

---

# *SCENARIOS, STORIES, USE CASES*

Through the Systems Development Life-Cycle

*Edited by*

**IAN ALEXANDER AND NEIL MAIDEN**



John Wiley & Sons, Ltd



---

# *SCENARIOS, STORIES, USE CASES*



---

# *SCENARIOS, STORIES, USE CASES*

Through the Systems Development Life-Cycle

*Edited by*

**IAN ALEXANDER AND NEIL MAIDEN**



John Wiley & Sons, Ltd

Copyright © 2004

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,  
West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): [cs-books@wiley.co.uk](mailto:cs-books@wiley.co.uk)

Visit our Home Page on [www.wileyeurope.com](http://www.wileyeurope.com) or [www.wiley.com](http://www.wiley.com)

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1T 4LP, UK, without the permission in writing of the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system for exclusive use by the purchaser of the publication. Requests to the Publisher should be addressed to the Permissions Department, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, or emailed to [permreq@wiley.co.uk](mailto:permreq@wiley.co.uk), or faxed to (+44) 1243 770620.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The Publisher is not associated with any product or vendor mentioned in this book.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

#### ***Other Wiley Editorial Offices***

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

#### ***British Library Cataloguing in Publication Data***

A catalogue record for this book is available from the British Library

ISBN 0-470-86194-0

Typeset in 10/12pt Times by Laserwords Private Limited, Chennai, India

Printed and bound in Great Britain by Biddles Ltd, King's Lynn

This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

# CONTENTS

PREFACE **xiii**

BIOGRAPHIES AND PHOTOGRAPHS **xvii**

## PART I

### OVERVIEW

#### CHAPTER 1 INTRODUCTION: SCENARIOS IN SYSTEM DEVELOPMENT

Context	3
Scope—What Does Scenario Mean, and What Does it Cover?	6
Stakeholders	6
Story	8
Situation, Alternative World	10
Simulation	10
Storyboard	12
Sequence	12
Structure	14
Through the Life Cycle	17
Types of System	18
Scenarios for Systems—Not Necessarily Software	19
Military Operations	19
Film-Making	20
Business ‘Systems’ and Processes	20
The Way Ahead	21
Keywords	21
References	21
Recommended Reading	22

#### CHAPTER 2 SCENARIO-BASED APPROACHES

Overview: The Crews Scenario Framework	25
The Framework	26
The Scenario Approaches Described in this Book	28
Conclusion	32
Keywords	32
References	32

## PART II

### SCENARIOS THROUGH THE SYSTEM LIFE-CYCLE: TECHNIQUES

#### CHAPTER 3 SCENARIOS IN REQUIREMENTS DISCOVERY

Applicability	39
Position in the Life Cycle	40
Key Features	40
Strengths	40
Weaknesses	41
What Is a Requirements Scenario?	41
A Business Event	42
Normal Case Scenario	43
Identifying Alternative Cases	49
Identifying Exception Cases	50
What-If Scenarios	51
From Scenarios to Atomic Requirements	51
Keeping Track of the Investigation	54
Who Produces the Scenarios?	54
Techniques for Building Scenarios	55
Text Scenarios	55
Story Boards	56
Scenario Process Models	57
Scenario Playthroughs	58
When to Use Scenarios	58
Keywords	59
References	59

#### CHAPTER 4 SCENARIOS FOR INNOVATION: DEVELOPING SCENARIO-BASED USER NEEDS ANALYSIS (SUNA)

Applicability	61
Position in the Life Cycle	62
Key Features	62
Strengths and Weaknesses	63
Technique	64

## vi CONTENTS

Method Outline	65
The Team	66
Inputs	66
Workshop 1	67
Between the Workshops	69
Workshop 2	71
Worked Example	73
Device Unifying Service	73
Inputs	74
Workshop 1 Preparation	74
Workshop 1—Day 1	74
Workshop 1—Day 2	76
Between the Workshops	76
Workshop 2	76
Post SUNA	76
Comparisons	77
Keywords	79
References	79
Recommended Readings	79
<b>CHAPTER 5</b> <i>RUNNING A USE CASE/SCENARIO WORKSHOP</i>	
<hr/>	
Applicability	81
Position in the Life Cycle	82
Key Features	82
Customer Collaboration	82
Multiple Requirements Models	83
The Iteration Process	83
Strengths	83
Weaknesses	84
Technique	85
A Word about Scenarios	86
Workshop Framework	86
Role Playing	88
Wall Modeling Use Cases: Storyboarding	89
Scenario Walkthroughs: Testing Use Cases	89
Collaboration Patterns	90
Doneness Testing	92
Iterating Through the Requirements	92
Worked Example	93
Workshop Pre-Work	94
Workshop 1: Monday	95
Workshop 2: Tuesday	97
Workshop 3: Wednesday	98
Workshop 4: Thursday	98
Comparisons	99

