



INSTITUTO FEDERAL DE EDUCAÇÃO,  
CIÊNCIA E TECNOLOGIA DO RIO  
GRANDE DO SUL  
CAMPUS Canoas

# FASE DE PROJETO Aula 07

Mariano Nicolao  
[mariano.nicolao@canoas.ifrs.edu.br](mailto:mariano.nicolao@canoas.ifrs.edu.br)

## **1. Introdução**

O gerenciamento de projetos de software é muito importante no contexto da engenharia de software. Ser o gerente de software em uma organização é um dos possíveis cargos que um engenheiro de software poderá assumir profissionalmente. Um gerente que atua com pessoas, processos, recursos diversos, etc (REZENDE, 2002).

Os gerentes de software realizam um trabalho similar a aquele que outros gerentes de outras áreas da engenharia fazem. No entanto, lembrar que a engenharia de software é diferente de outros tipos de engenharia, principalmente devido a que o software é intangível e não pode ser visto ou tocado, e isto dificulta o gerenciamento de software (SOMMERVILLE, 2007). No contexto da construção de um projeto de engenharia civil, o gerente do projeto pode ver o andamento da construção e no contexto do gerente de software precisa interagir com os projetistas para obter informações (relatórios e documentos que descrevem o estado do software que é desenvolvido) necessárias para examinar o progresso do projeto.

Na seção 2, se refere a gerenciamento de projetos de sistemas de software. Essa seção é organizada em três subseções: atividades de gerenciamento, planejamento de projeto e cronograma do projeto.

## **2. Gerenciamento de Projetos de Sistemas de Software**

No contexto do desenvolvimento de um sistema de software, um mau gerenciamento, geralmente, provocará em falhas no projeto, isto é acontecerá atraso na entrega do sistema, o custo aumentará, e haverá falhas no atendimento dos requisitos (SOMMERVILLE, 2007). A gestão efetiva de projetos de software foca: pessoal, produto de software, processo (modelo de processo mais apropriado) e projeto (plano de projeto) (PRESSMAN, 2010).

A responsabilidade do desenvolvimento de planos e cronogramas do projeto é do gerente de software. Também os gerentes devem supervisionar o desenvolvimento do software para verificar o uso de padrões exigidos e se o trabalho está dentro do prazo e orçamento previstos.

Uma equipe de desenvolvimento de software típica consiste em (BEZERRA, 2007): um gerente, analistas (devem ter conhecimento do domínio do negócio), projetistas

(avaliam alternativas de solução do problema e especificam uma solução computacional detalhada), arquiteto de software (elabora a arquitetura do sistema e identifica quais são os subsistemas que compõem o sistema assim como as interfaces entre esses subsistemas), programadores (responsáveis pela implementação do sistema), cliente usuário (especialista do domínio), cliente contratante e avaliadores de qualidade (asseguram o uso de padrões de qualidade estabelecidos pela organização).

## **2.1 Atividades de gerenciamento**

A seguir uma lista de atividades que são realizadas, algumas ou todas, pela maioria dos gerentes de projetos de software:

### Elaboração de proposta

A proposta descreve os objetivos do projeto e como ele será realizado. Geralmente inclui a estimativa de custos, o cronograma e justificativa. A elaboração de propostas é muito importante já que a elaboração de propostas, para muitas organizações de software, implica a possibilidade de realizar um contrato, isto é fundamental para a existência dessas organizações.

### Planejamento e desenvolvimento do cronograma do projeto

O planejamento de projeto implica identificar as atividades, marcos (ver a seção 2.2.1 deste documento) e produtos gerados por um projeto (SOMMERVILLE, 2007). É necessária a elaboração de um plano de projeto (ver a seção 2.2.1 deste documento) para guiar o desenvolvimento em direção aos objetivos do projeto.

### Custo de projeto

A estimativa de custos é uma atividade relacionada à estimativa dos recursos necessários para realizar o plano do projeto.

### Monitoramento e revisões de projetos

O gerente deve realizar o acompanhamento do progresso do projeto e comparar com o planejamento previsto, também deve ser monitorado os custos reais utilizados e os planejados. A maioria das organizações definem mecanismos formais de monitoração. Também, o gerente pode obter informações de forma informal, por exemplo através de discussões informais (por exemplo no canto do cafezinho ou no corredor) com alguns

membros da equipe, isto pode ser importante já que um problema, por exemplo defeito de software, poderá ser identificado, na conversa informal com algum membro da equipe, antes de ser relatado e agir rapidamente, objetivando resolver esse problema, alocando um especialista para resolver, e evitar um atraso no cronograma.

As revisões de gerenciamento do projeto objetivam a revisão geral do progresso e o desenvolvimento técnico do projeto e a verificação se o projeto e as metas da organização que está financiando o software estão alinhadas (SOMMERVILLE, 2007).

