 4.5	0.064 +- 0.192
Liderança	14.5 +- 0.81	0.285 +- 0.063

Tabela 13 – Ordenação dos atributos da Empresa A (conjunto de 2 classes)

Atributos	**Posição média**	**Mérito médio**
Pró-atividade	1.2 +- 0.4	0.323 +- 0.021
Capacidade de resolução de problemas complexos	2.4 +- 0.8	0.298 +- 0.032
Comunicação com os colegas	4 +- 1.48	0.254 +- 0.024
Foco nos resultados	5 +- 1.48	0.23 +- 0.029
Criatividade	6.7 +- 1.68	0.206 +- 0.025
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	7.5 +- 2.01	0.187 +- 0.026
Organização e planejamento	7.8 +- 3.22	0.191 +- 0.035
Conhecimento especializado	8.2 +- 3.28	0.189 +- 0.056
Experiência relevante	9.2 +- 3.12	0.173 +- 0.04
Empreendedorismo	10.2 +- 3.22	0.171 +- 0.037
Principal comportamento do desenvolvedor	10.4 +- 3.8	0.172 +- 0.052
Disposição para ajudar colegas quando solicitado	11.1 +- 4.35	0.156 +- 0.049
Foco no cliente	11.2 +- 1.72	0.158 +- 0.02
Liderança	13 +- 2.53	0.137 +- 0.039
Diversidade de habilidades	13.4 +- 3.1	0.136 +- 0.046
Tempo de trabalho (meses)	14.7 +- 1.68	0.123 +- 0.022

5.1.1.3 Classificação

A Tabela 14 e a Tabela 15 apresentam os melhores resultados obtidos através da aplicação dos algoritmos J48 e NaïveBayes nos dados da Empresa A, respectivamente. Os algoritmos foram aplicados em ambos conjunto original de 5 classes e no conjunto de 2 classes.

A Tabela 16 e a Tabela 17 mostram a aplicação exaustiva do algoritmo que associa a seleção de características com os algoritmos de classificação, para o J48 e para o NaïveBayes respectivamente.

Observando o teste exaustivo do J48, é possível notar que ele obteve uma melhor performance utilizando um pequeno número de características ao realizar a classificação. No caso do conjunto original de 5 classes, os melhores resultados foram obtidos selecionando de 2 a 4 características, e ao ser aplicado utilizando o conjunto de 2 classes, a melhor acurácia foi obtida selecionando apenas 2 características. É importante ressaltar também o aumento de aproximadamente 20% na acurácia do classificador utilizando a segunda classe de dados.

Diferentemente do J48, o NaïveBayes obteve uma melhor performance ao selecionar um maior número de características, de 13 a 15 no conjunto original de 5 classes e acima de 7 no conjunto de 2 classes. O NaïveBayes também teve uma melhoria de performance considerável ao utilizar o novo conjunto de dados (11%).

Ambos os algoritmos, na sua melhor configuração de quantidade de características selecionadas para a classificação, e utilizando o conjunto de 2 classes que proporcionou um aumenta em suas respectivas acurácias, produziram uma porcentagem de acertos de 79,5%. Apesar de ser uma porcentagem relevante, essa acurácia é menor que a acurácia obtida ao ser analisado o conjunto com os dados de todas as empresas

Tabela 14 – Aplicação do J48 para os diferentes conjuntos de classe da Empresa A

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	59%
Conjunto com 2 classes	79.50%

Tabela 15 – Aplicação do NaïveBayes para os diferentes conjuntos de classe da Empresa A

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	68.50%
Conjunto com 2 classes	79.50%

Tabela 16 – Aplicação exaustiva do J48 para a empresa A

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	57	79,5
3	59	76
4	59	75,5
5	59	75,5
6	57	75,5
7	57	75,5
8	57	75,5
9	57	75,5
10	56,5	75,5
11	56,5	74,5
12	56,5	74,5
13	56,5	74,5
14	56,5	74,5
15	56,5	74,5
16	56,5	74,5

Tabela 17 – Aplicação exaustiva do NaïveBayes para a empresa A

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	58	78
3	59,5	73
4	55,5	68
5	58	71,5
6	57	75,5
7	58	78
8	59	79,5
9	58	79,5
10	64	79,5
11	62,5	79,5
12	67	79,5
13	68,5	79,5
14	68,5	79,5
15	68,5	79,5
16	55,5	79,5

5.1.2 Análise da Empresa B

5.1.2.1 Pesquisa

A Empresa B obteve 10 avaliações de desenvolvedores feitas por um único supervisor. A Figura 6 e a Figura 7 mostram a distribuição dos desenvolvedores pelo conjunto de classes original e pelo conjunto de 2 classes, respectivamente.

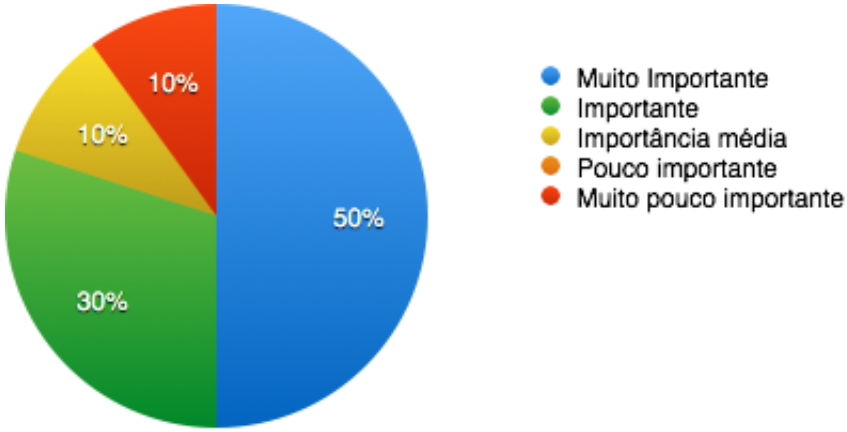

Figura 6 – Distribuição dos desenvolvedores pelo conjunto original de 5 classes (Empresa B)

