



Acta Scientiarum. Technology

ISSN: 1806-2563

eduem@uem.br

Universidade Estadual de Maringá
Brasil

Maracci, Francisco Virginio; Colanzi, Thelma Elita; Padovan Jubileu, Andrea; Creres Rosa, Marcelo
Vinícius; Said Branco, Emanuelle Fabiana

WAEI/MSE: uma instância do processo WAE para micro e pequenas empresas de software

Acta Scientiarum. Technology, vol. 31, núm. 2, 2009, pp. 141-149

Universidade Estadual de Maringá

Maringá, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=303226524002>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica

Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal

Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

WAEI/MSE: uma instância do processo WAE para micro e pequenas empresas de software

Francisco Virginio Maracci^{1*}, Thelma Elita Colanzi¹, Andrea Padovan Jubileu², Marcelo Vinícius Creres Rosa² e Emanuelle Fabiana Said Branco¹

¹Departamento de Informática, Universidade Estadual de Maringá, Av. Colombo, 5790, 87020-900, Maringá, Paraná, Brasil.

²Faculdade de Informática de Presidente Prudente, Universidade do Oeste Paulista, Presidente Prudente, São Paulo, Brasil.

*Autor para correspondência. E-mail: franciscomaracci@uol.com.br

RESUMO. Em Micro e Pequenas Empresas de Software (MPEs), geralmente, não há um processo sistemático de desenvolvimento de software. Um dos motivos é que essas empresas consideram os processos tradicionais extensos. Assim, normalmente, essas empresas são classificadas em nível 1 de maturidade do CMMI (*Capability Maturity Model Integration*). O objetivo deste artigo é apresentar uma instância do processo WAE (*Web Application Extension*), a fim de auxiliar especificamente as MPEs que desenvolvem software Web a alavancarem o nível de capacidade de seu processo de desenvolvimento de software. A instância foi validada durante sua aplicação no desenvolvimento de um software comercial com o apoio da Coordenadoria de Web da Unoeste e em outros três estudos de caso.

Palavras-chave: engenharia para Web, UML, CMMI, micro e pequenas empresas.

ABSTRACT. **WAEI/MSE: an instance of the WAE process for micro and small software enterprises.** In Micro and Small Enterprises (MSE) of software, there is usually not a systematic software development process. One of the reasons is that these enterprises consider the traditional processes extensive. Thus, these enterprises are usually classified in level 1 of maturity of the CMMI (*Capability Maturity Model Integration*). The objective of this paper is to present a proposal for a software development process, in order to aid specifically MSEs that build Web software, to help them increase the capacity level of their processes. The process was validated during its application in development of software with the support of the Unoeste Web Coordination.

Key words: Web engineering, UML, CMMI, micro and small enterprises.

Introdução

O setor de software é composto por uma maciça participação de micro e pequenas empresas – MPEs. Essas empresas, geralmente, não possuem processo sistemático de desenvolvimento de software que as direcione rumo à qualidade do produto. Um processo de qualidade conduz a um produto de qualidade. O CMMI é um modelo de referência que permite às empresas de software alavancarem seu nível de maturidade por meio da melhoria da capacidade de seus processos (YOO et al., 2006; HABRA et al., 2008). As MPEs, comumente, são classificadas no nível de maturidade 1 do CMMI em que os processos são, normalmente, caóticos e realizados de maneira *ad-hoc*. Nas organizações onde os processos têm essa característica, o sucesso está atrelado às competências individuais tornando a organização dependente desses indivíduos e, frequentemente, o cronograma e o orçamento previstos são excedidos.

É possível observar que muitos projetos de desenvolvimento de sistemas, neste tipo de organização, não atingem o resultado esperado, pela falta de gerenciamento do projeto e dos requisitos do produto pretendido. O nível 2 do CMMI apresenta um processo efetivo de planejamento e gerenciamento de projeto, em que os controles sobre os procedimentos, compromissos e atividades são bem fundamentados, além do gerenciamento dos requisitos. O planejamento e gerenciamento são baseados na experiência adquirida em projetos anteriores que obtiveram sucesso (YOO et al., 2006).

Com base nessas premissas, foi elaborado um projeto de pesquisa com o objetivo de investigar e, se necessário, adequar ou elaborar um processo de desenvolvimento de software para as organizações que desenvolvem software Web, a fim de torná-las mais eficientes e competitivas. Para isso, foram realizados estudos, a partir dos quais foi possível decidir pela criação de uma instância do processo

WAE (*Web Application Extension*) (CONALLEN, 2002) para MPEs com características apontadas pelo nível 1 de maturidade do CMMI. O objetivo dessa instância é possibilitar que esse grupo de empresas aprimore o nível de capacidade de seu processo de desenvolvimento de software, uma vez que não possui o processo formalizado e, geralmente, considera os processos tradicionais existentes extensos e muito burocráticos.

Este artigo está organizado em três seções. A seção Material e Métodos apresenta a metodologia do trabalho, além de uma breve revisão do processo WAE e de uma descrição detalhada da instância proposta para o WAE denominada de WAEI/MSE (*Web Application Extension Instance/Micro and Small Enterprises*). Nesta mesma seção, ainda, apresenta-se comparação sucinta entre os processos WAE e WAEI/MSE. A seção Resultados e Discussão aborda o processo de validação da instância e os resultados obtidos. E, por fim, a seção Conclusão, apresenta algumas considerações finais.

