

UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE SISTEMAS DE COMPUTAÇÃO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

DIOGO F. MELETTO
FABIO STAPAIT PEDARNIG
RODRIGO J. DA FONSECA
VITOR BOSCHI DA SILVA

GERENCIAMENTO DO TEMPO DO PROJETO



SÃO CARLOS

2010

DIOGO F. MELETTO
FABIO STAPAIT PEDARNIG
RODRIGO J. DA FONSECA
VITOR BOSCHI DA SILVA

GERENCIAMENTO DO TEMPO DO PROJETO

Trabalho apresentado à disciplina de Gerência de Projetos como requisito de avaliação parcial do curso de Bacharelado em Ciências da Computação da Universidade de São Paulo – Campus São Carlos.

Professora: Elisa Yumi Nakagawa

SÃO CARLOS

2010

Sumário

1 Contextualização.....	4
2 O que é gerência do tempo.....	4
3 Processos.....	5
4 Metodologias ágeis.....	11
5 SCRUM.....	12
6 SCRUM e PMBOK.....	12
7 Conclusões.....	13
8 Referências.....	13

1 Contextualização

Nos dois trabalhos anteriores a este, foi vistoso que é a Gerencia de Integração, que é o início do projeto, definição das partes envolvidas, contrato, definição de como as coisas vão funcionar, em outras palavras, a “gerência da gerência do projeto”, e foi visto sobre o que é Gerenciamento do Escopo do projeto, que trata dos limites de um projeto, ou seja, quais tarefas devem ou não ser feitas. O próximo passo é responder a pergunta: O quanto vai durar o projeto?

A gerência do tempo de projeto é uma das áreas de maior exigência dentro de um projeto, pois é uma das mais visíveis em sua gestão. Como foi visto em aula no começo do semestre, a qualidade de um projeto está fortemente relacionada a três fatores: custo, escopo e *tempo*. Então é fundamental ter a capacidade de se controlar o tempo de um projeto.



Figura 1: Elementos que determinam a qualidade de um projeto

O Guia PMBOK define 6 processos de gerenciamento de tempo. Fora o processo de controle, todos os outros processos são da fase de planejamento.

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring & Controlling Process Group	Closing Process Group
		⋮			
6. Project Time Management		6.1 Define Activities 6.2 Sequence Activities 6.3 Estimate Activity Resources 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
		⋮			

Figura 2: Processos de Gerenciamento do tempo do projeto no Guia PMBOK. Fonte: PMBOK – 4ª edição.

2 O que é gerência do tempo

A gerência do tempo refere-se a um conjunto de habilidades, ferramentas e técnicas usadas para gerenciar o tempo ao realizar tarefas específicas, projetos e metas. Este conjunto

engloba uma ampla gama de atividades, e estas incluem planejamento, alocação, estabelecimento de metas, delegação, análise do tempo gasto, monitoramento, organização, programação e priorização.

Os processos de gerenciamento do tempo, assim como os outros processos vistos anteriormente, interagem entre si e com os de outras áreas de conhecimento e podem envolver esforços de um grupo ou de uma pessoa, com base nas necessidades do projeto.

3 Processos

Para facilitar e tornar mais preciso o processo de desenvolvimento do cronograma do projeto, é necessário antes executar os processos de definição e sequenciamento de atividades, assim como estimar os recursos e tempo necessário para a finalização de cada uma delas.

3.1 Definir as atividades

O processo de definição das atividades toma como entrada a definição do escopo, os ativos organizacionais e os fatores ambientais relevantes ao projeto. Como ferramentas para definir as atividades, o PMBoK sugere o uso da técnica da decomposição, que consiste em quebrar cada uma das entregas definidas no escopo do projeto em diversas atividades, até que sejam simples o bastante para terem os recursos e tempo necessário para sua execução estimados com precisão. Como apoio para a técnica da decomposição, pode ser usado o Planejamento em Ondas, no qual as entregas vão sendo decompostas em atividades conforme o projeto avança e o entendimento sobre o projeto se torna mais claro, o que em geral resulta em uma lista de atividades mais precisa. Nos casos em que o projeto consiste em diversas etapas repetitivas (como no caso da construção de cada piso em um edifício), ou em entregas similares ao que já foi feito em algum projeto anterior, pode-se utilizar modelos para definir mais rápida e precisamente a lista de atividades. Além das técnicas já citadas, a consultoria de um especialista no(s) assunto(s) relevante(s) a cada entrega pode fornecer informações valiosas para o processo de definição das atividades, e produzir uma lista mais precisa e próxima da realidade.

