

<<软件项目管理>>

图书基本信息

书名：<<软件项目管理>>

13位ISBN编号：9787111305378

10位ISBN编号：711130537X

出版时间：2010年6月

出版时间：机械工业出版社

作者：(英)Bob Hughes,(英) Mike Cotterell

页数：378

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

Preparing the fifth edition of this book has reminded us that project management is not just a crucial element in successful software and IT development, but is also a fascinating topic in its own right. It is an intriguing mixture of the technical and the very human, of the rational and also the intuitive. Initially we offered this topic as an ancillary discipline for software engineers and IT practitioners. We have, however, become increasingly convinced that the discipline should have a more central role: that the question of how systems are implemented is a vital one to be asked at the same time as that of what a system is to do. Not many software books have lasted as long as this one. Clearly the principles of project management are less transient than those of software design and implementation, which have gone through some very major developments over recent years. However, project management has not been immune from change. One development has been the growth in project management bodies of knowledge such as those of the Project Management Institute (PMI) in the United States and the Association for Project Management (APM) in the United Kingdom. There has also been the development of project management standards such as PRINCE2. These developments are to be welcomed as externalizing and codifying good practice - indeed we have included an appendix on PRINCE2. However, we have resisted becoming a 'PMI' book or a 'PRINCE2' book. Partly this is because we believe that software project management, while incorporating all the key elements of generic project management, also has to deal with the peculiar problems associated with creating software. These include the relative intangibility of software, its extreme malleability, the intimate relationship it has with the systems within which it is embedded, and its sheer complexity. We also wanted to avoid means-end inversion where there was a focus on the recall of specific terminology and procedural detail at the expense of an understanding of underlying concepts and purpose. One new development that has been taken on board has been the growing awareness that a project is rarely an isolated activity but is almost always part of a broader programme of work aimed at meeting organizational and business objectives. There are also agile approaches, such as extreme programming, which have been a timely reminder that software development is an intensely human activity. In contrast to this emphasis on the highly productive, highly interactive co-located team, there is also a growth of dispersed or virtual projects where all or part of the development team is in another country or even continent. We noted these developments in previous editions but have expanded their treatment in this one. This greater emphasis on development team dynamics has led to the creation of a chapter devoted solely to these topics. One major problem has been the conflict between a desire to include all the topics that our reviewers would like to see and the desire for a concise volume that avoids 'bloating'. Sometimes there are topics and standards which appear to be current and of which one feels people should be aware. On closer inspection, the material for various reasons is less useful or relevant than one hoped. In this edition we have dropped an appendix on the British standard BS6079. This is because the new version of this has become what is essentially a general advisory guide on project management practice. As such it duplicates material already covered in this book. Some individual topics have also been dropped because it was felt that they really needed a deeper treatment better conveyed by a more specialist publication than this one: the internal rate of return (IRR) in project evaluation and the Hofstede analysis of national cultural characteristics are examples. In general, though, we have erred on the side of caution in retaining topics. It seems a long time since the first, rather slim, edition published in 1995. As novice authors, Cotterell and Hughes were very indebted to Dave Hatter and Martin Campbell Kelly who had a huge influence on the style of the book. Dave Hatter in particular emphasized the need for each chapter to have clear learning objectives: ideally the reader should finish the chapter feeling they had learnt a new skill. He also instilled the need to explain things clearly - to feel confident in using simple words to explain things that might at first appear complicated. We are aware that we have not always lived up to these values - and have been taken to task by our students and teachers from other institutions who have kindly acted as reviewers.

<<软件项目管理>>

内容概要

本书是经典的项目管理课程教材。

本版延续上一版清晰、易懂的叙述风格，采用步进式策划方法逐一分析了软件开发的各个环节，并通过丰富的实例和练习来阐明实践过程中软件项目管理的原则。

本书不仅适合作为计算机及相关专业的本科生和研究生的教材，而且适合于软件项目管理人员和软件开发人员阅读，还特别适合作为BCS/ISEB专业考试的参考书。

为了涵盖软件项目管理的新进展，本版进行了全面更新，新增和扩展的主题如下： 沟通策划

- 。 敏捷方法，包括XP（极限编程）、Scrum 和 DSDM。
- COCOMO II。
- 项目组合管理。
- 新增一章，主要是关于合作、分散和虚拟团队管理。
- 职业道德规范。

