

UNIVERSIDADE FEDERAL DO PARANÁ

ELIAS DELGOBO JUNIOR

USO DE MODELOS DE QUALIDADE DE SOFTWARE EM EMPRESAS DE  
DESENVOLVIMENTO DE SOFTWARE EM CURITIBA

CURITIBA

2015

ELIAS DELGOBO JUNIOR

USO DE MODELOS DE QUALIDADE DE SOFTWARE EM EMPRESAS DE  
DESENVOLVIMENTO DE SOFTWARE EM CURITIBA

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência, Gestão e Tecnologia da Informação, no Curso de Pós-graduação em Ciência, Gestão e Tecnologia da Informação, Setor de Sociais Aplicadas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Egon Walter Wildauer

CURITIBA

2015

UNIVERSIDADE FEDERAL DO PARANÁ. SISTEMA DE BIBLIOTECAS.  
CATALOGAÇÃO NA FONTE

Delgobo Junior, Elias

Uso de modelos de qualidade de software em empresas de desenvolvimento de software em Curitiba / Elias Delgobo Junior. - 2015. 96 f.

Orientador: Egon, Walter Wildauer..

Dissertação (Mestrado) - Universidade Federal do Paraná. Programa de Pós-Graduação em Ciência, Gestão e Tecnologia da Informação, do Setor de Ciências Sociais Aplicadas.

Defesa: Curitiba, 2015

1. Indústria de software - Curitiba (PR). 2. Indústria de software - Controle de qualidade. I. Wildauer, Egon Walter. II. Universidade Federal do Paraná. Setor de Ciências Sociais Aplicadas. - Programa de Pós-Graduação em Ciência, Gestão e Tecnologia da Informação. III. Título.

CDD 658.562

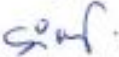
**TERMO DE APROVAÇÃO**

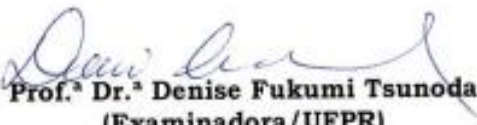
**Elias Delgobo Junior**


**“ADERÊNCIA DAS EMPRESAS DESENVOLVEDORAS DE SOFTWARE EM  
CURITIBA AOS MODELOS DE QUALIDADE”**

**DISSERTAÇÃO APROVADA COMO REQUISITO PARCIAL PARA  
OBTENÇÃO DO GRAU DE MESTRE NO PROGRAMA DE PÓS-  
GRADUAÇÃO EM CIÊNCIA, GESTÃO E TECNOLOGIA DA INFORMAÇÃO  
DA UNIVERSIDADE FEDERAL DO PARANÁ, PELA SEGUINTE BANCA  
EXAMINADORA:**

  
**Prof. Dr. Egon Walter Wildauer**  
(Orientador/UFPR)

  
**Prof. Dr. José Simão de Paula Pinto**  
(Examinador/UFPR)

  
**Prof.ª Dr.ª Denise Fukumi Tsunoda**  
(Examinadora/UFPR)

  
**Prof. Dr. Marcos Sfair Sunyé**  
(Examinador/UFPR)

**20 de fevereiro**

## RESUMO

A compreensão da qualidade como um atributo do software levou à importância da inclusão de modelos de qualidade de software no Brasil. A partir da década de 90 com as avaliações CMM e na década seguinte com as avaliações e MPS.BR o mercado brasileiro começou a implantar os modelos de qualidade que melhorassem os processos de software nas empresas de desenvolvimento de sistemas computacionais. Em outro aspecto, os modelos de qualidade de software para avaliação como o modelo SQUARE verificam por meio quantitativo se o produto tem qualidade e também interferem na satisfação do cliente. Foram selecionados três modelos referentes à qualidade do produto que refletem a evolução da avaliação do software (McCall, FURPS e SQUARE). A utilização dos modelos de avaliação de software é útil para garantir que o software produzido obtenha garantia da qualidade. A pesquisa também identificou três modelos de melhoria de processo de software (CMMI, SPICE e MPS.Br). Durante o processo de pesquisa se desenvolveu um instrumento de coleta de dados com o objetivo de identificar o uso de modelos de avaliação e de melhoria de processos em desenvolvedoras de software em Curitiba. Os resultados obtidos por meio do questionário foram formatados e analisados, das 75 empresas respondentes apenas 26 delas utilizam modelos de qualidade de software. As empresas que utilizam modelos de maturidade (independente do modelo) avaliaram o impacto do uso desses modelos como grau 5 (em um intervalo de 1-5) em seis critérios (custo, tempo de produção, desempenho, retrabalho, satisfação do cliente e manutenção do software), porém, na maior parte das empresa que utiliza modelos de qualidade utiliza modelos desenvolvidos na própria empresa com base em experiências vividas na empresa. O impacto do uso dos modelos é positivo no primeiro momento o processo de melhoria se torna caro e dispendioso, com gastos com treinamento e na mudança cultural da empresa, em um segundo estágio, depois da consolidação do uso dos modelos, a empresa passa a ter economia, com menos erros e gastos desnecessários. Por meio dos resultados desta pesquisa pode se concluir que existem modelos compatíveis com as necessidades das empresas desenvolvedoras, mas em função do custo as empresas adotam modelos próprios de maturidade e de avaliação de software.

Palavras-chave: Modelos de Qualidade de software. Informação. Qualidade de software. Curitiba (PR)

## ABSTRACT

The understanding of quality as a software attribute led to the importance of including software quality models in Brazil. From the 90s to the CMM reviews and a decade later with the reviews and MPS.BR the Brazilian market began to implement quality models that would improve software processes in the development of computer systems companies. In another aspect, the software quality models for evaluation as the model SQUARE check by quantitative means the product has quality and also interfere in customer satisfaction. Three models related to product quality that reflect the evolution of software evaluation (McCall, FURPS and SQUARE) were selected. The use of software evaluation models is useful to ensure that the software produced obtain quality assurance. The survey also identified three models of improvement of software process (CMMI, SPICE and MPS.Br). During the research process developed a data collection instrument in order to identify the use of valuation models and process improvement in software developers in Curitiba. The results obtained from the questionnaire were analyzed and formatted, the 75 respondent companies only use 26 software quality models. Companies using maturity models (regardless of model) evaluated the impact of the use of these models as grade 5 (in a range of 1-5) on six criteria (cost, production time, performance, rework, customer satisfaction and maintenance software), however, most of the company using quality models use models developed in-house based on experiences in the company. The impact of the use of models is positive at first the improvement process becomes expensive and costly, with spending on training and cultural change of the company, in a second stage, after the consolidation of the use of models, the company will have economy, with fewer errors and unnecessary spending. Through the results of this research can be concluded that there are models compatible with the needs of developing companies, but due to the cost companies adopt own maturity models and assessment software.

Keywords: Quality Model software. Information. Quality software. Curitiba (PR)



## LISTA DE QUADROS

Quadro 1 - Classificação das atividades econômicas segundo cnae 2.1 .....	50
Quadro 2 - Comparativo entre os modelos McCall, FURPS+ e SQUARE, dos atributos para Qualidade de software.....	57
Quadro 3 - Equivalência entre os cinco níveis do modelo cmmi e os sete níveis do modelo mps.br segundo franciscani; pestili (2012) .....	62

## LISTA DE FIGURAS

Figura 1 - Cclo PDCA.....	21
Figura 2 - ISO/IEC 15504.....	33



## LISTA DE GRÁFICOS

Gráfico 1 - Quantidade x porte das empresas que responderam o questionário.	64
Gráfico 2 - Utilização de modelos para avaliação de software em empresas de desenvolvimento em Curitiba .....	65
Gráfico 3 - Uso de modelos de qualidade de software para avaliação em microempresas de desenvolvimento de software .....	65
Gráfico 4 - Utilização de modelos de qualidade de processo de software em empresas de desenvolvimento de software .....	66
Gráfico 5 - Uso de modelos de qualidade de processo de software em microempresas de Curitiba .....	67
Gráfico 6 – Impacto dos modelos de qualidade no custo de produção de software .....	69
Gráfico 7 - Impacto dos modelos de qualidade no tempo de produção de software .....	70
Gráfico 8 - Impacto dos modelos de qualidade no desempenho do software ...	71
Gráfico 9 - Impacto dos modelos de qualidade no retrabalho da equipe de software .....	72
Gráfico 10 - Impacto dos modelos de qualidade na satisfação do usuário .....	73
Gráfico 11 - Impacto dos modelos de qualidade na manutenção do software ..	73

## LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AP	Atributos de Processo
ASQC	<i>American Society for Quality Control</i>
CBQP	Congresso Brasileiro de Qualidade e Produtividade
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
CNAE 2.1	Classificação Nacional de Atividades Econômicas
CONCLA	Comissão Nacional de CLAssificação
CWQC	<i>Company Wide Quality Control</i>
FPNQ	Fundação para o Prêmio Nacional da Qualidade
FURPS+	<i>Functionality, Usability, Reliability, Performance,</i>
<i>Supportability</i>	
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
INPM	Nacional de Pesos e Medidas
IPARDES	Instituto Paranaense de Desenvolvimento Econômico e Social
ISO	Organização Internacional para Padronização
JUSE	Associação japonesa de cientistas e engenheiros
MEG	Modelo de Excelência da Gestão
MPS.Br	Melhoria de Processo de Software Brasileiro
PDCA	<i>Plan, Do, Check, Action</i>
PEGQ	Projeto de Especialização em Gestão de Qualidade
PMBOK	Guia de Corpo de Conhecimento em Gestão de Projetos
PMMM	Modelo de Maturidade em Gestão de Projetos
PNQ	Prêmio Nacional da Qualidade
RAP	Resultados esperados dos Atributos de Processo
SEPL Geral	Secretaria de Estado do Planejamento e Coordenação
SPICE	<i>Software Process Improvement and Capability</i>
<i>dEtermination</i>	
SQUARE Software	Requisitos de Qualidade e Avaliação de Produtos de
TQC	Controle da Qualidade Total
TQM	Gestão da Qualidade Total
UFPR	Universidade Federal do Paraná

## SUMÁRIO

1	INTRODUÇÃO .....	10
1.1	PROBLEMA.....	10
1.2	OBJETIVOS .....	11
1.2.1	Objetivo Geral.....	11
1.2.2	Objetivos específicos.....	11
1.3	JUSTIFICATIVA .....	12
1.4	ESTRUTURA DO TRABALHO .....	13
2	REFERENCIAL TEÓRICO .....	14
2.1	EVOLUÇÃO DO TERMO QUALIDADE .....	14
2.2	QUALIDADE NO BRASIL.....	17
2.3	GESTÃO DA QUALIDADE E EXCELÊNCIA DA GESTÃO.....	18
2.4	QUALIDADE DE SOFTWARE.....	22
2.5	MODELOS DE QUALIDADE DE SOFTWARE .....	25
2.5.1	Modelo de qualidade de McCall.....	25
2.5.2	Modelo FURPS+.....	28
2.5.3	Modelo de qualidade SQUARE .....	29
2.5.4	Modelo de qualidade SPICE .....	32
2.5.5	Modelo de qualidade CMMI-DEV® .....	35
2.5.6	Modelo de qualidade MPS.Br-SW .....	38
2.6	QUALIDADE DA INFORMAÇÃO .....	43
2.7	INFORMAÇÕES QUE GARANTEM A QUALIDADE DE SOFTWARE..	44
3	METODOLOGIA.....	47
3.1	PROCESSO METODOLÓGICO.....	47
3.2	DELIMITAÇÃO E LIMITAÇÕES DA PESQUISA.....	50
3.3	DEFINIÇÃO DO INSTRUMENTO DE COLETA DE DADOS .....	53
3.5	TRATAMENTO DOS DADOS.....	54
3.6	ANÁLISE DE DADOS.....	56
4	RESULTADOS E ANÁLISES.....	57
4.1	RESULTADOS E ANÁLISES DO REFERENCIAL TEÓRICO.....	57

4.2	RESULTADOS E ANÁLISES DO INSTRUMENTO DE PESQUISA.....	63
5	CONSIDERAÇÕES FINAIS.....	76
	REFERÊNCIAS .....	79
	APÊNDICE A – INSTRUMENTO DE PESQUISA (QUESTIONÁRIO – PRÉ-TESTE)	
	83	
	APÊNDICE B – INSTRUMENTO DE PESQUISA (QUESTIONÁRIO) .....	89

## 1 INTRODUÇÃO

A busca pelo desenvolvimento de softwares com mais qualidade se tornou uma regra para as desenvolvedoras de software e a concorrência do mercado de software exige a melhoria contínua dos processos de produção, por essa razão foram desenvolvidos modelos de melhoria dos processos de software, esses modelos evoluíram com o passar do tempo e se tornaram referência no mundo todo. O Modelo CMMI, por exemplo, no ano de 2015 contava com 2.835 empresas certificadas nos EUA, 13.400 empresas certificadas em todo mundo, no Brasil eram 299 empresas no mesmo ano.

Nesse panorama os modelos de qualidade são utilizados não apenas para melhorar os processos de software, mas para certificar a empresa frente aos clientes e garantir que o software será produzido com base em processos definidos e documentados e que a empresa, com o tempo, se tornará melhor. A empresa também pode utilizar modelos de qualidade que possam avaliar o software como produto e buscar métricas que relacionem a qualidade do software à satisfação dos clientes e usuários.

### 1.1 PROBLEMA

No setor de desenvolvimento de software a entrega de um produto de qualidade é objetivo da maioria das empresas e se supõe que a qualidade do produto esta relacionada à qualidade dos processos de desenvolvimento desse produto. Na literatura se encontram modelos de qualidade de software que avaliam o produto de software e modelos de qualidade de software para melhoria da qualidade dos processos de software. Sobre a ótica do uso desses modelos se questiona “Quais são os modelos de qualidade de software para avaliar o produto e para melhoria dos processos de software mais utilizados nas empresas de desenvolvimento de software de Curitiba? E qual é o impacto do uso desses modelos na qualidade do produto?”

## 1.2 OBJETIVOS

Os objetivos deste trabalho estão relacionados em duas seções: objetivo geral e objetivos específicos.

### 1.2.1 Objetivo Geral

Identificar os modelos de qualidade de software para avaliar o produto e para melhoria dos processos de software utilizados no desenvolvimento de software pelas empresas desenvolvedoras de software em Curitiba.

### 1.2.2 Objetivos específicos

Os objetivos específicos estão divididos em:

- identificar os três modelos de qualidade de software para avaliar o produto e os três modelos de qualidade de software para melhoria dos processos de software na literatura;
- comparar os três modelos de qualidade de software para avaliar o produto entre si;
- comparar os três modelos de qualidade de software para melhoria do processo de software encontrados na literatura entre si;
- verificar, por meio de um instrumento de pesquisa (questionário), o uso dos seis modelos de qualidade de software no desenvolvimento de software em Curitiba;
- comparar os dados coletados por meio do instrumento de pesquisa, com os seis modelos de qualidade de software encontrados na literatura.

### 1.3 JUSTIFICATIVA

A compreensão da qualidade como um atributo do software levou à importância da inclusão de modelos de qualidade de software no Brasil. A partir da década de 90 com as avaliações CMM (*Capability Maturity Model*) e na década seguinte com as avaliações CMMI (*Capability Maturity Model - Integration*) e MPS.BR (Melhoria de Processos do Software Brasileiro) o mercado brasileiro começou a implantar os modelos de qualidade que melhorassem os processos de software nas empresas de desenvolvimento de sistemas computacionais<sup>1</sup>.

O uso desses modelos padroniza os processos de software, reduzindo o retrabalho e economizando tempo e recursos da empresa e por fim aumentando a satisfação do cliente. Em outro aspecto, os modelos de qualidade de software para avaliação verificam por meio quantitativo se o produto tem qualidade e também interferem na satisfação do cliente. O uso desses modelos de qualidade de software pelas empresas desenvolvedoras de software de Curitiba norteou esta pesquisa.

Do ponto de vista acadêmico, essa pesquisa visa à oportunidade de contribuir na identificação dos modelos de qualidade de software utilizados em desenvolvimento de software. Para o programa de pós-graduação a pesquisa contribui para o levantamento de informações relacionadas aos seis modelos de qualidade de software em desenvolvimento de software.

Para a comunidade do setor de desenvolvimento de software, esta pesquisa poderá auxiliar na compreensão da importância de conceitos que legitimam o uso dos modelos de qualidade de software no desenvolvimento de software em Curitiba. Para futuros pesquisadores que se interessarem em desenvolver estudos na área de qualidade de software, este trabalho, poderá auxiliar na pesquisa, por meio da literatura descrita no documento.

---

<sup>1</sup> Para este documento, “sistemas computacionais” é usado como sinônimo de “software”.



## 1.4 ESTRUTURA DO TRABALHO

Essa dissertação está organizada em seções, sendo que na primeira seção, **Introdução**, é apresentado um breve contexto a respeito de nuances que envolvem modelos de qualidade de software, seguido da apresentação do problema, dos objetivos geral e específicos e justificativa para desenvolvimento desta pesquisa.

A segunda seção, **Referencial Teórico**, está subdividida da seguinte forma: o item 2.1 apresenta conceitos e a evolução do termo qualidade. O item 2.2, apresenta um breve histórico da qualidade no Brasil; o item 2.3 apresenta conceitos de gestão da qualidade e excelência em gestão de acordo com o modelo brasileiro de excelência em gestão; o item 2.4 apresenta conceitos de qualidade de software; o item 2.5 apresenta seis modelos de qualidade de software divididos em seis subseções: 2.5.1 Modelo de qualidade de McCall; 2.5.2 Modelo FURPS+; 2.5.3 Modelo SQUARE; 2.5.4 Modelo SPICE 2.5.5 Modelo CMMI\_DEV; 2.5.6 Modelo MPS.Br-SW. O referencial teórico também apresenta conceitos de qualidade da informação no item 2.6 e as informações que garantem a qualidade de software no item 2.7.

A terceira seção, **Metodologia**, está dividida em seis seções: a primeira seção, 3.1, apresenta o processo metodológico; a segunda seção, 3.2 apresenta as delimitações e limitações da pesquisa; a terceira seção, 3.3, apresenta a definição do instrumento de coleta de dados; a quarta seção, 3.4, apresenta a coleta de dados; a quinta seção, 3.5, apresenta o tratamento dos dados; a sexta seção, 3.6, apresenta a análise de dados.

A quarta seção, **Resultados e Análise**, contém uma seção 4.1, que apresentam os resultados e análises obtidas por meio da pesquisa bibliográfica e a seção 4.2 que apresenta os resultados e análises do instrumento de pesquisa.

A quinta seção, **Considerações Finais**, apresenta as considerações finais deste trabalho.

## 2 REFERENCIAL TEÓRICO

Nesta seção são apresentadas informações sobre Qualidade de Software e sobre os modelos ISO/IEC 15504 (SPICE - *Software Process Improvement and Capability dEtermination*), McCall, FURPS+ (*Functionality, Usability, Reliability, Performance, Supportability*), MPS.Br (Melhoria de Processo de Software Brasileiro), CMMI (Modelo de Maturidade em Capacitação-Integração) e SQUARE (Requisitos de Qualidade e Avaliação de Produtos de Software).

### 2.1 EVOLUÇÃO DO TERMO QUALIDADE

A qualidade passou por quatro estágios durante a história. No primeiro estágio, até o final do século XIX, a concepção de qualidade era interpretada como atendimento as necessidades do cliente, o produto era customizado de acordo com a necessidade do cliente, porém fatores como confiabilidade<sup>2</sup>, conformidade<sup>3</sup>, metrologia<sup>4</sup>, tolerância<sup>5</sup> e especificação<sup>6</sup> ainda eram embrionários (CARVALHO, 2012).

Com a revolução industrial se iniciou o segundo estágio, a customização foi substituída pela padronização e produção em larga escala. A administração científica de Taylor e as linhas de montagens adicionaram os

---

<sup>2</sup>Confiabilidade é a extensão em que medidas repetidas de um fenômeno relativamente estável situam-se próximas umas das outras. (DICIO, 2014a)

<sup>3</sup>Conformidade é a qualidade do que é conforme ou de quem se conforma. Estado de duas ou de várias coisas iguais entre si.(DICIO, 2014b)

<sup>4</sup>Metrologia é a ciência que abrange todos os aspectos teóricos e práticos relativos às medições, qualquer que seja a incerteza em qualquer campo da ciência ou tecnologia. (INMETRO, 2014)

<sup>5</sup>Tolerância é a variação permissível da dimensão da peça, dada pela diferença entre as dimensões máxima e mínima.(BRAGA, 2014)

<sup>6</sup>Especificação é a determinação circunstanciada, descrição. Definição das características às quais deve responder uma instalação, uma construção, um material, uma confecção, um produto etc.(DICIO, 2014c)

inspetores às fábricas, que determinaram uma visão de controle de qualidade. Essa visão evoluiu a partir de 1924 com os gráficos de controle de Shewhart (1986) e na década de 30 com as ferramentas de controle estatístico, o desenvolvimento de sistemas de medidas, normas e inspeção por amostragem, que deu início ao terceiro estágio das eras da qualidade. Elton Mayo (1992) e a escola das Relações Humanas começaram a questionar a alienação no trabalho e a importância da motivação e participação do trabalhador, esse questionamento influenciou o modelo de qualidade japonês<sup>7</sup> após a Segunda Guerra Mundial.

Nas décadas de 40 e 50, com a fundação da *American Society for Quality Control* (ASQC), nos Estados Unidos e com a Associação Japonesa de Cientistas e Engenheiros (JUSE) a área de qualidade se solidificou. Juran (1998) apresentou um modelo de qualidade que envolvia planejamento e apuração dos custos da qualidade, Feigenbaum (1994) formulou o sistema de Controle da Qualidade Total (TQC) que influenciou o modelo ISO 9000 ABNT (2005) proposto pela Organização Internacional para Padronização (ISO). Crosby (1979) lançou os elementos que criaram o programa Zero Defeito. Deming e Juran exerceram forte influência no modelo japonês de qualidade, Controle de Qualidade por toda Empresa (CWQC).

