

Compromissos e processos na Qualidade de Software

Daniel Sant'Anna Andrade ¹

¹Universidade Federal Rural do Rio de Janeiro
Seropédica – RJ – Brazil

danielsantannaandrade@gmail.com

RESUMO

A qualidade de software é um ramo da engenharia de software que propõe a melhoria de todo o projeto, criando padrões e normas para serem validados antes de um produto ser considerado apto a ser entregue, facilitando a tomada de decisão por parte do gestor do projeto. Esses documentos além de auxiliarem, eles também gerando confiabilidade na produção do produto, visto que existe a utilização de certificados e selos que identificam que a empresa possui uma produção que se adequa aos padrões mais qualificados da área de tecnologia.

palavras-chaves: qualidade de software, padrões de qualidade, engenharia de software

1. INTRODUÇÃO

“A ideia de qualidade é aparentemente intuitiva; contudo, quando examinado mais longamente, o conceito se revela complexo. Definir qualidade para estabelecer objetivos é, assim, uma tarefa menos trivial do que aparenta a princípio.”(KOSCIANSKI, 2007, p. 17).

A necessidade da criação de um ramo da engenharia de software apenas para a qualidade dos produtos criados, se deu pela insatisfação de alguns serviços entregues, onde os softwares não atendiam ou então atendiam, mas de forma insatisfatória as necessidades dos clientes.

Com isso, viu-se a necessidade de criar técnicas formais para se configurar a existência de um padrão de qualidade.

Sommerville (2011, p. 454) aponta que para um desenvolvimento bem sucedido, existem 3 preocupações:

- A própria equipe de gerenciamento de qualidade deve criar os processos de validação a ser requerido pelo seu produto, criando também um documento que conste o produto realizado e as suas requisições.
- É necessária a aplicação de um procedimento para identificar se a confecção do software está seguindo os padrões estabelecidos.
- Implantação de um plano de qualidade para definir os processos e padrões a serem seguidos.

Um produto só tem um propósito, sanar as necessidades de um cliente. Esse único propósito se trata de um assunto muito subjetivo, pois cada cliente pode ter uma necessidade diferente ou se atentar a detalhes diferentes, podendo descartar ou não um produto com base nas suas necessidades. No geral, quanto mais difícil de medir um desses fatores de decisão, mais importante ele costuma ser para o cliente.

Geralmente, tem-se a ideia de que um software de qualidade é um software sem bugs ou erros de operação. Isso ocorre por conta da ideia de que é impossível um programa com problemas ser um programa bom. Muito pelo contrário, a qualidade de um software vem do que se foi decidido pelo grupo de qualidade, considerando técnicas e fatores que afetam a utilização do usuário.

2. GARANTIA DE QUALIDADE

Como dito por Vasconcelos (2006 , p.84) a Garantia de qualidade é um conjunto de atividades planejadas e sistemáticas, implementadas com base no sistema da qualidade da organização, promovendo a confiança no projeto, satisfazendo padrões relevantes de qualidade impostos no PMBOK.

