

UML

METODOLOGIAS E FERRAMENTAS

CASE

Volume 1

2ª Edição, revista e actualizada para o UML 2



CENTROATLANTICO.PT

Alberto Silva e Carlos Videira

ALBERTO MANUEL RODRIGUES DA SILVA

CARLOS ALBERTO ESCALEIRA VIDEIRA

UML, METODOLOGIAS E FERRAMENTAS CASE

2ª EDIÇÃO, VOL. I

Linguagem de Modelação UML, Metodologias e Ferramentas CASE na
Concepção e Desenvolvimento de Sistemas de Informação



CENTRO ATLANTICO .PT

Portugal/2005

Reservados todos os direitos por Centro Atlântico, Lda.

Qualquer reprodução, incluindo fotocópia, só pode ser feita com autorização expressa dos editores da obra.

UML, METODOLOGIAS E FERRAMENTAS CASE - 2ª EDIÇÃO, VOL. I

Colecção: Tecnologias

Autores: Alberto Manuel Rodrigues da Silva

Carlos Alberto Escalreira Videira

Direcção gráfica: Centro Atlântico

Revisão final: Centro Atlântico

Capa: Paulo Buchinho

© Centro Atlântico, Lda., 2005

Av. Dr. Carlos Bacelar, 968 – Escr. 1 – A

4764-901 V. N. Famalicão

Rua da Misericórdia, 76 – 1200-273 Lisboa

Portugal

Tel. 808 20 22 21

geral@centroatlantico.pt

www.centroatlantico.pt

Impressão e acabamento: Inova

1ª edição: Maio de 2005

ISBN: 989-615-009-5

Depósito legal: 226.829/05

Marcas registadas: Todos os termos mencionados neste livro conhecidos como sendo marcas registadas de produtos e serviços foram apropriadamente capitalizados. A utilização de um termo neste livro não deve ser encarada como afectando a validade de alguma marca registada de produto ou serviço.

O Editor e os Autores não se responsabilizam por possíveis danos morais ou físicos causados pelas instruções contidas no livro nem por endereços Internet que não correspondam às *Home-Pages* pretendidas.

Aos meus pais, António e Ermelinda

Alberto Rodrigues da Silva

À Elsa, Sofia, Guilherme e Mafalda

Carlos Videira

Prefácio à Segunda Edição

Quatro anos decorridos após a primeira edição do nosso livro “UML, Metodologias e Ferramentas CASE” tem sido constante a evolução nesta área da engenharia! Temas como o paradigma do desenvolvimento de sistemas de software baseado em modelos, o aparecimento de novas metodologias de desenvolvimento que pretendem reduzir o tempo dos projectos e a importância crescente atribuída à modelação do negócio e das organizações (*business modeling*) são apenas dois exemplos de actuais áreas de actuação e de investimento. No entanto a principal motivação para esta nova edição resulta da recente publicação de uma nova versão do UML (2.0), com algumas inovações importantes.

A segunda edição deste livro procura minimizar as alterações aos principais objectivos e à estrutura, relativamente à primeira edição. No entanto, é inevitável a introdução de novos temas, o refinamento de outros e a actualização de algumas matérias que entretanto sofreram evolução, ou que decorrem da natural evolução e maturidade dos autores. De entre as matérias que entretanto surgiram destaca-se um capítulo dedicado ao tema da modelação de dados (*data modeling*) em UML, nas suas diferentes representações ao nível conceptual, lógico e físico, e nas regras de mapeamento dos modelos UML para esquemas relacionais e DDL/SQL. No entanto, a principal alteração prende-se com a divisão do livro em dois volumes, o primeiro com as partes 1 e 2 e o segundo com as partes 3 e 4; esta divisão vem de encontro a alguns comentários que fomos recebendo, e tem também por objectivo separar matérias que podem interessar a públicos alvo distintos.

As matérias que sofreram alterações mais significativas incluem todos os capítulos da Parte 2 (que envolve a apresentação da linguagem UML), derivado ao facto de se ter realizado um esforço de actualização alinhado com a apresentação da versão 2 do UML. A evolução do UML 1.4 para o UML 2 produziu fortes alterações ao nível da sua própria arquitectura, um refinamento e aumento de qualidade da generalidade dos diagramas, e implicou também a introdução de novos diagramas. Adicionalmente, e na resposta às sugestões de alguns dos nossos leitores, utilizaremos um exemplo comum para suportar a explicação dos conceitos debatidos ao longo de toda a Parte 2. Em relação ao segundo volume, o capítulo referente ao RUP (*Rational Unified Process*) foi também substancialmente alterado, tendo em consideração a sua importância e significativas evoluções ao longo deste intervalo de tempo.

Finalmente, houve da parte dos autores e da editora um maior cuidado na verificação da produção final do livro, em particular nas características de impressão!

Lisboa, Março de 2005

Alberto Manuel Rodrigues da Silva

Carlos Alberto Escaleira Videira

Prefácio

Objectivos, Contexto e Motivação

O livro “UML, Metodologias e Ferramentas CASE” aborda tópicos importantes para a generalidade dos intervenientes nas actividades enquadradas na engenharia de software, designadamente as problemáticas (1) das linguagens de modelação de software, (2) do processo e das metodologias de desenvolvimento de software, e (3) das ferramentas CASE de suporte à modelação e ao próprio desenvolvimento. Pretende dar uma panorâmica abrangente sobre estes três aspectos de forma integrada e coerente. Embora o foco do livro seja nas fases de concepção de sistemas de software, discute o seu enquadramento de modo mais lato em áreas como o planeamento estratégico de sistemas de informação; as arquitecturas de sistemas de informação; ou mesmo a engenharia de software.