Nesse modelo, para eliminar a inspeção, Taiichi Ohno (1997) retornou a responsabilidade pela qualidade do que produziam para os trabalhadores, para que pudessem interromper a produção assim que uma não conformidade ocorresse no sistema. Ohno reservava um período para que os trabalhadores discutissem em conjunto melhorias no processo. Ishikawa contribuiu na difusão das sete ferramentas da qualidade (diagrama de Ishikawa, folhas de controle, análise de Pareto, histograma, diagrama de escada, gráficos de controle e fluxo de controle).

Os modelos TQC e CWQC se difundiram rapidamente e em 1987 surgiu o modelo normativo ISO 9000 para gestão da qualidade. As normas vigentes seguiram como requisitos para ingresso em várias cadeias produtivas, servindo como certificador, eliminando custos com auditores e utilizando

---

<sup>7</sup> O modelo de qualidade japonês pós Segunda Guerra considerava o trabalhador como parte indispensável na empresa e depositava confiança em todos os funcionários

auditores e certificadores credenciados para este fim. No ano 2000, a nova revisão da série ISO 9000:2000 trouxe a visão de Gestão da Qualidade no lugar de garantia introduzindo gestão por processo e, diretrizes e com foco no cliente, a última versão de 2005 ABNT(2005) manteve a visão de gestão da qualidade. O último programa de gestão da qualidade surgiu no final da década de 1980, na Motorola, chamado de Seis Sigma.

Segundo Crosby (1979) “a qualidade é a conformidade às exigências” nesse âmbito, um software só tem qualidade se está consoante aos requisitos para os quais foi desenvolvido. Segundo Weinberg (1993 p.6) “cada afirmação acerca de qualidade é uma afirmação sobre alguma(s) pessoa(s)” podendo ser explícita ou implícita, então se pergunta “quem é a pessoa por trás dessa definição de qualidade?” o autor ainda cita várias situações em que a qualidade é relativa: para usuários cujo trabalho é afetado pelos defeitos e para gerentes que são criticados pelos defeitos, defeito zero é alta qualidade; para usuários que costumam comprar milhares de cópias do software e para gerentes de projeto que tem orçamentos limitados, baixo custo de desenvolvimento e alta qualidade; para usuários cujo trabalho está esperando o *software* e para distribuidores que desejam colonizar um mercado antes de seus concorrentes, desenvolvimento rápido é alta qualidade.

Garvin (2002) classifica cinco abordagens diferentes de qualidade:

- transcendental: qualidade é sinônimo de excelência inata, absoluta e universalmente reconhecível; não é passível de crítica e não pode ser medida;
- baseada no produto: qualidade é uma variável precisa e mensurável, oriunda dos atributos do produto; melhor qualidade é sinônimo de maiores e melhores características do produto;
- baseada no usuário: qualidade é uma variável subjetiva que melhor se adéqua as necessidades do usuário;
- baseada na produção: qualidade é uma variável precisa e mensurável, conformidade com as especificações pré-definidas; foco na eficiência, não da eficácia;

- baseada no valor: um produto é considerado de qualidade quando possui alto grau atendimento as conformidades a um preço acessível.

As abordagens levantadas por Garvin podem ser usadas em um mesmo ambiente de produção melhorando o dialogo entre clientes e fornecedores.

## 2.2 QUALIDADE NO BRASIL

As discussões sobre qualidade no Brasil segundo Barçante (2013) se iniciaram em 1876 com a criação do Instituto Nacional de Pesos e Medidas (INPM) no Rio de Janeiro; 64 anos depois foi criada a Associação Brasileira de Normas Técnicas (ABNT) e em 1973 foi criado o Instituto Nacional de Metrologia, Normalização e Qualidade Industrial (INMETRO) substituindo o INPM no Rio de Janeiro. Na década de 80 Crosby, Deming, Ishikawa e Feigenbaum ministraram uma série de palestras para executivos brasileiros, na mesma década foi lançado o Projeto de Especialização em Gestão de Qualidade (PEGQ) e emitido o primeiro certificado ISO 9000 no Brasil.

Barçante (2013) comenta que na década seguinte, em 1990 foi lançada a série brasileira das normas ISO 9000 pela ABNT e no ano seguinte, representantes de 39 empresas privadas e públicas instituíram em São Paulo, a Fundação para o Prêmio Nacional da Qualidade (FPNQ), criada para administrar o Prêmio Nacional da Qualidade (PNQ) e as atividades decorrentes do processo de premiação.

O autor enfatiza que na década de 1990, IBM Sumaré, Xerox, Citibank, SERASA, Alcoa Alumínio S/A, Siemens e Caterpillar ganharam o PNQ. Também, em 1991 foi realizado o I Congresso Brasileiro de Qualidade e Produtividade (CBQP) em Vitória, ES. O mesmo congresso em 1995 teve sua quinta realização na cidade de Curitiba, PR. Na década seguinte o PNQ foi entregue a várias empresas, entre elas, novamente SERASA em 2000, Gerdau Aços Finos Piratini e Politeo Indústria e Comércio S/A em 2002 e Dana Albarus – Divisão de Cardans em 2003. O CBQP também foi realizado em

várias cidades como Ouro Preto, MG em 2001, Vitória, ES em 2002 e Gramado, RS em 2004; Nos anos de 2005 e 2006 foram realizados dois congressos de qualidade na produção de software pela PRO QUALITY. O Prêmio Nacional da Qualidade continua sendo apresentado, na sua 22ª edição em 2013, duas empresas foram premiadas, Elektro eletricidade e serviços S/A e a Companhia de saneamento básico do estado de São Paulo.

### 2.3 GESTÃO DA QUALIDADE E EXCELÊNCIA DA GESTÃO

A partir dos conceitos relacionados sobre qualidade, surgiu a necessidade de gerenciar as atividades relacionadas à qualidade. Segundo a ABNT(2005) gestão da qualidade “são atividades coordenadas para dirigir e controlar uma organização, no que diz respeito à qualidade”. Miguel (2006) explica que a gestão da qualidade se subdivide em planejamento, controle, garantia e melhoria da qualidade; as atividades de melhoria da qualidade podem estar presentes nas atividades das outras três.

Baseado nos conceitos de gestão da qualidade e na inter-relação entre os componentes da gestão da qualidade (planejamento, controle, garantia e melhoria da qualidade), a visão de GQ evoluiu para Controle Total da Qualidade (TQC). Durante a década de 50 com o controle de qualidade por toda empresa (CWQC), defendida pelos japoneses, nessa concepção, toda organização deveria se envolver e se comprometer com a qualidade. Para Miguel (2006 p. 90) a evolução do TQC resultou em uma variante voltada à gestão, a Gestão da Qualidade Total (TQM). A ideia geral do TQM é que a qualidade passe a abranger todas as atividades de gerenciamento organizacional, não apenas o controle.

Miguel (2006 p.90-91) aponta os sete elementos mais comuns da TQM: (1) liderança e apoio da alta direção – prover liderança no processo de mudança, exemplaridade e motivação da força de trabalho da organização; (2) relacionamento com os clientes – concentrar as atividades com foco nos clientes e estabelecer canais de comunicação; (3) gestão da força de trabalho – aplicar os princípios da gestão de recursos humanos, com base em um sistema de trabalho em equipe e com maior participação dos funcionários,

processos de recrutamentos e seleção, capacitação e treinamento; (4) relação com os fornecedores – utilizar práticas de seleção e qualificação de fornecedores, bem como meios de medição de desempenho, estabelecer relações a longo prazo visando colaboração mútua; (5) gestão por processos – definir os processos-chaves da organização, promover práticas preventivas, auto-inspeção, utilização de planos de controle e utilização de métodos estatísticos na produção; (6) projeto de produto – envolver todas as áreas funcionais no processo de desenvolvimento de produto, visando desenvolver um produto; (7) fatos e dados da qualidade – disponibilizar os dados e informações relativas à qualidade, como parte de um sistema de gestão transparente e de fácil visualização.

O mesmo autor aponta três modelos para representar a TQM. O primeiro modelo, de Lascelles e Dale considera a TQM como uma evolução da qualidade ao longo do tempo, representado pelas fases de inspeção, controle estatístico, garantia e gestão da qualidade. O modelo de Shiba é o segundo, comentado pelo autor, esse modelo defende que o TQM é “um sistema em evolução por meio da melhoria contínua de produtos e serviços na busca do aumento da satisfação do cliente” (MIGUEL, 2006 p. 92). O modelo de Shiba considera o foco no cliente, melhoria contínua, participação total, entrelaçamento social e atividade com foco na qualidade, elementos fundamentais para a qualidade.

O terceiro modelo apresentado por Miguel (2006 p. 94-95) é conhecido como “bloco de construção” e foi proposto por Zaire. Esse modelo estabelece o TQM como dependente de fases construtivas, estabelece três pontos principais, a fundação, os pilares e o topo. A fundação se refere à melhoria contínua, envolvimento dos funcionários e gerenciamento das atividades que adicionam valor, os pilares se referem ao controle estatístico de processo e da qualidade, ao controle da cadeia de fornecedores e usuários, sistema de gerenciamento e controle, flexibilidade do processo e projeto do local de trabalho, por último, no topo está à liderança, planejamento da qualidade e visão competitiva.

O TQM abrange diversas áreas funcionais das organizações e diversos conceitos como liderança, controles de processo, relacionamento com clientes



e fornecedores, (MIGUEL, 2006 p. 96-97) explica que uma evolução da TQM veio na necessidade de incorporar os diversos interessados (*stakeholders*) de uma organização na busca por excelência em desempenho. Esse enfoque considera que uma organização é constituída basicamente por pessoas e processos e deve ser estruturada para atender as demandas desses agentes internos e externos. A avaliação de uma organização com relação às práticas de gestão utilizadas e os resultados organizacionais de forma a atender as necessidades de todos os interessados são obtidas a partir de modelos de excelência que compõe um prêmio de qualidade.

O primeiro prêmio de qualidade lançado no mundo foi o Prêmio Deming de qualidade, criado no Japão em 1950. Trinta e oito anos depois o Prêmio Malcolm Baldrige, foi criado nos EUA com o objetivo de aumentar a competitividade das empresas americanas em resposta à invasão de produtos japoneses durante esse período. No Brasil, a avaliação do PNQ, administrado pela FPNQ, se baseia em critérios de excelência adotados mundialmente e similares ao prêmio norte-americano.

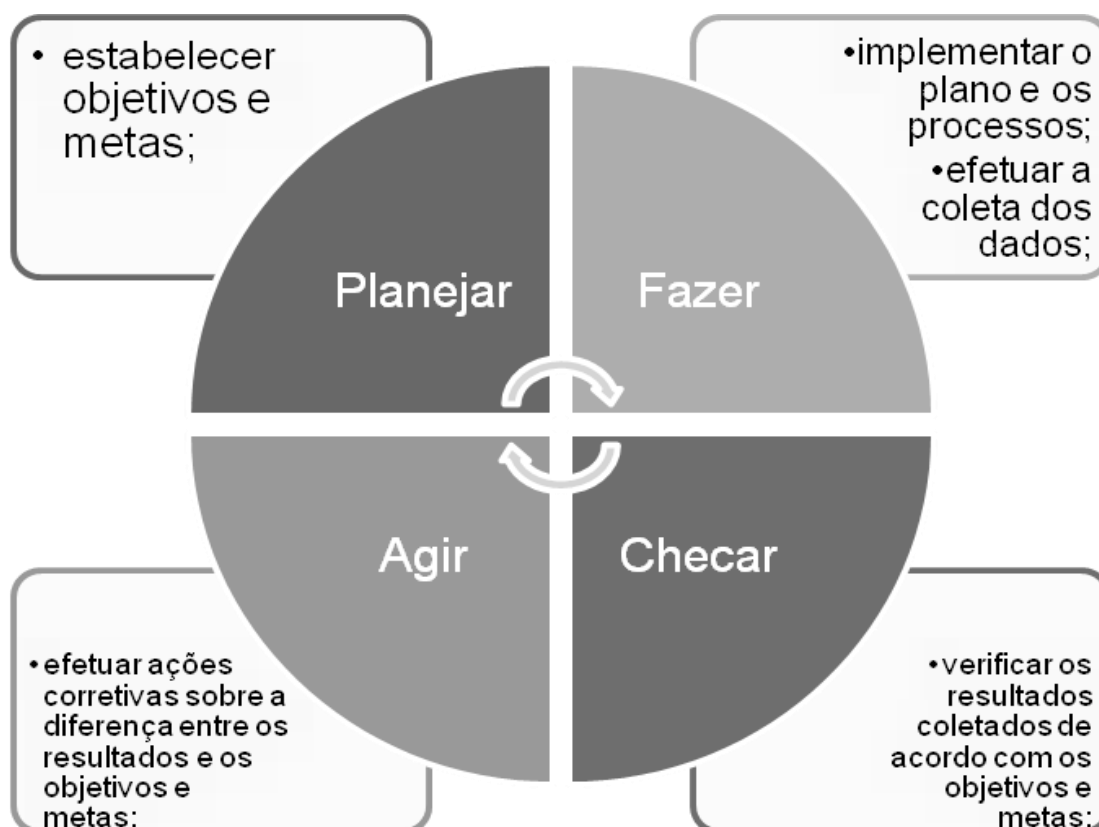
Em 2005 a FPNQ promoveu uma ampla reestruturação elegendo uma nova Governança na Assembleia Geral de Membros reformulou a logomarca passando a se chamar Fundação Nacional da Qualidade (FNQ). A FNQ, com a finalidade de disseminar os conhecimentos em excelência em gestão desenvolveu o Modelo de Excelência da Gestão® (MEG) FNQ(2014).

O modelo divide os processos de gestão em dois tipos: operacionais e gerenciais. Os processos operacionais usam como base o ciclo PDCA (*Plan, Do, Check, Action*) em que inicialmente se estabelece os objetivos e processos necessários para se obter resultados referentes às metas projetadas, no segundo passo se implementa o plano e os processos e efetua a coleta de dados. Na terceira fase do ciclo, se verifica os resultados coletados na fase anterior e na última fase do ciclo, se tomam ações corretivas sobre as diferenças entre os resultados reais e planejados. No início dos anos 90 a preocupação das organizações estava na gestão da qualidade aplicada aos processos que transformam matéria-prima em produtos. A partir dessa visão, se gerou a qualidade da gestão por meio de modelos de excelência.

Quanto aos processos gerenciais, são aqueles em que as informações são transformadas em decisões gerenciais. Nesse caso, a FNQ(2014) utiliza o

conceito de PDCL que substitui o 'A' de ação (*action*) pelo 'L' de aprender (*learn*) no ciclo PDCA simbolizando o aprendizado organizacional. A visão sobre os outros elementos do ciclo também muda, o planejamento ganha sentido de padronizar, assegurar a repetição, enquanto que a verificação se identifica por verificar se os padrões estão sendo cumpridos.

FIGURA 1 - CICLO PDCA



FONTE: SHEWHART (1986)

O MEG permite a verificação da incorporação do PDCL pela gestão da organização. Para FNQ (2014), o ciclo de controle PDCL é acrescido de um ciclo de aprendizado na ação. Ocorre uma avaliação dos processos, em seguida a melhoria desses e definições de práticas e padrões, e então retorna ao ciclo no planejamento da execução. A FNQ (2014) considera que uma boa prática de gestão deve possuir nove características: (1) metodologia e regras de funcionamento definidas; (2) pessoas responsáveis pela implantação, controle, etc.; (3) controle das execuções conforme os padrões e metodologia; (4) proatividade para solução antecipada e prevenção de situações

indesejáveis; (5) abrangência da prática – onde será aplicada?; (6) continuidade e periodicidade; (7) refinamento, avaliação e melhoramento; (8) integração com todas as partes interessadas; (9) resultados.

## 2.4 QUALIDADE DE SOFTWARE

De modo geral um software que atende os requisitos do cliente e as normas de desenvolvimento de software é considerado de qualidade (PRESSMAN, 2006). Sommerville (2003) explica que “a especificação deve ser orientada em direção às características do produto que o cliente deseja. Contudo, a organização de desenvolvimento pode também ter requisitos (como, os de facilidade de manutenção)” e “embora um produto de software possa atender à sua especificação, os usuários podem não considerá-lo um produto de alta qualidade”. Para Arthur (1994) “a qualidade não é apenas a ausência de defeitos. Na perspectiva do cliente, a qualidade significa tanto a presença de valor assim como a ausência dos defeitos”.

A qualidade do produto possui uma relação direta com a qualidade do processo empregado no desenvolvimento do software, por isso metodologias e padrões de qualidade como ISO 9000-3, CMMI e MPS.Br são importantes para a melhoria dos processos de software. O SWEBOK (2004) considera um produto como um artefato decorrente de um processo de construção do software, incluindo partes do programa final, documentação e relatório de análises de qualidade, e enfatiza que enquanto a maioria dos tratamentos de qualidade são descritos em termos de produto final e desempenho de software, praticas de engenharia de software requerem produtos intermediários importantes para avaliar todo processo de software.

Berwanger (2011, 5-6) comenta que o ciclo de vida de um software passa por oito fases (planejamento; análise e especificação de requisitos; projeto; implementação; testes; entrega e implantação; operação e; manutenção). A fase de planejamento fornece estrutura para que os gerentes desenvolvam estimativas de recursos, custos e prazos e se estabelece em todo projeto. Estabelecido o escopo do trabalho deve-se elaborar um plano de

projeto, configurando o processo a ser utilizado no desenvolvimento do software e atualizando o planejamento de acordo com o andamento do projeto, em cada fase do mesmo. Segundo Pressman (2006 p. 85) “o planejamento excessivo consome tempo e não produz frutos (...), mas falta de planejamento é uma receita para o caos.” Sommerville comenta que

o gerente de projeto deve prever os problemas que podem surgir e preparar soluções experimentais para esses problemas (...) o planejamento se inicia com uma avaliação de restrições que afetam o projeto (...) em conjunto com uma estimativa dos parâmetros para o projeto. Uma programação é traçada e as atividades definidas na programação são iniciadas ou liberadas para prosseguimento (...), depois de algum tempo as estimativas são examinadas e as discrepâncias são observadas. O plano sempre precisará ser modificado (SOMMERVILLE, 2003 p. 62-67).

A análise e a especificação de requisitos, segundo Sommerville (2003 p. 46-47) “destina-se a estabelecer quais funções são requeridas pelo sistema e as restrições sobre a operação e o desenvolvimento do sistema” geralmente são apresentados em dois níveis de detalhamento, um para os clientes com uma visão geral e outro com detalhes para os desenvolvedores do projeto. Pressman (2006, p.87) define um conjunto de cinco princípios operacionais para modelagem de análise, (1) o domínio da informação de um problema precisa ser representado e entendido, (2) as funções a serem desenvolvidas pelo software devem ser definidas, (3) o comportamento do software com base nas interações externas deve ser representado, (4) modelos que mostrem informações, funções e comportamento devem ser divididos de um modo que revele detalhes em forma de camada ou hierarquias (5) a tarefa de análise deve ir da informação essencial até os detalhes de implementação.

Na fase de projeto se descreve e molda a arquitetura do software, dados, interfaces, componentes e algoritmos. Segundo Sommerville (2003 p. 47) “o projeto envolve acrescentar formas e detalhes, à medida que o projeto é desenvolvido, com ‘retornos’ constantes, a fim de corrigir projetos anteriores”, podem-se encontrar erros e omissões de estágios anteriores. As informações de retorno permitem a melhoria do projeto. Pressman (2006, p.89) comenta

que “com exceção do projeto associado à infraestrutura do software, os elementos do modelo de projeto devem estar relacionados ao modelo de análise”. O autor também afirma que se deve considerar a arquitetura do sistema e o projeto de dados, projetar as interfaces com cuidado sintonizando as necessidades do usuário, a facilidade de uso e a integração para ajudar o testador na validação das funções dos componentes. Os componentes devem ser funcionalmente independentes e fracamente acoplados uns aos outros e ao ambiente externo. Os modelos devem ser facilmente reconhecidos e desenvolvidos da forma mais simples possível.

Na implementação (ou construção), o projeto deve ser traduzido para a plataforma, na(s) linguagem (ns) escolhida(s). Pressman (2006) descreve que “a atividade de construção compreende um conjunto de tarefas de codificação e testes que levam ao software operacional (...) pronto para ser entregue ao cliente ou usuário final”.

A fase de testes inclui vários níveis, incluindo testes de unidade, de integração, de sistemas, segurança, stress e desempenho. Em todos os níveis os testes efetuados e seus resultados documentados. Segundo Pressman (2006) “teste é um conjunto de atividades que podem ser planejadas antecipadamente (sic) e conduzidas sistematicamente”. Após a fase de testes, o sistema é posto em produção (fase de operação). Nessa fase é necessário treinar os usuários, configurar o ambiente de produção e caso necessário, converter bases de dados. Após a aceitação do produto o software passa a ser usado pelo cliente, na fase de operação.

Na fase de manutenção o software sofrerá reajustes por vários fatores, o cliente pode necessitar de mais desempenho ou de funcionalidades adicionais, ou mesmo por erros encontrados e produção. Dessa forma um novo projeto será iniciado, porém para atualizar e corrigir o produto desenvolvido.

Segundo a ISO/IEC 9126-1: 2003 (ABNT, 2003) a qualidade do *software* pode ser avaliada medindo-se os atributos internos (medidos com base no produto), externos (medidos com base em como o produto/processo/recurso se relaciona com o ambiente) ou os atributos de qualidade em uso. A finalidade é alcançar a qualidade necessária para atingir as reais necessidades do usuário. As necessidades de qualidade do usuário

podem ser especificadas como requisitos de qualidade pelas métricas de qualidade em uso, pelas métricas externas e pelas métricas internas.