Material e métodos

Ao longo da investigação bibliográfica, foram estudados os processos de software: UP (*Unified Process*) (JACOBSON et al., 1999), o processo proposto por Larman (2000), o processo WAE (CONALLEN, 2002) e XP (*Extreme Programming*) (BECK, 1998). Além disso, foi estudada a abordagem por estágios do modelo CMMI (YOO et al., 2006). Na sequência, o WAEI/MSE foi elaborado e, posteriormente, validado por meio de estudos de caso. A seguir, são apresentados o processo WAE e sua instância WAEI/MSE.

Modelagem de aplicações Web

O *Web Application Extension* (WAE) (CONALLEN, 2002) é um processo de desenvolvimento de software Web baseado na iteratividade, em que as fases planejamento, requisitos, análise, projeto, implementação, teste e avaliação são repetidas até que cada incremento esteja ajustado às tarefas que desempenhará e seja implantado (Figura 1). As atividades do processo e os artefatos produzidos são de responsabilidade das equipes de trabalho que em vários momentos se relacionam. As equipes do WAE são: arquitetura, requisitos, análise, UX (experiência do usuário), projeto, implementação, teste e implantação. Os artefatos produzidos no WAE são agrupados em conjuntos para melhor entendimento.

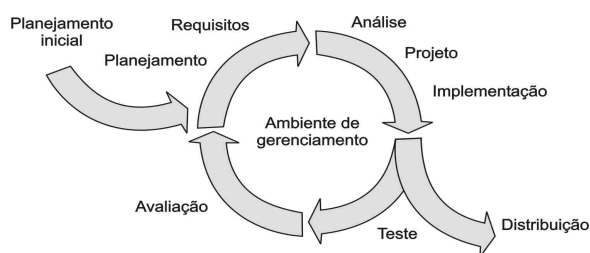


Figura 1. Processo iterativo (adaptado de CONALLEN, 2002).

O conjunto de gerenciamento de projeto descreve os planos do mesmo para mostrar o seu cronograma de desenvolvimento. No conjunto de domínio, é desenvolvido o modelo de domínio, que descreve o contexto do negócio e do domínio no qual o sistema se insere, e o glossário que inclui definições de siglas, termos-chave e conceitos. Na sequência, é desenvolvido o conjunto de requisitos que descreve o quê o sistema deverá fazer, tendo como artefatos: documento de diretrizes de experiência do usuário, documento de visão, modelo de caso de uso e especificação dos casos de uso.

O conjunto de análise permite transformar os requisitos do sistema em algo implementável por meio da descrição e do protótipo da arquitetura de software, do diagrama de sequência e do roteiro. As informações adquiridas com os conjuntos anteriores são, então, refinadas no conjunto de projeto para gerar o modelo que pode ser mapeado diretamente para o código. Os principais artefatos produzidos no conjunto de projeto são: classes de projeto e páginas Web.

No conjunto de implementação, os artefatos anteriormente produzidos são utilizados junto às ferramentas de desenvolvimento, mapeando os requisitos do sistema em código. São realizadas separadamente, pela equipe de UX, as partes visuais e, pela equipe de implementação, o domínio do negócio. Na sequência, inicia-se o conjunto de testes com o objetivo de aumentar a confiabilidade no sistema, verificando, assim, a conformidade dos componentes desenvolvidos (artefatos executáveis) em relação à documentação do software.

O conjunto de implantação descreve, por meio do plano de implantação, as instruções de instalação dos componentes do sistema. Neste momento, é necessário estabelecer o sistema e atualizá-lo, quando necessário, realizando a reiteração das atividades anteriormente citadas, revendo e refinando o progresso com as partes interessadas, avaliando cada iteração e ajustando, caso necessário, a próxima iteração.

Modelagem de aplicações Web para micro e pequenas empresas

Considerando os processos de software estudados, que aplicações Web exigem ciclos mais otimizados de desenvolvimento (NORTON, 1999 apud PRESSMAN, 2002) e, ainda, considerando pesquisas realizadas sobre a importância da documentação de software para o entendimento do domínio do negócio, da implementação e da aplicação (ARISHOLM et al., 2006), a WAEI/MSE procura ser enxuta inserindo somente os artefatos considerados de alta importância nas pesquisas para o entendimento dos domínios anteriormente citados. O processo WAEI/MSE utiliza o processo WAE como metamodelo, uma vez que se trata de um processo voltado especificamente para o desenvolvimento de software Web. Entretanto, poderiam ter sido utilizados outros modelos de processo, tal como o UP, ou até mesmo metodologias e diretrizes ágeis como o XP, integradas à instância.

Seguindo os mesmos princípios do WAE, a instância WAEI/MSE propõe um processo de desenvolvimento incremental, baseado em casos de uso, e indica o desenvolvimento do protótipo do produto. Essa instância consiste em cinco fases: Requisitos, Análise, Projeto, Implementação e Teste, apresentadas nas subseções a seguir. As atividades de cada fase são realizadas em paralelo pelas equipes de UX e de projeto. A equipe de UX é responsável pelo design gráfico que a aplicação deverá ter para satisfazer as necessidades dos usuários finais, cujo objetivo é melhor acessibilidade e usabilidade do sistema a ser desenvolvido (LARSEN; CONALLEN, 2002). Já a equipe de projeto é responsável por realizar a análise de negócio, requisitos funcionais e não-funcionais do sistema.

