

## DESENVOLVIMENTO DE UM SOFTWARE PARA CONTROLE GERAL DE UM LAVA-CAR.

ANTONIAZI, Lucas.<sup>1</sup>  
HELENA, André Augusto Ribeiro.<sup>2</sup>

### RESUMO

Todo software, de modo genérico, é uma entidade que se encontra em constante estado de mudança. Essas mudanças ocorrem por necessidades de corrigir erros existentes no software e/ou adicionar novos recursos e funcionalidades no software. Isto é, os sistemas computacionais, isto é aqueles que têm o software como um de seus principais elementos também sofrem mudanças frequentemente. Essa necessidade evolutiva de desenvolvimento de um software o torna predisposto a defeitos, isto se torna sendo não confiável, podendo resultar no atraso de entrega do software e também em custos acima do estimado. Com esses fatos, o crescimento em tamanho e complexidade dos sistemas de software exige que os profissionais da área raciocinem, projetem, codifiquem e se comuniquem por meio de componentes de software, e qualquer concepção ou solução de sistema passa então para o nível arquitetural. Neste sentido, o objetivo deste artigo é discutir e explorar o fato de que software não é construído como uma TV ou algum outro aparelho eletrônico, mas sim desenvolvido. E, todo desenvolvimento de software requer um processo de gestão de projeto, para que se possa ter a obtenção de um produto ou serviço com boa qualidade.

**PALAVRAS-CHAVE:** Desenvolvimento, Qualidade, Funcionalidades.

### ABSTRACT

All software, generally, is an entity that is in a constant state of change. These changes occur in need to correct any errors in the software and / or add new features and functionality in the software. That is, computer systems, ie those who have the software as one of its main elements also undergo changes frequently. This evolutionary need to develop a software makes it prone to defects, it becomes being unreliable and may result in the software delivery delay and also above the estimated cost. With these facts, the growing size and complexity of software systems requires that professionals in the area to reason, to design, encode and communicate by means of software components, and any design or system solution then passes into the architectural level. In this sense, the purpose of this article is to discuss and explore the fact that software is not built like a TV or any other electronic device, but rather developed. And, all software development requires a project management process, so that it may have to obtain a product or service with good quality.

**KEYWORDS:** Development, Quality, Features.

## 1. INTRODUÇÃO

Hoje em dia, a necessidade das empresas por um software informatizado é muito grande. A cada dia mais empresas estão migrando de sua forma antiga de trabalhar para começar a utilizar de novas tecnologias, que disponibilizam várias funções para controle e processamento de informações. Fazendo o uso destes softwares informatizados, é possível realizar vários controles dentro de uma empresa. Por exemplo, em uma empresa prestadora de serviços de lava-car, é possível verificar e controlar quantos veículos foram lavados durante o dia, e também qual foi o funcionário que realizou o serviço.

Segundo Cordeiro *et al* (2010) *apud* Junior (2012), Todo software é destaca-se como um dos principais instrumentos estratégicos para as organizações, e vem sendo utilizado pelas organizações como instrumento de apoio as diversas atividades e a tomadas de decisões. Entretanto, apesar da importância dos softwares no mercado como um instrumento competitivo, produzir software com qualidade ainda é um grande desafio.

No Brasil, a maioria das empresas desenvolvedoras de softwares é classificada como: micro, pequena e média, e tendem a enfrentar dificuldades na implementação dessas normas e modelos, uma vez que, fazem uso desses processos tradicionais de desenvolvimento e trabalham com grandes lotes de processos. Sendo assim, empresas de grande porte são mais propensas a utilizar estas normas.

De acordo com Junior (2007), com a crescente evolução do seu poder de processamento e armazenamento de informações e uma considerável redução de custos, os computadores estão cada vez mais presentes na sociedade atual, que se veem cada vez mais dependentes destas máquinas e de seus programas, também conhecidos como softwares. Estes, por sua vez, tem apresentado um significativo aumento na complexidade interna, fato este que favorece a maior incidência de erros, e conseqüentemente, queda na sua qualidade. Segundo Silva *et al* (2003) *apud* Junior (2007), a qualidade do software é menos adequada do que deveria. A qualidade do software é frequentemente suspeita.

As novas tecnologias prometem uma mudança qualitativa na forma que pode ocorrer à aprendizagem. A educação não é mais unidirecional, a informação agora circula de forma bidirecional, colaborativa e interdisciplinar e as novas técnicas de utilização de softwares informatizados começam a quebrar barreiras e ainda assim customizam a organização e a tornam mais econômicas.

<sup>1</sup>Acadêmico do curso de Sistemas de Informação da Faculdade Assis Gurgacz. E-mail: lucasantoniazi3@gmail.com

<sup>2</sup>Professor da Faculdade Assis Gurgacz. E-mail: andrehelena@gmail.com

Para Pereira Junior (2007), um processo de software pode ser definido como uma coleção de fatores necessários para a construção de software de alta qualidade. Segundo Ambler *et al* (1998) *apud* Junior (2007), um processo ainda pode ser definido como uma série de ações que na qual uma ou mais entradas são utilizadas para produzir uma ou mais saídas. Os termos de qualidade e confiabilidade tem merecido uma atenção especial dos pesquisadores quando se foca no desenvolvimento de software moderno. Existem vários processos de desenvolvimento gerados para nortear a elaboração de software e garantir que tanto a confiabilidade quanto a qualidade sejam asseguradas.