Keywords	100
References	100
Recommended Reading	101
<b>CHAPTER 6</b> <i>ALTERNATIVE WORLD SCENARIOS TO ASSESS REQUIREMENT STABILITY</i>	
<hr/>	
Applicability	103
Position in the Life Cycle	104
Key Features	104
Strengths	104
Weaknesses	104
Technique	105
Change in Systems	105
Requirements and Assumptions	105
The Effect of Time	106
Assessing Requirements Stability	106
Alternative World Scenarios	106
Generating Scenarios	106
Requirements	107
Assessing Stability	107
Reviewing Requirement Stability	108
Instability Propagation	109
Using the Stability Assessment Results	110
Worked Example	110
Goals/Requirements for MSAW	110
Initial Assessment	111
Results Presentation	113
Propagation	113
Comparisons	115
Keywords	116
References	116
Recommended Reading	117
On ‘Alternative World’ Scenarios	117
On Requirements Engineering	117
<b>CHAPTER 7</b> <i>NEGATIVE SCENARIOS AND MISUSE CASES</i>	
<hr/>	
Applicability	119
Roles in the Life Cycle	119
Key Features	120
Strengths	120
Weaknesses	120
Technique	120
Eliciting Hostile Roles	121
Eliciting Misuse Cases	122



Eliciting Exception Scenarios, Requirements, and Further Use Cases	123
Driving Design	125
Automatic Analysis of Use/Misuse Case Relationships	125
Design Trade-off and Conflict Analyses	126
Generating Acceptance Test Cases	126
Metaphorical Roles	127
Worked Example	128
Defeating Detection	128
Defeating the Handling of Intrusion	129
Comparisons	131
Failure Cases	131
FMEA, FTA, Functional Hazard Analysis, HazOp, and so on	132
No Crystal Ball	133
Abuse Cases	134
Negative Scenarios	135
Obstacles	136
Anti-Scenarios	136
i* and GRL	136
Exceptions	137
The Place of the Negative Scenario or Misuse Case	137
Keywords	138
References	138
Recommended Reading	139

## **CHAPTER 8** *AUTHORING USE CASES*

---

Applicability	141
Position in the Life Cycle	141
Key Features	142
Strengths	143
Weaknesses	143
Technique	143
Why Do We Need Guidance on Authoring Use Cases?	143
Use Case Attributes	144
Other Attributes	146
Guidelines for Authoring Use Cases	147
General Use Case Guidelines	148
Scenario/Use Case Style Guidelines	149
Scenarios/Use Case Content Guidelines	151
Short Example	155
Comparisons	158
Keywords	159

References	159
Recommended Reading	160

## **CHAPTER 9** *SYSTEMATIC SCENARIO WALKTHROUGHS WITH ART-SCENE*

---

Position in the Life Cycle	161
Applicability	161
Key Features	162
Strengths	162
Weaknesses	162
The Art-Scene Process and Environment	162
Art-Scene's Research Provenance	163
The Art-Scene Approach	164
The Structure and Representation of an Art-Scene Scenario	164
The Art-Scene Software Environment	166
Facilitating Scenario Walkthroughs	171
The Scenario Workshop Environment	173
Worked Example	173
Comparisons	176
Keywords	177
References	177
Recommended Reading	178

## **CHAPTER 10** *THE ROLE OF SCENARIOS IN CONTEXTUAL DESIGN: FROM USER OBSERVATIONS TO WORK REDESIGN TO USE CASES*

---

Applicability	180
Position in the Life Cycle	180
Key Features	181
Strengths	181
Weaknesses	185
Technique	185
Techniques and Worked Example	186
Contextual Inquiry	187
Interpretation Sessions	187
Work Modeling	188
Consolidation	190
Visioning	192
Storyboarding	194
User Environment Design	199
Paper Prototyping, Mock-Up Interviews and Initial User Interface Design	202
Using Contextual Design Deliverables to Generate Use Cases and Implementation Models	204

## **viii** CONTENTS

Lessons Learnt **205**  
Comparisons **206**  
Keywords **207**  
References **207**  
Recommended Readings **208**

### **CHAPTER 11** *A SCENARIO-BASED DESIGN METHOD FOR HUMAN-CENTRED INTERACTION DESIGN*

---

Overview **211**  
Applicability **211**  
Position in the Life Cycle **212**  
Key Features **212**  
Strengths **213**  
Weaknesses **213**  
The Method **213**  
    Scenarios in the Method **213**  
    User Stories **214**  
    Conceptual Scenarios **215**  
    Concrete Scenarios **215**  
    Use Cases **218**  
Other Artifacts in the Method **218**  
    Requirements and Problems **218**  
    Scenario Corpus **218**  
    Object Model **219**  
    Design Language **219**  
Processes of the Method **220**  
    Abstraction **220**  
    Design Constraints **221**  
    Formalise Design **221**  
    Specifying Requirements **221**  
    Conceptual Design **222**  
    Prototyping, Envisionment and Evaluation **222**  
    Physical Design **223**  
Summary **223**  
Worked Example **224**  
    The On-Line Partner Lens (OPaL) **225**  
User Story **225**  
    Conceptual Scenario **226**  
    Conceptual Design **229**  
    Use Cases **229**  
    Design Language **229**  
    Conclusion **231**  
Comparisons **232**  
Acknowledgements **233**

Keywords **234**  
References **234**  
Recommended Reading **234**

### **CHAPTER 12** *USE CASE-BASED SOFTWARE DEVELOPMENT*

---

Applicability **237**  
Position in the Life Cycle **238**  
Key Features **238**  
Strengths **238**  
Weaknesses **238**  
Technique and Worked Example **239**  
    The Use Case—Centred Requirements Framework **239**  
Summary and Comparisons **262**  
Keywords **263**  
References **263**  
Recommended Reading **264**