A Figura 2 ilustra os artefatos das fases do processo WAEI/MSE por equipe. Assim como o WAE, para cada incremento, todas as fases são executadas de forma sequencial, e a cada iteração são refinados os artefatos de cada fase até que cada um deles esteja conforme o desejado. Nesta figura, as setas indicam as dependências entre os artefatos.

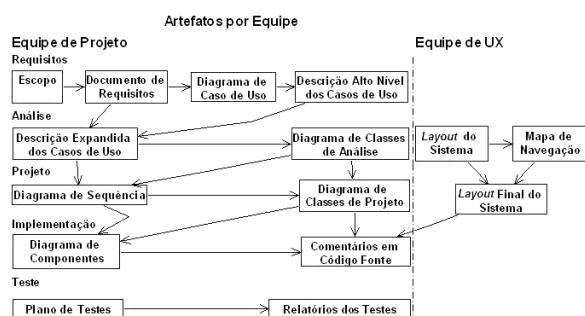


Figura 2. Artefatos produzidos no processo WAEI/MSE.

De forma geral, na Fase de Requisitos são descritos o domínio do projeto, as funções que o sistema realizará, as restrições do projeto e os recursos necessários para o desenvolvimento. Na Fase de Análise, as funcionalidades são descritas detalhadamente, os recursos necessários e as restrições levantadas são analisados. A Fase de Projeto utiliza os artefatos desenvolvidos nas fases anteriores para mostrar os detalhes internos do sistema, de acordo com o comportamento do negócio requerido, mostrando, por exemplo, as classes necessárias para cada funcionalidade. A Fase de Implementação utiliza-se dos artefatos já produzidos e de ferramentas para iniciar a implementação do projeto. A Fase de Teste visa à realização de validações e testes com o intuito de aumentar a confiabilidade do produto.

As próximas subseções descrevem cada fase do WAEI/MSE e o que deve ser realizado em cada uma delas. Para ilustrar a sua aplicação são mostrados os artefatos produzidos em um estudo de caso desenvolvido, que é apresentado parcialmente, por uma questão de limitação de espaço, mas pode ser encontrado na íntegra em <http://www.din.uem.br/~teclopes>.

Fase de requisitos

A Fase de Requisitos inicia-se pela definição do escopo do sistema que deve abordar, em linhas gerais, o domínio no qual o sistema está inserido, os benefícios que o sistema trará e suas funcionalidades, sem mencionar aspectos tecnológicos. Uma vez definido o escopo do sistema é necessário realizar a elicitación dos requisitos necessários para o seu desenvolvimento. Essa atividade é realizada pela equipe de projeto e permite o desenvolvimento do documento de requisitos, descrito por meio de declarações das funcionalidades do sistema denominadas requisitos funcionais, das tecnologias necessárias para a sua implementação e dos requisitos não-funcionais. Sugere-se o padrão IEEE Standard 830-1998 (POSTON, 1985) como formato para o Documento de Requisitos de Software.

É possível e recomendado incluir um estudo de viabilidade técnica e econômica para o projeto neste artefato. Na Figura 3, é apresentado um fragmento do Documento de Requisitos do ConfSystem, um sistema para controle administrativo e financeiro de uma empresa de confecções, que é utilizado neste trabalho de pesquisa para ilustrar vários outros artefatos.

Após elaborar o documento de requisitos, a equipe de projeto deve desenvolver o diagrama de caso de uso e suas descrições em alto nível, descrevendo, assim, as funcionalidades do sistema.

Larman (2000) afirma que a descrição em alto nível dos casos de uso deve possuir identificação, os atores participantes e uma descrição das ações dos atores e respostas do sistema ao realizar o caso de uso. Na Figura 4, pode-se observar a descrição do caso de uso Vender Produtos (à cima) e o diagrama de caso de uso do ConfSystem (à baixo).

Cadastros: Produtos: destina-se a cadastrar os produtos que serão fabricados e vendidos pela empresa. Dados necessários: tipo do produto, descrição do produto, tamanho, cor, modelo, quantidade, preço de venda, preço de custo, margem de lucro, estoque mínimo e foto do produto.
Movimentações: Venda: registra as vendas realizadas aos clientes. Ocorre quando um cliente efetua uma compra, confirma um orçamento previamente efetuado ou faz um pedido. Sendo assim, o funcionário deverá informar os seguintes dados: orçamento, representante, pedido, cliente, data da venda, itens (quantidade, valor unitário e valor total) e condição de pagamento. Para pagamento, à vista deverá ser informada também a data do pagamento. O sistema deverá atualizar o estoque de produtos, gerar contas a receber (no caso de pagamento a prazo) e calcular a comissão do representante, gerando uma conta a pagar.
Relatórios: Pessoas em débito: o relatório informará quais são os clientes que estão em débito com a empresa. Para tanto, é necessário informar uma data de início e término para a pesquisa que exibirá o nome do cliente, telefone, fatura, data do vencimento e valor total dos débitos.

Figura 3. Exemplo parcial do documento de requisitos do ConfSystem.

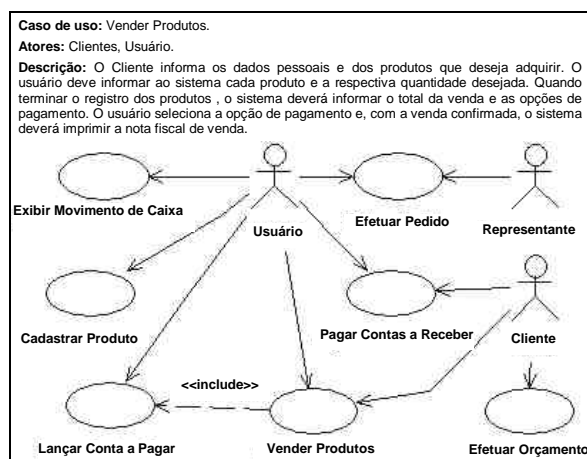


Figura 4. Descrição alto nível do caso de uso vender produto e diagrama de caso de uso parcial do ConfSystem.