Segundo Costa (2004), a construção de um sistema de computador apresenta varias oportunidades de melhorar seus conhecimentos na área de informática, obter experiência e adquirir uma melhor colocação no mercado de trabalho. Um sistema informatizado disponibiliza vários recursos e sua utilização traz melhorias para uma empresa, possibilitando um aumento da qualidade tanto em produtos quanto serviços. É possível realizar o controle de suas atividades simultaneamente e assim permitindo a clareza de informações e, também, o rápido acesso a qualquer tipo de informação que esteja cadastrada no sistema.

Conforme Berni (2010), o desenvolvimento de software não combina somente com técnicas e ferramentas, é predominante a uma atividade humana. Em uma empresa de software, os principais recursos para o desenvolvimento de atividades são providos pelas interações humano-computador. A necessidade de planejamento do processo produtivo torna-se indispensável para que haja um melhor aproveitamento desses recursos.

As usabilidades de softwares informatizados em empresas pequenas, de médio e grande porte estão com um avanço tecnológico muito grande em questão de tecnologia. Com essa revolução da tecnologia da informação, cada software desenvolvido, e que já está no mercado de trabalho, tende a sempre ter mais inteligência em suas funções, disponibilizando um acesso de melhor qualidade para as empresas que estarão a utilizando desse software.

Para as empresas que utilizam destas novas tecnologias da informação, também é possível ter controle total de sua empresa via remota, aonde somente conectado através da internet é possível realizar o controle e verificar o que se passa no decorrer do dia a dia dentro da empresa. Utilizando desta função, é possível realizar a busca de informações em tempo real, sem nenhuma perda de dados, gerando um resultado satisfatório a empresa, e também possibilitando realizar todos os controles possíveis de sua empresa, permitindo assim a eficiência e eficácia da informação.

Conforme Sommerville *et al* (1995) *apud* Berni (2010), o processo de software é um conjunto de atividades e resultados associados que produzem um produto de software. Para Schwartz *et al* (1975) *apud* Berni (2010), as principais fases de um projeto de software são: Especificação de requisitos, projeto de sistema, programação (codificação), verificação e integração. Em cada fase de desenvolvimento de software são executadas atividades inerentes para que os objetivos propostos possam ser atingidos. De acordo com Pressman *et al* (2006) *apud* Berni (2010), estas atividades constituem um conjunto mínimo para se obter um produto de software. Ainda existem questões de alocações de recursos e pessoal que precisam ser previstos quando do inicio de cada projeto.

De acordo com Pereira Junior (2007), outra particularidade em um desenvolvimento de software é o levantamento e identificação de requisitos ao longo de seu desenvolvimento. Quando se desenvolve um software comercial, geralmente já se tem a noção, mesmo que inexata, do sistema como um todo, ou seja, de todas as suas funcionalidades. Outra particularidade ainda pode ser observada. A alta rotação dos pesquisadores em uma pesquisa. Geralmente um software é desenvolvido para validar uma determinada parte de uma pesquisa, e existe uma equipe de desenvolvimento pré-definida e não se altera até o término do projeto. Além disso, no desenvolvimento de um software existem pessoas com funções distintas, como os analistas de sistemas, os administradores da base de dados, engenheiros de testes e os programadores.

## 2. REFERENCIAL TEÓRICO OU FUNDAMENTAÇÃO TEÓRICA

A indústria de software representa uma das mais importantes áreas de desenvolvimento econômico, impulsionando, desta forma, uma constante abertura de novas pequenas, médias e grandes empresas. De acordo com Pereira Junior (2007), o software hoje deve ser eficiente o suficiente para entregar a informação necessária, na hora correta, á pessoa interessada. Mas tão importante quanto a eficiência de um software é a sua qualidade. Um software que seja bastante eficiente, mas que não gere resultados satisfatórios ou que não as suas funcionalidades desejadas, obviamente não será bem sucedido.

Para Huzita *et al* Tait, Colanzi, e Quinaia (2007), No contexto desta pesquisa, considera-se que o desenvolvimento de software ocorre por meio de atividades de projetos executadas por membros de equipes que podem estar em locais geograficamente distintos. Além disso, os gerenciadores que são responsáveis por prover o suporte necessário, podem estar distribuídos, levando ao ambiente de desenvolvimento. Com isto, procura-se tirar o máximo proveito do produto de software, almejando um produto de qualidade com custos reduzidos.

Segundo Vilella, Travassos e Rocha (2004), diferentes abordagens têm sido propostas para melhorar a qualidade de processos e produtos. No entanto, pouca atenção tem sido dada á produção, armazenamento, compartilhamento e uso de conhecimentos relevantes para o desenvolvimento e manutenção de um software. Sempre que o conhecimento requerido não está disponível, o desenvolvedor do software tem que partir do zero para encontrar a

solução do problema, o que tem sérias implicações em termos da qualidade da solução. Verifica-se, desta forma, a importância da administração sistemática dos conhecimentos relevantes neste contexto.

