
Suppera Solutions

mPrime

Relatório Final do Release

Versão 1.0

<i>mPrime</i>	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

Histórico de Revisões

Data	Versão	Descrição	Autor
02/07/2005	1.0	Elaboração do Relatório do Release	Thiago Fernandes

<i>mPrime</i>	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

Conteúdo

1.	Introdução	4
2.	Escopo	4
3.	Aderência ao Plano	4
4.	Estórias do Usuário e Temas implementados	5
5.	Resultados dos Testes	6
6.	Ocorrência de Mudanças	7
7.	Pontos de Retrabalho necessário	8
8.	Referências	8

mPrime	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

Relatório Final do Release

1. Introdução

O mPrime é uma ferramenta que dará suporte ao processo de Gestão de Risco para Ambientes de Múltiplos Projetos de Desenvolvimento de Software.

Este documento tem como objetivo apresentar um relatório final do release. Nele se apresenta a relação entre o que foi planejado pela equipe de desenvolvimento e o que foi executado. Além de conter, uma avaliação da equipe com relação à adequação do processo de desenvolvimento para com o projeto. Para finalizar apresenta-se uma descrição das estórias de usuários que foram, a descrição da elaboração dos testes para cada uma delas, além de conter as mudanças ocorridas e os pontos que devem ser alterados para o próximo release.

2. Escopo

Este documento visa ao desenvolvimento do projeto durante o release em questão. Ele avalia e descreve todo este processo e apresenta as variações entre ele e o Plano de Projeto.

A metodologia de desenvolvimento escolhida pela Suppera Solutions foi Extremme Programming, com isso, ela tem por objetivo o desenvolvimento de funcionalidades de acordo com a necessidade do cliente. No início do processo documenta-se, junto a cliente, as estórias do usuário que serão executadas pelo sistema. A partir destas estórias, faz-se o planejamento do release.

Neste documento, discuti-se, inicialmente, a aderência do plano de projeto com relação ao processo de desenvolvimento. As estórias que foram implementadas e todos os possíveis temas surgidos durante esta fase são listados e as mudanças ocorridas durante o desenvolvimento são elucidadas. Avalia-se se as estimativas estipuladas foram consistentes e o quanto das atividades planejadas foram implementadas para a cliente, fazendo-se um balanço da produtividade da empresa. Esta é uma análise importante que deve ajudar para o planejamento do próximo release

Para finalizar a avaliação, os resultados obtidos com a realização dos testes unitários e teste de aceitação são apresentados. Esta é uma forma de mostrar como ocorreram os testes, como eles foram elaborados e qual o impacto deles no desenvolvimento geral do sistema.

O documento apresenta a avaliação do release, com a metodologia que realmente foi utilizada pela equipe, além de apresentar falhas e os pontos de mudanças para o próximo release, sendo de fundamental importância para a empresa.

3. Aderência ao Plano

O plano de projeto foi elaborado mediante as estimativas da equipe com relação ao conhecimento do projeto. Contudo, assim como define a metodologia de Extremme Programming, a idéia do projeto vai sendo incrementada continuamente juntamente com as decisões e necessidades do cliente.

O projeto do mPrime necessitou de um grande conhecimento teórico da área a que ele se propõe atuar. Isto porque para sua realização foram necessários um trabalho de graduação, uma tese de mestrado e uma de doutorado. Este estudo necessitou de um bom tempo para ser adquirido. Além disso, o plano de projeto foi idealizado concorrentemente com este aperfeiçoamento conceitual da equipe. Isto acarretou um passo muito importante para a estimativa da equipe para com o projeto.