Fase de análise

A Fase de Análise refina os artefatos produzidos anteriormente com o intuito de detalhar o comportamento esperado do software. Nesta fase, os artefatos produzidos pela equipe de projeto são: Descrição Expandida dos Casos de Uso, Diagrama de Classes de Análise. A equipe de UX desenvolve o *Layout* do Sistema e o Mapa de Navegação.

As descrições Alto Nível dos Casos de Uso realizadas na Fase de Requisitos agora são detalhadas e denominadas de Descrição Expandida de Caso de Uso. Este modelo de descrição é estruturado em

duas sequências de eventos: a típica que descreve os eventos ocorridos quando o caso de uso é executado normalmente e as sequências alternativas que descrevem as possíveis exceções (LARMAN, 2000). A Figura 5 ilustra este artefato para o Caso de Uso Vender Produtos.

O Diagrama de Classes de Análise é utilizado para mostrar as classes que poderão ser implementadas pelo sistema (Figura 6). Este diagrama ilustra as classes sem informar os detalhes que elas possuirão, como proposto por Larman (2000). Dessa forma, as classes deste artefato são consideradas conceitos do domínio e, por isso, não têm métodos.

Caso de Uso: Vender Produtos Atores: Cliente, Usuário Sequência Típica de Eventos 1 Este caso de uso inicia quando o cliente informa seus dados pessoais ao funcionário da empresa.(usuário). 2 Cliente confirma os dados. 5 Para cada produto que deseja adquirir, o Usuário informa os dados e confirma a inclusão do item. 7 Usuário confirma o final da inclusão de produtos. 9 Usuário seleciona e confirma a opção desejada pelo cliente.	Resposta do Sistema 3 Sistema busca e mostra os dados do cliente. 4 Sistema redireciona para a página de itens de venda. 6 Para cada produto confirmado, o Sistema verifica a quantidade disponível. 8 Sistema redireciona para a página de opções de pagamento e informa o total da venda. 10 Sistema registra a venda com suas parcelas e imprime a nota fiscal. Sequência Alternativa de Eventos 1 ... 9 O Cliente solicita o cancelamento da operação a qualquer momento, a venda deve ser cancelada e o caso de uso encerrado.
---	--

Figura 5. Descrição expandida do caso de uso vender produtos.

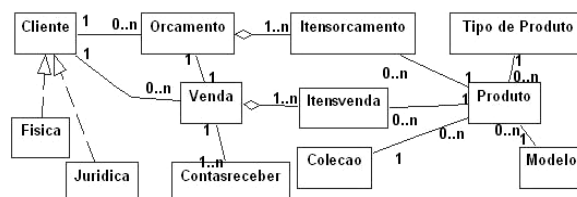


Figura 6. Diagrama de classes de análise (parcial).

O *Layout* do Sistema é constituído por representações gráficas das páginas Web, construídas pela equipe de UX, que, reunidas, constituirão a interface do sistema. Este *layout* auxilia as partes interessadas a compreenderem melhor o projeto e a identificarem falhas ou características ainda não observadas. No desenvolvimento do *layout*, deve-se aplicar testes

de acessibilidade e usabilidade com o objetivo de avaliar o sistema, antes de mostrá-lo ao cliente. Os *layouts* podem ser considerados como protótipos não-operacionais do sistema para a avaliação junto ao cliente e devem ser refinados até que atendam às necessidades dele.

O Mapa de Navegação é desenvolvido para informar quais são as possíveis navegações a serem realizadas pelos usuários, ou seja, partindo de uma determinada página, quais outras páginas podem ser alcançadas. Para elaboração deste artefato (Figura 7), utilizou-se a notação do diagrama de classes da UML, representando as páginas devidamente identificadas com o estereótipo `<<page>>` e os caminhos válidos entre elas.

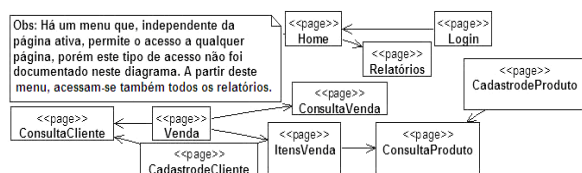


Figura 7. Mapa de navegação parcial do sistema ConfSystem.

Uma questão analisada foi: como modelar o Mapa de Navegação para um sistema que possui um menu por meio do qual, a partir de qualquer página, é possível acessar uma outra? A decisão tomada foi que este tipo de acesso não será contemplado no Mapa de Navegação do processo WAEI/MSE. Contudo, pode-se incluir uma nota no Mapa de Navegação informando que o sistema possui o menu que permite esse tipo de acesso, conforme ilustrado na Figura 7.

