

UMA PROPOSTA DE UTILIZAÇÃO DA PROTOTIPAÇÃO EVOLUCIONÁRIA COM TÉCNICAS DO TSP

Dienefer Santos¹

Dienefer Fialho²

Gabriela Córdova Medeiros³

Gilleanes Thorwald Araujo Guedes⁴

Resumo:

O objetivo deste artigo é apresentar uma abordagem de processo de desenvolvimento de software que combina características do modelo de Prototipação Evolucionária com diretrizes do Team Software Process (TSP). A Prototipação Evolucionária se mostra ideal para o desenvolvimento em situações em que os requisitos não são compreendidos totalmente, porém este processo pode muitas vezes ocasionar a degradação do código e a aplicação de técnicas de programação inadequadas devido ao fato de passar por muitas fases de validação com os clientes. Em contrapartida, o TSP sugere um conjunto de scripts de processos, métricas, formulários e métodos que fazem parte do Personal Software Process (PSP), estas técnicas são utilizadas para guiar o trabalho individual dos desenvolvedores de software. Estes elementos guiam os desenvolvedores em criar equipes eficazes através dos procedimentos estatísticos gerados que, quando empregados corretamente, provêm dados importantes para que os desenvolvedores de software conheçam melhor seus compromissos e dificuldades, tornando suas rotinas de trabalho mais previsíveis, eficientes e planejáveis com o intuito de acelerar a melhoria contínua do processo. Levando isto em consideração, produziu-se uma nova abordagem de processo com o objetivo de mitigar as desvantagens da Prototipação Evolucionária utilizando as técnicas TSP.

Palavras-chave: prototipação evolucionária, team software process, processo de software

Modalidade de Participação: Iniciação Científica

UMA PROPOSTA DE UTILIZAÇÃO DA PROTOTIPAÇÃO EVOLUCIONÁRIA COM TÉCNICAS DO TSP

¹ Aluno de graduação. dieneferfialho@gmail.com. Autor principal

² Aluno de graduação. dieneferfialho@gmail.com. Apresentador

³ Aluno de graduação. gabiceemi@gmail.com. Co-autor

⁴ Docente. gilleanesguedes@unipampa.edu.br. Orientador

UMA PROPOSTA DE UTILIZAÇÃO DA PROTOTIPAÇÃO EVOLUCIONÁRIA COM TÉCNICAS DO TSP

1 INTRODUÇÃO

Para que um software seja desenvolvido de forma consistente, é preciso aliar boas práticas da engenharia de software com um processo de desenvolvimento robusto e eficiente. De acordo com Sommerville (2007), um processo de desenvolvimento pode ser definido como um conjunto de atividades que leva à produção de um produto de software com maior qualidade.

O processo de Prototipação Evolucionária é definido por Wazlawick (2013) como um modelo que permite uma melhor adaptação às mudanças de requisitos, através da criação de protótipos que auxiliam na compreensão da arquitetura e da interface, o que é fundamental quando não é possível conhecer estes aspectos *a priori*.

O Team Software Process (TSP) proposto por Humphrey (2000), possui uma estrutura de processo definida, com foco em beneficiar as equipes de gerentes e engenheiros a organizar e produzir produtos de software de modo eficaz. É importante destacar que é necessário que os engenheiros tenham conhecimento sobre o Personal Software Process para que possam participar do TSP, a fim de que tenham resultados em produtos com maior qualidade.

Considerando que o processo de Prototipação Evolucionária tem tendência a degradar o código devido às muitas versões produzidas, o objetivo deste trabalho é propor um novo processo que combine características do modelo de Prototipação Evolucionária com diretrizes do TSP. O processo proposto tem o intuito de mitigar o desgaste de código causado pelas constantes iterações, bem como melhorar a qualidade do software produzido, por meio da capacidade de diminuir a quantidade de defeitos introduzidos e do aumento de produtividade que as práticas do TSP proporcionam.

Este artigo é estruturado da seguinte forma: na seção 2 é apresentada a fundamentação teórica, contextualizando a Prototipação Evolucionária e o Team Software Process (TSP); em seguida, na seção 3 descrevem-se os resultados e, após isso, apresentam-se as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados os conceitos de Prototipação Evolucionária e *Team Software Process* para contextualizar a proposta.

