

UML

METODOLOGIAS E FERRAMENTAS

CASE

Volume 2

2ª Edição, revista e actualizada



CENTROATLANTICO.PT

Alberto Silva e Carlos Videira

ALBERTO MANUEL RODRIGUES DA SILVA

CARLOS ALBERTO ESCALEIRA VIDEIRA

UML, METODOLOGIAS E FERRAMENTAS CASE

Vol. II

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



CENTRO**ATLANTICO**.PT

Portugal/2008

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 – VOL. II

Autores

Alberto Manuel Rodrigues da Silva

Carlos Alberto Escaleira Videira

Colecção

Tecnologias

Direcção gráfica

Centro Atlântico

Revisão

Centro Atlântico

Capa

Paulo Buchinho

Impressão e acabamento

PapelMunde

1ª edição

Março de 2008

ISBN

978-989-615-061-7

Depósito legal

/08

© Centro Atlântico, Lda., 2008

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

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 ao Segundo Volume da Segunda Edição

Seis 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 software baseado em modelos e o aparecimento de novas metodologias de desenvolvimento que pretendem acelerar e aumentar a produtividade dos projectos são apenas dois exemplos de áreas nas quais se podem observar progressos significativos. No entanto, a principal motivação para esta segunda edição resultou da publicação, em 2004, de uma nova versão do UML (2.0), com algumas inovações importantes.

Apesar de procurarmos nesta segunda edição deste livro minimizar as alterações aos principais objectivos e à estrutura, relativamente à primeira edição, é inevitável a introdução de novos temas, o refinamento de outros e a actualização de algumas matérias que entretanto sofreram evolução, ou devido à natural evolução e maturidade dos autores. 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 relativamente distintos.

Como já se tinha escrito no prefácio do Volume 1 desta segunda edição, as matérias que sofreram alterações mais significativas correspondem aos capítulos da Parte 2, incluídos no Volume 1, decorrente do esforço de actualização decorrente da evolução da versão 1.0 para a versão 2.0 do UML. No Volume 2, e uma vez que este livro pode ser lido de forma autónoma do primeiro, as suas duas partes, bem como os respectivos capítulos, foram renumerados de forma independente, de forma a começar no capítulo 1.

Devido à evolução que se assistiu nos últimos anos, designadamente ao nível das metodologias e das ferramentas que utilizam o UML, os capítulos que compõem este segundo volume foram significativamente revistos. Relativamente à Parte 1 (Metodologias) são apresentadas actualizações importantes nos capítulos sobre as metodologias RUP e ICONIX, e foi ainda introduzido um novo capítulo dedicado ao tema das metodologias ágeis, que aborda em particular o XP e o Scrum.

Relativamente à Parte 2 (sobre o tema Ferramentas CASE), o capítulo introdutório foi revisto de forma a incluir referências a novas iniciativas (por exemplo, o desenvolvi-

mento baseado em modelos). Em relação aos capítulos relativos às duas ferramentas UML, analisadas na primeira edição, decidimos avaliar o Rational Software Architect, uma das ferramentas que sucedeu ao Rational Rose, por continuar a ser uma das referências do mercado (obviamente reflectindo as versões mais recentes), e substituir o System Architect pelo Enterprise Architect, devido às capacidades desta última ferramenta e à sua crescente utilização no mercado. Devido à crescente importância do paradigma do desenvolvimento baseado em modelos (MDE – *Model Driven Engineering*), incluímos um novo capítulo que, para além de caracterizar melhor o âmbito deste conceito, descreve uma iniciativa de investigação – o ProjectIT – que se baseia nestes princípios, e na qual os autores têm vindo a participar ao longo dos últimos quatro anos.

Lisboa, Janeiro de 2008

Alberto Manuel Rodrigues da Silva

Carlos Alberto Escaleira Videira

Prefácio à Primeira Edição

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 despende 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 identificámos 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 baseados em 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, 17 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.

VOLUME 1 – ENQUADRAMENTO E LINGUAGEM UML

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

A Parte 2 (LINGUAGEM DE MODELAÇÃO UML) do primeiro volume 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.

VOLUME 2 – METODOLOGIAS E FERRAMENTAS CASE

A Parte 1 (METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE) do segundo volume apresenta exemplos concretos de propostas metodológicas de desenvolvimento de software, nomeadamente: o RUP (no Capítulo 1), o ICONIX (no Capítulo 2), e as metodologias ágeis, em particular o XP e o Scrum, descritos respectivamente no (Capítulo 3).

A Parte 2 (FERRAMENTAS CASE) do segundo volume apresenta a problemática das ferramentas CASE descrevendo o seu significado, evolução histórica e discutindo mecanismos de caracterização e avaliação (Capítulo 4). São apresentadas e analisadas duas das ferramentas que têm feito um investimento forte na modelação em UML, o Rational Software Architect, um dos sucessores do Rational Rose analisado na primeira edição do livro, e o Enterprise Architect, respectivamente nos Capítulos 5 e 6. No Capítulo 7 é apresentada a iniciativa ProjectIT que se enquadra no âmbito das abordagens de desenvolvimento baseadas em modelos.

