

UFCG – UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CCT - CENTRO DE CIÊNCIAS E TECNOLOGIA

DSC - DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

MESTRADO EM INFORMÁTICA

DISCIPLINA: GERÊNCIA DE PROCESSOS

PERÍODO: 2004.1 / 2º Bimestre

PROFESSOR: Francilene P. Garcia

ALUNO: Afrânio M. L. de Assis / Ecivaldo de S. Matos
{afranio,ecivaldo}@dsc.ufcg.edu.br

SCRUM - Metodologia para desenvolvimento de SI, em ambientes turbulentos

Campina Grande – Agosto de 2004

Sumário

1 – APRESENTAÇÃO	3
2 – INTRODUÇÃO	4
2.1 – AMBIENTES TURBULENTOS.....	4
2.2 – GERÊNCIA DE DESENVOLVIMENTO DE SOFTWARE EM AMBIENTES TURBULENTOS	4
3 – SCRUM	7
3.1 – VISÃO GERAL	7
3.2 – TERMINOLOGIA.....	8
3.3 – METODOLOGIA	8
3.3.1 – Planejamento	8
3.3.2 – Projeto Arquitetural.....	9
3.3.3 – Desenvolvimento (Sprint).....	9
3.3.4 – Encerramento.....	11
3.3.5 – Características	12
3.4 – RECURSOS HUMANOS	13
3.5 – VANTAGENS	14
3.6 – DESVANTAGENS	15
4 – XP & SCRUM	16
5 – CONCLUSÕES	17
6 – BIBLIOGRAFIA	18

1 – Apresentação

O objetivo deste documento é apresentar uma metodologia de desenvolvimento para ambientes turbulentos chamada SCRUM, metodologia extremamente focada nos aspectos gerenciais do processo de concepção de software.

Abordaremos esta metodologia de acordo com a estrutura abaixo:

- **Introdução** – Aborda um conceito de ambiente turbulento e uma metodologia de gerenciamento de projetos para esse tipo de ambiente, o Scrum.
- **Scrum** – Trata mais detalhadamente a metodologia Scrum, suas exigências, carências, vantagens e desvantagens
- **XP & Scrum** – Apresenta uma sugestão de utilização conjunta entre as metodologias *Extreme Programming* (XP) e Scrum.
- **Conclusão** – Apresenta as conclusões relacionadas ao uso do Scrum em ambientes turbulentos.
- **Bibliografia** – Expõe os itens bibliográficos utilizados durante a pesquisa.

2 – Introdução

2.1 – Ambientes Turbulentos

Na literatura temos visto modelos, metodologias e processos de desenvolvimento de software bem definidos, indicando procedimentos, técnicas e ferramentas a serem usados pelas equipes que desenvolvem software. Nesses processos são definidos, inicialmente, estimativas de prazos e recursos (humanos, financeiros e tecnológicos).

Entretanto, o que se observa na prática é que nem sempre a natureza do projeto permite que tais estimativas sejam estabelecidas inicialmente. Esses projetos se caracterizam por possuírem requisitos com alto grau de incertezas e conflitos de interesses e necessidades. Os ambientes de desenvolvimento desses projetos são conhecidos como Turbulentos.

Portanto, podemos caracterizar ambientes turbulentos por apresentar:

- Constantes mudanças de requisitos
- Possíveis mudanças dos integrantes da equipe de desenvolvimento do produto
- Pressão por parte dos clientes, que exigem resultados rápidos
- Imprevisibilidade que necessita de mecanismos de controle.

2.2 – Gerência de desenvolvimento de software em ambientes turbulentos

Gerenciar o desenvolvimento de software em ambientes turbulentos tende a ser uma atividade complexa, devido às características de tais ambientes. Portanto, faz-se necessário um processo de gerenciamento que possua mecanismos de controle flexíveis e que possa complementar outras práticas de engenharia.

Uma opção para o gerenciamento de desenvolvimento de software em ambientes turbulentos é o SCRUM, que é um aprimoramento da abordagem iterativa e incremental de desenvolvimento de software orientado a objetos.