O livro explica a necessidade da modelação no desenvolvimento de software, o que é o UML (*Unified Modeling Language*), como aplicar o UML no contexto mais abrangente das metodologias e processos de desenvolvimento, e como usar ferramentas CASE de forma a maximizar e automatizar algumas das tarefas relacionadas com a modelação, por exemplo, produção e gestão de documentação, geração de código, geração de esquemas de dados, *reverse engineering*, *round-trip engineering*, mecanismos de extensão, etc.

A aprendizagem e adopção dos temas abordados neste livro constituem uma vantagem decisiva para os intervenientes que os adoptarem consistentemente. Entre outros, salientamos os seguintes benefícios: melhor documentação dos sistemas e dos respectivos artefactos; aplicação de técnicas de modelação orientadas por objectos, mais fáceis de entender; reutilização desde as fases preliminares da concepção até à implementação; rastreabilidade dos requisitos ao longo de todo o processo; facilidade de comunicação entre todos os intervenientes envolvidos no processo; melhorias significativas em factores como sejam flexibilidade e produtividade; melhor gestão de requisitos; avaliação e manutenção de sistemas mais facilitadas. Estas características são naturalmente interdependentes entre si; por exemplo, uma maior qualidade da documentação produzida possibilita uma melhor comunicação entre os intervenientes de um projecto.

Todavia, os assuntos tratados neste livro são difíceis de adoptar nas organizações, por inúmeras razões. Antes de mais porque o ritmo de inovação tecnológica nesta área da engenharia tem-se processado a um ritmo particularmente intenso.

A segunda razão deve-se ao facto dos tópicos abordados neste livro exigirem uma formação significativa e principalmente uma adequada e correspondente actuação. Não basta dominar um conjunto alargado de conceitos e notações para especificar software, mas é fundamental aprender a aplicá-los de forma consistente, repetida e sistemática; adaptá-los às condicionantes e realidades de cada empresa, ou de cada projecto em particular; e ainda partilhar técnicas e métodos entre todos os indivíduos da empresa, ou de cada projecto, para que a comunicação entre todos os intervenientes seja maximizada e eficiente.

A terceira razão, consequência das razões anteriormente referidas, é o facto de ser oneroso a adopção efectiva e produtiva (dos tópicos abordados neste livro) no seio das empresas. Oneroso em termos do tempo inicial que é necessário despendido em formação, em termos da “resistência à mudança”, assim como o investimento necessário na selecção e aquisição de ferramentas CASE que potenciem significativamente as suas vantagens.

Este livro surge na sequência da experiência dos autores em actividades de investigação, mas principalmente em actividades de consultoria e de docência nas áreas de engenharia de software e de sistemas de informação.

Os temas abordados neste livro são na sua maioria influenciados pelo trabalho de unificação e de evangelização dos “três amigos”: Grady Booch, Ivar Jacobson e James Rumbaugh. Todavia, é da nossa exclusiva responsabilidade o estilo do livro, assim como a sua estrutura, conteúdo, exemplos e exercícios propostos (tal como as correspondentes gralhas e omissões decorrentes!). O livro condensa e integra informação dispersa por alguns livros da área, em particular os seguintes títulos: *OMG Unified Modeling Language Specification* [OMG99], *The Unified Modeling Language User Guide* [Booch99], *The Unified Software Development Process* [Jacobson99], *Visual Modeling with Rational Rose 2000 and UML* [Quatrani00] e *The Rational Unified Process* [Kruchten00]. No entanto, há inúmeros aspectos que o livro propõe e discute de forma única, dificilmente encontrados em qualquer dos livros referidos.

A nível internacional, existe um número relevante de títulos nesta área; contudo, há reconhecidamente, na língua Portuguesa uma lacuna muito significativa. Paralelamente, e em consequência da nossa experiência e responsabilidade de docência, supervisão e coordenação de trabalhos finais de curso e de investigação identificamos a necessidade e oportunidade de produzirmos este livro com vista a apoiar a aprendizagem da engenharia de software nos tópicos referidos.

A temática tratada neste livro é abrangente e a sua profundidade é, propositadamente, de nível intermédio. Inúmeros assuntos poderão ser analisados e aprofundados complementarmente, entre os quais destacam-se a título de exemplo os seguintes: arquitecturas de sistemas de software [Hofmeister99]; processos de negócio em contextos organizacionais [Penker00]; padrões de análise [Fowler96]; padrões de desenho em infra-estruturas de software (*frameworks*) [Souza99]; modelação de dados [Muller00]; modelação de aplicações segundo o paradigma dos agentes de software [Odell00], modelação de aplicações de tempo real [Selic94], ou modelação de aplicações interactivas [Nunes99]. Todos estes tópicos são importantes nos seus respectivos contextos de aplicação; muitos são alvo de intensa actividade de estudo e investigação. Todos eles apresentam, contudo, um denominador comum: baseiam-se no conhecimento introduzido, apresentado e discutido neste livro.