### **CHAPTER 13** *USER STORIES IN AGILE SOFTWARE DEVELOPMENT*

---

Applicability **265**  
Applicability II—Why User Stories? **266**  
Position in Life Cycle **267**  
Key Features **267**  
Strengths **268**  
Weaknesses **268**  
Technique **269**  
    Users: Tellers of Stories **269**  
    Story Characteristics and Typography **271**  
    Splitting Stories **273**  
    Ordering and Completing Stories **273**  
    Confirming Stories **274**  
    Communication—the Never Ending Story **275**  
Worked Example **275**  
Comparisons **278**  
Keywords **279**  
References **279**  
Recommended Reading **279**

### **CHAPTER 14** *USE CASES, TEST CASES*

---

Applicability **281**  
Position in the Life Cycle **281**

Key Features	282
Strengths	282
Weaknesses	282
Background: The V-Model	283
Technique	284
A One-to-Many Mapping	284
Slaying the Hydra	285
Prioritising the Tests	287
Tool Support	288
Do Scenarios Yield Test Cases or Test Case Classes?	289
Worked Example	290
Inputs	290
Input Validation	291
Identifying Classes of Test Cases	292
Generating Classes of Test Cases	294
Selecting Environmental Conditions for Test Cases	295
Comparisons	296
Testers' Domain Knowledge	296
Operational Scenarios	297
User Stories	297
Keywords	297
References	297
Recommended Reading	298

---

#### **CHAPTER 15** *PROJECT STORIES: COMBINING LIFE-CYCLE PROCESS MODELS*

---

Applicability	299
Position in the Life Cycle	299
Key Features	300
Strengths	300
Weaknesses	300
Technique	300
Introduction	300
Muddled Process Levelling	302
Systems Engineering 'in the small'	304
Systems Engineering 'in the large'	307
Combining Models	315
Meta-Process	316
Conclusion	323
Keywords	324
References	324
Recommended Reading	324

### **PART III**

#### *SCENARIOS IN ACTION: CASE STUDIES*

---

#### **CHAPTER 16** *STORY USE AND REUSE IN AUTOMOTIVE SYSTEMS ENGINEERING*

---

Type of Project	329
Applicability	329
Position in the Life Cycle	330
Roles Played by Scenarios	330
Strengths	330
Weaknesses	330
Introduction	330
Automotive Software Development	331
Stories in Automotive Software Development	334
Stories in Requirements Discovery, Negotiation and Communication	334
Stories in Requirements Analysis	336
Identifying Feature Interactions and Conflicts	338
Relating Stories to Features	339
Requirements Recycling by Reusing Stories	340
Lessons Learnt	342
Keywords	345
References	346

---

#### **CHAPTER 17** *USE AND MISUSE CASES IN RAILWAY SYSTEMS*

---

A: Use Cases for Train Control Requirements Discovery	347
Type of Project	347
Applicability	348
Position in the Life Cycle	348
Roles Played by Scenarios	348
Strengths	348
Weaknesses	348
Case Study	349
Background	349
Methods and Tools	350
Approach	350
Lessons Learnt	352
Keywords	353
B: Misuse Cases for a Seats Trade-Off	354
Type of Project	354
Applicability	354

## **X CONTENTS**

Position in the Life Cycle	354
Roles Played by Scenarios	354
Strengths	355
Weaknesses	355
Case Study	355
Lessons Learnt	360
Summary	361
Keywords	362
References	362

### **CHAPTER 18 SCENARIOS IN AIR TRAFFIC CONTROL (ATC)**

---

Applicability	363
Roles in the Life Cycle	364
Key Features	364
Strengths	364
Weaknesses	365
Background	366
The CORA-2 Project	366
The RESCUE Process	366
Scenario Walkthroughs	366
Use Case Modelling	366
Use Case Descriptions	368
Scenario Walkthroughs	370
Workshop Environment	371
Results	372
Requirements versus Time	372
Requirements Validation Using Use Cases	373
Domain-Specific Scenarios	373
Requirement Types	374
Requirements Management	375
Planning Workshops	375
How Many Scenarios Do We Need?	375
The Way Forward with Scenario Walkthroughs	376
Keywords	376
References	376

### **CHAPTER 19 SCENARIOS AND QUALITY REQUIREMENTS IN TELECOMMUNICATIONS**

---

Type of Project	379
Applicability	379
Position in the Life Cycle	380
Roles Played by Scenarios	380
Strengths	380
Weaknesses	380

Case Study	381
Scenarios	382
Quality Requirements	383
Architecturally Significant Requirements (ASRs)	384
Refinement of ASRs	385
Software Architecture	386
Examples in Documenting Specific Quality Characteristics	387
Lessons Learnt	389
Suggestions for Well-Understood Domains	390
Suggestions for Emerging Markets	390
Acknowledgements	391
Keywords	391
References	391

### **CHAPTER 20 SCENARIOS IN RAIL ROLLING STOCK WITH REVEAL**

---

Applicability	393
Position in the Life Cycle	394
Key Features	394
Strengths	394
Weaknesses	395
Case Study	396
Background	396
Our Approach to the Project	398
How Scenarios were Used	401
Postscript to Case Study	408
Lessons Learnt	408
Keywords	409
References	409

### **CHAPTER 21 EVALUATING SCENARIOS BY SIMULATION**

---

Type of Project	411
Applicability	412
Position in the Life Cycle	412
Roles Played by Scenarios	412
Strengths	412
Weaknesses	412
Case Study	413
Background to the Case Study	413
The RE Framework	415
Lessons Learnt	425
Keywords	426
References	426