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 escritos em itálico. As exceções são expressões vulgarmente adoptadas para o português (e.g., software, bit), expressões intensamente usadas ao longo do texto (e.g., Internet, Web, standard), ou nomes de empresas ou produtos de origem anglo-saxónica (e.g., MS-Word, Enterprise Architect).
- Frases e expressões que se pretendam destacar são escritas com ênfase (i.e., negrito).
- Exemplos de código, pseudo-código, 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.

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 este livro, bem como pelas inúmeras horas roubadas ao seu convívio.

Parte significativa da actividade que conduziu à realização deste livro foi desenvolvida ao longo de vários anos, no âmbito de duas instituições de excelência nacional, o Instituto Superior Técnico e o INESC-ID, às quais expresso o meu maior agradecimento. Agradeço igualmente a todos os colegas e alunos com quem tive o privilégio de conviver, aprender e ensinar durante estes anos.

Um agradecimento à editora Centro Atlântico e ao seu editor, Libório Manuel Silva, pelo interesse imediato na publicação do livro, pela disponibilidade manifestada, e pela grande paciência que foram tendo na expectativa da entrega desta nossa edição.

Por fim, um agradecimento a todos que de uma forma ou outra sugeriram, comentaram ou apenas criticaram partes diversas deste trabalho, ou com quem simplesmente fui partilhando a ‘ideia’ do livro, mais recentemente e em particular a Patrícia Dinis, Paula Martins, João Saraiva e David Ferreira.

Alberto Silva

Quero em primeiro lugar agradecer à minha família, pela sua dedicação, carinho e apoio incondicional, e a quem tive de “roubar” momentos importantes de convívio para poder participar neste projecto. Quero também agradecer aos meus amigos, de cujo convívio tive que prescindir para poder completar este livro.

A minha experiência como docente, colaborando com diversas instituições, quer ao nível de cursos de licenciatura, pós-graduações ou simples acções de formação, foi também um factor decisivo para a minha participação neste livro. As conversas que no contexto destes cursos fui mantendo com colegas e alunos proporcionaram uma permuta de opiniões e comentários que me ajudaram a melhorar a qualidade da presente obra e que contribuiu em diversos aspectos para a evolução do conteúdo de uma parte significativa deste livro.

Finalmente, à editora Centro Atlântico e ao Libório Manuel Silva deixo um agradecimento pela sua paciência em relação ao tempo que demorou a elaborar este segundo volume, e pelo seu interesse na publicação desta obra técnico-científica, valorizando a missão de educar para o futuro.

Carlos Videira

Lisboa, Janeiro de 2008

Alberto Manuel Rodrigues da Silva

Carlos Alberto Escaleira Videira

Contactos

Comentários, sugestões gerais ou pedidos de aquisição de exemplares deverão ser dirigidos à editora Centro Atlântico (ver contactos na ficha técnica deste livro).

Encontra-se disponível para consulta e *download*, em formato electrónico, partes complementares deste livro, nomeadamente: o seu resumo, o currículo abreviado dos autores, a lista de exercícios propostos e o guia de recursos electrónicos referidos em Apêndice A em <http://www.centroatl.pt/titulos/tecnologias/uml2-vol2.php3>.

Autores

Alberto Manuel Rodrigues da Silva é professor no Departamento de Engenharia Informática (DEI) do IST/UTL, sócio e director da empresa SIQuant - Engenharia do Território e Sistemas de Informação, e investigador sénior no INESC-ID Lisboa. 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 Sistema de Informação de nível licenciatura e pós-graduação. 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. É Coordenador da Região Sul do Colégio de Engenharia Informática da Ordem dos Engenheiros. É autor ou co-autor de 3 livros técnicos nacionais e mais de 100 artigos científicos em revistas, conferências e *workshops* nacionais e internacionais; é ainda editor de 2 livros internacionais.

Carlos Alberto Escalera 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, e está a concluir o seu doutoramento na área de Engenharia de Requisitos. 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 ao Segundo Volume da Segunda Edição	i
Prefácio à Primeira Edição	iii
Índice	xi

PARTE 1 – METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE 1

Capítulo 1 - Metodologia RUP	5
1.1 Introdução	5
1.2 Enquadramento Histórico	9
1.3 Características Principais	11
1.3.1 Metodologia Conduzida por Casos de Utilização	12
1.3.2 Metodologia Centrada numa Arquitectura	13
1.3.3 Metodologia Iterativa e Incremental	15
1.4 As 4+1 Visões	16
1.5 Visão Geral	17
1.5.1 Conceitos Gerais	17
1.5.2 Componente Dinâmica	20
1.5.3 Componente Estática	21
1.6 Fases e Iterações – A Componente Dinâmica	22
1.6.1 Concepção	23
1.6.2 Elaboração	24
1.6.3 Construção	25
1.6.4 Transição	26
1.6.5 Comentários Gerais	27
1.7 Disciplinas, Tarefas e Produtos de Trabalho – A Componente Estática	29
1.7.1 Disciplina de Gestão do Projecto	30
1.7.2 Disciplina de Modelação do Negócio	32
1.7.3 Disciplina de Requisitos	33