O SCRUM assume a pré-existência de projeto e código, o que acontece na maioria dos desenvolvimentos de software orientado a objetos. Ele pode ser usado para a construção de novos sistemas ou manutenção de sistemas já existentes.

Na gerência de processos, o SCRUM considera os seguintes fatores durante a disponibilização de uma versão:

- Requisitos do cliente – as necessidades do cliente que devem ser atendidas pelo software
- Pressão de tempo – cumprimento de prazos rigorosos para atender às necessidades impostas pelo mercado
- Competição – o software deve estar apto a competir com os demais produtos do mercado
- Qualidade – fatores de qualidade devem ser satisfeitos
- Visão – metas pré-definidas devem ser alcançadas
- Recursos – infra-estrutura necessária para o desenvolvimento da versão.

Vale realçar que o Scrum não possui práticas de engenharia, ou seja, não possui aspectos voltados diretamente à construção do produto, como método para levantamento de requisitos ou testes, apenas oferece um “*framework*” para o gerenciamento do processo de desenvolvimento.

O Scrum, na sua concepção, leva em consideração algumas variáveis que são relevantes no ambiente de desenvolvimento de qualquer produto, como:

- Disponibilidade de profissionais qualificados
- Estabilidade da tecnologia de implementação
- Estabilidade e poder das ferramentas
- Efetividade dos métodos
- Disponibilidade de profissionais em várias áreas do conhecimento
- Novas requisitos ao longo do projeto
- Metodologia

Essa metodologia pode ser utilizada na construção incremental de software em ambientes turbulentos, ou seja, onde os requisitos não são claros ou mudam com muita frequência.

3 – SCRUM

3.1 – Visão Geral

O Scrum tem como característica principal o fato de ser uma metodologia ágil voltada especificamente para gerência e controle das atividades de desenvolvimento. Essa metodologia se adapta de forma bastante consistente em ambientes turbulentos e possibilita o desenvolvimento de produtos de forma iterativa e incremental. Uma iteração dura de uma a quatro semanas. Ao fim de um conjunto de iterações, obtemos um incremento do software, ou seja, a implementação de um conjunto de requisitos escolhidos anteriormente de acordo com as necessidades do cliente.

Podemos ter no Scrum mais de uma equipe de desenvolvimento. Cada equipe é composta por pequenos times, entre três e seis pessoas. O número de integrantes da equipe de gerência varia de acordo com a complexidade e o tamanho do sistema a ser desenvolvido. O progresso das equipes é revisado freqüentemente, sempre tentando descobrir problemas o quanto antes. Toda versão executável, produzida por uma equipe, passa por uma rigorosa revisão, na qual o cliente também está presente. As revisões aumentam a cooperação entre os integrantes da equipe e aproximam estes dos clientes.

Uma das atividades gerenciais que fazem parte da metodologia Scrum é a gestão de riscos. Riscos que afetam o sucesso do projeto são continuamente identificados e avaliados em termos de impacto. Dois planos devem ser criados, um para evitar tais riscos e outro de contingência, este último será usado caso algum risco venha a se concretizar.

O fato do Scrum ser voltado para gerência e controle do desenvolvimento de software nos possibilita utilizá-lo em conjunto com outras práticas da engenharia de software. Tais práticas definem detalhes da análise, projeto, implementação e testes, mas possuem deficiências em relação à gerência de atividades.

3.2 – Terminologia

Nesta seção conceituamos alguns termos importantes que serão utilizados ao longo deste documento.

Backlog – Requisitos que ainda não foram trabalhados no desenvolvimento do produto. Dentre os itens de *backlog* podemos encontrar requisitos de funcionalidades, conserto de *bugs*, melhoramentos solicitados pelo cliente, aprimoramentos que resultaram em maior competitividade do produto no mercado e atualizações de tecnologia.

Release - Grupo de itens de *backlog* que serão trabalhados num conjunto de iterações. Ao fim desse conjunto de iterações teremos uma versão estável do produto que incorpora o grupo de itens de *backlog*. A liberação dessa versão considera fatores como qualidade, necessidades do cliente e competitividade no mercado.

Pacotes – Componentes ou objetos que devem ser modificados ou criados para implementar um item de *backlog*.