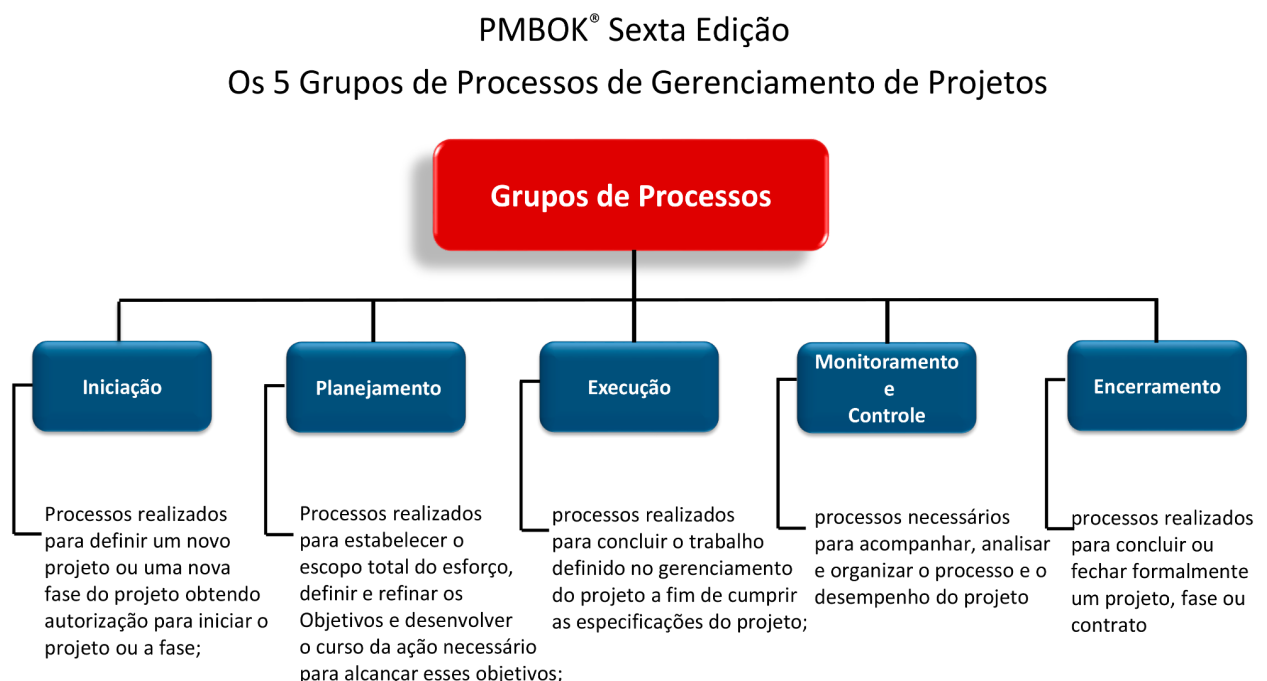


Figura baseada no PMBOK® - Sexta Edição

Fig 1: PROCESSOS DE GERENCIAMENTO DE PROJETOS SEGUINTE PMBOK - SEXTA EDICAÇÃO.

Uma equipe responsável pela garantia de qualidade tem como objetivo testar o produto criado e verificar se os requisitos impostos estão sendo validados gerando registros dos processos de testes. Essa equipe necessita ser uma equipe diferente da equipe de desenvolvimento para que exista uma visão diferente sobre o software e esse processo de garantia de qualidade seja mais eficaz.

A equipe de garantia de qualidade pode ser também a equipe de testes, porém, em casos de equipes separadas, é necessário analisar os testes que foram realizados para verificar sua validação.

É necessário que todos os envolvidos na construção do software tenham em mente a necessidade de produzir um produto de alta qualidade, incentivando todas as partes da produção a alcançar esse objetivo, gerando padrões e métricas necessários em todas as partes da produção.

Existem diferentes abordagens no gerenciamento de projetos, principalmente se tratando de projetos longos e de curto prazo. Onde projetos maiores e com várias equipes necessitará de uma documentação mais elaborada de forma que seja mais simples visualizar as abordagens de cada grupo. Já projetos menores não é necessária tanta documentação por existir uma maior informalidade na comunicação dos desenvolvedores, contudo ainda sim, partir para uma abordagem onde exista esse relatório de qualidade é essencial para que a produção não perca seu potencial.

Sommerville (2011, p. 454) comenta que devido a dificuldade em avaliar a qualidade de um software, a equipe de gerenciamento de qualidade precisa responder a algumas características do sistema para ter uma melhor avaliação. As perguntas são:

1. Durante o desenvolvimento os padrões de programação e documentação foram seguidos?
2. O software foi devidamente testado?
3. O software é suficientemente confiável para ser colocado em uso?
4. O desempenho do software é aceitável para uso normal?
5. O software é útil?
6. O software é bem estruturado e compreensível?