Para Paula *et al* (2001) *apud* Pereira Junior (2007), a qualidade de um software é proporcional ao seu grau de conformidade com os respectivos requisitos. Sendo assim, quanto mais preciso for um software, no sentido de atingir as suas necessidades identificadas, mais qualidade ele terá. A qualidade de um software aplicando um conjunto de métodos e ferramentas. Este conjunto de métodos e ferramentas é gerenciado por um processo. Têm-se, então, segundo Pressman *et al* (2006) *apud* Pereira Junior (2007), três fundamentos da Engenharia do software: Ferramentas, Métodos e processos. Os métodos proporcionam todos os detalhes de como fazer para construir um software. Neste fundamento, são estabelecidos os critérios e as formas de trabalho da equipe de desenvolvimento. Os métodos, quando aplicados de forma correta e coerente, permitem o gerenciamento ordenado do ciclo de desenvolvimento do software, fator este qual é crucial para a garantia da sua qualidade.

De acordo com Vilella, Travassos e Rocha (2004), os ADS (Ambientes de Desenvolvimento de Software), são sistemas computacionais que apoiam o desenvolvimento, reparo e melhoria de produtos de software bem como o controle e gerenciamento destas atividades. A ADS tem evoluído ao longo do tempo para fornecer apoio mais amplo e efetivo aos desenvolvedores de software, tendo surgido os Ambientes de Desenvolvimento de Software Centrados em Processos, onde visa focar com mais ênfase em cima do processo de desenvolvimento do software.

Segundo Silva Filho (2011), antigamente os projetos de desenvolvimentos de softwares alocavam uma pequena parcela ao software. Todos componentes de hardwares de um computador eram testados e analisados quase que exaustivamente, o que permitia a produção rápida de grandes de subsistemas e implicava em raros erros de projetos. Entretanto a facilidade de modificar o software, comparativamente ao hardware, motivou seu uso. A intensificação do uso do software numa larga variedade de aplicações o fez crescer em tamanho e complexidade. Isto acabou se tornando proibitivo analisar e testar exaustivamente, além de impactar no custo de manutenção.

Conforme DeMarco *et al* (2009) *apud* Berni (2010), não se pode controlar o que não se pode medir. O desenvolvimento de software com componentes de inovação, com quais se busca um maior retorno sobre investimentos pela diferenciação de mercado, são ainda mais arriscados e possuem processos de desenvolvimento mais imprevisíveis. Métricas de controle não funcionam muito bem nesse tipo de ambiente. DeMarco (2009), também afirma que a melhor forma para gerenciar o processo de desenvolvimento é organizar os colaboradores e controlar somente as variáveis de tempo e dinheiro.

Para Costa (2004), o objetivo a ser alcançado com o desenvolvimento de um sistema, será agilizar não somente a parte administrativa de uma empresa, mas todos os setores envolvidos. Segundo Renault, Bonilla e Frach (2009) *apud* Toniolo (2011, p.32), o reuso dos requisitos é a tendência de se utilizar o conhecimento em experiências anteriores no levantamento de requisitos e impulsionar a reutilização desse conhecimento.

De acordo com Toniolo (2011), isso se aplica de fato quando uma empresa tem uma equipe experiente na área de desenvolvimento, que mais facilmente essa equipe poderá encontrar formas de reuso de qualquer fase do projeto e, também, assim sendo mais focado no levantamento de requisitos. No que diz respeito às técnicas de reuso, tanto de código fonte como de requisitos ou até mesmo de qualquer parte de um projeto, López (2005) *apud* Toniolo (2011, p.32) elenca um total de seis técnicas que estão sendo relacionadas aos requisitos funcionais do sistema. São elas: Cenários, Casos de Uso, Diagramas de Atividades, Fluxo de Dados, Documento de Tarefa e Fluxo de Dados.

Segundo Toniolo (2011), essas técnicas são utilizadas de acordo com a experiência de cada equipe, e se dá na fase de elicitação de requisitos funcionais junto aos stakeholders do sistema. A viabilidade de se utilizar um sistema informatizado hoje está com um índice muito alto, pois varias empresas estão optando por mais qualidade e eficiência em seus produtos e serviços. Quando se fala em sistemas informatizados, é sempre focado muito no conceito de segurança de informações, justamente se o sistema for informatizado, isso se caracteriza uma preocupação maior em cima dos dados. Pois quando um sistema é informatizado não pode ter perda de informações, o quesito segurança da informação tem que atender o sistema como um todo, sendo assim, garantindo a integridade, qualidade e rapidez das informações.

De acordo com Berni (2010), a forma em que todo ambiente ou estrutura influencia para que um desenvolvimento de software seja realizado é a razão em que essas empresas adotam metodologias ágeis ou decidem utilizar metodologias tradicionais que devem ser tomadas pela organização de modo que possa honrar seus acordos comerciais, cumprindo cronogramas e desenvolvendo produtos com qualidade desejável. Em micro pequenas empresas, observasse que a adoção de mecanismos para a padronização do processo produtivo acaba burocratizando a empresa em excesso, restringindo sua flexibilidade. Segundo O'Connor *et al* (2007) *apud* Berni (2010), quanto menor a empresa, maior a aversão a documentação e burocracia.

