



UNISUL

UNIVERSIDADE DO SUL DE SANTA CATARINA

MOACIR CARLOS TONINI SALVADOR

PROCESSO DE HOMOLOGAÇÃO: UM ESTUDO DE CASO

Florianópolis

2015

MOACIR CARLOS TONINI SALVADOR

PROCESSO DE HOMOLOGAÇÃO: UM ESTUDO DE CASO

Monografia apresentada ao Curso de Especialização em Engenharia de Projetos de Software da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Especialista em Engenharia de Projetos de Software.

Orientador: Prof. Maria Inés Castiñeira, Dra.

Florianópolis

2015

MOACIR CARLOS TONINI SALVADOR

PROCESSO DE HOMOLOGAÇÃO: UM ESTUDO DE CASO

Esta monografia foi julgada adequada à obtenção do título de Especialista em Engenharia de Projeto de Software e aprovada em sua forma final pelo Curso de Especialização em Engenharia de Projeto de Software da Universidade do Sul de Santa Catarina.

Florianópolis, 11 de fevereiro de 2015.

Professora orientadora Maria Inés Castiñeira, Dra.
Universidade do Sul de Santa Catarina

Prof^a. Vera Regiane Niedersberg Schuhmacher, Dra.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Agradecer a minha família que sempre me deu apoio durante todo o curso de pós-graduação e principalmente nesta fase final

Aos colegas de turma que estiveram sempre juntos em diversos momentos sendo eles de estudo ou descontração, tendo eles como amigos que deixam saudades. A minha colega que se tornou uma grande amiga Claudiana. E aos colegas que estiveram em momentos importantes como a Anderson, William e Neri.

Agradecer a Softplan pelo incentivo e por disponibilizar as informações necessárias para a conclusão deste trabalho. Dentro da empresa quero agradecer a minha tutora Solange, e aos colegas que sempre estiveram dispostos a fornecer material necessário para o desenvolvimento deste trabalho Humberto Zilio, Luiz Giovanella, Márcia, Kazumi, Daniel e Wanderson.

Aos professores que estiveram sempre contribuindo para o desenvolvimento do trabalho especialmente a professora Maria Inés Castiñeira.

“É necessário abrir os olhos e perceber que as coisas boas estão dentro de nós, onde os sentimentos não precisam de motivos nem os desejos de razão. O importante é aproveitar o momento e aprender sua duração, pois a vida está nos olhos de quem sabe ver.”
(Gabriel Garcia Marquez)

RESUMO

O mercado de software cresceu muito nos últimos anos e cada vez mais empresas entram no ramo. Os clientes tornam-se cada vez mais exigentes e buscam um produto de qualidade para solucionar seus problemas. As empresas, por sua vez, aprimoram seus processos de desenvolvimento utilizando metodologias da engenharia de software. Neste trabalho foram estudados as etapas de desenvolvimento de software, descrevendo sucintamente as atividades que são realizadas em cada uma dessas etapas. Destacou-se a importância da qualidade do software e a atividade de validação do cliente como ponto importante para garantia da qualidade. Foi realizado o estudo de caso da implantação do processo de homologação em uma empresa de software, analisando os dados dos atendimentos registrados pelos clientes indicando erros no sistema, que foram abertos no período de um ano antes da implantação e um ano durante a implantação do processo. Também foi diagnosticada a importância de um processo de homologação, acompanhado pela empresa junto com o cliente; este acompanhamento durante o processo de homologação aumentou a qualidade do software instalado em produção, reduzindo os erros que impactam na operação do cliente. Como todos os processos, este também deve ser auditado para identificar não conformidades e pontos de melhoria, aumentando assim a qualidade do processo e, conseqüentemente, do produto final.

Palavras-chave: Qualidade. Desenvolvimento de Software. Processo de Homologação.

ABSTRACT

The software market has grown significantly in recent years and more and more companies are entering in this line of business. Customers are becoming more demanding and looking a quality product to solve your problems. Companies, in turn, improve their development processes using methodologies of software engineering. In this work was studied the software development stages, describing briefly the activities that are executing on each of these steps. It highlights the importance of software quality and the client validation activity as an important point for quality assurance. It was conducted a case study about of the implementation of the validation process in a software company, analyzing the data of the incidents registered during the period of one year before the implantation of the validation process and the incidents registered during the first year after the implantation of the process. In addition, a diagnose of the importance of a process of validation, followed by the software company together with the customer was done; this attendance during the validation process has increased the quality of the product installed in the production environment, reducing errors that have a direct impact in the customer operation. Even as any other process, it also should be audited to identify non-conformities and improvement points, thus increasing the quality of the process and, therefore, the final product.

Keywords: Quality. Software Development. Validation Process.

LISTA DE SIGLAS

UML – Unified Modeling Language

PML – Plano de Mudança e liberação

PHV – Plano de Homologação de Versão

PAV – Plano de Atualização de versão

RAV – Relatório Atualização de Versão

LISTA DE ILUSTRAÇÕES

Figura 1.	Processo de levantamento e análise de requisitos.....	17
Figura 2.	Tradução do modelo de análise para um modelo de projeto.....	19
Figura 3.	Custo relativo para corrigir um erro.....	25
Figura 4.	Fluxograma do processo de homologação anterior.....	26
Figura 5.	Fluxograma processo de homologação novo	28

LISTA DE GRÁFICOS

Gráfico 1.	Total de SALT abertas	32
Gráfico 2.	Total de SALT na 1ª semana de atualização.....	33

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	PROBLEMATIZAÇÃO	13
1.2	OBJETIVOS	14
1.2.1	Objetivos Gerais	14
1.2.2	Objetivos Específicos.....	14
1.3	JUSTIFICATIVA	14
1.4	PROCEDIMENTOS METODOLÓGICOS.....	15
1.5	ESTRUTURA DA MONOGRAFIA	15
2	REVISÃO DA LITERATURA.....	16
2.1	PROCESSO DE SOFTWARE	16
2.2	PROCESSO DE VERIFICAÇÃO E VALIDAÇÃO	20
2.2.1	Casos de Testes	23
2.3	GESTÃO DE QUALIDADE.....	23
2.4	CONSIDERAÇÕES FINAIS.....	25
3	ESTUDO DE CASO: PROCESSO DE HOMOLOGAÇÃO	26
3.1	PROCESSO DE HOMOLOGAÇÃO ANTERIOR.....	26
3.1.1	Problemas identificados	27
3.2	PROCESSO DE HOMOLOGAÇÃO NOVO.....	27
3.2.1	Melhorias identificadas.....	31
3.3	DADOS LEVANTANDOS	32
4	CONCLUSÕES E TRABALHOS FUTUROS.....	34
	REFERÊNCIAS	36

1 INTRODUÇÃO

A sociedade mundial avança cada dia mais quanto ao uso das tecnologias para auxiliar nas suas tarefas do cotidiano. Observa-se assim, que os sistemas informatizados tornam-se cada vez mais importantes e essenciais às empresas e às diversas organizações. Por outro lado é bastante visível a dificuldade das indústrias de softwares para se adequarem às necessidades constantemente crescentes do setor produtivo.

Na prática efetiva das empresas de desenvolvimento de softwares o “cliente”, ao solicitar um produto, mostra sua necessidade de resolver um problema e cabe à empresa contratada criar uma solução para o referido problema. Até a conclusão do desenvolvimento do produto final são diversas as etapas que são cumpridas para atender as necessidades dos seus usuários e a solução do problema ser entregue, de acordo com o contratado.

No processo de software segundo Sommerville (2003, p. 7), seguem-se as seguintes fases:

Especificação do software: A funcionalidade do software e as restrições em sua operação devem ser definidas.

Desenvolvimento do software: O software deve ser produzido de modo que atenda a suas especificações.

Validação do software: o software tem de ser validado para garantir que ele faz o que o cliente deseja.

Evolução do software: O software deve evoluir para atender às necessidades mutáveis do cliente.

Existem diferentes métodos de desenvolvimento de software que definem as atividades que são desenvolvidas em cada uma dessas fases, mas cada metodologia atende estas fases descrevendo diferentes atividades em cada uma delas.

A atividade de validação conforme Sommerville (2003) é a etapa em que o cliente recebe o produto, realiza as validações nele, de acordo com o que solicitou em contrato, e verifica se os requisitos levantados e implementados no produto entregue atendem suas necessidades. Após a aprovação do produto entregue, o cliente poderá realizar a implantação do software em seu ambiente para utilização em produção; problemas encontrados pelo cliente no produto que não impeçam seu trabalho, ou sejam problemas de baixo impacto na operação, serão tratados na evolução do software.