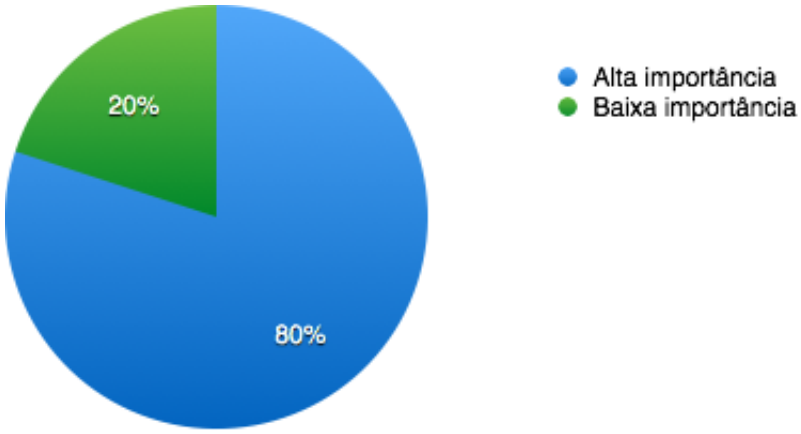

Figura 7 – Distribuição dos desenvolvedores pelo conjunto de 2 classes (Empresa B)

5.1.2.2 Seleção de Características

O resultado da aplicação do algoritmo de seleção de características para o conjunto original de 5 classes e para o conjunto de 2 classes é apresentado na Tabela 18 e na Tabela 19 respectivamente.

No caso da Empresa B, as características variam significativamente quando há a troca das classes sendo analisadas, e também diferem do resultado do conjunto de dados de todas as empresas.

Tabela 18 – Ordenação dos atributos da Empresa B (conjunto original de 5 classes)

Atributos	**Posição média**	**Mérito médio**
Capacidade de resolução de problemas complexos	1.3 +- 0.46	0.772 +- 0.078
Liderança	2.4 +- 1.02	0.652 +- 0.118
Tempo de trabalho (meses)	4.6 +- 1.43	0.562 +- 0.063
Foco nos resultados	5 +- 3.63	0.611 +- 0.179
Experiência relevante	5.3 +- 3.29	0.611 +- 0.179
Organização e planejamento	6.4 +- 1.8	0.506 +- 0.073
Conhecimento especializado	6.7 +- 2.1	0.515 +- 0.094
Diversidade de habilidades	8 +- 1.48	0.456 +- 0.06
Criatividade	9.7 +- 2.79	0.413 +- 0.119
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	11.1 +- 2.43	0.361 +- 0.076
Pró-atividade	11.2 +- 1.54	0.35 +- 0.04
Foco no cliente	11.2 +- 4.04	0.373 +- 0.133
Empreendedorismo	11.4 +- 1.74	0.36 +- 0.079
Disposição para ajudar colegas quando solicitado	12.4 +- 1.85	0.339 +- 0.08
Comunicação com os colegas	13.6 +- 2.15	0.318 +- 0.087
Principal comportamento do desenvolvedor	15.7 +- 0.64	0.235 +- 0.048

Tabela 19 – Ordenação dos atributos da Empresa B (conjunto de 2 classes)

Atributos	**Posição média**	**Mérito médio**
Foco nos resultados	2.9 +- 4.41	0.327 +- 0.131
Comunicação com os colegas	3.3 +- 1.42	0.244 +- 0.067
Experiência relevante	3.4 +- 4.25	0.327 +- 0.131
Criatividade	4.1 +- 2.07	0.26 +- 0.069
Capacidade de resolução de problemas complexos	5.6 +- 1.56	0.151 +- 0.052
Pró-atividade	8.1 +- 3.14	0.112 +- 0.028
Principal comportamento do desenvolvedor	8.1 +- 3.18	0.112 +- 0.028
Tempo de trabalho (meses)	8.6 +- 2.29	0.109 +- 0.033
Diversidade de habilidades	8.6 +- 3.5	0.119 +- 0.06
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	9.4 +- 1.74	0.093 +- 0.022
Empreendedorismo	9.6 +- 1.85	0.093 +- 0.022
Disposição para ajudar colegas quando solicitado	11.3 +- 1.42	0.063 +- 0.02
Conhecimento especializado	12.7 +- 1.42	0.043 +- 0.026
Organização e planejamento	12.9 +- 2.84	0.031 +- 0.043
Foco no cliente	13.3 +- 3.23	0.031 +- 0.043
Liderança	14.1 +- 3.67	0.023 +- 0.04

5.1.2.3 Classificação

A Tabela 20 e a Tabela 21 apresentam os melhores resultados obtidos através da aplicação dos algoritmos J48 e NaïveBayes nos dados da empresa B, respectivamente. Os algoritmos foram aplicados em ambos conjunto original de 5 classes e no conjunto de 2 classes.

A Tabela 22 e a Tabela 23 mostram a aplicação exaustiva do algoritmo que associa a seleção de características com os algoritmos de classificação, para o J48 e para o NaïveBayes respectivamente.

Observando o teste exaustivo do J48, é possível notar que, para o conjunto de classes original, o algoritmo obteve uma melhor performance utilizando um menor número de características (de 2 a 6). Já ao utilizar o conjunto de 2 classes, o classificador obteve a mesma acurácia independentemente do número de características utilizado. Utilizando o segundo conjunto de classes, o J48 teve um aumento de 10% em sua acurácia final.

No caso da Empresa B, o algoritmo NaïveBayes obteve uma performance extremamente semelhante ao J48, mantendo a performance praticamente constante de acordo com a variação das características selecionadas, obtendo uma melhoria de 10% ao utilizar o conjunto de 2 classes, e obtendo uma acurácia final de 80%. Assim como a Empresa A, a acurácia obtida pelos classificadores da Empresa B é menor que a acurácia obtida para o conjunto de dados de todas as empresas.