1.7.4	Disciplina de Análise e Desenho	35
1.7.5	Disciplina de Implementação	36
1.7.6	Disciplina de Testes	38
1.7.7	Disciplina de Instalação	39
1.7.8	Disciplina de Gestão da Configuração e das Alterações	40
1.7.9	Disciplina de Ambiente	41
1.8	Rational Method Composer	42
1.9	Caso de Estudo DGD	44
1.9.1	Enunciado do Caso de Estudo DGD	44
1.9.2	Resolução do Caso de Estudo DGD	46
1.10	Conclusão	58
1.11	Exercícios	59
Capítulo 2 -	Metodologia ICONIX	61
2.1	Introdução	61
2.2	Visão Geral	63
2.2.1	Análise de Requisitos	65
2.2.2	Análise e Desenho Preliminar	66
2.2.3	Desenho Detalhado	68
2.2.4	Implementação	69
2.3	Validações no ICONIX	69
2.4	Caso de Estudo WebDEI – Enunciado	71
2.4.1	Introdução	72
2.4.2	Arquitectura Geral	72
2.4.3	Tipos Básicos de Informação	73
2.4.4	Breve Descrição Funcional do Sistema	74
2.5	Caso de Estudo WebDEI - Resolução	76
2.5.1	Análise de Requisitos	76
2.5.2	Análise e Desenho Preliminar	85
2.5.3	Desenho	91
2.5.4	Implementação	96
2.6	Conclusão	98
2.7	Exercícios	99
Capítulo 3 -	Metodologias ágeis	101
3.1	Introdução	101
3.2	Manifesto para o Desenvolvimento Ágil de Software	102
3.3	XP – Extreme Programming	104
3.3.1	Valores	105
3.3.2	Princípios	106
3.3.3	Práticas	106
3.3.4	Papéis	108
3.3.5	Artefactos	111

3.3.6	Actividades	112
3.3.7	Um Dia Típico de Trabalho...	117
3.3.8	Discussão	120
3.4	Scrum	121
3.4.1	Papéis	122
3.4.2	Artefactos	124
3.4.3	Actividades	127
3.4.4	Discussão	128
3.5	Outros Processos Ágeis	129
3.5.1	Adaptive Software Development (ASD)	129
3.5.2	Crystal	129
3.5.3	PSP e TSP	130
3.6	Discussão	131
3.7	Exercícios	132

PARTE 2 – FERRAMENTAS CASE **135**

Capítulo 4 - Ferramentas CASE	139	
4.1	Introdução	139
4.2	Evolução Histórica	141
4.3	Mecanismos de Integração entre Ferramentas	147
4.4	Taxonomia das Ferramentas CASE	150
4.5	Vantagens e Problemas das Ferramentas CASE	153
4.6	Funcionalidades das Ferramentas CASE	154
4.7	Geração Automática de Artefactos	158
4.7.1	Round-Trip Engineering	158
4.7.2	Geração de Documentação	160
4.8	Avaliação de Ferramentas CASE	160
4.9	Ferramentas de Modelação para UML	162
4.9.1	Modelação de Bases de Dados	163
4.9.2	Modelação do Negócio	164
4.10	Conclusão	165
4.11	Exercícios	166
Capítulo 5 - Rational Software Architect	167	
5.1	Introdução	167
5.2	Evolução Histórica	168
5.3	Características Gerais	170
5.4	Técnicas de Modelação	174

5.4.1	Modelação em UML	175
5.4.2	Modelação de Dados	179
5.4.3	Modelação de Negócio	182
5.4.4	Outras Técnicas de Modelação	183
5.5	Elaboração de Documentação	184
5.6	Mecanismos de Extensibilidade	188
5.6.1	Mecanismos de Extensibilidade no Rational Rose	188
5.6.2	Mecanismos de Extensibilidade no RSA	189
5.7	Desenvolvimento baseado em modelos	194
5.7.1	Edição visual	194
5.7.2	Transformações	198
5.8	Conclusão	203
Capítulo 6 -	Enterprise Architect	205
6.1	Introdução	205
6.2	Características Gerais	206
6.3	Técnicas de Modelação	210
6.3.1	Modelação em UML	210
6.3.2	Outros modelos	212
6.3.3	Modelação de Dados	216
6.3.4	Modelação de Negócio	221
6.4	Suporte ao trabalho em equipa	222
6.4.1	Controlo de versões	223
6.4.2	Mecanismos de segurança	227
6.4.3	Mecanismos de auditoria	229
6.5	Mecanismos de geração de código	230
6.6	Transformações entre modelos	237
6.7	Mecanismos de extensibilidade	241
6.8	Documentação	243
6.9	Conclusão	247
Capítulo 7 -	Iniciativa ProjectIT	249
7.1	Introdução	249
7.2	ProjectIT-Enterprise – Ferramenta CSCW	252
7.2.1	Plataforma de Suporte – WebComfort	253
7.2.2	Dimensão Organizacional	255
7.2.3	Dimensão de Processos	257
7.2.4	Dimensão de Projectos	260
7.2.5	Outros Aspectos	264
7.3	ProjectIT-Studio – Ferramenta CASE	265
7.3.1	Aproximação ProjectIT	266
7.3.2	ProjectIT-Studio/Requirements	269
7.3.3	ProjectIT-Studio/UMLModeler	271