O processo descrito produz documentos listando todas as atividades definidas, assim como suas origens e atributos (atividades que exigem recursos ou especialidades específicas, por exemplo). Também é gerada uma lista de marcos importantes para o projeto, que podem ajudar no acompanhamento do projeto e também impor certas restrições sobre o cronograma.

3.2 Sequenciar as atividades

Com a lista de atividades finalizada, o próximo passo é determinar as relações de dependência entre as diversas atividades. O PMBoK lista três tipos de dependências: obrigatórias, arbitradas e externas. Uma dependência obrigatória é uma dependência imposta pela natureza do projeto (por exemplo, não é possível levantar as paredes de uma casa antes de se ter o alicerce construído). Dependências arbitradas, por outro lado, são impostas pelo gerente do projeto, para assegurar o cumprimento de boas práticas de desenvolvimento do projeto. A dependência externa é uma dependência entre uma atividade do projeto e algo externo ao projeto (por exemplo o fornecimento de um dispositivo ou material por um terceiro). Uma das técnicas empregadas para efetuar toda essa modelagem é o uso de diagramas de precedência, no qual são empregadas caixas para representar cada atividade, e setas conectando-as e representando as dependências entre elas. Esse método define quatro tipos de dependência entre atividades: término para início, término para término, início para início, e início para término. Além dessas dependências, é possível aplicar antecipações e

atrasos às atividades, por exemplo, pode-se determinar que a atividade B deve ser iniciada duas semanas após o início da atividade A, e que a atividade C começa quando faltar uma semana para o término da atividade A. Uma ferramenta utilizada para agilizar o desenvolvimento das dependências é o uso de modelos baseados em outros projetos já desenvolvidos pela empresa. Caso os projetos (ou sub-projetos) sejam similares, o uso de modelos pode prover uma economia de tempo e recursos substancial no desenvolvimento deste processo.

O resultado deste processo é a atualização dos documentos do projeto, principalmente referentes aos prazos e riscos, e um diagrama de dependências contendo todas as atividades e explicações sobre os motivos das dependências arbitradas possivelmente impostas ao projeto.

3.3 Estimar os recursos das atividades

O processo de estimativa de recursos é extremamente importante para o cumprimento do projeto dentro do prazo e custo estimado. Nele são determinados quais recursos (materiais e humanos) são necessários para a execução de cada uma das atividades definidas nos processos anteriores. O uso do conhecimento de especialistas é de grande valia e pode também fornecer abordagens alternativas para a execução das atividades, o que permite um melhor balanceamento entre tempo e custo do projeto. Estimativas fornecidas pelos fornecedores também fornecem uma maneira bastante precisa de determinar a quantidade de material necessária e o uso de softwares de gerenciamento de projeto podem ajudar na agregação de todas as informações, em especial no caso das atividades mais complexas, que podem ser divididas em sub-atividades e assim facilitar a estimativa dos recursos.

O processo de estimativa de recursos resulta num documento contendo a estrutura analítica dos recursos necessários para a execução de cada atividade, similar à WBS, além de um documento com os requisitos de cada atividade. Nesse processo, podem ser identificados riscos, conflitos de recursos e o custo do projeto passa a ser estimado com maior precisão, gerando a necessidade de atualização de diversos documentos produzidos nos processos anteriores.

3.4 Estimar as durações das atividades

O objetivo deste processo é determinar o tempo necessário para executar cada atividade, tendo em vista os recursos estimados pelo processo anterior. O calendário de recursos da empresa, assim como todos os documentos produzidos pelos processos anteriores a este fornecem as informações necessárias para o gerente estimar o tempo requisitado pelas atividades. Para ajudá-lo, especialistas das áreas relevantes devem ser consultados, e pode-se recorrer à analogias com outras atividades, e à tempos conhecidos para a execução de uma unidade de trabalho (por exemplo, tempo necessário para colocar 1 m² de piso). A estimativa por analogia pode ser muito rápida e precisa quando as atividades comparadas são realmente similares, e a equipes executoras também possuem qualificações parecidas. A estimativa paramétrica é muito precisa, mas somente é aplicável em uma pequena quantidade de atividades. Com o objetivo de melhorar a qualidade das estimativas, pode-se recorrer à estimativa de três pontos (piores caso, melhor caso e caso mais provável), que utiliza uma média ponderada dos três casos e também fornece uma faixa de prováveis prazos para as atividades. Neste processo também podem ser inseridas reservas, que são prazos adicionais ao realmente necessário, para que em caso de atrasos, o projeto não seja prejudicado como um todo.