### Seleção e avaliação de pessoal

O gerente de software precisa selecionar pessoas para trabalhar nos projetos. O ideal é que exista a melhor equipe possível, isto é, pessoas com experiência adequada para trabalhar no projeto (SOMMERVILLE, 2007). Na prática, isto não necessariamente acontecerá, então será preciso trabalhar com uma equipe disponível naquele momento, isto poderá acontecer devido a restrições no orçamento, ou a equipe ou pessoal com experiência não estará disponível para o período do projeto ou a organização quer desenvolver as habilidades de uma equipe inexperiente. Em relação a organização, o desenvolvimento de habilidades de uma equipe experiente, isto é tornar uma equipe inexperiente em uma equipe experiente é muito importante, já que isto vai possibilitar que depois de um período, a equipe terá condições de atender tecnicamente, de forma satisfatória, as necessidades da organização em termos de desenvolvimento de sistemas de software.

### Elaboração de relatórios e apresentações

Os gerentes de projeto devem se comunicar eficientemente, de forma verbal e escrita (SOMMERVILLE, 2007). Eles devem se comunicar com clientes, pessoas da alta gerencia, e com a equipe de desenvolvimento, para cada interação, o gerente de projeto deve comunicar considerando a pessoa com quem ele interage. Por exemplo, se o gerente fala com a equipe de desenvolvimento, então ele deve falar tecnicamente de forma a se entender com o pessoal de desenvolvimento, agora se o gerente fala com a alta gerencia, ele deve falar de tal forma (objetiva e clara - destacando as informações relevantes) que possa ser compreendido sem precisar aprofundar nos detalhes técnicos, já que pode ser difícil estabelecer um dialogo técnico com as pessoas da alta gerencia que geralmente não terão conhecimento técnico aprofundado. Também é importante

lembrar que a alta gerencia não dispõe de muito tempo, então o tempo disponível nas reuniões com as pessoas da alta gerencia deve ser utilizado da melhor forma possível. Os gerentes devem preparar relatórios e apresentações sobre o projeto para o cliente e direção da organização dele ou contratante. Eles devem redigir documentos concisos e coerentes que resumam as informações contidas em relatórios detalhados do projeto.

## 2.2 Planejamento de projeto

O gerenciamento eficiente de um projeto de software depende de um planejamento detalhado do progresso do projeto (SOMMERVILLE, 2007).

O plano elaborado inicialmente deve ser utilizado como guia que deve evoluir em função do avanço do projeto e mais informações se tornarão disponíveis. À medida que as informações se tornam disponíveis, durante a realização do projeto, o plano deve ser revisado.

O planejamento de projeto para desenvolvimento de software é um processo iterativo representado na Figura 1 (SOMMERVILLE, 2007), apenas concluído quando o

- *Estabeleça as restrições (data de entrega, pessoal disponível, orçamento, ...) do projeto*
- *Faça a avaliação inicial dos parâmetros (estrutura, tamanho, ...) do projeto*
- *Defina os marcos do projeto e os produtos a serem entregues*
- *while projeto não foi concluído ou cancelado loop*
  - Elabore um cronograma do projeto*
  - Inicie as atividades de acordo com o cronograma*
  - Aguarde (por um período)*
  - Examine o progresso do projeto*
  - Revise as estimativas de parâmetros do projeto*
  - Atualize o cronograma do projeto*
  - Renegocie as restrições do projeto e os produtos a serem entregues*
  - if surgirem problemas then*
    - Inicie revisão técnica*
  - end if*

**Fonte: (Sommerville, 2007)**

**Figura 1: Planejamento do projeto – processo iterativo**

### **2.2.1 O plano de projeto**

O plano de projeto estabelece os recursos disponíveis para o projeto, a estrutura analítica do projeto e um cronograma para a realização do trabalho.

A estrutura do plano para o processo de desenvolvimento, na maioria das vezes deve incluir as seguintes seções (SOMMERVILLE, 2007):

#### Introdução

Descrição dos objetivos do projeto e estabelece as restrições, tais como orçamento, prazo, etc que afetam o gerenciamento do projeto.

#### Organização do projeto

Descreve o modo como a equipe de desenvolvimento está organizada, pessoas envolvidas e seus papéis na equipe.

#### Análise de riscos

Descrição dos possíveis riscos<sup>1</sup> do projeto e as propostas de estratégias de redução de riscos (SOMMERVILLE, 2007).

#### Requisitos de recursos de hardware e de software

Descrição de hardware e software necessários para realizar o desenvolvimento. Se for necessário realizar a aquisição de recursos, então devem ser realizadas as estimativas de preços e prazos de entrega.