Outro fator muito importante é o desempenho do produto. Mesmo que o produto tenha funcionalidades inesperadas na qual o usuário consegue contornar para atingir o seu objetivo, isso não será um empecilho tão grande para o usuário a ponto de afetar tanto o desempenho. Porém, produtos que demoram muito para dar resposta da tarefa requisitada, serão rejeitados por grande parte dos usuários.

Também é fato que dificilmente um software será otimizado em todos os aspectos, sendo prioridade da equipe de qualidade definir um plano sobre quais aspectos serão os mais importantes deixando outros menos importantes de lado. Alguns desses aspectos devem ser tratados como extremamente importantes, necessitando estar extremamente otimizado no sistema.

3. PROCESSO DE FABRICAÇÃO

O processo de fabricação de um software é diferente do processo de fabricação de um produto na indústria. Na indústria, quando um sistema é configurado conforme as necessidades de um produto, ele sempre irá produzir todos os produtos no mesmo padrão. Já no desenvolvimento de software, isso é um processo mais complexo.

O desenvolvimento de software se trata de um processo criativo, sendo muito complicado manter um mesmo padrão de uma produção para a outra, pois não se trata simplesmente de repetir padrões e processos, tendo muitas variáveis externas que corroboram na produção.

Por conta disso, até mesmo a problemas individuais dos desenvolvedores afetam completamente o desenvolvimento. Pressões de prazos e cronogramas também auxiliam na qualidade final do produto.

Isso demonstra que os acontecimentos durante o processo de fabricação de software também têm grande importância na qualidade final do produto. As métricas e estratégias definidas não são totalmente uma indicação de que a qualidade será sempre alcançada. Porém, o gerenciamento e a equipe de qualidade pode contornar os problemas que forem gerados no desenvolvimento com constantes testes, melhorar a eficiência final do produto.

Também é importante apontar, que muitas características do software só serão descobertas com o uso a longo prazo, necessitando de atualizações para consertar problemas futuros.

4. PADRÕES

Segundo Kosciński(2007, p.19), foi durante o período da Segunda Guerra Mundial que os primeiros padrões de qualidade foram criados, como: ASQC (American Society for Quality Control), ABNT (Associação Brasileira de Normas Técnicas), a ISO (International Standardization Organization), entre outras. Esse fato se deu pela grande quantidade de produção de material de qualidade, principalmente referente a indústria bélica, necessitando de uma melhoria nas produções para acompanhar a evolução e necessidade nesse período. O Japão foi um grande inovador nesse cenário durante esse período, principalmente pelos diagramas de causa e efeito (modelo Ishikawa/diagrama espinha de peixe).

Os padrões de desenvolvimento para serem aplicados, necessitam de instruções e metodologias que se apliquem ao resultado desejado. As escolhas das ferramentas são extremamente importantes, pois os padrões de garantia de qualidade serão criados em cima deles.

Claramente, os padrões têm como principal objetivo enriquecer a qualidade do produto, necessitando de uma aplicação eficiente para não gerar perda de tempo e de trabalho.

Sommerville (2011, p. 454), disserta que existem dois tipos de padrões relacionados a qualidade de software: padrões de produto e padrões de processo.

- Padrões de produto são aplicados ao desenvolvimento do produto, na qual se utilizam padrões de documentos, padrões de documentação e padrões de codificação.
- Padrões de processo são aplicados às boas práticas, as definições de especificações, ferramentas, processos de validação, e os documentos na qual se basearam para a geração desse projeto.

Existe uma infinidade de padrões muitas vezes específicos para uma determinada área de atuação. É importante que os desenvolvedores do produto sejam consultados na hora da determinação dos padrões a serem utilizados para que não sejam descartados no meio do projeto por não se adequarem ao desenvolvimento. E como um padrão pode não acabar se adaptando a todo software feito por uma empresa, existe também a possibilidade de modificar um padrão existente para se adequar ao projeto atual, realizando pequenas alterações e as documentando no manual de padrões da empresa.

