Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

Tópicos Avançados de Programação Orientada a Objetos

Padrão Prazo

Adriano Saturno Muniz

Rachel de Paula

Sumário

1 Padrão Prazo

1.1 Intenção e Objetivo

Provê um meio de determinar prazos viáveis para projetos de software, isto é, minimiza a possibilidade dos projetos serem entregues com atraso.

1.2 Também conhecido como

Cronograma

1.3 Motivação

A determinação de prazos corretos para projetos é muito importante pois os requerentes do sistema podem previamente agendar tarefas presumindo que o projeto será entregue na data acordada. O atraso na entrega pode causar transtornos. Além disso, se o prazo não for suficiente e os desenvolvedores tiverem que fazer a implementação as pressas ou trabalhar horas excessivas, a qualidade do produto irá decair devido ao cansaço e a pressão existente para a entrega no prazo. A má qualidade do produto denigre ainda mais a imagem da equipe de desenvolvimento do que o atraso na entrega.

Considere a seguinte situação: o cliente faz uma requisição de um determinado projeto e o analista fica incumbido de fazer uma análise para determinar o tempo de desenvolvimento necessário.

Uma possivel solução, visando agilidade em dar uma resposta ao cliente, consistiria em determinar o prazo tendo como base uma análise superficial do problema. Esta solução, apesar de mais rápida, não é adequada, pois a possibilidade do projeto não ser concluído na data acordada é muito grande. Um dos principais causas de erro em análise decorre da falta de discernimento em relação a complexidade do problema. Muitos problemas aparentemente simples após uma análise mais aprofundada se revelam muito mais complexos e requerem mais esforço que o superficialmente previsto.

Outra soluçao consiste em fazer uma análise detalhada, obter um prazo para implementar o projeto e repassá-lo ao cliente sem nenhuma margem de segurança. Essa solução pode ser muito arriscada pois os contratempos de implementação foram ignorados : problemas nos servidores, falta de luz, doença, enfim qualquer coisa que possa impedir o desenvolvimento do sistema por algum tempo.

Conclui-se então que a determinação de prazos de desenvolvimento de projetos não deve ser baseada em análises superficiais e sempre deve ser acrescentada uma margem de segurança para possíveis contratempos. Durante a análise do prazo também deve ser levadas em consideração as fases de teste e documentação. Com relação a fase de testes, além dos testes constantes durante o desenvolvimento devem ser acrescentados pelo menos dois períodos de teste gerais: um de todo o sistema, logo após o desenvolvimento, seguido de uma fase de ajustes e correções, e outro de teste das eventuais correções feitas no primeiro período. É importante resaltar que a fase de teste deve ser a última antes da entrega do projeto ao cliente, garantindo-se dessa forma que o projeto sempre será entregue testado.

O padrão prazo provê um meio de determinar prazos dividindo os projetos em atividades cujo período necessário para o desenvolvimento é mensuravel. O padrão identifica também as atividades que podem ser executadas em paralelo, de modo a aproveitar os recursos disponíveis para que o prazo de desenvolvimento seja o menor possível.

1.4 Aplicabilidade

Use o padrão prazo quando:

- A data de entrega do projeto é crucial, não pode ser desrespeitada e a qualidade do produto não pode ser deixada de lado a fim de entregar o projeto dentro do prazo pré-estabelecido.
- uma análise inicial do projeto revelou um período curto de desenvolvimento: os períodos de desenvolvimento podem ser "auditados"pelo padrão.
- os projetos são complexos demais e a estimativa de tempo de desenvolvimento parece impossível. O padrão prazo divide tais projetos em atividades menores mensuráveis.