7.3.4	ProjectIT-Studio/MDDGenerator	273
7.4	Conclusões	275
Apêndice A – Guia de Recursos Electrónicos		279
	Standards, Organizações Normalizadoras, Iniciativas e Fóruns	280
	Links Relevantes	280
	Leituras Recomendadas	280
	Ferramentas UML Comerciais	281
	Ferramentas UML Open Source	282
Apêndice B – Glossário, Siglas e Abreviaturas		283
	B.1 Glossário	284
	B.2 Siglas mais Usadas	285
	B.3 Abreviaturas	286
Apêndice C – Elementos UML Predefinidos		287
	C.1 Palavras Reservadas	287
	C.2 Estereótipos Standard	291
Referências		295
Índice Remissivo		307

Parte 1

Metodologias de Desenvolvimento de Software

- *O que é um ritual? – perguntou o Principezinho.*
- *Também é uma coisa que toda a gente se esqueceu – respondeu a raposa. – É o que faz com que um dia seja diferente dos outros dias e uma hora, diferente das outras horas. Os meus caçadores, por exemplo, têm um ritual. À quinta-feira, vão ao baile com as raparigas da aldeia. Assim a quinta-feira é um dia maravilhoso. Eu posso ir passear para as vinhas. Se os caçadores fossem para o baile num dia qualquer, os dias eram todos iguais uns aos outros e eu nunca tinha férias.*

Antoine de Saint-Exupéry, O Principezinho.

A Necessidade de uma Metodologia

A Parte 1 deste segundo volume marca o retomar do livro na análise dos conceitos mais abrangentes da engenharia de software e dos sistemas de informação. Depois de termos apresentado a linguagem UML, ao longo da segunda parte do primeiro volume, colocam-se naturalmente diversas questões: Como é que podemos utilizar o UML para modelar sistemas de software? Quando é que devemos usar os diferentes tipos de diagramas? Com que nível de detalhe? Qual o melhor processo para definir relações entre os vários tipos de diagramas, por exemplo, entre diagramas de casos de utilização e diagramas de interacção? A resposta a estas e outras questões passa por conhecer e aplicar uma determinada metodologia que oriente a utilização do UML ao longo do processo de desenvolvimento (ou simplesmente em actividades de modelação) de software.

O que é uma Metodologia?

Tal como discutido no Capítulo 2 do primeiro volume, o processo de desenvolvimento de software consiste genericamente num conjunto de fases, tarefas e actividades, realizadas por intervenientes que desempenham várias funções, de modo a elaborarem diversos artefactos e que conjuntamente contribuem para a produção de um sistema de software. O conceito de metodologia implica adicionalmente a definição de aspectos que torna ‘concretizável’ a noção de processo, designadamente a utilização de técnicas, notações e ferramentas.

O processo de desenvolvimento de software é um acto complexo, com múltiplas variáveis e inúmeras vezes com qualidade reduzida, ultrapassando os prazos e os orçamentos previstos. Não há receitas milagrosas, por isso um processo deve ser adaptado e configurado conforme o perfil das empresas, dos projectos e dos intervenientes envolvidos. Não acreditamos na produção de software em massa e automatizada. Esta actividade não trivial é e será sempre uma combinação entre engenharia e arte, entre trabalho individual e trabalho de equipa.

Nesta primeira parte do segundo volume do livro aprofundamos a problemática das metodologias de desenvolvimento de software através da análise e discussão de duas propostas concretas que adoptam o UML como linguagem de modelação (o RUP e o ICONIX) e fazendo referência a um conjunto de metodologias e técnicas, normalmente designadas pela expressão de «abordagens ágeis», que têm ganho importância quer em termos de investigação, quer na sua utilização em projectos do mundo real.

Esta primeira parte pressupõe que o leitor tenha algum conhecimento sobre os conceitos que estão normalmente associados à definição de metodologias. Estes conceitos foram de alguma forma apresentados e discutidos na primeira parte do primeiro volume, e por isso não são aqui repetidos.

Organização da primeira parte do segundo volume

O Capítulo 1, “Metodologia RUP”, apresenta o *Rational Unified Process*, uma metodologia completa aplicada ao desenvolvimento de software. Encontra-se fortemente integrado com o UML, ou não tivesse resultado da continuação dos esforços dos mesmos autores. O RUP segue um modelo iterativo e incremental, centrado numa arquitectura e conduzido por casos de utilização. É uma metodologia que se baseia no paradigma da orientação por objectos, e a abrangência das suas propostas contempla a possibilidade de adaptação à realidade de cada projecto ou organização.