4.1. ISO

Trata-se de um sistema internacional criado pela International Organization for Standardization, que cria normas a serem utilizadas em todos os lugares do mundo, de forma que padronize os processos de produção/trabalho. Sendo no Brasil representada pela ABNT (Associação Brasileira de Normas Técnicas). Ela é um sistema de gestão da qualidade que comprova a qualidade de uma empresa através da sua certificação, dando vantagem competitiva às empresas que possuem seu selo ao passar credibilidade e confiabilidade.

“A ISO possui uma família de normas que trata de qualidade: a ISO série 9000. Nesta família está a ISO 9001, que trata dos requisitos para sistemas de gestão da qualidade. Uma organização que possui um sistema de gestão da qualidade de acordo com a norma ISO 9001 pode solicitar a certificação e obter o “selo de conformidade ISO 9001”. A versão brasileira da ISO 9001 chama-se ABNT NBR ISO 9001.”(TEMPLUM, [s.d.]

4.2. ISO 9001

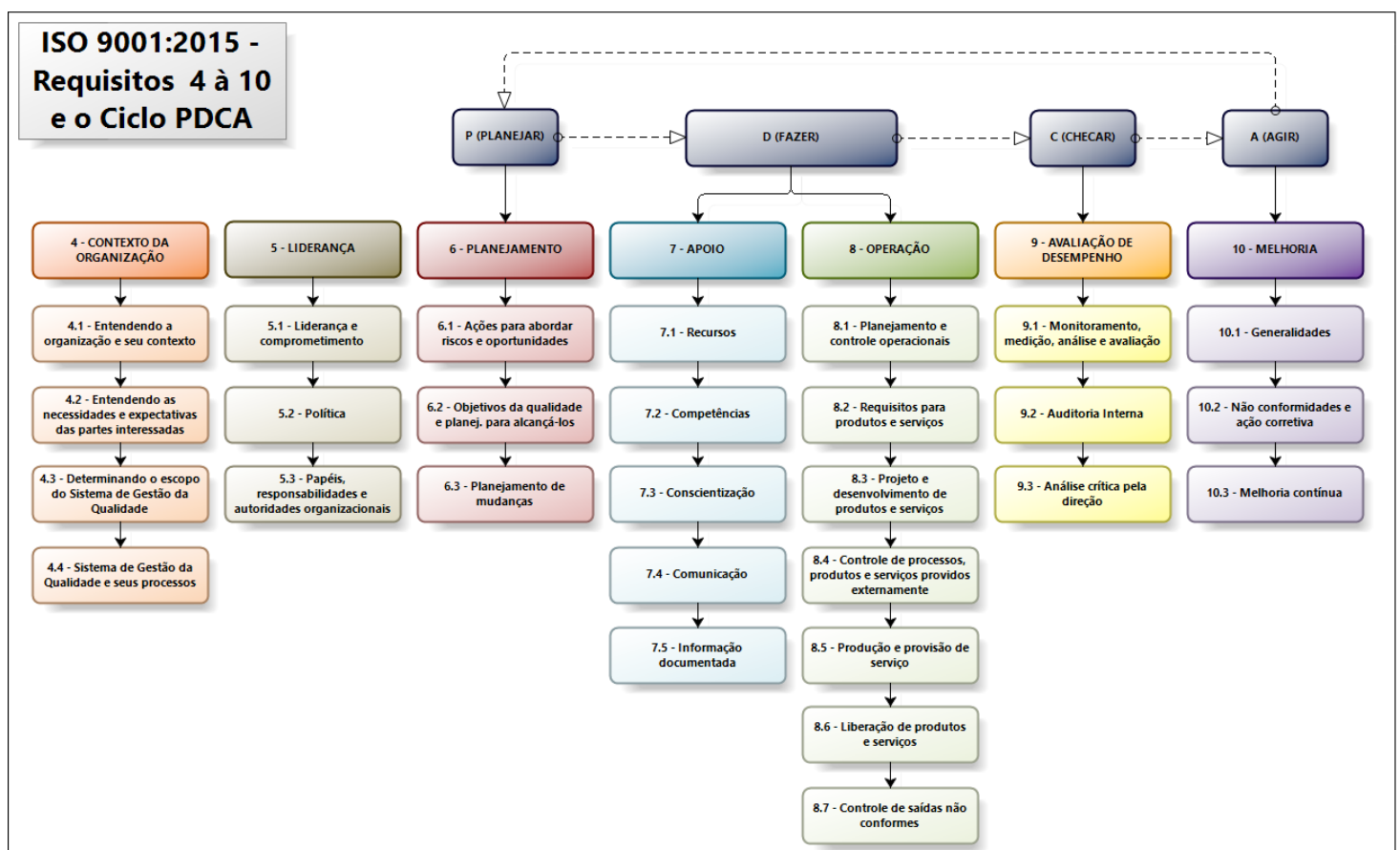


Fig 2: REQUISITOS DA ISO 9001:2015.

Dentro das ferramentas de padrões, há a ISO 9001 que é um framework de padrões. Essa ISO propõe os padrões de qualidade, como também seus processos de qualidade e

métodos a serem utilizados. Ela propõe que seja mantido registro de todos os processos organizacionais que foram utilizados ao longo do tempo em um manual de qualidade da empresa, criando uma espécie de histórico das metodologias mais importantes. Não é necessário que se formalize todos os processos de qualidade utilizados com manuais os detalhando, é possível que esse processo seja feito de forma mais simples com apenas incremento de informação sucinta.

“A ISO 9001 é um sistema de gestão com o intuito de garantir a otimização de processos, maior agilidade no desenvolvimento de produtos e produção mais ágil a fim de satisfazer os clientes e alcançar o sucesso sustentado.”(TEMPLUM, [s.d.]

É importante também destacar que as ISOs não irão garantir que todo produto de uma empresa que possua sua certificação terá máxima qualidade no seu software. A ISO 9001 garante que a empresa possui seus procedimentos e normas e que os utiliza, ou pelo menos se propõe a utilizá-los. Passando uma confiança, mas não uma certeza de que todas as práticas e propostas da norma estão sendo seguidas para gerar um produto com sua máxima qualidade.