3.3 – Metodologia

3.3.1 – Planejamento

É na fase de planejamento que o *backlog* é desenvolvido. Ele será constituído pelos requisitos que serão trabalhados ao decorrer de todo o processo de desenvolvimento do produto.

Uma série de outras atividades é executada nesta fase e, como o nome mesmo diz, é nesta etapa em que há um planejamento macro do que será gerado ao longo do processo. Dentre as diversas atividades, destacam-se:

- Seleção da *release* para desenvolvimento imediato
- Definição de prazos e funcionalidades
- Mapeamento dos pacotes necessários para a *release*
- Definição do time de profissionais para a *release*

- Avaliação dos riscos e estratégia de controle
- Revisão e possível ajuste dos itens do *backlog* e dos pacotes
- Validação ou nova seleção de infra-estrutura e de ferramentas de desenvolvimento
- Estimativas de custo (desenvolvimento, materiais, marketing e treinamento).

Vale a pena observar que ao final do desenvolvimento de uma *release* há um retorno à etapa de planejamento, de forma que um novo ciclo de atividades será planejado, levando em conta os itens que restaram no *backlog*, além dos que foram adicionados ao longo dos ciclos de desenvolvimento, como será visto mais à frente, no item 3.3.3.

3.3.2 – Projeto Arquitetural

Podendo ser considerada uma sub-etapa de Planejamento, porém merecendo destaque, a etapa de concepção do projeto arquitetural do sistema envolve as seguintes atividades:

- Revisão dos itens de *backlog* selecionados
- Identificação das mudanças necessárias para implementar os itens de *backlog*
- Realização da análise de domínio
- Refinamento da arquitetura para suportar os novos requisitos
- Organização da reunião de revisão (trabalho em equipe).

3.3.3 – Desenvolvimento (Sprint)

A fase de desenvolvimento é composta por ciclos iterativos que implementam um conjunto de itens de *backlog* definidos na fase de planejamento. Cada uma dessas iterações é chamada *sprint*. Um *sprint* é um conjunto de atividades de desenvolvimento conduzidas durante um período pré-definido, usualmente uma a quatro semanas. Esse período varia de acordo com a

complexidade dos requisitos a serem trabalhados, com a avaliação de riscos e com o grau de detalhes desejado. A gestão de risco é realizada continuamente ao longo da fase de desenvolvimento, os planos de risco são constantemente atualizados. Cada *sprint* é composto de ou mais times, resultando em uma engenharia concorrente. A gerência é responsável por atribuir tarefas a cada uma das equipes.

Durante o desenvolvimento encontramos os seguintes macros processos:

- Reunião com os times para revisar planos de distribuição.
- Distribuição, revisão e ajuste dos padrões com os quais o produto estará em conformidade.
- Iterações (*sprints*) até que o produto esteja pronto para distribuição.

As atividades dos times que compõe um *sprint* são:

- **Desenvolvimento** – Composta pelas seguintes sub-atividades:
 - Definição de modificações necessárias nos **pacotes** para implementar os itens de *backlog* em questão.
 - Análise de domínio.
 - Projeto.
 - Produção de código.
 - Testes.
 - Documentação.
- **Empacotamento** – Criação de a uma versão executável de software.
- **Revisão** - Reunião diária com todos os times para apresentação do trabalho realizado, revisão do progresso de cada time, descrição de problemas e definição de novas tarefas. A reunião dura em média quinze minutos, ela propicia uma maior integração entre os membros da equipe, o compartilhamento de conhecimentos e a minimização de riscos.
- **Ajuste** – Conclusão de uma reunião de revisão. Visa consolidar as informações adquiridas durante uma reunião.

Cada *sprint* é seguido por uma reunião. Nesta reunião estão presentes e participam todos os times envolvidos no *sprint* e o pessoal de gerência de projeto.

Clientes, vendedores e pessoal de marketing também podem participar da reunião. Durante a reunião são revistos os requisitos funcionais implementados, as mudanças efetuadas nos pacotes e o software executável produzido. A forma como itens de *backlog* foram implementados pode ser modificada de acordo com o resultado das revisões. Novos itens de *backlog*, que se apresentem necessários durante a reunião, podem ser adicionados na lista de *backlog*, isto pode provocar mudanças em relação à seleção de requisitos para versões futuras. A fase de desenvolvimento é apresentada graficamente na figura 1.