1.5 Estrutura

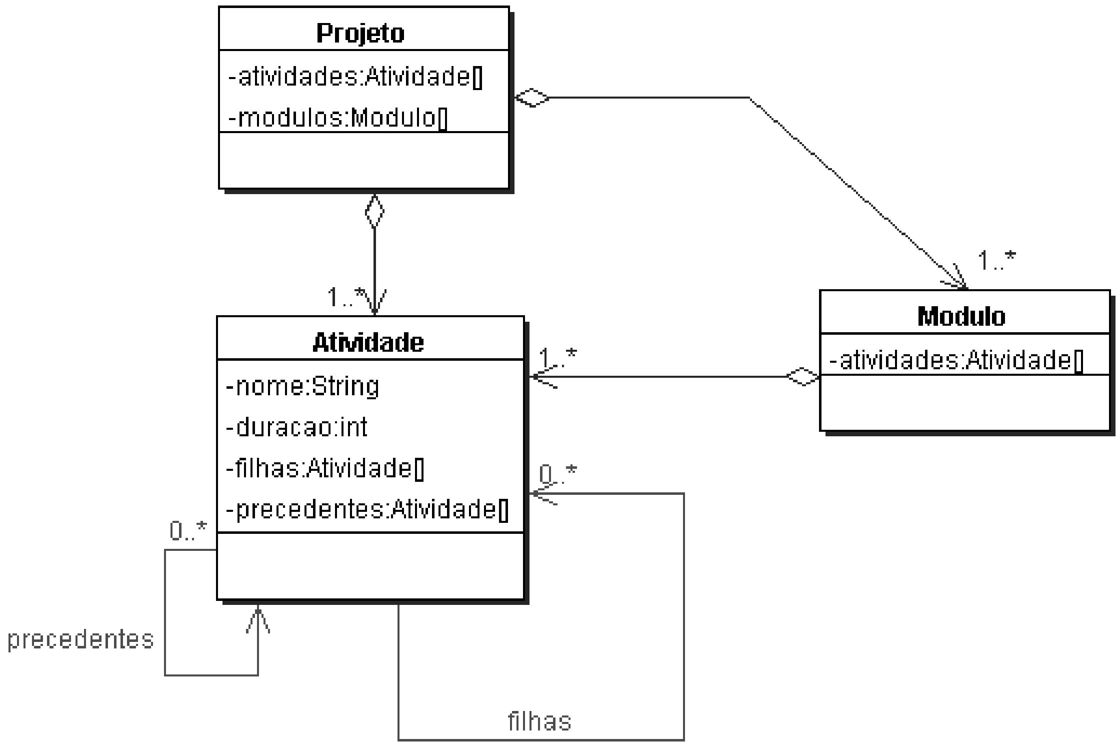

Figura 1: Diagrama de Classes

.