Nesta pesquisa de artigo científico, será utilizada a metodologia de desenvolvimento da engenharia do software, chamada de RUP (*Rational Unified Process*), que traduzido para o português significa Processo Unificado Racional. Para IBM *et al* (1998) *apud* PARAÍSO (2011), O RUP é utilizado de forma customizada, pois ele é um processo muito complexo. O RUP apresenta seis das melhores práticas para um bom desenvolvimento de como desenvolver o software iterativamente, gerar requisitos, gerar arquitetura baseadas em componentes, modelar o software visualmente, verificar

a qualidade do software e controlar as mudanças do software. Logo abaixo, na figura 1, está demonstrando a forma em que o RUP classifica a sua ordem de desenvolvimento.

FIGURA 1 – Modelo de desenvolvimento.

### Modelo de Desenvolvimento Tradicional



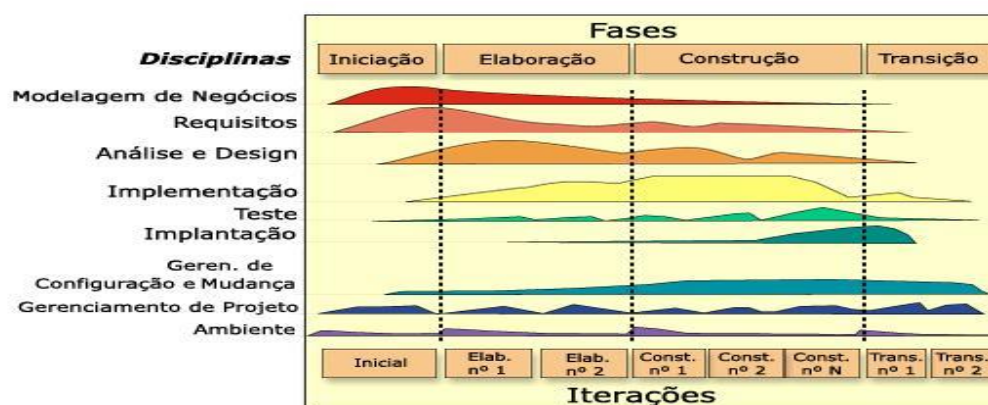
Fonte: RUP – *Rational Unified Process* (2013).

No modelo de desenvolvimento do RUP, por primeiro vem os requisitos, que tem por finalidade analisar e verificar qual é o problema ou necessidade do usuário em questão de software. O design é a parte em que se preocupa com o desenvolvimento do visual gráfico para o software, deixando assim de uma forma mais agradável e intuitiva para o usuário final estar fazendo o uso, poupando assim tempo para o usuário executar suas tarefas mais rapidamente e um custo menor com um software bem planejado.

A codificação é a parte em que o desenvolvimento do sistema realmente ocorre aonde o programador se encarrega de escrever em linguagem de programação para que o computador entenda quais são as devidas tarefas em que o software terá que atender. Já a parte de teste, foca mais em testar se o software está executando corretamente o que o usuário final solicitou, é nos testes que também é feito as correções de bugs e de possíveis erros do software. E por fim a implantação é a parte que se encarrega de realizar a implantação do software para o cliente, fazendo assim a instalação e as devidas configurações necessárias para o usuário final fazer o uso do software da melhor forma possível.

Com o uso da metodologia do RUP é possível realizar a modelagem do software. Essa metodologia trabalha em cima de fases, que, em cada fase do RUP será realizada alguma determinada função a respeito do desenvolvimento software. Atualmente existem quatro fases no desenvolvimento, fase de iniciação, elaboração, construção e transição. Logo abaixo na figura 2, está demonstrando corretamente as quatro fases do RUP.

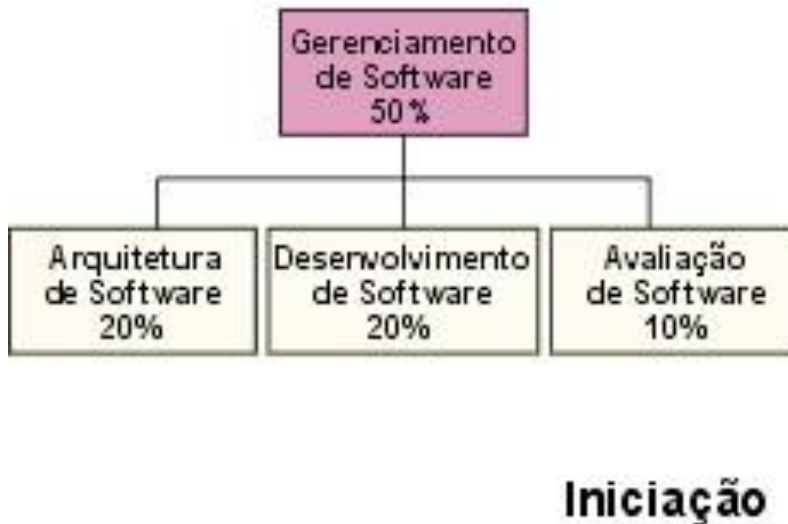
FIGURA 2 – Fases do RUP.