Fase de Projeto

Os artefatos produzidos na Fase de Projeto são:

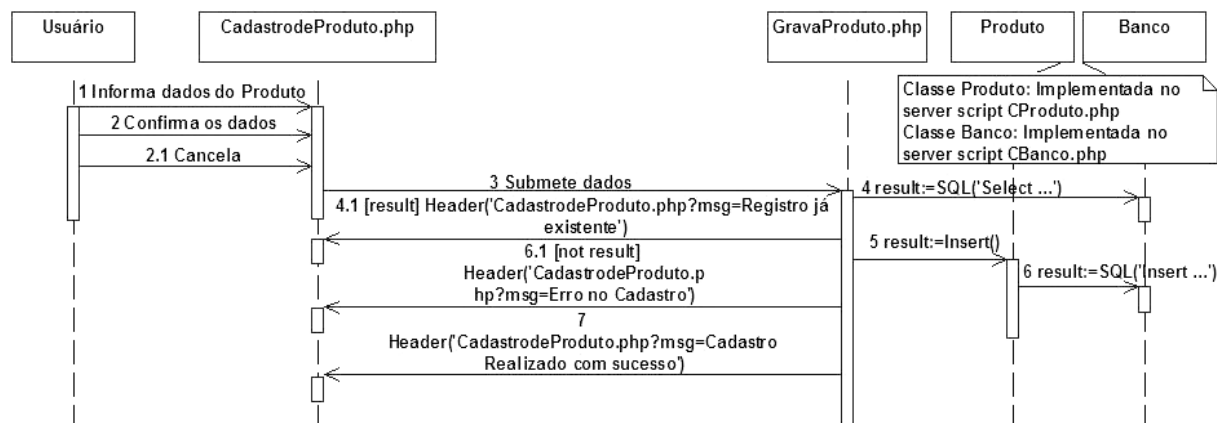


Figura 8. Diagrama de sequência para o caso de uso cadastrar produto.

Diagrama de Sequência, Diagrama de Classes de Projeto e *Layout* Final do Sistema.

Os Diagramas de Sequência são desenvolvidos para cada caso de uso do sistema e deverão mostrar a sequência típica e a alternativa dos eventos, relatadas na Descrição Expandida dos Casos de Uso, devidamente enumeradas. Para melhor demonstrar o comportamento de um software Web, o Diagrama de Sequência proposto ilustrará as páginas do cliente e do servidor, as classes e os atores, utilizando-se, respectivamente, dos estereótipos `<<client script>>`, `<<server script>>`, `<<class>>` e `<<actor>>`. A Figura 8 ilustra um Diagrama de Sequência para o Caso de Uso Cadastrar Produto. Os métodos representados e as classes utilizadas nos Diagramas de Sequência determinarão quais serão as classes e os possíveis métodos a serem implementados pelo sistema. Entretanto, observe-se, por meio da Figura 8, que o Diagrama de Sequência também pode representar as interações Cliente/Servidor de um documento ou entre vários documentos de uma aplicação Web. Assim, é possível representar: (i) a abstração dos métodos funcionais para o desenvolvimento do Diagrama de Classes de Projeto, (ii) o complemento e relação dos `<<client script>>`, `<<server script>>` e `<<page>>` com o Mapa de Navegação e (iii) os elementos de entrada para o Diagrama de Componentes, permitindo a rastreabilidade das atividades em caso de manutenção e validação.

O Diagrama de Classes de Projeto, agora refinado, deverá possuir as classes que serão implementadas e os atributos e métodos pertencentes a elas (Figura 9).

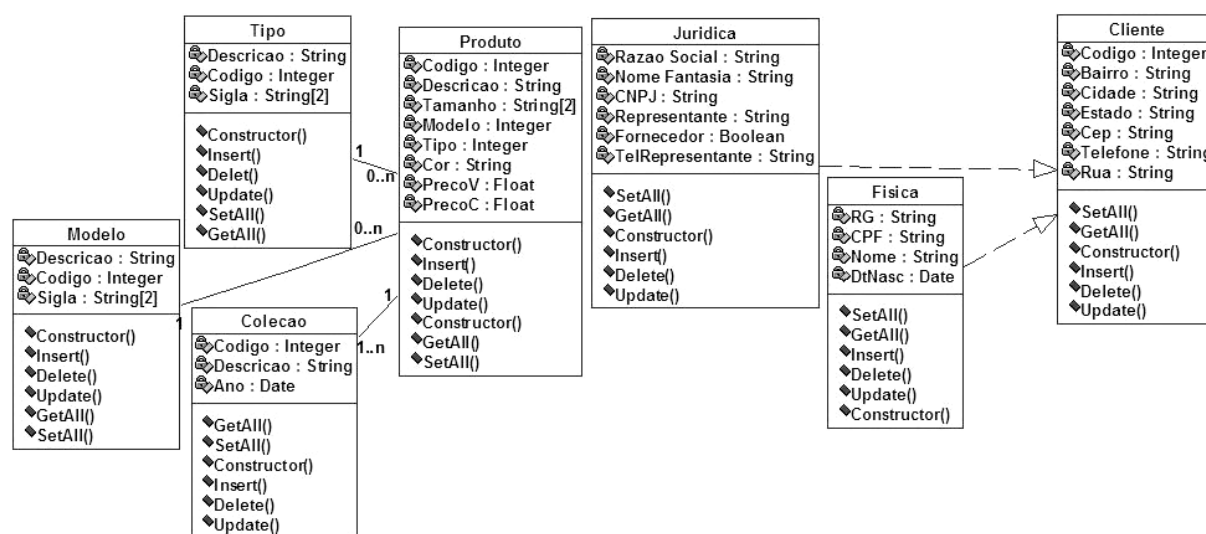


Figura 9. Diagrama de classes de projeto (parcial).