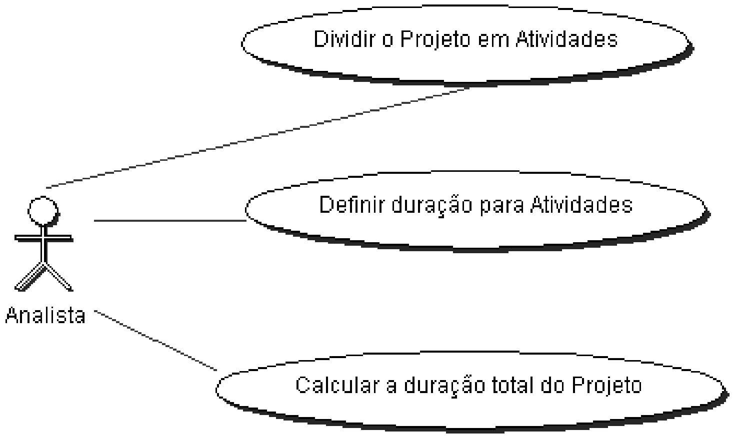

Figura 2: Diagrama de Casos de uso

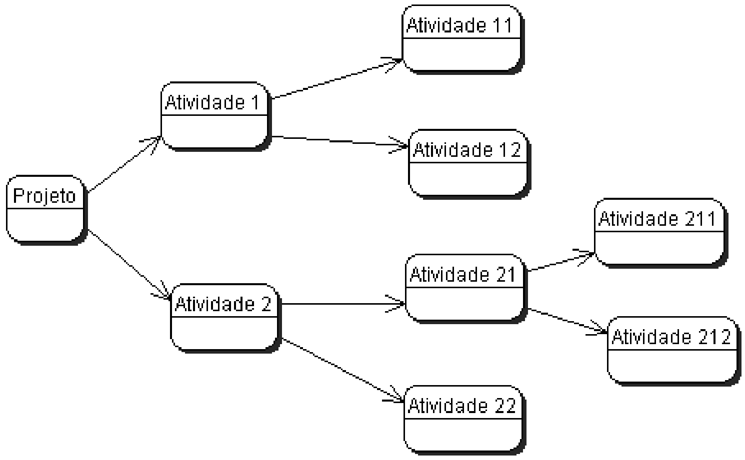

Figura 3: Diagrama de Divisao do Projeto

1.6 Participantes

Projeto:

- é dividido em Atividades que representam a quebra do mesmo em tarefas menores para facilitar a quantificação do tempo de desenvolvimento.
- também é divido em Módulos que representam como o projeto será entregue, isto é, módulos representam quais atividades serão entregues em conjunto. Alguns projetos não precisam ser entregues inteiros, eles podem ser entregues em módulos.

Atividade:

- pode ser dividida em atividades filhas.
- se não tem atividades filhas então deve ter um tempo de desenvolvimento. Caso contrário a duração da atividade é determinada através da duração das atividades filhas.
- tem precedentes, isto é, quais atividades devem ser desenvolvidas para que esta atividade seja implementada. O relacionamento de precedêcia não pode conter ciclos. Por exemplo:
Atividade A precede Atividade B
Atividade B precede Atividade C
Atividade C precede Atividade A
Esta relação de precedência não é permitida

Módulo:

- representação de como o projeto será entregue.
- contém as atividades que serão desenvolvidas em conjunto e entregues na mesma data.

1.7 Colaborações

- Antes de iniciar um módulo os precedentes das atividades dentro do módulo que estão contidos em outros módulos devem ter sido desenvolvidos.
- A atividade filha colabora com a atividade mãe de modo que o desenvolvimento de todas as atividades filhas representa o desenvolvimento da atividade mãe.

1.8 Consequências

O padrão prazo tem os seguintes benefícios e deficiências:

- Possibilita a determinação de um prazo coerente para a entrega de projetos. É muito difícil estimar com precisão o tempo de desenvolvimento de um projeto. Aplicando o padrão prazo divide-se o projeto em atividades elementares e determina-se o tempo de desenvolvimento destas atividades, tornando a estimativa de tempo total de desenvolvimento do projeto muito mais simples.

- Provê um meio de identificar as atividades que podem ser desenvolvidas em paralelo. Atividades que não tem inter-dependência podem ser desenvolvidas em paralelo. O desenvolvimento de atividades em paralelo torna possível, desde que hajam recursos suficientes, a diminuição do tempo total de desenvolvimento de um projeto.

- Consome tempo. Em muitos casos a determinação do prazo de desenvolvimento deve ser feita antes da aprovação do cliente. Se o cliente não aprovar o projeto, o tempo gasto em análise é desperdiçado. Além disso, a análise do prazo consome um tempo que poderia ser gasto em desenvolvimento.

1.9 Implementação

- Dividindo o projeto em Atividades. O tempo de desenvolvimento de uma atividade não pode ser subestimado. Caso o analista não tenha certeza da duração da atividade ela deve ser quebrada em atividades filhas mensuráveis.

- Definindo a duração das Atividades. Determinar o tempo de desenvolvimento de atividades elementares requer análise de uma pessoa experiente, que realmente saiba estimar com precisão tal tempo de desenvolvimento.

- Calculando o tempo de desenvolvimento total para o projeto. Deve-se tomar cuidado com a precedência das atividades. Se a Atividade A depende da Atividade B elas não podem ser executadas em paralelo. Por Exemplo:

Módulo 1:
Atividade A - duração 2 dias.
Atividade B - duração 3 dias.
Atividade A depende da Atividade B.