A ISO/IEC 9126: 2003 (ABNT NBR, 2003) descreve que a totalidade das características do *software*, do ponto de vista interno, é chamada de qualidade interna. Esta medida é avaliada com relação aos requisitos de qualidade interna, enquanto que qualidade externa é a totalidade das características do *software* do ponto de vista externo, a qualidade é medida e avaliada quando o *software* é executado em ambiente simulado. Qualidade em uso se refere à visão do usuário, com o uso em um ambiente e um contexto especificado. A qualidade do software é um fator estratégico, segundo Paladini (2012 p.28) “qualidade é uma relação da organização com o mercado [...] dessa sociedade de hoje saem os consumidores de amanhã”. A qualidade não está apenas no produto ou serviço prestado, ou nos métodos empregados, está no somatório desses fatores. Nesse contexto serão apresentados modelos de qualidade voltados à qualidade de software.

## 2.5 MODELOS DE QUALIDADE DE SOFTWARE

Os modelos de qualidade de software como ISO/IEC 25000 (SQUARE), McCall, FURPS+ definem um conjunto de atributos e métricas padronizadas que avaliam o software como produto, enquanto os modelos MPS.Br, CMMI e SQUARE definem um conjunto de atividades de desenvolvimento e manutenção de *software* padronizados, que auxiliam no processo de melhoria da qualidade em empresas que desenvolvem software. Nessa seção são apresentados os conceitos sobre os modelos SQUARE, McCall, FURPS+, MPS.Br, CMMI e SPICE.

### 2.5.1 Modelo de qualidade de McCall

McCall propôs, em 1977, um modelo de avaliação da qualidade de *software* com três perspectivas e 11 fatores de qualidade do software que

devem fornecer um retrato completo da qualidade do software (ANJOS JUNIOR, 2009; FLEMING, 2014; ALMEIDA, 2010):

- Revisão do produto: capacidade de alterar o produto, [1] manutenção (esforço exigido para localizar e reparar erros em um software), [2] flexibilidade (efetuar mudanças necessárias) e [3] testabilidade (validar os requisitos do software);
- Transição do produto: identifica fatores de qualidade que influenciam a capacidade de adaptar o software a novos ambientes, [4] portabilidade (capacidade de transferir o *software* de um ambiente de software e hardware para outro), [5] reusabilidade (medida na qual o software [ou parte dele] pode ser reusado em outras aplicações) e [6] interoperabilidade (esforço exigido para acoplar um software a outro);
- Operação do produto: identifica fatores de qualidade que influenciam o uso do *software*, [7] confiabilidade (medição das falhas do software, execução das funções pretendidas com precisão exigida), [8] eficiência (uso de recursos do sistema, incluindo *hardware* e *software*), [9] integridade (controle de acesso aos dados e ao software), [10] exatidão (correção, satisfazer as especificações e objetivos visados pelo cliente) e [11] usabilidade (esforço do usuário para aprender a operar, preparar a entrada e interpretar a saída de um software).

O modelo de McCall apresenta 21 métricas, são:

- auditabilidade: medida da facilidade com que se pode verificar a conformidade a padrões;
- acurácia: medida da precisão dos tratamentos e do controle;
- padrões de comunicação: medida em que padrões de interfaces de máquinas, protocolos e larguras de banda são usados;
- inteireza: quanto à implementação total da função pretendida foi conseguida;
- concisão: medida da compactação do programa em nível de linhas de código por função;



- consistência: medida do uso de técnicas de projeto e documentação uniformes ao longo do ciclo de desenvolvimento do software;
- padrões de dados: medida do uso de padrões de estruturas e tipos de dados;
- tolerância a erros: medida dos danos que ocorrem quando o programa executa um erro;
- eficiência de execução: medida do desempenho de um programa em tempo de operação;
- expansibilidade: medida da amplitude da aplicabilidade dos componentes de um programa;
- generalidade: medida da amplitude da aplicabilidade dos componentes de um programa;
- independência de hardware: o quanto o software é desvinculado do hardware em que opera;
- instrumentação: o quanto o programa monitora sua própria operação e identifica os erros que venham a ocorrer;
- modularidade: medida da independência funcional dos componentes de um programa;
- operabilidade: medida da facilidade de operação de um programa;
- segurança: medida da disponibilidade de mecanismos que controlem ou protejam programas e dados;
- autodocumentação: medida do quanto o código fonte apresenta documentação significativa;
- simplicidade: medida do quanto o programa é entendido facilmente;
- independência do software: medida de quanto o programa é independente de particularidades não padronizadas da linguagem de programação, sistemas operacionais e ambientes;
- treinamento: medida da capacidade do software de auxiliar novos usuários na utilização do sistema.

O modelo apresentado por McCall se difundiu de forma que, outros modelos como o FURPS+ e as Normas ISO 9126 se basearam nesse modelo.

### 2.5.2 Modelo FURPS+

O modelo FURPS (*Functionality, Usability, Reliability, Performance, Supportability*) (QUALIDADEBR, 2008) é baseado no modelo de qualidade de software de McCall. Criado por Robert Grady na *Hewlett-Packard* é um acrônimo que representa as categorias que podem ser usadas na definição de requisitos de qualidade de *software*:

- *functionality* (Funcionalidade): representa os requisitos funcionais (características) do software. É uma categoria com diversas subcategorias que variam de acordo com a aplicação. Sua medição considera, principalmente, o cumprimento dos requisitos especificados;
- *usability* (Usabilidade): é o atributo que avalia a estética, o design, as prevenções de erros, documentações e consistência da interface do usuário;
- *reliability* (Confiabilidade): refere-se à integridade, conformidade e interoperabilidade do software. Os requisitos a serem considerados são: frequência e gravidade de falha; possibilidade de recuperação; possibilidade de previsão; exatidão; tempo médio entre falhas (MTBF);
- *performance* (Desempenho): avalia os requisitos de desempenho do software. Como medida podem ser usados: rendimento, tempo de resposta, consumo de memória, utilização da CPU, capacidade de carga e disponibilidade da aplicação;
- *supportability* (Suportabilidade): os requisitos de suportabilidade agrupam várias características, como: testabilidade, adaptabilidade, manutenibilidade, compatibilidade, configurabilidade, instalabilidade, escalabilidade, localizabilidade entre outros.

O símbolo de “+” identifica categorias adicionais que normalmente representam limitações:

- requisitos de design: se referem a restrições relacionadas ao desenho do software, pode incluir linguagens de programação, bibliotecas de classes, ferramentas de desenvolvimento;
- requisitos de implementação: especifica ou restringe o código ou a construção de um software, por exemplo: padrões obrigatórios, linguagens de programação, políticas de integridade de banco de dados, limites de recursos e ambientes operacionais;
- requisitos de interface: restringem as funcionalidades inerentes à interface do software com o cliente;
- requisitos físicos: especificam uma limitação física, por exemplo: configuração física de redes ou *hardware*.

O modelo FURPS+ integra o Processo Unificado da Rational (RUP – *Rational Unified Process*) de desenvolvimento de software e deve ser utilizado para levantamento de requisitos de software.

### 2.5.3 Modelo de qualidade SQUARE

O modelo de qualidade SQUARE (Requisitos de Qualidade e Avaliação de Produtos de Software) ABNT NBR ISO/IEC 25000:2008 (ABNT, 2008) evoluiu das normas ISO/IEC 9126 e ISO/IEC 14598 e propõe uma série de atributos de qualidade distribuídos em seis características principais:

- funcionalidade: capacidade de um software de promover funcionalidades que satisfaçam o usuário em suas necessidades declaradas e implícitas, dentro de um determinado contexto de uso. Suas sub-características são:
  - adequação: mede o quanto o conjunto de funcionalidades é adequado às necessidades do usuário;

- acurácia (ou precisão): representa a capacidade do software de fornecer resultados precisos ou com a precisão dentro do que foi acordado/solicitado;
  - interoperabilidade: trata da maneira como o *software* interage com outro(s) software(s) especificados;
  - segurança: mede a capacidade do software de proteger as informações do usuário e fornecê-las apenas (e sempre) às pessoas autorizadas;
- 
- confiabilidade: se mantém no nível de desempenho nas condições estabelecidas. Suas sub-características são:
    - maturidade: capacidade do software em evitar falhas decorrentes de defeitos no software;
    - tolerância a falhas: capacidade do software em manter o funcionamento adequado mesmo quando ocorrem defeitos nele ou nas suas interfaces externas;
    - recuperabilidade: capacidade de um software se recuperar após uma falha, restabelecendo seus níveis de desempenho e recuperando os seus dados;
- 
- usabilidade: capacidade do produto de software ser compreendido, seu funcionamento aprendido, ser operado e ser atraente ao usuário. Suas sub-características são:
    - inteligibilidade: facilidade com que o usuário pode compreender as suas funcionalidades e avaliar se o mesmo pode ser usado para satisfazer as suas necessidades específicas;
    - apreensibilidade: facilidade de aprendizado do software para os seus potenciais usuários;
    - operacionalidade: como o produto facilita a sua operação por parte do usuário, incluindo a maneira como ele tolera erros de operação;

- atratividade: envolve características que possam atrair um potencial usuário para o software, pode incluir desde a adequação das informações prestadas para o usuário até os requintes visuais utilizados na sua interface gráfica;
- eficiência: O tempo de execução e os recursos envolvidos devem ser compatíveis com o exigido pelo cliente. Suas sub-características são:
  - comportamento em relação ao tempo: avalia se os tempos de resposta (ou de processamento) estão dentro das especificações;
  - utilização de recursos: avalia tanto os recursos consumidos quanto a capacidade do software em utilizar os recursos disponíveis;
- manutenibilidade: capacidade (ou facilidade) do produto de software ser modificado, incluindo tanto as melhorias ou extensões de funcionalidade quanto às correções de defeitos. Suas sub-características são:
  - analisabilidade: facilidade em se diagnosticar eventuais problemas e identificar as causas das deficiências ou falhas;
  - modificabilidade: facilidade com que o comportamento do software pode ser modificado;
  - estabilidade: capacidade do software de evitar efeitos colaterais decorrentes de modificações introduzidas;
  - testabilidade: capacidade de se testar o software modificado, tanto quanto as novas funcionalidades quanto as não afetadas diretamente pela modificação;
- portabilidade: capacidade de transferência do software, de um ambiente para outro. Suas sub-características são:
  - adaptabilidade: capacidade do software se adaptar a diferentes ambientes sem a necessidade de ações adicionais (configurações);

- capacidade para ser instalado: facilidade com que pode se instalar o software em um novo ambiente;
- coexistência: mede o quão facilmente um software coexiste com outros instalados no mesmo ambiente;
- capacidade para substituir: capacidade que o software possui, de substituir outro software especificado, em um contexto de uso e ambiente específicos. Este atributo interage tanto com adaptabilidade quanto com a capacidade para ser instalado.

O modelo SQUARE avalia a qualidade de software sob a ótica do produto, por meio da funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade, porém, sob a ótica do processo de desenvolvimento do software, dos requisitos do usuário à entrega do produto final, existe um processo de desenvolvimento que pode comprometer a qualidade do software; neste caso são necessários modelos de qualidade de software que abordem o processo de desenvolvimento.

#### 2.5.4 Modelo de qualidade SPICE

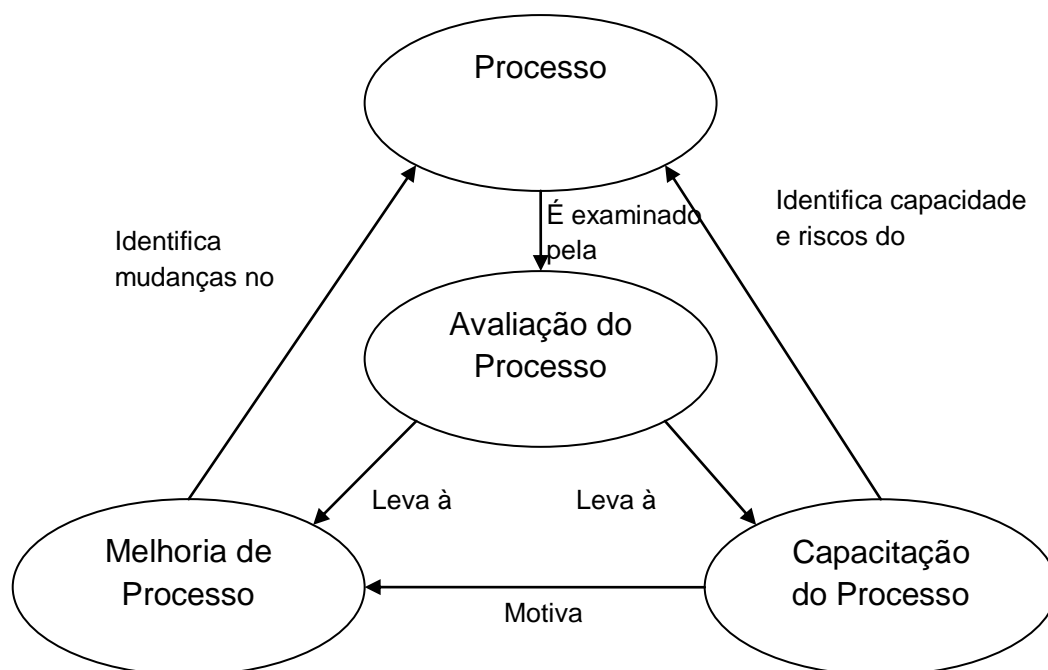
O modelo de qualidade SPICE ABNT NBR ISO/IEC 15504: 2008 (ABNT, 2008) propõe a melhoria dos processos de software e a determinação da capacidade de processos de uma empresa que desenvolve software. O modelo define duas dimensões: dimensão de processo e a dimensão de capacidade. A dimensão de processo corresponde à definição de um conjunto de processos universais e fundamentais para engenharia de software. A dimensão de capacidade corresponde à definição de um modelo de medição com base na identificação de um conjunto de atributos que permitem determinar a capacidade de um processo para atingir seus propósitos.

O modelo SPICE é dividido em cinco partes, a primeira apresenta os conceitos e vocabulários usados no documento, a segunda apresenta os requisitos mínimos para a realização de uma avaliação visando consistência e repetibilidade, a terceira parte é um guia para interpretação dos requisitos para realização de uma avaliação, a quarta parte é um guia para determinação da

capacidade e melhoria de processos da empresa, a última parte é um modelo de referência de processos aderente a segunda parte do modelo. Este trabalho trata da terceira e quarta parte da ISO 15504.

A figura 2 representa a relação entre o processo e sua avaliação, nessa relação o processo é avaliado, o que leva a melhoria e a capacitação do processo que auxilia na motivação da melhoria desse processo. Por fim o processo de melhoria identifica mudanças no processo, assim como a capacitação do processo identifica riscos e capacidades do processo que é reavaliado iniciando o ciclo novamente.

FIGURA 2 - ISO/IEC 15504



FONTE: OLIVEIRA (2009)

A dimensão do processo é baseada na norma ISO/IEC 12207 e apresenta cinco categorias de processos:

- cliente-fornecedor (*customer-supplier*): processos que afetam diretamente o cliente. Subcategorias: CUS.1 aquisição do software, CUS.2 gerencia de necessidades do cliente, CUS.3 fornecimento do software, CUS.4 operação do software, CUS.5 promoção do serviço ao cliente;
- engenharia de software: processos que envolvem a transformação de requisitos em um produto de software.

Subcategorias: ENG.1 desenvolvimento de requisitos e projeto do sistema<sup>8</sup>, ENG.2 desenvolvimento de requisitos de software, ENG.3 desenvolvimento do projeto de software, ENG.4 implementação do projeto do software, ENG.5 integração e teste do projeto do software, ENG.6 integração e teste do sistema, ENG.7 manutenção do sistema;

- apoio (*support*): processos de suporte que podem auxiliar outros processos em qualquer estágio do ciclo de vida do software. Subcategorias: SUP.1 desenvolvimento da documentação, SUP.2 gerência de configuração, SUP.3 execução da garantia da qualidade, SUP.4 execução da verificação dos produtos de trabalho, SUP.5 execução da validação dos produtos de trabalho, SUP.6 execução das revisões conjuntas, SUP.7 execução das auditorias, SUP.8 resolução de problemas;
- gerência (management): processos gerenciais. Subcategoria: MAN.1 gerenciamento do projeto, MAN.2 gerenciamento da qualidade, MAN.3 gerenciamento dos riscos; MAN.4 gerenciamento de subcontratantes.
- organização: processos associados às atividades gerais da empresa. Subcategorias: ORG.1 construção do negócio; ORG.2 definição do processo; ORG.3 melhoria do processo; ORG.4 fornecimento de recursos treinados; ORG.5 fornecimento de infraestrutura organizacional.

A dimensão de capacidade de processo estabelece uma escala de seis níveis crescentes, com início no nível 0 e termino no nível 5:

- nível 0 (Incompleto): o processo não é implementado ou falha na realização do propósito, não existem evidências que comprovem os resultados alcançados;
- nível 1 (Executado): o propósito do processo é geralmente alcançado, mas a execução não é rigorosamente planejada e acompanhada.

---

<sup>8</sup> Neste trabalho se entende sistema como algo que engloba software, hardware e usuários



- nível 2 (Gerenciado): o processo fornece produtos de trabalho de acordo com os procedimentos especificados, planejados e controlados. Os produtos de trabalho são gerados conforme os padrões e requisitos especificados.
- nível 3 (Estabelecido): o processo é definido por meio de princípios de engenharia de software e de um processo padrão da organização que disponibiliza os recursos necessários.
- nível 4 (Previsível): o processo definido é executado de maneira consistente dentro de limites de controle definidos, de modo a atingir os objetivos do processo.
- nível 5 (Em otimização): o desempenho do processo é otimizado para atender às necessidades de negócios atuais e futuros.

O modelo SPICE se originou do modelo ISO/IEC 12207, de onde herdou a arquitetura dos processos de ciclo de vida do software e do CMM (Modelo de Maturidade em Capacitação) de onde herdou o conceito de níveis de maturidade. A abordagem permite relacionar os níveis de maturidade a cada processo do ciclo de vida do software.

A avaliação do processo pode ser utilizada para compreender o estado dos processos da empresa e para determinar se os processos da empresa estão adequados para um requisito ou uma classe de requisitos específica ou ainda para determinar a adequação dos processos para um contrato ou classe de contratos específicos.

O modelo SPICE encoraja a empresa a desenvolver um programa de aperfeiçoamento de acordo com as necessidades comerciais da empresa.

#### 2.5.5 Modelo de qualidade CMMI-DEV®

O CMMI (Modelo de Maturidade em Capacitação-Integração) (SEI, 2010), evolução do modelo CMM (Modelo de Maturidade em Capacitação), foi desenvolvido pelo SEI (Instituto de Engenharia de Software) nos Estados Unidos. É um modelo de maturidade para melhoria do processo de

desenvolvimento e manutenção de produtos e serviços. A versão mais atual, 1.3, publicada em 2010 apresenta três modelos:

- CMMI para Desenvolvimento (CMMI-DEV): voltado ao processo de desenvolvimento de software;
- CMMI para Aquisição (CMMI-ACQ): voltada ao processo de aquisição e terceirização de bens e serviços de software;
- CMMI para Serviços (CMMI-SVC): voltado para processos de empresas prestadoras de serviços de software.

Esta seção trata do modelo para desenvolvimento de software. Esse modelo possui três dimensões: (1) pessoas; (2) procedimentos e métodos e; (3) ferramentas e equipamentos, que são alinhados por meio dos processos.

O CMMI possui duas representações, a contínua e a por níveis (ou estágios). A representação contínua permite que a organização escolha uma determinada área de processo e melhore os processos relacionados a esta área. A empresa pode se concentrar em um ponto problemático associado a um processo isolado ou pode trabalhar em várias áreas que estejam fortemente ligadas aos objetivos estratégicos da empresa. A representação por níveis oferece uma forma sistemática e estruturada para melhoria de processo. Nessa abordagem as áreas de processo são organizadas em níveis de maturidade, o que reduz a necessidade de escolhas associadas à melhoria de processo. A aprovação de cada estágio assegura a estrutura base para o nível seguinte, sendo que esta representação é caracterizada por cinco níveis:

- nível 1 (Inicial) [*Ad-hoc*]: os processos são caóticos, a empresa, normalmente não suporta processos. O sucesso depende de “pessoas-chave”, a empresa é incapaz de reproduzir os sucessos realizados no desenvolvimento e manutenção;
- nível 2 (Gerenciado): os processos são planejados, executados e monitorados de acordo com uma política empregada pela empresa, com pessoas especializadas para a função e produzem produtos controlados. Os processos envolvem os *stakeholders* (interessados) relevantes;

- nível 3 (Definido): os processos são bem caracterizados e entendidos, são descritos em padrões, procedimentos, ferramentas e métodos. O processo é estabelecido e melhorado ao longo do tempo;
- nível 4 (Quantitativamente gerenciado): a empresa estabelece objetivos quantitativos para qualidade e desempenho de processo que é compreendida em termos estatísticos e gerenciada por toda vida dos projetos;
- nível 5 (Em otimização): O objetivo é melhorar o desempenho do processo continuamente por meio de melhorias incrementais e inovadoras de processos e tecnologias.