**PART IV****THE WAY AHEAD****CHAPTER 22** *PUTTING SCENARIOS INTO PRACTICE*

- Which Kind of Scenario, When? **431**  
 Which Scenario Technique, When? Conclusions from Part 2 **432**  
 Who, and Why? **435**  
 Scenarios Everywhere? **436**  
 How Does Project Scale Affect Scenario Usage? **437**  
 Into Practice: Conclusions from the Part 3 Case Studies **439**  
 Do Scenarios Replace Requirements? **440**  
 Getting Started **443**  
 Keywords **444**  
 References **444**

**CHAPTER 23** *TEACHING COMPUTER SCIENTISTS TO MAKE USE*

- Challenges in Teaching Students to Make Use **445**  
 Comparisons—Scenarios for Making Use **446**  
 Using Cases to Teach Scenario-Based Usability Engineering **449**  
   Why Case-Based? **449**  
   The Usability Case Study Library **450**  
   Browsing the Usability Cases **452**  
   Case-Based Learning Activities **454**  
   Student Performance and Reactions **455**  
 Strengths and Weaknesses **458**  
 Discussion and Future Directions **459**  
 Acknowledgements **461**  
 Keywords **461**  
 References **462**  
 Recommended Readings **463**

**CHAPTER 24** *WHAT SCENARIOS (STILL) AREN'T GOOD FOR*

- Continuous Behaviour **465**  
 Very Large Systems **466**  
 Fragmentary Models **466**  
 Episodic, Allusory **467**  
 Domain-Specific? **467**  
 Which Representation? **467**  
 Open-Ended **468**  
 Tacit Knowledge **468**  
 Non-Functional Requirements (NFRs) **468**  
 Summary **468**  
 References **469**

**CHAPTER 25** *THE FUTURE OF SCENARIOS*

- Introduction: Horses for Courses **471**  
 Towards a Framework **471**  
 Representation **471**  
 Process **472**  
 Domain Knowledge **473**  
 COTS **473**  
 Dissemination **473**  
 Summary **474**  
 References **474**

**APPENDIX 1** *SCENARIO-BASED SYSTEM DEVELOPMENT TEMPLATES* **475****APPENDIX 2** *EXERCISES* **499****APPENDIX 3** *ANSWERS TO EXERCISES* **501***GLOSSARY* **507***INDEX* **513**



---

# PREFACE

## *A quoi ça sert?*

*What's it for?*

President Chirac of France,  
on being shown a famous white elephant,  
the Millennium Dome, Greenwich

## Communicating Needs

Much of the recent history of large engineering projects—software or systems—has been a tale of waste, error, mismanagement, over-optimism, and lack of proper planning for likely costs and risks. Projects come in late, over budget, and with miserably reduced functionality. Systems sometimes never work, fail on their first period of operational stress, or are permanently unreliable and costly to maintain. We will not name names, as it is fruitless to play the blame game; indeed, engineering systems badly and passing blame are two sides of the same coin. In any case, it is all too easy to find examples in news reports of the demise of famous projects.

By the way, we do not think this is a software issue: it seems to affect complex systems of diverse kinds. The solution cannot therefore be a matter of finding better software-specific tools and techniques; it must be something that helps master complexity.

People have suggested many possible cures for this disease. Most come down to two things:

- the needs that systems are supposed to fulfil ought to be defined much earlier and far more carefully;
- people on projects ought to be made aware of and become skilled in techniques to define needs adequately.

We think that a critical element that is therefore lacking is communication and, in particular, skill in techniques for communicating needs. With the other authors of this book, we believe that the scenario is one of the most powerful techniques for discovering and communicating requirements and often the first choice for organising them.

On a lighter note, scenarios pass the party test, where the requirements engineer has to explain what he or she does for a living to a stranger in 15 seconds. Where

“I’m a systems, erm, a requirements engineer, and I help to specify complex systems . . .”

gets a glazed look every time; the description

“I get people to tell the stories of what their systems are meant to do, so they build the right thing”

always seems to work (and even arouse interest). Story telling is so obviously sensible that it seems surprising that it has taken so long to become a mainstream engineering activity.

## Scenarios and Requirements

If you are wondering whether we recommend replacing requirements entirely with scenarios in all circumstances, we can at once tell you that we think that distinctly unwise, for several reasons:

- The main strength of scenarios is in telling the story of functional behaviour; it is possible to cover various non-functional aspects with stories, but it is doubtful whether such coverage could ever be comprehensive—even if that were desirable.
- Many engineers, organisations, and standards bodies are strongly attached to traditional requirement forms (like ‘The system shall . . . ’), and if those forms work for those people, they should continue using them—anyway, they may have little choice if they have to comply with standards. People work better with familiar artefacts and work processes, even if these sometimes seem to outsiders to be sub-optimal.
- Making a scenario approach work well often requires flair, experimentation, and the courage to take risks, for example, running active workshops rather than writing up requirements in a back room. The implied style of engineering simply does not suit everybody.
- the needs governing large projects are complex and require a range of information structures including stakeholder and goal models, business rules, algorithms and formal specifications of behaviour, interface definitions (protocols, data structures, hardware connections), and commercial and physical constraints (like cost, size, and weight), many of which cannot be framed as scenarios.