E por fim, projetos pequenos não necessitam de normas robustas como a ISO 9001 para serem desenvolvidos, se preocupando mais com a utilização de métodos ágeis a fim de tirar um pouco da burocracia e mesmo assim possuir um padrão que gera excelência e qualidade no produto.

5. REVISÃO

“Cada vez mais, as verificações e validações de software têm sido consideradas ferramentas úteis no contexto da garantia da qualidade de software. Através delas, são obtidas visões mais concretas com relação a aspectos de qualidade de alguns produtos de software”(VASCONCELOS, 2006, p. 109)

Revisões e inspeções têm a principal finalidade de verificar as partes do projeto que já foram prontas, averiguando se o projeto está de acordo com a documentação, identificando os problemas que forem encontrados ou que estejam diferentes do que foi proposto na proposta de qualidade de software. Os gerentes de projetos recebem os relatórios de revisões e inspeções e utilizam para decidir o que deverá ser realizado posteriormente. Esses relatórios são construídos com base nos documentos gerados ao longo do desenvolvimento, testes e do processo de revisão e inspeção da equipe de qualidade e desenvolvimento.

A revisão por si só não serve somente para verificar se os padrões pré-estabelecidos em etapas mais iniciais do projeto estão sendo utilizados corretamente, mas sim para procurar por erros na produção e recursos que acabaram ficando de fora. Assim, elas conseguem ampliar a qualidade de software levando a correção prévias de problemas que só apareceriam muito mais a frente no desenvolvimento.

Como é uma etapa de detecção de erros, é de extrema importância que a atenção dos gerentes esteja voltada para qual erro foi encontrado e como corrigi-lo do que para quem produziu esse erro, pois como o desenvolvimento é um processo criativo, apontar dedos e culpar os desenvolvedores pelos problemas pode afetar negativamente a produção como um todo.