Assim que o produto foi implantado e inicia sua utilização pelo cliente, problemas que não foram encontrados durante a fase de validação podem impactar as atividades rotineiras, sejam eles falhas de definições de requisitos, desenvolvimento divergente às regras definidas ou *bugs* no código, estes precisarão passar pela etapa de manutenção para realização

de ajustes e evolução do software, para melhor atender ao cliente. Dependente do tamanho do projeto ou criticidade do software esses problemas não encontrados durante a validação podem parar as atividades do cliente, sendo então esta uma das etapas fundamentais no processo de desenvolvimento.

1.1 PROBLEMATIZAÇÃO

As empresas de software disponibilizam aos seus clientes novas versões do seu produto conforme definido em contrato e de acordo com a organização das empresas desenvolvedoras. Nestas versões estão correções de erros, alterações solicitadas pelo cliente, alterações devido à legislação e melhorias identificadas pela empresa. Antes da entrega da nova versão ao cliente, as empresas necessitam realizar seus testes para entregar seu produto com qualidade e evitar transtornos quando o sistema for atualizado por seus clientes.

Recomenda-se que o cliente tenha um ambiente para realizar a homologação ao receber uma nova versão, sendo neste ambiente realizada uma bateria de testes para validar que o sistema não tem erros que impactem no andamento da produção.

Entende-se por homologação uma auditoria para verificar produtos que são submetidos a testes para ver se atendem às exigências estabelecidas por uma autoridade competente. A esse respeito Tatsch (2010) esclarece:

O ambiente de homologação é um ambiente onde o cliente deverá testar as funcionalidades do sistema que serão posteriormente colocadas em produção ou refeitas, caso não ocorra à aprovação por parte deste cliente. Lembrando que este ambiente não é o servidor de testes para os desenvolvedores. O servidor a ser utilizado como ambiente de aceite, deverá possuir configuração similar ao utilizado como ambiente de produção, inclusive, com atualização constante dos registros existentes em produção, para que o cliente possa simular situações muito próximas das que encontrará no seu dia a dia.

No entanto, sabe-se que muitos clientes não possuem estrutura para ter outro ambiente ou demanda de pessoal que possam desprender este tempo para realizar esta atividade, com isso, quando atualizado o software em produção, possíveis falhas podem ser introduzidas e impactar diretamente na sua produção.

A ausência de um processo acompanhado de homologação gera impacto na produção, tendo como resultados atrasos em entregas ou paralização de atividades. Já para as empresas que desenvolvem é necessário um grande trabalho para correção destes erros para que o cliente possa realizar novamente seu trabalho, já que algumas vezes não basta retornar para a versão anterior, haja vista que nestas atualizações podem ocorrer mudanças na estrutura

do banco de dados, correndo o risco desta operação de retorno de versão ser viável apenas restaurando backup da base de dados.

1.2 OBJETIVOS

Os objetivos são apresentados em objetivos gerais e específicos.

1.2.1 Objetivos Gerais

Reconhecer a importância de um processo de homologação acompanhado dentro das empresas de software e realizar um estudo de caso com a análise e proposta de melhoria no processo de homologação de uma empresa de software, possibilitando maior qualidade no software atualizado em produção.

1.2.2 Objetivos Específicos

- Pesquisar na literatura as boas práticas para homologação de software.
- Relacionar os resultados apresentados com o processo de homologação da empresa estudo de caso.
- Propor um novo processo de homologação com as recomendações de melhorias.
- Analisar os resultados obtidos com o levantamento do número de atendimentos de erro após a atualização em produção antes e após a implantação do novo processo de homologação.

1.3 JUSTIFICATIVA

O uso de software se tornou cada vez mais imprescindível nas organizações e no nosso dia-a-dia. O software como um produto desenvolvido pelo ser humano é passível de falhas pois muitas vezes quando o cliente solicita o desenvolvimento ainda tem dúvidas sobre suas necessidades e regras envolvidas, e durante o processo de desenvolvimento ocorrem

diversas alterações até a entrega do produto quando o cliente realiza a instalação deste em seu ambiente.

Ao ser entregue o produto, o cliente realiza suas validações e verifica se o produto está adequado às suas necessidades e realiza sua instalação. Ao iniciar a sua utilização podem ocorrer falhas que podem parar suas atividades. Desta forma, o processo de validação deve ser criterioso para evitar impactos nas suas atividades e evitar prejuízos, principalmente financeiros.

1.4 PROCEDIMENTOS METODOLÓGICOS

Pesquisa é definida por Silva (2005, p. 20) como “um conjunto de ações, propostas para encontrar a solução para um problema, que têm por base procedimentos racionais e sistemáticos. A pesquisa é realizada quando se tem um problema e não se têm informações para solucioná-lo”. O planejamento e a execução de pesquisa compreendem o processo sistematizado, seguindo algumas etapas como: escolha do tema, revisão bibliográfica, justificativa, formulação do problema, determinação dos objetivos, metodologia, coleta de dados, tabulação de dados, análise e discussão dos resultados, conclusão da análise dos resultados e redação e apresentação do trabalho científico.

O estudo desenvolvido neste trabalho é de cunho exploratório, com abordagem de natureza qualitativa. O método de procedimento adotado é monográfico com estudo de caso na empresa Estudo de Caso, utilizando as técnicas de pesquisa bibliográfica, baseada em livros, artigos científicos e sites, e de pesquisa com base em dados *in loco*.

1.5 ESTRUTURA DA MONOGRAFIA

O capítulo um aborda a introdução, problematização, objetivos, justificativa e procedimentos metodológicos.

O processo de software, processo de validação e verificação, casos de testes e gestão da qualidade são abordados no capítulo dois.

O estudo de caso de processo de homologação em uma empresa de software, e os resultados deste processo são apresentados no capítulo três.

No capítulo cinco é apresentada as conclusões finais e sugestão de trabalhos futuros.

2 REVISÃO DA LITERATURA

Este capítulo aborda os seguintes assuntos: processo de software, processo de validação e verificação, gestão de qualidade de software.

2.1 PROCESSO DE SOFTWARE

A construção de um software é um processo complexo e trabalhoso, para tanto a Engenharia de Software apresenta modelos que auxiliam o processo de desenvolvimento, organizando as fases e atividades que serão desenvolvidas.

O desenvolvimento começa com engenharia de requisitos. Podemos definir que esta fase trata de uma ponte ao projeto de software, na qual se tem a concepção do problema a ser solucionado e deve-se estabelecer uma solução para o mesmo. Na definição de Pressman (2006, p. 117),

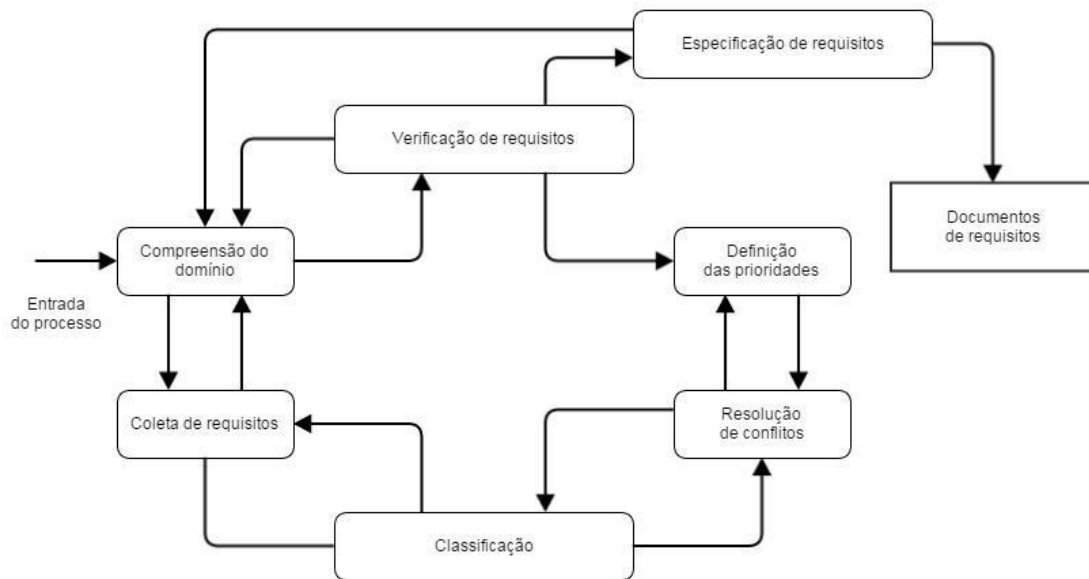
[...] a passagem pela ponte nos leva muito acima do projeto, permitindo à equipe de software examinar o contexto do trabalho de software a ser realizado; as necessidades específicas que o projeto e a construção precisam satisfazer; as prioridades que estabelecem a ordem na qual o trabalho deve ser completado e a informação, funções e comportamentos que terão um impacto no projeto resultante.

