

PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

Second edition

Bob Hughes (editor)



Computer Weekly readers can get 20% discount
on this book. Enter code **CWB3** at
www.bcs.org/books/projectmanagement2

PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

BCS, THE CHARTERED INSTITUTE FOR IT

Our mission as BCS, The Chartered Institute for IT, is to enable the information society. We promote wider social and economic progress through the advancement of information technology science and practice. We bring together industry, academics, practitioners and government to share knowledge, promote new thinking, inform the design of new curricula, shape public policy and inform the public.

Our vision is to be a world-class organisation for IT. Our 70,000 strong membership includes practitioners, businesses, academics and students in the UK and internationally. We deliver a range of professional development tools for practitioners and employees. A leading IT qualification body, we offer a range of widely recognised qualifications.

Further Information

BCS, The Chartered Institute for IT,
First Floor, Block D,
North Star House, North Star Avenue,
Swindon, SN2 1FA, United Kingdom.

T +44 (0) 1793 417 424

F +44 (0) 1793 417 444

www.bcs.org/contact



PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

Second edition

Bob Hughes (editor)
Roger Ireland, Brian West, Norman Smith,
David I. Shepherd



© 2012 BCS Learning & Development Ltd

All rights reserved. Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted by the Copyright Designs and Patents Act 1988, no part of this publication may be reproduced, stored or transmitted in any form or by any means, except with the prior permission in writing of the publisher, or in the case of reprographic reproduction, in accordance with the terms of the licences issued by the Copyright Licensing Agency. Enquiries for permission to reproduce material outside those terms should be directed to the publisher.

All trade marks, registered names etc. acknowledged in this publication are the property of their respective owners. BCS and the BCS logo are the registered trade marks of the British Computer Society charity number 292786 (BCS).

Published by BCS Learning & Development Ltd, a wholly owned subsidiary of BCS The Chartered Institute for IT, First Floor, Block D, North Star House, North Star Avenue, Swindon, SN2 1FA, UK.
www.bcs.org

ISBN: 978-1-78017-118-0

PDF ISBN: 978-1-78017-119-7

ePUB ISBN: 978-1-78017-120-3

Kindle ISBN: 978-1-78017-121-0

British Cataloguing in Publication Data.

A CIP catalogue record for this book is available at the British Library.

Disclaimer:

The views expressed in this book are of the author(s) and do not necessarily reflect the views of the Institute or BCS Learning & Development Ltd except where explicitly stated as such. Although every care has been taken by the authors and BCS Learning & Development Ltd in the preparation of the publication, no warranty is given by the authors or BCS Learning & Development Ltd as publisher as to the accuracy or completeness of the information contained within it and neither the authors nor BCS Learning & Development Ltd shall be responsible or liable for any loss or damage whatsoever arising by virtue of such information or any instructions or advice contained within this publication or by any of the aforementioned.

Typeset by Lapid Digital Services, Chennai, India.

Printed at CPI Antony Rowe Ltd, Chippenham, UK.

CONTENTS

	Figures and tables	viii
	Preface	ix
	Useful URLs	xi
1	PROJECTS AND PROJECT WORK	1
	Learning outcomes	1
	1.1 Projects	1
	1.2 Successful projects	2
	1.3 Project management	5
	1.4 System development life cycle	6
	1.5 Project management and the development life cycle	12
	1.6 Elements of project management	13
	1.7 Development process models	15
	1.8 The project plan	20
	1.9 The business case	24
	1.10 Implementation strategies	27
	1.11 Post-implementation review	28
	Sample questions	29
	Answers to sample questions	29
	Pointers for activities	30
2	PROJECT PLANNING	32
	Learning outcomes	32
	2.1 Introduction	32
	2.2 Approaches to planning	33
	2.3 Product flow diagram	35
	2.4 Activity planning	37
	2.5 Resource allocation	44
	2.6 Using software tools for planning	49
	Sample questions	50
	Answers to sample questions	50
	Pointers for activities	51
3	MONITORING AND CONTROL	55
	Learning outcomes	55
	3.1 Introduction	55
	3.2 The project control life cycle	55
	3.3 Monitoring progress	56
	3.4 Applying control	58

	3.5 Purpose and types of reporting meetings	59
	3.6 Taking corrective action	60
	3.7 Graphical representation of progress information	63
	Sample questions	67
	Answers to sample questions	67
	Pointers for activities	68
4	CHANGE CONTROL AND MANAGEMENT	70
	Learning outcomes	70
	4.1 Introduction	70
	4.2 Definition of change	71
	4.3 Change management roles and responsibilities	73
	4.4 The change management process	74
	4.5 Configuration management	76
	Sample questions	78
	Answers to sample questions	78
	Pointers for activities	79
5	QUALITY	80
	Learning outcomes	80
	5.1 Introduction	80
	5.2 Definitions of quality	81
	5.3 Quality characteristics	82
	5.4 Quality criteria	82
	5.5 Quality control versus quality assurance	84
	5.6 Quality plan	85
	5.7 Detecting defects	85
	5.8 Dynamic testing	90
	5.9 Evaluating suppliers	93
	5.10 ISO 9001:2008	93
	5.11 Capability maturity models	94
	Sample questions	95
	Answers to sample questions	95
	Pointers for activities	96
6	ESTIMATING	100
	Learning outcomes	100
	6.1 Introduction	100
	6.2 What we estimate and why it is important	100
	6.3 Expert judgement	101
	6.4 Approaches to estimating	103
	6.5 A parametric approach	104
	6.6 Estimating by analogy	108
	6.7 Checklist	109
	Sample questions	110
	Answers to sample questions	111
	Pointers for activities	111

7	RISK	113
	Learning outcomes	113
	7.1 Introduction	113
	7.2 Risk management	114
	7.3 Identifying risks	115
	7.4 Assessing the risk	117
	7.5 Quantitative approaches to risk	118
	7.6 The qualitative approach to project risk assessment	119
	7.7 Deciding the appropriate actions	122
	7.8 Planning, monitoring and control	124
	7.9 Summary	126
	Sample questions	126
	Answers to sample questions	126
	Pointers for activities	127
8	PROJECT ORGANISATION	129
	Learning outcomes	129
	8.1 Introduction	129
	8.2 Programmes and projects	130
	8.3 Identifying stakeholders and their concerns	130
	8.4 The organisational framework	131
	8.5 Desirable characteristics of a project manager	136
	8.6 Project support office	137
	8.7 Project team	138
	8.8 Matrix management	139
	8.9 Team building	141
	8.10 Team dynamics	142
	8.11 Management styles	143
	8.12 Communication methods	144
	8.13 Conclusion	145
	Sample questions	146
	Answers to sample questions	146
	Pointers for activities	146
	Index	149