Tabela 20 – Aplicação do J48 para os diferentes conjuntos de classe da Empresa B

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	70%
Conjunto com 2 classes	80%

Tabela 21 – Aplicação do NaïveBayes para os diferentes conjuntos de classe da Empresa B

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	70%
Conjunto com 2 classes	80%

Tabela 22 – Aplicação exaustiva do J48 para a empresa B

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	70	80
3	70	80
4	70	80
5	70	80
6	70	80
7	70	80
8	60	80
9	60	80
10	60	80
11	60	80
12	60	80
13	60	80
14	60	80
15	60	80
16	60	80

Tabela 23 – Aplicação exaustiva do NaïveBayes para a empresa B

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	70	70
3	70	80
4	70	80
5	70	80
6	70	80
7	70	70
8	70	80
9	70	80
10	70	80
11	70	80
12	70	80
13	70	80
14	70	80
15	70	80
16	70	80

5.1.3 Análise da Empresa C

5.1.3.1 Pesquisa

A Empresa C obteve 18 avaliaçãos de desenvolvedores, dadas por 2 diferentes supervisores. Apesar de serem times diferentes, ambos fazem parte da mesma indústria de software, por isso decidiu-se mesclar os dados e fazer uma avaliação conjunta. A Empresa C também obteve mais 2 avaliações dadas por um terceiro supervisor, mas os dados foram excluídos dessa análise pelo propósito desse time ser bastante diferente dos dois primeiros. A Figura 8 e a Figura 9 mostram a distribuição dos desenvolvedores pelo conjunto de classes original e pelo conjunto de 2 classes, respectivamente.

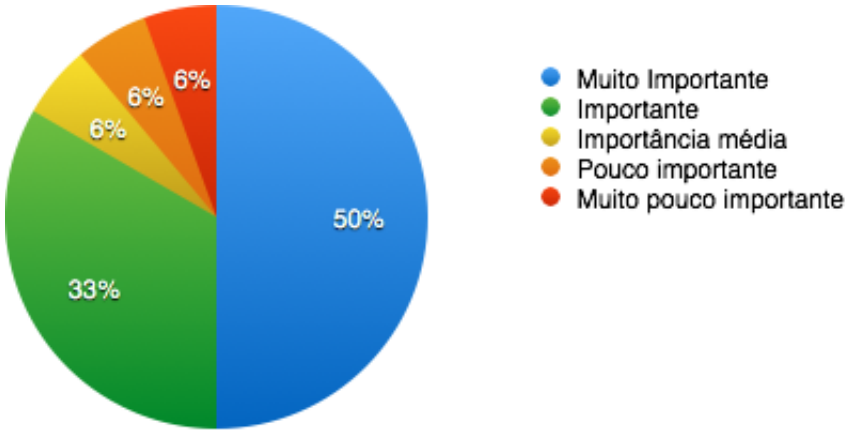

Figura 8 – Distribuição dos desenvolvedores pelo conjunto original de 5 classes (Empresa C)

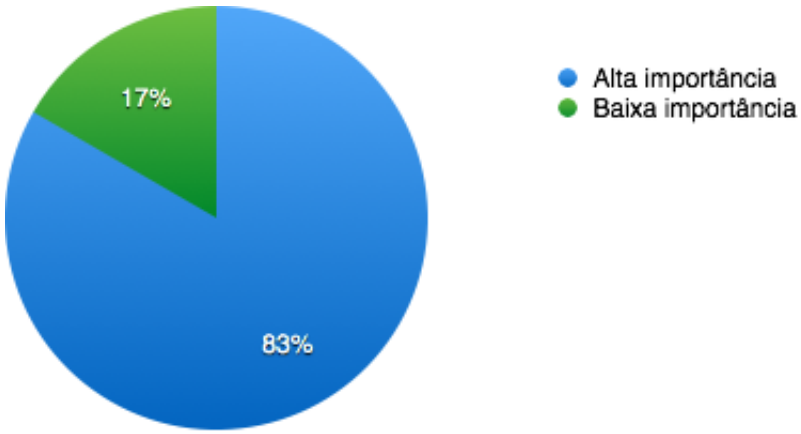

Figura 9 – Distribuição dos desenvolvedores pelo conjunto de 2 classes (Empresa C)

5.1.3.2 Seleção de Características

O resultado da aplicação do algoritmo de seleção de características para o conjunto original de 5 classes e para o conjunto de 2 classes é apresentado na Tabela 24 e na Tabela 25 respectivamente.

No caso de Empresa C, ao analisar os atributos mais relevantes em ambas as tabelas nota-se uma nítida tendência pelos atributos pertencentes ao grupos de características técnicas, inclusive, ao comparar os dados dessas tabelas com os das tabelas que representam o resultado da seleção de características para o conjunto de dados de todas as empresas, a diferença mais notável entre os atributos relevantes é a ausência do atributo "Pró-atividade"nos resultados da Empresa C, que aparece como um dos mais relevantes no resultado geral.

Tabela 24 – Ordenação dos atributos da Empresa C (conjunto original de 5 classes)

Atributos	**Posição média**	**Mérito médio**
Capacidade de resolução de problemas complexos	1.1 +- 0.3	0.643 +- 0.044
Experiência relevante	2.2 +- 0.4	0.581 +- 0.037
Conhecimento especializado	3.3 +- 0.9	0.518 +- 0.058
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	3.4 +- 0.66	0.504 +- 0.036
Diversidade de habilidades	5.2 +- 0.6	0.386 +- 0.046
Principal comportamento do desenvolvedor	6.4 +- 0.8	0.338 +- 0.048
Comunicação com os colegas	8.1 +- 1.45	0.301 +- 0.029
Foco nos resultados	8.5 +- 2.77	0.3 +- 0.08
Empreendedorismo	9.6 +- 2.06	0.275 +- 0.037
Pró-atividade	10.7 +- 2	0.267 +- 0.058
Tempo de trabalho (meses)	11.2 +- 1.66	0.261 +- 0.035
Foco no cliente	11.3 +- 1.68	0.255 +- 0.045
Criatividade	11.9 +- 2.55	0.246 +- 0.056
Liderança	13.6 +- 2.01	0.21 +- 0.055
Organização e planejamento	14.2 +- 1.54	0.196 +- 0.042
Disposição para ajudar colegas quando solicitado	15.3 +- 0.78	0.186 +- 0.044

Tabela 25 – Ordenação dos atributos da Empresa C (conjunto de 2 classes)