Os problemas encontrados durante as revisões e inspeções devem ser registrados como todos os passos do gerenciamento de qualidade do produto, gerando sua prévia correção e

auxiliando a utilizações futuras e retomadas de tópicos a fim de conferir se tudo o que foi proposto foi atendido da melhor forma possível.

Mesmo com a efetividade desse processo de inspeção e revisão, ele não é sempre utilizado pela equipe de qualidade em todas as empresas ou em todos os projetos. Muitas vezes isso se dá ao fato de ter que disponibilizar toda uma equipe apenas para realizar esse processo, gerando mais custo e mais tempo para a finalização do projeto, ou então por se tratar de utilizar uma metodologia ágil, onde os próprios desenvolvedores irão verificar e refatorar o código vez ou outra para não deixar acumular problemas.

5.1. PROCESSO DE REVISÃO

Antes de se iniciar a revisão é realizado um planejamento de revisão, onde é decidido o que será revisado, verificando o projeto como um todo e como ele influencia o software que está sendo construído. O grupo que irá fazer parte desse processo de revisão precisa estar alinhado com o que os padrões e modelos já definidos propõem para as partes que serão revisadas.

Durante a revisão, toda a equipe deverá fazer uma breve verificação do que foi proposto e do que foi realizado, de forma que passe por todos os pontos pré-estabelecidos. Apesar de ter um líder no grupo de revisão é bom que as decisões sejam tomadas como um todo, detalhando sempre as decisões realizadas durante a revisão.

Após a revisão a equipe de desenvolvimento precisa ler o documento criado durante a revisão para realizar a alteração dos erros encontrados no software, e se possível, realizando também a refatoração do código para melhor estruturá-lo. Ao final da verificação do documento, o líder da equipe de revisão precisa verificar se os problemas encontrados foram sanados.

Caso necessário, esse processo deverá ser realizado novamente para consertar novos problemas ou então verificar se tudo foi realizado conforme está documentado no registro de revisão.

5.2. PROCESSO DE INSPEÇÃO

As inspeções são parecidas com os testes por também verificar o que foi produzido até o momento da inspeção. A grande diferença para os testes é que durante as inspeções, não será verificado totalmente uma funcionalidade, apenas uma parte da mesma fragmentando a verificação completa e reduzindo a necessidade de uma.

Essas inspeções consistem na verificação de linha a linha do código, verificando a lógica utilizada, não utilização de boas práticas, erros e bugs, e falta de concordância entre os diversos pontos do projeto. Utilizando uma check-list de pontos importantes para serem avaliados naquele produto.

5.3. DEFEITOS E FALHAS

Defeito é referente a algo que não foi executado no código de forma corretamente, pode estar associado a um erro do programador, fato muito comum, pois muitas vezes as

possibilidades de execução são infinitas e não se pensa em todas elas de uma vez só no momento em que se escreve o código. Defeitos não são empecilhos que impedem o funcionamento, e sim que geram resultados inesperados em uma determinada execução.

Falha é um problema provocado por um defeito que tem a capacidade de paralisar o funcionamento do software podendo muitas vezes levar a perda de processos.

É possível também que existam falhas ou defeitos que não dão sinais visíveis de sua existência, podendo comprometer o software no futuro. E também, nem sempre falhas e defeitos são provocados por problemas do código, mas sim por problemas de conexão de bancos de dados ou links não mais funcionais.

6. MEDIÇÕES E MÉTRICAS

Segundo Vasconcelos(2006, p.105), as métricas são ferramentas essenciais na gerência de software. Os engenheiros medem as características de seus produtos para obter uma visão geral de vários aspectos, como: atendimento aos requisitos, processo de desenvolvimento, esforço e custo, e outros. Com essas medições, o cliente pode determinar se esse produto irá atender às suas necessidades em qualidade satisfatória.

As medições e métricas se baseiam em utilizar padrões e documentos para avaliar a introdução de novas estratégias ou tecnologias a serem introduzidas no projeto ou na empresa. Ela irá funcionar ao realizar uma avaliação geral antes da introdução da novidade e após um tempo de uso se realiza uma nova avaliação para comparar com a avaliação antiga.