Tanto a abordagem contínua, quanto a abordagem por níveis são compostas por 22 áreas de processo, são:

- Análise e Resolução de Causas (CAR);
- Gestão de Configuração (CM);
- Análise e Tomada de Decisões (DAR);
- Gestão Integrada de Projeto +IPPD (IPM +IPPD);
- Medição e Análise (MA);
- Implantação de Inovações na Organização (OID);
- Definição dos Processos da Organização +IPPD (OPD +IPPD);
- Foco nos Processos da Organização (OPF);
- Desempenho dos Processos da Organização (OPP);
- Treinamento na Organização (OT);
- Integração de Produto (PI);
- Monitoramento e Controle de Projeto (PMC);
- Planejamento de Projeto (PP);
- Garantia da Qualidade de Processo e Produto (PPQA);
- Gestão Quantitativa de Projeto (QPM);
- Desenvolvimento de Requisitos (RD);
- Gestão de Requisitos (REQM);
- Gestão de Riscos (RSKM);
- Gestão de Contrato com Fornecedores (SAM);
- Solução Técnica (TS);

- Validação (VAL);
- Verificação (VER);

Cada área de processo deve satisfazer metas genéricas, as metas são: GG 1 Processo Executado, GG2 Processo Gerenciado, GG 3 Processo Definido, GG 4 Processo Gerenciado Quantitativamente e GG 5 Processo em Otimização. As metas são estruturadas para que cada meta sirva como base para a seguinte, sendo assim, um processo gerenciado é um processo executado, um processo definido é um processo gerenciado, um processo gerenciado quantitativamente é um processo definido e um processo em otimização é um processo gerenciado quantitativamente.

A empresa que preferir pode buscar uma certificação CMMI para comprovar o nível em que está, o custo da certificação varia de empresa para empresa, de acordo com o porte e a nível desejado, uma certificação do nível 2 por exemplo, pode variar de U\$ 100.000,00 à U\$ 700.000,00 considerando apenas o custo da certificação, que se soma ao custo de treinamento, levantamento de processos, criação de equipes de auditoria e de qualidade e com as alterações dos processos culturais da empresa. O processo CMMI é uma das bases para o modelo brasileiro de qualidade de processo de software MPS.Br.

#### 2.5.6 Modelo de qualidade MPS.Br-SW

O Modelo de Melhoria de Processo de Software Brasileiro (MPS.Br), também conhecido por Melhoria de Processo de Software (MPS) (SOFTEX, 2012) foi desenvolvido pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) e visa à melhoria do processo de software, preferencialmente, em micro, pequenas e médias empresas; e baseado nas normas ISO/IEC 12207:2008, ISO/IEC 15504, ISO/IEC 20000, CMMI-DEV® e CMMI-SVC®, O MPS está dividido em quatro componentes:

- Modelo de Referência MPS para Software (MR-MPS-SW): contém os níveis de maturidade, processo e atributos do processo de melhoria de software voltados ao desenvolvimento de software, em conformidade com as normas ISO/IEC 15504-2;

- Modelo de Referência MPS para Serviços (MR-MPS-SV): contém as definições dos níveis de maturidade, processos e atributos do processo de software voltados a serviços de software, em conformidade com os requisitos de modelos de referência de processo da Norma Internacional ISO/IEC 15504-2;
- Método de avaliação: guia de boas práticas para a aquisição de software e serviços;
- Modelo de Negócio: descreve regras de negócio para implementação do MPS-SW e MPS-SV.

Esta seção apresenta características do MR-MPS-SW. Os processos desse modelo são descritos em termos de propósito, resultado e informações adicionais. O propósito descreve o objetivo geral a ser atingido durante a execução do processo, enquanto os resultados esperados estabelecem os resultados a serem obtidos com a implementação do processo. A capacidade do processo é representada por um conjunto de Atributos de Processo (AP) atendidos por meio dos Resultados esperados dos Atributos de Processo (RAP), requeridos em todos os processos. São nove atributos:

1. AP 1.1 – O processo é executado;
2. AP 1.2 – O processo é gerenciado;
3. AP 2.2 – Os produtos de trabalho do processo são gerenciados;
4. AP 3.1 – O processo é definido;
5. AP 3.2 – O processo está implementado;
6. AP 4.1 – O processo é medido;
7. AP 4.2 – O processo é controlado;
8. AP 5.1 – O processo é objeto de melhorias incrementais e inovações;
9. AP 5.2 – O processo é otimizado continuamente.

Esse modelo define sete níveis de maturidade de acordo com a evolução de processos de software, sendo “A” o nível mais alto e G o nível mais baixo. À medida que a empresa evolui nos níveis de maturidade, os RAPs devem ser atingidos e acumulados para o nível seguinte. Os níveis de A a G são relacionados a seguir:

- G (Parcialmente Gerenciado): composto pelos processos de Gerência de Projetos e de Requisitos; o propósito do processo de gerência de projeto é estabelecer e manter planos que definam as atividades, recursos e responsabilidades do projeto, assim como prover informações sobre o andamento do projeto. O propósito do processo de Gerência de Requisitos é gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto. Este nível deve atender aos atributos de processo AP 1.1 e AP 2.1.
  
- F (Gerenciado): composto pelo nível de maturidade G, acrescido dos processos de Aquisição, Garantia da Qualidade, Gerência de Configuração, Gerência de Portfólio de Projetos e Medição; o propósito do processo de Aquisição é gerenciar a aquisição de produtos (software e serviços caso estes sejam inclusos na entrega do produto) que satisfaçam às necessidades do cliente. O propósito do processo de Gerência de Configuração é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos. O propósito do processo de Garantia da Qualidade é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos, procedimentos e padrões estabelecidos. O propósito do processo de Gerência de Configuração é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos; o propósito do processo de Gerência de Portfólio de Projetos é iniciar e manter projetos que sejam necessários, suficientes e sustentáveis, de forma a atender os objetivos estratégicos da organização. O propósito do processo de Medição é coletar, armazenar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos,

de forma a apoiar os objetivos organizacionais. Este nível deve atender aos atributos de processo AP 1.1, AP 2.1 e AP 2.2.

- E (Parcialmente Definido): composto pelos níveis G e F e os processos de Avaliação e Melhoria do Processo Organizacional, Definição do Processo Organizacional, Gerência de Recursos Humanos e Gerência de Reutilização; o processo de Gerência de Projetos passa a atuar com base no processo definido para o projeto e nos planos integrados; o propósito do processo Avaliação e Melhoria do Processo Organizacional é determinar o quanto os processos padrão da empresa contribuem para alcançar os objetivos de negócio da empresa e para apoiar o planejamento, realização e implantação de melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos. O propósito do processo Definição do Processo Organizacional é estabelecer e manter um conjunto de ativos de processos organizacionais e padrões do ambiente de trabalho usáveis e aplicáveis às necessidades de negócio da empresa. O propósito do processo Gerência de Recursos Humanos é prover a organização e os projetos com os recursos humanos necessários e manter as competências adequadas, da empresa, às necessidades do negócio. O propósito do processo Gerência de Reutilização é gerenciar o ciclo de vida dos ativos reutilizáveis. Este nível deve atender aos atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2 e AP 3.22.
- D (Largamente Definido): composto pelos níveis G, F, E e os processos de Desenvolvimento de Requisitos, Integração do Produto, Projeto e Construção do Produto, Validação, e Verificação. O propósito do processo Desenvolvimento de Requisitos é definir os requisitos do cliente, do produto e dos componentes do produto. O propósito do processo Integração do Produto é compor os componentes do produto, produzindo um produto integrado consistente com seu projeto, e demonstrar que os requisitos funcionais e não funcionais são satisfeitos para

o ambiente alvo ou equivalente. O propósito do processo Projeto e Construção do Produto é projetar, desenvolver e implementar soluções para atender aos requisitos. O propósito do processo Validação é confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido; o propósito do processo Verificação é confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados. Este nível deve atender aos atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2.

- C (Definido): composto pelos níveis G, F, E, D e os processos Desenvolvimento para Reutilização, Gerência de Decisões e Gerência de Riscos; o propósito do processo Desenvolvimento para Reutilização é identificar oportunidades de reutilização sistemática de ativos na organização e, se possível, estabelecer um programa de reutilização para desenvolver ativos a partir de engenharia de domínios de aplicação. O propósito do processo Gerência de Decisões é analisar possíveis decisões críticas usando um processo formal, com critérios estabelecidos, para avaliação das alternativas identificadas; o propósito do processo Gerência de Riscos é identificar, analisar, tratar, monitorar e reduzir continuamente os riscos em nível organizacional e de projeto. Este nível deve atender aos atributos de processo. Este nível deve atender aos atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2.
- B (Gerenciado quantitativamente): composto pelos níveis G, F, E, D, C e o processo de Gerência de Projetos, passa a atender aos objetivos de gerenciamento quantitativo e os atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1 e AP 4.2.
- A (Em Otimização): composto pelos níveis G, F, E, D, C e B e os atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1 e AP 4.2.



O Modelo MPS.Br teve seu início em 2003 e se tornou popular entre empresas de médio e pequeno porte brasileiras por ser compatível com os modelos ISO e CMMI e ser mais viável do ponto de vista econômico, pois permite que as empresas incorporem os processos de forma mais gradual durante os sete níveis de maturidade, que os outros dois modelos que possuem seis e cinco níveis respectivamente. A certificação MPS.Br varia de acordo com o porte e as necessidades da empresa, mas uma certificação do nível F pode variar de R\$ 60.000 à R\$ 300.000,00 e tem validade de 3 anos.

## 2.6 QUALIDADE DA INFORMAÇÃO

Informação é diferente de dado e de conhecimento, Davenport e Prusak (1998 p. 3-5) comentam que “dados são um conjunto de fatos distintos e objetivos, relativos a eventos”, já a informação é “uma mensagem, geralmente na forma de um documento ou uma comunicação audível ou visível (...) tem um emissor e um receptor (...) tem por finalidade mudar o modo como o destinatário vê algo, exercer algum impacto sobre seu julgamento e comportamento”, porém apenas o receptor pode compreender a mensagem passada como informação, caso contrário ainda é uma mensagem ou um dado, ou seja, um dado só se torna informação quando tem um significado para quem o recebe.

Ter informação em si não basta, é necessário ter informação de qualidade, confiável e com exatidão para evitar problemas dessa categoria. “Os gestores precisam avaliar constantemente a qualidade das informações coletadas” (STRAUHS et al., 2012 p. 29). Segundo Carvalho e Tavares (2001 p. 12) “a constatação de que as empresas (...) captam, produzem e emitem informação traz como consequência a necessidade de métodos racionais de captação, análise, processamento e disseminação das mesmas”, dentro de uma organização muitas informações são transitadas, As autoras formulam o conceito de Informação Essencial como “essencialmente útil ao negócio de uma organização e que se apresenta de forma limpa, racionalizada e sistematizada”.

A informação deve ser usada de forma estratégica pela organização, porém, uma preocupação deve ser a qualidade dessa informação, Calazans (2008) comenta que o tema começou a ser discutido em 1998, no Seminário do *Nordic Council for Scientific Information and Research Libraries* onde alguns autores começaram a discutir e propor dimensões sobre esse conceito.

## 2.7 INFORMAÇÕES QUE GARANTEM A QUALIDADE DE SOFTWARE

Para que um software seja desenvolvido com qualidade é importante que exista um controle de todos os processos referentes à criação do mesmo. Sommerville (2003 p. 466) comenta que o “controle da qualidade envolve supervisionar o processo de desenvolvimento de software, a fim de assegurar que os procedimentos e os padrões de garantia de qualidade sejam seguidos”. O processo de controle tem seus procedimentos e relatórios, os procedimentos devem ser diretos e de fácil compreensão pelos engenheiros do projeto.

Os processos de construção dos produtos devem ser revisados periodicamente e aperfeiçoados, porém, a construção de sistemas de informação exige uma atenção especial ao fator humano. Sommerville (2003, p. 479-480) comenta que para desenvolvimento de software quatro fatores ([1] tecnologia de desenvolvimento; [2] qualidade do processo; [3] qualidade do pessoal e; [4] custo, tempo e cronograma) influenciam na construção de um novo produto, a influência de cada um desses fatores depende do tamanho e tipo de projeto. Para grandes projetos com subsistemas e produzido por diferentes equipes o principal problema é a integração, por isso o fator que mais influencia é a qualidade do processo, enquanto que para pequenos projetos que exigem uma equipe menor, o fator que mais influencia é a capacidade da equipe de desenvolvimento. Se a equipe tem um alto nível e experiência, provavelmente o sistema será de alta qualidade, porém se a equipe tem menos experiência e menor capacidade, um processo bem definido poderá limitar o dano, mas ainda assim não produzirá um software de qualidade. Para equipes pequenas é importante ter uma tecnologia de

desenvolvimento estruturada, pois a equipe não pode dedicar muito tempo para procedimentos administrativos. Os engenheiros dedicam a maior parte do tempo projetando e programando o sistema, neste caso ferramentas de desenvolvimento podem afetar significativamente a produtividade.

Para a produção de um software de qualidade, de acordo com Sommerville (2003, p. 479-480), são necessárias informações relacionadas aos quatro fatores. O planejamento do projeto deve incluir informações sobre ferramentas de desenvolvimento, linguagens utilizadas, ferramentas usadas durante o projeto e especificações destas ferramentas; Com relação à qualidade do pessoal, os gerentes devem coletar e obter dados sobre os *stakeholders*. Todos os envolvidos no projeto influenciam de alguma forma a qualidade do sistema desenvolvido pois uma equipe de desenvolvimento com qualidade tem conhecimento necessário para produzir um software de forma rápida, concisa e com a preocupação de manter as funcionalidades, suportabilidade, desempenho e confiança.

Informações relacionadas ao tempo de desenvolvimento, tempo de entrega, cronograma, e custos do projeto mantêm a visão de cliente como um *stakeholder* e visa garantir a entrega do projeto em um prazo aceitável. A documentação gerada nos processos deve ser atualizada de acordo com o andamento do projeto, a documentação auxilia no treinamento de novos funcionários e torna rápido e fácil a compreensão das atividades de desenvolvimento de um software.

Segundo Rezende (1997, p. 173) os objetivos da documentação de software são para divulgar e deixar claro e transparente o que o sistema faz, como utilizá-lo, vantagens, operações e requisitos para funcionamento, mostrar as técnicas ou ferramentas utilizadas para seu desenvolvimento, tempo, custo, equipe, cronograma, metodologia, integração, etc. Esta documentação é utilizada para orientar e treinar o usuário e possibilitar a garantia da continuidade do funcionamento do produto. O mesmo autor enfatiza que a qualidade da documentação está calcada em três pilares, devendo:

- (1) ser completa – padronização elimina falhas, nem sempre o que é irrelevante para um, é irrelevante para outro, não deve estar faltando nenhuma informação;

- (2) possuir precisão – estar de acordo com as especificações, e ser revisada por duas pessoas, com visões diferentes; e
- (3) conter clareza – completa, concisa, com fácil identificação, localização e organização.

O retorno da visão do cliente é muito importante para a continuidade do ciclo de vida de um software. Maldonado (2010, p.53) comenta que “como parte dos seus pacotes de serviços as software-houses implantam serviços de atendimento ao usuário, com o objetivo de garantir a plena utilização das suas soluções de software” por estarem mais próximo do cliente, estes serviços se constituem no principal canal de comunicação com o cliente. Para Gronros citado por Pereira, Bellini e Luce (2010), os clientes demandam ofertas mais holísticas, não apenas produtos e serviços, de modo que a vantagem se torna em gerenciar os elementos adicionais da oferta mais eficientemente que a concorrência.

O reuso da informação para a continuidade do processo de construção de sistemas, informações levantadas durante a construção de um sistema e com o cliente durante o uso do mesmo, podem ser usadas no desenvolvimento de novos produtos e produtos mais eficientes e fáceis de usar. Dingsoyr (2000, p.56) relata que há um foco especial no reuso de experiências de projetos anteriores para aumento de qualidade em novos projetos. Podem-se aproveitar diretamente os artefatos gerados como também criar um ambiente de aprendizagem entre os desenvolvedores de uma equipe ou projeto.

As informações do desenvolvimento, manutenção e uso do software, geradas durante todo processo de desenvolvimento, manutenção e uso desse software devem ser utilizadas para auxiliar a qualidade dos softwares em desenvolvimento e que serão desenvolvidos e também para informar aos outros setores da organização as informações necessárias.

### 3 METODOLOGIA

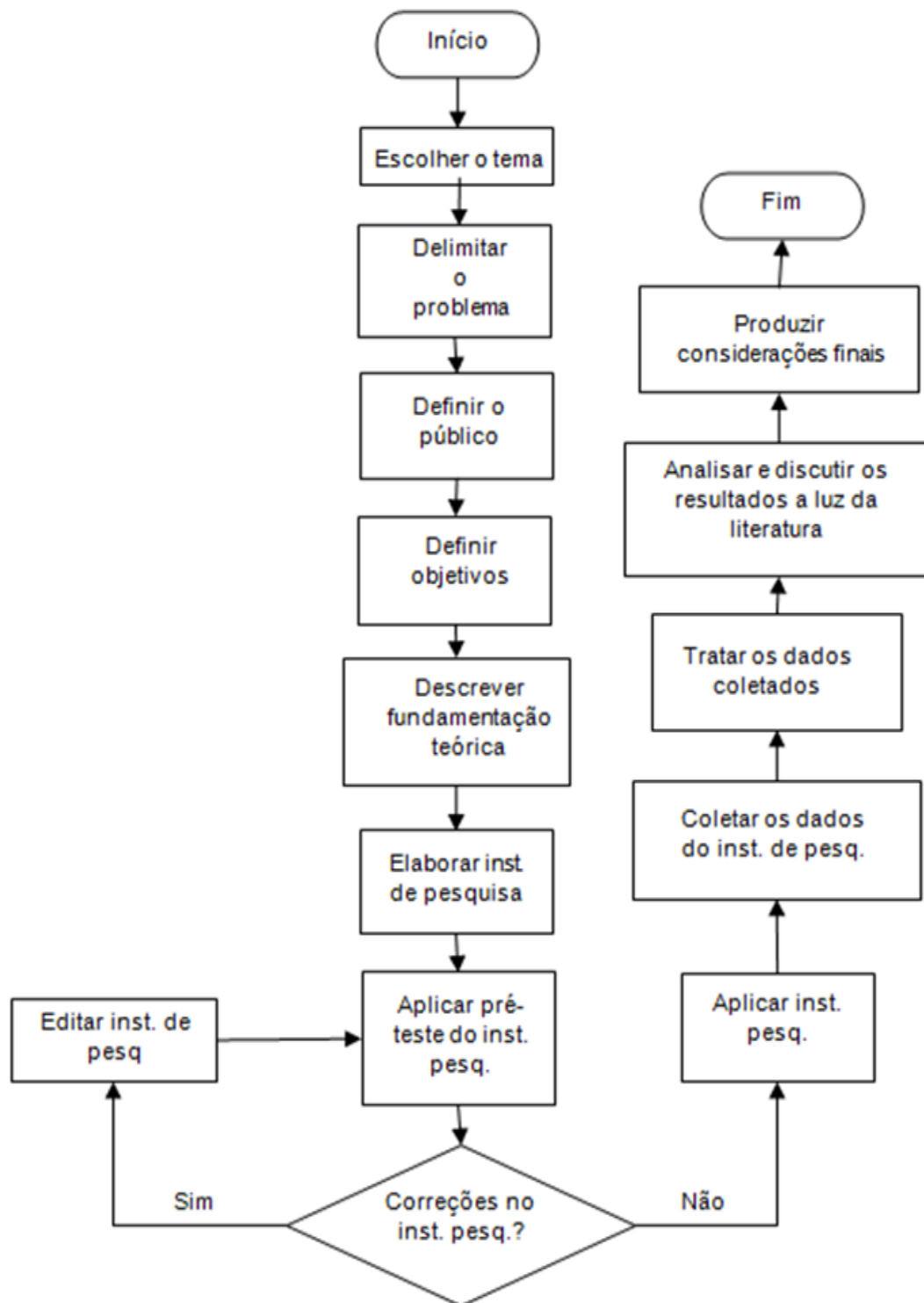
Nessa seção é apresentada a metodologia da pesquisa. A metodologia está dividida em cinco seções: a primeira seção, 3.1, apresenta o processo metodológico; a segunda seção, 3.2, apresenta as delimitações e limitações da pesquisa; a terceira seção, 3.2 apresenta a definição do instrumento de coleta de dados; a quarta seção, 3.4, apresenta a coleta de dados; a quinta seção, 3.5, apresenta o tratamento dos dados e a sexta seção, 3.6, apresenta a análise de dados.

#### 3.1 PROCESSO METODOLÓGICO

O processo de elaboração da pesquisa seguiu as etapas: escolha do tema, definição do problema, definição do público, definição dos objetivos (geral e específicos), elaboração da fundamentação teórica, elaboração do instrumento de pesquisa, validação do instrumento de pesquisa por meio de um pré-teste, aplicação da pesquisa, coleta dos dados, tratamento de dados, análise e discussão resultados da pesquisa com os dados coletados e por fim as conclusões. O fluxograma 1 apresenta a sequência das etapas.

Com base no tema, Modelos de Qualidade de *Software* e uso dos Modelos de Qualidade de Software em desenvolvimento de *software*, se definiu o problema e a questão a ser respondida, “Quais são os modelos de qualidade de software para avaliar o produto e para melhoria dos processos de software mais utilizados nas empresas de desenvolvimento de software de Curitiba? E qual é o impacto do uso desses modelos na qualidade do produto?” (seção 1.1). A partir do problema, definiram-se os objetivos geral (seção 1.2.1) e específicos (seção 1.2.2); com base no problema e nos objetivos identificaram-se os temas a serem estudados para responder a pergunta proposta.

FLUXOGRAMA 1 - METODOLOGIA DESTA PESQUISA



FONTE: O AUTOR (2014)

Os temas identificados foram estudados para desenvolver o referencial teórico. Foi realizada uma pesquisa bibliográfica em livros, artigos e bases de

dados. As principais bases utilizadas foram o Google Scholar, o portal de periódicos CAPES, e o acervo SCIELO, as duas últimas com acesso concedido pela Universidade Federal do Paraná (UFPR). Os livros foram identificados com base nas referências encontradas em artigos, nas bibliotecas da UFPR e livrarias, por indicação de professores e colegas e as páginas web foram identificadas pelas referências de artigos e por buscadores web como Google e indicação de professores.