Neste capítulo, apresenta-se uma panorâmica geral do RUP, em particular as suas diferentes visões: a visão dinâmica (baseada em ciclos, fases e iterações) e a visão estática (baseada em disciplinas, actividades e produtos de trabalho). Por fim, discute-se o RUP através da sua aplicação a um caso prático, de dimensão e complexidade não trivial, e ainda as evoluções mais recentemente efectuadas pela IBM, ao nível de suporte de ferramentas para a aplicação da metodologia. Como se poderá constatar, os conceitos base do RUP permanecem os mesmos desde as versões iniciais, mas houve uma tentativa de responder às críticas que lhe foram sendo efectuadas ao longo dos anos, agilizando a sua aplicação e melhorando o suporte por parte de ferramentas.

O Capítulo 2, “Metodologia ICONIX”, apresenta a metodologia ICONIX, para desenvolvimento de software segundo uma abordagem muito prática, algures entre a complexidade e abrangência do RUP e a simplicidade e o pragmatismo das abordagens ágeis. O ICONIX é conduzido por casos de utilização, iterativo e incremental, tal como o RUP, mas sem a complexidade que este preconiza. Por outro lado, é relativamente pequeno e simples, tal como o XP, mas sem eliminar as tarefas de análise e de desenho que aquele não desenvolve significativamente. Discute-se o ICONIX através da sua aplicação na modelação do caso de estudo WebDEI, o qual consiste num sistema de informação Web multi-camada, desenvolvido exclusivamente com tecnologia Java.

Apesar de não ser uma metodologia muito divulgada e por isso não conhecida do grande público, decidimos manter este capítulo no presente livro, devido quer à simplicidade da metodologia ICONIX, quer ao facto de incluir as actividades básicas, que muitas vezes são realizadas mesmo por quem não conhece os fundamentos teóricos das metodologias orientadas por objectos, baseadas em UML. A principal razão tem a ver com o facto do ICONIX ser uma das metodologias em que a relação entre os conceitos de metodologia e de modelação é mais forte, por ter sido concebida de raiz para a aplicação das técnicas de modelação UML.

Finalmente o Capítulo 3 descreve os princípios subjacentes a um conjunto de diversas abordagens, designadas por «abordagens ágeis». A sua origem assenta em princípios e valores comuns, que estão descritos no «Manifesto Ágil», e reflectem preocupações que vão além dos tradicionais aspectos metodológicos mais técnicos, ao incluírem

também aspectos de natureza social, humana e trabalho de equipa. O seu enfoque é sobretudo nas pessoas, na qualidade do seu trabalho, na definição de técnicas que permitam atingir o verdadeiro objectivo (o desenvolvimento do código de um sistema de informação), e principalmente na criação de um ambiente de trabalho agradável e motivador.

Entre as várias abordagens ágeis, destaca-se em particular o *Extreme Programming* (XP), que adquiriu uma popularidade significativa junto do grande público, o que também motivou a publicação de várias referências bibliográficas. Mais recentemente, o Scrum tem merecido a atenção crescente quer da comunidade académica, quer dos profissionais que desenvolvem sistemas no dia-a-dia. Por isso, optámos por descrever os princípios genéricos das abordagens ágeis, e de seguida por fazer uma descrição mais detalhada das práticas relacionadas com o XP e o Scrum, que apresentem aspectos distintivos mas complementares. O capítulo termina com uma breve referência a outras abordagens, como seja Crystal, ASD, PSP e TSP.

1.11 Exercícios

- Ex.1. Tendo em conta os seus conhecimentos sobre o RUP, quais são os tipos de empresas que maior partido poderão tirar desta metodologia de desenvolvimento? Justifique a sua afirmação
- Ex.2. Relativamente à metodologia de desenvolvimento RUP, aplicam-se normalmente critérios uniformes para definir o tempo de duração da fase de concepção, em função de experiências do passado, e introduzem-se alguns factores de ajustamento que podem aumentar ou diminuir esse tempo. Para além dos factores referidos no livro, indique outros que na sua opinião poderão influenciar os valores das referidas estimativas.
- Ex.3. Imagine o desenvolvimento de um sistema de informação de complexidade relativamente baixa, cuja duração total não excederia três meses. Seria possível e desejável aplicar o RUP? Em caso afirmativo, que alterações ou adaptações introduziria?
- Ex.4. O RUP refere que a aplicação do *workflow* de modelação do negócio é opcional. Em que circunstâncias é que recomendaria a sua inclusão num projecto de desenvolvimento de software.
- Ex.5. Complete o diagrama de casos de utilização apresentado na Figura 1.24, relativamente ao actor Cidadão.
- Ex.6. Identifique quais os principais riscos que o projecto poderá apresentar, e proponha algumas medidas de correcção.
- Ex.7. Elabore o diagrama de classes simplificado (o modelo de domínio) referente ao caso de estudo apresentado.