#### Estrutura analítica

Estabelece a estrutura analítica do projeto em atividades e identifica os marcos e os produtos a serem entregues com cada atividade.

Um marco é um ponto final reconhecível de uma atividade do processo de software. A cada marco, deve existir uma saída formal, como um relatório, que possa ser apresentado à gerência. Por exemplo, no contexto do processo de

requisitos, podemos identificar como marcos: relatório de viabilidade, requisitos de sistema. Os relatórios

1 Risco: é um evento não desejado que provoca consequências negativas (PFLEEGER, 1998)

dos marcos, também, não precisam ser documentos extensos, podem ser simples e breves sobre o que foi concluído.

Um produto é um resultado de projeto entregue ao cliente. É geralmente disponibilizado no fim de alguma fase importante do projeto, como a especificação de requisitos, análise ou projeto.

Os marcos podem ser resultados internos do projeto que não serão entregues ao cliente. Neste caso os marcos são usados pelo gerente do projeto para verificar o progresso.

### Cronograma do projeto

Apresenta as atividades, prazo estimado para realizar cada atividade, dependências entre atividades e a alocação de pessoas nas atividades.

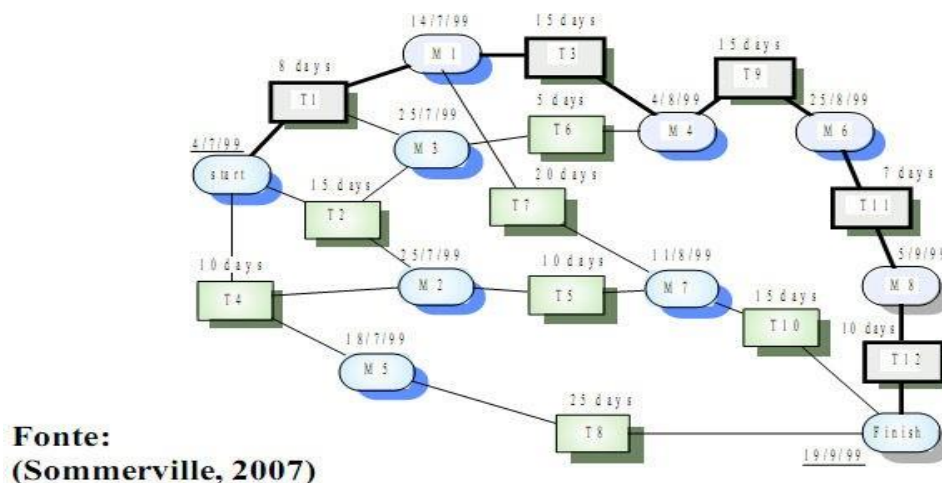
### Mecanismos de monitoração e elaboração de relatórios

Consiste na definição de relatórios de gerenciamento que devem ser produzidos, quando devem ser produzidos e os mecanismos de monitoração de projeto usados.

## **2.3 Cronograma do projeto**

O cronograma representa a divisão do trabalho total de um projeto em atividades separadas e a estimativa do tempo necessário para completar essas atividades. Deve ser considerado no cronograma os recursos necessários (pessoas envolvidas e outros) para realizar cada atividade. Os cronogramas são geralmente representados através de diagramas que mostram a duração e dependências de atividades, e alocação de pessoal.

Um diagrama (rede) de atividades mostra as dependências entre as diferentes atividades que constituem um projeto. No exemplo do diagrama da Figura 2, são mostradas as atividades ou tarefas (representadas através de um retângulo), a duração (em dias) associada e dependências (por exemplo a tarefa T3 depende de T1 – M1).



**Figura 2: Exemplo de diagrama (rede) de atividades.**

Um diagrama de barras<sup>2</sup> mostra quando uma atividade começa e quando termina. Também no diagrama de barras pode ser mostrada a alocação de pessoal às atividades do projeto de software.

A seguir são apresentados dos exemplos de cronograma.

### Exemplo 1 de cronograma

Na Figura 3, é apresentado um exemplo de cronograma, no contexto de um projeto de software, mostrando as atividades e o período de tempo necessário para a realização das mesmas. A duração prevista para esse exemplo de projeto de software é de um ano e as atividades consideradas são: engenharia de requisitos, análise, projeto, implementação, teste de unidade e subsistemas, testes de integração, validação do sistema, treinamento de usuários e liberação do sistema. Para cada atividade, no cronograma da Figura 3, deve ser colocado o nome do responsável pela atividade. Em algumas atividades também pode ser colocado os nomes dos envolvidos na realização da atividade, por exemplo no contexto da engenharia de requisitos (colocar o cliente e/ou usuários envolvidos), validação do sistema (colocar o cliente e/ou usuários envolvidos), treinamento (colocar os usuários ou setor da organização envolvido) e liberação do sistema (colocar o cliente e/ou usuários envolvidos).