O *Layout* Final do Sistema representa, por meio de imagens, o design gráfico final do sistema, depois de aprovado pelo cliente. Para tanto, devem ser detalhados, quando necessário, as fontes, tamanhos, cores e outros atributos necessários à construção do software.

A Fase de Implementação inicia-se após o término da Fase de Projeto e os artefatos produzidos são: Diagrama de Componentes e Comentários em Código Fonte.

Os Diagramas de Componentes são utilizados neste processo com o intuito de representar a relação entre os arquivos *client script*, *server script* e *class* que um caso de uso implementa e utiliza. As dependências representam a utilização de um arquivo pelo outro. O estereótipo <<implements>> indica que um determinado arquivo implementa as classes ligadas a ele. Quando, na modelagem de um sistema, o Diagrama de Componentes tiver muitos arquivos, é aconselhável utilizar vários Diagramas de Componentes parciais ou pacotes (por exemplo, um diagrama de componentes por pacote de classes). Este diagrama também contempla a navegação entre os arquivos de sistema, dando continuidade à

sequência explicitada no Mapa de Navegação e Diagrama de Sequência, ou seja, a relação entre os arquivos *html*, *css* e *php*.

Fase de implementação

No desenvolvimento de um software Web, ainda que o escopo seja pequeno, a quantidade de arquivos produzida é relativamente grande. A documentação do inter-relacionamento destes arquivos é um ponto forte proposto pelos artefatos produzidos nesta fase. A Figura 10 ilustra um Diagrama de Componentes para o caso de uso Cadastrar Produto, em que são apresentados os arquivos de conexão com o banco de dados (Conexao.php e CBanco.php), arquivos para efetivação do cadastro (CadastrodeProduto.php, GravaProduto.php, ExcluiProduto.php e ConProduto.php) e o arquivo que implementa as classes (Cproduto.php). O arquivo Conexao.php realiza a conexão com o banco de dados e o arquivo CBanco.php é responsável por implementar a classe Banco que realiza as operações de SQL no banco de dados. A representação proposta não apresenta o mesmo conceito de componentes da UML.

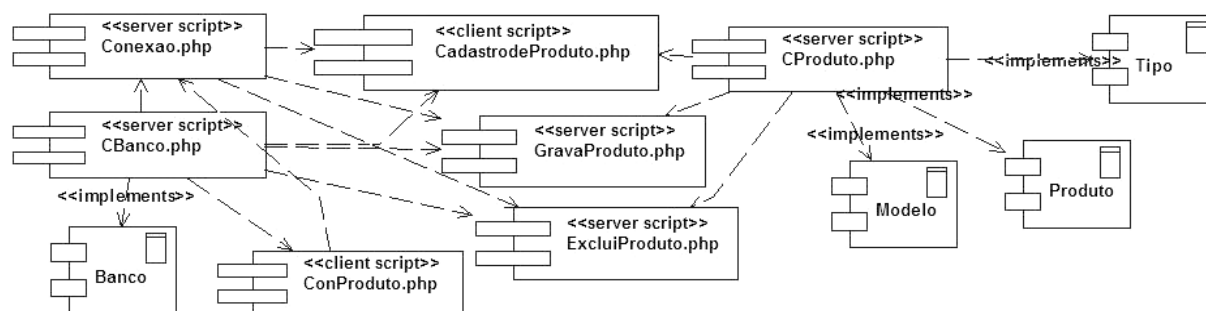


Figura 10. Diagrama de componentes para o caso de uso cadastrar produto.

Os Comentários em Código Fonte são recomendados pelo processo WAEI/MSE, pois são considerados muito importantes para entendimento posterior, objetivando a evolução do sistema, implementação de mudanças e manutenções. É necessário comentar classes, métodos, atributos, laços e funções que sejam de difícil compreensão, ou cujo objetivo não seja claramente entendido.

Fase de teste

O processo WAEI/MSE recomenda que atividades de verificação, validação e teste (VV&T) sejam realizadas durante todo o ciclo de vida do software. Especificamente, foi introduzida uma Fase de Teste na qual devem ser conduzidas estas atividades.

Colanzi (1999) apud Maldonado et al. (2007) e Vincenzi et al. (2003) afirmam que as atividades de teste de software devem acontecer em níveis diferentes, verificando primeiramente as unidades do software, posteriormente a integração do software e, finalmente, o teste de sistema que direciona os esforços de teste para quando o software já está totalmente integrado, identificando erros e características que estejam contra a especificação do sistema. Recomenda-se, adicionalmente, a aplicação do teste de aceitação conduzido com o usuário final.

No processo WAEI/MSE, a Fase de Teste deve iniciar com a definição do plano de teste a ser aplicado, no qual devem estar mencionados os casos de teste a serem realizados, os objetivos a serem alcançados, como e quando realizá-los. Após a realização de cada caso de teste, deve ser desenvolvido um relatório do teste realizado. O processo WAEI/MSE, porém, não enfatiza quais os tipos de atividades de VV&T a serem realizados.

Breve comparação entre os processos WAE e WAEI/MSE

Minimizar a documentação criada, em um processo de desenvolvimento de software, é o principal motivo da criação do WAEI/MSE, considerando que empresas enquadradas no nível 1 do CMMI possuem seu processo de software imaturo. Sendo assim, o WAEI/MSE sugere um processo de desenvolvimento de software reduzido que pode ser melhorado e adaptado, tornando-se cada vez mais maduro dentro das empresas que o utilizam.