2.7 Exercícios

- Ex.8. Tendo em conta os seus conhecimentos sobre o ICONIX, quais são os tipos de empresas que maior partido poderão tirar desta metodologia de desenvolvimento?
- Ex.9. Quais são as principais características do ICONIX?
- Ex.10. Quais são os diagramas UML cuja aplicação não é relevante segundo o ICONIX?
- Ex.11. Quais são os principais tipos de objectos usados na actividade de análise de robustez?
- Ex.12. O que são «diagramas de robustez»? Qual a sua importância no contexto do ICONIX?
- Ex.13. Qual é a posição do ICONIX relativamente à relação entre casos de utilização e requisitos? Discuta justificadamente se concorda ou não com essa posição.
- Ex.14. Faça a descrição textual dos casos de utilização apresentados na Figura 2.11, relativamente ao caso de estudo WebDEI.
- Ex.15. Qual é a sequência de passos preconizada no ICONIX para se passar de um diagrama de robustez (conjuntamente com a descrição textual do caso de utilização) para um diagrama de sequência?
- Ex.16. Modifique o diagrama de classes da fase de desenho, apresentado na Figura 2.19, relativamente ao caso de estudo WebDEI, considerando que tanto os *servlets* (derivados a partir da classe *WebDEIServlet*) como o *WebDEIServer* têm acesso às classes de entidade (e.g., *Evento*, *Docente*, *AreaCientifica*), e que em vez de trocarem strings, trocam instâncias dessas classes serializadas.

3.7 Exercícios

Ex.17. Considere o desenho de um modelo de especificação dos elementos de um processo, que envolva os seguintes conceitos: (1) Actividade (*activity*); (2) Produto (*work product* ou *artefact*); (3) Papel (*role*); (4) Disciplinas (*discipline*); e (5) Ciclo de vida: Fases (*phase*) e Iterações (*iteration*). Aplique esse modelo ao XP.

Ex.18. Considere o desenho de um modelo de especificação dos elementos de um processo, que envolva os seguintes conceitos: (1) Actividade (*activity*); (2) Produto (*work product* ou *artefact*); (3) Papel (*role*); (4) Disciplinas (*discipline*); e (5) Ciclo de vida: Fases (*phase*) e Iterações (*iteration*). Aplique esse modelo ao Scrum.

Ex.19. Enumere e discuta as principais limitações de processos ágeis como o XP.

Ex.20. Discuta a adopção do XP em (1) Projectos com/para a administração pública; e (2) Projectos internos numa operadora de telecomunicações.

Ex.21. Discuta a adopção do Scrum em (1) Projectos com/para a administração pública; e (2) Projectos internos numa operadora de telecomunicações.

Ex.22. Seleccione a melhor resposta (apenas uma) das questões abaixo. No XP o "*Release planning game*" é disputado por:

- A. Programadores
- B. Clientes
- C. Programadores e Clientes
- D. Programadores, Clientes, Monitorizador

Ex.23. Seleccione a melhor resposta (apenas uma) das questões abaixo. No XP o "*Iteration planning game*" é disputado por:

- A. Programadores
- B. Clientes
- C. Programadores e Clientes
- D. Programadores, Clientes, Monitorizador

Ex.24. Seleccione a melhor resposta (apenas uma) das questões abaixo. O XP recomenda que uma tarefa seja realizada em 2 a 4:

- A. Horas
- B. Dias
- C. Semanas
- D. Meses

Ex.25. Seleccione a melhor resposta (apenas uma) das questões abaixo. O XP recomenda que uma iteração seja realizada em 2 a 4:

- A. Horas
- B. Dias
- C. Semanas
- D. Meses

Ex.26. Seleccione a melhor resposta (apenas uma) das questões abaixo. No Scrum o "*Sprint Planning*" é realizado por:

- A. *Product Owner*
- B. Clientes
- C. *Product Owner, Team*
- D. *Product Owner, Team, ScrumMaster*

Ex.27. Seleccione a melhor resposta (apenas uma) das questões abaixo. No Scrum o "*Sprint Review*" é realizado por:

- A. *Product Owner*
- B. Clientes
- C. *Product Owner, Team*
- D. *Product Owner, Team, ScrumMaster*

Ex.28. Seleccione a melhor resposta (apenas uma) das questões abaixo. O Scrum recomenda que uma tarefa seja realizada entre 4 a 16:

- A. Horas
- B. Dias
- C. Semanas
- D. Meses

Ex.29. Seleccione a melhor resposta (apenas uma) das questões abaixo. O Scrum recomenda que um *Sprint* seja realizado em 2 a 4:

- A. Horas
- B. Dias
- C. Semanas
- D. Meses

Ex.30. Discuta as principais vantagens e desvantagens da aplicação de um processo ágil, como o XP, relativamente a um processo mais ‘pesado’ como o RUP.

Ex.31. Para cada um dos processos ágeis referidos neste capítulo (i.e., ASD, PSP e TSP, DSDM, *Agile Modeling*, OpenUP, FDD, AUP, MSF), investigue, discuta e identifique claramente os seguintes conceitos: (1) papéis; (2) terminologia de fases, iterações e actividades; e (3) artefactos principais.

Ex.32. Compare o Scrum com o XP do ponto de vista do papel do Gestor de Projecto. (Sugestão: liste as suas principais responsabilidades e tarefas)

Ex.33. Discuta a aplicação do Scrum em complementaridade com o XP.