O plano de projeto esteve adequado ao release, contudo, ele foi extenso o bastante para a equipe. O planejamento foi bem executado com relação à identificação das estórias do usuário, elas foram bem distribuídas com relação às iterações, o risco pode ser identificado, analisado, planejado, monitorado e controlado. Logo, ele se adequou a metodologia desenvolvida pelo SEI (Software Engineering Institute). Contudo, segundo o planejamento do release, a categorização de projetos deveria ser a primeira estória do usuário, o que foi verificado, posteriormente, pela equipe que não era eficiente. Isto consumiu algum tempo

mPrime	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

da equipe, e acabou não sendo produtivo. Isto porque, para haver a categorização do projeto era preciso entender mais quais atributos poderiam ser alterados do próprio MS Project, e este era um dado que só foi adquirido, a medida em que, se utilizava a biblioteca do mesmo. Logo, faltava conhecimento da equipe com os métodos que poderiam ser reaproveitados.

Com relação às estimativas de tempo, é importante observar o pouco tempo que a equipe teve. Isto porque ela é pequena e teve que trabalhar com várias novas tecnologias que não era de seu conhecimento. Contudo, é possível observar que houve esforço suficiente entre os membros para a elaboração de todas as fases da gestão de risco definida pelo SEI.

É possível concluir que, exceto pelas mudanças de cronograma causadas pela escassez de tempo mediante a quantidade de atividades, o plano de release foi adequado para o processo de desenvolvimento. Esta adequação diz respeito principalmente as histórias que foram implementadas, a sequência de implementação das mesmas.

O quão aderente ao planejado foi o release. As estimativas foram corretas? Os resultados esperados foram alcançados?

4. Estórias do Usuário e Temas implementados

As estórias do usuário foram elucidadas a partir de uma reunião feita inicialmente com a cliente. A partir da descrição de todas as estórias que deveriam ser elaboradas durante o release, aconteceu a reunião de planejamento de release. E a partir deste documento, foi feita a demarcação semanal das reuniões com a cliente para a descrição específica de cada uma das estórias que seriam elaboradas durante a iteração em questão. Estas reuniões aconteceram sempre às quartas-feiras e nela estavam presentes, toda a equipe juntamente com a cliente. Nestas ocasiões, discutiam-se, também, dúvidas que surgiram durante a iteração iniciada na semana anterior e que necessitavam da decisão da cliente como gerente de risco.

Segue abaixo as estórias que foram trabalhadas durante o release:

- I. **Identificar Riscos** - Determina que riscos são prováveis de afetar o projeto e procede a documentação das características de cada um deles.
- II. **Quantificar Riscos** - Analisa numericamente a probabilidade de ocorrência do risco e suas conseqüências nos objetivos do projeto, bem como a extensão do risco global do projeto.
- III. **Qualificar Riscos** - Avalia qual a probabilidade de ocorrência e o grau de impacto dos riscos identificados no Plano de Gestão de Risco. Prioriza o risco de acordo com o seu efeito potencial sobre os objetivos do projeto.
- IV. **Responder Riscos** - Desenvolver planos que possam sempre que possível eliminar o risco.
- V. **Controlar Riscos** - Consiste na implementação de ações corretivas, previstas no Plano de Contingência e em atualizações no Plano de Gerência de Riscos.
- VI. **Monitorar Riscos** - Desenvolver rotinas de acompanhamento da execução do projeto, de forma a espelhar a realidade e as tendências de risco ao longo do ciclo de vida do projeto.
- VII. **Planejar Gestão Riscos** - Define que atividades de gerenciamento de risco deverão ser executadas pelo projeto.

As estórias foram implementadas nesta ordem, e seguiram a seqüência definida pela cliente. Segundo Extreme Programming, as funcionalidades do sistema são implementadas e apresentadas para a cliente, em ordem de prioridade de cada uma, quanto maior for a necessidade dela, mais rápido se inicia sua codificação. Esta é uma forma de deixar a cliente bem próxima da equipe de desenvolvimento, além de priorizar as preferências dela. Além disso, esta seqüência seguiu a ordem de desenvolvimento das fases da gestão de riscos descrita pelo SEI, possibilitando, a toda a equipe, um conhecimento modular e contínuo da

<i>mPrime</i>	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

gestão de riscos.