作者简介

Bob Hughes曾在产业界和高等教育界担任各种职务，现在是英国布莱顿大学信息管理学院信息系统的负责人。

他还是BCS/ISEB项目管理认证考试的主考官和相关培训课程的主讲老师。

书籍目录

Preface Guided tour Acknowledgements I Introduction to software project management 1.1 Introduction 1.2 Why is software project management important? 1.3 What is a project? 1.4 Software projects versus other types of project 1.5 Contract management and technical project management 1.6 Activities covered by software project management 1.7 Plans, methods and methodologies 1.8 Some ways of categorizing software projects 1.9 Stakeholders 1.10 Setting objectives 1.11 The business case 1.12 Project success and failure 1.13 What is management? 1.14 Management control 1.15 Conclusion Annex 1 Contents list for a project plan 1.16 Further exercises 2 Project evaluation and programme management 2.1 Introduction 2.2 A business case 2.3 Project portfolio management 2.4 Evaluation of individual projects 2.5 Cost-benefit evaluation techniques 2.6 Risk evaluation 2.7 Programme management 5.8 Managing the allocation of resources within programmes 2.9 Strategic programme management 2.10 Creating a programme 2.11 Aids to programme management 2.12 Some reservations about programme management 2.13 Benefits management 2.14 Conclusion 2.15 Further exercises 3 An overview of project planning 3.1 Introduction to Step Wise project planning 3.2 Step O: Select project 3.3 Step 1: Identify project scope and objectives 3.4 Step 2: Identify project infrastructure 3.5 Step 3: Analyse projec characteristics 3.6 Step 4: Identify project products and activities 3.7 Step 5: Estimate effort for each activity 3.8 Step 6: Identify activity risks 3.9 Step 7: Allocate resources 3.10 Step 8: Review/publicize plan 3.11 Steps 9 and 10: Execute plan/lower levels of planning 3.12 Conclusion 3.13 Further exercises 4 Selection of an appropriate project approach 4.1 Introduction 4.2 Build or buy? 4.3 Choosing methodologies and technologies 4.4 Choice of process models 4.5 Structure versus speed of delivery 4.6 The waterfall model 4.7 The spiral model 4.8 Software prototyping 4.9 Other ways of categorizing prototypes 4.10 Incremental delivery 4.11 Agile methods 4.12 Atern/Dynamic Systems Development Method 4.13 Extreme programming (XP) 4.14 Managing iterative processes 4.15 Selecting the most appropriate process model 4.16 Conclusion 4.17 Further exercises 5 Software effort estimation 5.1 Introduction 5.2 Where are estimates done? 5.3 Problems with over- and under- estimates 5.4 The basis for software estimating 5.5 Software effort estimation techniques 5.6 Bottom-up estimating 5.7 The top-down approach and parametric models 5.8 Expert judgement 5.9 Estimating by analogy 5.10 Albrecht function point analysis 5.11 Function points Mark II 5.12 COSMIC full function points 5.13 COCOMO 13: a parametric productivity model 5.14 Conclusion 5.15 Further exercises 6 Activity planning 6.1 Introduction 6.2 The objectives of activity planning 6.3 When to plan 6.4 Project schedules 6.5 Projects and activities 6.6 Sequencing and scheduling activities 6.7 Network planning models 6.8 Formulating a network model 6.9 Adding the time dimension 6.10 The forward pass 6.11 The backward pass 6.12 Identifying the critical path 6.13 Activity float 6.14 Shortening the project duration 6.15 Identifying critical activities 6.16 Activity-on-arrow networks 6.17 Conclusion 6.18 Further exercises 7 Risk management 7.1 Introduction 7.2 Risk 7.3 Categories of risk 7.4 A framework for dealing with risk 7.5 Risk identification 7.6 Risk assessment 7.7 Risk planning 7.8 Risk management 7.9 Evaluating risks to the schedule 7.10 Applying the PERT technique 7.11 Monte Carlo simulation 7.12 Critical chain concepts 7.13 Conclusion 7.14 Further exercises 8 Resource allocation 8.1 Introduction 8.2 The nature of resources 8.3 Identifying resource requirements 8.4 Scheduling resources 8.5 Creating critical paths 8.6 Counting the cost 8.7 Being specific 8.8 Publishing the resource schedule 8.9 Cost schedules 8.10 The scheduling sequence 8.11 Conclusion 8.12 Further exercises 9 Monitoring and control 9.1 Introduction 9.2 Creating the framework 9.3 Collecting the data 9.4 Visualizing progress 9.5 Cost monitoring 9.6 Earned value analysis 9.7 Prioritizing monitoring 9.8 Getting the project back to target 9.9 Change control 9.10 Conclusion 9.11 Further exercises 10 Managing contracts 10.1 Introduction 10.2 Types of contract 10.3 Stages in contract placement 10.4 Typical terms of a contract 10.5 Contract management 10.6 Acceptance 10.7 Conclusion 10.8 Further exercises 11 Managing people in software environments 11.1 Introduction 11.2 Understanding behaviour 11.3 Organizationbehaviour: a backgrou

nd 11.4 Selecting the right person for the job 11.5 Instruction in the best methods 11.6 Motivation 11.7
The Oldham Hackman job characteristics model 11.8 Stress 11.9 Health and safety 11.10 Some ethical and
professional concerns 11.11 Conclusion 11.12 Further exercises 12 Working in teams 12.1 Introduction
12.2 Becoming a team 12.3 Decision making 12.4 Organizational structures 12.5 Coordination
dependencies 12.6 Dispersed and virtual teams 12.7 Communication genres 12.8 Communication plans
12.9 Leadership 12.10 Conclusion 12.11 Further exercises 13 Software quality 13.1 Introduction 13.2
The place of software quality in project planning 13.3 The importance of software quality 13.4 Defining software
quality 13.5 ISO 9126 13.6 Product versus process quality management 13.7 Quality management systems
13.8 Process capability models 13.9 Techniques to help enhance software quality 13.10 Testing 13.11
Quality plans 13.12 Conclusion 13.13 Further exercises Appendix A PRINCE2 - an overview Appendix B
Answer pointers Further reading Index

章节摘录

插图：The feasibility study assesses whether a project is worth starting -that it has a valid business case. Information is gathered about the requirements of the proposed application. Requirements elicitation can, at least initially, be complex and difficult. The stakeholders may know the aims they wish to pursue, but not be sure about the means of achievement. The developmental and operational costs, and the value of the benefits of the new system, will also have to be estimated. With a large system, the feasibility study could be a project in its own right with its own plan. The study could be part of a strategic planning exercise examining a range of potential software developments. Sometimes an organization assesses a programme of development made up of a number of projects.

<<软件项目管理>>

编辑推荐

《软件项目管理(英文版·第5版)》：经典原版书库

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>