Durante esta fase são realizadas algumas atividades, dependendo de cada projeto. Para Presmann (2006, p. 118) essas atividades são definidas nas seguintes funções: “*concepção, levantamento, elaboração, negociação, especificação, validação e gestão*”.

De acordo com Presmann (2006, p.118) na **Concepção**, “a intenção é estabelecer um entendimento básico do problema, o pessoal que quer uma solução, a natureza da solução desejada e a efetividade da comunicação e colaboração preliminar entre cliente e desenvolvedor”.

Já no **Levantamento** tem-se o objetivo de levantar as informações com os usuários, clientes e outros interessados para definição dos limites do projeto, entendimento do problema. As pessoas que estão envolvidas no levantamento são chamadas de stakeholders. O levantamento é realizado através de questionários, entrevistas e observações aos processos em execução e documentação utilizada. A seguinte imagem, Figura 1, apresenta a representação do fluxo das atividades que são desenvolvidas nesta etapa, segundo Sommerville (2006, p. 105 e 106). Sendo necessário compreender o que será desenvolvido para poder realizar a coleta e classificação dos requisitos, resolver os conflitos dos requisitos com os stakeholders, definir as prioridades e realizar a verificação dos requisitos levantados.

Figura 1. Processo de levantamento e análise de requisitos.



Fonte: Adaptado de SOMMERVILLE (2006, p.106)

Na etapa de **Elaboração** as informações levantadas são utilizadas para iniciar o processo de criação da documentação técnica, de forma a criar uma modelagem do sistema. Para a documentação técnica são utilizados diversos diagramas da UML, que darão suporte ao desenvolvimento do software.

Já no processo de **Negociação** são evidenciadas algumas solicitações de desenvolvimento que não possuem conformidade com a possibilidade do projeto, problema esse decorrente por diversos fatores como custo, tempo ou tecnologia. No entanto deve-se negociar o escopo mais adequado ao projeto que está em desenvolvimento.

Quanto a **Especificação**, Presmann (2006, p. 118) define que “[...] é o produto de trabalho final produzido pelo engenheiro de requisito. Ele serve como fundamento das atividades de engenharia de software subsequente. Ela descreve a função e o desempenho de um sistema baseado em computador e as restrições que governarão o seu desenvolvimento”. Esta documentação deve ser clara e de fácil entendimento tendo em vista que será verificada pelos stakeholders que podem não ter domínio técnico, mas sim do negócio.

Logo após, a **Validação** tem a função de revisar a documentação elaborada, sendo realizada por: engenheiros de requisitos, clientes e usuários. Neste sentido Presmann (2006, p.120) contribui afirmando que,

A validação de requisitos examina a especificação para garantir que todos os requisitos do software tenham sido declarados de modo não ambíguo; que as inconsistências, omissões e erros tenham sido detectados e corrigidos e que os

produtos de trabalho estejam de acordo com as normas estabelecidas para o processo, o projeto e o produto.

A atividade de validação é a atividade desempenhada para que seja elevada a garantia de qualidade. Os erros descobertos nesta fase irão contribuir para que sejam corrigidos os problemas apresentados, na sua maioria, com menor custo, tendo em vista que o requisito não passou para as demais fases do desenvolvimento do software. Sommerville (2003, p. 116) destaca as principais verificações que são feitas na atividade de validação, sendo elas:

1. *Verificação de validade* Um usuário pode pensar que um sistema é necessário para realizar certas funções. Contudo, mais estudos e análise podem identificar funções adicionais ou diferentes, que são exigidas. Os sistemas têm diversos usuários com necessidades diferentes e qualquer conjunto de requisitos é inevitavelmente uma solução conciliatória da comunidade de usuários.
2. *Verificações de consistência* Os requisitos em um documento não devem ser conflitantes, ou seja, não devem existir restrições contraditórias ou descrições diferentes para uma mesma função do sistema.
3. *Verificação de completeza* O documento de requisitos deve incluir requisitos que definam todas as funções e restrições exigidas pelo usuário do sistema.
4. *Verificações de realismo* Utilizando o conhecimento da tecnologia existente, os requisitos devem ser verificados, a fim de assegurar que eles realmente podem ser implementados. Essas verificações devem também levar em conta o orçamento e os prazos para o desenvolvimento do sistema.
5. *Facilidade de verificação* Para reduzir o potencial de divergências entre cliente e fornecedor, os requisitos do sistema devem sempre ser escritos de modo que possam ser verificados. Isso significa que um conjunto de verificações pode ser projetado para mostrar que o sistema entregue cumpre com esses requisitos.

Para maximizar a satisfação do cliente na explicação de Presmann (2006, p. 127), pode-se realizar “A *Implantação da Função de Qualidade (IFQ) (QualityFunction Deployment, QFD)* que é uma técnica que traduz as necessidades do cliente para requisitos técnicos do software”. Podemos elencar os requisitos nas seguintes formas:

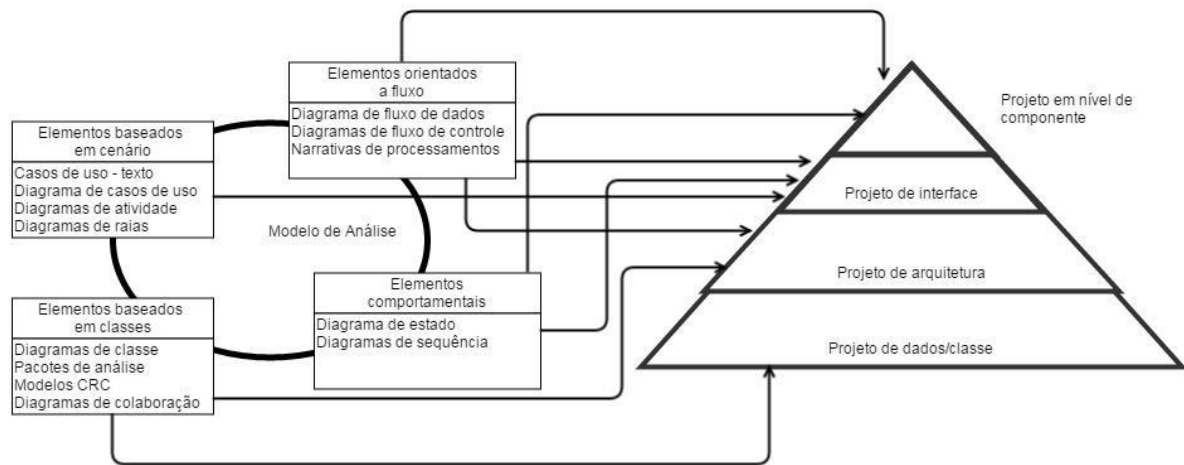
Requisitos normais: são os requisitos que foram acordados em reuniões com o cliente, que refletem os objetivos do produto.

Requisitos esperados: são os requisitos essenciais que o cliente não solicita, mas se não estiver presente no produto causara insatisfação. Como performance e facilidade de utilização.

Requisitos exitantes: não foram solicitados pelo cliente, mas seu desenvolvimento aumenta a satisfação do cliente.

Logo após o processo de engenharia de requisitos se inicia o projeto de software, localizado no centro da engenharia de software, sendo aplicado independente do modelo de desenvolvimento utilizado, como evidenciado na Figura 2, Presmann (2006, p. 187).

Figura 2. Tradução do modelo de análise para um modelo de projeto



Fonte: Adaptado de PRESMANN (2006, p. 187).

Os elementos produzidos durante a análise são base para produção da documentação do projeto. Parafraseando as ideias de Sommerville (2003, p. 27) neste processo são realizadas as seguintes atividades:

Agrupar os requisitos: Os requisitos são analisados e reunidos em grupos relacionados.

Identificar subsistemas: São identificados diferentes subsistemas que podem, individual ou coletivamente, atender aos requisitos.

Atribuir requisitos a subsistemas: Os requisitos são atribuídos a subsistemas.

Especificar a funcionalidade de subsistemas: As funções específicas fornecidas por cada subsistema são definidas.

Definir as interfaces de subsistemas: Essa tarefa envolve a definição das interfaces que são fornecidas e exigidas por cada subsistema.

A importância do projeto de software é destacada por Presmann (2006, p.187) ao defini-lo como,

[...] única palavra – *qualidade*. Projeto é a etapa na qual a qualidade é incorporada na engenharia de software. O projeto nos fornece representações do software que podem ser avaliadas quanto à qualidade. É o único modo pelo qual podemos traduzir precisamente os requisitos do cliente em um produto ou sistema de software acabado. O projeto de software serve de base para todos os passos de engenharia de software e de suporte de software que se seguem.