Audiência do Livro

O livro pretende servir como referência de suporte a um número restrito de disciplinas de nível de ensino superior na área de sistemas de informação. Consequentemente, o livro adopta um estilo tendencialmente pedagógico através da apresentação e discussão de exemplos, da narrativa de histórias e factos reais, ou pela proposta de exercícios académicos.

O primeiro perfil de leitores deste livro vai directamente para os alunos de licenciatura e de cursos de pós-graduação em engenharia informática ou em informática de gestão. Pressupõe-se que os leitores já “sabem” implementar aplicações informáticas; e que neste livro procuram aprender a reflectir sobre o processo de desenvolvimento de software, e aprender técnicas e práticas consistentes e sistemáticas para o realizar.

Adicionalmente, este livro é relevante para um número mais alargado de leitores, em particular para investigadores, gestores informáticos, responsáveis pelo processo de desenvolvimento de software, analistas-programadores, e outros que necessitem de especificar de forma mais ou menos detalhada sistemas de software.

O livro pressupõe um conjunto de pré-requisitos que o leitor deverá possuir para o poder usufruir devidamente. É suposto o leitor possuir um conhecimento razoável sobre as bases da informática e dos sistemas de computadores, tais como noções essenciais de programação, de bases de dados e de sistemas operativos.

Organização do Livro

A segunda edição do livro encontra-se organizada em 2 volumes, 4 partes, 14 capítulos e 2 apêndices conforme se resume de seguida. No primeiro volume incluem-se as 2 primeiras partes e no segundo as restantes; os apêndices são incluídos em ambos os volumes.

A Parte 1 (INTRODUÇÃO E VISÃO GERAL) apresenta os conceitos gerais, visão histórica e enquadramento da realização deste livro. Inclui os capítulos 1, 2 e 3.

A Parte 2 (LINGUAGEM DE MODELAÇÃO UML) é constituída por 7 capítulos complementares, sendo que o Capítulo 4 dá a visão histórica e geral do UML e o Capítulo 9 descreve sucintamente alguns aspectos considerados “avançados”, não essenciais para o leitor que apenas pretende usar e aplicar as características básicas do UML. Os restantes capítulos (Capítulos 5, 6, 7 e 8) constituem o centro desta parte do livro e deverão ser lidos de forma sequencial conforme proposto. Adicionalmente, o Capítulo 10 discute vários aspectos relacionados com a problemática de modelação de dados aos seus diferentes níveis, em UML; e descreve regras de mapeamento de modelos de dados de uma representação de alto nível (e.g., em UML) para uma representação mais próxima do desenho físico das bases de dados.

A Parte 3 (METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE) apresenta a problemática geral das metodologias e processos de desenvolvimento de software, com exemplos concretos baseados em duas propostas reais de metodologias, o RUP e o ICONIX, descritos respectivamente nos Capítulos 11 a 12.

A Parte 4 (FERRAMENTAS CASE) apresenta a problemática das ferramentas CASE descrevendo o seu significado, evolução histórica e discutindo mecanismos de caracterização e avaliação (Capítulo 13). São apresentadas e analisadas duas ferramentas CASE, o Rose da IBM (empresa que entretanto adquiriu a Rational) e o System Architect da Popkin, respectivamente nos Capítulos 14 e 15.

No Apêndice A (“Guia de Recursos Electrónicos”) apresenta-se de modo classificado um conjunto significativo de recursos electrónicos sobre os temas abordados neste livro.

No Apêndice B (“Glossário, Siglas e Abreviaturas”) apresentam-se três tabelas com informação relativa ao glossário, as siglas, e as abreviaturas adoptadas ao longo de todo o livro.

Em “Referências” listam-se, por ordem alfabética, todas as referências bibliográficas utilizadas ao longo do livro.

Por fim, inclui-se o “Índice Remissivo” que constitui um mecanismo típico de trabalho e de consulta neste género de literatura.

Notação Adoptada

Ao longo do livro são adoptadas genericamente as seguintes regras de notação textual:

- Nomes e expressões em inglês são escritas em itálico. As excepções são expressões vulgarmente adoptadas para o Português (e.g., software, bit), expressões intensamente usadas ao longo do texto (e.g., Internet, Web, applet, standard), ou nomes de empresas ou produtos de origem anglo-saxónica (e.g., MS-Word, Rational Rose).
- Frases e expressões que se pretendam destacar são escritas com ênfase (i.e., negrito).
- Exemplos de código, pseudocódigo, nomes de classes, ou endereços electrónicos são apresentados numa fonte de tamanho fixo (i.e., Courier).

Os exemplos apresentados neste livro aparecem enquadrados por uma moldura correspondente, conforme ilustrado neste mesmo parágrafo.

Conceito

Há ao longo do livro um cuidado particular na introdução dos inúmeros conceitos que o mesmo analisa e discute. De forma a facilitar a identificação desses conceitos, colocamos na margem esquerda do respectivo texto a marca visual “Conceito” conforme apresentado neste parágrafo. Recomenda-se ao leitor a utilização do índice remissivo para consultar a definição de qualquer dos conceitos tratados neste livro.

Por fim, relativamente à representação de diagramas será utilizada, sempre que for adequado, e por razões óbvias, a linguagem UML.

Agradecimentos

Um agradecimento muito especial à minha família por todo o amor e suporte que tive para poder realizar mais este trabalho, bem como pelas inúmeras horas roubadas ao seu convívio.

Um agradecimento também aos colegas do Judo Clube Portugal e outros amigos cujo convívio me proporcionou os momentos de relaxamento necessário para a produção deste livro.