FIGURES AND TABLES

Figure 1.1	The waterfall model	16
Figure 1.2	An incremental model	18
Figure 1.3	An iterative model	19
Figure 2.1	A product breakdown structure diagram	35
Figure 2.2	A product flow diagram	36
Figure 2.3	Activity on arrow network	37
Figure 2.4	Activity on node network	38
Figure 2.5	A network activity fragment with activity durations	39
Figure 2.6	Earliest start and finish days (ES = 'earliest start'; EF = 'earliest finish')	40
Figure 2.7	Latest start and finish dates	41
Figure 2.8	Layout of an activity box	42
Figure 2.9	Activity box for 'Do C'	42
Figure 2.10	The activity span	43
Figure 2.11	Canal Dreams project: write/test software activity (SDes = System designer, Prog = Programmer, w = week)	45
Figure 2.12	A resource histogram for each resource type	46
Figure 2.13	Gantt chart	48
Figure 3.1	A Gantt chart that has been updated with actual progress	64
Figure 3.2	A cumulative resource chart	65
Figure 3.3	Earned value graph	66
Figure 3.4	A Gantt chart that has been updated with actual progress up to week 4	69
Figure 5.1	A simplified V model	86
Figure 7.1	Risk management framework	115
Figure 7.2	Probability impact grid	122
Figure 7.3	Risk register	124
Figure 7.4	Risk record	125
Table 1.1	Net present value calculation	26
Table 1.2	Payback period calculation	27
Table 2.1	Numbers of each resource type needed in each week	45
Table 6.1	Relationship between top-down/bottom-up and the three main estimating approaches	109
Table 7.1	Mapping qualitative and quantitative assessments of risk probability	120
Table 7.2	Mapping qualitative and quantitative assessments of cost impact	120
Table 7.3	Mapping qualitative and quantitative assessments of scope impact	121
Table 7.4	Mapping qualitative and quantitative assessments of time impact	121
Table 8.1	Typical matrix organisation	140

PREFACE

This book is designed to give a practical introduction to IT project management principles and techniques. The first edition was intended more specifically to support candidates for the BCS Foundation Certificate in IS Project Management. This edition still supports this qualification, but updates some of the material and broadens its practical application.

Taking this qualification is not itself a daunting challenge: it consists of an hour-long 40-question multiple choice examination. However, the intention was never just to help cram for an examination. While there might be an immediate concern to pass a test, for most people the more important motivation was to gain guidance on planning and managing an IT project. The text was designed to help those from an IT practitioner background who were beginning to take on project management responsibilities. However, it is not just IT developers who have to grapple with IT projects: users often have to bear the brunt of IT-driven business change and have their own project responsibilities that can have a decisive impact on project success. An additional aim was to give these IT users some insights into IT project management issues. The text therefore goes beyond simply helping people to tick the right boxes in a test and aspires to support novice IT project leaders in their place of work.

When learning about any new topic, a good starting point is a text which provides a simple explanation of the basics. This can provide a foundation that allows you to go on and grasp more advanced concepts. A measure of the success of the first edition was that it started to be used for purposes for which it was not primarily designed. The text started to appear in the reading lists of courses where the overall syllabus was broader and the assessment more demanding than the foundation certificate. One example of this was the BCS Higher Education Qualification Diploma in Project Management (an 'academic' BCS qualification comparable to a UK university award and taken mainly by overseas candidates). Some of the changes for the second edition have been made in response to this unexpected use. The focus still remains on the foundations but care has been taken to provide links to other, more detailed project management material. Wherever possible, alternatives to the terminology we have used are provided for techniques and concepts to allow easier cross-reference to other bodies of knowledge. For example, 'steering committee', 'project board' and 'project management board' all refer to largely the same concept in project management.

We have put in links to further material using a  symbol for those who want to explore a topic more deeply. Some material in the basic text goes beyond what is needed for the BCS foundation syllabus and these have been marked with a  symbol to indicate an 'advanced topic'.

It may be heretical to say this in a project management text, but successful projects do not depend only on good project management and some of the links provided are to material on complementary disciplines that can assist positive project outcomes. (The BCS Diploma in Business Analysis to which the Foundation Certificate in IS Project Management can contribute supports this view.)

The BCS Foundation Diploma syllabus has been very stable in recent years and there have been no massive changes in content in the new edition. Some inadvertent gaps have been filled: for example, more has been added on the question of deciding whether to build or buy an IT solution. A suggestion to acknowledge the growing interest in agile approaches has been incorporated. The main principles of project agility – such as the focus on iterations and increments in project delivery – had been well-established before the term ‘agile’ was adopted in the context of software development, so it has been easy to signpost those elements of our approach that dealt with them. It was also suggested that more quantitative approaches to risk assessment be discussed, and this has been done.

The BCS Foundation Certificate course’s focus is different from that of PRINCE2. PRINCE2 is a UK government-sponsored set of procedures for managing major projects. In our view, it effectively describes an information system for a project that allows it to be run in a controlled and efficient manner. Although PRINCE2 is really an administrative standard that will tell you what decisions need to be taken, when they need to be taken and by whom, it offers little guidance about how decisions are made: it does not claim to be a set of project management principles and techniques.

The BCS syllabus can also be distinguished from more general introductory courses on project management by its focus on IT projects. While the core elements of project management remain the same regardless of the type of project, there are some significant differences in emphasis with IT projects. The description of the IT-focused system development life cycle has already been mentioned, but there are other topics – like testing and the measurement of functionality to support the estimation of system size – which get more attention here than in more general project management courses.

The following people contributed the material for the text:

Norman Smith Chapters 1 and 4
 Bob Hughes Chapters 2 and 6
 Roger Ireland Chapters 3 and part of 8
 Brian West Chapters 5 and part of 8
 David I. Shepherd Chapter 7

Any defects and errors are almost certainly those of the editor, Bob Hughes. Sue McNaughton and Elaine Boyes at the BCS drove the publication project for the first edition along. The original development of the Foundation Certificate as a whole has involved many BCS staff, including Malcolm Sillars, Rebecca Stoddart, Imelda Byrne, Steve Causer and Carol Lewis. Jutta Mackwell was instrumental in the creation of this second edition. Roger Ireland has been a painstaking reviewer, and Karen Greening managed the production of the book from the author’s word-processed manuscript to the final version that appears here.

The book is dedicated, as was the last, to the memory of Jimmy Robertson.

