

## **Metodologia Ágel: Extreme Programming (XP)**

Lucas Fagundes Teixeira, Wagner Rodrigo Da Silva

Curso de Engenharia de Software – Centro de Ensino Superior (CESJF)

Caixa Postal 36.100.000 – Juiz de Fora – MG – Brasil

lucasft\_01@outlook.com, wagnerrodrigo.pan@gmail.com

### **RESUMO**

Este artigo tem como objetivo o esclarecimento à respeito da metodologia ágil extreme programming (XP). Em particular será mostrada as principais características e as práticas desta, que serão tabeladas para melhor visualização. Além disso, será feita uma breve comparação com uma metodologia tradicional para expor algumas das principais diferenças e vantagens em usar este framework mais atual. Finalmente o ciclo de vida do XP será explicado em um único tópico em conjunto com uma imagem bem didática a respeito do seu funcionamento.

Palavras-chaves: Metodologia ágel, Extreme programming.

### **ABSTRACT**

This article aims to clarify the agile methodology extreme programming (XP). In particular will be shown the main characteristics and practices of this, which will be tabulated for better visualization. In addition, a brief comparison will be made with a traditional methodology to expose some of the key differences and advantages in using this most current framework. Finally the XP life cycle will be explained in a single topic together with a well didactic image about its operation.

Keywords: Agile methodology, Extreme programming.

## **1 INTRODUÇÃO**

As metodologias ágeis surgiram para suprir certos entraves no desenvolvimento que as metodologias tradicionais não conseguiam tratar. Feedback rápido com o cliente e rápida adaptação com os inevitáveis problemas ao longo do projeto são apenas algumas de outras complicações não tratadas pelo sistema arcaico. Os resultados dessas dificuldades podem ser notados em alguns dados recolhidos em 1995 onde, usando como base 8380 projetos, mostram que apenas

16,2% dos projetos foram entregues respeitando os prazos e os custos e com todas as funcionalidades especificadas. Aproximadamente 31% dos projetos foram cancelados antes de estarem completos e 52,7% foram entregues, porém com prazos maiores, custos maiores ou com menos funcionalidades do que especificado no início do projeto.

A idéia das metodologias ágeis é o enfoque nas pessoas e não em processos ou algoritmos. Além disso, existe a preocupação de gastar menos tempo com documentação e mais com a implementação. Uma característica das metodologias ágeis é que elas são adaptativas ao invés de serem preditivas. Com isso, elas se adaptam a novos fatores decorrentes do desenvolvimento do projeto, ao invés de procurar analisar previamente tudo o que pode acontecer no decorrer do desenvolvimento.

Dentre as várias metodologias ágeis existentes, as mais conhecidas são a Extreme Programming e a Scrum. No entanto, focaremos em abordar apenas a respeito do XP neste artigo.

## **2 Metodologias**

### **2.1 Metodologia Tradicional**

As metodologias mais tradicionais devem ser aplicadas apenas em situações em que os requisitos do software são estáveis e requisitos futuros são previsíveis. Estas situações são difíceis de serem atingidas, uma vez que os requisitos para o desenvolvimento de um software são mutáveis. Dentre os fatores responsáveis por alterações nos requisitos estão a dinâmica das organizações, as alterações nas leis e as mudanças pedidas pelos stakeholders, que geralmente têm dificuldades em definir o escopo do futuro software.

### **2.2 Extreme Programming**

A Extreme Programming (XP) é uma metodologia ágil para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se modificam rapidamente.

Conforme Tabela 1, a metodologia XP, busca trazer valores e motivações para toda equipe envolvida no projeto de desenvolvimento. Levar confiança ao

cliente de que suas necessidades serão respeitadas diante de um informativo de prioridades é de vital importância para que o projeto não seja interrompido.

**Tabela 1 – Valores de Extreme Programming**

FeedBack	Comunicação
Simplicidade	Coragem

Mostradas como uma série de atividades a serem seguidas, as práticas de XP norteiam a equipe desenvolvedora durante o projeto. Os valores citados anteriormente, somados as práticas ( Tabela 2), resultam em um grupo de boas atitudes.

**Tabela 2 – Práticas de Extreme Programming**

Cliente Presente	Jogo do Planejamento	Integração Contínua
Pequenas Versões	Metáforas	Desenvolvimento Orientado a Testes
Projeto Simples	Time Coeso	Refatoração
Testes de Aceitação	Ritmo Sustentável	Padrões de Codificação
Reuniões em Pé	Posso Coletiva	Programção em Pares

### 2.2.1 A Equipe e Seus Membros

Dentro de uma equipe XP, não existem pessoas com ações mais importantes que as outras, não existindo hierarquia. Cada membro faz sua contribuição ao projeto da melhor maneira que for possível.