Foram selecionados três modelos referentes à qualidade do produto que refletem a evolução da avaliação do software, sendo o modelo de McCall, a base para os outros dois modelos. O modelo FURPS+ é voltado para requisitos, enquanto o modelo SQUARE, apresenta as características levantadas por McCall de forma detalhada. A seleção do modelo SQUARE se deu pela representatividade, desenvolvido pela ISO o modelo é divulgado internacionalmente. O modelo McCall foi selecionado por ser o precursor do modelo SQUARE, enquanto a seleção do FURPS+ se dá pelo complemento do McCall em relação à visão de levantamento de requisitos de software e pelo uso do modelo no desenvolvimento com a metodologia RUP (Processo Unificado da Rational).

O modelo SPICE apresenta seis níveis, enquanto que os modelos CMMI e MPS.Br apresentam cinco e sete níveis respectivamente. O modelo SPICE é um modelo normativo internacional; o modelo CMMI é baseado nas melhores práticas para desenvolvimento, manutenção e serviços de software; o modelo MPS.Br é baseado no modelo CMMI e no modelo SPICE, porém, adaptado para empresas que não podem obter um investimento alto no início do processo de capacitação da empresa.

A pesquisa realizada na fundamentação teórica obteve resultados que serão discutidos na seção de Resultados e Análises. O instrumento de pesquisa, coleta, tratamento e análise dos dados serão discutidos nas próximas seções.

### 3.2 DELIMITAÇÃO E LIMITAÇÕES DA PESQUISA

Essa seção apresenta as delimitações e limitações da pesquisa decorrentes de possíveis erros de coleta, análise e interpretação dos dados. A pesquisa abrange as empresas do setor de TI conforme a classificação CNAE 2.1 CONCLA – Comissão Nacional de Classificação (2014) presente no quadro 1. Essa classificação abrange todas as empresas que desenvolvem programas de computador.

A pesquisa foi realizada em Curitiba por motivo de restrição física no deslocamento do pesquisador para as empresas respondentes. Os dados da população foram obtidos por meio da Junta Comercial do Paraná (JUCEPAR), uma instituição vinculada ao Ministério do Desenvolvimento, Indústria e Comércio Exterior que realiza registro e cadastro de empresas mercantis no Paraná, no ano de 2014. Essa instituição foi escolhida por apresentar, na pesquisa, a possibilidade de buscar empresas ativas no ano de 2014, diferente dos institutos IPARDES e IBGE que possuíam uma base com atualizações do ano de 2011.

<b>Classificação CNAE 2.1</b>	
Seção J	Informação e Comunicação
Divisão 62	Atividades dos Serviços de Tecnologia da Informação
Grupo 620	Atividades dos Serviços de Tecnologia da Informação
<b>Classes</b>	<b>Quantidade em Curitiba</b>
DESENVOLVIMENTO DE PROGRAMAS DE COMPUTADOR SOB ENCOMENDA (6201-5)	476
DESENVOLVIMENTO E LICENCIAMENTO DE PROGRAMAS DE COMPUTADOR CUSTOMIZÁVEIS (6202-3)	229
DESENVOLVIMENTO E LICENCIAMENTO DE PROGRAMAS DE COMPUTADOR NÃO-CUSTOMIZÁVEIS (6203-1)	145
<b>TOTAL</b>	<b>850</b>

QUADRO 1 - CLASSIFICAÇÃO DAS ATIVIDADES ECONÔMICAS SEGUNDO CNAE 2.1  
 FONTE: (CONCLA, 2014)



A pesquisa foi realizada com base no ano de 2014. Segundo as informações desse ano existiam 476 empresas de desenvolvimento de programas<sup>9</sup> de computador sob encomenda, 229 empresas de desenvolvimento e licenciamento de programas de computador customizáveis e 145 empresas desenvolvimento e licenciamento de programas de computador não customizáveis em Curitiba. O quadro 1 apresenta valores por classe e total da população da pesquisa em Curitiba. Considera-se que a subdivisão por classes contida nesse quadro não influencia no fator estudado em função do critério de escolha ser que todas são desenvolvedoras de programas.

Por meio do questionário as empresas foram classificadas pelo porte, da seguinte forma:

- microempresa - renda operacional bruta anual menor ou igual a R\$ 2,4 milhões;
- pequeno porte- renda operacional bruta anual maior que R\$ 2,4 milhões e menor ou igual a R\$ 16 milhões;
- médio porte- renda operacional bruta anual maior que R\$ 16 milhões e menor ou igual a R\$ 90 milhões;
- médio-grande porte - renda operacional bruta anual maior que R\$ 90 milhões e menor ou igual a R\$ 300 milhões;
- grande porte - renda operacional bruta anual maior que R\$ 300 milhões;

Com base na população, e conhecendo a amostra pesquisada foi calculado o erro amostral por meio do calculo amostral para população finita e variável contínua apresentado na equação (1):

$$n = \frac{Z^2 * p * q}{d^2} \quad (1)$$

Sendo que:

n – amostra;

Z – intervalo de confiança;

---

<sup>9</sup> Neste documento programa possui o mesmo significado de software.

p – estimativa de ocorrência do atributo estudado;

q – estimativa de não ocorrência do atributo estudado;

d – erro amostral.

Considerando a amostra de 75 empresas, com intervalo de confiança de 95%, equivalente a 1,96 pela tabela de confiança Z e considerando a estimativa de ocorrência das empresas de software de 0,5 e de não ocorrência de 0,5 (todas as empresas possuem a mesma estimativa de participar e de não participar da pesquisa), se obtêm:

$$75 = \frac{1,96^2 * 0,5 * 0,5}{d^2}$$

Isolando o erro amostra se obtém:

$$d^2 = \frac{1,96^2 * 0,5 * 0,5}{75}$$

Elevando ao quadrado o intervalo de confiança e multiplicando os valores do denominador se obtém:

$$d^2 = \frac{0,9604}{75} \rightarrow d^2 = 0,0128$$

Passando o exponencial como raiz para a o lado direito da equação se obtêm:

$$d = 0,1132 = 11,32\%$$

Com uma população de 850 empresas e uma amostra de 75 empresas se obtêm um erro amostral de 11,32% em relação à população, a amostra foi probabilística aleatória simples, pois todas as empresas possuíam a mesma probabilidade de responder ao questionário.

### 3.3 DEFINIÇÃO DO INSTRUMENTO DE COLETA DE DADOS

Como ferramenta de coleta de dados, optou-se por um questionário aplicado à área de desenvolvimento de software. Por meio da ferramenta (Apêndice B - Questionário) serão obtidos os dados que posteriormente descreveram a amostra. O instrumento contém 11 questões voltadas a modelos de qualidade de *software*.

A primeira questão visa identificar a empresa para evitar respostas repedidas, a segunda questão visa conhecer o porte econômico da empresa para verificar se existe relação entre o porte da empresa e o uso de modelos de qualidade. A terceira questão verifica a utilização de modelo(s) de qualidade para avaliação do produto, como o modelo McCall e FURPS+. A quarta questão verifica o uso de um modelo de qualidade de processo e manutenção de software como SPICE e CMMI. A quinta questão é dependente da quarta, caso o respondente tenha escolhido um dos modelos apresentados na questão anterior, deve informar qual o nível em que a empresa se encontra.

A sexta questão depende das questões 3 e 4, caso as duas questões sejam respondidas de forma negativa, o respondente deve informar se houve tentativa de aplicação de algum modelo. Na sétima questão, caso a resposta seja sim, o respondente deve explicar qual o modelo escolhido e as dificuldades encontradas com a adoção desse modelo. A oitava questão avalia o impacto da utilização de modelos de qualidade de software sobre o custo, tempo, desempenho, retrabalho, satisfação do usuário e manutenção do software.

A questão nove avalia a importância das características dos produtos de software, de acordo com a visão da empresa em questão, com base na funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e limitações de interface, implementação, física e/ou de design. A décima questão avalia a importância dos processos de software, de acordo com a visão da empresa. Consideram-se os processos que afetam diretamente o cliente, que envolvem a transformação de requisitos em um produto, de suporte, que podem auxiliar outros processos, gerenciais e associados a

atividades gerais da empresa. A última verifica a visão do respondente quanto aos modelos de melhoria dos processos de software e manutenção.

### 3.4 COLETA DE DADOS

A coleta de dados foi composta de três fases: pré-teste do questionário, aplicação do questionário e coleta dos questionários. A seguir serão descritas cada uma das fases.

O pré-teste (Apêndice A) possui a finalidade de validar o questionário proposto, a fim de eliminar problemas de entendimento com relação ao texto, de eliminar perguntas incoerentes ou repetidas e validar o instrumento de coleta. O pré-teste foi aplicado em três empresas de desenvolvimento de software, dessas empresas nenhuma apontou alterações no questionário. O tempo de resposta desse questionário variou de 2 a 5 minutos.

Para desenvolvimento do instrumento de coleta e da coleta de dados usou-se uma ferramenta web gratuita de formulários disponibilizada pela organização Google, para tratamento dos dados utilizou-se o *Microsoft Excel*®. A escolha de uma ferramenta eletrônica para coleta de dados agiliza a tabulação e preparo dos dados, já que a ferramenta atualiza os dados conforme as questões forem respondidas.

### 3.5 TRATAMENTO DOS DADOS

Os dados foram extraídos em forma de planilha do Google docs que já efetua um pré-tratamento das respostas. A ferramenta do Google adiciona as datas e horários de resposta em uma coluna paralela às respostas e cada questão com subquestões a ferramenta converte em um número correspondente de subquestões, por exemplo, a questão 9 (sobre a

importância dos produtos de acordo com a visão da empresa) contém 7 subquestões em escala intervalada, essa questão foi subdividida em 7 colunas.

Os dados foram tratados por meio do *Microsoft Excel*. Inicialmente a coluna de indicação de data e hora que o Google docs adicionou e a coluna com os nomes das empresas participantes foram retiradas para não identificar as empresas que não queria que seu nome fosse divulgado na pesquisa e porque os nomes foram utilizados para evitar que mais de um funcionário da mesma empresa respondesse o questionário encaminhado por isso não influenciariam na análise dos dados.

Das 76 empresas respondentes, apenas uma empresa não respondeu de forma satisfatória, essa empresa respondeu apenas as questões 3 e 4, sobre o uso dos modelos de qualidade e deixou todas as outras questões em branco, por esse motivo foi eliminada no tratamento dos dados.

O campo de porte empresarial foi agrupado de acordo com o porte das empresas, os campos referentes às questões 3 e 4 foram agrupados de acordo com o tipo de modelo escolhido e seus dados foram comparados com o porte da empresa para verificar se o uso (ou não) de modelos de qualidade se relaciona ao porte da empresa. A coluna referente à questão 5 foi cruzada com as colunas das questões 3 e 4 devido a sua dependência e com o porte da empresa para verificar se o porte está relacionado ao uso de determinado modelo de qualidade.

Os campos da questão 6, 7 e 8 foram relacionado aos campos das questões 3 e 4, devido à dependência desses campos, a questão 8 foi analisada campo a campo em função das questões 3 e 4, para verificar qual(is) modelo(s) foi(ram) utilizado(s) pela empresa e qual a visão dessa empresa sobre o impacto gerado pelo uso desse modelo(s).

As colunas da questão 9 e 10 foram agrupadas de acordo com as subquestões e cada subquestão foi cruzada por meio de um cálculo de correlação com as demais subquestões, para verificar a importância das características dos produtos e processos de software na visão das empresas respondentes. A questão 11, por ser descritiva optou-se por analisar individualmente cada resposta.

### 3.6 ANÁLISE DE DADOS

Na fase de análise dos dados do questionário foi utilizada estatística descritiva, recursos de visualização como gráficos de barras e correlação entre variáveis. Para facilitar a visualização dos dados, efetuar os cálculos e as análises as ferramentas, Microsoft Excel® e Draw.io foram usadas.

A análise descritiva possui a finalidade de descrever os dados que foram analisados por meio de distribuição de frequência e medidas de dispersão. Essa análise tem o objetivo de apresentar a visão dos respondentes com relação utilização e importância dos modelos de qualidade de software. A distribuição de frequência foi utilizada para descrever e analisar a distribuição das respostas ao questionário. As medidas de dispersão serão utilizadas para analisar a dispersão das respostas do questionário. Os gráficos serão utilizados para facilitar a interpretação dos resultados. As análises tiveram foco nos objetivos deste trabalho, portanto foram analisados os modelos de qualidade, com foco no produto, encontrados na literatura de acordo com os atributos de qualidade de software e os modelos voltados ao processo de software, com base na evolução da maturidade da empresa. Também, após as análises do questionário, foram comparados os resultados da literatura com os resultados obtidos por meio do questionário.

## 4 RESULTADOS E ANÁLISES

Nessa seção serão discutidos os resultados, obtidos por meio da pesquisa bibliográfica realizada para esta dissertação e por meio da pesquisa de campo.

### 4.1 RESULTADOS E ANALISES DO REFERENCIAL TEÓRICO

Essa pesquisa identificou três modelos de qualidade de software para avaliação do produto. O quadro 2 apresenta um comparativo entre os modelos McCall, FURPS+ e SQUARE, com base nos atributos para avaliação da qualidade de um produto de software presentes no modelo McCall (seção 2.5.1).

QUADRO 2 - COMPARATIVO ENTRE OS MODELOS MCCALL, FURPS+ E SQUARE, DOS ATRIBUTOS PARA QUALIDADE DE SOFTWARE.<sup>10</sup>

Nº	Atributos para qualidade do software	McCall	FURPS+	SQUARE
1	Usabilidade	Usabilidade	Usabilidade, requisitos de design e interface	Usabilidade (inteligibilidade, apreensibilidade, operacionalidade e atratividade)
2	Confiabilidade	Confiabilidade	Confiabilidade (exatidão, recuperação de falhas)	Confiabilidade (maturidade, tolerância, recuperação de falhas)
3	Manutenção	Manutenção	Suportabilidade (subatributo manutenção)	Manutenção (analísabilidade, modificabilidade, estabilidade e testabilidade)
4	Portabilidade	Portabilidade	Suportabilidade (subatributos compatibilidade,	Portabilidade (adaptabilidade, instalabilidade,

<sup>10</sup> Os significados dos atributos de qualidade de software estão explicados nos itens 2.5.1, 2.5.2 e 2.5.3 do referencial teórico.

			instalabilidade)	coexistência e substituibilidade)
5	Testabilidade	Testabilidade	Suportabilidade (subatributo testabilidade)	Manutenção (subatributo testabilidade)
6	Flexibilidade	Flexibilidade	Suportabilidade (adaptabilidade e configurabilidade)	Funcionalidade (subatributo adequação)
7	Eficiência	Eficiência	Desempenho, Requisitos de implementação e físicos	Eficiência (comportamento em relação ao tempo e utilização de recursos)
8	Integridade	Integridade	Funcionalidade (caso faça parte das especificações)	Funcionalidade (subatributo segurança)
9	Interoperabilidade	Interoperabilidade	Suportabilidade (comportabilidade e instabilidade)	Funcionalidade (subatributo interoperabilidade)
10	Exatidão	Exatidão	Confiabilidade (subatributo exatidão)	Acurácia (precisão)
11	Reusabilidade	Reusabilidade	-	-

FONTE:O AUTOR (2015)

O modelo McCall apresenta 11 atributos para qualidade de um software, enquanto o modelos FURPS+ apresenta 9 atributos para qualidade, e o modelo SQUARE apresenta 6 atributos, desses atributos somente a confiabilidade se torna um vínculo comum entre os três modelos, sendo que no modelo FURPS+ a confiabilidade se divide em exatidão e recuperação de falhas, no modelo SQUARE se divide em maturidade, tolerância e recuperação de falhas, o modelo McCall considera recuperação de falha e precisão como características de confiabilidade, mas não desdobra em subitens. Para o atributo confiabilidade, os três modelos são equivalentes.

O modelo SQUARE possui 4 subitens para usabilidade (inteligibilidade, apreensibilidade, operacionalidade e atratividade) enquanto o modelo FURPS+ possui o atributo de usabilidade e dois atributos que influenciam na usabilidade



(requisitos de design e interface). Para o desenvolvimento de software em que a usabilidade seja um fator de importância, pode-se considerar a utilização do modelo SQUARE associado aos atributos do modelo FURPS+ que influenciam na usabilidade.

O atributo manutenção, para o modelo FURPS+ é um subitem de suportabilidade, enquanto que no modelo McCall é considerado um atributo de qualidade e no modelo SQUARE é um atributo de qualidade com quatro subatributos (analisabilidade, modificabilidade, estabilidade e testabilidade). O modelo SQUARE é mais indicado para projetos que tenham como exigência a manutenção do software.

O atributo Portabilidade não existe para o modelo FURPS+ existem subatributos de suportabilidade, compatibilidade e instalabilidade que se relacionam com a portabilidade, enquanto que o modelo SQUARE considera adaptação, instalação, coexistência e substituição do software como subatributos de portabilidade. A utilização do modelo SQUARE é mais indicado para o desenvolvimento de software que necessitem de portabilidade.

O atributo testabilidade para o modelo FURPS+ é incluído como subatributo de suportabilidade, enquanto que para o modelo SQUARE é incluído como subatributo de manutenção. Ambos os modelos consideram o teste como um elemento pós-desenvolvimento do software. O modelo McCall considera a testabilidade um elemento que facilita a alteração do produto, é o modelo mais indicado para sistemas com prioridade em testabilidade e atualizações contínuas.

O atributo flexibilidade não existe para o modelo FURPS+, porém os subatributos adaptabilidade e configurabilidade estão associados à flexibilidade do software. O modelo SQUARE apresenta adequação como um atributo compatível com flexibilidade. Para o desenvolvimento de um software pode-se avaliar a flexibilidade, a configurabilidade e a adaptabilidade do sistema como um grupo de atributos.

O atributo eficiência é similar no modelo McCall e no modelo SQUARE, porém, no modelo FURPS+ se identifica o desempenho e os requisitos de implementação e físicos que interferem na eficiência do software e por esse fator se assemelha aos dois outros modelos. Um software de qualidade deve

considerar a eficiência do software, mas deve medi-la também por meio dos requisitos de implementação e físicos do software.

Com relação à integridade apenas o modelo McCall a considera como um atributo de qualidade, porém, foram identificados atributos dos outros modelos que complementam o valor de integridade, para o modelo FURPS+ se inclui na funcionalidade, caso seja um item exigido pelo cliente, no caso do modelo SQUARE se considera segurança um item complementar a integridade do software, por tanto, a avaliação de segurança deve considerar a integridade do software.

O atributo interoperabilidade esta presente no modelo SQUARE como um subatributo de funcionalidade e no modelo FURPS+ os subatributos comportamento e instalabilidade se relacionam ao atributo na forma como o software é instalado e como se comporta perante uma ação do usuário. Os três atributos podem ser trabalhados juntamente em um único projeto.

O atributo exatidão está associado à confiabilidade para o modelo FURPS+ e a acurácia para o modelo SQUARE, para os modelos de qualidade de software a qualidade do software está ligada diretamente a precisão e certeza das funções do software. Os três modelos apresentados consideram esse atributo importante.

O atributo reusabilidade presente no modelo McCall não possui atributo similar nos modelos FURPS+ e SQUARE. No ponto de vista dos dois modelos a reutilização de um software não é um atributo de qualidade e sim uma característica do software que depende dos requisitos de funcionalidade. Para o desenvolvimento de um sistema que necessite reutilização do software ou parte do software, o melhor modelo é o de McCall.

A utilização dos modelos de avaliação de software é útil para garantir que o software produzido obtenha garantia da qualidade, os três modelos apresentados no quadro 2 (McCall, FURPS+ e SQUARE) podem ser usados (integral e parcialmente) para avaliação da qualidade de software e podem ser usados em conjunto, de acordo com as necessidades da empresa, porém, o fato de uma empresa não utilizar modelos para avaliar a qualidade dos softwares produzidos, não significa que o produto não tenha qualidade, por esse motivo o instrumento de pesquisa (Apêndice B, questão 4) dessa

dissertação questiona a implementação de técnicas que controlem e avaliem o produto de software com base nos modelos estudados.

Este trabalho também está fundamentado em modelos de qualidade que se referem ao processo de desenvolvimento de software, esses modelos possuem níveis de maturidade que dimensionam a maturidade de uma empresa que desenvolve software. O modelo MPS.Br (ver seção 2.5.6) desenvolvido no Brasil, segundo Franciscani e Pestili (2012) possui equivalência com o modelo CMMI (ver seção 2.5.5), no seguinte formato: o nível 1 do CMMI que se refere ao nível inicial (processos caóticos e sem gerenciamento) não possui relação com nenhum dos níveis do MPS.Br, pois para que a certificação do primeiro nível do modelo brasileiro ocorra, a empresa deve ter os processos parcialmente gerenciados; o nível 2 (processos gerenciados; planejados, executados e monitorados de acordo com uma política empregada pela empresa) do CMMI equivale aos níveis G e F do MPS.Br; o nível 3 do CMMI (processos definidos e descritos por meio de padrões, procedimentos, ferramentas e métodos) se equivale aos níveis E, D e C do MPS.Br; Os níveis 4 e 5 do CMMI são totalmente equivalentes respectivamente aos níveis B e A do MPS.Br. A figura 2 apresenta essa equivalência.

O MPS.Br visa aderência às normas ISO 15504 (SPICE), além do CMMI, por essa razão o modelo brasileiro possui processos que não existem no modelo CMMI, são:

- nível F: processo de Gerência de Portfólio de Projetos;
- nível E: processo de Aquisição de Pessoal e Gerência de Conhecimento que não estão presentes na área de Treinamento Organizacional do CMMI;
- nível E: processo de Gerência de Reutilização;
- nível C: processo Desenvolvimento para Reutilização;

O MPS.Br é um modelo voltado para empresas de micro, pequeno e médio porte pois divide os processos iniciais do CMMI em várias partes, assim se torna mais fácil e econômico adquirir os processos exigidos para cada nível de maturidade pois como os processos são estendidos em sete níveis, a empresa pode concentrar investimentos em menos processos, por exemplo, para o nível 2 do CMMI são necessários sete processos, enquanto que para o

nível G do MPS.Br são necessários dois processos e para o nível F são necessários 4 processos. As normas ISO possuem o modelo SPICE (ver seção 2.5.4), com seis níveis equivalentes ao CMMI; O nível 0 do SPICE equivale ao nível 1 do CMMI; o nível 1 e 2 do SPICE equivalem ao nível 2 do CMMI; os níveis 3, 4 e 5 do SPICE são equivalentes respectivamente aos níveis 3, 4 e 5 do CMMI.