Other vital ingredients of a successful project include

- realistic and supportive managers, including one who champions the project;
- effective training for engineers, that is, practical knowledge that changes their behaviour;
- sufficient contact with stakeholders, whether through traditional meetings and reviews, or through some form of participatory design or inquiry cycle, to ensure that the project is working from valid requirements;
- sufficient openness within teams to enable people to speak out when absurd plans are placed on the table (“test and debug a million lines of safety-critical air traffic control software in three months”).

But these are not all within our scope; scenarios don’t do everything. However, much of the book is in one way or another about helping to ensure sufficient contact with stakeholders, and the book will, we hope, help to inform engineers in a practical way about using scenarios.



## Scope: A Wealth of Purposes and Techniques

In this book, we present a range of scenario techniques from light, sketchy, and agile to careful and systematic. There is no single ‘right way’ to use scenarios; we celebrate diversity in requirements discovery and modelling. There is supposedly a saying among French cooks that the English have only two sauces: brown Windsor soup (salted gravy thickened with flour) and custard (sweetened milk thickened with corn flour). Obviously, if such a thing were true, the English diet would be somewhat monotonous. Happily, there are as many ways of using scenarios as there are French sauces—for every palate, season, and occasion, and like sauces, each basic scenario technique has any number of variations.

It would have been possible while editing to impose a uniform style and ‘voice’ on all the contributed chapters, but while we have arranged for a common chapter structure and cross-references, we have chosen to encourage authors to speak in their own way. This may help readers to see that people—engineers and researchers—come to technical issues from different directions, with their own backgrounds and preconceptions, just as project stakeholders do. No one on a project has a monopoly on truth; a major strength of scenario approaches is that they allow stakeholders to share and own a description of what they want. Indeed, each step of an operational scenario may be the responsibility of a different player.

Equally, there are many kinds of scenario structures, and these may well be applicable in projects of different types. The question of which approach is best for a given type of project is open, and in the final part of this book, we sketch some preliminary answers to it.

What all the scenario techniques described here have in common is the motivation to improve industrial practice, a clearly defined approach which has been applied to projects and has a grounding in theory.

We have taken care to ensure a consistent framework for each contribution. There are no tall claims here for commercial tools; equally, there are no chapters asserting elegant but untried academic hypotheses.

## Structure of This Book

The book is structured as a whole to put across the message that scenarios work and are good for your projects,

**Part I** provides an Overview of the nature and use of Scenarios.

**Part II** looks at how to apply Scenarios through the System Life cycle. It is introduced by an overview of the chapter structure used in this part of the book, and then by two chapters that review what scenarios are and how they are used. Then the chapter authors describe their techniques in their own words, but in a fixed structure, which we hope makes the different approaches easy to compare and contrast. Each chapter includes a Comparisons section to guide the reader to related chapters and to help weave the book into a unified whole. The chapters are supported not only by references to the literature but also by recommendations for further reading.

**Part III** presents industrial experiences of Scenarios in Action: Case Studies. It begins with an overview of the chapter structure used in this part of the book. Then the chapter authors tell their stories in their own words, but again in a structure that we hope will help you to select the experiences most relevant to your projects. Where appropriate, the text is cross-referenced to the techniques described in other chapters.

**Part IV** reasons and speculates a little about the future of Scenarios in The Way Ahead. Chapter 22, Putting Scenarios into Practice, reflects on the lessons learnt from the techniques and case studies in Parts II and III—the book itself serving as the basis for some very preliminary research. Part of the Way Ahead lies in the dissemination of what we already know and in the education of tomorrow’s engineers; this challenge is discussed in Chapter 23, Teaching Computer Scientists to Make Use.

**The Appendices** are designed to help make this a practical guide by explaining the terms used and by providing a set of scenario-based engineering templates to get you started, with simple exercises in their use—and providing answers to the exercises.

---

## BIOGRAPHIES AND PHOTOGRAPHS

### Ian Alexander



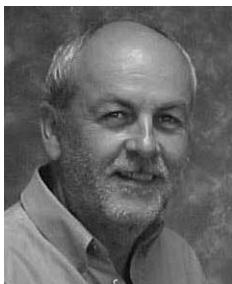
Ian Alexander is an independent consultant specialising in requirements engineering. He is an experienced instructor and has written training courses for a range of organisations. He is the author of the Scenario Plus toolkits for the DOORS requirements tool. His principal research interest is in improving the requirements engineering process by modelling business goals, processes, constraints, and scenarios. He is currently exploring whether Use Cases can assist reuse of specifications for automobile control systems. He was lead author of *Writing Better Requirements* published by Addison-Wesley, 2002. He helps to run the BCS Requirements Engineering Specialist Group and the IEE Professional Network for Systems Engineering. He is a Chartered Engineer.

### Kent Beck



Kent Beck is the founder and director of TRI. He has pioneered patterns for software development, the xUnit family of testing frameworks, the Hot-Draw drawing editor framework, CRC cards, re-factoring, and most recently Extreme Programming. He is the author of *Extreme Programming Explained*, *Planning Extreme Programming*, and *The Smalltalk Best Practice Patterns*. He lives on 20 acres in rural southern Oregon with his wife, five children (one sadly now gone to college), four dogs, two sheep, and a variable number of domestic fowl.

### David Benyon



David Benyon has held the post of Professor of Human–Computer Systems at Napier University, Edinburgh, since 1996. He obtained his MSc at Leicester Polytechnic and his PhD at the Open University, where he designed and implemented courses in human–computer interaction. He contributed to the 1994 book *Human-Computer Interaction* by Preece, J. et al. His research focus is to shift HCI to the idea of navigation of information space. He is working to replace the 1994 book *Designing Interactive Systems* and is conference co-chair for the ACM conference of the same name, DIS2004. He has attracted well over €1 m of funding to support his research and published over 100 conference and journal papers.