Evidenciamos que a engenharia de software tem evoluído quanto aos conceitos de projeto de software. Parafraseando Presmann (2006, p. 190-197), esses conceitos são:

Abstração: é demonstrar a solução de forma abstrata, podendo utilizar a linguagem do domínio do problema que está sendo resolvido, fazendo com que os envolvidos consigam compreender o que deve ser desenvolvido.

Modularidade: para resolver os problemas que são complexos podemos dividi-los em componentes que se integram, facilitando a resolução dos problemas. Assim, devem-se organizar estes módulos nomeando-os e interligando-os. Para tanto, o número de módulos deve ficar na média para não aumentar os custos, pois quanto mais módulos e mais integrações os custos de desenvolvimento aumentam.

Ocultamento da informação: os módulos são projetados de forma a serem específicos e seu conteúdo não deve ser visualizado por outros módulos, no entanto será visível apenas aos módulos que se integram, no que se refere aos elementos essenciais, este ocultamento de informação facilita a manutenção e testes que são realizados.

Independência funcional: quanto mais independentes forem os módulos se facilita sua implementação e manutenção. Para avaliar qualitativamente um projeto utiliza-se dois critérios: sua coesão, sendo que cada módulo realiza uma tarefa e requer pouca interação com demais módulos, e o acoplamento, que trata-se da conexão entre os componentes, devendo ser mais simples evitando problemas propagados no desenvolvimento.

Refinamento: durante a elaboração são descritas as funções de forma objetiva e em cada refinamento a descrição vai sendo mais detalhada e abstrata, deve-se então conceituar a função e não descrever seu funcionamento.

Refabricação: ao refabricar um software os algoritmos são analisados para identificar redundâncias e elementos não utilizados, como também identificar problemas de construção ou falhas de projeto que devem ser corrigidas.

Classes de Projeto: no desenvolvimento do projeto são definidos o conjunto de classes que fornecem detalhes do projeto, fornecendo base para a implementação da interface do usuário, os métodos e regras de negócio e a persistência de dados.

Após o desenvolvimento inicia-se o processo de instalação de sistema, neste processo o sistema é colocado no ambiente em que será operado.

2.2 PROCESSO DE VERIFICAÇÃO E VALIDAÇÃO

A verificação e validação (V & V) são definidas por Sommerville (2003, p.358) como “o nome dado aos processos de verificação e análise que asseguram que o software

cumpra com suas especificações e atenda às necessidades dos clientes que estão pagando por ele”.

Sendo que as atividades de cada um destes processos são diferentes, conforme Presmann (2006, p. 289) a “*Verificação* se refere ao conjunto de atividades que garante que o software implementa corretamente uma função específica.” Já a norma da ISO 12207 define que “o processo de validação é um processo para determinar se os requisitos e o produto final, sistema ou produto de *software* construído, atendem ao uso específico pretendido.”

No processo de V & V, Presmann (2006, p. 358) descreve duas técnicas utilizadas para realizar a análise do sistema, sendo elas:

1. *As inspeções de software*, que analisam e verificam as representações do sistema, como o documento de requisitos, os diagramas de projeto e o código-fonte do programa. As inspeções podem ser aplicadas em todos os estágios do processo e ser complementadas por alguma análise automática do texto de origem de um sistema ou dos documentos associados. As inspeções de software e as análises automatizadas são técnicas estáticas de V & V, uma vez que não requerem que o sistema seja executado.
2. *Os testes de software*, que envolvem executar uma implementação do software com os dados de teste e examinar as saídas dele e seu comportamento operacional, a fim de verificar se ele está sendo executado conforme o esperado. Os testes são uma técnica dinâmica de verificação e validação porque trabalham com uma representação executável do sistema.

Este processo de V & V deve ser realizado desde o início do desenvolvimento do projeto de software até sua entrega final, por pessoas da empresa, outras organizações e o cliente que solicitou o produto. A realização adequada deste processo tem a função de garantir a qualidade do produto, reduzindo os custos das correções dos defeitos e aumentando a confiança do cliente no produto recebido.

O processo de verificação é realizado por auditorias do processo, revisões técnicas formais que, conforme Presmann (2006, p. 583), têm como objetivo principal “achar erros durante o processo, de modo que eles não se transformem em defeitos depois da entrega do software. O benefício [...] é a descoberta antecipada de erros, de forma que eles não se propaguem para o passo seguinte do processo de software.”

No entanto, para realizar a validação segundo a ISO 12207 (1998, p. 22) é necessária a elaboração de um plano que deve ser documentado incluindo os seguintes itens:

- a) Itens sujeitos à validação;
- b) Tarefas de validação a serem executadas;
- c) Recursos, responsabilidades e cronograma para validação; e
- d) Procedimentos para encaminhar relatórios de validação ao adquirente e outras partes envolvidas.

Como todo o processo de desenvolvimento de software, a validação também deve ser auditada, elencando as não conformidades e apontando melhorias para garantir melhor qualidade no produto entregue ao seu cliente.

Na tarefa de validação do processo, com a documentação de requisitos e casos de testes e um ambiente para realização do plano de teste, a ISO 12207 (1998, p.23) sugere que se testem conforme a seguir:

- a) Teste de estresse, limites e entradas específicas.
 - b) Teste do produto de *software* para verificar sua habilidade em isolar e minimizar efeitos de erros; isto é, degradação suave em caso de falha, pedido de assistência do operador em caso de estresse, de exceder limites e de condições específicas.
 - c) Teste para que usuários representativos possam executar, com sucesso, suas tarefas pretendidas usando o produto de *software*.
- 6.5.2.4 Validar que o produto de *software* satisfaça seu uso pretendido.
- 6.5.2.5 Testar o produto de *software*, quando apropriado, nas áreas selecionadas do ambiente-alvo.

No teste de validação existe o teste de aceitação, sendo esta etapa de responsabilidade do usuário do software ou clientes; nesta etapa o cliente realiza seus testes validando se o que requisitou foi entregue de forma adequada atendendo suas necessidades. Esta etapa compreende a homologação do software entregue ao cliente, para avaliar se a versão entregue está adequada para realizar a instalação em seu ambiente de trabalho ou produção.

Incluir o teste de aceitação no processo de desenvolvimento do software, segundo Bastos (2012, p. 204) define algumas vantagens:

- Detecção antecipada de possíveis defeitos, permitindo, inclusive, que novos prazos venham a ser negociados. Lembre-se de que, quanto antes for encontrado um defeito, mais barato será corrigi-lo.
- A participação do usuário também no processo de teste permitirá mais eficiência em sua execução, na preparação do Plano de Teste e, depois, nos casos de teste. O usuário também poderá ser um importante aliado no momento de procurar recursos e ferramentas.
- Detecção de novas necessidade e definição de novos requisitos pelo usuário.

A realização deste processo ocorre de forma paralela ao processo de desenvolvimento do software, neste são realizadas as seguintes atividades parafraseando Bastos (2012, p. 204):

- Elaboração do plano de teste com a definição dos critérios de aceitação;
- Definição de um ambiente adequado para realização dos testes, preferencialmente deve ser o mais semelhante ao ambiente em que será instalado;
- Criação de um cronograma com prazos e o produto que será testado;

Documentar o relatório das não conformidades e defeitos encontrados para que sejam realizadas as manutenções necessárias para o aceite;

Monitorar as correções dos defeitos identificados;

Confirmar o aceite do produto entregue.

O teste de aceitação pode ser realizado durante todo o processo de desenvolvimento, sendo conduzido por um grupo de usuários capacitados, o grupo pode ser formado por usuários do cliente ou uma equipe especializada é contratada a fim de inspecionar o produto. A realização de testes pode ser feita em produtos intermediários ou no produto final entregues para realização da tarefa.

2.2.1 Casos de Testes

Assim como a especificação de requisitos os casos de testes são um documento que especifica os procedimentos do teste. Bastos (2012, p. 153) recomenda que um bom caso de teste deva conter os seguintes itens:

- Identificação das condições de testes:
- Pré-condições;
- Pós-condições;
- Critério de aceitação.
- Identificação dos casos de testes (o que testar).
- Detalhamento da massa de entradas e de saída.
- Critérios especiais, caso necessários, para a geração da massa de teste.
- Especificação das configurações de ambiente no qual o teste será executado: sistema operacional, ferramentas necessárias, origem dos dados etc. (onde testar).
- Definir o tipo de implementação do teste: automática/manual (como testar).
- Definir o cronograma, ou seja, em qual fase esse teste será executado (quando testar).
- Listar as interdependências, caso existam, entre os casos de teste.