QUADRO 3 - EQUIVALÊNCIA ENTRE OS CINCO NÍVEIS DO MODELO CMMI E OS SETE NÍVEIS DO MODELO MPS.BR SEGUNDO FRANCISCANI; PESTILI (2012)

CMMI	MPS.Br
5 - Em Otimização - Análise Casual e Resolução, Inovação e melhoria organizacional.	A - Em Otimização - análise de causas de problemas e resolução.
4 - Quantitativamente Gerenciado - Desempenho do Proc. Organizacional.	B - Gerenciado Quantitativamente - gerência quantitativa do projeto.
3 - Definido - Foco no processo da organização, definição do proc. da organização, treinamento organizacional, gerência integrada de projeto, gerência de risco, desenvolvimento de requisitos, solução técnica, integração de produtos, verificação, validação e análise de decisão e resolução.	C - Definido - análise de decisão do projeto, gerência de riscos e desenvolvimento de reutilização  D - Largamente Definido - desenvolvimento de risco, integração do produto, projeto e construção do produto, verificação e validação.  E - Parcialmente Definido - gerência de recursos humanos, avaliação e melhoria do proc. org., definição do proc. organizacional e gerência de reutilização.
2 - Gerenciado - gerência de requisitos, planejamento de projeto, acompanhamento e controle de projeto, ger. de acordo com fornecedores, ger. qualidade de proc. e produto, gerência de configuração, medição e análise.	F - Gerenciado - medição, gerência de configuração, aquisição e garantia de qualidade.  G - Parcialmente Gerenciado - gerência de requisitos e gerência de projetos.
1 – Inicial.	

FONTE: FRANCISCANI; PESTILI (2012)

Segundo Baradel (2007 p. 24) o modelo SPICE é geralmente aplicado em organizações envolvidas com planejamento, gestão, monitoramento, controle e melhoria de processos de aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de software. O mesmo autor comenta que o SPICE é um modelo mais abrangente que o CMMI em função do uso de outros modelos como referência e por isso possui um nível de controle maior que o

CMMI com relação aos contratos comerciais, serviços de manutenção e instalação.

O modelo SPICE demonstra uma avaliação e melhoria contínua dos processos de software propondo autoconhecimento da empresa como parte do processo de melhoria e com a obrigação executar as práticas definidas no modelo. Na visão desse modelo, a preocupação com a melhoria dos processos e o comprometimento da empresa por consequência, melhoram a visão e satisfação do cliente em longo prazo, por parte da empresa e do produto desenvolvido pela empresa.

O modelo CMMI possui uma visão de mercado, por isso necessita de uma certificação para avaliar o nível de maturidade da empresa de software. Essa visão apresenta ao cliente a qualidade dos processos empregados pela desenvolvedora de software, o modelo CMMI é o mais utilizado no mundo em função da certificação que comprova a maturidade da empresa e por consequência a qualidade do produto.

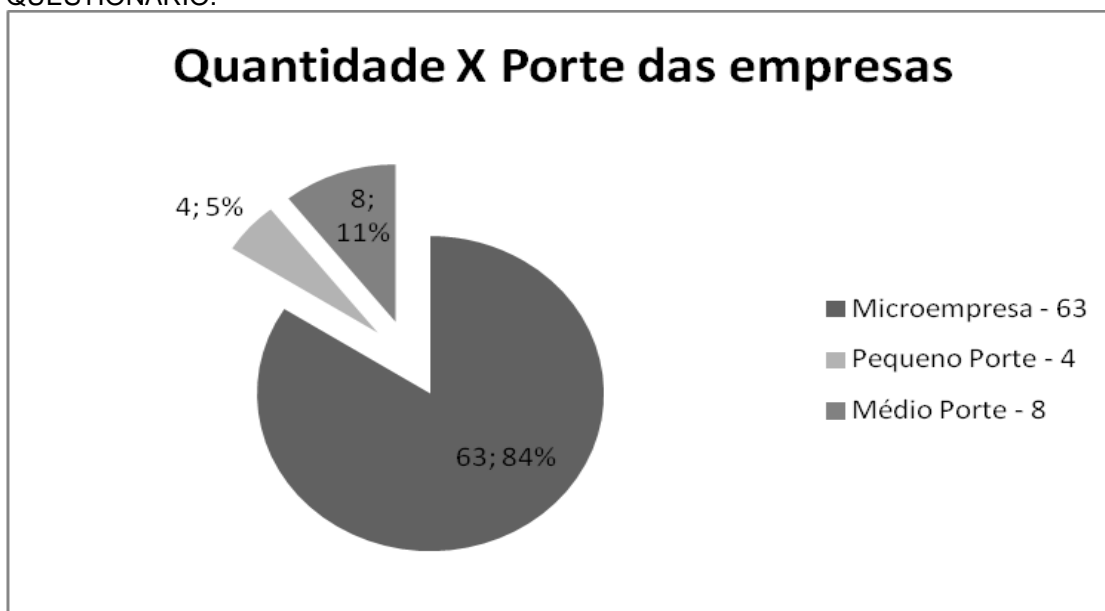
Existe um risco com relação a essa certificação, a certificação não comprova necessariamente que os processos são de qualidade, uma certificação CMMI nível 2 comprova que a empresa está se esforçando para melhorar os processos, mas ainda possuem falhas e os processos ainda não estão definidos e não são medidos de forma quantitativa. Nesse caso a certificação está comprovando a incapacidade da empresa, essa visão normalmente não é observada por empresas de desenvolvimento de software que buscam certificação, mas para absorver mais clientes.

## 4.2 RESULTADOS E ANALISES DO INSTRUMENTO DE PESQUISA

Com relação aos dados obtidos por meio do instrumento de pesquisa, foram contatadas 330 empresas em Curitiba, no período de 03 de novembro de 2014 a 30 de janeiro de 2015, destas, 133 empresas concordaram em participar da pesquisa, porém, apenas 76 empresas responderam ao questionário durante o período da pesquisa de campo, das 76 empresas respondentes 1 não respondeu de forma satisfatória o questionário e foi

excluída dos resultados, 63 empresas são microempresas, 4 de pequeno porte e 8 de porte médio, não houve respondentes de empresas de médio-grande porte e de grande porte. O total de respondentes é apresentado no gráfico 1.

GRÁFICO 1 - QUANTIDADE X PORTE DAS EMPRESAS QUE RESPONDERAM O QUESTIONÁRIO.



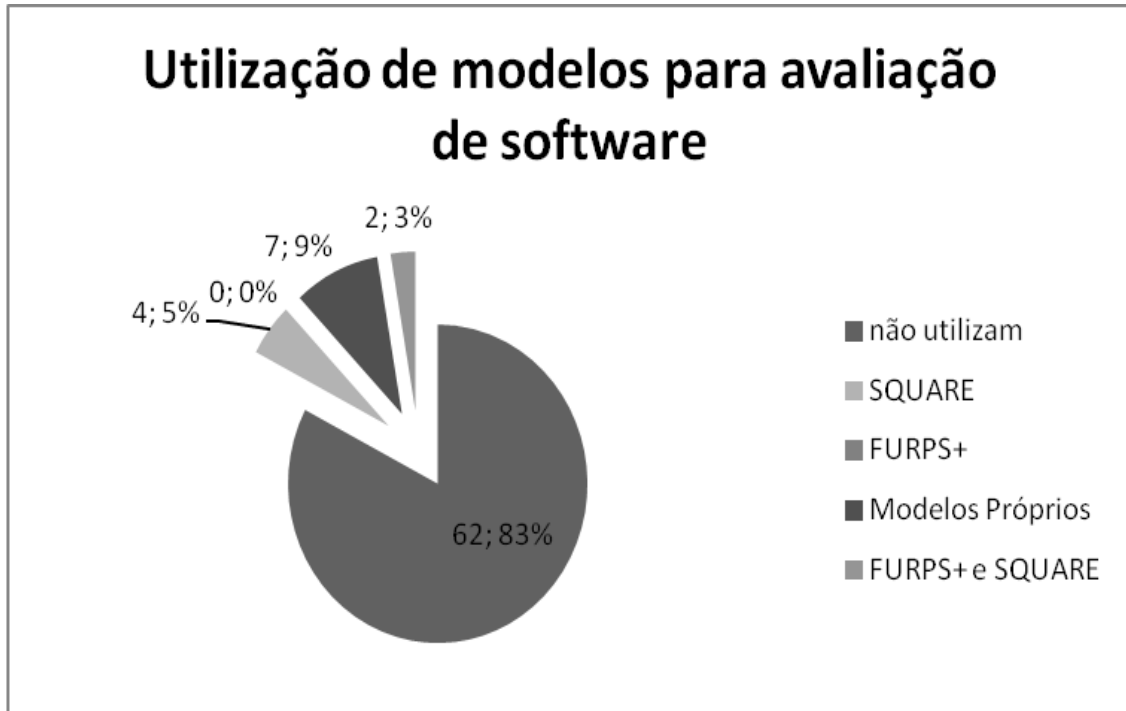
FONTE: O AUTOR (2015)

As duas primeiras perguntas do questionário foram utilizadas para classificar a empresa de acordo com o porte e para evitar que funcionários da mesma empresa respondessem o questionário. A terceira pergunta do questionário se refere ao uso de modelos de qualidade para avaliação do software. A pergunta é necessária para identificar se as empresas utilizam os modelos de qualidade para avaliação de software apresentados neste documento ou se utilizam outro modelo não listado. Com essa pergunta se esperava que houvesse uma quantidade acima de 10% de empresas que utilizassem um ou mais modelos e neste caso a porcentagem foi de 17% (somatório de todas as respostas afirmativas quanto ao uso de modelos de qualidade para avaliação de software), 83% das empresas responderam que não utilizam, enquanto 5% responderam que utilizam modelo SQUARE, 3% responderam que utilizam modelo FURPS+ e SQUARE e 8% possuem modelos próprios de qualidade. O gráfico 2 apresenta os resultados.

Das microempresas respondentes 54 não usam modelos, 4 utilizam modelo SQUARE, 2 utilizam FURPS+ e SQUARE e 3 utilizam modelos

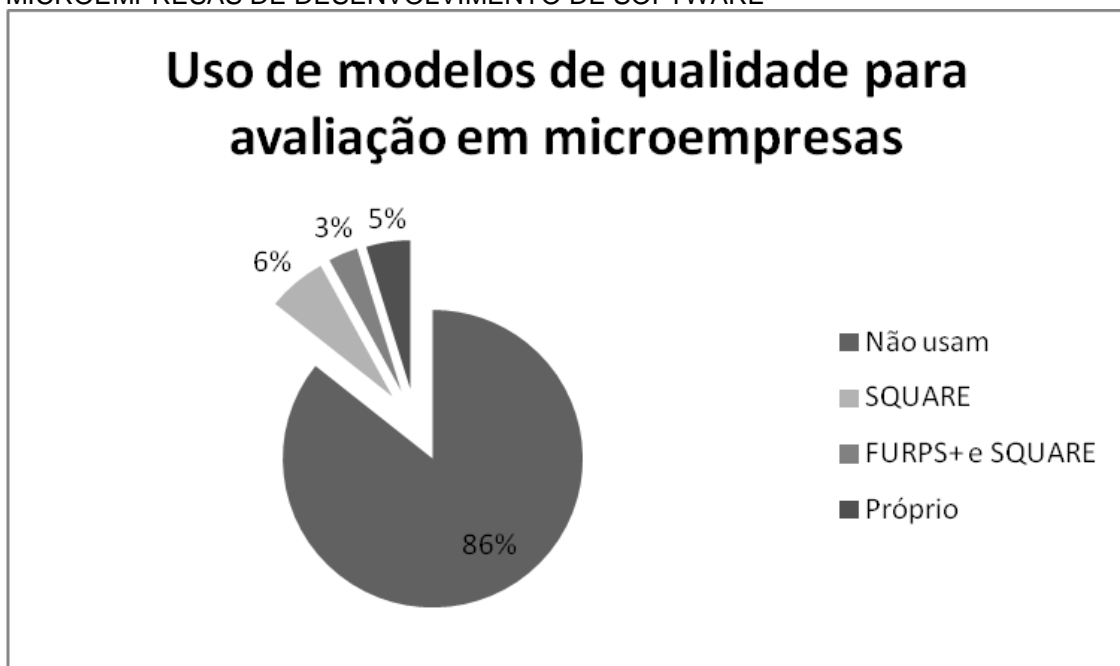
desenvolvidos com base nas experiências da empresa. O gráfico 3 apresenta os resultados.

GRÁFICO 2 - UTILIZAÇÃO DE MODELOS PARA AVALIAÇÃO DE SOFTWARE EM EMPRESAS DE DESENVOLVIMENTO EM CURITIBA



FONTE: O AUTOR (2015)

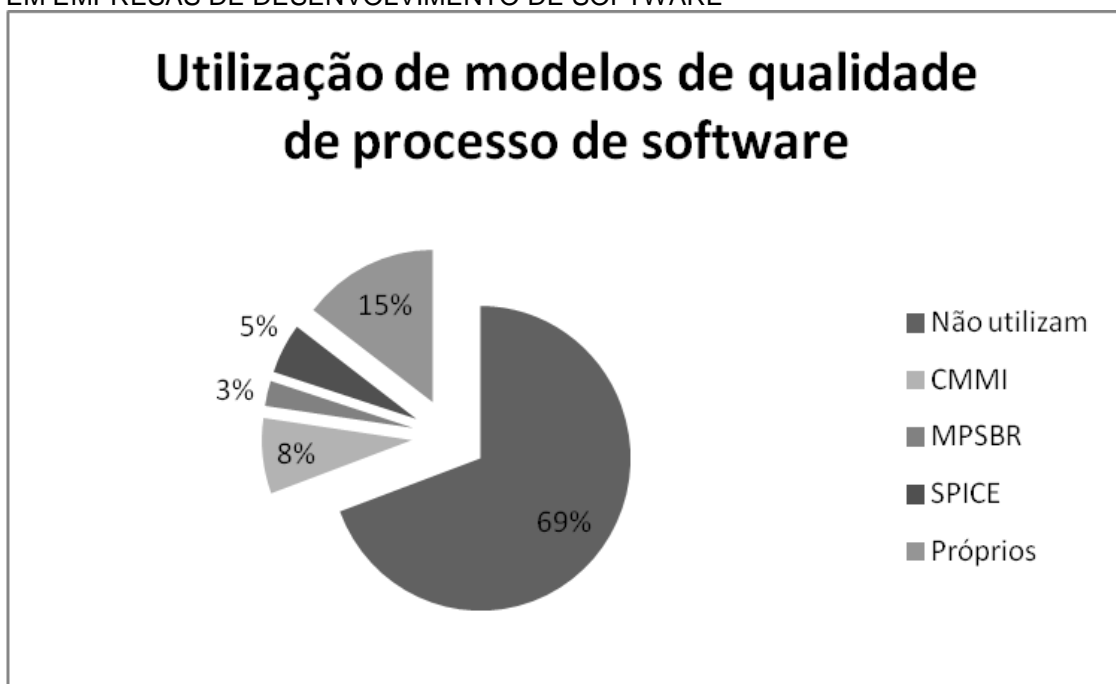
GRÁFICO 3 - USO DE MODELOS DE QUALIDADE DE SOFTWARE PARA AVALIAÇÃO EM MICROEMPRESAS DE DESENVOLVIMENTO DE SOFTWARE



FONTE: O AUTOR (2015)

As quatro empresas de pequeno porte não utilizam modelos de qualidade, enquanto que das 8 empresas de médio porte respondentes, 2 empresas não utilizam modelo e 6 utilizam modelos baseados na experiência da empresa. Esse resultado demonstrou que as empresas de desenvolvimento de software estão preocupadas com a qualidade de seus produtos e um número significativo de empresas utiliza avaliação dos seus produtos com base em modelos e experiências da empresa.

GRÁFICO 4 - UTILIZAÇÃO DE MODELOS DE QUALIDADE DE PROCESSO DE SOFTWARE EM EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE



FONTE: O AUTOR (2015)

A quarta pergunta do questionário se refere ao uso de modelos de qualidade para processos de software. A pergunta é necessária para identificar se as empresas utilizam os modelos de qualidade de processo apresentados neste documento ou se utilizam outro modelo não listado. Com essa pergunta se esperava que houvesse uma quantidade acima de 10% de empresas que utilizassem um ou mais modelos e neste caso a porcentagem foi de 31% (somatório de todas as respostas afirmativas quanto ao uso de modelos de qualidade para avaliação de software), 69% das empresas responderam que não utilizam modelos de qualidade de processo de software, 15% responderam que utilizam modelos próprios baseados na experiência da empresa e em

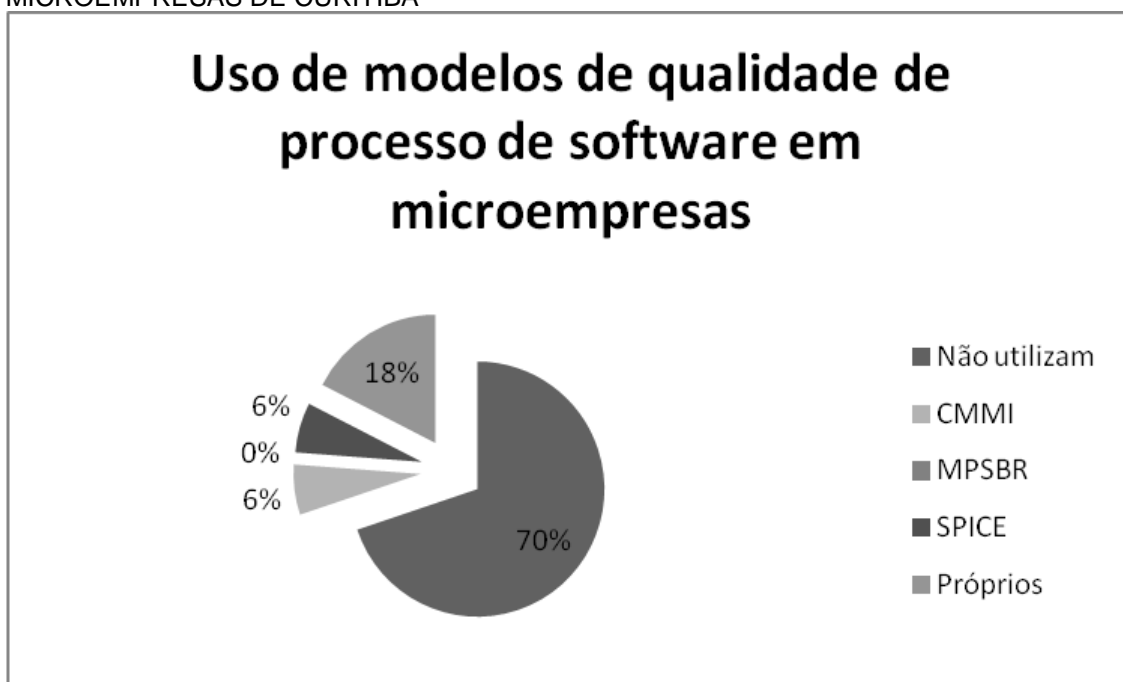


outros modelos como CMMI e MPS.Br, 8% utilizam CMMI, 5% utilizam SPICE e 3% utilizam o modelo MPS.Br. O gráfico 4 apresenta os resultados.

Das 63 microempresas respondentes 44 não usam modelos, 4 utilizam modelo CMMI, 4 utilizam SPICE e 11 utilizam modelos desenvolvidos com base nas experiências da empresa. O gráfico 5 apresenta os resultados.

Esse resultado demonstrou que as empresas de desenvolvimento de software estão preocupadas com a qualidade de seus processos e um número significativo de empresas utiliza modelos de qualidade de processos de software com base em modelos e experiências da empresa.

GRÁFICO 5 - USO DE MODELOS DE QUALIDADE DE PROCESSO DE SOFTWARE EM MICROEMPRESAS DE CURITIBA



FONTE: O AUTOR (2015)

Para as empresas que não utilizam modelos de qualidade de software para avaliação e para processos, se questionou se houve tentativa de implantação de algum modelo de qualidade. Essa questão é necessária para identificar se a empresa já conhece ou tentou implantar um determinado modelo e quais foram as dificuldades para a implantação desse modelo. Esperava-se que uma parcela de 10 % das empresas já estivesse em um estágio inicial de utilização de um modelo de software. Em 37% dos casos, houve ou está iniciando o processo de implementação de algum modelo

(CMMI-DEV e MPS.Br-SW). Esse dado demonstra que as empresas de desenvolvimento tem a preocupação em melhorar os processos de desenvolvimento e a avaliação dos produtos, porém, ainda estão buscando o melhor modelo e o processo de implantação pode se tornar demorado em função dos investimentos necessários, seja econômico ou de pessoal.