O uso de XP contribuiu bastante com o desenvolvimento do projeto, uma vez que, a equipe necessitou aprender diversos conceitos que não eram do seu cotidiano e que cabiam a cliente. Desta forma, foi possível que a cliente estivesse sempre próxima aos desenvolvedores, diminuindo contradições entre requisitos e validando as funcionalidades implementadas constantemente. Logo, se um detalhe não ficou de acordo com o esperado pela cliente ela já avisava, com bastante antecedência, evitando perda de tempo em implementações desnecessárias.

Lista das histórias implementadas e de possíveis temas implementados no release.

5. Resultados dos Testes

A metodologia da Suppera adotou dois tipos de testes para este primeiro release. Os testes unitários, que são implementados em uma linguagem de programação e executados através de um programa e os testes de aceitação que são negociados com a cliente na reunião semanal, neste momento ela define quais são os requisitos que ela considera primordiais e como eles devem ser feitos, até mesmo em nível de interface gráfica, dependendo da história a ser implementada.

Os testes unitários foram produzidos pelos desenvolvedores do sistema para cada uma de suas classes, antes mesmo da própria implementação da história, induzindo um desenvolvimento direcionado a cobrir o maior número de casos possíveis. Isto porque, antes de codificar, o programador deverá pensar em todas as possibilidades de fluxo. Segundo pesquisa feita pela equipe, centros de uso de XP definem que o tempo gasto para implementar uma funcionalidade isoladamente era quase o mesmo de implementar os testes e depois a classe. Além disso, após os testes, as correções de código necessárias são feitas seguidas de uma nova sessão de testes, formando um ciclo entre alterações e os testes das mesmas. Isto gera um código mais confiável e otimizado, conforme a metodologia de desenvolvimento adotada (eXtreme Programming). Utilizar o teste unitário ajudou na definição do que realmente precisava ser feito, eles definiam todos as atividades que necessitavam ser testadas e, conseqüentemente, implementadas. Desta forma, evitou-se que os codificadores trabalhassem desnecessariamente, em métodos e classes que não seriam utilizadas. E finalmente, os testes contribuíram para a definição de quando a tarefa estava realmente terminada, já que ela só termina quando todos os testes estão rodando corretamente. O programa utilizado pela equipe para a execução dos testes foi o NUnit, ferramenta grátis disponibilizada na rede, e que pode ser integrado facilmente ao Visual Studio 2003, ferramenta utilizada pela equipe.

mPrime	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

Os testes de aceitação foram definidos pelo usuário na fase inicial do projeto e são os critérios de aceitação do software conforme a estratégia de entrega e representa exatamente a métrica de aderência do sistema desenvolvido ao universo do cliente. Eles garantem o correto funcionamento de cada requisito a ser desenvolvido. Testes de unidade, embora essenciais, não são suficientes. Os testes de aceitação estão diretamente ligados com as histórias do usuário. Eles visam testar o sistema do ponto de vista do usuário, de modo que são menos suscetíveis a alterações. Como o sistema é testado com todos os componentes interligados e configurados, inclusive bancos de dados e gerenciadores de filas, há garantias de que cada serviço oferecido esteja funcionando. Além disso, ao escrever os testes, é preciso ser bastante rigoroso e tentar prever todas as ações corretas e incorretas do usuário. Ao fazer isso, muitas vezes nos deparamos com situações que não haviam sido previstas pela análise inicial feita pela equipe. Nestes casos, é possível refinar esta análise inicial e trazer novos elementos que a equipe de desenvolvimento deve passar a levar em consideração. Em resumo, os testes de aceitação, além de serem uma importante ferramenta para garantir a corretude do sistema, é também uma importante ferramenta para refinar a análise e garantir que a equipe esteja sempre trabalhando no que é importante para a cliente.

Referenciar os resultados obtidos em relação aos testes unitários e testes de aceitação

6. Ocorrência de Mudanças

Assim como define a metodologia XP, o desenvolvimento do projeto é orientado a mudanças. Elas acontecem naturalmente durante todo o processo de implementação, até porque ele é feito incrementalmente. Logo, várias mudanças ocorreram, principalmente com relação a forma como as histórias do usuário seriam implementadas. Isto porque, segundo as histórias que foram implementadas e citadas acima, elas seguem uma descrição das fases da gestão de risco segundo o SEI. Contudo, a forma como estas histórias seriam apresentadas ao usuário foi sendo alterada gradualmente e de acordo com as exigências da cliente.

