

ENGENHARIA DE SOFTWARE:

metodologia extreme programming (xp)

Rodrigo Ferreira dos Anjos

RESUMO: O trabalho faz um descritivo sobre a metodologia XP (*Extreme Programming*) e um comparativo com metodologia clássica usada ao longo do processo de desenvolvimento de software. O estudo surgiu da necessidade de se entender como foi criada a metodologia XP, seus conceitos e funcionalidades, para que se possa ter o mínimo de conhecimento sobre o assunto. Este artigo tem como objetivo esclarecer, de maneira resumida, o que é a metodologia XP, seus valores e como ela é aplicada no desenvolvimento de projetos e softwares, bem como fazer-se entender a diferença entre este método e o método tradicional. Pretende-se mostrar o desenrolar de como a metodologia funciona no decorrer do desenvolvimento com base nos valores por ele (método) estabelecidos.

Palavras-chave: Extreme Programming, Metodologia Ágil, Engenharia de Software, Kent Beck, Desenvolvimento de Software Ágil.

1 EXTREME PROGRAMMING

A programação extrema ou programação ágil nasceu da necessidade de se fazer desenvolvimento em tempos menores e com o mínimo de problemas possível. Criada por Kent Beck na década de 90 tornou-se referência mundial a partir de 2001 quando, juntamente com outros métodos como, por exemplo, o SCRUM, foram estabelecidas características comuns entre eles, criando assim a “Aliança Ágil”.

Para que seja possível essa agilidade no processo, algumas medidas e valores foram criadas. Estas práticas tornam-se totalmente eficazes quando levadas ao extremo. São elas: Comunicação, Simplicidade, Feedback, Coragem e Respeito.

1.1 COMUNICAÇÃO

A comunicação entre o cliente e a equipe de desenvolvimento deve ser a mais clara e eficaz possível. Isso permite que todos os minuciosos detalhes sejam

abordados com a devida atenção e agilidade. Este método busca aproximar todos os envolvidos no projeto afim de todos se comuniquem por igual.

1.2 SIMPLICIDADE

Este valor diz respeito a sempre buscar fazer uma implementação do modo mais simples possível para atender as necessidades atuais do cliente, ou seja, codificar alguma funcionalidade somente para o problema existente hoje. Com isso o cliente pode aprender durante o projeto e gerar um melhor feedback para a equipe.

1.3 FEEDBACK

Quando o cliente aprende sobre a ferramenta e/ou sobre o sistema que está sendo desenvolvido ele consegue conduzir o crescimento do projeto dando seu parecer para a equipe desenvolvedora., isso garante que as atenções estejam voltadas para o que realmente irá dar valor ao projeto.

1.4 CORAGEM

A XP defende que para que se tenha sucesso na metodologia é preciso ter coragem. Fazer mudanças de paradigmas, dar e receber feedbacks sem medo das consequências, não preservar códigos ruins. Não pode ter medo dos riscos que aparecem no decorrer do projeto, deve-se saber como lidar com eles para resolve-los da forma mais rápida e eficaz possível.

1.5 RESPEITO

Respeito é o que dá sustentação aos outros valores. Somente haverá uma boa comunicação, a real preocupação com o projeto se houver respeito dentro da equipe.

2 PRÁTICAS XP

As boas práticas da XP podem ser definidas em um conjunto de atividades cujas as equipes envolvidas utilizam durante o projeto. Para que todo o trabalho tenha um final glorioso é preciso confiar nessas atividades e ter a ciência de que devem ser aplicadas em conjunto. São elas:

- **PLANEJAMENTO:** Como a XP se apoia em requisitos atuais para o desenvolvimento, é preciso separar o que realmente é necessário do que não é;
- **ENTREGAS FREQUENTES:** Fazer entregas menores tempos, com tamanhos pequenos conforme os requisitos aparecem, mas contendo o que interessa para o negócio;
- **METÁFORA:** Descrever o software sem usar termos técnicos para que seja o mais transparente possível para o cliente;
- **PROJETO SIMPLES:** Deixar o software mais simples possível, mas contendo os requisitos necessários e atuais. Requisitos futuros devem ser adicionados somente quando estes aparecerem;
- **TESTES:** Fazer testes mesmo antes da implementação, isso porque a validação é primordial para a XP;
- **PROGRAMAÇÃO EM PARES:** Desenvolver o software utilizando duplas é sempre mais produtivo, uma vez que ambos têm ideias diferentes e visões diferentes sobre o mesmo aspecto, podendo assim, identificar com mais facilidade possíveis erros que apareçam;
- **REFATORAÇÃO:** Sempre que possível, melhorar o código sem que este perca suas funcionalidades. Isto pode ser feito em todas as etapas do desenvolvimento;
- **PRIORIDADE COLETIVA:** Esta prática garante que todos os membros de uma equipe possam fazer modificações no código, desde que estas alterações agreguem valor ao produto final;
- **INTEGRAÇÃO CONTÍNUA:** Para que os desenvolvedores estejam sincronizados com relação ao projeto (software). Para isso é necessário que o código construído várias vezes ao dia, evitando assim, surpresas indesejadas e problemas maiores futuros;

