

Uma Visão Ágil sobre o Processo de Desenvolvimento de Software do MPDFT

Marcelo Henrique de Oliveira Lima

2014

Resumo

Palavras-chaves: Processo Unificado. MPDFT. Métodos Ágeis.

Introdução

1 Processo Unificado

Um processo de software é um conjunto de atividades que leva à produção de um produto de software ([SOMMERVILLE, 2007](#)). O Processo Unificado (PU) para desenvolvimento de software surgiu como uma alternativa ao já reconhecidamente ineficiente modelo sequencial ou em cascata. O modelo em cascata foi o primeiro a ser publicado e teve forte influência dos processos mais gerais de engenharia de sistema ([SOMMERVILLE, 2007](#)). Esse modelo (cascata) caracteriza-se pelo encadeamento das fases e pelo fato da fase seguinte não iniciar antes da atual ter terminado. Alguns estudos de sucesso/falha mostram que projetos que utilizaram o modelo em cascata obtiveram alta taxa de falha. Acredita-se que esse modelo ganhou forte adoção baseado em boatos e crenças ao longo dos anos ([LARMAN, 2007](#)).

Já o Processo Unificado (PU) ([JACOBSON; BOOCH; RUMBAUGH, 1999](#)) é fortemente baseado em um modelo iterativo e incremental. Esse modelo se mostrou mais eficiente em situações que os requisitos não estão completamente definidos e a entrega de resultados deve ser antecipada o máximo possível, além de reduzir a quantidade de defeitos. [Larman \(2007\)](#) cita em seu livro que o desenvolvimento iterativo e evolutivo considera normal que a fase de desenvolvimento comece antes dos requisitos terem sido definidos em detalhes; a realimentação é usada para esclarecer e aperfeiçoar as especificações em evolução.

Colocar referências do Larman.

Talvez gráfico das baleias.

O PU pode ser dividido em três visões: dinâmica, estática e melhores práticas???

Disciplinas de Engenharia

1. Modelagem de Negócios
2. Requisitos
3. Análise e Projeto (Design)
4. Implementação
5. Teste
6. Implantação

Disciplinas de Apoio/Suporte

1. Ambiente
2. Gestão de Configuração e Mudança
3. Gerência de Projeto

Falar sobre visões dinâmicas, estáticas e

Falar sobre disciplinas, workflows, itens de trabalho etc.

1.1 MPDFT-UP

Seguindo recomendações do Tribunal de Conta da União (TCU) presentes no acórdão nº 1.603/2008 ([TCU, 2008a](#)) e publicadas no Diário Oficial da União ([TCU, 2008b](#)), que visam melhorias na governança de tecnologia da informação na administração pública federal. No item 9.1.4 desse acórdão é proposta a utilização de medidas que estimulem a adoção de metodologias de desenvolvimento de sistemas. Foi feito então um esforço para adequar-se às recomendações. O Ministério Público do Distrito Federal de Territórios (MPDFT) desenvolveu um processo denominado MPDFT-UP ([MPDFT, 2014](#)). É um processo iterativo e incremental baseado no processo unificado. A utilização do PU não é tão crítica em relação a necessidades e pressões de mercado. Mas, por se tratar de um órgão público, é necessária especial atenção à documentação dos sistemas desenvolvidos, formalização de demandas, designação de responsáveis pelo desenvolvimento e pelos aceites.

Seguindo o entendimento de que não existe um processo ideal várias organizações desenvolveram abordagens inteiramente diferentes para o desenvolvimento de software ([SOMMERVILLE, 2007](#)).

Os processos evoluíram para explorar as capacidades das pessoas em uma organização e as características específicas dos sistemas que estão sendo desenvolvidos ([SOMMERVILLE, 2007](#)).

Falar sobre as visões implementadas pelo MPDFT-UP. Disciplinas, workflows etc.

Etapas que compõem a metodologia de desenvolvimento de sistemas

1.	Entendimento inicial	Breve descrição
2.	Planejamento e gerência de projeto	
3.	Levantamento e análise de requisitos	
4.	Implementação e manutenção	
5.	Testes	
6.	Homologação	
7.	Treinamento e implantação	
8.	Avaliação	

1. Entendimento inicial
2. Planejamento e gerência de projeto
3. Levantamento e análise de requisitos
4. Implementação e manutenção
5. Testes
6. Homologação
7. Treinamento e implantação
8. Avaliação

2 Métodos Ágeis

Por volta da década de 1990 houve um movimento envolvendo desenvolvedores que não concordavam com o caminho que estava sendo tomado pelos processos de software. A burocracia, o peso do processo em detrimento do próprio produto a ser desenvolvido, o foco nos artefatos que não necessariamente agregam valor, todos esses fatores foram motivos para a migração que leva a outro extremo, a redução brusca da burocracia. Tal processo foi tão extremo que levou a criação de um dos processos de desenvolvimento mais populares do mundo ágil, a programação extrema (eXtreme Programming - XP) ([BECK, 1999](#)).