4.11 Exercícios

- Ex.34. Qual a importância da definição e aplicação dos standards de integração de ferramentas? Quais as razões pelas quais, na sua opinião, eles têm falhado na realização dos objectivos propostos.
- Ex.35. Um dos grandes objectivos há muito tempo perseguido pelas ferramentas CASE é a geração automática de código. No entanto, até à data estas iniciativas não têm tido o impacto esperado. Indique algumas razões pelas quais tal pode ter acontecido.
- Ex.36. De que modo a evolução da tecnologia tem condicionado o sucesso das ferramentas CASE?
- Ex.37. Segundo a taxonomia de ferramentas CASE apresentada neste capítulo, classifique em termos de importância (fundamental, importante, pouco importante, sem relevância) cada uma das categorias apresentadas, justificando a sua resposta.
- Ex.38. Pensa que seria possível utilizar uma ferramenta de modelação de processos, segundo as abordagens tradicionais, para suportar o processo RUP? Justifique a sua resposta.

APÊNDICE A – GUIA DE RECURSOS ELECTRÓNICOS

Neste apêndice é compilado um conjunto significativo de referências electrónicas (*links*) relacionadas com os temas abordados neste livro. Esta compilação está organizada num conjunto de categorias, de modo a facilitar ao leitor a procura e navegação de informação. As categorias consideradas são as seguintes:

- **Standards e Organizações Normalizadoras:** São tipicamente consórcios de empresas e de centros de investigação cujo objectivo é a promoção, divulgação de informação sobre os métodos de A&D, linguagens de modelação, e processos de desenvolvimento de software, ou então referências disponíveis *online* sobre algumas normas referidas ao longo do texto.
- **Leituras Recomendadas:** Constituem um conjunto de referências clássicas, suficientemente simples de entender, que abordam diversos aspectos que poderão facilitar a compreensão dos temas abordados.
- **Catálogos de Informação (Meta-Informação):** Estes recursos constituem excelentes pontos de partida para uma procura geral sobre os temas abordados; apresentam um número significativo de referências electrónicas para distintos locais, organizados por distintos critérios; alguns destes recursos providenciam ainda jornais *online*, grupos de discussão e/ou listas de correio-electrónico. Poderão incluir referências para recursos bibliográficos existentes na Internet, nomeadamente para livros, revistas, artigos publicados em conferências, etc.
- **Empresas:** Lista de empresas cuja actuação tem, ao longo do tempo, apresentado um significado relevante para os temas abordados neste livro. Algumas empresas não são incluídas nesta lista pelo facto de serem referidas no ponto seguinte.
- **Ferramentas CASE:** Lista de empresas e ferramentas de modelação de software. Esta lista, sem pretender ser exaustiva, apresenta algumas das referências mais relevantes nesta categoria de ferramentas, podendo servir como ponto de partida para o leitor efectuar a sua própria investigação.

Standards, Organizações Normalizadoras, Iniciativas e Fóruns

Título	URL
IDEF Family of Methods	www.idef.com
Object Management Group (OMG) UML	www.omg.org/uml
Object Management Group (OMG) MDA	www.omg.org/mda
OCL (Object Constraint Language)	www.klasse.nl/ocl/
OOAD_UML Group	groups.yahoo.com/group/OOAD_UML/
Precise UML (pUML) Group	www.cs.york.ac.uk/puml
Software Engineering Body of Knowledge	www.swebok.org
Software Engineering Institute	www.sei.cmu.edu
Software Engineering Institute Product Line Practice Initiative	www.sei.cmu.edu/productlines/plp_init.html
UML Zone (DevX)	www.devx.com/enterprise/Door/10606
UML Forum (1)	www.uml-forum.com
UML Forum (2)	groups.yahoo.com/group/uml-forum/
2U Consortium (Unambiguous UML)	www.2uworks.org

Links Relevantes

Título	URL
Catalysis	www.catalysis.org
Extreme Programming (Donovan Wells)	www.extremeprogramming.org
Extreme Programming (Ron Jeffries)	www.xprogramming.com
ICONIX Software Engineering	www.iconix.sw.com
Manifesto Ágil	agilemanifesto.org
Methodology.org	www.itmweb.com/methodology
Projecto ProjectIT	isg.inesc-id.pt/alb/ProjectIT@81.aspx
Projecto XIS	berlin.inesc-id.pt/projects/xis/
RUP (Rational Unified Process)	www-306.ibm.com/software/awdtools/rup/
Scrum	www.scrumalliance.org

Leituras Recomendadas

Título	URL
Agent UML Web Site	aot.ce.unipr.it/auml
CETUS Links: Object-Oriented Analysis & Design Methods	www.cetus-links.org/oo_ooa_ood_methods.html
CETUS Links: Object-Oriented Analysis & Design Tools	www.cetus-links.org/oo_ooa_ood_tools.html
CETUS Links: Unified Modeling Language (UML)	www.cetus-links.org/oo_uml.html
Data Modeling	www.datamodel.org
Guide to selecting a UML modeling tool	www.objectsbydesign.com/tools/modeling_tools.html
Objects by Design	www.objectsbydesign.com