Atributos	**Posição média**	**Mérito médio**
Conhecimento especializado	1 +- 0	0.26 +- 0.028
Experiência relevante	2.3 +- 0.46	0.24 +- 0.026
Qual a sua avaliação sobre a produtividade do desenvolvedor em questão?	3.1 +- 0.83	0.227 +- 0.035
Capacidade de resolução de problemas complexos	4 +- 0.45	0.216 +- 0.029
Diversidade de habilidades	5.6 +- 0.66	0.179 +- 0.027
Disposição para ajudar colegas quando solicitado	6 +- 1.55	0.172 +- 0.034
Foco nos resultados	7.1 +- 1.51	0.16 +- 0.03
Pró-atividade	8.1 +- 1.97	0.147 +- 0.026
Tempo de trabalho (meses)	8.7 +- 1	0.133 +- 0.024
Organização e planejamento	10 +- 1	0.115 +- 0.025
Foco no cliente	10.8 +- 1.4	0.091 +- 0.028
Criatividade	12 +- 1.41	0.076 +- 0.015
Comunicação com os colegas	13.2 +- 0.75	0.064 +- 0.011
Liderança	14.4 +- 1.62	0.048 +- 0.015
Empreendedorismo	14.5 +- 1.02	0.048 +- 0.015
Principal comportamento do desenvolvedor	15.2 +- 1.08	0.041 +- 0.015

5.1.3.3 Classificação

A Tabela 26 e a Tabela 27 apresentam os melhores resultados obtidos através da aplicação dos algoritmos J48 e NaïveBayes nos dados da empresa C, respectivamente. Os algoritmos foram aplicados em ambos conjunto original de 5 classes e no conjunto de 2 classes.

A Tabela 28 e a Tabela 29 mostram a aplicação exaustiva do algoritmo que associa a seleção de características com os algoritmos de classificação, para o J48 e para o NaïveBayes respectivamente.

Observando o teste exaustivo do J48, é possível notar que ele obteve uma melhor performance utilizando apenas 2 características ao realizar a classificação utilizando o primeiro conjunto de classes. Já ao utilizar o conjunto de 2 classes, o classificador obteve a mesma acurácia independentemente do número de características utilizado. Utilizando o segundo conjunto de classes, o J48 teve um aumento de 8% em sua acurácia final.

Diferentemente do J48, o NaïveBayes, ao utilizar o primeiro conjunto de classes obteve melhor performance selecionando um número intermediário de características (7). Já ao utilizar o conjunto de 2 classes, o classificador atingiu sua acurácia máxima utilizando 2 ou 3 características, um número relativamente baixo. Nesse caso, o algoritmo obteve um ganho de performance de quase 15% se comparado com quando utilizou o conjunto original de 5 classes.

Tabela 26 – Aplicação do J48 para os diferentes conjuntos de classe da Empresa C

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	77%
Conjunto com 2 classes	85%

Tabela 27 – Aplicação do NaïveBayes para os diferentes conjuntos de classe da Empresa C

Classe	**Porcentagem de acertos**
Conjunto original de 5 classes	79.50%
Conjunto com 2 classes	94%

Tabela 28 – Aplicação exaustiva do J48 para a empresa C

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	77	85
3	74	85
4	74	85
5	74	85
6	74	85
7	75	85
8	75	85
9	75	85
10	75	85
11	75	85
12	75	85
13	75	85
14	75	85
15	75	85
16	75	85

Tabela 29 – Aplicação exaustiva do NaïveBayes para a empresa C

Número de características	**% de acertos - conjunto original de 5 classes**	**% de acertos - conjunto de 2 classes**
2	70,5	91,5
3	65	94
4	70,5	94
5	75,5	905
6	77,5	93,5
7	79,5	92,5
8	78	89,5
9	61,5	89,5
10	54,5	89,5
11	56	89,5
12	54,5	89,5
13	55,5	93,5
14	51,5	93,5
15	54,5	91,5
16	56	91,5

Capítulo 6

Discussão

6.1 Considerações Iniciais

O primeiro ponto a ser considerado nessa discussão é a criação do novo conjunto de classes. Como mostrado na Tabela 3, baseado no conjunto original de classes, que possui 5 diferentes classes, essas 5 classes foram agrupadas em apenas 2, criando um novo conjunto de classes que provaram melhorar a performance de todos os algoritmos de classificação aplicados, tanto para o conjunto geral de dados, que continha dados de todas as empresas, como para o conjunto gerado pelas três empresas que atingiram o número mínimo para obter uma análise individual. Como é possível observar na Tabela 4 e na Tabela 5, esse novo conjunto de classes não alterou significativamente a importância dos atributos para a classificação, assim, pode-se concluir que essa nova configuração de classes preserva o sentido da classificação original feita pelos supervisores por causa da pequena variação na posição dos atributos na ordenação.

6.2 Resultados Gerais (Todas Empresas)

Vale a pena discutir as 3 primeiras características (mais importantes), que aparecem em ambas as ordenações. são elas:

- Pró-atividade
- Capacidade de resolução de problemas complexos
- Avaliação de produtividade do desenvolvedor

Uma delas é a produtividade do desenvolvedor, sob o ponto de vista do supervisor; nesse caso, produtividade representa a quantidade de trabalho entregue. Isso não foi uma surpresa pois, como mencionado na introdução desse trabalho, esse é a métrica clássica para avaliação de performance dos desenvolvedores. Por outro lado, as outras duas características trazem novas informações relevantes à discussão.

Pró-atividade é uma característica comportamental de perfil do desenvolvedor, e ao se avaliar os dados utilizando o novo conjunto de classes, é tida como a característica que mais influencia a avaliação dos supervisores sobre os desenvolvedores.

Capacidade de resolver problemas complexos no entanto nos leva à uma direção oposta apresentada pela métrica clássica (quantidade de entregas), porque geralmente faz com que a produção possua uma menor taxa de *outputs* (LOC ou FP) sobre *inputs* (recursos, tempo) consumido.

