

Documento de Melhores Práticas para Desenvolvimento de Software

**Ariel Cristina Ferreira da Costa
Caroline Gonçalves**

**Faculdades Integradas Camões
Curso de Análise e Desenvolvimento de Sistemas**

Prof. Rafael Gorski

Resumo. *Este documento descreve as melhores práticas para um desenvolvimento de software desde o levantamento da documentação que define como o software irá atender o cliente, a especificação em que é o conjunto dessa documentação que define o requisitos qual deve ser comunicado para o time de desenvolvimento.*

Abstract. *This document describes the best practices for a software development since the documentation that defines how the software will serve the client, the specification in which is the set of this documentation that defines the requirements which must be communicated to the development team.*

1. Entender a Necessidade do Cliente

1.1. Antes de qualquer tomada de decisão para o início do desenvolvimento de um software, é necessário entender qual a real necessidade do cliente. Para tanto é preciso de alguém que consiga compreender a situação pelo qual o cliente está passando e repassar essas informações com clareza para o documento para que não haja futuras alterações ou desentendimentos pela equipe de desenvolvimento, pois estas alterações podem gerar custos excessivos qual não era esperando pelo cliente.

Algumas abordagens para extrair o máximo de informações, é iniciar com os stakeholders do processo e interagir para começar a descoberta dos requisitos. Para isso é necessário conhecer algumas das técnicas de levantamento de requisitos e escolher a melhor abordagem para iniciar. Lembrando que pode-se utilizar duas técnicas ou mais para melhor análise dos problemas. Abaixo segue algumas das técnicas:

1.2. Levantamento orientado a pontos de vista

A abordagem levantadas como ponto de vista, reconhece os diferentes pontos de vista a serem considerados de um problema, pois algumas podem mostrar alguma duplicidade, mas de algum modo apresenta requisitos comuns. Nesta etapa os analistas se reúnem com os stakeholders para fazerem o brainstorming que é uma técnica utilizada para geração de ideias.

1.3. Etnografia

Nesta técnica, é utilizada a observação do trabalho diário e fazendo anotações em cima das tarefas realizadas. A Etnografia é eficaz na descoberta de dois tipos específicos de requisitos que seriam os requisitos derivados da maneira como elas trabalham e não como elas dizem como deveriam trabalhar e os requisitos da cooperação e conscientização das atividades de outras pessoas.

1.4. Workshops

O trabalho do workshop é acionar o trabalho em equipe, onde teremos os stakeholders representando a organização e um facilitador para conduzir o workshop. O facilitador deve ser alguém com postura de mediador e observador e é necessário ter locais e horários previamente agendados além do assunto definido e a documentação do seminário.

1.5. Entrevistas

Uma das técnicas mais comuns, a entrevista é importante para coletar bons dados e resultados na fase inicial, pois o entrevistado expõe seus problemas e ideias, porém é necessário planejar a entrevista para que não haja dispersão no assunto alongando a entrevista.

1.6. Questionários

Utilizamos o questionário quando não é possível ter acesso aos usuários principais por estarem em diversos locais diferentes. Neste caso o questionário é elaborado em cima das informações necessárias para dar início ao levantamento da documentação. O questionário deve ser claro e conciso e deve-se ter o controle para saber se todas as pessoas receberam e responderam devidamente o questionário.

2. Documentação

Após entender a necessidade do cliente, é importante documentar todas as expectativas em forma de requisitos. Todas as etapas devem ser definidas para atender a necessidade do cliente. O conjunto deste documento forma a especificação que serve também como um contrato e a comunicação para quem irá desenvolver o software. Antes da documentação ser entregue ao time de desenvolvimento, é importante que um arquiteto de software analise os requisitos para definir o documento em desenho que normalmente é feito em forma de notação UML. A documentação além de servir como contrato, também é o meio para verificar todo o passo a passo realizado para a criação do software e pode ser consultado por novos integrantes da equipe que não conheçam ou não participaram da evolução da criação do mesmo.

3. Protótipo

É recomendável desenvolver protótipos no início do projeto para apresentar ao cliente para que seja possível dar a ele uma visão do que o software irá se tornar com base no levantamento de requisitos definido no início do projeto junto com o cliente. O protótipo deve ser criado de forma que seja possível realizar alterações caso o cliente deseje algumas mudanças.

4. Equipe, Ferramentas e Testes.

Para o sucesso do desenvolvimento do software é necessário contar com uma equipe engajada, com papéis bem definidos para que haja um bom relacionamento e principalmente comunicação devido a todas as adversidades que possam aparecer durante o desenvolvimento para que todas sejam resolvidas da melhor forma possível. É necessário também a definição das ferramentas que serão utilizadas, sempre avaliando o custo de acordo com o orçamento do projeto, como por exemplo, a

linguagem de programação, plataforma de desenvolvimento entre outros pensando além do custo a economia do tempo para entregar em tempo hábil acordado com o cliente.

Além da equipe e da correta escolha das ferramentas, é importante lembrar-se de sempre realizar testes verificando todos os tipos de usuários, funcionalidades e prevenção de possíveis erros. Para melhor obtenção de dados, seria interessante testar o softwares com pessoas fora da equipe para que possam visualizar problemas que não foram percebidos pelos desenvolvedores.

5. Metodologias Ágeis

A documentação que será entregue ao desenvolvedor qual irá implementar a solução utilizando como referência a especificação de requisitos tem como função definir e comunicar para futuras referências. Em caso de documentações muito detalhadas, será difícil lidar com alterações, como por exemplo alteração de requisitos, pois este exige custo alto e riscos dependendo da forma da alteração que pode alterar também toda a documentação.

Tendo em vista este tipo de problema, as Metodologias Ágeis procuram incentivar formas naturais de comunicação para reduzir ao máximo a necessidade de documentação. No entanto há algumas limitações, como esta metodologia visa a redução da necessidade da documentação, para consultas, referências entre outros, é necessário que a aplicação seja realizada somente em pequenas equipes, para pequenos incrementos e até mesmo em um espaço físico pequeno para melhor comunicação.

Portanto a redução da documentação, não significa eliminá-la e sim analisar e avaliar o custo para todo o projeto e em caso seja menor, a intenção é manter o projeto documentado.

6. Referências

Técnicas para Levantamento de Requisitos: <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>

Melhores Práticas para Desenvolvimento de Sistemas e Software:
<https://www.ibm.com/developerworks/br/rational/library/systems-software-lifecycle-development/index.html>