Neste processo podem ser identificados novos riscos e uma estimativa mais precisa do prazo dos projetos é obtida, gerando a necessidade de atualização dos documentos relevantes. Também é produzido um documento fornecendo a duração de todas as atividades, necessário para o desenvolvimento do cronograma final do projeto.

3.5 Desenvolver o cronograma

Desenvolver o cronograma é o processo de análise de sequências das atividades, suas durações, recursos necessários e restrições do cronograma visando criar o cronograma do projeto.

Entradas

- **Lista das atividades:** Descrito no processo 1.
- **Atributos das atividades:** Descrito no processo 1.
- **Diagramas de rede do cronograma do projeto:** Descrito no processo 2.
- **Requisitos dos recursos da atividade:** Descrito no processo 3.
- **Calendários dos recursos:** Descrito no processo 3.
- **Estimativas da duração da atividade:** Descrito no processo 4.
- **Declaração do escopo do projeto:** A declaração do escopo do projeto contém premissas e restrições que podem gerar um impacto no desenvolvimento do cronograma do projeto.
- **Fatores ambientais da empresa:** Os fatores ambientais da empresa que podem influenciar o processo Desenvolver o cronograma incluem, mas não estão limitados à ferramenta de elaboração do cronograma que pode ser usada no desenvolvimento do mesmo.
- **Ativos de processos organizacionais:** Os ativos de processos organizacionais que podem influenciar o processo Desenvolver o cronograma incluem, mas não estão limitados a:
 1. A metodologia de elaboração do cronograma e
 2. Calendário do projeto.

Ferramentas e Técnicas

- **Análise da rede do cronograma:** A análise de rede do cronograma é uma técnica que gera o cronograma do projeto. Usa várias técnicas analíticas, tais como o método do caminho crítico, o método da cadeia crítica, análise e - se e o nivelamento de recursos para calcular as datas de início e término mais cedo e mais tarde para as partes incompletas das atividades do projeto.
- **Método do caminho crítico:** O método do caminho crítico fornece o caminho crítico do projeto, que é a sequência de atividades que devem ser concluídas nas datas programadas para que o projeto possa ser concluído dentro do prazo final. Se o prazo final for excedido, é porque no mínimo uma das atividades do caminho crítico não foi concluída na data programada. É importante entender a sequência do caminho crítico para saber onde você tem e onde você não tem flexibilidade.
- **Método da cadeia crítica:** A cadeia crítica é uma técnica de análise de rede do cronograma que modifica o cronograma do projeto para que se leve em conta a

limitação de recursos. Inicialmente, o diagrama de rede do cronograma do projeto é construído usando-se como entradas as estimativas de durações com suas dependências necessárias e suas restrições definidas.

Saídas

- **Cronograma do Projeto:** O cronograma do projeto inclui pelo menos uma data de início e de término planejadas para cada atividade. Embora um cronograma possa ser apresentado em formato tabular, é com mais frequência apresentado graficamente, usando-se um ou mais dos seguintes formatos:
 - **Gráficos de marcos:** esses gráficos assemelham-se aos gráficos de barras, porém identificam somente o início ou término agendado para as entregas mais importantes e interfaces externas-chaves. Um exemplo está representado no cronograma de marcos da Figura 1.
 - **Gráfico de barras:** esses gráficos, onde as barras representam as atividades, mostra as datas de início e término da atividade, assim como as durações esperadas. Esses são relativamente de fácil leitura e são frequentemente usados em apresentações gerenciais. Para controle e comunicação gerencial, a atividade mais ampla e mais abrangente, algumas vezes chamada de atividade sumarizadora, é usada entre marcos ou através de múltiplos pacotes de trabalho interdependentes, sendo mostrada em relatórios de gráfico de barras. Um exemplo está representado no cronograma resumo da Figura 1 que é apresentado num formato estruturado EAP.
 - **Diagramas de rede do cronograma do projeto:** esses diagramas, com informações sobre as datas das atividades, normalmente mostra tanto a lógica da rede do projeto como suas atividades do seu caminho crítico. Podem ser apresentados no formato de diagrama de atividade no nó, como mostrado na Figura 6-7, ou no formato de diagrama de rede do cronograma com escala de tempo que às vezes é chamado de gráfico de barras lógico, como mostrado no cronograma detalhado na Figura 1. Esse exemplo mostra como cada pacote de trabalho é planejado como uma série de atividades relacionadas.