2.1 MODELO DE PROTOTIPAÇÃO EVOLUCIONÁRIA

De acordo com Wazlawick (2013), a Prototipação Evolucionária permite que, em casos onde é difícil se comunicar com o cliente ou quando os requisitos não estão bem claros, se torne mais fácil chegar a um acordo sobre o que deve ser desenvolvido. A prototipação começa com os requisitos que são melhor compreendidos e, conforme o desenvolvimento for avançando, esse protótipo é refeito, adaptado e alinhado às fases de elaboração do projeto. Duas abordagens de prototipação se destacam, a primeira é chamada de *throw-away*, onde os protótipos gerados tem como finalidade apenas estudar aspectos do sistema, compreender melhor os requisitos e reduzir riscos, sendo então descartados. Na abordagem *cornerstone*,

que é na qual iremos focar neste artigo, além de ser utilizada com o mesmo objetivo da anterior, neste cenário o protótipo vai evoluindo até se tornar o sistema a ser entregue.

Na fase inicial deste modelo, os desenvolvedores devem identificar as funcionalidades básicas do software. Em seguida, no projeto e implementação do protótipo inicial, uma representação dos aspectos do software é definida, baseada nos requisitos previamente elicitados e em características visíveis ao usuário. Ainda nesta etapa, ocorre a implementação de um protótipo com as devidas funcionalidades exigidas. Este protótipo passa pela validação, que é feita através de uma reunião realizada com o cliente. Quando o protótipo não é aprovado, a equipe de desenvolvimento volta a fazer mudanças até que se torne aceitável, conforme a avaliação do cliente. Assim que aprovado, parte-se para a etapa de finalização e liberação do produto, onde este deve ser implementado levando-se em consideração os critérios de qualidade.

A Prototipação Evolucionária tem como prioridade encontrar os defeitos nos protótipos iniciais para que não sejam propagados ao sistema final. Entretanto, Pressman (2011) reconhece que ao serem realizadas muitas validações com os clientes, acabam sendo empregues algumas técnicas de programação inadequadas na criação dos protótipos, o desenvolvedor pode acabar se habituando com estas alternativas, permitindo que as mesmas se tornem parte do sistema final. Wazlawick (2013) aponta outro ponto fraco, este modelo não é muito efetivo em relação à previsão de cronograma da equipe de gerência do projeto, já que é difícil avaliar quando cada fase decorre.

2.2 TEAM SOFTWARE PROCESS (TSP)

Conforme declarado por Pressman (2011), o TSP consiste em um conjunto de *scripts* de processos, métricas, formulários e padrões/métodos que fazem parte do *Personal Software Process* (PSP) para guiar o trabalho individual dos desenvolvedores de software. O *Personal Software Process* (PSP) foi proposto por Humphrey (2000), onde alega que os indivíduos são diferentes do mesmo modo que o planejamento de trabalho dos mesmos. Para isso são utilizados um conjunto de métodos, formulários e roteiros para planejar, medir e gerenciar o trabalho individual que tem como objetivo geral a produção de produtos de software sem defeitos, respeitando prazos e custos planejados.

Os scripts de processo servem para documentar os critérios de entrada do processo, suas etapas e os critérios de saída, com o objetivo de guiar o desenvolvedor ao usar o processo. As métricas servem para medir o processo e o produto. Elas fornecem informações sobre como o processo está funcionando e o status do trabalho. Os formulários fornecem uma estrutura conveniente e consistente para coletar e reter os dados. Os padrões/métodos provêm definições consistentes que orientem o trabalho e a coleta de dados. A fim de prevenir defeitos, o TSP também estabelece o gerenciamento da qualidade através das anotações de causas dos erros que já ocorreram.

Estes elementos guiam os desenvolvedores em criar equipes eficazes através dos procedimentos estatísticos gerados que, quando empregados corretamente, provêm dados importantes para que os desenvolvedores de software conheçam melhor seus compromissos e dificuldades, tornando suas rotinas de trabalho mais previsíveis, eficientes e planejáveis com o intuito de acelerar a melhoria contínua do processo.

De acordo com Humphrey (2000), a metodologia TSP conduz o desenvolvimento de software através de vários ciclos rápidos até atingir o produto final. Cada ciclo guia a equipe através de sete fases, sendo elas: lançamento, estratégia de planejamento, requisitos, projeto, implementação, teste e análise, também conhecida como *postmortem*.