Se a precedência das atividades não for levada em consideração o analista poderia chegar a conclusão errada de que para desenvolver o módulo 1 seriam necessários 3 dias se as atividades fossem executadas em paralelo. Porém o tempo correto de desenvolvimento do módulo 1 é de 5 dias pois a atividade 2 só pode ser iniciada depois da atividade 1 ter sido concluída.

1.10 Exemplo de Uso

Este exemplo foi extraído do livro [PFL91] e cita as atividades necessárias para um projeto de construção de uma casa.

Atividade 1: Preparação do Local	
Atividade 1.1: Vistoria do Terreno	
Atividade 1.2: Obtenção de licenças/autorizações	
Atividade 1.3: Escavação	
Atividade 1.4: Obteção de Materiais	
Atividade 2: Construindo o Exterior	
Atividade 2.1: Fundação	
Atividade 2.2: Paredes Externas	
Atividade 2.3: Encanamento Externo	
Atividade 2.4: Sistema elétrico externo	
Atividade 2.5: Reboque Externo	Atividades
Atividade 2.6: Pintura Externa	
Atividade 2.7: Portas e acessórios	
Atividade 2.8: Telhado	
Atividade 3: Terminando o Interior	
Atividade 3.1: Encanamento interno	
Atividade 3.2: Sistema elétrico interior	
Atividade 3.3: Paredes internas	
Atividade 3.4: Pintura interior	
Atividade 3.5: Piso	
Atividade 3.6: Portas e acessórios	