Ambas essas características, pró-atividade e capacidade de resolver problemas, são necessárias em times envolvidos na solução de problemas complexos invés de sistemas canônicos onde as tarefas são mais previsíveis. A área de recursos humanos pode conduzir um melhor processo de contratação sabendo que os supervisores dos times de desenvolvimento de software avaliam essas características como requisitos fundamentais.

Em suma, a métrica clássica de medir a quantidade de entregas continua sendo importante na hora de avaliar a importância dos desenvolvedores, porém, para mitigar os riscos levantados na Seção 1.1, os supervisores devem levar em conta outras características, sejam elas técnicas ou do perfil do desenvolvedor. Um desenvolvedor que possui uma boa capacidade de resolução de problemas complexos, obviamente deve ser alocado para a resolução de tais problemas, e avaliar sua performance pela quantidade de LOC tende a ser um erro, principalmente se for comparar com problemas mais simples. Por outro lado, o supervisor deve estar atento aos desenvolvedores com um perfil mais dinâmico, que buscam tarefas invés de esperar, eles podem, com o acompanhamento apropriado, vir a ser os desenvolvedores que irão bater as metas de produção.

O alinhamento da avaliação de importância pelos supervisores, considerando mais que a métrica clássica, e a busca dessas características técnicas e comportamentais no perfil de candidatos, pelas áreas de contratação das empresas são práticas que podem aumentar a performance de um time / empresa como um todo.

Sobre o resultado dos classificadores, cada um deles teve um ganho em sua acurácia em torno de 20% utilizando o novo conjunto de classes. Quando avaliando todas as empresas em conjunto, o que obteve a melhor performance foi o algoritmo NaïveBayes, com uma acurácia de 85.62%, o que pode ser considerado um resultado de sucesso. O algoritmo J48 também obteve uma acurácia significativa, de 80%.

O uso desses classificadores pode ajudar os supervisores a conduzir uma análise mais coerente do perfil do seu time e de cada desenvolvedor, auxiliando na implementação da prática mencionada acima, de considerar mais do que a quantidade de entregas para avaliar a importância / performance dos desenvolvedores, e também na evolução do time, de maneira individual e coletiva, trabalhando sobre cada ponto que ainda precisa ser melhorado.

6.3 Resultados Individuais (Por Empresa)

Primeiramente, ao se falar sobre a análise individual aplicada a cada empresa, é importante mencionar a distribuição dos desenvolvedores pelas classes de importância. Ambas as empresas A, B e C, apresentaram uma semelhança em relação à distribuição dos seus desenvolvedores pelas 5 classes originais, que se dá por uma grande concentração de desenvolvedores nas classes mais altas e uma pequena nas classes mais baixas. Esse comportamento reforça a hipótese de que os supervisores possam ter ficados receosos em classificar seus desenvolvedores nas classes mais baixas. Para diminuir esse viés, foi criado esse novo conjunto de 2 classes que agrupa as 3 classes mais baixas em uma nova classe chamada "Baixa importância", e as duas classes mais altas na classe chamada "Alta importância".

Ao serem analisadas as características que mais influenciam a avaliação dos supervisores (considerando apenas a ordenação que utilizou o novo conjunto de 2 classes), por cada empresa, é possível notar algumas grandes diferenças, tanto entre as empresas quanto em relação ao conjunto geral de dados (todas as empresas). Para todas as empresas, encontra-se pelo menos uma característica técnica entre as top três características mais importantes. Nas empresas A e B, foram encontradas características comportamentais (Pró-atividade), de habilidade interpessoal (Comunicação com os colegas) e de compromisso em relação à empresa (Foco no resultado) com melhor posicionamento que a métrica clássica de produtividade (Quantidade de entregas).

Já a Empresa C preza mais por características técnicas, visto que dentre as top 5 primeiras características estão as 4 características técnicas e a métrica clássica de produtividade. É possível atribuir essas diferenças nos resultados às diferenças na cultura e nos valores de cada empresa, vários estudos já foram feitos sobre como a cultura corporativa pode interferir na produtividade dos desenvolvedores (EDMANS, 2011; JONES, 2000; SCUDDER; KUCIC, 1991; Agrell, A. and Gustafson, 1994; GUZZO, 1988; MCLEAN; SMITS; TANNER, 1996; TURCOTTE; RENNISON, 2004).

Em relação à classificação, todas as empresas obtiveram um aumento de 10% a 15% na acurácia dos classificadores ao utilizar o novo conjunto de classes de importância. A Empresa A, coincidentemente, obteve uma acurácia de 79.5% para ambos os algoritmos J48 e NaïveBayes. A Empresa B, atingiu uma acurácia também igual entre os classificadores, de 80%. Ambas as acurácias máximas das empresas A e B foram menores que a acurácia máxima atingida para o conjunto de dados de todas as empresas. Este não era o resultado, visto que, ao construir classificadores customizados por empresa, esperava-se uma acurácia maior que a geral pela eliminação de diferentes perspectivas dos diferentes supervisores.

A Empresa C por sua vez foi a que obteve a maior acurácia, utilizando o algoritmo NaïveBayes, de 94%. Esse resultado foi considerado um sucesso, pois supera em quase

10% a acurácia geral, possuindo uma taxa de erro de apenas 6%. Além disse, é importante notar que foram agrupados avaliações de 2 supervisores, diferentemente das empresas A e B onde um único supervisor foi responsável pela avaliação. Atribui-se esse alta taxa de acertos à uma cultura bem estabelecida da empresa e a critérios claros de avaliação de importância pelos supervisores da Empresa C.

6.4 Ameaças À Validade

Por fim, é importante apontar algumas ameaças à validade desse estudo. O número limitado de desenvolvedores e empresas participantes nesse estudo, e o fato de todas estarem estabelecidas na mesma cidade podem limitar a generalização dos resultados para outros contextos. A variabilidade na cultura das empresas participantes e nos critérios que cada supervisor usa para avaliar seus desenvolvedores também podem impactar o resultado geral. Apesar disso, é possível observar várias intersecções no resultado de diferentes empresas, e das empresas com o resultado geral, o que pode mitigar parte desse risco. A classificação inicial de importância proporcionado pelos supervisores tendem a serem mais positivas, talvez porque eles não gostariam de dizer que mantém desenvolvedores com baixa importância em seus times. Porém a nova classificação proposta mitiga parte desse risco.