Parte significativa da actividade que conduziu à realização deste livro foi desenvolvida no âmbito de duas instituições que procuram a excelência da sua actividade, o Departamento de Engenharia Informática do Instituto Superior Técnico e o INESC-ID, às quais não posso deixar de endereçar o meu expresso agradecimento, bem como a todos os colegas e alunos com quem tive o privilégio de conviver, aprender e ensinar durante este período. Em particular, aos alunos da primeira e segunda edição da Pós-Graduação em Sistemas de Informação (POSI'1999 e POSI'2000) do Instituto Superior Técnico, com os quais ensaiei e testei uma parte preliminar deste livro; ao núcleo organizativo do POSI, nomeadamente aos Prof. José Tribolet e Prof. Paulo Guedes, pelo convite que me endereçaram; e ao meu monitor desses cursos, Eng. Miguel Goulão, com quem discuti alguns dos tópicos e exemplos apresentados.

Um agradecimento à editora Centro Atlântico, na pessoa do Dr. Libório Manuel Silva, pelo seu interesse imediato na publicação do livro e pela sua activa e persistente atitude de estar no nosso pequeno mercado nacional de literatura técnico-científica.

Por fim, um agradecimento a todos os colegas que de uma forma ou outra sugeriram, comentaram ou apenas criticaram partes preliminares deste trabalho, ou com quem simplesmente fui partilhando a “ideia” do livro.

Alberto Silva

Quero em primeiro lugar agradecer à minha família, pela sua dedicação, carinho e apoio incondicional, sem a comunicação da qual dificilmente teria participado neste projecto. Quero também agradecer aos meus amigos, de cujo convívio tive que prescindir para poder completar este livro.

Um factor decisivo para a minha participação neste livro foi a experiência como docente, especialmente na Universidade Autónoma de Lisboa, onde tenho estado ligado a disciplinas relacionadas com os temas abordados neste livro. Nesse sentido, gostaria de agradecer ao Prof. José Luís Ferreira e ao Eng. Miguel Gonçalves toda a colaboração e incentivo que me têm dado. Um agradecimento particular a todos os alunos das várias disciplinas que leccionei, pois o esforço de preparação das mesmas contribuiu para a evolução do conteúdo de uma parte significativa deste livro.

Um agradecimento também para outros colegas com quem mantive, ao longo destes meses de trabalho, uma permuta de opiniões e críticas que me ajudaram a melhorar a qualidade da presente obra.

Finalmente, à editora Centro Atlântico e ao Dr. Libório Manuel Silva deixo um agradecimento pelo seu interesse na publicação desta obra técnico-científica, valorizando a missão de educar para o futuro.

Carlos Videira

Lisboa, Março de 2005

Alberto Manuel Rodrigues da Silva

Carlos Alberto Escalreira Videira

Contactos

A maior parte dos comentários, sugestões gerais ou pedidos de aquisição de exemplares deverão ser dirigidas à editora Centro Atlântico (via www.centroatlantico.pt ou geral@centroatlantico.pt).

Comentários técnicos, sugestões ou pedidos de esclarecimentos podem ser dirigidos directamente aos autores via correio electrónico: Alberto Rodrigues da Silva (Alberto.Silva@acm.org) e Carlos Videira (cvideira@acm.org).

Encontram-se disponíveis para consulta e *download*, em formato electrónico, partes complementares deste livro, nomeadamente: o seu resumo, índice, o currículo abreviado dos autores e a lista de exercícios propostos, em <http://www.centroatl.pt/titulos/tecnologias/uml2-vol1.php3>.

Autores

Alberto Manuel Rodrigues da Silva é professor do Departamento de Engenharia Informática (DEI) do IST/UTL, investigador sénior do INESC-ID, e consultor da empresa SIQuant – Engenharia do Território e Sistemas de Informação. É doutor e mestre em Engenharia Informática e Computadores pelo IST/UTL, e licenciado em Engenharia Informática pela FCT/UNL. Lecciona actualmente cadeiras da área de Sistemas de Informação de nível licenciatura, pós-graduação e mestrado. Supervisiona a realização de vários trabalhos finais de curso, teses de mestrado e de doutoramento. Tem interesses profissionais e científicos em sistemas de informação dinâmicos e distribuídos em larga escala; engenharia de requisitos, desenvolvimento baseado em modelos, processos e ferramentas de sistemas de informação; e negócios suportados electronicamente. No INESC-ID é membro e co-fundador do Grupo de Sistemas de Informação e do Laboratório de Ambientes Virtuais e Cooperativos. Actualmente desempenha os seguintes cargos de gestão: Coordenador do Grupo do Sistemas de Informação do INESC-ID; Coordenador dos programas de Mestrado e de Doutoramento em Engenharia Informática e Computadores do IST/UTL; e Vogal da Região Sul do Colégio de Engenharia Informática da Ordem dos Engenheiros. É autor e co-autor de 3 livros técnicos e cerca de 50 artigos científicos em revistas, conferências e workshops nacionais e internacionais.