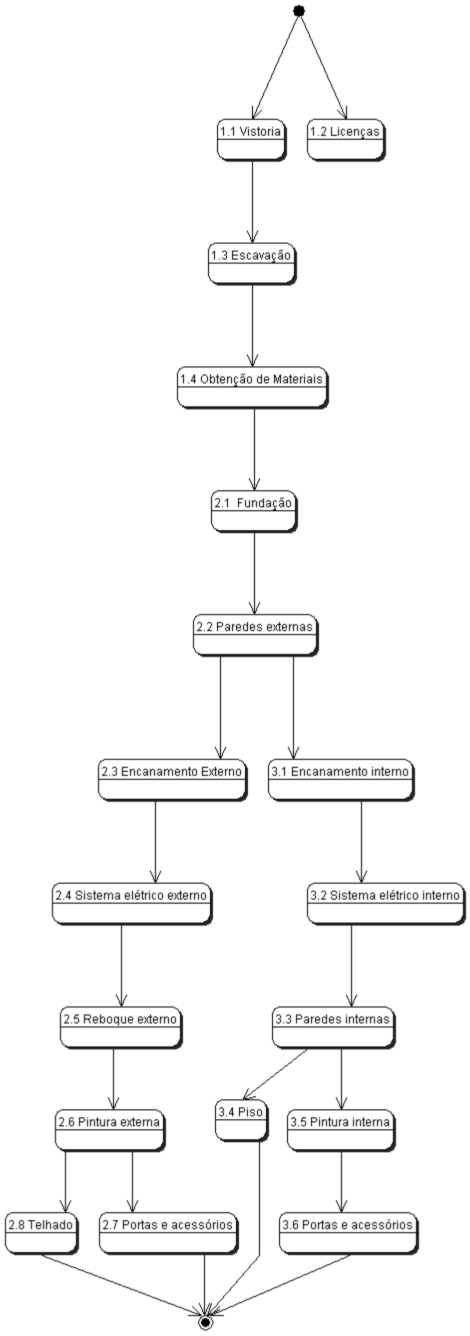

Figura 4: Atividades

Atividade 1: Preparação do Local	
Atividade 1.1: Vistoria do Terreno: 3 dias	
Atividade 1.2: Obtenção de licenças/autorizações: 15 dias	
Atividade 1.3: Escavação: 10 dias	
Atividade 1.4: Obteção de Materiais: 10 dias	
Atividade 2: Construindo o Exterior	
Atividade 2.1: Fundação: 15 dias	
Atividade 2.2: Paredes Externas: 20 dias	
Atividade 2.3: Encanamento Externo: 10 dias	
Atividade 2.4: Sistema elétrico externo: 10 dias	
Atividade 2.5: Reboque Externo: 8 dias	Cronograma
Atividade 2.6: Pintura Externa: 5 dias	
Atividade 2.7: Portas e acessórios: 6 dias	
Atividade 2.8: Telhado: 9 dias	
Atividade 3: Terminando o Interior	
Atividade 3.1: Encanamento interno: 12 dias	
Atividade 3.2: Sistema elétrico interior: 15 dias	
Atividade 3.3: Paredes internas: 9 dias	
Atividade 3.4: Pintura interior: 18 dias	
Atividade 3.5: Piso: 11 dias	
Atividade 3.6: Portas e acessórios: 7 dias	

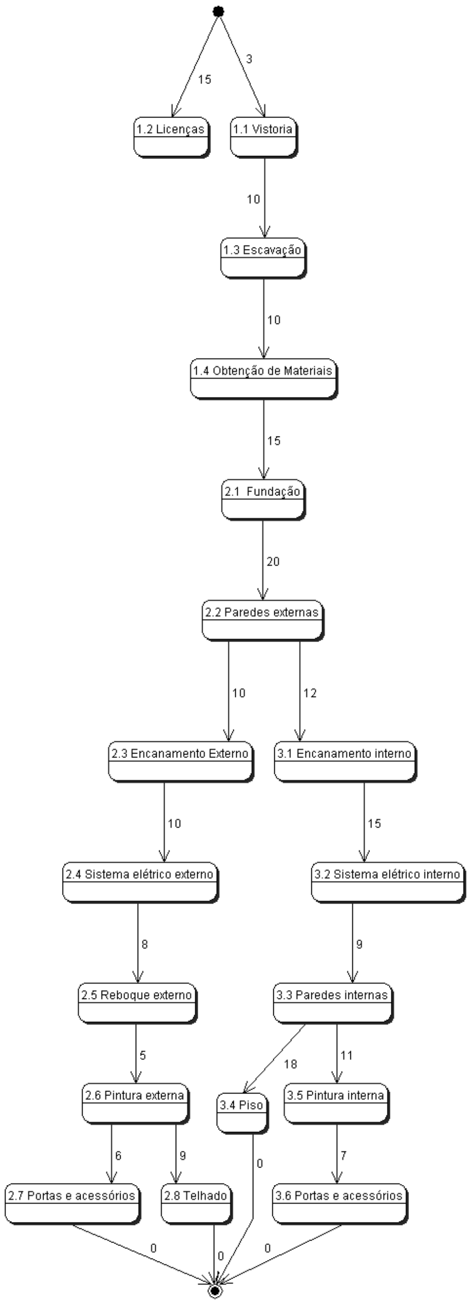

Figura 5: Cronograma de Atividades

O Prazo total do Projeto é dado pelo caminho mais longo no gráfico. No gráfico acima um dos caminho mais longos contém as seguintes atividades: 1.1, 1.3, 1.4, 2.1, 2.2, 3.1, 3.2, 3.3, 3.5 e 3.6, este caminho requer 54 dias. Portanto o projeto iria levar no mínimo 54 dias.

1.11 Usos Conhecidos

Este padrão ou variações do mesmo são utilizados em técnicas de Engenharia de Software.

2 Bibliografia

PFL91 Shari Lawrence Pfleeger, "Software Engineering - The Production of Quality Software, Second Edition", (Prentice Hall 1991)

GHJV E. Gamma, R. Helm, R. Johnson, J. Vlissides "Padrões de Projeto - Soluções Reutilizáveis de Softwares Orientado a Objetos", (Tradução Luiz A. Meirelles Salgado - Bookman 2000)