- 40h/SEMANA: Não é necessário fazer horas extras durante a semana, caso seja necessário e estas passem de duas semanas consecutivas, é bom rever o projeto, pois existe um problema bem grande envolvido;
- CLIENTE PRESENTE: A participação do cliente ao longo do desenvolvimento é fundamental, já que ele deve estar disponível para tirar dúvidas nesse período, evitando que o projeto sofra atrasos ou criações erradas;
- CÓDIGO PADRONIZADO: Uma arquitetura de software padronizada é interessante no sentido de que todos os desenvolvedores podem fazer suas alterações sem que profissionais futuros se atrapalhem na hora de rever este mesmo código.

3 METODOLOGIA XP x METODOLOGIAS TRADICIONAIS (CASCATA)

As metodologias tradicionais são realizadas através de rígidas regras, onde o projeto deve ser guiado em sequência e os testes, cuja etapa é fundamental, são realizados somente no final do desenvolvimento. Esta prática da vantagem ao modelo XP, pois os testes são feitos e refeitos em todas as etapas do processo de desenvolvimento, deixando o projeto mais flexível com relação ao erros e problemas que aparecem.

Na XP a definição dos requisitos é obrigatórios e é atualizada ao longo do desenvolvimento, em contrapartida, no método cascata, estes requisitos são usados para a criação de uma versão base que servirá para todos os passos que vem à frente, mas não deixa de ser obrigatório também.

Quando é preciso criar o projeto do software a ser desenvolvido em cascata, este somente começa assim que a definição dos requisitos estiver finalizada, já na metodologia ágil, este ocorre informalmente com o cliente conforme o desenrolar do projeto.

Quando o assunto é desenvolvimento o método XP pula na frente outra vez, pois não precisa seguir uma sequência de passos pré-determinados pelo projetista do software implementando as funcionalidades uma a uma no projeto. Esta etapa vai sendo desenvolvida de acordo com as necessidades do cliente.

Já na etapa final do projeto, no modo tradicional a equipe desenvolvedora entrega o software completo, podendo o cliente gostar ou não do resultado. Contudo

a XP faz a entrega do produto em partes durante o seu progresso. Com isso, caso o cliente desejar alterar alguma funcionalidade no software, esta correção é feita em tempo real.

REFERÊNCIAS

A IMPORÂNCIA da Engenharia de Software. *Engenharia de Software, Blogspot*, 2008. Disponível em: <http://engenhariadesoftware.blogspot.com/2007/03/programacao-extrema-xp-html>. Acesso em: 17 set. 2017.

BRANDÃO, Daniel. Extreme Programming (xp) – Resumo. *SlideShare*, 2013. Disponível em: <https://pt.slideshare.net/DanielBrando3/extreme-programming-xp-resumo>. Acesso em: 17 set. 2017.

FONSECA, Daniel. Conceitos Básicos sobre Metodologias Ágeis para Desenvolvimento de Software (Metodologias Clássicas x Extreme Programming). Disponível em: <http://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-para-desenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>. Acesso em: 16 set. 2017.

GONÇALVES, Júnior. Metodologia XP (eXtreme Programming) – Comparativos entre as metodologias. *Hiperbytes*. Disponível em: <http://hiperbytes.com.br/xp/metodologia-xp-extreme-programming-comparativos-entre-as-metodologias/>. Acesso em: 17 set. 2017.

MEDEIROS, Higor. Introdução ao Extreme Programming (XP). *Devmedia*. Disponível em: <http://www.devmedia.com.br/introducao-ao-extreme-programming-xp/29249>. Acesso em: 16 set. 2017.

SILVA, Carlos Henrique Martins da. eXtreme Programming (XP). *SlideShare*, 2014. Disponível em: <https://pt.slideshare.net/CarlosHenrique372/e-xtreme-programming-xp-40079788/6>. Acesso em: 16 set. 2017.