Carlos Alberto Escaleira Videira é actualmente consultor na área de sistemas de informação, e assistente no Departamento de Ciências e Tecnologias da UAL. Desempenhou funções de coordenação na área de Informática em diferentes empresas e participou em diversos projectos como consultor. Tem um mestrado em Engenharia Electrotécnica e Computadores pelo IST/UTL e uma licenciatura em Engenharia Informática pela FCT/UNL. Lecciona actualmente disciplinas de Planeamento de Sistemas de Informação, Engenharia de Software, Qualidade de Sistemas de Informação e Negócios Electrónicos de nível de licenciatura e pós-graduação. Tem interesses profissionais e científicos em temas relacionados com Planeamento Estratégico de Sistemas de Informação, Engenharia de Software, Sistemas de Informação, Gestão de Projectos e Negócios Electrónicos.

Índice

Prefácio à Segunda Edição _____	VII
Prefácio _____	IX
Índice _____	XIX

PARTE 1 – INTRODUÇÃO E VISÃO GERAL _____ **1**

Capítulo 1 - Enquadramento e Conceitos Gerais _____	3
1.1 Introdução _____	3
1.2 O Impacto das Tecnologias de Informação _____	4
1.3 Produto e Processo _____	6
1.4 Sistemas de Informação _____	7
1.5 Arquitectura de Sistemas de Informação _____	9
1.6 Objectivos do Desenvolvimento de Sistemas Informação _	12
1.7 Problemas no Desenvolvimento de Sistemas Informação _	14
1.8 Planeamento Estratégico de Sistemas de Informação ____	17
1.9 Engenharia de Software _____	18
1.10 Conclusão _____	20
1.11 Exercícios _____	22
Capítulo 2 - O Processo de Desenvolvimento de Software ____	23
2.1 Introdução _____	23
2.2 Processos e Metodologias _____	24
2.3 Modelos e Modelação _____	27
2.3.1 Importância da Modelação _____	28
2.3.2 Princípios da Modelação _____	29
2.4 Boas Práticas no Desenvolvimento de Software _____	30

2.5	Fases do Processo de Desenvolvimento de Software	32
2.5.1	Tarefas Transversais	37
2.5.2	Planeamento	38
2.5.3	Análise	40
2.5.4	Desenho	42
2.5.5	Implementação	42
2.5.6	Testes	43
2.5.7	Instalação	45
2.5.8	Manutenção	46
2.6	Processos de Desenvolvimento de Software	47
2.6.1	Processos em Cascata	47
2.6.2	Processos Iterativos e Incrementais	50
2.7	Conclusão	52
2.8	Exercícios	54

Capítulo 3 - Evolução das Metodologias de Desenvolvimento de Software _____ **55**

3.1	Introdução	55
3.2	A Programação como Fonte de Inovação	56
3.3	O Desenvolvimento Ad-Hoc	60
3.4	As Metodologias Estruturadas	62
3.4.1	Contexto e Motivação	62
3.4.2	Conceitos Básicos	63
3.4.3	Técnicas e Notações mais Utilizadas	65
3.4.4	Principais Metodologias	68
3.5	Metodologias Orientadas por Objectos	70
3.5.1	Contexto e Motivação	72
3.5.2	Conceitos Básicos	72
3.5.3	Técnicas e Notações mais Utilizadas	80
3.5.4	Principais Metodologias	81
3.6	Outras Abordagens	83
3.6.1	Metodologias Ágeis	83
3.6.2	MDA, Desenvolvimento Baseado em Modelos	84
3.7	Comparação de Metodologias	85
3.7.1	Gestão de Requisitos e Facilidade de Manutenção	86
3.7.2	Representação da Realidade	87
3.7.3	Outros Aspectos	88
3.8	Conclusão	89
3.9	Exercícios	90

PARTE 2 – LINGUAGEM DE MODELAÇÃO UML ___ **93**

Capítulo 4 - UML – Visão Geral	101
4.1 Introdução	101
4.2 Visão Histórica	103
4.3 Tipos de Elementos Básicos	106
4.4 Tipos de Relações	107
4.5 Tipos de Diagramas Principais	108
4.5.1 Diagramas de Casos de Utilização	109
4.5.2 Diagramas de Modelação da Estrutura	110
4.5.3 Diagramas de Modelação do Comportamento	110
4.5.4 Diagramas de Arquitectura	113
4.6 Mecanismos Comuns do UML	114
4.6.1 Notas (ou Anotações)	114
4.6.2 Mecanismos de Extensão	115
4.6.3 Tipos de Dados	117
4.7 Organização dos Artefactos - Pacotes	118
4.7.1 Representação Gráfica	118
4.7.2 Relações entre Pacotes	119
4.7.3 Tipos de Pacotes	122
4.7.4 Modelação de Grupos de Elementos	123
4.8 Molduras e Fragmentos	124
4.9 Aspectos Específicos do UML 2	126
4.9.1 Objectivos do UML 2	126
4.9.2 Diagramas do UML 2	128
4.10 Exercícios	130
Capítulo 5 - UML – Casos de Utilização	131
5.1 Introdução	131
5.2 Casos de Utilização	133
5.2.1 Casos de utilização e Cenários	134
5.2.2 Relações entre Casos de Utilização	136
5.3 Diagramas de Casos de Utilização	142
5.3.1 Actores	143
5.3.2 Casos de Utilização Abstractos e Concretos	144
5.4 Proposta de Metodologia	144
5.5 Exemplos e Recomendações	147
5.6 Exercícios	152