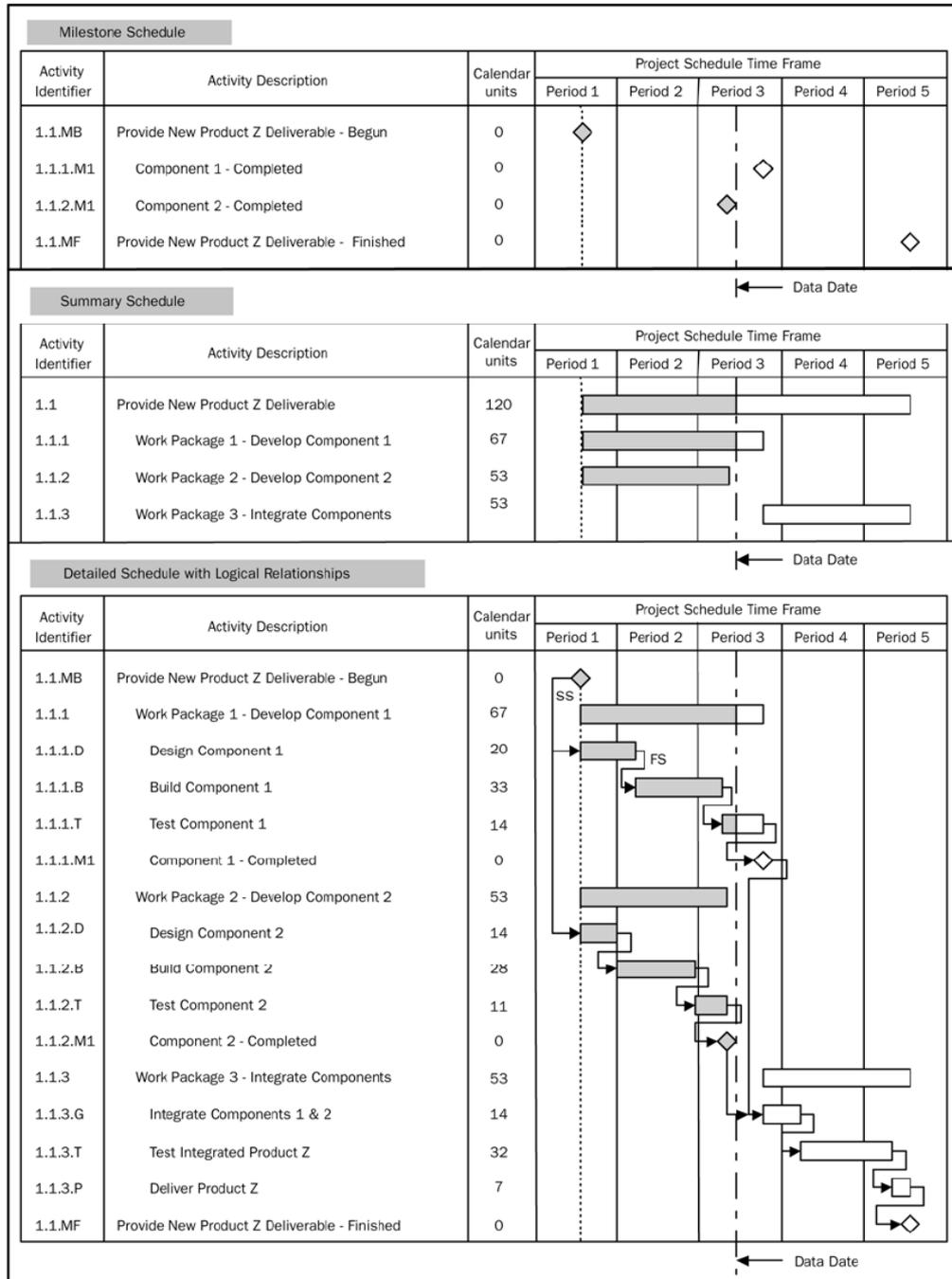


Figura 3: Tipos de cronograma. Fonte: PMBOK – 4ª edição.