O caso de testes deve ser claro para permitir que a validação da funcionalidade do software atenda aos requisitos que foram levantados e desenvolvidos.

2.3 GESTÃO DE QUALIDADE

O dicionário Michaelis define qualidade como “1- Atributo, condição natural, propriedade pela qual algo ou alguém se individualiza, distinguindo-se dos demais; maneira de ser, essência, natureza. 2- Excelência, virtude, talento. [...] 4-Grau de perfeição, de precisão, de conformidade a um certo padrão”.

No entanto, Presmann (2006, p.578) destaca que: “a qualidade se refere a características mensuráveis – coisa que nós podemos comparar com padrões conhecidos tais como comprimento, cor, propriedades elétricas e maleabilidade. Todavia, o software, que é essencialmente uma entidade intelectual, é mais difícil de caracterizar que objetos físicos.”

Algumas características que podem ser utilizadas para medir um software são: número de linhas de código, número de pontos por função, conformidade com requisitos, normas específicas do processo de desenvolvimento, satisfação do cliente.

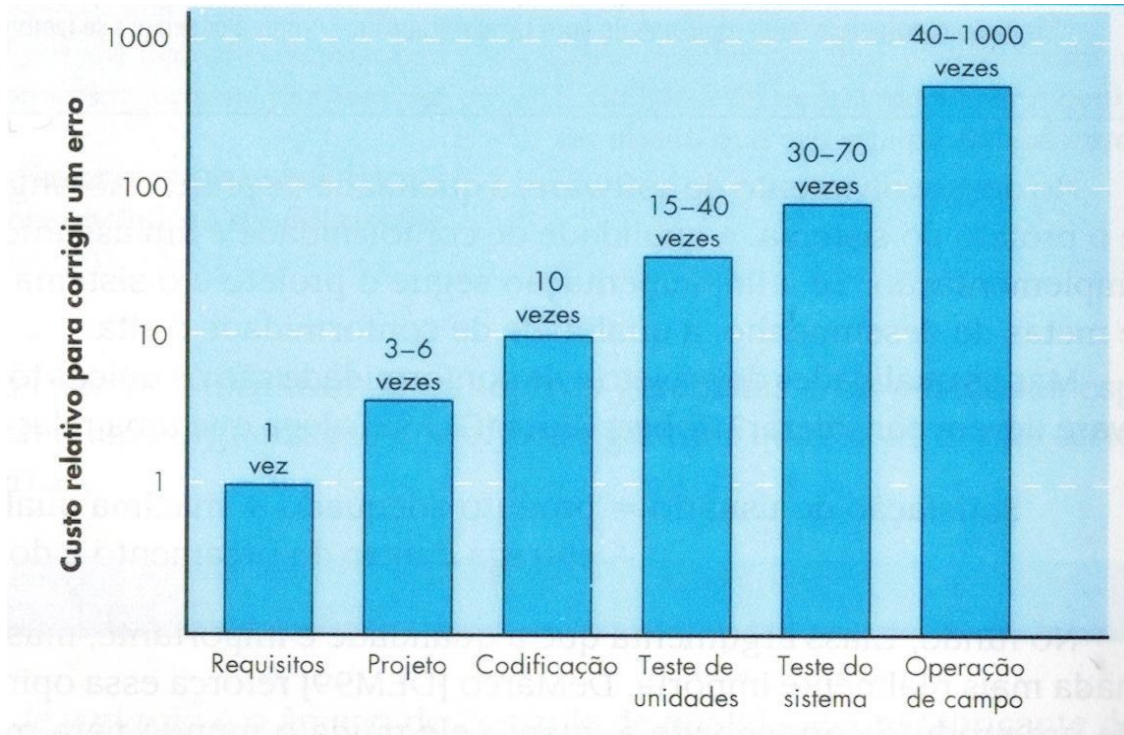
Presmann (2006, p. 578) divide a qualidade em duas espécies, “*Qualidade de projeto* refere-se a características que os projetistas especificam para um certo item. *Qualidade de conformidade* é o grau com que as especificações do projeto são seguidas durante a fabricação.”

Realizar o controle da qualidade consiste em monitorar e controlar as especificações e mensurações definidas no processo de desenvolvimento, fazendo comparativo com resultados anteriores a fim de reduzir os defeitos desenvolvidos. (PRESMANN, 2006)

Conforme Presmann (2006) é através do controle podemos fazer a garantia da qualidade, realizando auditorias durante o processo, avaliando a realização das atividades de acordo com as definições e padrões, a completeza de cada uma destas e se atende os requisitos definidos.

Na gestão da qualidade também deve ser avaliado o custo da qualidade dentro do projeto, conforme vai avançando as fases de desenvolvimento os custos para correção de falhas aumentam, então quando antes os erros forem mapeados e corrigidos menor serão seus custos; a Figura 3exibe um gráfico com a relação de custos com as fases do processo de desenvolvimento. Presmann (2006, p.579) classifica e identifica os custos em: “custos de prevenção incluem planejamento da qualidade, revisões técnicas formais, equipamento de teste, treinamento. Os custos de avaliação incluem atividades para obter entendimento da condição do produto na “primeira execução” de cada projeto.”

Figura 3. Custo relativo para corrigir um erro.



Fonte: Presmann (2006, p. 580)

2.4 CONSIDERAÇÕES FINAIS

Ao analisar o processo de desenvolvimento podemos perceber que a engenharia define etapas no desenvolvimento com atividades específicas em cada uma destas, as etapas se dividem em "*concepção, levantamento, elaboração, negociação, especificação, validação e gestão*", destaca-se a importância da validação e verificação durante todo o processo, sendo esta atividade deve ser realizada tanto pela empresa desenvolvedora quanto o cliente, para que os erros sejam encontrados na fase inicial do processo gerando menor custo de sua correção.

A realização de testes é mostrada como item importante para a qualidade do produto entregue, para tanto é necessário a criação de casos de testes para servir como guia e critério de aceite do produto.

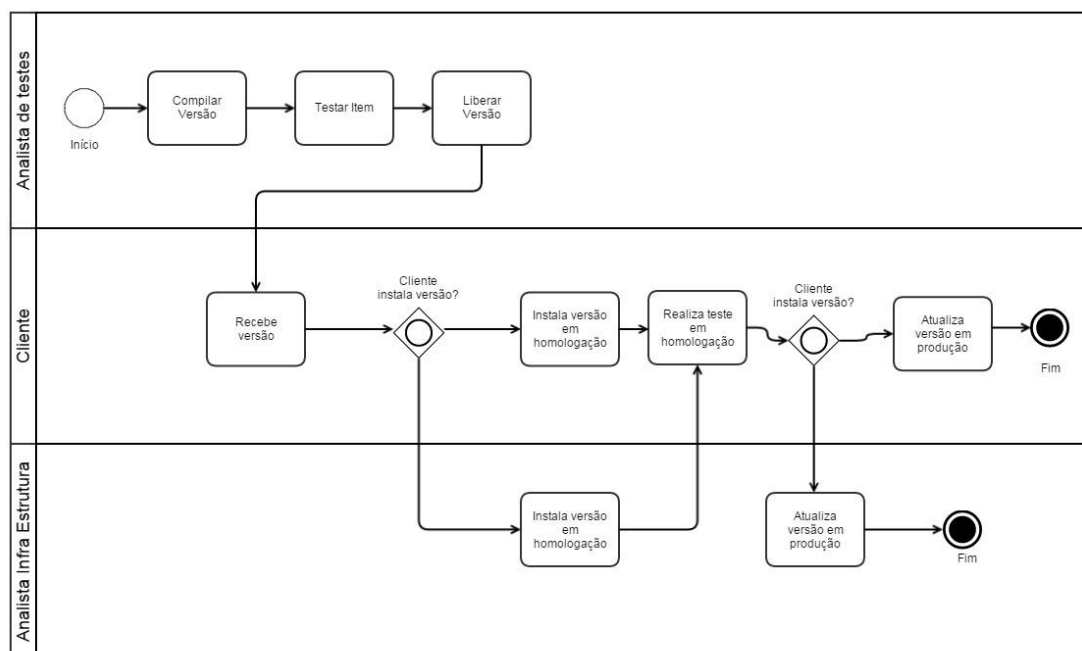
3 ESTUDO DE CASO: PROCESSO DE HOMOLOGAÇÃO

Como estudo de caso do processo de homologação foi utilizada a empresa Softplan com o sistema de tribunal de justiça de primeiro grau. Inicialmente é definido o processo de homologação anterior, depois é definido o processo de homologação novo.