Capítulo 6 - UML – Modelação da Estrutura	155
6.1 Introdução	155
6.2 Classes	156
6.2.1 Declaração de Atributos	157
6.2.2 Declaração de Operações	159
6.3 Relações	161
6.3.1 Relação de Dependência	161
6.3.2 Relação de Generalização	164
6.3.3 Relação de Associação	165
6.4 Interfaces	173
6.5 Instâncias e Objectos	178
6.6 Diagramas de Classes e Diagramas de Objectos	181
6.7 Exemplos e Recomendações	181
6.8 Exercícios	187
Capítulo 7 - UML – Modelação do Comportamento	191
7.1 Introdução	191
7.2 Interacções	192
7.2.1 Objectos e Ligações	193
7.2.2 Mensagens	194
7.2.3 Representação Gráfica de Mensagens	195
7.2.4 Tipos de Mensagens	196
7.3 Diagramas de Interação	197
7.3.1 Diagramas de Sequência	198
7.3.2 Diagramas de Comunicação	200
7.3.3 Equivalência Semântica	203
7.3.4 Diagramas de Interação e Casos de Utilização	205
7.3.5 Reutilização de Interações	206
7.3.6 Diagrama de Visão Geral da Interação	207
7.3.7 Combinação de Fragmentos de Interação	208
7.3.8 Diagrama Temporal	210
7.4 Diagrama de Estados	212
7.4.1 Estados	213
7.4.2 Transições	214
7.4.3 Eventos	216
7.4.4 Acções e Actividades	217
7.4.5 Ordem da realização de operações	218
7.4.6 Sub-Estados	219
7.4.7 Outros Aspectos dos Diagramas de Estados	221

7.5	Diagramas de Actividade	222
7.5.1	Decisões	224
7.5.2	Caminhos Concorrentes	225
7.5.3	Partições de Actividades	225
7.5.4	Actividades e Objectos	229
7.5.5	Utilizações Típicas	230
7.5.6	Outros Aspectos dos Diagramas de Actividade	235
7.6	Exercícios	236
Capítulo 8 - UML – Modelação da Arquitectura		241
8.1	Introdução	241
8.2	Componentes e Nós	242
8.2.1	Componentes no UML 1	242
8.2.2	Componentes no UML 2	244
8.2.3	Nós	246
8.2.4	Relações entre Nós e Componentes	247
8.3	Diagramas de Componentes	248
8.4	Diagramas de Instalação	251
8.5	Exercícios	255
Capítulo 9 - UML – Aspectos Avançados		259
9.1	Introdução	259
9.2	A Arquitectura do UML	260
9.2.1	A Estrutura do UML a Quatro Camadas	260
9.2.2	A Camada Metamodelo, UML 1.x	262
9.2.3	Mecanismos de Extensão, UML 1.x	265
9.2.4	A Camada Metamodelo, UML 2	268
9.3	Perfis UML	271
9.3.1	Referências a Perfis UML	272
9.3.2	Perfil para Processos de Desenvolvimento Software	273
9.3.3	Perfil para Modelação de Negócios	277
9.3.4	Perfil para Modelação de Aplicações Web	278
9.4	Sistemas de Componentes e Reutilização	280
9.4.1	Definição de Componente	280
9.4.2	Famílias de Aplicações	281
9.4.3	Sistemas de Componentes	281
9.4.4	Reutilização	283

9.5	Tipos Parametrizáveis	284
9.5.1	Classes Parametrizáveis	285
9.5.2	Padrões de Desenho	286
9.6	XMI (XML Metadata Interchange) e Diagram Interchange	290
9.7	Conclusão	291
9.8	Exercícios	293
Capítulo 10 - UML – Modelação de Dados		295
10.1	Introdução	295
10.2	Processo de Desenho de Bases de Dados com o UML	297
10.2.1	Nível Conceptual	299
10.2.2	Nível Lógico	300
10.2.3	Nível Físico	301
10.3	Regras de Mapeamento de Modelos Lógicos UML em Esquemas Relacionais	303
10.3.1	Paradigma Relacional vs. Orientado por Objectos	303
10.3.2	Classes em Relações e Objectos em Tuplos	305
10.3.3	Identificação de Objectos em Chaves Primárias	306
10.3.4	Mapeamento de Relações de Associação	306
10.3.5	Mapeamento de Relações de Generalização	311
10.3.6	Síntese das Regras de Mapeamento	314
10.4	Exemplos de Aplicação	316
10.5	Exercícios	321
Apêndice A – Guia de Recursos Electrónicos		323
	Standards, Org. Normalizadoras, Iniciativas e Forums	325
	Empresas e Links Relevantes	325
	Leituras Recomendadas	325
	Jornais e Catálogos de Informação	326
	Ferramentas CASE	326
Apêndice B – Glossário, Siglas e Abreviaturas		329
B.1	Glossário	330
B.2	Siglas mais Usadas	331
B.3	Abreviaturas	332

Apêndice C – Elementos UML Predefinidos	333
C.1 Palavras Reservadas	333
C.2 Estereótipos Standard	337
Referências	341
Índice Remissivo	351

Parte 1 – Introdução e Visão Geral

Uma empresa de software de sucesso é aquela que consistentemente produz software de qualidade que vai ao encontro das necessidades dos seus utilizadores. Uma empresa que consegue desenvolver tal software, de forma previsível, cumprindo os prazos, com uma gestão de recursos, quer humanos, quer materiais, eficiente e eficaz, é uma empresa que tem um negócio sustentado.