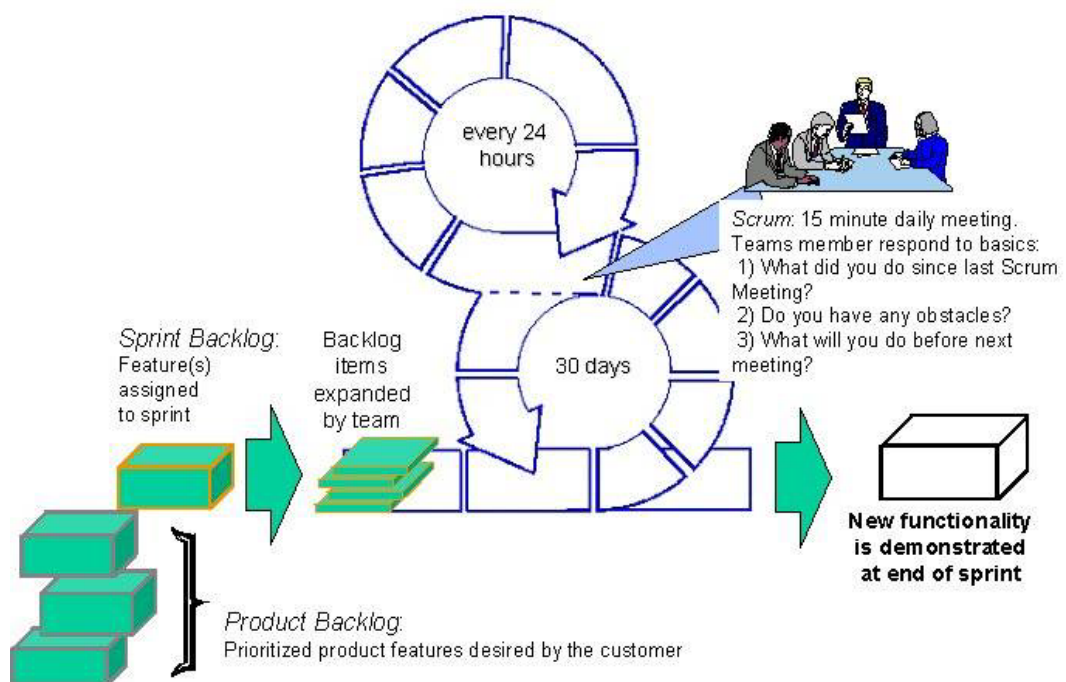


Figura 1 - Fase de desenvolvimento

3.3.4 – Encerramento

Ao final de um conjunto de *sprints*, quando a gerência percebe que tem uma versão estável com as novas funcionalidades requeridas, com qualidade e pronta para competir, essa nova versão é declarada encerrada e o processo entra em uma fase encerramento. Esta fase prepara essa versão estável para ser lançada no mercado. Entre as atividades da fase de encerramento encontramos:

- Testes de Sistema

- Testes de Integração
- Criar documentação do usuário
- Preparar material para marketing e treinamento

A fase de encerramento é ilustrada na figura 2.