Nesse sentido, a instância não propõe o desenvolvimento do documento de arquitetura, visto que as MPEs, geralmente, desenvolvem software Web usando o padrão arquitetônico Cliente Web Magro, em que todas as regras de negócio são realizadas no servidor (CONALLEN, 2002).

Contudo, para sistemas mais complexos, as equipes de desenvolvimento podem optar por incluir o documento de arquitetura no processo WAEI/MSE. A Tabela 1 apresenta a comparação entre os artefatos produzidos pelo WAE e pela sua instância.

Tabela 1. Comparação dos artefatos produzidos em WAE e WAEI/MSE.

	WAE	WAEI/MSE	
Conjuntos	Artefatos	Artefatos	Fases
Gerenciamento de Projeto	Plano de Gerenciamento de Alteração Plano de Projeto Plano de Iteração	Escopo Doc. de Requisitos Diag. de Casos de Uso	Fase de Requisitos
Conjunto de Domínio	Glossário	Descrição Alto	
Conjunto de Requisitos	Modelo de Domínio Visão Diretrizes de Experiência do Usuário Casos de Uso e Especificações Diagrama de Sequência Diagrama de Atividades	Nível dos Casos de Uso	
Conjunto de Análise	Doc. de Arquitetura de Software Protótipo de Arquitetura Diagrama de Sequência Classe de Análise Diagrama de Estado Roteiro Diagrama de Colaboração	Descrição Expandida dos Casos de Uso Diag. de Classes de Análise Layout do Sistema Mapa de Navegação	Fase de Análise
Conjunto de Projeto	Classe de Projeto Página Web	Diagrama de Sequência Diag. de Classes de Projeto Layout Final do Sistema	Fase de Projeto
Conjunto de Implementação	Código-Fonte Componente Binário Página Web com Script	Diag. de Componentes Comentários em Código Fonte	Fase de Implementação
Conjunto de Teste	Plano e Procedimento de Teste Script de Teste Resultados de Teste	Plano de Teste Relatório dos Testes	Fase de Teste
Conjunto de Implantação	Plano de Implantação Componente Binário Arquivos de Implantação		Não-existente

Resultados e discussão

Como resultado deste trabalho de pesquisa, foi desenvolvida uma instância do processo original proposto por Conallen (2002), anteriormente descrita e exemplificada. Esta seção apresenta algumas validações do processo WAEI/MSE e seus resultados.

Inicialmente, a validação da instância foi realizada documentando e desenvolvendo em PHP, o módulo de vendas e de contas a receber para o ConfSystem. Nesta primeira aplicação, o WAEI/MSE mostrou-se viável no contexto de MPEs. A instância foi validada também, junto a uma equipe de analistas e desenvolvedores da Coordenadoria de Web da Unoeste. Por meio da apresentação dos artefatos produzidos e do módulo desenvolvido, a equipe

consultada a considerou adequada, pois os artefatos gerados pelo WAEI/MSE permitiram uma visão em comum para os analistas e desenvolvedores do sistema e, conseqüentemente, o tempo utilizado para aplicar o processo não foi considerado perdido. Atualmente, a instância vem sendo utilizada no currículo do Curso Superior de Tecnologia em Desenvolvimento Web da mesma universidade com o intuito de validá-la no contexto acadêmico.

Uma outra validação realizada consistiu de estudos de caso utilizando o processo em duas microempresas de desenvolvimento de software Web situadas em Maringá, Estado do Paraná.

A primeira empresa existe há dois anos e não utilizava um processo sistemático de desenvolvimento de software. Como ela possui mais de 30 clientes, percebeu a necessidade emergencial de adotar algum processo de desenvolvimento de software Web. O processo foi aplicado durante o desenvolvimento de um sistema de gerenciamento de projetos de uma ONG denominada Escrever o Futuro. Os desenvolvedores da empresa destacaram o fato de o software estar sendo desenvolvido em PHP sem usar princípios de orientação a objetos (OO). Neste contexto, usar artefatos como o diagrama de classes não permite o mapeamento direto para banco de dados relacional.

A segunda empresa está no mercado há nove anos e atende cerca de 50 clientes. Insatisfeita com o processo de desenvolvimento de software adotado, ela resolveu adotar o processo WAEI/MSE em um novo projeto, que consiste no desenvolvimento de uma locadora virtual, utilizando a abordagem orientada a objetos em PHP. Os fatores que a motivaram a utilizar a instância do WAE foi a existência do mapa de navegação e do diagrama de componentes por contemplarem conteúdos intrínsecos a software Web, como navegação no *website* e relacionamento entre os arquivos de implementação. Além disso, esses artefatos também foram elogiados pela primeira empresa.

Os principais problemas encontrados que enfatizaram pontos fracos na aplicação do processo WAEI/MSE, de acordo com as empresas, foram o pequeno número de colaboradores para o desenvolvimento dos projetos e a falta de conhecimentos em engenharia de software, principalmente, em produzir diagramas UML. Outro problema percebido foi que, no momento da implementação dos projetos, as empresas não realizaram, como proposto pelo processo, a separação das camadas de aplicação e de regras de negócio em diferentes *scripts* (*client script* e *server script*,

respectivamente). Os desenvolvedores afirmaram que, por falta de tempo, não puderam mudar a forma de desenvolvimento porque essa mudança acarretaria atraso no projeto, sendo assim, não realizaram a separação das camadas.