A possível ausência de fatores para a avaliação dos supervisores também é uma ameaça, porém o levantamento de fatores baseou-se em vários estudos, que também buscavam os fatores de maior relevância na avaliação de desenvolvedores Tabela 2, e também houve uma preocupação em selecionar características distintas, não apenas técnicas ou comportamentais, diminuindo esse risco.

Capítulo 7

Conclusões

Foi identificado, como mostrado na Seção 1.1 uma situação-problema nas empresas atualmente, que é a subjetividade na avaliação dos desenvolvedores por parte dos supervisores. Esse problema leva as empresas a terem problemas maiores, de retenção e motivação dos funcionários. Este estudo se propôs então a ajudar as empresas e, se possível fornecer uma ferramenta que auxilia e empresa a enfrentar com mais eficácia esses problemas.

O estudo se baseou em duas principais perguntas. A primeira questionava quais seriam os critérios mais importantes utilizados pelos supervisores ao realizarem sua avaliação sobre os desenvolvedores. Para responder tal pergunta, primeiramente, foi preciso levantar uma série de critérios e optar por aqueles que aparentassem estar mais ligados à realidade das empresas de hoje. Esse levantamento foi realizado utilizando o framework GQM e como resultado foram obtidas 16 métricas, divididas em 4 diferentes grupos, como mostrado na Seção 2.1. Uma vez com as métricas levantadas, foram utilizados algoritmos de seleção de características que ordenaram as 16 métricas por ordem de influência na classe. Foi possível observar ainda intersecções nas métricas mais relevantes da ordenação geral (todas as empresas) com as ordenações individuais das três empresas analisadas, dando mais relevância ao resultado final.

A segunda pergunta questionava se seria possível atingir um classificador de alta acurácia, utilizando os critérios propostos, tanto para o conjunto que reuniam os dados de todas as empresas, quanto para cada empresa que atingisse o mínimo de dados necessários para uma análise individual. Como mostrado no Capítulo 4 e no Capítulo 5, essa pergunta também é considerada como respondida com êxito, visto que foi obtido um classificador com acurácia maior que 85% para o conjunto de dados de todas as empresas, e para as 3 empresas que se qualificaram para uma análise individual, os melhores classificadores obtiveram uma acurácia de cerca de 80%, sendo que uma empresa obteve um classificador com uma acurácia de 94%.

Nesse estudo foi proporcionado então um conjunto de critérios utilizados pelos supervisores de empresas de T.I, para avaliar seus desenvolvedores, além de ordenar esses

critérios pela taxa de influência na classificação de importância, provendo uma análise em quais são os fatores mais relevantes. Considerando apenas o conjunto de duas classes criado para otimizar os resultados, as 5 características que mais influenciam a avaliação de importância são:

- ❏ Pró-atividade
- ❏ Avaliação de produtividade do desenvolvedor
- ❏ Capacidade de resolução de problemas complexos
- ❏ Foco nos resultados
- ❏ Experiência relevante

Levar em consideração essas características no momento da contratação e no acompanhamento individual de cada desenvolvedor tendem formar times de grande importância dentro das organizações.

Além disso, foi criado um classificador de alta acurácia, que pode ajudar, por exemplo, os gerentes de recursos humanos a procurarem candidatos que possuam as características necessárias e que consequentemente possuam um bom potencial de se tornarem parte importante do time, e também pode auxiliar o supervisor a realizar uma avaliação muito mais concreta, saindo da área da subjetividade no momento de avaliar a importância dos seus desenvolvedores. Existe uma taxa de erro inerente ao classificador (no caso da avaliação geral, é de cerca de 15%), logo, não é recomendável utilizar apenas o classificador sozinho. Ele serve como um instrumento para agregar valor no momento da avaliação do supervisor ou na contratação de um novo funcionário, aliando com uma avaliação qualitativa do desenvolvedor seguindo as características recomendadas nesse mesmo estudo.

Foram obtidos bons resultados com esse estudo sobre a avaliação da importância dos desenvolvedores sob a ótica do supervisor. Porém, entende-se que muito mais pode ser realizado, pois este é um problema significativo e a pesquisa também possui diversas áreas ainda a serem exploradas. Segue uma lista de algumas das possibilidades que para trabalhos futuros:

1. Expandir a aplicação da pesquisa, aplicando em mais empresas situadas em diferentes regiões do país e até do mundo, bem como expandir as características utilizadas;
2. Realizar uma análise qualitativa, em cima dos dados de cada empresa, considerando como a cultura e os valores da empresa influenciam na avaliação de cada critério;
3. Aplicar esse classificador em colaboradores de repositórios de software *open-source*. Vasilescu et al. (VASILESCU; FILKOV; SEREBRENIK, 2013) realizou um estudo buscando encontrar relação entre a busca de conhecimento em sites de perguntas

e respostas e a produtividade do desenvolvedor, porém utilizou a métrica clássica de produtividade, nesse caso, número de *commits*. Utilizar as métricas levantadas neste estudo como base para validar os resultados ou mesmo detectar e avaliar as diferenças, pode ser uma sugestão de trabalho futuro para ambos os estudos.

Por se tratar de um problema importante para todas as empresas atualmente, entende-se que, além das possibilidades propostas, existem mais inúmeras possibilidades de trabalhos futuros que poderiam usar este estudo como base.

Referências

ABDEL-HAMID, T.; MADNICK, S. E. **Software Project Dynamics: An Integrated Approach**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991. ISBN 0-13-822040-9.

Agrell, A. and Gustafson, R. The Team Climate Inventory (TCI) and group innovation: A psychometric test on a Swedish sample of work groups. **Journal of Occupational and Organizational Psychology**, v. 67, p. 143–151, 1994.

ALPER, S.; TJOSVOLD, D.; LAW, K. S. Conflict Management, Efficacy, and Performance in Organizational Teams. **Personnel Psychology**, v. 53, p. 625–642, 2000. ISSN 0031-5826. Disponível em: <http://doi.wiley.com/10.1111/j.1744-6570.2000.tb00216.x>.