Com as medições e métricas é capaz de verificar eficientemente os impactos que a estratégia ou tecnologia gerou durante o período em que esteve sendo avaliada e os gestores tomam decisões para o futuro do projeto. Assim, facilita caso o gestor perceba que projeto não receba um impacto positivo da nova utilização de tecnologia e necessite tomar uma decisão se permanece com ela ou a descarta.

As métricas de software se dividem em duas: métricas de controle ou métricas de previsão. As métricas de controle são relacionadas a correção do software e tempo gasto na análise dos erros gerados. Já as métricas de previsão se baseiam em indicadores do projeto, ligação das principais áreas e como elas irão impactar no projeto como um todo.

As métricas de software se dividem em duas: métricas de controle ou métricas de previsão. As métricas de controle são relacionadas a correção do software e tempo gasto na análise dos erros gerados. Já as métricas de previsão se baseiam em indicadores do projeto, ligação das principais áreas e como elas irão impactar no projeto como um todo. Essas duas métricas serão utilizadas posteriormente para o gestor tomar decisões referente ao andamento do projeto, com as métricas de controle indicando possíveis mudanças e as métricas de previsão indicando o que irá precisar e quanto tempo levará para implementar as mudanças/correções.

O futuro dessa área propõe a existência de ferramentas que sejam capazes de avaliar a qualidade de forma automatizada, apontando problemas, erros e modificações possíveis para aumento da qualidade, porém no momento ainda é tudo feito de forma muito manual e não existe nenhuma ferramenta automática que tenha uma taxa de acerto satisfatória.

6.1. COLETA DE DADOS

A coleta de dados permite entender melhor a ideia geral de como os usuários estão lidando com o produto, através da análise desses dados. Essa é uma parte onde a opinião do usuário é extremamente importante, pois dessa forma é capaz de descobrir algum problema, caso as solicitações de usuários sejam mais para alterar partes do produto ou então descobrir o que falta no produto, caso as solicitações dos usuários sejam mais para acrescentar funções nas ferramentas já prontas ou então novas ferramentas que muitas vezes nem a equipe de produção tenha pensado.

A coleta de dados permite entender melhor a ideia geral de como os usuários estão lidando com o produto, através da análise desses dados. Essa é uma parte onde a opinião do usuário é extremamente importante, pois dessa forma é capaz de descobrir algum problema, caso as solicitações de usuários sejam mais para alterar partes do produto ou então descobrir o que falta no produto, caso as solicitações dos usuários sejam mais para acrescentar funções nas ferramentas já prontas ou então novas ferramentas que muitas vezes nem a equipe de produção tenha pensado.

Essa interação com o cliente auxilia muito na longevidade do sistema, alterando as funções que não são muito utilizadas ou que não funcionam. Essa relação com o cliente gera uma confiança e faz com que o mesmo não procure outras alternativas no mercado para resolver as suas necessidades, além de criar um público fiel ao produto.

Esses dados de modificação do software muitas vezes necessitam de informações sobre a usabilidade do cliente para verificar a total necessidade ou não da mudança.

7. REFERÊNCIAS

DE VASCONCELOS, Alexandre Marcos Lins et al. Introdução à Engenharia de Software e à Qualidade de Software. Minas Gerais: Universidade Federal de Lavras, 1. ed. 2006.

FURNIEL, Igor. ISO 9001 – Sistema de Gestão da Qualidade. **Templum**. [s.d.] Disponível em: <https://certificacaoiso.com.br/iso-9001/>. Acesso em 25/11/2021.

KOSCIANSKI, André; DOS SANTOS SOARES, Michel. Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. Novatec Editora. 2. ed. 2007.

LANDO, Andressa. O que é ISO e por que certificar? **Templum**. [s.d.] Disponível em: <https://certificacaoiso.com.br/o-que-e-iso-e-por-que-certificar/>. Acesso em 25/11/2021.

SOMMERVILLE, I. Engenharia de software. São Paulo: Pearson Prentice Hall. 9. ed. 2011.