## David Bush



David is completing an engineering doctorate at University College London, as a mature student, examining the application of Goal Based Requirements approaches to large-scale industrial projects. He has significant experience in systems engineering and project Management in IT, aerospace, and defence projects and is currently a Principal Engineer in the UK National Air Traffic Services Ltd, working in the development of systems, software, and safety engineering techniques. He is Secretary to the British Computer Societies Requirements Engineering Special Interest Group, a member of the Management Committee of the UCL Centre for Systems Engineering, a Chartered Engineer, and a member of the Chartered Management Institute.

## John Carroll



John M. Carroll is the Edward Frymoyer Chair Professor of Information Sciences and Technology at the Pennsylvania State University. His research interests include methods and theory in human–computer interaction, particularly as applied to networking tools for collaborative learning and problem solving. He has written or edited 14 books, including *Making Use* (MIT Press, 2000), *HCI in the New Millennium* (Addison-Wesley, 2001), *Usability Engineering* (Morgan-Kaufmann, 2002, with M.B. Rosson), and *HCI Models, Theories, and Frameworks* (Morgan-Kaufmann, 2003). He serves on nine editorial boards for journals, handbooks, and series; he is a member of the US National Research Council’s Committee on Human Factors and editor-in-chief of the *ACM Transactions on Computer-Human Interactions*. He received the Rigo Career Achievement Award, from ACM (SIGDOC), and the Silver Core Award from IFIP and was elected to the CHI Academy. In 2003, he received the CHI Lifetime Achievement Award from ACM.

## Andrew Farncombe



Andrew Farncombe has a first class honours degree and spent his early career in the software industry. He subsequently moved into the defence and aerospace sector where he held a number of senior technical and management positions including that of technical director and where he led the codification of systems engineering knowledge and experience for one of the groups as a whole. At John Boardman Associates, he has applied systems engineering to the aerospace and transportation industries. Andrew is Visiting Professor of Systems Engineering at Cranfield University.

## Chris Fowler



Chris Fowler has a degree in psychology and sociology and a Ph.D in cognitive psychology. From 1977 to 1990, he worked in various HE Institutes, undertaking teaching and research into cross-cultural psychology, research methods, human–computer interaction, and the use of computers in education. In 1990, he joined the Human Factors Division of BT Labs. Whilst at BT, he worked on introducing human factors into various software design methodologies, created and managed the Education & Training Research Group, and set up BT’s Asian Research Centre in Malaysia. In April 2002, he became Professor and Director of Chimera, a new institute of socio-technical innovation and research set up at the University of Essex with support from BT.

## Ellen Gottesdiener



Ellen Gottesdiener is Principal Consultant of EBG Consulting, Inc. She works with project teams to help them explore requirements, shape their development processes, and collaboratively plan their work. Before becoming a consultant, Ellen had a 13-year career with CIGNA Corp. as a developer, analyst, trainer, and project manager. She is a Certified Professional Facilitator and an expert in using facilitated workshops in software development projects for developing project charters, defining requirements, and conducting retrospectives. She also presents seminars and advises on industry conferences. Ellen has written numerous articles on requirements, facilitated workshops, methods, and modelling. She is the author of *Requirements by Collaboration: Workshops for Defining Needs* (Addison-Wesley, 2002).

## Peter Haumer



Dr. Peter Haumer is a content developer for the IBM Rational Unified Process product platform. Currently, he is working as the content architect for the future generations of IBM’s integrated process architecture. Before joining the RUP team, he worked as a Senior Professional Services Consultant for IBM’s Rational Software Brand. He assisted and coached customers on how to be successful with the Rational Unified Process platform and Rational tools, performing on-site consulting and providing training courses. His areas of work include requirements management, object-oriented analysis, and design for enterprise application architectures, as well as Software process implementation. He is also a member of Rational’s steering committees for Model-Driven Development, Software Process Adoption, as well as Business Modelling, Requirements Management, and Rational XDE education. Before joining Rational, he worked in basic research in the areas of requirements engineering and flexible CASE tool architectures.

## Karen Holtzblatt



Dr. Karen Holtzblatt, President and CEO of InContext Enterprises, is the co-developer of the customer-centred process Contextual Design. She originated this approach to field data collection and pioneered its introduction into working product design and engineering teams. In 1992, Karen Holtzblatt and Hugh Beyer founded InContext to provide design and consulting services to clients backed by the Contextual Design method. Their book *Contextual Design: Defining Customer Centered Systems*, published by Morgan Kaufmann, is a key reference for anyone doing or teaching customer-centred design. InContext works with leaders in the technology industry, including SAP, Microsoft, Hewlett-Packard, IBM, Novell, Motorola, Nokia, Thomson Corporation, and others.

## Frank Houdek



Frank Houdek is a senior researcher and project leader at the DaimlerChrysler research centre in Ulm, Germany. After finishing his Ph.D. in the field of systematic process improvement and empirical software engineering, he worked in different requirements engineering research projects, and he is involved in technology transfer activities in passenger car and commercial vehicles development. His research interests are requirements engineering processes and requirements recycling. He is an IEEE CS member and part of the steering group of the Requirements Engineering Group of the German Computer Science Society.