Na fase de lançamento são definidos os membros da equipe e seus respectivos papéis, que são escolhidos baseados em sua experiência, conhecimento e maturidade. Além disso, são definidas as reuniões que tem como objetivo verificar o andamento do projeto e o desempenho da equipe.

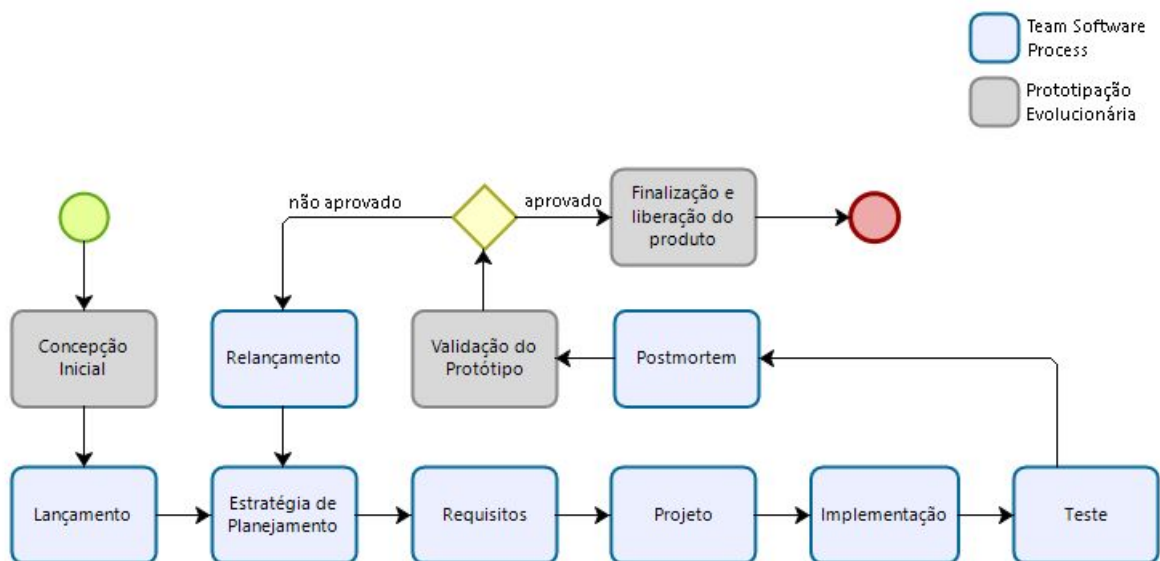
Uma estratégia é estabelecida para realizar as tarefas durante o ciclo, criando um modelo conceitual, estimativas de tamanho e tempo de desenvolvimento do produto. Nesta fase a equipe começa a produzir o plano de gerência de configuração, que é fundamental para o controle da versão do produto. Se houver algum problema na estrutura do processo, a estratégia é retificada. É necessário um plano detalhado para saber com exatidão o que deve ser feito e quando será feito, com a finalidade de planejar um método de trabalho equilibrado entre a equipe.

A fase de requisitos é então iniciada, na qual realiza-se a especificação dos mesmos, prevendo o que o produto será, este artefato posteriormente é utilizado para avaliar se o software desenvolvido está de acordo com o planejado. A equipe irá decidir como construir o produto durante a fase de projeto, produzindo uma especificação completa e precisa de como o produto será construído. Um dos passos definidos nesta etapa é a decisão quanto aos componentes e as funcionalidades que serão desenvolvidas em cada ciclo.

A etapa de implementação é a geração do código fonte do produto, seguindo o que foi proposto pelo projeto, uma das características desta fase no TSP é a utilização de padrões. Os módulos desenvolvidos no ciclo são por fim testados e integrados para formar o produto final, com o objetivo de verificar se tais módulos produzirão um produto de qualidade e atenderão aos requisitos solicitados. No *postmortem* ocorre a análise, que tem como objetivo prover um método para avaliar todo o processo de desenvolvimento e estipular melhorias.

3 RESULTADOS E DISCUSSÃO

A nossa proposta introduz técnicas do TSP no modelo de Prototipação Evolucionária com o objetivo de conter as desvantagens apontadas anteriormente, como, degradação do código e técnicas de programação inadequadas. A abordagem de prototipação escolhida foi a *cornerstone* pois, como citado na seção 2.1, essa abordagem permite que os protótipos evoluam até o produto final. A figura abaixo demonstra o fluxo das mesclagem dos dois modelos de acordo com esta concepção:



Fonte: do autor, 2018.