- **Linha de base do cronograma:** Uma linha de base do cronograma é uma versão específica do cronograma do projeto desenvolvido a partir da análise de rede do mesmo. É aceita e aprovada pela equipe de gerenciamento como a linha de base do cronograma com datas de início da linha de base e datas de término da linha de base. A linha de base do cronograma é um componente do plano de gerenciamento do projeto.

3.6 Controlar o cronograma

Controlar o cronograma é o processo de monitoramento do andamento do projeto para atualização do seu progresso e gerenciamento das mudanças feitas na linha de base do cronograma.

Entradas

- **Plano de gerenciamento do projeto:** O plano de gerenciamento do projeto descrito no Gerenciamento de Integração contém o plano de gerenciamento do cronograma e a linha de base do mesmo. O plano de gerenciamento do cronograma descreve como este será gerenciado e controlado. A linha de base do cronograma é comparada aos resultados reais para determinar se uma mudança, ação corretiva ou preventiva é necessária.
- **Cronograma do projeto:** A versão mais recente do cronograma do projeto com a anotações indicando atualizações, atividades terminadas e atividades iniciadas até a data dos dados indicada.
- **Informações sobre o desempenho do trabalho:** Informações a respeito do progresso do projeto, tais como quais atividades foram iniciadas, os seus progressos e quais foram concluídas.

Ferramentas e Técnicas

- **Análise de desempenho:** As análises de desempenho medem, comparam e analisam o desempenho do cronograma como as datas reais de início e término, porcentagem completa e duração restante para o trabalho em andamento.
- **Análise da variação:** Medições do desempenho do cronograma (VP, IDP) são usadas para avaliar a magnitude de variação à linha de base do cronograma. A variação da folga total também é um componente essencial de planejamento para avaliar o desempenho de tempo do projeto.
- **Software de gerenciamento de projetos:** *Um software* de gerenciamento de projetos para agendamento fornece a habilidade de se rastrear datas planejadas versus datas reais e prever os efeitos de mudanças no cronograma.

Saídas

- **Medição do desempenho do trabalho:** Os valores da VP e do IDP calculados para os componentes da EAP, em particular os pacotes de trabalho e contas de controle, são documentados e comunicados às partes interessadas.
- **Atualizações de ativos de processos organizacionais:** Os ativos de processos organizacionais que podem ser atualizados incluem, mas não se limitam a:
 - Causas das diferenças;
 - Ação corretiva escolhida e as razões e
 - Outros tipos de lições aprendidas a partir do controle do cronograma do projeto.
- **Solicitações de mudança:** A análise de variação do cronograma, conjuntamente às revisões dos relatórios de análise de progresso, resultados de medições de desempenho e modificações no cronograma do projeto podem resultar em solicitações de mudanças

na linha de base do cronograma e/ou de outros componentes do plano de gerenciamento.

4 Metodologias ágeis

Todo o desenvolvimento de software pode ser caracterizado como um ciclo de solução de um problema, no qual são encontrados quatro estágios distintos: situação atual, definição do problema, desenvolvimento técnico e integração da solução. Realisticamente, é difícil compartmentalizar as atividades dessa forma, pois interferências ocorrem dentro e entre estágios. Devido a isso, existem modelos que processo para engenharia de software que pretendem trazer uma ordem para a atividade de desenvolvimento de forma a ajudar no controle e na coordenação de um projeto de software real.

O modelo conhecido como cascata sugere uma abordagem sistemática sequencial para o desenvolvimento de software, que começa no nível de sistema e progride através da análise, projeto, codificação, teste e manutenção. Há uma tentativa de definir, em detalhes, todos ou a maioria dos requisitos antes da programação. Igualmente, há uma tentativa de definir um plano ou cronograma confiável logo no começo.