USEFUL URLS

IT Project management qualifications and syllabuses

- BCS Professional certification: Foundation Certificate in IS Project Management <http://certifications.bcs.org/content/ConTab/2>
- BCS Professional certification: Foundation Certificate: Programme and Project Support Office Essentials <http://certifications.bcs.org/content/ConTab/3>
- BCS Higher Education Qualifications: Diploma in IT Project management. An 'academic' examination at university 2nd Year level popular with overseas candidates. www.bcs.org/upload/pdf/dippmsyll.pdf

Agile project management approaches

- DSDM Consortium. This group is responsible for the DSDM Atern agile project management framework www.dsdm.org/
- Scrum Alliance has a set of resources supporting the Scrum agile framework. www.scrumalliance.org/
- **Some professional bodies** – APM and PMI have their own qualifications
- PROMSG: the BCS Project Management Specialist Group www.proms-g.bcs.org
- Association for Project Management: the UK professional body for generic project management (rather than just IT) www.apm.org.uk
- Project Management Institute: US-based professional body www.pmi.org. A UK chapter of PMI exists www.pmi.org.uk
- International Project Management Association. A global umbrella association to which most national project management professional bodies are affiliated <http://ipma.ch>

Planning tools

- Microsoft Project : probably the most widely used project planning tool www.microsoft.com/project/en-us/project-management.aspx
- Oracle Primavera: another, perhaps more industrial, project planning tool (and much else) www.oracle.com/eppm

- Smartsheet is an easy to use tool for small one-off projects where there is a need to do things quick and simple way

Quality

- TickIT: UK initiative to tailor ISO9001 to specifically IT development. www.tickit.org
- Details of the SEI Capability Maturity Model (CMMI). www.sei.cmu.edu/cmmi/

Estimation and measurement

- UK Software Metrics Association contains useful information and/or links relating to function points of various types. www.ukσμα.co.uk/

Project organisation

- PRINCE2, the UK government- sponsored standard for project management procedures. www.prince-officialsite.com/

General keeping up to date

- Project Management Today, trade magazine www.pmtoday.co.uk/

6 ESTIMATING

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- *the effects of over- and under-estimating;*
- *effort versus duration;*
- *the relationship between effort and cost;*
- *estimates and targets;*
- *use of expert judgement, including its advantages and disadvantages;*
- *the Delphi approach;*
- *top-down estimating*
- *bottom-up estimating;*
- *the use of analogy in estimating.*

6.1 INTRODUCTION

In Chapter 2, we explained how to draw up a plan for a project. One of the things that we did was to allocate an estimated duration to each of the activities to be carried out. This allowed us to calculate the overall duration of the project and to identify when we would need to call upon the services of individuals to carry out their tasks. In this chapter, we will explore further the ways in which these estimates can be produced.

6.2 WHAT WE ESTIMATE AND WHY IT IS IMPORTANT

6.2.1 Effort versus duration

As well as estimating the time from the start to the end of an activity, it is also necessary to assess the amount of effort needed. Duration should not be confused with **effort**. For example, if it takes one worker two hours to clear a car park of snow then, all other things being equal, it takes two workers only one hour. In both cases, the effort is two hours but the activity **duration** is two hours in one case and only one

hour in the other. There can be cases where the duration is longer: for example, where someone only works in the afternoons on a particular task. In fact, a problem is that activities often take longer than planned even though the effort has not increased. This may happen, for instance, when you have to wait for approval from a higher level of management before a job is signed off. This distinction between effort and duration can be particularly important when assessing the probable cost of a project, as on some projects staff costs are governed by the hours actually worked (typically where staff complete timesheets), while on others the costs are governed by the time in which people are employed on the project (even if there is not always work for them to do).

6.2.2 The effects of over- and under-estimating

If effort and duration are under-estimated, the project can fail because it has exceeded its budget or has been delayed beyond its agreed completion date. This may be so even when staff have worked efficiently and conscientiously. Allocating less time and money than is really needed can also affect the quality of the final project deliverables: team members may work hard to meet deadlines but, as a consequence, produce sub-standard work.

On the other hand, estimates that are too generous can also be a problem. If the estimate is the basis for a bid to carry out some work for an external customer, then an excessively high estimate may lead to the work being lost to a competitor. Parkinson's Law ('work expands to fill the time available') means that an excessively generous estimate may lead to lower productivity. If a task is allocated four weeks when it really needs only three, there is a chance that, with the pressure removed, staff will take the planned time.

6.2.3 Estimates and targets

Identifying the exact time it will take to do something is very difficult because, if the same task is repeated a number of times, each instance of the task execution is likely to have a slightly different duration. Take going to work by car. It is unlikely that on any two days this will take exactly the same amount of time. The journey time will vary because of factors such as weather conditions and the pressure of traffic. This means that an estimate of effort or time is really a most likely effort/time with a range of possibilities on each side of it. Within this range of times we can choose a **target** – we can go for an 'aggressive' target which may get the job done quickly, but with a strong possibility of failure, or a more generous estimate which is likely to expand the length of time needed, but have a safer chance of the target being met. The target, if at all reasonable, can become a **self-fulfilling prophecy** – with the commuting example, if you know that you are going to be late you may take steps to speed up, perhaps by taking an alternative route if the normal one is congested. Estimating can thus have a 'political' aspect. Some managers may reduce estimates, either to gain acceptance for a proposed project, or as a means of pressurising developers to work harder. There are clearly risks involved in such an approach, as well as possible ethical issues.

6.3 EXPERT JUDGEMENT

6.3.1 Using expert judgement

Where do you start if you want to produce reasonable estimates? Although estimating is treated as a separate, isolated topic in project management and

information systems development, it in fact depends on the completion of other tasks that provide information for estimates. For a start, you need to know:

- What activities are going to be carried out during the course of the project;
- How much work is going to have to be carried out by these activities.

For example, to work out how long it will take to install some software on all the workstations in an organisation, we need to know approximately how long it takes to install the software on a single workstation and how many workstations there are in the organisation. We may also need to know how geographically dispersed the workstations are. The best person to tell us about these things would be someone familiar with the tasks to be carried out and the environment in which they are done. As a general rule, the best people to estimate effort are those who are experts in the area. As a consequence, most guides to estimating identify **expert opinion** or **expert judgement** as an estimating method.

Although ‘phoning a friend’ can be a very sensible approach, there remains the question of how the experts themselves derive their estimates. There is a possibility that they have their own experts upon whom they can call, but at some point someone must sit down and work out the estimate based on their own judgement – and the likelihood is that they will end up using the analogy approach described below.

The advantages of using expert judgement include the following:

- It involves the people with the best experience of similar work in the past and the best knowledge of the work environment;
- The people who are most likely to be doing the work are involved with the estimating process – they will be more motivated to meet the targets set if they have had a hand in setting them in the first place.