BANKER, R. D.; DATAR, S. M.; KEMERER, C. F. Factors affecting software maintenance productivity: An exploratory study. In: **Proceedings of the International Conference on Information Systems**. Pittsburgh, PA: [s.n.], 1987. p. 160–175.

_____. **A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects**. 1991. 1–18 p.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. **Encyclopedia of Software Engineering**, Wiley, v. 2, p. 528–532, 1994. Disponível em: <http://maisqual.squoring.com/wiki/index.php/TheGoalQuestionMetricApproach>.

BOEHM, B. W. Software Engineering Economics. **IEEE Transactions on Software Engineering**, SE-10, 1984. ISSN 0098-5589.

_____. Improving Software Productivity. **Computer**, IEEE Computer Society, Los Alamitos, CA, USA, v. 20, n. 9, p. 43–57, 1987. ISSN 0018-9162.

BOEHM, B. W. et al. **Software Cost Estimation with Cocomo II with Cdrom**. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN 0130266922.

_____. The TRW Software Productivity System. In: **Proceedings of the 6th International Conference on Software Engineering**. Los Alamitos, CA, USA: IEEE Computer Society Press, 1982. (ICSE '82), p. 148–156. Disponível em: <http://dl.acm.org/citation.cfm?id=800254.807757>.

BOEHM, B. W.; PAPACCIO, P. N. Understanding and controlling software costs. **Software Engineering, IEEE Transactions on**, v. 14, n. 10, p. 1462–1477, 1988. ISSN 0098-5589.

BOYATZIS, R. E. **The competent manager : a model for effective performance**. [S.l.]: New York: Wiley, 1982.

_____. Competencies in the 21st century. **Journal of Management Development**, v. 27, n. 1, p. 5–12, 2008. Disponível em: <http://dx.doi.org/10.1108/02621710810840730>.

BRIAND, L.; EMAM, K. E.; BOMARIUS, F. COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment. **Proceedings of the 20th International Conference on Software Engineering**, 1998. ISSN 0270-5257.

Brooks Jr, F. P. No silver bullet-essence and accidents of software engineering. **IEEE computer**, v. 20, n. 4, p. 10–19, 1987.

BROOKS, W. Software Technology Payoff: Some Statistical Evidence. **J. Syst. Softw.**, Elsevier Science Inc., New York, NY, USA, v. 2, n. 1, p. 3–9, 1981. ISSN 0164-1212. Disponível em: <http://dx.doi.org/10.1016/0164-1212(81)90041-8>.

CALOW, H. Maintenance productivity factors-a case study. In: **Software Maintenance, 1991., Proceedings. Conference on**. [S.l.: s.n.], 1991. p. 250–253.

CHATZOGLOU, P. D.; MACAULAY, L. A. **The importance of human factors in planning the requirements capture stage of a project**. 1997. 39–53 p.

CLINCY, V. A. Software Development Productivity and Cycle Time Reduction. **J. Comput. Sci. Coll.**, Consortium for Computing Sciences in Colleges, USA, v. 19, n. 2, p. 278–287, 2003. ISSN 1937-4771. Disponível em: <http://dl.acm.org/citation.cfm?id=948785.948824>.

COLE, A. Runaway Projects - Cause and Effects. **Software World (UK)**, v. 26, n. 3, 1995.

CORAM, M.; BOHNER, S. The impact of agile methods on software project management. In: **Engineering of Computer-Based Systems, 2005. ECBS '05. 12th IEEE International Conference and Workshops on the**. [S.l.: s.n.], 2005. p. 363–370.

de Barros Sampaio, S. C. et al. A Review of Productivity Factors and Strategies on Software Development. In: **Software Engineering Advances (ICSEA), 2010 Fifth International Conference on**. [S.l.: s.n.], 2010. p. 196–204.

DEMARCO, T.; TIMOTHY, L. **peopleware - productive projects and teams**. [S.l.: s.n.], 1987.

DUTRA, J. S. **Competências: Conceitos e Instrumentos para a Gestão de Pessoas na Empresa Moderna**. 1ª. ed. [S.l.: s.n.], 2004.

EDMANS, A. Does the stock market fully value intangibles? Employee satisfaction and equity prices. **Journal of Financial Economics**, v. 101, p. 621–640, 2011. ISSN 0304405X.

Faria Sueli, O. V. F. d. F. L. . D. F. Competências do profissional da informação: uma reflexão a partir da Classificação Brasileira de Ocupações. **Ciência da Informação**, scielo, v. 34, p. 26–33, 2005. ISSN 0100-1965. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652005000200003&nrm=iso>.

FINNIE, G. R.; WITTIG, G. E.; PETKOV, D. I. **Prioritizing software development productivity factors using the analytic hierarchy process**. 1993. 129–139 p.

FLEURY, M. T. L.; FLEURY, A. **Construindo o conceito de competência**. 2001. 183–196 p.

GUZZO, R. A. Productivity research: Reviewing psychological and economic perspectives. In: . [S.l.]: Jossey-Bass Publishers, 1988. p. 63–81.

HANSON, S. J.; ROSINSKI, R. R. Programmer Perceptions of Productivity and Programming Tools. **Commun. ACM**, ACM, New York, NY, USA, v. 28, n. 2, p. 180–189, 1985. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2786.2791>.

HANTOS, P.; GISBERT, M. Identifying software productivity improvement approaches and risks: Construction industry case study. **IEEE Software**, v. 17, p. 48–56, 2000. ISSN 07407459.

HOLMES, G.; DONKIN, A.; WITTEN, I. H. W EKA : A Machine Learning Workbench.

JONES, C. **Programming Productivity: Steps Toward a Science. McGraw-Hill Series in Software Engineering and Technology.** [S.l.]: McGraw-Hill, 1986.

_____. **Applied software measurement: assuring productivity and quality**. [s.n.], 1997. ISBN 0070328137. Disponível em: <http://books.google.ch/books/about/Applied_software_measurement.html?id=U9JQAAAAMAAJ&redir_esc=y>.

_____. **Software Assessments, Benchmarks, and Best Practices**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000. ISBN 0-201-48542-7.