Grady Booch, James Rumbaugh, Ivar Jacobson. *The Unified Modeling Language User Guide.*

Fazer software não é uma tarefa fácil. Fazer software de qualidade é ainda mais difícil. A generalidade dos resultados obtidos ao longo do tempo tem sistematicamente apresentado padrões de baixa qualidade, de custos e prazos completamente ultrapassados. Neste aspecto, a indústria de software deve ser caso único na sociedade actual, pois apesar da taxa de sucesso dos projectos ser relativamente baixa, o interesse das organizações pelo desenvolvimento de sistemas informáticos tem aumentado constantemente, não se vislumbrando qualquer alternativa. Tudo isto porque as organizações reconhecem que o recurso informação é estratégico e fonte de vantagens competitivas importantes.

O facto dos resultados dos projectos informáticos estarem normalmente abaixo das expectativas, e os diversos problemas que de forma consistente vêm ocorrendo desde o início da utilização das tecnologias de informação, torna extremamente relevantes as várias iniciativas que possam ser desenvolvidas com o objectivo de ultrapassar estes problemas. Por isso, vale a pena analisar os diversos esforços que foram efectuados ao longo do tempo, e perceber por que alguns não foram totalmente efectivos na resolução dos problemas, enquanto outros, bem sucedidos, são apontados como melhores práticas a aplicar sistematicamente.

Esta primeira parte do livro pretende dar um enquadramento das questões relacionadas com o desenvolvimento de software, de forma a “aguçar o apetite” dos leitores para os capítulos subsequentes do livro, onde são apresentadas várias ideias, técnicas, métodos e ferramentas que os autores deste livro acreditam que poderão desempenhar um papel decisivo na melhoria dos diversos problemas referidos na primeira parte.

Organização da Parte 1

O Capítulo 1, “Enquadramento e Conceitos Gerais”, faz o enquadramento e define o âmbito do livro em questões mais vastas relacionadas com as tecnologias de informação, de forma a transmitir a mensagem ao utilizador que há questões importantes relacionadas com o desenvolvimento de software cuja resolução passa pela realização de actividades e aplicação de técnicas que serão discutidas nos restantes capítulos, enquanto outras saem fora do âmbito deste livro. Apresenta ainda os problemas que os sistemas de informação enfrentam actualmente e algumas definições que são relevantes para a compreensão do livro.

O Capítulo 2, “O Processo de Desenvolvimento de Software”, pretende fornecer ao leitor uma visão geral sobre as actividades relacionadas com o desenvolvimento de software, nomeadamente sobre a sua organização, sequência e objectivos a atingir. São ainda clarificados alguns conceitos relacionados com as etapas do desenvolvimento de software.

O Capítulo 3, “Evolução das Metodologias de Desenvolvimento de Software”, procura dar uma visão histórica de como o desenvolvimento de software foi encarado ao longo do tempo, na perspectiva da aplicação de metodologias e respectivas técnicas, e quais as principais motivações para os diversos saltos qualitativos que ocorreram.

Capítulo 1 - ENQUADRAMENTO E CONCEITOS GERAIS

Tópicos

- Introdução
- O Impacto das Tecnologias de Informação
- Produto e Processo
- Sistemas de Informação
- Arquitectura de Sistemas de Informação
- Objectivos do Desenvolvimento de Sistemas de Informação
- Problemas no Desenvolvimento de Sistemas de Informação
- Planeamento Estratégico de Sistemas de Informação
- Engenharia de Software
- Conclusão
- Exercícios

1.1 Introdução

O principal objectivo deste livro é apresentar a linguagem de modelação UML (Parte 2) e demonstrar a sua aplicação de forma a facilitar todo o desenvolvimento de software, quer seja directamente como técnica de modelação de software, quer seja na sua utilização em metodologias de desenvolvimento (Parte 3) ou em ferramentas de apoio (Parte 4).

Actualmente é já indiscutível que o UML representa um papel relevante no desenvolvimento de software. Consideramos por isso importante enquadrar o leitor

deste livro nos principais problemas, objectivos e conceitos relacionados com os sistemas de informação e com o seu desenvolvimento. Neste primeiro capítulo, esta abordagem será efectuada de forma ainda muito genérica, e será concretizada nos dois capítulos seguintes.

É ainda importante que o leitor compreenda a relevância de outros conceitos e actividades, que devem ser aplicados no âmbito dos sistemas de informação, mas que não se encontram no âmbito deste livro; estamos a falar, por exemplo, das noções de arquitectura de sistemas de informação e do planeamento estratégico de sistemas de informação. São áreas que estão ao nível da concepção de sistemas de informação, com preocupações de natureza estratégica e que apenas serão brevemente equacionadas neste livro.

1.2 O Impacto das Tecnologias de Informação

É hoje em dia lugar comum ouvir-se falar da importância que a informática ocupa na nossa vida. O impacto e a rápida evolução ao longo dos últimos 40 anos das tecnologias relacionadas com os sistemas de informação têm colocado sucessivos desafios às empresas. De forma a tirar partido das potencialidades destas tecnologias, é necessário um grande investimento em software e hardware. Este impacto é visível não só nas grandes organizações de âmbito internacional, mas atinge também as pequenas e médias empresas.

Desde que surgiram, as tecnologias de informação potenciaram o aparecimento de novas indústrias, como sejam as consultoras de sistemas de informação ou as relacionadas com negócios na Internet, ou reforçaram a importância de outras, nomeadamente as ligadas à indústria de telecomunicações. Têm também provocado uma redefinição das responsabilidades e das interacções entre os parceiros da cadeia de valor de várias indústrias. Nos anos mais recentes, as tecnologias de informação têm mesmo posto em causa modelos tradicionais de fazer negócio.