There are, however, some balancing risks:

- The task to be carried out may be a new one of which there is no prior experience;
- Experts can be prone to human error – they may, for example, underestimate the time that they would need to carry out a task in case a larger figure suggests that they are less capable;
- It can be difficult for the project planner to evaluate the quality of an estimate that is essentially someone else’s guess;
- Large, complex tasks may require the expertise of several different specialists.

6.3.2 The Delphi approach

One method that attempts to improve the quality of expert judgement is the **Delphi technique** which originated in the Rand corporation in the USA. There are different versions of this, but the general principle is that a group of experts are asked to produce, individually and without consulting others, an estimate supported by

some kind of rationale. These are all forwarded to a moderator who collates the replies and circulates them back to the group as a whole. Each member of the group can now read the anonymous estimates and supporting rationales of the other group members. They may now submit a revised opinion. Hopefully, the opinions of the experts should converge on a consensus.

The justification for the technique is that it should lead to people's views being judged on their merits and undue deference will not be paid to more senior staff or the more dominant personalities.

6.4 APPROACHES TO ESTIMATING

We are now going to discuss key approaches to estimating. However, first we are going to explain the terms **bottom-up** and **top-down**. Note that these are not specific estimating methods, but describe a way of grouping estimating methods.

6.4.1 Bottom-up

In essence, we break the task for which an estimate is to be produced into component sub-tasks and then break the component sub-tasks into sub-sub-tasks and so on, until we get to elements that we think would not take one or two people more than a week to complete. The idea is that you can realistically imagine what can be accomplished in one or two weeks in a way that would not be possible for one or two months. To get an overall estimate of the effort needed for the project, you simply add up all the effort for the component tasks.

This method is also sometimes called **analytical** or **activity-based estimating**. Some people (especially those who are or who have been software developers) find the name 'bottom-up' confusing because the first part of the process is really top-down!

ACTIVITY 6.1

Which planning product identified in Chapter 2 could be the basis for an initial bottom-up estimate?

A bottom-up estimate is recommended where you have no accurate historical records of relevant past projects to guide you. A disadvantage of the method is that it is very time-consuming as you have, in effect, to draw up a detailed plan of how the project is to be carried out first. It could be argued that you are going to have to do this anyway. However, it may be a very tedious and speculative task if you have been asked for a rough estimate at the feasibility study stage of the project proposal.

ACTIVITY 6.2

You have been asked to organise the recruitment of staff for the new network support centre needed for the Canal Dreams ebooking enhancement. Identify the component activities in this overall task, as you would for the first stage of the bottom-up approach to estimating effort.

6.4.2 Top-down

With the top-down approach, we look for some overall characteristics of the job to be done and, from these, produce a global effort estimate. This figure is nearly always based on our knowledge of past cases.

An example of top-down estimating is when house owners have to make decisions about the sum for which they should insure their house. The question here is the probable cost of rebuilding the house in the event of it being destroyed, for example by fire. Most insurance companies produce a handy set of tables where you can look up such variables as the number of storeys your house has, the number of bedrooms, the area of floor-space, the material out of which it has been constructed and the region in which it is located. For each combination of these characteristics a rebuilding cost will be suggested. The insurance company can produce such tables as it has access to many historical records of the actual cost of rebuilding houses.

This is essentially a top-down approach because only one global figure is produced. In the unhappy case of a fire actually occurring, this figure would not help a builder to calculate how much effort would be needed to dig the foundations, build the walls, put on the roof and all the other individual components of the building operation. A builder may be able to use past experience of the proportion of total costs usually consumed by each type of activity, such as foundation digging.

6.5 A PARAMETRIC APPROACH

The base estimate created when using a top-down approach can be derived in a number of ways. In estimating the costs of rebuilding a house, a **parametric** method was used. This means that the estimate was based on certain variables or parameters (for example, the number of storeys in the house and the number of bedrooms). These parameters can be said to 'drive' the size of the house to be built: you would expect a house with three storeys and five bedrooms to be physically bigger than a bungalow with only two bedrooms. These parameters are therefore sometimes called **size drivers**. You would also expect the three-storey building to need more work, or effort, to build than the bungalow. These parameters are therefore also sometimes called **effort drivers**.

6.5.1 Size drivers and productivity

Earlier we had an example where a technician was allocated the job of installing a new piece of software on every workstation in an organisation. Clearly, the more workstations there are, the bigger the job and the longer its duration. Hence the number of workstations is a size driver and an effort driver for this activity.

ACTIVITY 6.3

Identify the possible size and effort drivers in the Canal Dreams ebooking enhancement for each of the following activities:

- (a) Creating training material for users;
- (b) Analysing business processes;
- (c) Carrying out acceptance tests;
- (d) Writing and testing software.

In order to produce an estimate of effort using this method, we also need a productivity rate. For example, in addition to the number of workstations we would need to know the average time needed to install the software on a single workstation. This time per workstation would be the **productivity rate**. If this rate was 12 minutes per workstation and there were 50 workstations then we could guess the overall duration of the job would be around 50×12 minutes – that is, about ten hours.

The usual way to obtain the productivity rate is from records of past projects. Where these are not available within an organisation, it is sometimes possible to obtain ‘industry’ data that relates not to projects in a single organisation, but to projects in a particular industrial sector. This kind of information can help managers compare the productivity in their organisation with that of others – this is sometimes called **benchmarking**. If they find that they have much lower productivity, this may spur them on to search for more productive ways of working. However, caution needs to be practised if the reason for using industry data is that local project data is missing: there can be large differences in productivity between organisations, because organisations and their businesses are so different.

ACTIVITY 6.4

In the earlier example about the time needed to drive to work, identify:

- (a) the size driver;
- (b) the productivity rate;
- (c) other factors that may cause a variation in the time it takes to get to work.

The additional factors are called **productivity drivers**. A key productivity driver when it comes to developing and implementing IT systems is experience. When putting a figure on how long a technical activity like developing software code is going to take, more experienced estimators will try to find out who will be doing the work.

Productivity drivers vary from activity to activity, but other drivers often include:

- the availability of tools to assist in the work;
- communication overheads, including the time it takes to get requirements clarified and approved;
- the stability of the environment – that is, the extent to which the work has to cope with changes to requirements or resources;
- the size of the project team: there is a tendency for larger jobs involving lots of people to be less efficient than smaller ones because more time has to be spent on management, planning and co-ordination at the expense of ‘real work’.

The problems that can affect productivity are often considered at the same time as risks to the project in general (see Chapter 7).