3.1 PROCESSO DE HOMOLOGAÇÃO ANTERIOR

A empresa não tinha um processo de homologação bem definido com uma equipe com atividades exclusivas para o processo de homologação, bem como um acompanhamento mais frequente da realização deste processo pelo cliente. A Figura 4 mostra o fluxograma do processo de homologação de versão seguida das atividades descritas.

Figura 4. Fluxograma do processo de homologação anterior



Fonte: o autor, 2015

No processo anterior a equipe de desenvolvimento liberava os itens da versão para equipe de testes, a qual realizava os testes individualizados para aprovar cada item. Após os itens aprovados e de acordo com o funcionamento a versão era compilada pela equipe de testes.

Com a versão compilada, então, era liberada para o cliente, sendo que este era comunicado via e-mail informando as alterações e novas implementações desenvolvidas para que a versão pudesse ser testada.

O cliente estando com a versão nova poderia fazer a atualização em seu ambiente de homologação. Neste ambiente realizava os testes para simular operações que são realizadas na produção, bem como validar as alterações e nova implementações desenvolvidas na versão em homologação.

Assim que concluída a etapa de testes pelo cliente, este realizava a solicitação da atualização da versão em produção. Para os clientes que possuem contrato, a atualização é realizada pela equipe de infraestrutura da empresa desenvolvedora; para os demais clientes a atividade é realizada por equipe especializada do próprio cliente.

3.1.1 Problemas identificados

Após análise do processo anterior com a literatura descrita no capítulo 2 verifica-se que há falta de auditorias no processo que era realizado. Não existia um plano de testes com os itens que deveriam ser testados para garantir a qualidade do software e os critérios de aceite.

Não ocorria o acompanhamento da homologação realizada pelo cliente, de forma que após a entrega da versão ao cliente ele poderia realizar este processo de validação ou não, solicitando diretamente a instalação da versão no ambiente de produção ou instalando diretamente neste ambiente, sem realização de testes, causando grandes impactos aos usuários finais e ao negócio do cliente após instalação.

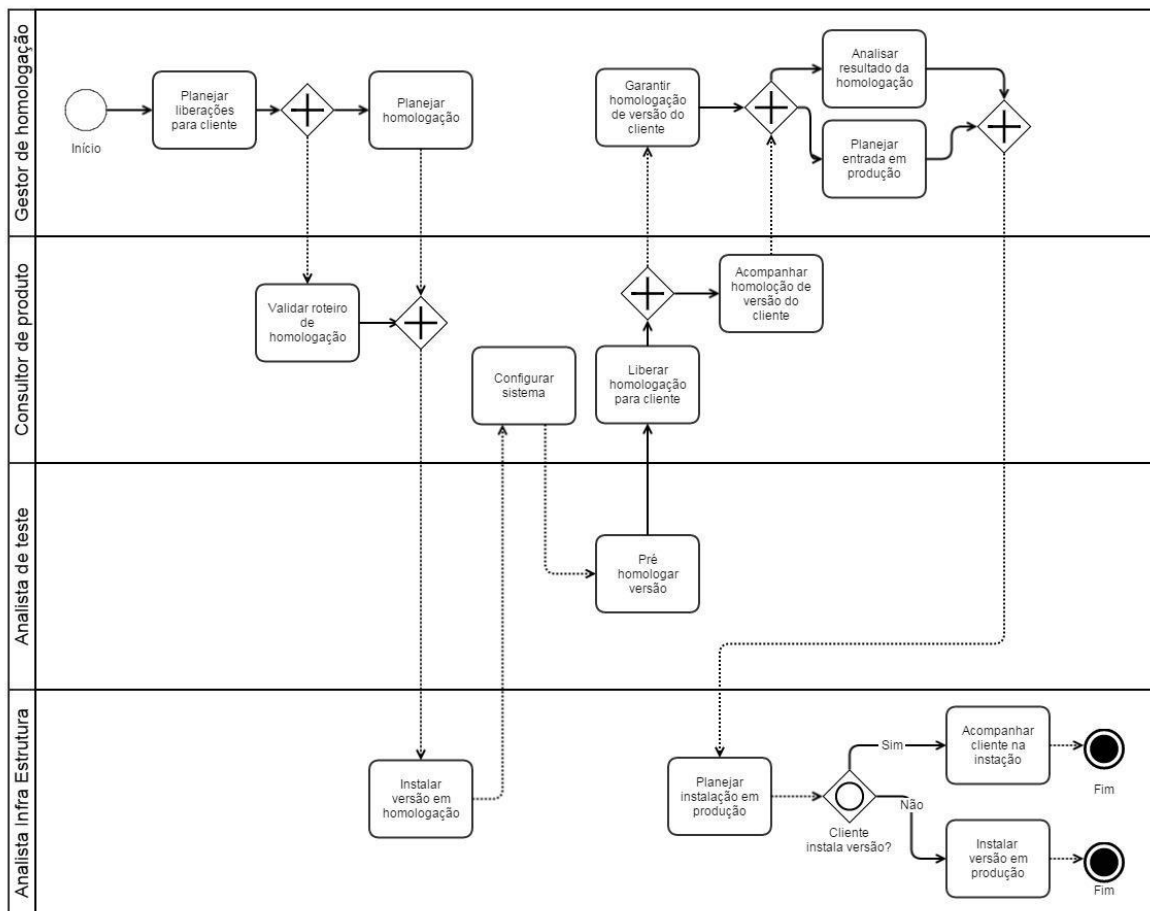
A equipe de testes não realizava um teste completo no software para validar que todas as funcionalidades estavam funcionando corretamente, conforme já ocorriam na versão anterior do sistema; as integrações também não eram testadas de forma completa, garantindo que na nova versão ainda estavam funcionando adequadamente.

3.2 PROCESSO DE HOMOLOGAÇÃO NOVO

Após estudos pela equipe de qualidade e gerencias criou-se um processo de homologação criando uma equipe especializada para as tarefas de homologação, bem como tarefas bem definidas para melhorar a qualidade do produto recebido pelo cliente e atualizado em produção. Essa equipe é formada por gestor de homologação que coordena a equipe e as

homologações a serem realizadas e analistas que realizam os testes. Para execução do processo de homologação definiu-se o fluxograma apresentado na Figura 5 com o novo processo de homologação e suas atividades descritas abaixo.

Figura 5. Fluxograma processo de homologação novo



Fonte: o autor, 2015

Para iniciar o processo de homologação o Gestor de Homologação elabora um cronograma da homologação, esta atividade é iniciada assim que a equipe de desenvolvimento realiza a liberação de uma versão cheia. Nesta atividade inicial de *planejar a liberação ao cliente* são definidas as datas de atualização em ambiente de homologação, cronograma das atividades da homologação e previsão de data para realizar a atualização em produção.

Em seguida o Gestor de Homologação na *atividade planejar homologação* define os critérios para aceite da homologação, realizando a negociação com o cliente e equipe interna para instalação e configuração da versão de homologação, comunica o cliente o cronograma da homologação e solicita à equipe responsável a instalação da versão em homologação.

A definição dos critérios de aceite é realizada em conjunto com o consultor do cliente, sendo que os resultados são classificados como “Reprovado que impede entrada em produção”, estes são erros, de acordo com o cliente e equipe interna, que impedem o trabalho em produção e devem ser corrigidos para finalizar a homologação; “Reprovado que pode ser tratado posteriormente” que são problemas de menor impacto que podem ser corrigidos após a homologação, desde que a quantidade de erros nesta classificação não ultrapasse 10% dos itens durante o processo; e “Aprovado” que não apresentaram problemas nos testes realizados.

Na negociação com o cliente será definido o melhor período para realização do processo, quais são as pessoas responsáveis para comunicação e aprovação de aceite da homologação, os responsáveis para atividade de testes no cliente e o roteiro a ser seguido. Se existirem integrações, as entidades que estiverem envolvidas devem ser integradas e comunicadas neste processo.

Como integrante do processo, o Consultor de produto irá *validar o roteiro de homologação*, caso para o sistema do cliente ainda não exista um roteiro, este deve ser elaborado, ele deve estar atualizado contendo as lições aprendidas em homologações e testes anteriores, deve indicar os testes que devem ser realizados simulando o ambiente de produção, indicando a necessidade de simulações de carga e stress e o comportamento esperado do software, principalmente para avaliar o desempenho em situações críticas de processamento. Caso o roteiro contenha alterações e novas implementações ele deve destacar os testes neste ponto do software e demais pontos que podem ter sido afetados pela alteração.

Assim que for autorizado e acordado com o cliente inicia-se a atividade de *instalar versão em homologação*, a equipe responsável pela atividade cria um PAV – Plano de atualização de versão –, que irá indicar o roteiro de atividades que devem ser feitas, apontando os scripts e executáveis que devem ser executados e disponibilizados no ambiente indicado; após a conclusão da atividade serão comunicados os interessados. Deve-se garantir que o sistema está operacional e registrar a instalação feita.