## Pericles Loucopoulos



Pericles Loucopoulos holds the chair of Information Systems Engineering at the University of Manchester Institute of Science & Technology (UMIST), in Manchester, UK, where he has worked since January 1984, following a period of many years in industry. His research interests focus on the provision of information processing systems that support large, complex, and dynamic organisational systems. To this end, his research addresses both systems engineering issues and issues relating to organisational objectives, strategy, and business processes. He is co-editor-in-chief of the *Journal of Requirements Engineering* and also serves on editorial boards of four other international journals. He is the co-author of five books, the co-editor of one book, and the author or co-author of over 100 journal and conference papers.

## Catriona Macaulay



Dr Catriona Macaulay's two sons sometimes let her go for long enough to allow her to play at being Programme Leader of the new BSc honours programme in Interactive Media Design at Dundee University/Duncan of Jordanstone College of Art and Design. Her research interests are in the areas of experience design/human-computer interaction, design ethnography, and interactive soundscape design. She also works as a user experience design consultant in industry. Mostly, though she builds pirate ships out of Lego, many of which have met tragic ends because of the large number of crocodiles, witches, and dragons they encounter.

## Neil Maiden



Neil Maiden is Professor of Systems Engineering and Head of the Centre for Human-Computer Interface Design, an independent research department in City University's School of Informatics. He has been directing interdisciplinary research in requirements engineering for 15 years and has worked on numerous EPSRC- and EU-funded research projects. He has over 100 refereed academic publications in journals, conferences and workshops.

## Alistair Mavin



Alistair Mavin is a requirements engineer with Praxis Critical Systems Limited, a UK-based company specialising in requirements, systems, software, and safety engineering. He has undertaken requirements engineering projects in a range of industries including defence, aerospace, rail, automotive, and local government. He has been involved in a number of capability enhancement projects, which have the aim of improving engineering capability within the client organisation. He previously worked in the Centre for HCI Design at City University in London, where he was involved in research and consultancy in requirements engineering.

## Suzanne Robertson



Suzanne Robertson is a principal and founder of the Atlantic Systems Guild. Suzanne is co-author of *Mastering the Requirements Process* (Addison-Wesley 1999). Current work includes research and consulting on stakeholders and all aspects of requirements. The product of this research is *Volere*, a complete requirements process and template for assessing requirements quality and for specifying requirements. She is editor of the requirements column in IEEE Software magazine.

## Con Rodi



Con Rodi retired from the United States Air Force after a 30-year career as a fighter pilot and communications officer. He is now a graduate student at Virginia Tech pursuing a doctorate in Computer Science. He has bachelor's and master's degrees in computer science from the University of Utah and Stanford University respectively. Con's research interests include collaborative systems, case studies supporting scenario-based design, and weblogs as they relate to community computing.

## Perminder Sahota



Parm is a systems engineer with Praxis Critical Systems. He has experience with carrying out requirements engineering work in the transport and aerospace domain. He has specific skills in requirements elicitation, analysis and management, use case and scenario modelling, and dependency modelling and object oriented analysis and design. Parm holds a degree in computer science from City University, where he specialised in requirements engineering and object-oriented analysis and design.

## Mary Beth Rosson



Mary Beth Rosson is Professor of Information Sciences and Technology at Pennsylvania State University. Her research interests include scenario-based design and evaluation; the use of network technology to support collaboration, especially in learning contexts; and the psychological issues associated with the use of high-level programming languages and tools. She is co-author of *Usability Engineering: Scenario-Based Development of Human-Computer Interaction* (Morgan Kaufmann, 2002) and author of *Instructor's Guide to Object-Oriented Analysis and Design with Application* (Benjamin Cummings, 1994), as well as numerous articles, book chapters, and tutorials. Dr. Rosson is active in both ACM SIGCHI and ACM SIGPLAN, serving in numerous technical programs as well as in conference organisation roles for the CHI and OOPSLA annual conferences.



**Camille Salinesi**

Dr. Camille Salinesi is a senior lecturer of computer science in the Department of Mathematics and Informatics at the University of Paris 1 Panthéon—Sorbonne. He does research in information systems engineering, process engineering, and requirements engineering and has published more than 20 referred papers on use case & scenario elicitation and authoring, and on information system evolution. Dr Salinesi is an active animator for the RE community: he was involved in fundamental research projects such as NATURE and CREWS; he has organised a number of conferences and has been guest editor for several journals. He belongs to the RE discussion group of AFIS (French Association of Systems Engineering). He used to be a project leader and RE consultant in projects with French companies.

**Juha Savolainen**

Juha Savolainen is a researcher and project manager at the Nokia Research Center (NRC) in Helsinki, Finland. He assists Nokia's business units to create innovative products and solutions for complex problems. His current work involves research and consulting in requirements engineering, product line development, and software architecture. He loves cooking, inventing new mobile applications while traveling, and watching Finnish ice hockey.

**Ramin Tavakoli**

Ramin Tavakoli Kolagari is writing his PhD thesis on requirements engineering and product lines at the DaimlerChrysler Research Centre in Ulm, Germany. In addition, he is engaged in requirements engineering projects and in technology transfer activities for commercial vehicles development. His research interests are—in addition to the topics covered by his PhD—software engineering for embedded systems, agile software development, and category theory.