6.5.2 Function points

There was a time when almost all IT projects involved writing software of some description. This is now increasingly less the case for many reasons, one of which is the tendency to use ‘off-the-shelf’ software applications. However, there are still many cases where software has to be written specially, and these situations can cause particular challenges for the estimation of development effort.

If we use a parametric approach, the first question is what to use as size drivers. If IT is old enough to have any real ‘traditions’, then one of the longest established of these would be to use **lines of code** as the size driver for software development. (When software is written, the programmer writes the instructions – as lines of code – in a form which is comprehensible to human experts. This ‘code’ is an electronic document which can be changed, added to and printed. When the code is to be executed by the computer, the document is ‘read’ by a special piece of software which converts it into a format that the computer can interpret automatically.)

From this very brief explanation it can be seen that:

- the code is a very technical product – it would need a software expert to estimate the number of lines of code;
- you will not know the exact number of lines of code until quite near the end of the project; most other size drivers are known at the beginning, or at least at an early stage, of the project.

Things are also complicated by there being many programming languages. Some are more ‘powerful’ than others – that is, they need fewer lines of code to carry out a particular procedure.

Rather than use this technical unit of size which is invisible to everyone except the software developers, it is more convenient to use as the size drivers counts externally apparent features of the software application. This would be rather like using the number of storeys, the floor space and so on to estimate the cost of a house, rather than the number of bricks. With software applications, this can be done with function points.



ADVANCED TOPIC Function points

For the purposes of the Foundation Certificate, you do not need to know the details of the rules of function point counting. There are at least two major systems of function point counting and some of the detailed rules are rather arcane, to say the least. The following description should be enough to give a general idea of the approach. (It is based on one particular version – Mark II, or Symons, function points – simply because this is, in our view, the simplest method for getting an understanding of the general principles of the approach.)

- (a) The size drivers are features of the software that are apparent to the user. In general, users are aware of the **transactions** that they can carry out when using a software application. A transaction is where the user **inputs** something into the computer (normally by typing), the computer carries out a procedure and comes up with a **response**, normally in the form of an **output**, and the computer system is left in a **consistent state**. This is similar to a use case in UML.

When booking a boating holiday using the new Canal Dreams ebooking system, you make a mistake (for example, typing in an invalid date) and an error message is displayed. Although an input has been followed by an output (the error message), the system is not in a consistent state: only half the booking details have been set up. In this case the processing so far would not be regarded as a transaction: either the whole booking would be rejected or the processing would continue until a complete and correct set of booking details had been input.

- (b) For each type of transaction, a count is made of the number of items of information that are input and output, and the number of tables of information that are accessed. In general, it can be assumed that the more of these there are, the more lines of code will have to be written, and the more work there will be for the system developers.
- (c) The counts are weighted to take account of the relative difficulty of implementing each type of feature. For example, a simple output is normally easier to implement than an input. With inputs you often have to carry out error checking, which adds to the developer's work. To take account of these differences in difficulty, the feature counts are weighted appropriately. In the Mark II method, inputs are weighted 0.58, outputs 0.26 and entity (or table) accesses 1.66. Effectively this means that the weighting between inputs, outputs and entity accesses is about 2:1:6. The use of such peculiar numbers is because the inventor of this method wanted the resulting function point counts to be about the same as for the American method (specified by the international function point user group, IFPUG) and hoped to achieve this by making the weightings add up to 2.50 (that is, $0.58 + 0.26 + 1.66$).

A restriction on the use of function point counting is that it assumes that there is a human operator initiating transactions and receiving outputs from the system. COSMIC function points are an alternative approach that can be used to measure the size of embedded software which interfaces with other software and hardware layers rather than human users.

6.5.3 An example of function point counting

Within the Canal Dreams ebooking system, there is a transaction which records the final payment made by a customer for a booking for which they have already paid a deposit. There are **three inputs** for a new payment:

- date;
- customer account reference;
- amount.

There are **four** possible **outputs** from the transaction:

- payment reference, a number allocated automatically by the computer system;
- customer name;
- customer address;
- an error message.

To carry out this transaction, a CUSTOMER ACCOUNT table and a PAYMENT table are accessed, giving **two entity accesses**. The function point count for this transaction is therefore:

$$(3 \times 0.58) + (4 \times 0.26) + (2 \times 1.66), \text{ that is, } 6.10.$$

What does this 6.10 really represent? It is best regarded as an index value that gives an idea of the amount of processing carried out by the transaction. For a single, isolated transaction, this measure is not very accurate. However, if you were able to add up the function point counts for all the transactions in an information system, then it is likely that the count for the application as a whole would be a useful indicator of its size.

We can use a function point count to find out the relative productivity of development projects that have already been completed. We may find that the average number of function points implemented per day is around five. This may seem a rather small number, but 'development effort' here includes the whole development cycle, from requirements gathering to testing. When a new project proposal comes along, a preliminary investigation may suggest that the delivered system would have a count of about 250 function points. The estimated effort is therefore in the region of $250/5$ days – that is, 50 days.

6.6 ESTIMATING BY ANALOGY

The function point approach (and, indeed, the more generic approach of using size drivers and productivity rates) is based on the assumption that we have the details of the size driver values and actual effort of past projects. Often, however, such records do not exist. For smaller organisations particularly, the IT projects that have been previously implemented may all seem to have their own peculiarities. For example, some may have involved the installation of off-the-shelf packages,

others may have required specially written software, some a mixture of the two, and so on. This seems to suggest that previous experience is not a stable basis for estimating the effort for new projects. However, in this kind of situation the **analogy** or **comparative** approach could be used.

The main steps with this method are as follows.

- (a) Identify the key characteristics of the new project.
- (b) Search for a previous project which has similar characteristics.
- (c) Use the actual effort recorded for the previous project as the base estimate for the new one.
- (d) Identify the key differences between the old and the new projects (it is unlikely that the old project is an exact match for the new one).
- (e) Adjust the base estimate to take account of the identified differences.

An analogy approach can be used to create a top-down estimate for a project. Where there is no past project which seems to be a useful analogy for the new project, an estimator can use analogy to select parts of old projects that seem similar to components of the current project (using analogy as part of a bottom-up approach).

As Table 6.1 shows, both analogy and the parametric approaches can be used either at the overall level of a project or for estimating the effort needed for components. The activity-based approach – breaking down the overall task into smaller components – seems almost by definition to be a bottom-up approach.

	activity-based/ analytical	parametric	analogy/ comparative
top-down		✓	✓
bottom-up	✓	✓	✓

6.7 CHECKLIST

As a project planner you may often need to use the effort estimates produced by experts from technical areas in which you are not knowledgeable. Are there any

ways in which you can realistically review these estimates? It may be possible to assess the plausibility of the estimates by asking the estimator the questions below.