Fonte: PARAÍSO, (2011, p.22).

Na fase de iniciação do RUP, é a fase de criação do escopo do projeto do software. Nela é realizada a avaliação de risco, uma estimativa de recursos necessários e cronogramas dos marcos temporais mais importantes. Todos os atores envolvidos na iteração do sistema também são identificados nesta fase. Logo abaixo na figura 3, está demonstrando a forma em que o RUP trabalha na fase de iniciação.

FIGURA 3 – Fase de Iniciação.



Fonte: *Rational Software Corporation* (2001).

A fase de elaboração cuida dos pontos de especificações e eliminações de maior risco, buscando complementar o levantamento e documentação voltada para a arquitetura do sistema, revisa a modelagem de negócio para os projetos e inicia a versão do manual do usuário. Logo abaixo na figura 4, está demonstrando a forma em que o RUP trabalha na fase de elaboração.

FIGURA 4 – Fase de Elaboração.



Fonte: *Rational Software Corporation* (2001).

Na fase da construção, se caracteriza como o início do desenvolvimento do sistema, ao final de cada ciclo, e pode surgir uma versão utilizável do processo. Logo abaixo na figura 5, está demonstrando a forma em que o RUP trabalha na fase de Construção.

FIGURA 5 – Fase de Construção.

## Construção



Fonte: Rational Software Corporation (2001).

E por último a fase de transição, que se encarrega de realizar a implantação, entrega acompanhamento e qualidade do software para usuário final. Logo abaixo na figura 6, está demonstrando a forma em que o RUP trabalha na fase de Transição.

FIGURA 6 – Fase de Transição.

## Transição



Fonte: *Rational Software Corporation* (2001).

Conforme Pereira Junior (2007), todo software, por sua vez, pode ser definido como o produto com que a engenharia de software se preocupa. Em outras palavras, é o código executável por um computador, produzido por engenheiros e programadores do software, que reflete as implementações das necessidades de uma determinada

situação. Tal produto é hoje o fator crucial no quesito competitividade empresarial. O software pode influenciar, por exemplo, na tomada de decisões nos negócios. Estes fatos tornam inviável a existência de uma grande empresa que queria ser competitiva no mercado sem a sua utilização.

### 3. METODOLOGIA

Este capítulo tem a finalidade de apresentar a metodologia utilizada no desenvolvimento da pesquisa que em tese será realizado o desenvolvimento de um sistema que controla e gerencia algum determinado lava-car, que está dividida em uma caracterização geral do método utilizado, que traz as definições dos tipos de pesquisa e abordagens; procedimentos metodológicos, que apresenta e descreve as etapas de desenvolvimento da pesquisa.

#### 3.1 CARACTERIZAÇÃO GERAL DO MÉTODO

Essa pesquisa se classifica como uma pesquisa de campo, exploratória, com abordagem qualitativa e, também, quantitativa. Os resultados obtidos serão transformados em informações para a análise descritiva do que realmente o sistema terá que atender. A abordagem qualitativa será utilizada, também, na análise das informações colhidas dos proprietários das empresas, quando do levantamento de suas necessidades.

De acordo com Piana (2009), a pesquisa de campo é o tipo de pesquisa que pretende buscar a informação diretamente com a população pesquisada. Ela exige do pesquisador um encontro mais direto. Neste caso, o pesquisador precisa ir ao espaço aonde o fenômeno ocorre ou ocorreu e reunir um conjunto de informações a serem documentadas.

Segundo Moresi (2003), a pesquisa qualitativa é particularmente útil como uma ferramenta para determinar o que é importante para os clientes e por que é importante. Esse tipo de pesquisa fornece um processo a partir do qual questões-chaves são identificadas e perguntas são formuladas, descobrindo o que importa para os clientes e por que. De acordo com Moresi (2003), a pesquisa qualitativa revela áreas de consenso, tanto positivo quanto negativo nos padrões de respostas. Ela também determina quais ideias geram uma forte reação emocional. Além disso, é especialmente útil em situações que envolvem o desenvolvimento e aperfeiçoamento de novas ideias.

Conforme Moresi (2003), a pesquisa quantitativa é apropriada para medir tanto opiniões, atitudes e preferências como comportamento. Se você quer saber quantas pessoas usam um produto ou serviço ou tem interesse em um novo conceito de produto, a pesquisa quantitativa é o que você precisa. Ela também é usada para medir um mercado, estimar o potencial ou volume de um negócio e para medir o tamanho e a importância de segmentos de mercado.

Segundo Gil (2008), a pesquisa exploratória proporciona maior afinidade com o problema (explícita-lo). Pode envolver levantamento bibliográfico, entrevistas com pessoas experientes no problema pesquisado. Geralmente assume a forma de pesquisa bibliográfica e estudo de caso. Uma pesquisa exploratória tem por objetivo familiarizar-se com o assunto ainda pouco conhecido, pouco explorado. Como qualquer exploração, depende muito da intuição do pesquisador, por ser um tipo de pesquisa específica.

Conforme Berni (2010), pequenas empresas de software não estão adotando metodologias prescritivas justamente pelo atrelamento a burocracia pertinente aos modelos que são propostos. Em um ambiente em constante modificação, onde velocidade de respostas e entrega de produtos é o diferencial competitivo, um modelo de processo adaptável é o intento das organizações.