Beck [Beck, 2000] diz que o risco é o problema básico do desenvolvimento de software e cita alguns exemplos desses riscos:

- Estouro do prazo de entrega: chega o dia da entrega e há a necessidade de se contar para o cliente que o software não estará pronto nos próximos 6 meses;
- Cancelamento do projeto: após inúmeros deslizos, o projeto é cancelado antes mesmo de ter entrado em produção;
- Degradação do sistema: o software é colocado em produção mas, após alguns anos, o custo para se fazer mudanças ou corrigir defeitos é tão grande que é melhor substituí-lo;
- Muitos defeitos: o software é colocado em produção mas o grande número de defeitos faz com que ele nunca seja usado;
- Regras de negócio erradas: o software é colocado em produção mas não resolve os problemas para os quais ele foi desenvolvido;
- Mudanças nas regras de negócio: o software é colocado em produção, no entanto, o problema para o qual ele foi desenvolvido já não existe mais e um novo problema, mais urgente, já está há seis meses no lugar desse último;
- Funcionalidades desnecessárias: o software tem um grande número de funcionalidades interessantes mas nenhuma delas traz lucro para o cliente; e
- Mudança da equipe: após dois anos, todos os bons programadores da equipe começam a odiar o projeto e abandonam a equipe.

De acordo com Larman [Larman, 2005], o desenvolvimento iterativo e evolutivo, em contraposição ao ciclo de vida em cascata, envolve a imediata programação e teste de um sistema parcial em ciclos repetidos. Também considera normalmente que o desenvolvimento começa antes que os requisitos tenham sido definidos em detalhe; a realimentação é usada para esclarecer e aperfeiçoar as especificações em evolução.

Ainda segundo Larman, o desenvolvimento iterativo e evolutivo, também conhecido como desenvolvimento ágil, em contraposição ao ciclo de vida em cascata, envolve a imediata programação e teste de um sistema parcial em ciclos repetidos. Também considera normalmente que o desenvolvimento começa antes que os requisitos tenham sido definidos em detalhe; a realimentação é usada para esclarecer e aperfeiçoar as especificações em evolução.

5 SCRUM

Scrum é um método ágil para desenvolvimento de software criado por Ken Schwaber e Jeff Sutherland no início da década de 1990 [Schwaber, 2004].

Segundo Schwaber, o Scrum é um processo simples para gerenciamento de projetos complexos, com poucas regras e que são fáceis de se aprender. É utilizado em trabalhos complexos, nos quais é impossível se prever tudo o que vai ocorrer. O Scrum já foi usado com sucesso em milhares de projetos em centenas de organizações nos últimos 10 anos.

O coração do Scrum mora na iteração. O time verifica os requisitos, considera as tecnologias disponíveis, e avalia suas próprias habilidades e capacidades. A partir daí, determina como implementar a funcionalidade, modificando essa abordagem diariamente conforme são encontradas novas complexidades, dificuldades e surpresas. A equipe, então, imagina o que deverá ser feito e escolhe o melhor método para fazer isso. Esse processo criativo é o coração da produtividade do Scrum.

Existem três papéis no Scrum: o Dono do Produto, a Equipe e o *Scrum Master*. O gerenciamento de todas as responsabilidades no projeto é dividido entre esses três papéis. O Dono do Produto, é o responsável por representar os interesses de todos, olhando ao mesmo tempo o projeto e o sistema sendo produzido. Ele cria uma lista de requisitos do sistema, chamada de *Product Backlog*, uma lista dos retornos em investimento pretendidos com o sistema, chamada de ROI (do inglês *Return On Investment*) e os planos de entrega das iterações. Também é responsável por garantir que as funcionalidades mais importantes contidas no *Product Backlog* serão implementadas primeiro. A Equipe é a responsável por implementar as funcionalidades. Equipes se auto-gerenciam e se auto-organizam, além de serem multifuncionais. Todos os membros da Equipe são responsáveis pelo sucesso da iteração e do projeto como um todo. O *Scrum Master* é o responsável pelo processo do Scrum, por ensinar o Scrum a todos envolvidos no projeto, por implementar o Scrum de tal modo que ele se ajuste à cultura da organização e tenha os benefícios desejados, e por assegurar que todos sigam as práticas e regras do Scrum.