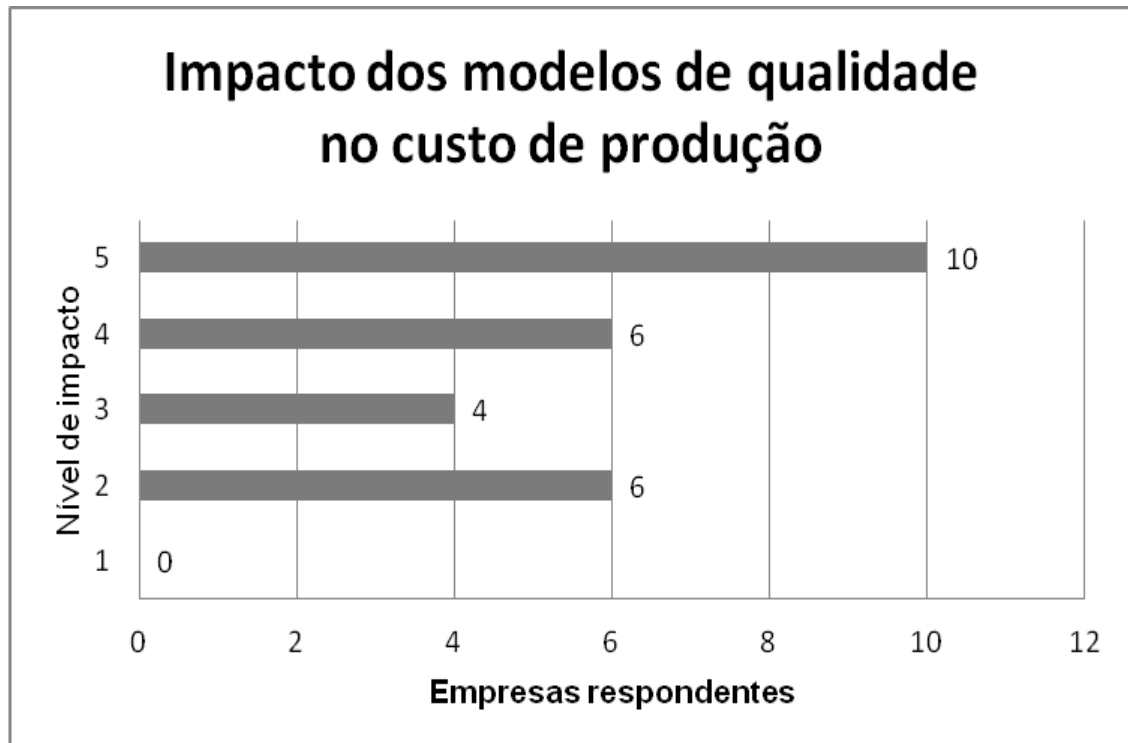
Para as empresas que adotam modelos de qualidade ou que iniciaram o processo de implantação se perguntou qual o impacto da utilização do modelo de qualidade no custo e tempo de produção, desempenho, retrabalho, satisfação do cliente e manutenção do software em escala intervalada (1-5 sendo que 1 representa o grau mais baixo de impacto e 5 representa o grau mais alto de impacto). Essa questão é necessária para identificar se o uso dos modelos de qualidade para avaliação e de processo produzem impacto positivo ou negativo no custo e tempo de produção, desempenho, retrabalho, satisfação do cliente e manutenção do software. Nessa questão se obteve 26 respostas, 3 de empresas que utilizavam apenas modelos de qualidade para processo, 13 empresas que utilizavam apenas modelos de qualidade de avaliação de software e 10 empresas que utilizavam os dois tipos de modelos. Quanto ao custo de produção, das 26 empresas respondentes 10 empresas consideram um grau máximo de custo de produto, 6 consideram um impacto de grau 4, porém 6 empresas consideram um impacto 2 (gráfico 6). Esse resultado mostra que a utilização de modelos de qualidade de software tem influencia nos custos de produção como citado por Santos e Weber (2013):

O principal objetivo do programa MPS.BR (...) foi desenvolver e disseminar um modelo de processos de software visando a estabelecer um caminho economicamente viável para que organizações, incluindo as Pequenas e Médias Empresas (PME), alcancem os benefícios da melhoria de processos e da utilização de boas práticas da engenharia de software em um intervalo de tempo justo a um custo razoável. (SANTOS E WEBER, 2013, p. 5)

O uso dos modelos de avaliação e de processo reflete um alto custo em curto prazo, pois são necessários treinamentos, consultorias, contratação de profissionais especializados e certificados, o controle da qualidade do produto mediante avaliação produz um custo direto sobre o produto. Porém em longo prazo, o controle e prevenção dos erros reduzem a necessidade de retrabalho

da equipe e a padronização dos processos reduz o trabalho e o tempo de desenvolvimento sem reduzir a qualidade do produto.

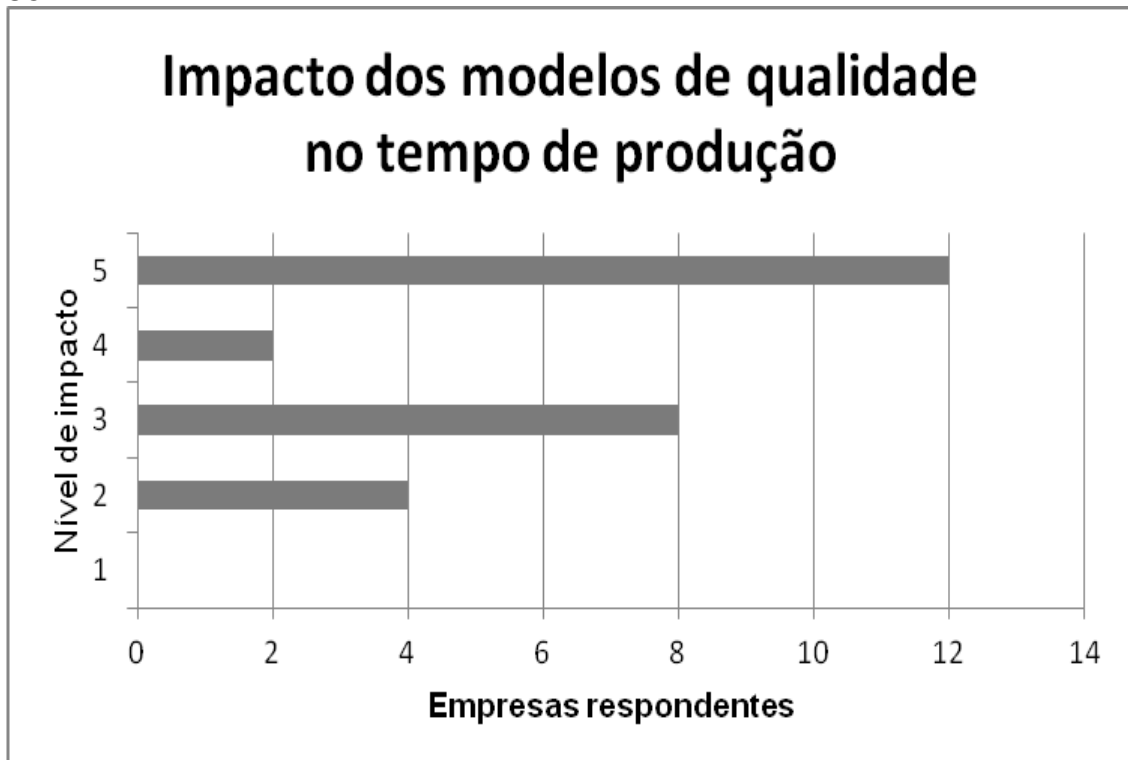
GRÁFICO 6 – IMPACTO DOS MODELOS DE QUALIDADE NO CUSTO DE PRODUÇÃO DE SOFTWARE



FONTE: O AUTOR (2015)

. A visão de ganho com relação ao tempo, descrito por Santos e Weber (2013) no trecho anterior pode ser verificado no gráfico 7, com relação ao tempo de produção das 26 empresas respondentes 12 consideram um impacto de grau 5 no tempo de produção, enquanto quatro empresas consideram um impacto de grau 2 e 8 empresas consideram um impacto de grau 3, nenhuma das empresas considera que o grau de impacto mínimo.

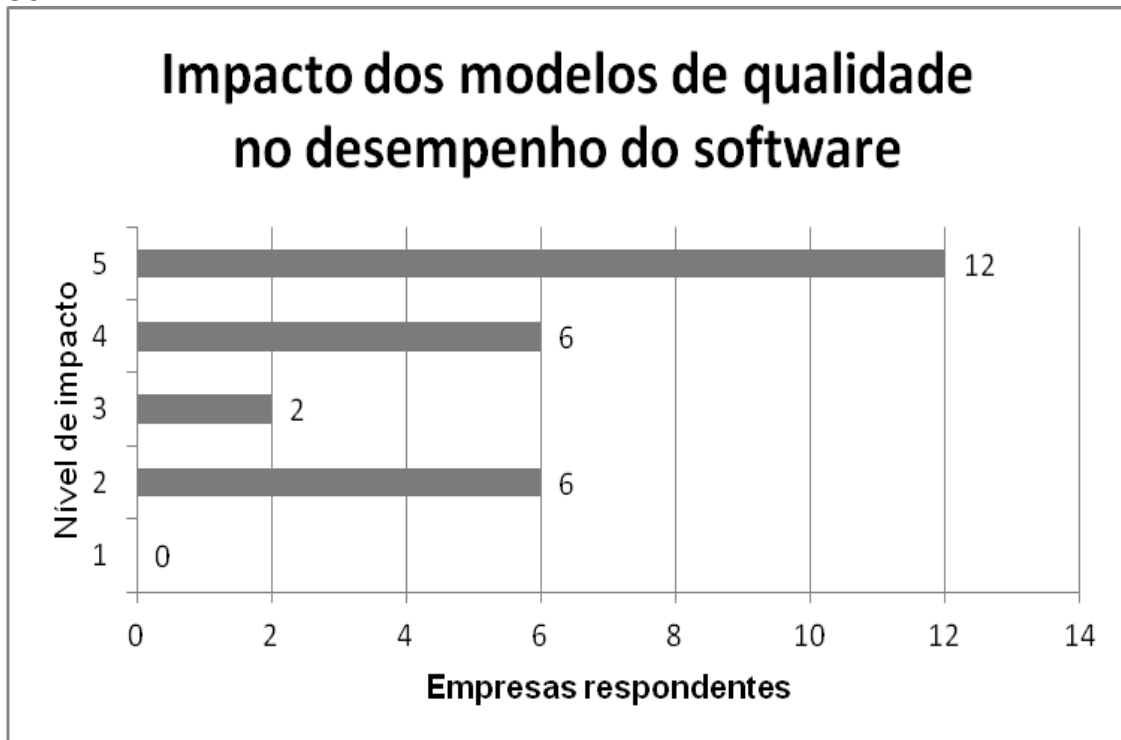
GRÁFICO 7 - IMPACTO DOS MODELOS DE QUALIDADE NO TEMPO DE PRODUÇÃO DE SOFTWARE



FONTE: O AUTOR (2015)

Com relação ao desempenho do software das 26 empresas respondentes 12 consideram um impacto de grau 5 no desempenho do software, enquanto 6 empresas consideram impacto 2 e 6 empresas consideram um impacto 4, nenhuma das empresas considera que o grau de impacto seja muito pouco ou nenhum. De modo geral os respondentes consideram um impacto do uso dos modelos de qualidade no desempenho do software. O gráfico 8 apresenta o resultado. Segundo os modelos McCall, FURPS+ e SQUARE, o desempenho é um atributo fundamental para avaliação da qualidade do software.

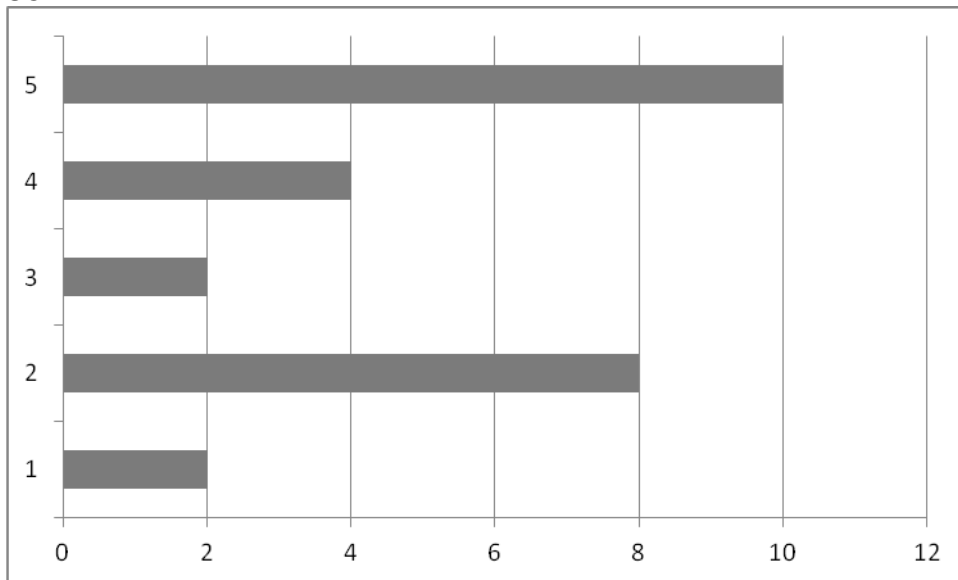
GRÁFICO 8 - IMPACTO DOS MODELOS DE QUALIDADE NO DESEMPENHO DO SOFTWARE



FONTE: O AUTOR (2015)

Com relação ao retrabalho da equipe, das 26 empresas respondentes 10 consideram um impacto de grau 5 no desempenho do software, 6 empresas consideram impacto de grau 4 e 8 empresas consideram um impacto de grau 2, duas das empresas considera o impacto de grau 1. De modo geral os respondentes consideram um impacto da qualidade no retrabalho da equipe de desenvolvimento, definir os processos organizacionais (nível E do MPS.Br; nível 2 do CMMI; e nível 2 do SPICE), gerência de recursos e de reutilização (nível E do MPS.Br) pode evitar o retrabalho da equipe de desenvolvimento, o gráfico 9 apresenta o resultado.

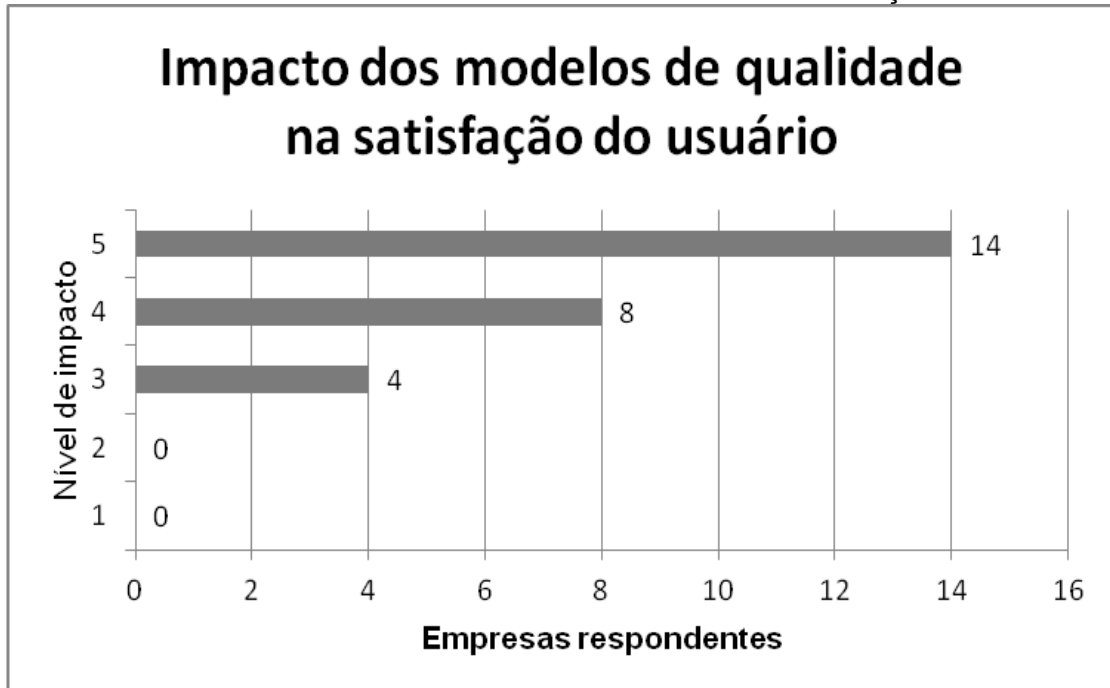
GRÁFICO 9 - IMPACTO DOS MODELOS DE QUALIDADE NO RETRABALHO DA EQUIPE DE SOFTWARE



FONTE: O AUTOR (2015)

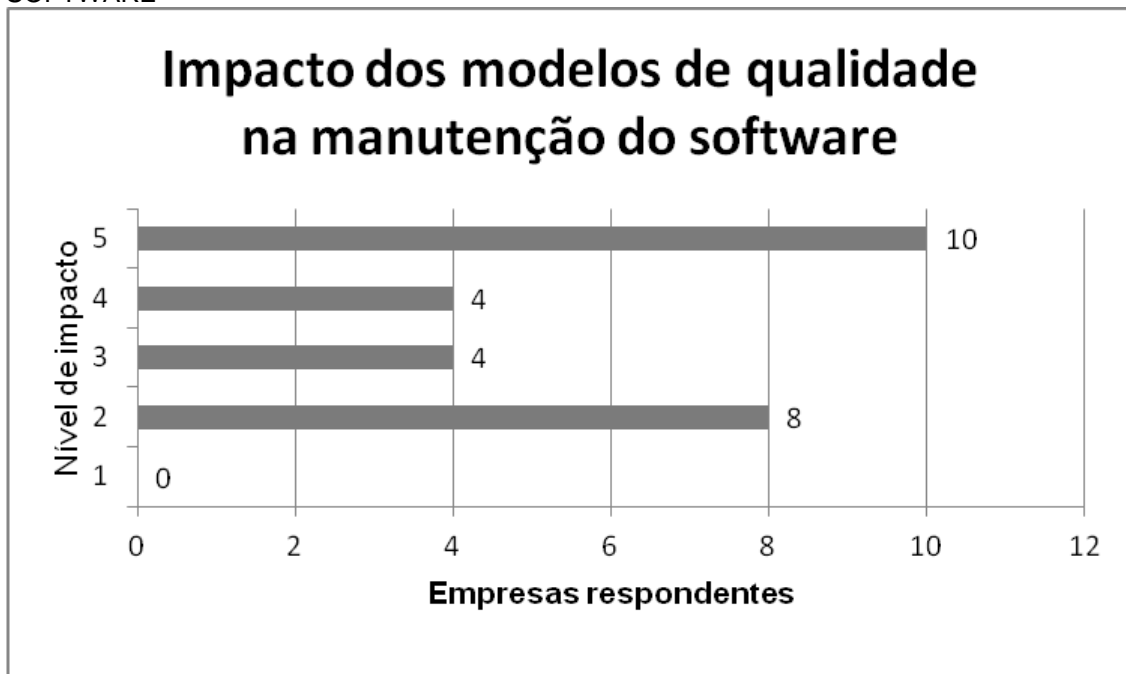
Para satisfação do usuário, 14 empresas consideram um impacto de grau 5 no desempenho do software, 8 empresas consideram impacto de grau 4, 4 empresas consideram um impacto de grau 3. Todos os respondentes consideram um impacto do uso dos modelos de qualidade na satisfação do usuário. Santos e Weber (2013) comentam que é necessário relacionar indicadores de desempenho à medidas de satisfação do cliente, o modelo SPICE relaciona a importância dos processos que afetam o cliente como dimensões do processo de melhoria de software. Os modelos de avaliação de software controlam métricas de usabilidade, funcionalidade e eficiência que refletem diretamente na satisfação do usuário. O gráfico 10 apresenta o resultado.

GRÁFICO 10 - IMPACTO DOS MODELOS DE QUALIDADE NA SATISFAÇÃO DO USUÁRIO



FONTE: O AUTOR (2015)

GRÁFICO 11 - IMPACTO DOS MODELOS DE QUALIDADE NA MANUTENÇÃO DO SOFTWARE



FONTE: O AUTOR (2015)

Com relação manutenção do software, 10 respondentes consideram um impacto de grau 5, quatro empresas consideram impacto de grau 4, quatro empresas consideram um impacto de grau 3 e oito empresas consideram que impacto de grau 2. De modo geral os respondentes consideram que existe impacto dos modelos de qualidade na manutenção do software, os modelos

McCall, FURPS+ e SQUARE, confirmam essa afirmação, pois os três modelos apresentam manutenção como atributo da qualidade do software. O gráfico 6 apresenta o resultado.

Na pesquisa se questionou o entendimento da empresa sobre a importância das características do software (Apêndice B - questão 9) em escala intervalada (1-5 sendo que 1 representa a menor importância e 5 representa a maior importância), sob sete critérios: [1] funcionalidade, [2] confiabilidade, [3] usabilidade, [4] eficiência, [5] manutenibilidade, [6] portabilidade e [7] limitações de interface, implementação, física e/ou de design. Essa questão não era dependente de outras questões, por isso todas as empresas poderiam responder, 69 empresas responderam a essa pergunta. Porém analisando as respostas se percebeu uma possibilidade de duplicidade nas respostas, pois “na opinião da empresa” pode variar de acordo com o projeto de software e com as necessidades e requisitos do cliente e usuário por esse fator a questão 9 foi eliminada dos resultados. A questão 10 também repete a mesma duplicidade e também foi retirada dos resultados.

O questionário continha uma resposta descritiva, em que o respondente poderia expressar sua opinião sobre os modelos de qualidade. De modo geral, as respostas expressaram a necessidade de implementar um modelo de qualidade como “seria importante implementar”, “indispensável”, “são de extrema importância para melhorar a qualidade de software”, porém, um respondente alega que o uso de um modelo em desenvolvimento poderia aumentar a burocracia e reduzir a eficiência no desenvolvimento e manutenção de software. Outro respondente considera que a evolução do mercado de desenvolvimento faz com que os processos atuais de modelos de qualidade se tornem defasados.

As análises dos resultados obtidos com a pesquisa demonstraram que existe uma gama de modelos que podem ser utilizados para melhoria da qualidade do software e da empresa, de acordo com as necessidades da empresa que desenvolve software, porém, uma porcentagem alta de empresas não implanta nenhum modelo de qualidade. Algumas empresas desenvolvem o seu próprio modelo baseado em experiências adquiridas com o tempo ou adaptado de modelos já existentes. De um modo geral as empresas se



preocupam com os atributos de qualidade do software e com os processos que resultam em um sistema computacional.