<sup>2</sup> Diagrama de barras: também chamado de diagrama de Gantt. Mostra quando uma atividade começa e termina. Deve ser lido de esquerda a direita.



Período	1º Trimestre			2º Trimestre			3º Trimestre			4º Trimestre		
Atividades	1	2	3	4	5	6	7	8	9	10	11	12
Engenharia de Requisitos <i>Nome do responsável: A</i>												
Análise <i>Nome do responsável: B</i>												
Projeto <i>Nome do responsável: C</i>												
Implementação <i>Nome do responsável: D</i>												
Testes de unidade e subsistemas <i>Nome do responsável: E</i>												
Testes de integração <i>Nome do responsável: F</i>												
Validação de sistema <i>Nome do responsável: G</i>												
Treinamento de usuários e disponibilização de documentação <i>Nome do responsável: H</i>												
Liberação do sistema <i>Nome do responsável: I</i>												

**Figura 3: Exemplo de cronograma para o desenvolvimento de um sistema de software**

#### Exemplo 2 de cronograma

No formato de cronograma da Figura 4 (ainda sem a estimativa do tempo para cada atividade), é considerado um desenvolvimento incremental<sup>3</sup> de software, para isto nesse exemplo são considerados três incrementos no contexto da construção do sistema de software. Após a realização da especificação de requisitos, para cada incremento, as tarefas a serem realizadas são: análise, projeto, implementação e testes de unidade, incremento e integração. Posteriormente a construção dos três incrementos, serão realizadas as seguintes atividades: validação do sistema, treinamento de usuários e liberação de usuários.

<sup>3</sup> : Desenvolvimento incremental de software: consultar o texto da Unidade 2 – Processo de Software.

Período	1º Trimestre			2º Trimestre			3º Trimestre			4º Trimestre		
Atividades	1	2	3	4	5	6	7	8	9	10	11	12
Engenharia de Requisitos												
Nome do responsável:												
Incremento 1: Análise												
Nome do responsável:												
Incremento 1: Projeto												
Nome do responsável:												
Incremento 1: Implementação												
Nome do responsável:												
Incremento 1: Testes de unidade e incremento												
Nome do responsável:												
Incremento 2: Análise												
Nome do responsável:												
Incremento 2: Projeto												
Nome do responsável:												
Incremento 2: Implementação												
Nome do responsável:												
Incremento 2: Testes de unidade, incremento e integração												
Nome do responsável:												
Incremento 3: Análise												
Nome do responsável:												
Incremento 3: Projeto												
Nome do responsável:												
Incremento 3: Implementação												
Nome do responsável:												
Incremento 3: Testes de unidade, incremento e integração												
Nome do responsável:												
Validação do sistema (considerando todos os incrementos)												
Nome do responsável:												
Treinamento de usuários e disponibilização da documentação												
Nome do responsável:												
Liberação do sistema												
Nome do responsável:												

**Figura 4 : Exemplo de formato de cronograma para o desenvolvimento incremental no contexto de um sistema de software**

Cada uma das atividades dos exemplos de cronograma das Figuras 3 e 4 podem ser colocadas, também, de forma mais detalhada em outro cronograma. Por exemplo, a atividade Projeto, da Figura 3, poderia ser detalhada da seguinte forma:

#### Atividade: Projeto com 3 meses de duração.

Sub-atividade 1: Projeto de arquitetura (análise de alternativas e definição da arquitetura mais apropriada)

Duração: duas semanas de duração

Sub-atividade 2: Especificação de cada subsistema da arquitetura, interfaces e interações entre subsistemas da arquitetura

Duração: um mês de duração

Sub-atividade 3: Especificação de todos os componentes envolvidos na arquitetura

Duração: duas semanas de duração

Sub-atividade 4: Especificação de estruturas de dados e algoritmos

Duração: um mês de duração

### **3. Resumo**

No contexto do gerenciamento de projetos de sistemas de software, foram apresentados, neste documento, os seguintes tópicos: atividades de gerenciamento, planejamento de projeto e cronograma do projeto.

### **Referências Bibliográficas**

Bezerra E. Princípios de Análise e Projetos de Sistemas com UML, 2a edição. Elsevier- Editora Campus. 2007.

Pfleeger S. L. Software Engineering: theory and practice. Prentice-Hall.

1998. Pressman R. Engenharia de Software, 6a edição, AMGH Editora.

2010.

Rezende D. A. Engenharia de software e sistemas de informação, 2a edição. Brasport, 2002.

Sommerville, I. Engenharia de Software. 8a Edição. Addison Wesley. 2007.