Ao longo do tempo, o papel das tecnologias de informação nas organizações sofreu diversas alterações. Actualmente, as tecnologias de informação encontram-se na origem de mudanças significativas ao nível dos modelos de negócio das empresas, e constituem um elemento fundamental para a obtenção de vantagens estratégicas e competitivas. Por isso, a respectiva implementação nas organizações deve ser cuidadosamente planificada e estruturada, de modo a garantir o alinhamento com os objectivos estratégicos do negócio.

A implementação de sistemas de informação requer um investimento significativo (financeiro, tecnológico e de recursos humanos), pelo que estas intervenções deve-

rão merecer o apoio e o comprometimento dos órgãos de topo das organizações. A justificação destes volumes de investimento deve ser efectuada demonstrando qualitativamente e quantitativamente o seu valor estratégico e o impacto positivo nas organizações.

No entanto, muitos gestores não conseguem perceber o verdadeiro alcance de todas estas tecnologias, quer por questões de formação, quer pela sua anterior experiência com sistemas antiquados e obsoletos, que constituíam verdadeiros entraves à satisfação dos requisitos do negócio, e não funcionavam como potenciadores do seu crescimento. Por outro lado, os intervenientes da área de informática criaram de si próprios no passado uma imagem muito técnica, pouco alinhada com as reais necessidades do negócio, o que contribuiu decisivamente para a não caracterização da informática como uma área estratégica dentro das empresas.

A progressiva importância que os sistemas de informação têm nas organizações pode ser constatada através de vários exemplos:

- No passado era comum o responsável da informática depender hierarquicamente do director financeiro, enquanto este reportava directamente à administração. Pelo contrário, actualmente são cada vez menos as organizações em que esta situação se mantém, ficando a área de informática ao mesmo nível que os restantes departamentos e reportando directamente ao órgão que define a estratégia da organização (a administração); a informática passa assim a ser considerada como uma área estratégica, em pé de igualdade com as restantes áreas da organização.
- A indústria de software, ou de forma mais geral todas as relacionadas com as tecnologias de informação, é actualmente uma das mais importantes em todo o planeta e uma das principais responsáveis pelo crescimento contínuo da economia mundial durante a última década. Este fenómeno é também visível ao nível das individualidades, já que, segundo as estatísticas, o homem que é actualmente o mais rico do mundo é um dos principais responsáveis pela maior empresa de software (estamos obviamente a falar de Bill Gates e da Microsoft).
- O crescente peso na economia e na sociedade das empresas ligadas às tecnologias de informação, cujas acções são algumas das mais importantes das transaccionadas nas bolsas de valores internacionais.
- A importância destas empresas tem motivado a crescente preocupação dos governos em garantir o acesso livre ao mercado e a tentar evitar posições monopolistas. É o caso do litígio que opôs o governo americano e a Microsoft,

onde assistimos à disputa em torno de questões por vezes pouco racionais; no entanto, e independentemente da nossa posição pessoal, o governo americano actua de forma semelhante à dos seus antecessores há algumas décadas atrás, em relação a empresas de outras indústrias chave, como eram na altura a do petróleo e do aço.

Muitos outros exemplos poderiam ser dados, mas a conclusão óbvia é que nos tornámos dependentes das tecnologias de informação, quer do ponto de vista pessoal quer profissional.

1.3 Produto e Processo

A importância das tecnologias de informação na nossa vida é sobretudo concretizada pelas funcionalidades que são implementadas ao nível do software, e que são disponibilizadas com o suporte de um conjunto de dispositivos diversos (hardware). O primeiro pode ser considerado o componente lógico dos sistemas de informação, o segundo o componente físico.

Não existe uma definição rigorosa e inequívoca de software. Diversos autores [Pressman2000, Schach1999] encaram o software como o resultado final de um processo, ao qual designam por “Engenharia de Software”. O que é um facto é que o software não é dádiva da natureza, nem é objecto de uma produção numa linha de montagem, realizada de forma perfeitamente automática, sem qualquer intervenção humana, criativa e subjectiva.

Quando falamos em “processo” esta palavra implica desde logo a definição de um conjunto de actividades uniformizadas, a aplicar sistematicamente, que se encontram agrupadas em fases. Cada uma destas fases tem os seus intervenientes, aos quais são atribuídas responsabilidades, que possui diversas entradas e que produz saídas. Do ponto de vista da garantia da qualidade do produto final (o software), é fundamental que o processo seja realizado segundo parâmetros que permitam também aferir a respectiva qualidade, isto é, não conseguiremos otimizar o resultado final sem uma preocupação no processo que o produz.

Se pensarmos que o desenvolvimento do software é um processo que deve ser baseado na aplicação de técnicas e práticas rigorosas, sistemáticas, eficientes e controláveis, podemos concluir que este se aproxima bastante de outras realizações humanas, como a construção de qualquer obra de engenharia civil (por exemplo, a construção da ponte Vasco da Gama em Lisboa). Daí o nome de “Engenharia de Software” precisamente como tentativa de trazer para esta actividade a preocupação da aplicação de técnicas de engenharia ao desenvolvimento de software, por exemplo, modelar antes de realizar; estimar diversos factores antes de