## 5 CONSIDERAÇÕES FINAIS

Essa seção apresenta uma síntese do estudo realizado, descrevendo os últimos comentários referentes ao trabalho, retomando os objetivos propostos, que nortearam o desenvolvimento deste trabalho e as principais contribuições desta pesquisa.

A pesquisa identificou na literatura três modelos de qualidade de software para avaliação do produto (McCall, FURPS+ e SQUARE), os modelos foram comparados entre si e se constatou que os três modelos possuem atributos semelhantes, como usabilidade, confiabilidade e manutenção, porém, cada modelo agrupava esses atributos de forma diferente, tornando melhor que os outros dois para determinadas necessidades, por exemplo, o modelo McCall possui um atributo chamado reusabilidade, que não é compatível com nenhum atributo dos outros modelos estudados, por essa razão esse modelo pode ser utilizado em projetos de software que possuam a necessidade de desenvolver sistemas que possam ser usados parcial ou integralmente em outros sistemas.

O modelo FURPS+ possui um atributo para usabilidade como os outros dois, porém, esse modelo considera mais dois atributos que influenciam na usabilidade, que são os requisitos de design e interface. Por esse fator o modelo FURPS+ é mais indicado para projetos em que a usabilidade tenha um foco maior que os outros atributos de qualidade, como software para deficientes visuais.

O modelo SQUARE é indicado para avaliação de software que necessitem de cuidados quanto às funcionalidades do software, pois esse modelo possui quatro dimensões (adequação, acurácia, interoperabilidade e segurança) relacionadas as funcionalidades. Os três modelos podem ser usados em conjunto ou de acordo com as necessidades da empresa.

A pesquisa também identificou três modelos de melhoria de processo de software (CMMI, SPICE e MPS.Br). O modelo CMMI é conhecido e utilizado no mundo todo, com mais de 20 mil empresas certificadas possui cinco níveis de maturidade, sendo que o primeiro nível é considerado como caos e o último é o

nível da melhoria contínua, muitas empresas buscam o certificado CMMI para adquirir mais clientes, porém a certificação do modelo não garante a qualidade da empresa e sim a melhoria contínua dos processos. O modelo SPICE é mais utilizado na Europa e não necessita certificação como os outros dois modelos. Possui uma visão de melhoria contínua e em longo prazo e considera que com o esforço e melhoria dos processos, ao longo do tempo o cliente perceberá a melhoria. O modelo MPS.Br é um modelo brasileiro compatível com os outros dois, .

Durante o processo de pesquisa se desenvolveu um instrumento de coleta de dados com o objetivo de identificar o uso de modelos de avaliação e de melhoria de processos em desenvolvedoras de software em Curitiba,

Os resultados obtidos por meio do questionário foram formatados e analisados, das 75 empresas respondentes apenas 26 delas utilizam modelos de qualidade de software. As empresas que utilizam modelos de maturidade (independente do modelo) avaliaram o impacto do uso desses modelos como grau 5 (em um intervalo de 1-5) em seis critérios (custo, tempo de produção, desempenho, retrabalho, satisfação do cliente e manutenção do software), porém, na maior parte das empresa que utiliza modelos de qualidade utiliza modelos desenvolvidos na própria empresa com base em experiências vividas na empresa.

O impacto do uso dos modelos é positivo e negativo em dois estágios da melhoria dos processos, no primeiro momento o processo de melhoria se torna caro e dispendioso, com gastos com treinamento e na mudança cultural da empresa, em um segundo estágio, depois da consolidação do uso dos modelos, a empresa passa a ter economia, com menos erros e gastos desnecessários. Por meio dos resultados desta pesquisa pode se concluir que existem modelos compatíveis com as necessidades das empresas desenvolvedoras, mas em função do custo as empresas adotam modelos próprios de maturidade e de avaliação de software.

O desenvolvimento do trabalho encontrou limitações e dificuldades quando aos participantes da pesquisa de campo, das 330 empresas contatadas, apenas 133 aceitaram responder ao questionário e dessas apenas 75 empresas responderam de forma válida no período proposto para a pesquisa.

Como sugestão para trabalhos futuros se destaca a adoção de entrevistas com equipes de desenvolvimento para aprofundar a pesquisa de forma a identificar o uso dos modelos de qualidade de software desenvolvidos nas empresas de software.

## REFERÊNCIAS

ABNT. **ABNT NBR ISO 9000**. Rio de Janeiro: ABNT, 2005. p. 35

ABNT. **ABNT NBR ISO/IEC 25000:2008**. [s.l.] ABNT, 2008. p. 42

ANACLETO, Alessandra. VON WANGENHEIM, Christiane. **Método e Modelo de Avaliação para Melhoria de Processos de Software em Micro e Pequenas Empresas**. Simpósio Brasileiro de Qualidade de Software, Porto Alegre, Brazil, 2005.

ALMEIDA, Kenner Paulo de Moraes. **Avaliação da qualidade de software ERP de acordo com a norma ISO/ICE 9126**. 2010. 100 f. Monografia (Ciência da Computação) – Universidade Federal de Lavras, Lavras, MG, 2010.

ANJOS JUNIOR, E. R. DOS. **Modelo McCall**. Disponível em: <<http://engenhariadesoftwares.blogspot.com.br/2009/05/em-1977-mccall-propos-um-modelo-para.html>>. Acesso em: 29 set. 2014.

ARCHER, Renato. **Metodologia de Avaliação da CERTICS para Software**. CTI Renato Archer. 2013.

ARTHUR, L. J. **Melhorando a qualidade do software: Um guia completo para o TQM**. Rio de Janeiro: Infobook, 1994.

BARÇANTE, L. C. **Qualidade Total: uma visão brasileira**. [s.l: s.n.]. p. 62

BARADEL, Nicolas dos Santos. **CMMI e SPICE abordagem e estudo comparativo**. 2007. 41 f. Monografia (Ciência da Computação) – Faculdade de Jaguariúna, Jaguariúna, SP, 2007.

BERWANGER, E. **Apostila Engenharia de Software**, 2011.

BRAGA, E. **Metrologia**. Disponível em: <<http://goo.gl/t5xrVI>>. Acesso em: 4 out. 2014.

CALAZANS, A. T. S. Qualidade da informação: conceitos e aplicações. **TransInformação**, v. 20, n. 1, p. 29–45, 2008.

CARVALHO, G. M. R. DE; TAVARES, M. DA S. **Informação e Conhecimento: Uma abordagem organizacional**. Rio de Janeiro: Qualitymark, 2001. p. 152

CARVALHO, M. M. História da Gestão da Qualidade. In: **Gestão da qualidade: teorias e casos**. 2. ed. São Paulo: Elsevier Ltd, 2012.

CONCLA. **CNAE 2.0**. Disponível em: <<http://www.cnae.ibge.gov.br/>>. Acesso em: 27 jul. 2014.

CROSBY, P. B. **Quality is Free: The art of Making Quality Certain**. New York: McGraw-Hill, 1979.

DAVENPORT, T. H.; PRUSAK, L. **Conhecimento Empresarial: Como as organizações gerenciam o seu capital intelectual**. Rio de Janeiro: Campus, 1998. p. 237

DICIO, D. O. DE P. **Confiabilidade**. Disponível em: <<http://www.dicio.com.br/conformidade/>>. Acesso em: 4 out. 2014a.

DICIO, D. O. DE P. **Conformidade**. Disponível em: <<http://www.dicio.com.br/conformidade/>>. Acesso em: 4 out. 2014b.

DICIO, D. O. DE P. **Especificação**. Disponível em: <<http://www.dicio.com.br/especificacao/>>. Acesso em: 4 out. 2014c.

DINGSOYR. An Evaluation of Research on Experience Factory. **Workshop on Learning Software Organisations at the international conference on Product-Focused Software Process Improvement**, p. 55–66, 2000.

DOS ANJOS, Lúcio André Mendonça. Moura, Hermano Perrelli. **Um Modelo para Avaliação de Produtos de Software**. UFPE. Recife. 2004.

FEIGENBAUM, A. **Controle de Qualidade Total**. 3. ed. São Paulo: Makron, 1994.

FLEMING, I. **McCall's Quality Model - 1977**. Disponível em: <<http://www.sqa.net/index.htm>>. Acesso em: 29 set. 2014.

FNQ. **Sistemas de Gestão**. Disponível em: <<http://www.fnq.org.br/informe-se/noticias/faca-o-download-do-e-book-sobre-sistemas-de-gestao>>. Acesso em: 4 out. 2014.

GARVIN, D. **Gerenciando a qualidade – a visão estratégica e competitiva**. 3. ed. Rio de Janeiro: Qualitymark, 2002.

HILSDORF, C. **O que é benchmarking**. Disponível em: <<http://www.administradores.com.br/artigos/negocios/o-que-e-benchmarking/48104/>>. Acesso em: 6 out. 2014.

INMETRO. **Metrologia Científica**. Disponível em: <<http://www.inmetro.gov.br/metcientifica/>>. Acesso em: 4 out. 2014.

JURAN, J. M. **Juran's quality handbook**. 5. ed. New York: McGraw-Hill, 1998.

KERZNER, H. **Strategic Planning for Project Management Using a Project Management Maturity Model**. [s.l.] John Wiley e Sons, 2001. p. 107

MALDONADO, M. Engenharia do conhecimento aplicado ao serviço de atendimento ao cliente na indústria do software. **Revista chilena de ingeniería**, v. 18, n. 1, p. 53–63, 2010.

MARCONI, M. DE A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

MAYO, E. **The Human Problems of an Industrial Civilization**. Salem: Ayer Company, 1992. p. 194

MIGUEL, P. A. C. Gestão da Qualidade: TQM e Modelos de Excelência. In: **Gestão da qualidade: teorias e casos**2. Rio de Janeiro: Campus, 2006. p. 355.

MTE. **Números de estabelecimentos, segundo a classe da CNAE 2.0 - Paraná e Município de Curitiba - 2006-2012**. Disponível em: <<http://portal.mte.gov.br/portal-pdet/>>. Acesso em: 8 jul. 2014.

NBR ISO/IEC 9126. **NBR ISO/IEC 9126 Engenharia de software - Qualidade de produto**Rio de JaneiroABNT, , 2003.

OHNO, T. **O sistema Toyota de Produção: além da produção em larga escala**. [s.l.] Bookman, 1997. p. 147 p.

OLIVEIRA, Sandro Ronaldo Bezerra. **Qualidade de Processo de Software Normas ISO 12207 e 15504**.Universidade Federal do Pará, Pará, 2009.

PALADINI, E. P. Perspectiva Estratégica da qualidade. In: **Gestão da qualidade: teorias e casos**. 2. ed. São Paulo: Elsevier Ltd, 2012.

PEREIRA, R. DE C. DE F.; BELLINI, C. G. P.; LUCE, F. B. Namoro ou Amizade? A visão de clientes e fornecedores sobre relacionamentos de negócios no setor de software. **Revista Eletronica de Sistemas de Informação**, v. 9, n. 2, 2010.

PRESSMAN, R. **Engenharia de Software**. [s.l: s.n.]. v. 6 edp. 711

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2. ed. Novo Hamburgo, RS: FEEVALE, 2013.

QUALIDADEBR. **FURPS+**. Disponível em: <<http://qualidadebr.wordpress.com/2008/07/10/furps/>>. Acesso em: 28 set. 2014.

REZENDE, D. A. **Engenharia de Software Empresarial**. Rio de Janeiro: Brasport, 1997. p. 192

SANTOS, Gleison. WEBER, Kival Chaves. **Lições Aprendidas na Gestão do Programa MPS.BR**. Softex. 2013. p. 5

SEI. **CMMI® for Development, Version 1.3: CMMI-DEV, V1.3.** Pittsburgh: Carnegie Mellon University, 2010. p. 482

SHEWHART, W. A. **Statistical Method from the Viewpoint of Quality Control.** New York: Dover Publications, 1986.

SOFTEX. **MPS.BR - Melhoria de Processo do Software Brasileiro: Guia Geral MPS de Software.** [s.l.] SOFTEX, 2012. p. 59

SOMMERVILLE, I. **Engenharia de Software.** 6. ed. São Paulo: Addison Wesley, 2003.

STRAUHS, F. DO R. et al. **Gestão do conhecimento nas organizações.** Curitiba: Aymarã Educação, 2012. p. 128

SWEBOK. **Guide to the software engineering body of knowledge.** Los Alamitos, California: Angela Burgess, 2004.

WEBER, K. C.; ARAUJO, E.; MACHADO, C. A. F.; SCALET, D.; SALVIANO, C. F.; ROCHA, A. R. C. da. **Modelo de referência e método de avaliação para melhoria de processo de software - versão 1.0 (MR - MPS e MA - MPS).** In: Simpósio Brasileiro de Qualidade de Software, 4, 2004, p. 1-14.

WEINBERG, G. M. **Software com Qualidade: pensando e idealizando sistemas.** São Paulo: Makron Books, 1993. p. 385



**APÊNDICE A – INSTRUMENTO DE PESQUISA (QUESTIONÁRIO – PRÉ-TESTE)**

# Questionário - Uso de modelos de qualidade de software em empresas de desenvolvimento

A presente pesquisa visa efetuar um levantamento do uso de modelos de avaliação do software e de modelos de maturidade utilizados em empresas de desenvolvimento de software.

## 1. Escreva o Nome da organização que esta envolvido:

## 2. Selecione o porte da sua empresa:

Segundo o BNDES (2014), existem 5 níveis de classificação de uma empresa

- Microempresa - renda operacional bruta anual menor ou igual a R\$ 2,4 milhões
- Pequeno porte- renda operacional bruta anual maior que R\$ 2,4 milhões e menor ou igual a R\$ 16 milhões
- Médio porte- renda operacional bruta anual maior que R\$ 16 milhões e menor ou igual a R\$ 90 milhões
- Médio-grande porte - renda operacional bruta anual maior que R\$ 90 milhões e menor ou igual a R\$ 300 milhões
- Grande porte - renda operacional bruta anual maior que R\$ 300 milhões

## 3. Sua empresa utiliza modelo(s) de qualidade de software para avaliação do produto software?

Selecione mais de um , caso se aplique

- Não
- Modelo de McCall
- Modelo FURPS+
- Modelo SQUARE NBR Normas ISO/IEC 25000
- Outro:

## 4. Sua empresa utiliza modelo de qualidade de processo e manutenção de software?

- Não
- Modelo SPICE NBR Normas ISO/IEC 15504.
- Modelo CMMI-DEV.
- Modelo MPSBr-SW.
- Outro:

## 5. Caso tenha escolhido um dos três modelos apresentados na questão 4, escreva o nível em que sua empresa se encontra:

## 6. Caso tenha respondido Não, nas respostas 3 e/ou 4, houve ou está ocorrendo a tentativa de implementação de algum modelo de qualidade de software na empresa?

- Não
- Sim

7. Caso tenha respondido Sim na questão anterior, qual é o modelo quais as dificuldades encontradas com a adoção deste modelo:

8. Caso tenha respondido algo diferente de Não nas respostas 3 e/ou 4, com relação ao uso de modelo(s) de qualidade de software, qual é o grau de impacto do uso do(s) modelo(s) no(a):

Classifique o grau de impacto da utilização do(s) modelo(s) de qualidade, sendo que 1 representa pouco ou nenhum impacto e 5 representa alto impacto

	1	2	3	4	5
Custo de produção de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tempo de Produção de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Desempenho do Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Retrabalho da Equipe de Desenvolvimento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Satisfação do Usuário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Manutenção do Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**9. Classifique a importância das características dos produtos de acordo com a visão da sua empresa:**

Classifique o grau de relevância que as características dos softwares desenvolvidos na empresa possuem, considere um panorama geral dos produtos

	1	2	3	4	5
Funcionalidade: capacidade do software de promover funcionalidade que satisfaçam o usuário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confiabilidade: maturidade, tolerância a falhas e recuperação quando necessário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usabilidade: capacidade do software de ser compreendido e suas funcionalidades serem aprendidas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eficiência: tempo de execução, uso de recursos são características que indicam eficiência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Manutenabilidade: facilidade de manutenção de software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Portabilidade: capacidade de transferência do software, de um computador para outro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Limitações de interface, implementação, física e/ou de design	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### 10. Classifique a importância dos processos de software de acordo com a visão da sua empresa:

Classifique o grau de relevância que os processos de software possuem na empresa, considere um panorama geral dos processos

	1	2	3	4	5
Processos que afetam diretamente o cliente, como aquisição do software, instalação e assistência técnica	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Processos que envolvem a transformação de requisitos em um produto de software, como desenvolvimento de requisitos, projeto do software, desenvolvimento do projeto de software e testes de integração do sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Processos de suporte que podem auxiliar outros processos em qualquer estágio do ciclo de vida do software, como desenvolvimento de documentação, gerência de configuração, garantia da qualidade, verificação e validação do software e auditorias



Processos gerenciais, como gestão de projeto, gestão de riscos e gestão da qualidade



Processos associados as atividades gerais da empresa, como, definição de processos, melhoria de processos, treinamento e provisão de infraestrutura organizacional



**11. Independente de usar algum modelo de melhoria dos processos de software e manutenção, qual a sua opinião sobre os mesmos?**

**Caso queira receber os resultados finais desta pesquisa, anote seu endereço de e-mail:**

Enviar



100% concluído.

*Nunca envie senhas em Formulários Google.*

**APÊNDICE B – INSTRUMENTO DE PESQUISA (QUESTIONÁRIO)**

# Questionário - Uso de modelos de qualidade de software em empresas de desenvolvimento

A presente pesquisa visa efetuar um levantamento do uso de modelos de avaliação do software e de modelos de maturidade utilizados em empresas de desenvolvimento de software.

## 1. Escreva o Nome da organização que esta envolvido:

## 2. Selecione o porte da sua empresa:

Segundo o BNDES (2014), existem 5 níveis de classificação de uma empresa

- Microempresa - renda operacional bruta anual menor ou igual a R\$ 2,4 milhões
- Pequeno porte- renda operacional bruta anual maior que R\$ 2,4 milhões e menor ou igual a R\$ 16 milhões
- Médio porte- renda operacional bruta anual maior que R\$ 16 milhões e menor ou igual a R\$ 90 milhões
- Médio-grande porte - renda operacional bruta anual maior que R\$ 90 milhões e menor ou igual a R\$ 300 milhões
- Grande porte - renda operacional bruta anual maior que R\$ 300 milhões

## 3. Sua empresa utiliza modelo(s) de qualidade de software para avaliação do produto software?

Selecione mais de um , caso se aplique

- Não
- Modelo de McCall
- Modelo FURPS+
- Modelo SQUARE NBR Normas ISO/IEC 25000
- Outro:

## 4. Sua empresa utiliza modelo de qualidade de processo e manutenção de software?

- Não
- Modelo SPICE NBR Normas ISO/IEC 15504.
- Modelo CMMI-DEV.
- Modelo MPSBr-SW.
- Outro:

## 5. Caso tenha escolhido um dos três modelos apresentados na questão 4, escreva o nível em que sua empresa se encontra:

## 6. Caso tenha respondido Não, nas respostas 3 e/ou 4, houve ou está ocorrendo a tentativa de implementação de algum modelo de qualidade de software na empresa?

- Não
- Sim



7. Caso tenha respondido Sim na questão anterior, qual é o modelo quais as dificuldades encontradas com a adoção deste modelo:

8. Caso tenha respondido algo diferente de Não nas respostas 3 e/ou 4, com relação ao uso de modelo(s) de qualidade de software, qual é o grau de impacto do uso do(s) modelo(s) no(a):

Classifique o grau de impacto da utilização do(s) modelo(s) de qualidade, sendo que 1 representa pouco ou nenhum impacto e 5 representa alto impacto

	1	2	3	4	5
Custo de produção de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tempo de Produção de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Desempenho do Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Retrabalho da Equipe de Desenvolvimento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Satisfação do Usuário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Manutenção do Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**9. Classifique a importância das características dos produtos de acordo com a visão da sua empresa:**

Classifique o grau de relevância que as características dos softwares desenvolvidos na empresa possuem, considere um panorama geral dos produtos

	1	2	3	4	5
Funcionalidade: capacidade do software de promover funcionalidade que satisfaçam o usuário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confiabilidade: maturidade, tolerância a falhas e recuperação quando necessário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usabilidade: capacidade do software de ser compreendido e suas funcionalidades serem aprendidas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eficiência: tempo de execução, uso de recursos são características que indicam eficiência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Manutenabilidade: facilidade de manutenção de software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Portabilidade: capacidade de transferência do software, de um computador para outro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Limitações de interface, implementação, física e/ou de design	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**10. Classifique a importância dos processos de software de acordo com a visão da sua empresa:**

Classifique o grau de relevância que os processos de software possuem na empresa, considere um panorama geral dos processos

	1	2	3	4	5
Processos que afetam diretamente o cliente, como aquisição do software, instalação e assistência técnica	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Processos que envolvem a transformação de requisitos em um produto de software, como desenvolvimento de requisitos, projeto do software, desenvolvimento do projeto de software e testes de integração do sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

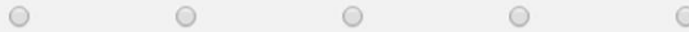
Processos de suporte que podem auxiliar outros processos em qualquer estágio do ciclo de vida do software, como desenvolvimento de documentação, gerência de configuração, garantia da qualidade, verificação e validação do software e auditorias



Processos gerenciais, como gestão de projeto, gestão de riscos e gestão da qualidade



Processos associados as atividades gerais da empresa, como, definição de processos, melhoria de processos, treinamento e provisão de infraestrutura organizacional



**11. Independente de usar algum modelo de melhoria dos processos de software e manutenção, qual a sua opinião sobre os mesmos?**

**Caso queira receber os resultados finais desta pesquisa, anote seu endereço de e-mail:**

Enviar



100% concluído.

*Nunca envie senhas em Formulários Google.*