De acordo com Berni (2010), ao se tentar prever, ao início do projeto, todos os fatores que influenciarão as atividades de desenvolvimento e, a partir disso, planejar sua execução acaba engessando sua evolução. Conforme Brooks *et al* (1987) *apud* Berni (2010), especificar totalmente um software antes do início de sua implementação é impossível. Com base nestas justificativas, surgem novas metodologias para o desenvolvimento de software que pregam que a elaboração de software é um processo criativo e experimental, e não deve ser associado de forma rígida a cronogramas, prazos e escopos de desenvolvimentos.

#### 3.2 PROCEDIMENTOS METODOLÓGICOS

Para que todos os projetos de desenvolvimento de software sejam realizados, são seguidas algumas regras utilizadas no decorrer do desenvolvimento do projeto. Sempre quando se vai começar a criar um software, deve ser criado um projeto antes. Neste projeto existem algumas regras que deverão ser seguidas. Em algumas dessas regras, estão encontrados os levantamentos dos requisitos para o software, conforme foi mostrado anteriormente na figura 1. O levantamento de requisitos se caracteriza em verificar com o usuário final, analisar o que ele realmente precisa, qual é o tipo da necessidade que ele necessita que seja sanada pelo software. Sendo assim, o levantamento de requisitos serve para verificar e analisar as melhores formas de resolver o problema de software do cliente.

Para Ricarte (2002), em um desenvolvimento de software existem cinco ordens de ignorância, a falta de ignorância, falta de conhecimento, falta de consciência, falta de processos e a meta ignorância. A falta de ignorância caracteriza-se como quando o desenvolvedor conhece alguma coisa e pode demonstrar esse assunto. Falta de

conhecimento é quando o desenvolvedor não sabe alguma coisa, mas sabe identificar este fato. Falta de consciência é quando o desenvolvedor não sabe de alguma coisa e nem sabe que não sabe. Falta de processo é quando ele não sabe como fazer para descobrir que há coisas que ele não sabe. E por último a meta-ignorância, que o desenvolvedor não sabe sobre as cinco ordens de ignorância no desenvolvimento de um software.

De acordo com Ziv *et al* (1997) *apud* Berni (2010), a incerteza é inevitável ao desenvolvimento de software. Conforme Wagner *et al* (2005) *apud* Berni (2010), não é possível especificar completamente um sistema interativo. Nota-se que o processo de produção de software apresenta alguma característica que o fazem diferenciar-se de outras áreas de produção industrial. Um dos pontos principais desta é a não linearidade desse tipo de processo.

Quando se trata de desenvolvimento de um software, para todas as empresas que fornecem esses serviços de desenvolvimento, geralmente é comum que essas empresas utilizam de metodologias de desenvolvimento já existentes no mercado de trabalho, que auxilia no desenvolvimento desse projeto que futuramente será um software. Sempre em questão de desenvolvimento de um software é dada muita ênfase na questão de qualidade e segurança da informação, e também a rapidez de entrega do software. Muitas empresas se preocupam pouco na questão de qualidade e segurança da informação, aonde assim deixam um pouco a desejar no seu software final, em quantas outras empresas se preocupam mais com a qualidade e segurança da informação e acabam deixando a desejar no quesito de rapidez de entrega desse software, para que o usuário final possa estar utilizando para controlar e gerenciar sua empresa.

Segundo Woo *et al* (2006) *apud* Fernandes, Tait e Bruzarosco (2012, p.104), ao empregar uma metodologia consistente pode-se economizar uma quantidade considerável de tempo, dinheiro e reduzir o número de defeitos no produto desenvolvido. Para todo projeto de software a equipe tem que delimitar algumas ações que irão ser executadas durante o desenvolvimento do projeto. Como por exemplo, o tipo de linguagem de programação que irá utilizar para o desenvolvimento, qual a estrutura de banco de dados que se mais se adequa para determinada ocasião e também quais serão os métodos corretos a serem utilizados no decorrer do desenvolvimento do projeto e futuramente com resultado final de um software.

Existem várias Linguagens de programação no mercado de trabalho com qual é possível estar trabalhando. Neste caso de desenvolvimento de software, a linguagem de programação que será utilizada para o processo de desenvolvimento desse projeto de software será a Object Pascal, que por fim é, também, uma linguagem de programação orientada a objetos. Para que se possa ser feito o uso desta linguagem de programação, é recomendado utilizar algum tipo de IDE (*Integrate development Environment*), ou seja, é um ambiente integrado para que o desenvolvimento de algum software possa ser realizado. Atualmente será trabalhado com uma IDE chamada Borland Delphi 7, que nada mais é do que uma IDE de desenvolvimento criada pela empresa Borland, e assim chamada de Delphi 7.

Todo software que atenda algum tipo de comércio, tanto como de lava-car, como qualquer outro tipo de comércio, necessita de um Banco de Dados para armazenar, assim, todas as informações do seu estabelecimento conforme o seu funcionamento de cada dia. Neste caso, será utilizada a versão de banco de dados MySQL 4.1, onde a linguagem que será utilizada se chama SQL (*Structured Query Language*), também conhecida como Linguagem de Consulta Estruturada, e como IDE dessa linguagem será utilizada o SQLyog, que é uma IDE de desenvolvimento que atende os requisitos da linguagem MySQL 4.1.