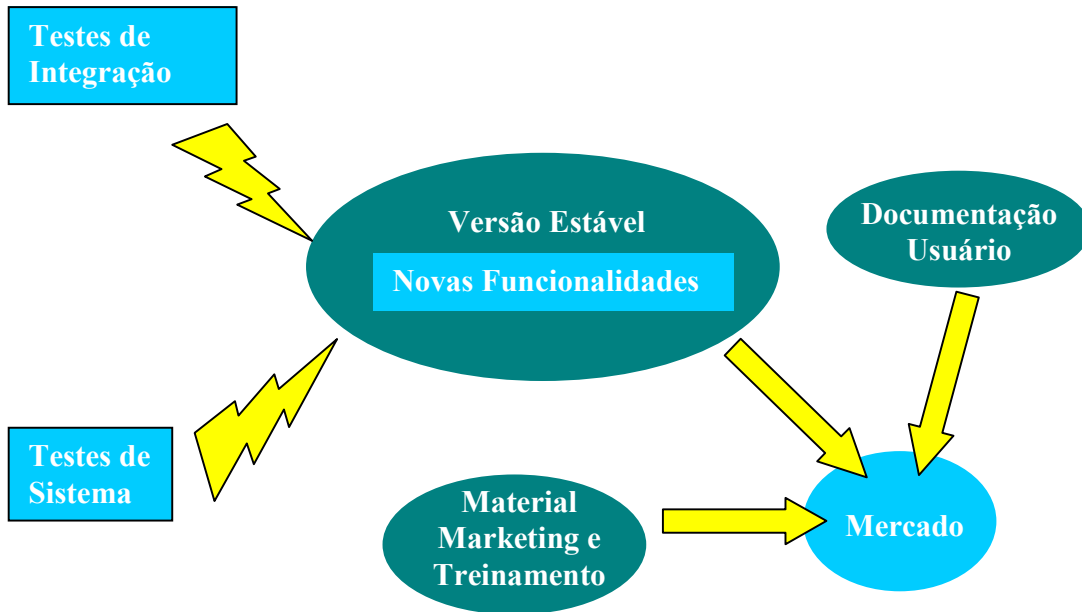


Figura 2 - Fase de encerramento

3.3.5 – Características

Para a melhor compreensão do funcionamento do Scrum é importante atenção a algumas características, tais que diferem o Scrum de outros modelos de gerenciamento, tornando-o singular.

As fases de planejamento e encerramento possuem entradas e saídas bem definidas. A entrada da fase de planejamento é o conjunto de requisitos iniciais, que juntos formam o *backlog*. Como saída tem-se a release estruturada, com os itens (requisitos) que serão trabalhados, a equipe de profissionais, etc. A fase de encerramento possui como entrada uma versão estável do produto e como saída uma versão pronta para o mercado, ou seja, testada, devidamente documentada e com material de marketing preparado.

A fase de desenvolvimento é um processo empírico flexível e não linear. Esta flexibilidade é um dos maiores diferenciais do Scrum. A equipe de desenvolvimento durante os *sprints* se comunicam bastante e decidem sobre os problemas ou necessidades do projeto nessa fase. Entretanto há um controle sutil externo, ao nível de resultados.

Outro aspecto importante é que as características do projeto podem ser mudadas a qualquer momento, até chegar à fase de encerramento. Contudo, durante um determinado ciclo (*sprint*), não há mudança de requisitos (externos), o que foi planejado é o que deve ser desenvolvido. Isso assegura a equipe de desenvolvimento acerca do que deve ser o seu trabalho, mesmo que mais tarde, em outros *sprints*, ocorram mudanças.

3.4 – Recursos Humanos

O pessoal envolvido no desenvolvimento de um projeto que utiliza a metodologia Scrum é formado por desenvolvedores, gerentes e algumas entidades externas, tais como pessoal de marketing, vendedores e clientes.

Em processos tradicionais os clientes são mantidos distantes da equipe de desenvolvimento do produto, isto devido ao medo de que os clientes possam complicar muito o processo e por julgar a interferência desnecessária. O Scrum, entretanto, incentiva e facilita a participação dos clientes em revisões e reuniões. Com essa participação teremos uma maior probabilidade de que as versões do software atendam de forma precisa as expectativas do cliente.

O pessoal que faz parte da equipe de projeto pode ser enquadrado em uma das duas categorias:

- Gerência – Liderada pelo gerente de projeto. Essa categoria tem uma participação fundamental na definição das funcionalidades de uma versão (juntamente com os clientes) e do tempo de desenvolvimento. Ela também é responsável pelo gerenciamento do progresso do desenvolvimento, das mudanças e dos riscos. Outra

função importante é manter sempre atualizado o *backlog*, de forma que ele reflita as necessidades do ambiente.