A concepção inicial é a primeira etapa, onde procura-se ter um entendimento básico do problema e definir o escopo. Na etapa seguinte, no lançamento do projeto, determina-se o papel de cada membro, suas estimativas de esforço e o objetivo da equipe com base na concepção inicial. Com o lançamento do projeto finalizado, parte-se para o planejamento, em que será produzido uma estratégia para direcionar o desenvolvimento do trabalho.

A documentação de especificação de requisitos começa a ser produzido na etapa seguinte, chamada requisitos, contendo todos os requisitos funcionais e não-funcionais conhecidos do produto até o momento. As etapas de projeto e implementação ocorrem da forma como foi citada anteriormente na seção 2.2, após a implementação ser concluída, o código deve ser testado e documentado utilizando o padrão de formulários previsto no TSP. E por fim a última etapa do fluxo do TSP, a postmortem, que faz o uso da documentação gerada durante o processo, que fazem parte do conjunto de artefatos do PSP. Esta análise abrange tanto os dados qualitativos quanto os dados quantitativos e permite determinar quais elementos do projeto foram bem-sucedidos ou malsucedidos, auxiliando no planejamento da equipe.

Com uma versão do protótipo já integrada, deve-se passar para a etapa de validação do protótipo. Esta etapa requer a participação do cliente, para se certificar de que o sistema atende às necessidades e expectativas do mesmo. Caso o protótipo não seja aprovado, um novo ciclo se inicia a partir da etapa de relançamento do projeto, considerando as observações do cliente, as iterações acontecem até chegar a um protótipo que seja aprovado. Quando o protótipo é aprovado, parte-se para finalização e liberação do produto.

4 CONSIDERAÇÕES FINAIS

O uso de modelos de processos bem definidos para o desenvolvimento de software proporciona benefícios no que diz respeito a qualidade dos produtos gerados. Entretanto, mesmo um processo bem definido, não é perfeito e está sempre passível de melhoria já que compreende diversas variáveis, que vão do nível organizacional até a capacidade dos recursos humanos envolvidos. Este artigo descreve alguns conceitos de processos de software e propõe uma nova abordagem, combinando o Processo de Prototipação Evolucionária com diretrizes do TSP.

Existem cenários onde os requisitos não são facilmente identificados, Prototipação Evolucionária mostra-se como uma abordagem eficiente nestas ocasiões, pois permite que o desenvolvimento seja realizado em iterações a partir de uma concepção inicial, até que todos os requisitos sejam identificados. Porém sua natureza iterativa tende a gerar problemas, como desgaste de código devido a defeitos inseridos no mesmo durante os muito protótipos, falta de documentação, retrabalho e atrasos em entregas.

Ao elaborar um projeto seguindo o TSP, as equipes são auto-gerenciadas, planejando o próprio trabalho, acompanhando o progresso e realizando uma sequência de tarefas bem definidas. Evitando que a qualidade baixe por prover documentação, revisões e inspeções, e o essencial que é as práticas de implementação adequadas. Essas diretrizes permitem que, por exemplo, um desenvolvedor aborde um problema de uma maneira organizada e consistente desde o início do ciclo de desenvolvimento, podendo assim, mitigar os problemas gerados nas fases incrementais.

Apesar de esta abordagem ainda não ter sido testada em um cenário real, conclui-se que é possível potencializar o ganho de produtividade, qualidade de código e controle do projeto por meio da inserção de técnicas do TSP no processo de Prototipação Evolucionária. Como trabalho futuro será feita a validação desta técnica através de um projeto de pesquisa,

os voluntários serão divididos entre equipes que utilizam Prototipação Evolucionária com TSP e equipes que utilizarão apenas a Prototipação Evolucionária.

REFERÊNCIAS

HUMPHREY, Watts S. **Introduction to the team software process**. Addison-Wesley Professional, 2000.

PRESSMAN, S. Roger. Engenharia de Software: Uma Abordagem Profissional. 7ª. **Edição. Rio de Janeiro—RJ. Editora McGraw-Hill**, 2011.

SOMMERVILLE, Ian et al. **Software engineering**. Addison-wesley, 2007.

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. Elsevier Brasil, 2013.