**Tabela 3 – Membros de Equipe de Extreme Programming**

Gerente de Projetos	Treinador
Analista de Testes	Redator Técnico
Desenvolvedor	Rastreador

### 2.2.2 Ciclo de Vida

Um projeto XP passa pelas seguintes fases: exploração, planejamento inicial, iterações do release, produção, manutenção e morte.

**a) Exploração:**

Objetivo: Investigar possíveis soluções e verificar a viabilidade de tais soluções.

Funcionamento: Os programadores elaboram possíveis arquiteturas e tentam visualizar como o sistema funcionará considerando o ambiente tecnológico (hardware, rede, software, performance, tráfego) onde o sistema irá rodar.

**b) Planejamento inicial:**

Objetivo: Entrar em acordo com o cliente em uma data de entrega para o primeiro release.

Funcionamento: Os programadores, juntamente com o cliente, definem as histórias (use case simplificados) a serem implementadas e as descrevem em cartões. Os programadores assinalam uma certa dificuldade para cada história e, baseados na sua velocidade de implementação, dizem quantas histórias podem implementar em uma iteração. Depois, os clientes escolhem as histórias de maior valor para serem implementadas na iteração – isso é chamado planejamento de iteração. O processo então se repete até terminar as iterações do release. O tempo para cada iteração deve ser de uma a três semanas e para cada release de dois a quatro meses.

**c) Iterações do release:**

Objetivo: Ter uma idéia melhor das tecnologias e do domínio do problema de modo que as iterações poderão ser mais curtas nos releases subsequentes e fazer estimativas mais confiáveis com o que se aprendeu das iterações passadas.

Funcionamento: São escritos os casos de teste funcionais e de unidade. Os programadores vão seguindo mais ou menos o seguinte fluxo de atividades na seguinte ordem (Em cada iteração): escrita dos casos de testes; projeto e refatoramento; codificação; realização dos testes; e integração. À medida que esse fluxo vai sendo seguido, o sistema vai sendo

construído segundo os princípios, valores e práticas apresentados nas seções anteriores

#### d) Manutenção:

Objetivo: Reajustar o processo para conseguir adequar às imprevisibilidades que vão surgindo no decorrer do processo. Exemplo: implementação de novas funcionalidade, chegada de novos membros e a reescrita do código para o seu melhoramento.

Funcionamento: Estando em XP sempre simultaneamente produzindo novas funcionalidades, mantendo o sistema existente rodando, incorporando novas pessoas na equipe e melhorando o código. Mecanismos como: refatoramento, introdução de novas tecnologias, e introdução de novas idéias de arquitetura podem ser utilizados em um projeto XP.

#### e) Morte:

Objetivo: Dar fim à um projeto, seja pelo motivo do cliente já estar satisfeito, seja pela inviabilidade em dar continuidade ao desenvolvimento.

Funcionamento: Quando não mais existir novas histórias, é o momento de finalizar o projeto. É o momento de escrever algumas páginas (de 5 a 10 páginas) sobre a funcionalidade do sistema, um documento que auxilie no futuro a saber como realizar alguma alteração no sistema. Toda a equipe que trabalhou no sistema deve ser reunida para reavaliação. Aproveite a oportunidade de analisar o que pode ter causado queda no sistema e o que fez o projeto avançar.

Figura 1: Ciclo de Vida do XP



Fonte: DevMedia

### **3 Considerações Finais**

Neste trabalho foram apresentadas as principais partes da metodologia ágil extreme programming. Os valores e práticas foram tabelados e o ciclo foi detalhadamente explicado. Toda essa discussão ajuda na compreensão da importância de uma boa metodologia para um bom desenvolvimento de software.

Enquanto as tradicionais possuem um modelo mais engessado para conseguir levar do início ao fim o mesmo escopo, as metodologia ágeis, com toda a sua flexibilidade, conseguem se adaptar aos mais variados ambientes e situações. Encarar os problemas de frente, ao invés de modelar um minimundo perfeito sem nenhuma alteração é uma das principais consequências de adotar um modelo mais flexível e transparente ao cliente. Isso faz com que o número de adeptos de metodologias ágeis cresçam cada vez mais.

### **REFERÊNCIAS**

Gomes, Fabio. Integrando XP as principais metodologias ágeis. Disponível em: <<http://www.devmedia.com.br/integrando-xp-as-principais-metodologias-ageis/30989/>>. Acesso em: Agosto. 2017.

Dias, Thiago e Oliveira, Jader. Adoção da Metodologia Extreme Programming para Construção de Software. CEFET-MG, Divinópolis.

Soares, Michel dos Santos. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. UNIPAC-MG, Conselheiro Lafaiete

Bona, Cristina e Thiry, Marcello. Processo de Software: Um estudo de Caso em XP. UFSC, Santa Catarina e UNIVALI, Vale do Itajaí.