Após a instalação da versão de homologação a consultoria de produto ou equipe responsável no cliente irá *configurar o sistema* de acordo com a documentação da RAV – Roteiro Atualização de Versão, assim que concluída esta atividade o ambiente está preparado para realização dos testes.

Após o ambiente configurado o Analista de Teste realiza a *Pré homologação versão* nesta atividade; com o RHV (Roteiro de Homologação de Versão) e PHV (Plano de Homologação de Versão) serão realizados testes que devem ser evidenciados e registrados com a devida classificação, sendo eles “Aprovado”, “Reprovado – impede entrada em

produção” e “Reprovado – pode ser tratado posteriormente”, os erros encontrados devem ser encaminhados para equipe responsável para as devidas correções e os itens que não foram testados devem ser justificados.

Assim que a versão no ambiente de homologação estiver com uma versão com as correções dos itens identificados na pré homologação feitas, é então *Liberada homologação para o cliente*, comunicando-se formalmente via e-mail os responsáveis pela homologação no cliente, destacando a versão que será testada anexando a RAV e o PHV.

Com a comunicação formal ao cliente que a versão está liberada para homologação o Gestor de Homologação deve *Garantir homologação de versão do cliente*, em contatos com o cliente deve acompanhar a realização da homologação até que seja formalizado o TAE – Termo de aceite de entrega–, que indica que a versão foi homologada pelo cliente.

O consultor de produto deve acompanhar a homologação de versão do cliente, validar os resultados do RHV e os critérios de aceite para comunicação via e-mail com o cliente e envolvidos com o processo, acompanhar a homologação com o cliente conforme foi planejado no Plano de homologação. Os erros encontrados pelo cliente durante os testes feitos devem ser registrados em chamado para que a equipe de correção faça as correções e libere uma nova versão para conclusão da homologação. Ao final dos critérios de aceite serem atendidos registra-se a conclusão e aceite da homologação.

Ao final da homologação o Gestor de homologação deverá fazer a análise dos resultados da homologação, medindo e tabulando o número de erros encontrados conforme a classificação, tempo necessário para realização do processo e lições aprendidas.

Com o aceite da versão homologada pelo cliente inicia então o Planejamento da entrada em produção pelo Gestor de homologação, informar o período de interrupção do sistema ao cliente para que o mesmo comunique os interessados, agendar e solicitar o backup e plantão junto aos DBA, comunicar as equipes internas sobre a atualização programada, comunicar e deixar de sobreaviso as equipes de suporte ao cliente, monitoramento e DBAs para identificação e correções de problemas decorrente da atualização.

Com o agendamento da atualização o analista de infraestrutura deverá elaborar o PML – Plano de Mudança e liberação–, sendo preenchido conforme os dados indicados pelo Gestor de homologação, com informações pertinentes a scripts e executáveis, juntamente com o roteiro de atualização.

O analista de infraestrutura na data acordada com o cliente e a empresa deverá realizar a Instalação da versão em produção, para realização da atividade deve estar com

executáveis, scripts e demais arquivos a serem instalados, acompanhado da PML criada e o PHV. Ao final da instalação deve conferir se a versão está compatível e os executáveis disponíveis para atualização no cliente (usuário final), realizar login no sistema e verificar se está executando normalmente.

Assim que concluída a instalação o analista de infraestrutura deverá Acompanhar cliente na instalação, no período acordado com o cliente para atualizar deve-se acompanhar a realização da instalação e os envolvidos devem ser comunicados.

3.2.1 Melhorias identificadas

Após a implantação do processo de homologação os clientes começaram a se preocupar mais em realizar testes em homologação, bem como uma equipe especializada dentro da empresa realiza testes no ambiente de homologação, a fim de reduzir os erros introduzidos na atualização em produção que prejudicam a operação.

O primeiro dia após atualização da versão costumava ser complicado para o cliente, tendo em vista que algumas operações fundamentais apresentavam problema, ou falha de comportamento e a equipe de manutenção precisava realizar as correções para uma nova atualização do ambiente no mesmo dia.

Após a implantação do processo utilizando o plano de testes e com critérios de aceite bem definidos, as funcionalidades fundamentais são testadas e são itens que definem o aceite e atualização da versão, para que as operações do cliente não parem em produção.

As versões eram disponibilizadas para os clientes e havia muita diferença entre as versões que estavam atualizadas em produção entre eles, exigindo assim que os erros corrigidos fossem sincronizados em diferentes versões. Assim que a equipe iniciou o planejamento de homologação as versões entre os clientes ficaram mais homogêneas.

Os problemas encontrados após a entrada da versão em produção são elencados para verificação do motivo deles não terem sido identificados durante a etapa de homologação, para que nas próximas versões erros semelhantes não ocorram após atualização.

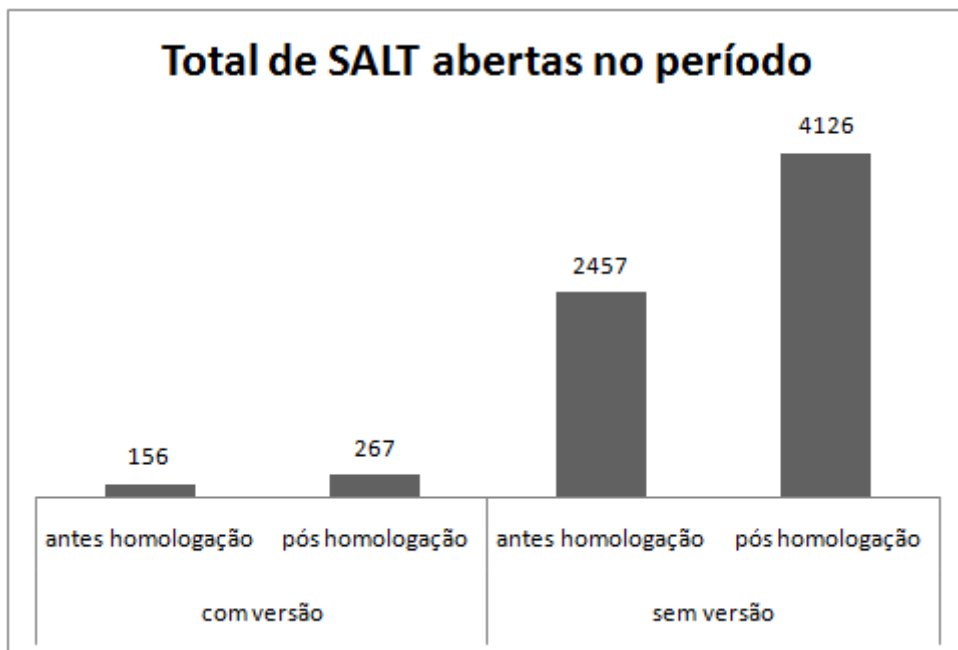
O processo de homologação continua sendo auditado e melhorias estão sendo estudadas para que a qualidade do software aumente, bem como a satisfação do cliente.

3.3 DADOS LEVANTANDOS

Como forma de verificação dos resultados do processo anteriormente descrito alguns dados foram coletados e analisados. Assim sendo foram levantadas as SALTs – Solicitação de Alteração – abertas, ou seja, os chamados abertos para o suporte de problemas na utilização de sistema, neste caso tratando apenas os abertos como erro de sistema. O período considerado foi de 01 de julho de 2012 a 31 de julho de 2014, tratando um ano antes da implantação do processo e um ano desde o início do processo, sendo que são considerados sete clientes que utilizam o processo de homologação.

A partir destes dados levantados segue o Gráfico 1 que mostra o número de SALTs abertas; considera-se “com versão” as referentes a primeira semana ao ser atualizada a versão e as “sem versão” aquelas abertas nos demais períodos. Percebe-se que o número de atendimentos quase dobrou, este aumento ocorreu principalmente pela entrada de um cliente de grande porte que realiza a abertura de sessenta por cento de todos os atendimentos.