Um projeto Scrum começa com uma visão do sistema a ser desenvolvido, que pode ser vaga no início, talvez escrita numa linguagem mais voltada ao mercado do que à tecnologias, e que vai se tornando mais clara a medida em que se progride. O *Scrum Master* cria o *Product Backlog* com os requisitos funcionais e não funcionais. Ele também é o responsável por fazer com que seja criada uma visão que maximize seu ROI.

Todo o trabalho é executado em *Sprints*. Cada *Sprint* é uma iteração de 30 dias. Cada *Sprint* é iniciado com uma reunião de planejamento, onde o *Product Owner* e a Equipe se reúnem para decidir o que será feito no próximo *Sprint*. A partir do *Product Backlog* o *Product Owner* escolhe as funcionalidades com maior prioridade e a Equipe dirá o que e quanto será necessário para que essa funcionalidade esteja disponível no próximo *Sprint*. Essas reuniões não devem ultrapassar oito horas, já que o foco é trabalhar e não em pensar em trabalhar.

Todos os dias, a equipe se reúne para uma reunião diária chamada *Daily Scrum*. Nessa reunião, os membros da equipe deverão responder a três questões: O que foi feito no projeto desde a última *Daily Scrum*? O que se planeja fazer entre hoje e a próxima *Daily Scrum*? Quais são as dificuldades encontradas para esse *Sprint* e para o projeto?

No final de cada *Sprint*, é necessária uma reunião de revisão do *Sprint*. Essa é uma reunião, de no máximo quatro horas, onde a Equipe apresenta ao *Product Owner* o que foi desenvolvido e já conversa sobre o que será desenvolvido em seguida.

6 SCRUM e PMBOK

Todos os processos pertencentes à área de conhecimento Gerenciamento de Tempo do PMBOK possuem processos similares no Scrum:

- Definir atividades: esse processo pode ser realizado na definição do Product Backlog, quando são definidas as funcionalidades que o projeto deverá ter;
- Sequenciar as atividades: o *Product Owner* define as prioridades do Product Backlog e, assim, define uma sequência para as atividades do projeto;
- Estimar os recursos das atividades: os recursos podem ser definidos na reunião do *Sprint* e também no *Daily Scrum*;
- Estimar a duração das atividades: a duração das atividades são estimadas na reunião do *Sprint*, atualizando assim o *Sprint Backlog* através dos *Scrum Points*;
- Desenvolver o cronograma: o cronograma também é definido no *Sprint Backlog*; e
- Controlar o cronograma: o cronograma é atualizado diariamente e pode ter seu progresso medido através do *Burndown Chart*.

É válido lembrar que o PMBOK diz que os processos podem ser adaptados e utilizados com o rigor que o gerente de projetos achar necessário.

7 Conclusões

Gerenciamento do Tempo é saber quando um projeto vai iniciar e quando vai terminar. Uma boa gestão do tempo do projeto é um dos principais fatores que determinam a qualidade de um projeto. Como algumas atividades são dependentes umas das outras, um atraso no meio do projeto poderá ter repercussão até o final, e se descoberto tarde pode ser catastrófico para o projeto, por exemplo, adiar a entrega do projeto.

Mesmo que haja um atraso no projeto, através de uma boa gestão do tempo do projeto, é possível saber em quanto tempo o projeto irá atrasar e com antecedência suficiente para minimizar os danos.

8 Referências

[PMBOK] Project Management Institute, Guia PMBOK, 4ªed, 2009.

[Beck, 2000] Beck, K., **Extreme Programming Explained: Embrace Change**, first edition, Addison Wesley, 2000.

[Cockburn, 2001] Cockburn, A., **Agile Software Development**, Draft version: 3b, Alistair Cockburn, 2001.

[Larman, 2005] Larman, C., **Utilizando UML e Padrões**, terceira edição, São Paulo - SP, Bookman, 2005.

[Pressman, 2002] Pressman, R., **Engenharia de Software**, quinta edição, Rio de Janeiro - RJ, McGraw-Hill, 2002.

[Schwaber, 2004] Schwaber, K., **Agile Project Management with Scrum**, first edition, Washington, Microsoft Press, 2004.

[Sommerville, 2003] Sommerville, I., **Engenharia de Software**, sexta edição, São Paulo - SP, Addison Wesley, 2003.