- Desenvolvimento – Times de desenvolvimento são pequenos, cada um composto por desenvolvedores, documentadores e pessoal de controle de qualidade. Os times contêm entre três e seis pessoas. Cada uma é designada para trabalhar com um conjunto de pacotes (ou objetos), portanto, deve conhecer a fundo os itens de *backlog* associados a esse conjunto de pacotes. Os times definem as mudanças nos pacotes que serão necessárias para implementar um item de *backlog*, eles também são responsáveis por gerenciar os problemas causados por essas mudanças. Os times podem ser formados com base em:
 - Conjunto de funcionalidades - Times que atuam no desenvolvimento relacionado a um determinado conjunto de funcionalidades.
 - Camadas do sistema – Times que atuam no desenvolvimento relacionado a determinadas camadas do sistema.

Desenvolvedores são selecionados com base no perfil exigido pela atividade a ser desenvolvida.

3.5 – Vantagens

Dentre as vantagens do Scrum podemos destacar:

- Divisão de responsabilidades – Cada integrante da equipe possui o seu papel bem definido, assim temos um trabalho direcionado onde cada um é responsável por seus atos.
- Processo ágil e flexível – O processo se adapta ao ambiente, incorporando as inúmeras mudanças no decorrer do projeto.
- Foco em controle e gerenciamento – O Scrum dedica-se exclusivamente a atividades de controle e gerência, tais como a minimização do risco e maximização de qualidade.

- Times pequenos – Pequenas equipes têm maior produtividade, colaboração e são mais fáceis de serem gerenciadas.

3.6 – Desvantagens

Dentre as desvantagens do Scrum podemos destacar:

- Ausência de práticas de Engenharia de Software (técnicas e notações) e ferramentas.
- Necessidade de associação com outras metodologias.
- Dificuldades na implementação de mudanças ocorridas durante um *sprint*.

4 – XP & SCRUM

O Scrum pode ser considerada uma metodologia ágil que trata apenas de aspectos gerenciais do desenvolvimento de produtos, no nosso caso produto de software. Portanto, o Scrum não possui técnicas nem procedimentos relativos à engenharia de software, ou seja, não trata aspectos específicos (não gerenciais) de produção de sistemas, como tratamento de requisitos, método de análise e concepção de interface, entre outros. Considerando estes aspectos, o XP (Extreme Programming), é um parceiro ideal para o SCRUM.

XP também é uma metodologia ágil, desprovida de práticas de gerenciamento e focada na qualidade de produtos de software especialmente desenvolvidos em ambientes turbulentos¹. XP e Scrum possuem práticas e regras complementares. Combinadas e bem utilizadas, elas podem prover a implementação de funcionalidades incrementalmente, melhoria constante na qualidade do processo e do software, adequando-se melhor às necessidades dos usuários.

Em linhas gerais, as principais características indicadoras do uso em conjuntos das duas metodologias são:

- As duas metodologias são ágeis
- Scrum possui práticas e regras de gerenciamento para maximizar a produtividade
- Scrum não possui práticas de engenharia
- XP provê as práticas integradas de engenharia
- XP não possui práticas de gerenciamento

¹ XP, assim como SCRUM, também pode ser utilizado em ambientes sem características de turbulência. Nesses casos os resultados, bem como as vantagens e desvantagens da utilização do conjunto XP&SCRUM podem ser diferentes do que o exposto neste texto.

5 – Conclusões

O Scrum, através do seu modelo flexível e de reuniões freqüentes, provê o aumento da comunicação e cooperação entre os membros da equipe, se comparado às metodologias densas. Além disso, possibilita a detecção e remoção com maior facilidade das causas que estão atrapalhando o desenvolvimento e entrega dos produtos.

Somando-se às considerações acima, após esta pesquisa, concluímos que o Scrum estimula o controle e organização do desenvolvimento de múltiplos produtos e projetos inter-relacionados, onde milhares de pessoas podem estar envolvidos, incentivando a produtividade.

6 – Bibliografia

- Linda Rising and Normam S. Janoff. The Scrum Software Development Process for Small Teams. *AG Communication Systems*.
- Ken Schwaber. SCRUM Development Process. *Advanced Development Methods*.
- <http://www.controlchaos.com>. Visitado em 02/08/2004.
- Rogério Guaraci dos Santos Giulian e Dalton Luz www.ime.usp.br/~gdaltonl/ageis/ageis_6pp.pdf. Visitado em 02/08/2004.
- <http://www.fawcette.com>. Visitado em 02/08/2004.