KIRA, K.; RENDELL, L. A. The Feature Selection Problem: Traditional Methods and a New Algorithm. In: **Proceedings of the Tenth National Conference on Artificial Intelligence**. AAAI Press, 1992. (AAAI'92), p. 129–134. ISBN 0-262-51063-4. Disponível em: <http://dl.acm.org/citation.cfm?id=1867135.1867155>.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artificial Intelligence**, v. 97, n. 1-2, p. 273–324, dez. 1997. ISSN 00043702. Disponível em: <http://www.sciencedirect.com/science/article/pii/S000437029700043X>.

LAKHANPAL, B. **Understanding the factors influencing the performance of software development groups: An exploratory group-level analysis**. 1993. 468–473 p.

LALSING, V.; KISHNAH, S.; PUDARUTH, S. People Factors in Agile Software Development and Project Management. **International Journal of Software Engineering & Applications**, v. 3, p. 117–138, 2012. Disponível em: <http://airccse.org/journal/ijsea/papers/3112ijsea09.pdf>.

LOKAN, C. J. Impact of Subjective Factors on Software Productivity. In: **Proceedings of 7th Australian Conference on Software Metrics**. Melbourne: [s.n.], 2001.

MAXWELL, K. D.; FORSELIUS, P. Benchmarking software development productivity. **Software, IEEE**, v. 17, p. 80–88, 2000. ISSN 0740-7459.

MCLEAN, E. R.; SMITS, S. J.; TANNER, J. R. The importance of salary on job and career attitudes of information systems professionals. **Information and Management**, v. 30, p. 291–299, 1996. ISSN 03787206.

MELO, C. et al. Agile Team Perceptions of Productivity Factors. In: **Agile Conference (AGILE), 2011**. [S.l.: s.n.], 2011. p. 57–66.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

RASCH, R. H. An Investigation of Factors That Impact Behavioral Outcomes of Software Engineers. In: **Proceedings of the 1991 Conference on SIGCPR**. New York, NY, USA: ACM, 1991. (SIGCPR '91), p. 38–53. ISBN 0-89791-389-2. Disponível em: <http://doi.acm.org/10.1145/111084.111090>.

SCACCHI, W.; HURLEY, D. Understanding software productivity. In: **Software Engineering and Knowledge Engineering: Trends for the Next Decade**. [s.n.], 1995. v. 4, p. 273–316. Disponível em: <http://books.google.com/books?hl=en&lr=&id=bDlZa4KvFGgC&oi=fnd&pg=PA273&dq=Understanding+software+productivity&ots=TFYqI5HTtY&sig=QBz5suT4_1Fwjkp-NwO0ZMeGEIY>.

SCHWABER, K. **Agile Project Management With Scrum**. Redmond, WA, USA: Microsoft Press, 2004. ISBN 073561993X.

SCUDDER, R. A.; KUCIC, A. R. Productivity Measures for Information Systems. **Inf. Manage.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 20, n. 5, p. 343–354, 1991. ISSN 0378-7206. Disponível em: <http://dx.doi.org/10.1016/0378-7206(91)90033-X>.

SHARP, H. et al. Models of Motivation in Software Engineering. **Inf. Softw. Technol.**, Butterworth-Heinemann, Newton, MA, USA, v. 51, n. 1, p. 219–233, 2009. ISSN 0950-5849. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2008.05.009>.

SHARPE, J. L.; CANGUSSU, J. W. A productivity metric based on statistical pattern recognition. In: **Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International**. [S.l.: s.n.], 2005. v. 1, p. 59–64 Vol. 2. ISSN 0730-3157.

SHIRAZI, A.; MORTAZAVI, S. Effective management performance a competency-based perspective. **International Review of Business Research Papers**, v. 5, n. 1, p. 1–10, January 2009. ISSN 1832-9543.

SYMONS, C. Software Industry Performance: What You Measure Is What You Get. **Software, IEEE**, v. 27, n. 6, p. 66–72, 2010. ISSN 0740-7459.

TURCOTTE, J.; RENNISON, L. W. The Link between Technology Use, Human Capital, Productivity and Wages: Firm-level Evidence. **International Productivity Monitor**, v. 9, p. 25–36, 2004. Disponível em: <http://ideas.repec.org/a/sls/ipmsls/v9y20043.html>.

VASILESCU, B.; FILKOV, V.; SEREBRENIK, A. StackOverflow and GitHub: associations between software development and crowdsourced knowledge. In: **Proceedings of the 2013 ASE/IEEE International Conference on Social Computing. IEEE**. [S.l.: s.n.], 2013. p. 188–195.

VOSBURGH, J. et al. Productivity factors and programming environments. In: **ICSE '84 Proceedings of the 7th International Conference on Software Engineering**. [s.n.], 1984. p. 143–152. ISBN 0-8186-0528-6. Disponível em: <http://dl.acm.org/citation.cfm?id=800054.801963>.

WAGNER, S.; RUHE, M. **A Structured Review of Productivity Factors in Software Development Technical** . [S.l.], 2008.

WALLACE, L.; KEIL, M.; RAI, A. Understanding Software Project Risk: A Cluster Analysis. **Inf. Manage.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 42, n. 1, p. 115–125, 2004. ISSN 0378-7206. Disponível em: <http://dx.doi.org/10.1016/j.im.2003.12.007>.

WALSTON, C. E.; FELIX, C. P. **A method of programming measurement and estimation**. 1977. 54–73 p.

WOHLIN, C.; AHLGREN, M. Soft factors and their impact on time to market. **Software Quality Journal**, v. 4, p. 189–205, 1995. ISSN 09639314.

WOHLIN, C.; ANDREWS, A. A. Assessing Project Success Using Subjective Evaluation Factors. **Software Quality Journal**, v. 9, p. 43–70, 2001. ISSN 09639314.

YU, W. D.; SMITH, D. P.; HUANG, S. T. Software productivity measurements. In: **Computer Software and Applications Conference, 1991. COMPSAC '91., Proceedings of the Fifteenth Annual International**. [S.l.: s.n.], 1991. p. 558–564.