## Joy Van Helvert



Joy van Helvert joined BT in 1995 after an IT career in government, and in the last seven years, has worked at a senior professional level in a socio-technical research group specialising in E-learning. In April 2002, the research group transferred to the University of Essex to become Chimera—Institute for socio-technical innovation and research. Her work within the group has included formal and organisational learning research and the development of scenario-based methodologies that have been used successfully as a consultancy product and on European collaborative projects (5<sup>th</sup> Framework and EURESCOM). She is an experienced facilitator and is currently researching cross-cultural communication using ICT.

## David West



David West is a Professor at New Mexico Highlands University (Ph.D., University of Wisconsin, 1988). He teaches systems analysis and design, introduction to business and informational systems, informational modelling and databases, website authoring and management, and enterprise information modelling and databases. He has had more than 23 refereed articles and/or invited appearances at academic and professional conferences, and multiple presentations at national and international conferences. West has worked as a consultant to more than 50 corporate clients (many of them Fortune 100 companies), as well as international clients in India.

## Thomas Zink



Thomas Zink was with DaimlerChrysler research for two years helping in projects applying new requirements engineering approaches. His research focus was on use cases and stories for requirements recycling in a product family context. Thomas recently joined Nokia at the product creation site in Ulm/Germany.

PART



---

*OVERVIEW*



---

# INTRODUCTION: SCENARIOS IN SYSTEM DEVELOPMENT

**Ian Alexander**

*Scenario Plus, London, UK*

---

**S**CENARIOS ARE a powerful antidote to the complexity of systems and analysis. Telling stories about systems helps ensure that people—stakeholders—share a sufficiently wide view to avoid missing vital aspects of problems. Scenarios vary from brief stories to richly structured analyses, but are almost always based on the idea of a sequence of actions carried out by intelligent agents. People are very good at reasoning from even quite terse stories, for example detecting inconsistencies, omissions, and threats with little effort. These innate human capabilities give scenarios their power. Scenarios are applicable to systems of all types, and may be used at any stage of the development life cycle for different purposes.

---

## CONTEXT

---

Scenarios are simple, human things. This book reveals that there are many possible variations on the theme, and the Scope section below introduces some of the concepts; but the basic idea is just a story: someone does this, someone else does that:

*The driver walks towards the car and presses his key.*

*The car recognises the driver, unlocks the doors, and adjusts the driving seat, steering wheel, radio, and mirrors to the driver's preferred settings.*

“Scenarios are arguably the starting point for all modelling and design” (Sutcliffe 2003). Since systems either do something that somebody wants, or are shelfware, and scenarios describe how to do things, it seems hard to disagree with the idea that scenarios are the way to begin a development—if not also to continue, as several authors in this book argue.

More and more modelling notations are being invented—I saw one in a new gadget being shown off at a trade fair, and it had no fewer than 26 symbols that somebody believed

were necessary for requirements analysis—and no end to the madness is in sight. I have no idea how the developers of such things imagine that ordinary people are going to use anything so intricate, and so far removed from life. I do have a lively idea of the blank looks I'd get from practical engineers if I tried to pull any such trick on them.

The use of the narrative scenario in engineering seems in one way to be a kind of reaction against too much technology, too fast. There is no need to be a Luddite to wonder what is being missed in the race to construct ever more complex, formal, and unfamiliar models for ever more risky projects.

Scenarios allow us to take a backward glance. They use a simple, traditional activity—storytelling—to provide a vital missing element, namely a view of the whole of a situation. And they do this at low cost, at least once people are trained in their use (Sommerville and Sawyer 1997).

The scenario perspective looks at what people do and how people use systems in their work, with concrete instead of abstract descriptions, focus on particular instances rather than generic types, being work- and not technology-driven, open-ended and fragmentary rather than complete and exhaustive, informal, rough, and colloquial instead of formal and rigorous, and with envisioned rather than specified outcomes (Carroll 1995).

Analysis means ‘dissolving [into component particles]’, which is fine and very necessary; but it also means looking at details, which as everyone knows is a way of not seeing the wood for the trees. Engineers love analysis and design: our profession’s occupational hazard is diving into detail, ignoring the people involved, and what they may want.

Using scenarios in analysis is thus paradoxical. Analysis is about refinement, precision, and completeness with respect to the parts of a problem. But scenarios are basically holistic. Whether in terse and summary form, or written out at length in a carefully studied sequence—or even in a complex analytical framework with multiple paths ingeniously fitted together—the scenario is in essence, a single thing that conveys a human meaning. And that meaning is enhanced by the reader from her own experience; the story told in the written scenario slips effortlessly into the context of the network of meaning in the reader’s mind.

“What are you doing?” sobbed the Djinni.

“I’m throwing you back into the sea”, said the Fisherman.

“Let me out of this bottle” wailed the desperate Djinni, “and I’ll make you richer than King Solomon”.

“You are a tricky Djinni”, answered the Fisherman. “You deserve the same fate as the King in the story of The King and the Doctor.”

“What story is that?” inquired the Djinni.

*The Djinni and the Fisherman, in The Thousand and One Nights (850 AD onwards)*

Stories are quite insistent on one point: a tale is not over until it’s finished in every detail. The Djinni is not just playing for time by exploring side issues: he’s thinking out other options and tricks that might result in a better outcome—from his dark and devious point of view. It goes without saying that the exploration is by storytelling, and the Fisherman has to use all his cunning to outwit his immensely powerful opponent. Scenario-based techniques such as searching for Exceptions, Functional Hazard Analysis, and Misuse Cases (*see* Chapter 7) make use of the power of storytelling to explore likely