Com relação ao planejamento da equipe na implementação das histórias, houve duas grandes mudanças no planejamento do Release. Uma delas foi a ordem da implementação da história de categorização de projeto. Esta história definia como projetos deveriam ser categorizados para que através desta técnica compará-los. Ela foi planejada para ser uma das primeiras a ser executada, contudo a equipe percebeu que ela estava intimamente relacionada com a ação do agente inteligente, o que dificultava sua implementação, já que o agente só seria implementado mais tarde. A mudança ocorrida foi transferir a implementação desta funcionalidade para ocorrer concorrentemente com a do agente inteligente. Logo, a equipe responsável pela implementação do agente ficou responsável pela validação da categorização de projeto, definindo quais atributos seriam comparados e qual o critério de pesos para cada um deles. Isto ocorreu na última reunião com os clientes e ambos, Cristine Gusmão e Hermano Perrelli, confirmaram a não-trivialidade desta atividade, uma vez que os pesos podem variar para cada empresa. Logo, decidiu-se que esta história seria adiada, para que uma pesquisa mais concreta fosse feita com um conjunto de gerentes de projeto, a fim de modular um sistema de pesos default para esta categorização. Mesmo assim, ainda foi decidido que o sistema daria liberdade para o cliente redefinir estes pesos.

Outra alteração que ocorreu com o planejamento do release foi o adiamento da implementação do agente inteligente. Uma vez que, diferentemente do que foi informado a equipe, a ontologia não estava pronta. Ela havia sido definida conceitualmente, e quanto a sua implementação só havia uma parte pronta e mesmo assim, esta se encontrava incompleta. Por causa deste contra-tempo, foi preciso alocar recursos exclusivos para a implementação da ontologia. Todavia, embora a implementação do agente inteligente tenha sido adiada, a sua concepção teórica foi toda definida e uma estratégia de desenvolvimento foi criada. Esta foi uma forma de a equipe avaliar realmente todas as estâncias necessárias para a implementação do agente, evitando que novas estimativas errôneas fossem feitas por conta de dados não consistentes. Além disso, a ontologia foi definida para todas as categorias de riscos sugeridas pelo SEI.

mPrime	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

Citar mudança de requisitos, aparecimentos de concorrentes, mudança na infra-estrutura.

7. Pontos de Retrabalho necessário

As áreas de retrabalho do sistema estão intimamente relacionadas com as mudanças que foram realizadas durante este release. O principal ponto é a implementação do agente inteligente. Uma vez que existe a ontologia, a ontoPRIME, é preciso que se desenvolva a base de caso e a base de regras para o sistema. Como a estratégia de desenvolvimento do agente foi feita, este será um dos primeiros passos para o próximo release. Para isto, é preciso que toda a equipe aprenda melhor os conceitos teóricos de inteligência artificial, e para isto, uma palestra será ministrada pelo gerente de inteligência artificial, com o intuito de explicar tais conceitos e o plano de desenvolvimento. Toda a equipe deve trabalhar com esta implementação, diferentemente do primeiro release, em que uma parte da equipe foi toda alocada para a solução da ontologia. Esta é uma forma, de validar ainda mais a metodologia XP e difundir entre todos os integrantes o conhecimento sobre o agente inteligente, tendo a intenção de agilizar o seu desenvolvimento