Selection list of UML tools.(Objects by Design)	www.objectsbydesign.com/tools/umltools_byCompany.html
The UML Bibliography (Mark Richters)	www.db.informatik.unibremen.de/umlbib/home.html

Ferramentas UML Comerciais

Empresa	Ferramenta	URL
Aonix	Software through Pictures	www.aonix.com/index.html
Artisan	Artisan Studio	www.artisansw.com
Borland	Together	www.borland.com/us/products/together/index.html
Casewise	Casewise Corporate Modeler	www.casewise.com
Computer Associates	Erwin, Bpwin	www.ca.com
Compuware	OptimalJ	www.compuware.com/products/optimalj/
Gentleware	Poseidon	www.gentleware.com
IBM Rational	Rational Software Architect	www-306.ibm.com/software/awdtools/architect/swarchitect
IBM Rational	Rational Software Modeler	www-306.ibm.com/software/awdtools/modeler/swmodeler/
IDS Scheer AG	Aris	www.ids-scheer.com
Mega International	Mega Modeling Suite	www.mega.com
Metastorm	ProVision	www.metastorm.com/products/mpea.asp
Microgold Software	WithClass	www.microgold.com
Microsoft	Visio	www.microsoft.com/office/visio
No Magic's	MagicDraw UML	www.magicdraw.com
Oracle	Designer, Developer	www.oracle.com
Grandite	Silverrun	www.silverrun.com
Softeam	Objecteering	www.objecteering.com
Sparx Systems	Enterprise Architect	www.sparxsystems.com
Sybase	PowerDesigner	www.sybase.com
Telelogic	Doors, Focal Point, System Architect, Rhapsody, TAU	www.telelogic.com
Troux Technologies	Metis Architect	www.troux.com/products/metis_architect/
Universidade de Paderborn, Alemanha	Fujaba	www.es.upb.de/cs/fujaba/
Visible Systems Corporation	Visible Analyst	www.visible.com
Visual Object Modelers	Visual UML	www.visualobject.com

Ferramentas UML Open Source

Título	URL
Argo UML	<i>argouml.tigris.org</i>
Eclipse Modeling Tools	<i>www.eclipse.org/modeling/mdt/?project=uml2</i>
MonoUML	<i>monouml.sourceforge.net</i>
Papyrus UML	<i>www.papyrusuml.org</i>
Star UML	<i>staruml.sourceforge.net/en/</i>
TopCased	<i>www.topcased.org</i>
Umbrello	<i>uml.sourceforge.net/index.php</i>
UMLLet	<i>www.umlet.com</i>

Devido à evolução que se assistiu nos últimos anos, designadamente ao nível das metodologias de desenvolvimento de sistemas de informação e das ferramentas CASE que suportam em particular a modelação em UML, os capítulos que compõem este Volume 2 foram significativamente revistos e expandidos relativamente à 1ª edição do livro. Este livro, Volume 2, complementa o respectivo Volume 1, e encontra-se estruturado em duas partes.

A Parte 1 (Metodologias) introduz actualizações importantes nos capítulos sobre as metodologias RUP e ICONIX, e também um novo capítulo dedicado ao tema emergente das metodologias ágeis que aborda, em particular, o XP (Extreme Programming) e o Scrum, e discute os seus princípios, conceitos e boas-práticas.

A Parte 2 (Ferramentas CASE) revê o capítulo introdutório de forma a actualizar referências a novas iniciativas, como por exemplo, ligadas ao paradigma do desenvolvimento baseado em modelos (MDE – Model Driven Engineering). Relativamente aos capítulos que descreviam duas ferramentas CASE, analisadas na 1ª edição, decidiu-se avaliar o Rational Software Architect, uma das ferramentas que sucedeu ao Rational Rose, por continuar a ser uma referência do mercado, e substituir a descrição do System Architect pela do Enterprise Architect, devido às capacidades desta última ferramenta e à sua crescente utilização e popularidade. Inclui-se, por fim, um novo capítulo que descreve a iniciativa de investigação ProjectIT, na qual os autores têm vindo a participar ao longo dos últimos quatro anos, e onde se volta a discutir, de forma abrangente, as questões das metodologias e do seu suporte por ferramentas adequadas.

Alberto Manuel Rodrigues da Silva é professor no Departamento de Engenharia Informática (DEI) do IST/UTL, sócio e director da empresa SIQuant - Engenharia do Território e Sistemas de Informação, e investigador sénior no INESC-ID Lisboa. 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 e pós-graduação. Supervisiona a realização de vários trabalhos finais de curso, teses de mestrado e de doutoramento. No INESC-ID é membro e co-fundador do Grupo de Sistemas de Informação. É Coordenador da Região Sul do Colégio de Engenharia Informática da Ordem dos Engenheiros. É autor ou co-autor de 3 livros técnicos nacionais e mais de 100 artigos científicos em revistas, conferências e *workshops* nacionais e internacionais; é ainda editor de 2 livros internacionais.

Carlos Alberto Escalreira 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, e está a concluir o seu doutoramento na área de Engenharia de Requisitos. 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.