Com o uso destas ferramentas, todo desenvolvimento de software fica de alguma forma, mais fácil e mais rápido de ser realizado. Atualmente o ramo de tecnologia da informação está tendo um grande avanço a cada dia que se passa, aumentando assim o desempenho e qualidade de novos softwares que estão sendo desenvolvidos. Essas linguagens de programação, tanto para o desenvolvimento do software, como, para o banco de dados deste software, necessitam da interação de um programador de software para criar as devidas funções lógicas que o software irá executar, pois não tem como algum software ser desenvolvido sozinho a partir do zero.

### 3.2.1 Local de Realização da Pesquisa

Esta pesquisa será realizada na cidade de Santa Lúcia – Paraná, buscando atender a uma demanda por tecnologia por parte das empresas do comércio e prestação de serviço do local, aonde tem um número grande de empresas que necessitam de sistemas informatizados que façam o controle geral de toda a sua estrutura empresarial.

Neste caso, com essa pesquisa é possível relatar um grande índice de empresas que não estão voltadas a tecnologia da informação, pois essa e algumas cidades da região com pouco número de habitantes acabam não conseguindo acompanhar o nível de outras cidades com um porte de habitantes maior.

## 4. ANÁLISES E DISCUSSÕES

O presente estudo deste artigo apresentou uma ferramenta de software que foi desenvolvida para controle geral de um lava-car, onde, é possível realizar a integração de dados, monitoramento e gerar informações de possíveis



diagnósticos ou intervenções realizadas pelo software. Pequenas empresas têm necessidades diferentes de grandes organizações produtoras de software. Com o excesso de burocracia ou demasiada especificações de processos no decorrer de um desenvolvimento de software, pode causar uma perda grande para a empresa, reduzindo assim a capacidade da empresa de produzir mais com mais qualidade e em menos tempo. Se todo software for planejado corretamente seguindo das metodologias já disponíveis, terá uma boa chance de ser um software de qualidade, onde possivelmente terá um mercado excelente e muito amplo de utilização.

Conforme analisado, a perda de foco no desenvolvimento e aperfeiçoamento de um produto pode representar o fracasso no andamento de qualquer projeto. Assim quando todas as empresas de desenvolvimento de software utilizam de metodologias ou processo de produção adaptados as realizadas enfrentadas pelas organizações, é de fundamental importância para sua sobrevivência e crescimento em meio o mercado competitivo. Dessa forma, faz-se necessário a busca por meio de sistematizar o processo produtivo de empresas desenvolvedoras de software que se adequam as características mencionadas. As metodologias fornecem possibilidades de organização e planejamento para a produção de software, permitindo que o processo seja simplificado e adaptável.

## 5. CONSIDERAÇÕES FINAIS

O objetivo deste artigo é representar uma abordagem de desenvolvimento de software capaz de auxiliar e controlar uma empresa prestadora de serviços, com ênfase em um lava-car. De acordo com os resultados obtidos desta pesquisa, constatou-se que as empresas de um modo geral apresentam dificuldades no que diz respeito à documentação ineficiente, ausência de dados históricos e de agilidade no processamento de informações. Estas informações auxiliam na determinação de quais práticas as empresas pesquisadas tem maior necessidade em se aprimorar para que seus produtos e serviços sejam executados de forma mais eficiente.

O uso dos conhecimentos de domínio durante o processo de desenvolvimento de softwares é um fator diferencial para desenvolvedores que não conheçam o domínio ou não tem experiência em desenvolver algum tipo de software. Neste contexto, percebe-se que muito pode ser feito para melhorar a qualidade do software no decorrer do desenvolvimento, facilitando assim futuramente também a manutenção e desenvolvimento de novas versões deste software.

Durante o desenvolvimento do software muitos recursos de informações são tratados, e em muitas situações, é essencial estabelecer conexões entre elas para se obter o conjunto necessário de informações para apoiar a realização de uma atividade. Portanto, é importante manter a semântica dos itens nos ambientes de desenvolvimento de software, de modo que sempre esteja sempre evoluindo o processo atual para um processo atualizado e melhorado.

Uma importante contribuição do presente trabalho é a possibilidade das empresas implantarem algumas práticas ou mesmo documentos, de forma gradativa, sem que implique em muitos impactos em seus métodos de trabalho. Caso uma determinada empresa identifique que a forma de documentar a atividade de testes, necessita de um detalhamento de informação mais elaborado, sem que isto implique em muitas mudanças nos demais documentos ou atividades, já desempenhadas na empresa.

Naturalmente em todo desenvolvimento de software existem bastantes dificuldades e vários obstáculos. No processo de desenvolvimento destes softwares, certamente que encontrará desafios, mas suas vantagens justificam e compensam todos os esforços para desenvolvê-lo. Todo software deve transcender o paradigma de sistemas de informação e comunicação, possibilitando um rápido acesso as suas funcionalidades, com uma boa qualidade durante a sua execução. Logo, todo software que atenda a essas características, é um software que está de acordo com o paradigma de sistemas de informação e comunicação.