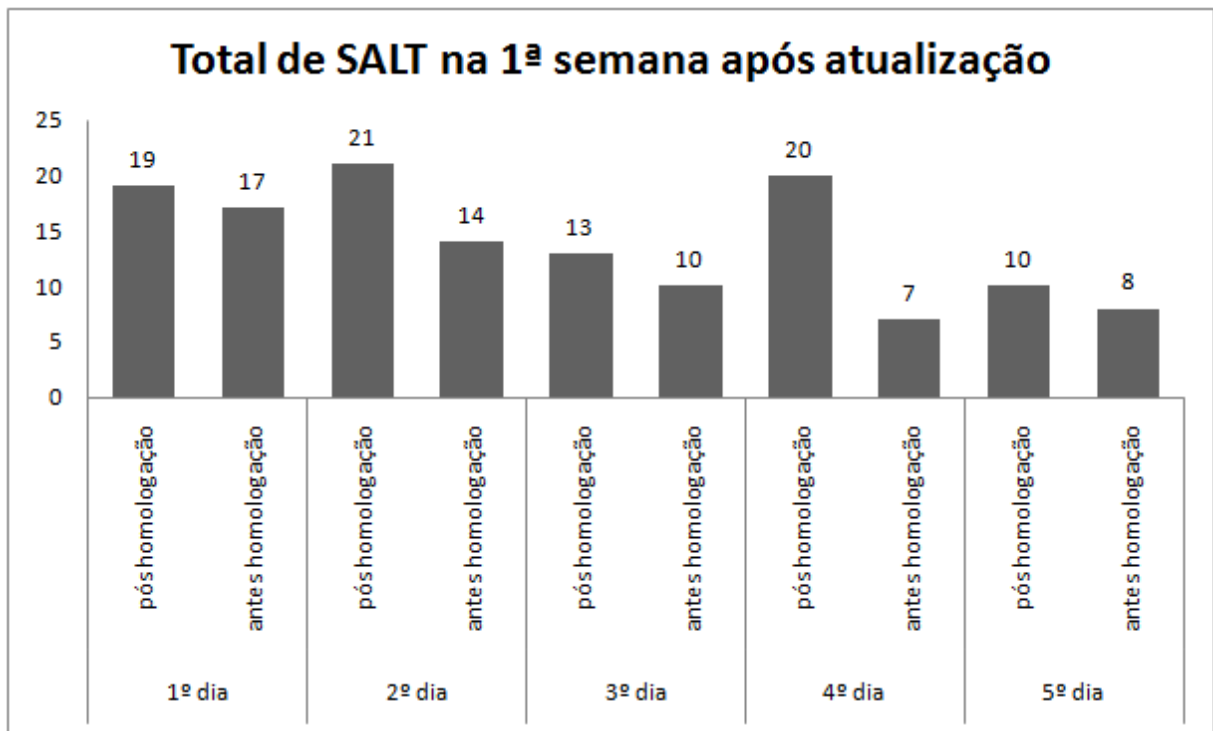
Gráfico 1. Total de SALT abertas



Fonte: o autor, 2015.

No Gráfico 2 estão os atendimentos em que foram necessários envio de uma versão de correção pela equipe de manutenção corretiva. Percebe-se que o número de SALTs é maior após a implantação do processo, sendo este decorrente ao número de atendimentos terem aumentado no último período conforme gráfico 1.

Gráfico 2. Total de SALT na 1ª semana de atualização



Fonte: o autor, 2015.

Percebe-se ainda pelo gráfico 2 que aumentou o número de problemas identificados pelo cliente e corrigidos pela empresa no quarto dia após a atualização, sendo assim problemas que são de baixo impacto na produção, ou ocorrem em situações muito específicas que não são comumente feitas por todos os usuários.

Trabalhando diretamente com o suporte ao cliente e manutenção corretiva podemos perceber que os problemas mais graves que impactavam diretamente a operação que ocorriam no primeiro dia após atualização reduziram. Com isto, diminuiu a necessidade de envios de versões e sua instalação durante o expediente do cliente, podendo estas versões de correções serem instaladas no ambiente de produção após o expediente do primeiro ou segundo dia.

4 CONCLUSÕES E TRABALHOS FUTUROS

Nos últimos anos percebe-se um crescente aumento de empresas de desenvolvimento de software, fazendo assim que ocorra aumento da concorrência no mercado, e o surgimento de novas tecnologias e produtos atendendo a diferentes setores do mercado. Para tanto o cliente passou a exigir um produto com mais qualidade e empresas conceituadas no mercado. Em contrapartida as empresas estão se aperfeiçoando, certificando-se na ISO 9001 e níveis de maturidade, como o CMMI e MPSBr.

A engenharia de software cria metodologias de desenvolvimento, definindo em cada uma das fases do desenvolvimento as atividades que devem ser feitas. Estas atividades são estudadas através de boas práticas que estão em uso pelo mercado. As empresas por sua vez ao escolher uma metodologia de desenvolvimento cria e adapta as suas necessidades e documenta seus processos e fluxos de trabalho.

Para garantir uma qualidade de produto final, os processos são utilizados para definir padrões no desenvolvimento de atividades, bem como a produção de toda a documentação padronizada e armazenada. Um processo deve ser constantemente monitorado a fim de identificar não conformidades e identificar pontos de melhorias no processo.

Quando consideramos o desenvolvimento de software trata-se de um produto final que foi solicitado por um cliente para solucionar um problema. Durante todo o desenvolvimento o cliente deve validar o produto que está sendo produzido.

O cliente como o principal stakeholders do projeto, no início do processo de desenvolvimento, isto é, na fase de engenharia de requisito, ao ser concluída a documentação com os requisitos deve realizar a aprovação, após isso é que estes requisitos serão projetados e desenvolvidos.

Ao concluir o desenvolvimento em que é entregue uma versão para o cliente, inicia-se a homologação do produto entregue, nesta se realiza uma bateria de testes no programa gerado. Os testes devem seguir um plano de testes para validar as funcionalidades descritas na documentação de requisitos aprovada. Esta fase é fundamental para diagnosticar erros não identificados pelas equipes de testes da empresa desenvolvedora e a correção destes erros antes que o produto seja atualizado em seu ambiente de produção e prejudique a operação de funcionamento.

Neste trabalho foi apresentado inicialmente a revisão bibliográfica com a explicação sobre o processo de desenvolvimento e suas etapas, destacando a etapa de validação na qual estão as atividades de homologação realizada pelo cliente. As atividades de

verificação e validação são importantes para qualidade do produto instalado. Realizado o estudo de caso em uma empresa que não possuía um processo de homologação bem definido e passou a utilizar, levantando e analisando os dados para verificação do resultado da implantação do processo de homologação dentro da empresa.

Conforme o estudo de caso pode se perceber que um processo de homologação definido e implantando, aumenta a qualidade do produto ao ser atualizado na produção e reduzindo as falhas que tenham grande impacto nas operações de funcionamento. Criando um plano de testes no qual as funcionalidades críticas são critérios de aceite de versão, lembrando ainda que as integrações realizadas pelo software devem ser testadas e serão criteriosamente avaliados.

Verificou-se que o trabalho de uma equipe que tenha como principal atividade acompanhar o cronograma da homologação e monitorar as atividades do processo de homologação melhora os resultados obtidos com o processo.

Como todo o processo deve-se auditar e identificar pontos de melhoria e não conformidades que devem ser registradas. As melhorias serão discutidas pela equipe, bem como sugestões vindas do próprio cliente serão avaliadas e se adequadas ao processo em funcionamento.

Como trabalho futuros sugere-se realizar o estudo em diferentes empresas que utilizam o processo de homologação, para levantamento de dados e resultados obtidos com a implantação do processo. Realizar entrevistas com clientes das empresas pesquisadas para avaliar o processo e sugerir melhorias caso necessário.

REFERÊNCIAS

BASTOS, Anderson. **Base de conhecimento em teste de software**. 3 ed. São Paulo: Martins Fontes, selo Martins. 2012

_____. **NBR ISO/IEC 12207 - tecnologia de informação: processos de ciclo de vida de software**. Rio de Janeiro: ABNT, 1998. Disponível em: <<https://www.passeidireto.com/arquivo/1091264/nbr-iso-12207---tecnologia-de-informacao--processos-de-ciclo-de-vida-de-sofwar>> . Acesso em: 27 set. 2014

PRESMANN, Roger S. **Engenharia de Software**. Tradução de Rosângela Delloso Penteadó. 6 ed. São Paulo: McGraw-Hill, 2006.

QUALIDADE. In: DICIONÁRIO Michaelis. Disponível em: < <http://michaelis.uol.com.br/>> . Acesso em: 27 set. 2014

SILVA, Edna Lúcia da, MENEZES, EsteraMuszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. –4 ed. rev. atual. Florianópolis:UFSC, 2005. Disponível em: <https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf>. Acesso em: 30 jan. 2015.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução de André Maurício de Andrade Ribeiro. 6 ed. São Paulo: Addison Wesley, 2003.

TATSCH, Evaldo, Jr. **A ITIL e o ambiente de aceite ou homologação**. Publicado 14 de julho de 2010. Disponível em: <<http://www.profissionaisiti.com.br/2010/07/a-til-e-o-ambiente-de-aceite-ou-homologacao/>> . Acesso em 2014.