Conforme citado por [Larman \(2007\)](#), o PU é completamente adaptável e aberto a boas práticas de outros métodos. A introdução do PU não visa

diminuir o valor desses outros métodos - muito pelo contrário ([LARMAN, 2007](#)).

<http://manifestoagil.com.br/> ([BECK, 2014](#))

2.1 Scrum

Scrum das trincheiras.

2.2 XP

2.3 Kanban

2.4 Abordagens Ágeis no MPDFT

Falta de comprometimento do Product Owner. Em geral, os POs não estão interessados ou não têm tempo para lidar com as responsabilidades do papel. As dificuldades com a estrutura organizacional. Como o modelo de gestão é mais voltado para uma organização matricial, há dificuldades no gerenciamento de projetos utilizando técnicas ágeis.

3 Modelo Híbrido

Limites bem definidos entre os modelos de processos é algo muito bom do ponto de vista didático, mas, no mundo prático, é inviável se utilizar somente de um modelo de desenvolvimento de software. Conforme explicado anteriormente, os modelos possuem vantagens e desvantagens, o que os tornam interessantes sob determinadas situações. Contudo, instituições sólidas buscam uma padronização no modo de trabalho, pois um modo de trabalho uniforme viabiliza o aprimoramento dos processos de software, no qual a diversidade de processos de software ao longo da organização é reduzida ([SOMMERVILLE, 2007](#)). Isso visa melhorias na comunicação e redução na curva de aprendizado de novos colaboradores.

Uma vez que o Processo Unificado é bastante flexível e aberto é incentivada a inclusão de práticas interessantes de outros métodos iterativos tais como: eXtreming Programming (XP) ([BECK, 1999](#)), Scrum ([SCHWABER; BEEDLE, 2002](#))

4 Recomendações

Apoio à projetização do órgão.

Melhorias na gestão de configuração: políticas de controle de versão e estilo de código.

Melhorias no fluxo do processo: liberar casos de uso para desenvolvimento e teste. Posteriormente, liberar o roteiro de teste para o desenvolvimento.

Abordagem mais agilista.

Incentivo ao desenvolvimento orientado a teste.

Definições de tecnologias e arquitetura.

Melhorias no processo de arquitetura. Incluir prova de conceito no fluxo dos projetos.

Considerações finais

Adotar de maneira mais direta as outras disciplinas do PU.

A gestão de configuração e mudança é fundamental para a padronização do processo e redução na curva de aprendizado.

Padronização dos repositórios de projetos. Ex.: gerencia, requisitos, projeto, implementação, teste, design etc.

Criação de uma política de gestão de configuração.

Encoding padrão.

Ajustes/melhorias na estrutura organizacional para facilitar a adoção de métodos ágeis. Equipes mais projetizadas.

Definição de tecnologias suportadas.

Definições de arquitetura e de uma possível arquitetura de referência.

Referências

BECK, K. Embracing change with extreme programming. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 32, n. 10, p. 70–77, out. 1999. ISSN 0018-9162. Disponível em: <<http://dx.doi.org/10.1109/2.796139>>. Citado 2 vezes nas páginas 3 e 4.

BECK, K. *Manifesto para o desenvolvimento ágil de software*. 2014. Manifesto para o desenvolvimento ágil de software. Disponível em: <<http://manifestoagil.com.br/>>. Acesso em: 20 mar. 2014. Citado na página 4.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *The Unified Software Development Process*. Pearson Education, Limited, 1999. (Addison-Wesley Object Technology). ISBN 9780321822000. Disponível em: <<http://books.google.com.br/books?id=t4UVnwEACAAJ>>. Citado na página 1.

LARMAN, C. *Utilizando UML e Padrões*. 3. ed. São Paulo: Bookman, 2007. ISBN 9788560031528. Disponível em: <<http://books.google.com.br/books?id=ZHtcynS03DIC>>. Citado 3 vezes nas páginas 1, 3 e 4.

MPDFT, M. P. do Distrito Federal e T. *MPDFT-UP*. 2014. Processo de Desenvolvimento de Sistemas do MPDFT. Disponível em: <<https://intranet.mpdft.mp.br/mpdft-up/>>. Acesso em: 03 mar. 2014. Citado na página 2.

SCHWABER, K.; BEEDLE, M. *Agile software development with scrum*. Prentice Hall, 2002. (Series in agile software development). ISBN 9780130676344. Disponível em: <<http://books.google.com.br/books?id=BpFYAAAAYAAJ>>. Citado na página 4.

SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007. Citado 4 vezes nas páginas 1, 2, 3 e 4.

TCU, T. de Contas da U. 2008. Acórdão nº 1.603/2008. Disponível em: <<http://www.tcu.gov.br/Consultas/Juris/Docs/judoc%5CAcord%5C20080814%5C008-380-2007-1-GP.doc>>. Acesso em: 23 mar. 2014. Citado na página 2.

TCU, T. de Contas da U. 2008. 243 p. Acórdão nº 1.603/2008. Disponível em: <<http://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?jornal=1&pagina=243&data=18/08/2008>>. Acesso em: 23 mar. 2014. Citado na página 2.