- What methods were used to produce the estimates?
- How is the relative size of the job measured (in other words, what are the size/effort drivers)?
- How much effort was assumed would be required for each unit of the size driver (in other words, what productivity rates are you assuming)?
- Can a past project of about the same size be identified which had about the same effort?
- If a job with a comparable size cannot be identified, can past jobs which had similar productivity rates be found?

SAMPLE QUESTIONS

XYZ ORGANISATION SCENARIO

Staff have managed to develop information systems at a rate of five function points per staff day. A new system has been assessed as requiring 120 function points to implement, but the staff available are relatively inexperienced and are only 80% as productive as the staff usually used in such projects.

1. An under estimate of effort is MOST likely to lead to which of the following?

- (a) decreased productivity
- (b) decreased quality
- (c) a less competitive bid for a contract
- (d) a longer project duration

2. Which of the following estimating methods is MOST likely to be used bottom-up?

- (a) parametric
- (b) algorithmic
- (c) Delphi
- (d) activity-based

Both questions 3 and 4 use this scenario.

3. In the XYZ scenario, of which of the following is 80% of the value?

- (a) a size driver
- (b) an effort driver
- (c) a productivity rate
- (d) a productivity driver

4. In the XYZ scenario, what would be the best estimate of effort for the project?

- (a) 30 days
- (b) 25 days
- (c) 24 days
- (d) 20 days

ANSWERS TO SAMPLE QUESTIONS

1. (b) 2. (d) 3. (d) 4. (a)

POINTERS FOR ACTIVITIES**ACTIVITY 6.1**

The work breakdown structure (or possibly the product breakdown structure).

ACTIVITY 6.2

Among the activities that may need to be carried out are:

- Create/agree job descriptions
- Create job advertisements
- Collect and assess applications and curricula vitae (CVs) from potential employees
- Invite selected candidates
- Interview candidates
- Notify successful and unsuccessful candidates
- Request, await and check references
- Confirm appointment
- Arrange induction
- Carry out induction processes

This set of activities offers some good illustrations of the difference between elapsed time and effort. There will be some points – for example where you are waiting for references – where little effort is expended but time will be passing.

ACTIVITY 6.3

The following are suitable answers:

- (a) The number of functions that users need to be able to use.
- (b) The number of different types of system user (as each will need to be interviewed for their requirements), and the number of different operations carried out in the system.
- (c) The number of functions to be tested and the number of input and output data items to be tested.
- (d) The number of different functions in the system, the number of inputs, outputs and tables accessed.

ACTIVITY 6.4

- (a) The size driver would be the distance driven to work.
- (b) The productivity rate would be the average speed of the car.
- (c) We have already suggested that the weather and the amount of traffic congestion could have an effect on the travel time.

In this case, the weather and traffic do not increase the size of the job to be done – the distance to work remains the same. These factors are best seen as influences on the productivity rate. In order to assess more accurately the time it takes me to go to work, I could take account of these intermittent constraints on my speed. I may be aware, for instance, that the rush-hour traffic in the morning tends to be significantly less heavy during school holidays. I could therefore perhaps allow myself to start off to work a few minutes later when it is half-term. On the other hand, I may start earlier if the weather is foggy, as I know that this can slow down the traffic.

INDEX

- acceptance of risk 122
- acceptance testing 11, 16, 18, 85, 86, 92
- active communication methods 144
- activity on arrow networks 37
- activity on node networks 37–8
- activity-based estimating 103, 109
- activity-based planning 34–5
- activity boxes 42
- activity networks, project planning 37–9
- activity span 42–3
- adourning, team building phase 141
- agile project practices 17
- analogy approach, estimating 108–9
- analytical estimating 103
- assessment of risk 117–18
- autocratic style of management 143
- avoidance of risk 122–3

- backlog 17, 59
- baselines 71
 - Gantt charts 63, 64
 - project plans 71–2
 - variations on 72
- Belbin, team roles 142–3
- benchmarking 105
- Boehm's Top Ten Software Project Risks 116
- bottom-up estimating 103–4
- budget constraints 8
- business assurance
 - co-ordinator 135
- business case 1, 24–5, 32
 - identification of 8–9
 - net present value 25–6
 - payback period 26–7
 - reports 9
- business objectives 3

- Canal Dreams project scenario 4
- capability maturity models (CMM) 94–5
- cash flow 25, 26, 27
- change control and management 14, 70–1
 - configuration management 76–8
 - definitions of change 71–2
 - process of 74–6
 - roles and responsibilities 73
- change control board (CCB) 73
- change feasibility group, role of 73
- change implementation group 73
- change management 70–1
 - process 74–6
 - roles and responsibilities 73
- change manager, role of 73
- change procedures 5
- change register 73
- change requestor, role of 73, 74
- changeover methods 27–8
- checklists of risks 115–16
- checkpoint meetings 59
- checkpoint reports 59
- closure of project 12
- co-ordinator, team role 142
- code, lines of, size driver 106
- communication methods 144–5
- communication plans 24, 145
- comparative estimating 108–9
- completer finisher, team role 142
- configuration control 77–8
- configuration item identification 77
- configuration items (CIs) 77
- configuration management 76–8, 138
- configuration management
 - database (CMDB) 77
- configuration management system (CMS) 77
- configuration manager/librarian 77
- configuration status accounting 77
- conformance to standard, quality definition 81
- constraints 8–9, 37–8
- construction phase of project 11, 16
- consultative management style 143, 144
- contingency plans 123
- contingency pools 60–1
- control of projects 5, 13–15
 - applying control 58
 - corrective action, applying 60–2
 - project control life cycle 55–6
 - reporting meetings 59–60
 - standard control strategies 61–2
- control of risk 124–5
- corrective action 60–2
- COSMIC function points 107
- cost impact, risk assessment 120
- cost planning 22–3
- cost requirements 5
- cost specification, iron triangle 3
- critical path 41
- Crosby, Philip 80
- cumulative resource charts 64–5

- deadlines 3, 5, 17, 56, 61, 62
- defects
 - detection of 85–7
 - removal of 87–90
- deliverables 7, 33
 - monitoring of 56, 57
 - and risk exposure 117
 - time, cost and scope 57
- Delphi approach 102–3
- democratic style of management 143
- design phase of project 10–11, 16, 86–7
 - peer review of 88
 - systems design 85–6
- desk checking 87
- development life cycle models 15, 15–20
 - agile project practices 17
 - incremental model 17, 18
 - iterative model 17, 19–20
 - waterfall model 15–16
- development life cycle processes, splitting up 12–15
- development staff, sourcing 23–4
- documents 77
 - business case 24–5
 - Gantt charts 48–9, 63–4
 - product plans 33–4
 - project initiation document 20–2, 58
 - requirements 36
 - review of 88
 - risk log/register 124–5
- dry-run 88
- duration versus effort 100–1
- Dynamic Systems Development Method/Attern Method (DSDM Attern) 17
- dynamic testing 90–2