Outro ponto que foi discutido pela equipe e que será executado no próximo release é a estrutura de implementação da equipe. As reuniões deverão acontecer um pouco antes e durante a semana haverá um outro encontro com a cliente para validação das tarefas. Isto porque neste release a equipe só possuía uma reunião com a cliente e quando havia alguma dúvida que só ela poderia decidir, ficava um pouco complicado entrar em contato com ela. Logo, com duas reuniões é possível validar os requisitos das estórias do usuário que serão implementadas na iteração em uma delas e validar a execução das tarefas e do planejamento da equipe em outra, evitando desta forma que a equipe trabalhe desnecessariamente. Além disso, haverá duas reuniões semanais de trabalho da equipe, já que na primeira só havia uma. Estas duas reuniões servem para dividir o trabalho, em uma delas será feita a implementação do sistema, já que muitos integrantes não trabalham apenas com a Suppera, neste momento cada um deve desenvolver as tarefas que ficou responsável. Esta técnica evita que alguém fique sem implementar porque seu trabalho depende de outra pessoa, pois todos os integrantes estarão no mesmo ambiente, qualquer dúvida surgida poderá ser discutida imediatamente. Além do mais, como o mPrime lida com conceitos que não são cotidianos da equipe, o mais difícil é começar a implementação, logo, desta forma, a equipe estará toda reunida para o início da implementação, tentando diminuir estes problemas iniciais. Contudo, os integrantes podem ficar responsáveis por algumas atividades que serão feitas fora do horário de implementação. Já a segunda reunião, que ocorrerá durante a semana servirá para resolver qualquer problema que houver com a integração do sistema, pois em XP deve haver implementação contínua e neste momento resolve-se qualquer problema ocorrido e se finaliza a iteração, gerando um instalador para o uso da cliente. Uma nova metodologia de trabalho já foi criada pelo gerente do projeto e o gerente de desenvolvimento e discutida com toda a equipe.

Com relação aos recursos humanos da empresa, estes também foram reestruturados. A estrutura de trabalho não funcionou tão bem quanto deveria. Os cargos criados para a equipe seguiram a necessidade que o projeto demandava, nem sempre, se preocupando com os cargos sugeridos por XP e que podem ser de extrema valia para o projeto. Uma nova estrutura também foi definida pelos gerentes e discutida com a equipe. Esta será uma forma de deixar a organização hierárquica da empresa mais próxima aos conceitos de XP, além de validá-las ou não para a dinâmica da empresa.

Citar problemas encontrados em algumas áreas que causam a necessidade de retrabalho nessas áreas em outros releases

8. Referências

Livros:

- Sommerville, Ian. Software Engineering, 5th ed., Addison-Wesley, 1995.
- Program Development in Java. Liskov, Barbara & Guttag, John. 2001.

<i>mPrime</i>	Versão: 1.0
Relatório Final do Release	Data da Versão: 02/05/2005
Suppera_RelatorioRelease	

- Addison Wesley – Extreme Programming Explained
- Addison-Wesley - Martin Fowler – Refactoring -Improving the Design of Existing Code
- Wesley - Extreme Programming Applied - Playing to Win

Trabalhos Científicos:

- Cynthia Belleza Bernardino. Design interativo no desenvolvimento de software. 2005. 80 f. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Universidade Federal de Pernambuco. Orientador: Alex Sandro Gomes.
- Bruno Celso Cunha de Freitas. Um modelo para o gerenciamento de múltiplos projetos de software aderente ao CMMI. 2005. 93f. Trabalho de Conclusão do Curso. (Graduação em Ciência da Computação) – Universidade Federal de Pernambuco. Orientador: Hermano Perreli de Moura.
- Ciro Carneiro Coelho. MAPS: Um Modelo de Adaptação de Processos de Software. 2003. 178f. Dissertação para obtenção do grau de mestre. (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco. Orientador: Hermano Perreli de Moura.
- Gerências de Riscos em Engenharia de Software. Christine Martins Gomes de Gusmão. 2005. Exame de Qualificação. (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco. Orientador: Hermano Perreli de Moura.

Sites:

- http://www.cin.ufpe.br/~if683/referencias/slides/introducao_xp.pdf;
- http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_pmbok_p.html;
- <http://www.xispe.com.br/index.html>;
- <http://labase.nce.ufrj.br/blog/leonardo/2005/06/28/1119982096000.html>;
- <http://xp.edugraf.ufsc.br/>;
- http://www.pmir.org/PMI21_Documentos.htm.
- <http://www.NUnit.org>
- <http://msdn.microsoft.com/library>