Em geral, os pontos fortes destacados pelas empresas foram que o processo lhes proporcionou: melhor controle do projeto, maior facilidade de entendimento e melhor documentação do produto final. As empresas afirmaram que continuarão a utilizar o processo WAEI/MSE para projetos de grande porte, porém, para projetos menores, a opinião é de que o processo apresentado é relativamente grande e que acarreta atrasos. Portanto, elas preferem correr o risco de ter problemas futuros por falta de documentação no momento da manutenção do que adotar um processo de software sistemático em todos os projetos.

Conclusão

O objetivo do trabalho abordado neste artigo foi desenvolver uma proposta de processo de desenvolvimento de software Web para MPes, a fim de torná-las mais eficientes e competitivas. No decorrer da pesquisa, observou-se que, geralmente, os processos tradicionais são considerados extensos por este grupo de organizações. Portanto, decidiu-se por desenvolver uma instância do processo WAE, mantendo-se algumas características do processo original, porém deixando-o mais objetivo e conciso. Para otimizar o desenvolvimento de software Web, mantendo a qualidade do processo e do produto final, foram introduzidos no WAEI/MSE somente os artefatos com maior prioridade para o desenvolvimento do software (ARISHOLM et al., 2006).

Ao validar o processo proposto, detectou-se que atingiu os objetivos requeridos quando utilizado para documentar/modelar sistemas em que as equipes possuem previamente bom entendimento, pelas experiências já vivenciadas, do domínio do negócio a ser considerado. No entanto, as informações levantadas, na realização dos estudos de caso, serão utilizadas para melhorar características que não foram contempladas anteriormente.

O WAEI/MSE procura ser enxuto, porém flexível para agregar melhorias. Entretanto, trabalhos futuros podem contribuir para a sua melhoria, como, por exemplo, o detalhamento das atividades de VV&T a serem realizadas, a validação quantitativa do processo WAEI/MSE e o estudo de formas de relação do processo WAEI/MSE com UP, XP e MPS.BR.

Conclui-se que o WAEI/MSE atingiu seus objetivos quanto ao melhoramento dos processos de organizações classificadas no nível 1 do CMMI. As

organizações que utilizaram o processo destacaram que, dentre as principais colaborações do WAEI/MSE para a melhoria do processo de desenvolvimento de software da organização, estão: o melhor controle das atividades e do cronograma do projeto e a maior facilidade de entendimento do processo e dos artefatos produzidos, o que influenciou positivamente no momento de realizar as manutenções. Observa-se, ainda, que os pontos fortes apresentados na seção anterior sobressaem aos pontos fracos, de acordo com as organizações envolvidas; quanto aos pontos fracos, pesquisas estão sendo realizadas para a melhoria das características apontadas por estas organizações. As organizações destacaram que continuarão a utilizar o WAEI/MSE para o desenvolvimento de seus projetos.

Finalmente, esta pesquisa contribuiu para o desenvolvimento tecnológico científico e para que novas pesquisas sejam realizadas, tendo como base os resultados obtidos.

Referências

- ARISHOLM, E.; BRIAND, L. C.; HOVE, S. E. The impact of UML documentation on software maintenance: an experimental evaluation. **IEEE Transactions on Software Engineering**, v. 32, n. 6, p. 365-381, 2006.
- BECK, K. Extreme programming: a humanistic discipline of software development. **Fundamental Approaches to Software Engineering**, v. 1382, p. 1-6, 1998.
- CONALLEN, J. **Building Web Applications with UML**. 2. ed. [s/l]: Addison-Wesley, 2002.
- HABRA, N.; ALEXANDRE, S.; DESHARNAIS, J. M.; LAPORTE, C. Y.; RENAULT, A. Initiating software improvement in very small enterprises – Experience with a light assessment tool. **Information and Software Technology**, v. 50, n. 7-8, p. 763-771, 2008.
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. The unified process (Reprinted from The Unified Software Development Process). **IEEE Software**, v. 16, n. 3, p. 96, 1999.
- LARMAN, C. **Utilizando UML e padrões: uma introdução a análise e projeto orientados a objetos**. Porto Alegre: Bookman, 2000.
- LARSEN, G.; CONALLEN, J. Engineering web-based systems with UML assets. **Annals of Software Engineering**, v. 13, n. 1-4, p. 203-230, 2002.
- MALDONADO, J. C.; DELAMARO, M. E.; JINO, M. **Introdução ao teste de software**. Rio de Janeiro: Campus, 2007.
- POSTON, R. M. Preventing software requirements specification errors with IEEE-830. **IEEE Software**, v. 2, n. 1, p. 83-86, 1985.
- PRESSMAN, R. S. **Engenharia de software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.
- VINCENZI, A. M. R.; MALDONADO, J. C.; DELAMARO, M. E.; SPOTO, E. S.; WONG, W. E. Component-based software: an overview of testing. In: CECHICH, A.; PIATTINI, M.; VALLECILLO, A. (Ed.). **Component-based software quality: methods and techniques**. Heidelberg: Springer, 2003. p. 99-127. (Series Lecture Notes in Computer Science, v. 2693).
- YOO, C.; YOON, J.; LEE, B. A Unified model for implementation of both ISO 9001: 2000 and CMMI by ISO-certified organizations. **Journal of Systems and Software**, v. 79, n. 7, p. 954-961, 2006.

Received on March 25, 2008.

Accepted on July 3, 2008.

License information: This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.