Além disso, percebeu-se uma tendência cada vez maior de empresas utilizando destes softwares informatizados, caracterizando assim uma nova forma de melhoria para as empresas desenvolvedoras. Assim, se se torna de alguma forma com mais facilidade aplicar o uso de estratégias de negócios em uma empresa. Possibilitando assim um resultado final de software com boa qualidade e bastante competitivo no mercado de trabalho.

Com essa ideia, é possível realizar uma avaliação do software, tanto no seu processo de desenvolvimento quanto no software pronto e já em execução. Com essas informações, podem influenciar sobre a qualidade no desenvolvimento e no uso dos softwares. Ao mesmo tempo em que um software é desenvolvido, o profissional que o está desenvolvendo já faz alguns tipos de testes de avaliações, gerando assim um relatório de melhoria e com possíveis soluções propostas causadas por algum tipo de erro ou bug no software.

Concluindo, a concepção de um processo de desenvolvimento de software, envolve além do conteúdo visto nesta pesquisa de artigo, pois em um processo de desenvolvimento, sugere-se fazer abordagens em vários estudos, para assim verificar quais os melhores conceitos, teorias e metodologias a serem utilizadas no decorrer do desenvolvimento. A tendência de softwares nas áreas comerciais e também de prestação de serviços, é de cada vez mais se adaptar o uso do software com o estilo do usuário, adequando o software de acordo com a necessidade e, também utilizando de estratégias que auxiliam no decorrer do processo de desenvolvimento do software.

## REFERENCIAS

BERNI, J.C.A. **Gestão para o processo de desenvolvimento de software científico** - utilizando uma abordagem ágil e adaptativa na microempresa. Universidade Federal de Santa Maria (RS), 2010 (Dissertação de Mestrado).

COSTA, N.T **Projeto para desenvolvimento de um sistema para controle de estoque**. Faculdade de Ciências Aplicadas de Minas. Uberlândia, 2004. (Dissertação de Mestrado).

FERNANDES, D.B; TAIT, T.F.C; BRUZAROSCO, D.C. **Uma Contribuição para o Processo de Desenvolvimento de Software em Micro e Pequenas Empresas Desenvolvedoras de Software**. VIII Simpósio Brasileiro de Sistemas de Informação, Maringá – Paraná, 2012.

GIL, R.L. **Tipos de pesquisa**. Licenciatura em Ciências Biológicas. Disciplina de Pesquisa do Ensino de Ciências e Biologia, São Paulo, 2008.

HUZITA, E.H.M; TAIT, T.F.C; COLANZI, T.E; QUINAIA, M.A. **Um Ambiente de Desenvolvimento Distribuído de Software DISEN**. Universidade Estadual de Maringá, Maringá – PR, 2007.

JERONIMO JUNIOR, H. **Desenvolvimento Sustentável e Responsabilidade Social**: as contribuições da engenharia de produção. análise e viabilidade de aplicação do pensamento enxuto para atender os processos de produção de softwares do modelo de maturidade mps.BR, XXXII Encontro nacional da Engenharia de Produção Belo Horizonte Bento Gonçalves – RS, 2012.

MORESI, E. **Metodologia da Pesquisa**. Programa de pós-graduação stricto sensu em gestão do conhecimento e tecnologia da informação, Universidade Católica De Brasília – UCB, Brasília, 2003.

PARAÍSO, C.M.S. Desenvolvimento de um sistema de controle acadêmico – acadsystem. Faculdade de Tecnologia São José dos Campos, 2011 (Trabalho de Graduação).

PEREIRA JUNIOR, M. Concepção de um processo de desenvolvimento de específico para software científico. Centro Federal de Educação Tecnológica de Migas Gerais. Belo Horizonte – MG, 2007. (Dissertação de Mestrado).

PIANA, M.C. **A pesquisa de campo**. A construção do perfil do assistente social no cenário educacional. Editora UNESP, São Paulo, 2009.

RATIONAL SOFTWARE CORPORATION. **Diretrizes**: plano de desenvolvimento de software. Disponível em: [http://www.wthree.com/rup/process/modguide/md\\_prpln.htm](http://www.wthree.com/rup/process/modguide/md_prpln.htm). Acessado em: 07/10/2015.

RUP – RATIONAL UNIFIED PROCESS. **Modelo de desenvolvimento tradicional**. Disponível em: <https://carledwinj.wordpress.com/2013/08/25/rup-rational-unified-process/> Acessado em: 07/10/2015.

SILVA FILHO, A.M. Desenvolvimento de software requer processo e gestão. **Revista Espaço Acadêmico**, n.123, p.46-56, 2011.

TONIOLO, C.M. Reuso de requisitos para famílias de produtos em sistemas embarcados. **Revista Ibirapuera**, n.1, p.31-35, 2011.

VILLELA, K; TRAVASSOS, G.H; ROCHA, A.R.G. **Definição e Construção de Ambientes de Desenvolvimento de software orientados a organização**. Programa de Engenharia de Sistemas e Computação. Universidade Federal do Rio de Janeiro (RJ), 2004. (Artigo).