- earned value analysis (EVA) 65–6
- effort 39
 - vs. duration 100–1
- effort/size drivers 56, 104
 - lines of code as 106
- elapsed times, estimating 39–43
- entry requirements 86–7
- estimating 13–14, 100
 - by analogy 108–9
 - assessing plausibility of estimates 109–10
 - bottom-up approach 103–4
 - Delphi approach 102–3

- effects of over- and under-estimating 101
- effort versus duration 100–1
- elapsed time 39–43
- estimates and targets 101
- expert judgement 101–2
- parametric approach 104–8
- top-down approach 104
- 'events' in life of project 39
- exception plans 61, 62, 71
- exception reports 60, 61
- Executive 132
- exit requirements 87
- expert judgement, estimating method 101–2
- Extreme Programming (XP) 17
- failure of systems 81, 83, 98
- feasibility study 1, 8
 - for change requests 73, 74–5
- feedback loops, waterfall model 16
- financial expenditure, monitoring of 56–7
- fitness for purpose, quality definition 81, 82
- float 41
- formal communication methods 145
- formal monitoring methods 57–8
- forming, team building phase 141
- free float 48
- function points counting 106–8
- functional requirements 4, 82
- Gantt charts 48–9
 - progress information 63–4
- generic risks 116
- gold-plating 116
- highlight reports 58, 59–60
- identification
 - of business case 8–9
 - of configuration items 77
 - of required resources 44
 - of risks 115–17
 - of stakeholders 130
- impact of risk 117, 118
- mitigating actions 123
- probability impact grid 121–2
- and risk exposure 118–19
- implementation 11, 16, 18
 - of request for change 76
 - requirements 87
 - strategies 27–8
- implementer, team role 142
- incremental model 17, 18
- informal communication methods 145
- informal monitoring 58
- initiation process 8
- inputs 10
 - function point counting 107, 108
- inspection process 88–9
- installation 11, 57, 68
- integration testing 91–2
- intermediate products 33, 34
- internal physical design 10
- iron triangle 3
- ISO 9001:2008, quality management systems 93–4
- issue management 14
- issues log 59
- iterative models 17, 19–20, 43
- joint application development (JAD) sessions 9
- leadership styles 143–4
- legal requirements 4
- lessons learnt report 12
- logical design 10
- maintenance 12
 - of logs/risk registers 124–5, 138
- management styles 143–4
- matrix management 139–40
- matrix organisational structure 140
- maturity models 94–5
- meantime to repair (MTTR) 98
- meantime between failure (MTBF) 98
- meetings
 - arranging of, role of PSO 138
 - inspection 89
 - reporting 59–60
- milestones 7, 21, 39, 56, 118
- mitigating actions, risk 122, 123–4
- monitor evaluator, team role 142
- monitoring and control 14, 21–2, 55
 - applying control 58
 - graphical representation of progress 63–6
 - of progress 56–8
 - project control life cycle 55–6
 - reporting meetings 59–60
 - of risks 124–5
 - taking corrective action 60–2
- net present value 25–6
- norming, team building phase 141
- objectives of successful projects 2–5
- off-the-shelf software
 - evaluation of 10
 - pros and cons of 7–8
- one-shot/once-through approach 15–16
- opportunity, risk viewed as 117–18
- organisation 15
- organisational constraints 9
- organisational fit 24
- organisational framework 131–6
- outputs 10
 - function point counting 107, 108
- outsourcing 24, 26, 124
- overrun of projects 60
- pair programming 90
- parallel-running 27–8
- parametric approach to estimating 104–8
- passive communication methods 144
- payback period 26–7
- peer reviews 88
- performing, team building phase 141
- physical design 10
- pilot changeover 28
- planning 13–14
 - approaches to 33–5
 - of projects 20–4
 - of risk 124–5
- plant, team role 142
- portability 82
- post-implementation review (PIR) 12, 28–9
- PRINCE2, project management method 6, 113, 129, 131–2
- prioritisation of risks 121–2
- probability 118
- probability of risk 120–1
 - and risk exposure 118–19
- probability impact grid 121–2
- process maturity 94–5
- process requirements 86–7
- product-based planning 33–4
- product breakdown structure
 - diagrams 33, 34–5
- product definitions 34, 82
- product flow diagrams (PFDs) 35–7
- productivity drivers 105–6
- productivity rate 105
- programme and project support office (PPSO) 137
- programme board 140
- programme board meetings 60
- programme director 130
- programme management 140
- programme management office (PMO) 137
- programme manager 130
- programmes, projects grouped into 60, 130
- progress (information)
 - graphical representation of 63–6
 - monitoring of 56–8
- project assurance 14–15
- project assurance team 135
- project authority 132
- Project Board 132–3
- project board meetings 59–60
- project control life cycle 55–6
- project evaluation review 28–9
- project initiation document (PID) 20–2
- project management 5–6, 13–15
- project management board 9, 132
- project management methods 6
- project manager
 - desirable characteristics of 136–7
 - role of 5, 73, 133–4
- project organisation 129
 - communication methods 144–5
 - management styles 143–4
 - matrix management 139–40
 - organisational framework 131–6
 - programmes and projects 130
 - project manager skills 136–7
 - project support office 137–8
 - project team 138–9
 - stakeholders 130
 - team building 141–2
 - team dynamics 142–3
- project planning 32
 - activity planning 37–43
 - approaches to planning 33–5
 - product flow diagrams (PFDs) 35–7
 - resource allocation 44–9
 - software tools for 49
- project plans 20
 - communication planning 24
 - cost planning 22–3
 - project initiation document (PID) 20–2
 - quality planning 24
 - resource planning 23–4
 - scheduling 22
- project progress, graphical representation of 63–4
- project repository/library 77

- project risk plan 124–5
- project sponsor 2, 131, 132
- project support office (PSO) 137–8
- project teams 138–9
- projects 1
 - attributes of 2
 - objectives of successful 2–5
- prompts, project risks 115–16
- prototypes 9, 20
- proximity of risk 118
- qualitative approach to risk
 - assessment 119–22
- quality 80–1
 - capability maturity models 94–5
 - characteristics of 82
 - control vs. assurance 84–5
 - criteria of 82–3
 - definitions of 81
 - detecting defects 85–90
 - dynamic testing 90–2
 - ISO 9001:2008 93–4
 - plans 85
 - supplier evaluation 93
- quality assurance 84–5, 133
 - project assurance team 135
- quality control 84–5
 - and the V model 85–6, 90
- quality criteria 24, 34, 82–3
- quality management system (QMS) 84
- quality plans 24, 84, 85
- quality requirements 5
- quality strategy 84
- quantitative financial criteria
 - net present value 25–6
 - payback period 26–7
- quantitative risk assessment 118–19
 - mapping to qualitative 120–1
- regression testing 92
- relationship-orientated leadership style 143
- replanning of project 61–2
- reporting cycle 58
- reporting meetings 59–60
- reporting structure 58
- requests for change (RFCs) 73
 - approval of 75–6
 - assessment of feasibility 74–5
 - implementing 76
 - recording of 138
 - submission and review of 74
- requirements elicitation and analysis 9–10, 16
- requirements specification 85–6
- requirements statement 10, 11
- resource allocation 44–9
- resource clashes 47
- resource constraints 38
- resource contingency 60–1
- resource histograms 46
- resource investigator, team role 142
- resource planning 23–4
- resource types 44
- responsibility assignment matrix (RAM) 23
- reviews 12
- risk 113–14
 - assessing 117–18
 - deciding appropriate actions 122–4
 - identifying 115–17
 - management of 114–15
 - planning, monitoring and control 124–5
 - qualitative approach to 119–22
 - quantitative approaches to 118–19
 - risk avoidance 122–3
 - risk evaluation criteria 117
 - risk exposure 117–19, 124
 - risk logs 124–5
 - risk management 14
 - risk management framework 114–15
 - risk owner 125
 - risk probability 120–1
 - risk proximity 118
 - risk records 125
 - risk reduction leverage (RRL) 119, 122
 - risk registers 124–5
 - risk tolerance line 122, 124
 - roles and responsibilities
 - change management 73
 - project organisation framework 131–5
 - S-curve charts 64–5
 - schedule planning 22
 - scope of project 3, 5
 - changes to 72
 - reducing 62
 - scope impact, risk assessment 121
 - Scrum, agile practices 17
 - self-fulfilling prophecy 101
 - Senior Supplier 132
 - Senior User 132
 - set-up of project 9
 - shaper, team role 142
 - shortfalls, risk checklist 116
 - show-stoppers 124
 - size/effort drivers 56
 - function point counting 107–8
 - lines of code as 106
 - and productivity 104–6
 - SMART criteria 3
 - software
 - design 10
 - function point counting 106–8
 - lines of code as size driver 106
 - off-the-shelf 7–8, 10
 - for project planning 49
 - specific risks 116
 - specifications for project 3
 - sprints 17
 - staff increases 61
 - staff, sourcing of 23–4
 - stage manager, role of 134
 - stakeholders 3, 15, 24, 130, 136, 145
 - standards 81, 82, 84, 85, 93–4, 132
 - static testing 90
 - steering committee 9, 132
 - steering committee meetings 60
 - storming, team building phase 141
 - success criteria (objectives) 3–4
 - summary risk profile 121–2
 - suppliers, evaluation of 93
 - system development life cycle (SDLC) 6–7
 - acceptance testing 11
 - construction process 11
 - design 10–11
 - identification of business case 8–9
 - implementation/installation 11
 - initiation process 8
 - models 15–16
 - project closure 12
 - project set-up 9
 - requirements elicitation and analysis 9–10
 - review and maintenance 12
 - stages 12–13
 - systems design 85–6
 - systems testing 92
 - targets and estimates 101
 - task-orientated leadership style 143
 - team building 141–2
 - team dynamics 142–3
 - team leaders 134–5
 - approval of changes 75
 - reporting to project manager 139
 - team meetings 59
 - team members, role of 135
 - team worker, team role 142
 - teams, project 138–9
 - technical assurance
 - co-ordinator 135
 - technical constraints 8, 38
 - technical specialist, team role 142
 - temporary staff, sourcing of 23–4
 - termination of project 62
 - terms of reference 5, 8, 9, 133
 - testing 19, 44, 78, 83, 86
 - integration 91–2
 - regression 92
 - static 90
 - systems 86, 92
 - unit 86, 90–1
 - user acceptance 11, 16, 18, 85, 86, 92
 - TickIT Guide 94
 - time constraints 8
 - time estimations 39–43
 - time extensions 62
 - time impact, risk assessment 121
 - time recording, project control 137
 - time specification, iron triangle 3
 - timeboxes, incremental model 17
 - timesheets 58
 - tolerance line, risk 122, 124
 - tolerances 60–1, 71
 - top-down estimating 104
 - Top Ten Software Project Risks, Boehm 116
 - total float 49
 - transactions and function point counting 107–8
 - transfer of risk 123
 - Tuckman-Jensen model, team building 141
 - uncertainty 5, 12, 117, 118, 122
 - unit testing 90–1
 - use of resources, monitoring of 56–7
 - user acceptance testing 11, 16, 18, 85, 86, 92
 - user assurance co-ordinator 135
 - V model 85–6
 - variations on baselines 72
 - version numbers
 - configuration control 77
 - iterative model 19, 43
 - walkthroughs 89–90
 - waterfall model 15–16
 - work-based planning 35–7
 - work breakdown structures (WBS) 34

PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

Second edition

Bob Hughes (editor)

Ideal for IT practitioners with project management responsibilities, this book explains the principles of IT-related project management. This fully updated second edition covers key areas such as project planning, monitoring and control, change management, risk management and communication between project stakeholders.

The only textbook tailored specifically to match the syllabus of the BCS Foundation Certificate in IS Project Management, this book will not only help you pass an exam, it will also be a great tool in the successful management of IT projects.

- Contains new material on risk assessment and agile approaches to project management
- Includes sample questions and answers to reinforce the learning process
- Official textbook for the BCS Foundation Certificate in IS Project Management

ABOUT THE AUTHOR

Bob Hughes worked on IT projects in the telecommunications, energy and local government sectors before becoming an academic at the University of Brighton. He is the author of a number of books, as well as chief moderator of both the BCS Professional and Higher Education Qualifications in IT Project Management.

You might also be interested in:

NLP FOR PROJECT MANAGERS

Make things happen with neuro-linguistic programming
Peter Parkes

PROJECT MANAGEMENT IN THE REAL WORLD

Elizabeth Harrin

The structured approach, with clearly laid out learning objectives, will appeal to trainers and academics as well as practitioners, and the content is lucidly written and supported by some excellent diagrams.

This practical book hits several important targets and is a valuable update of a popular standard text.

**Miles Shepherd, Vice President,
Association for Project Management**

Business; Management;
Information technology



ISBN 9781780171180

