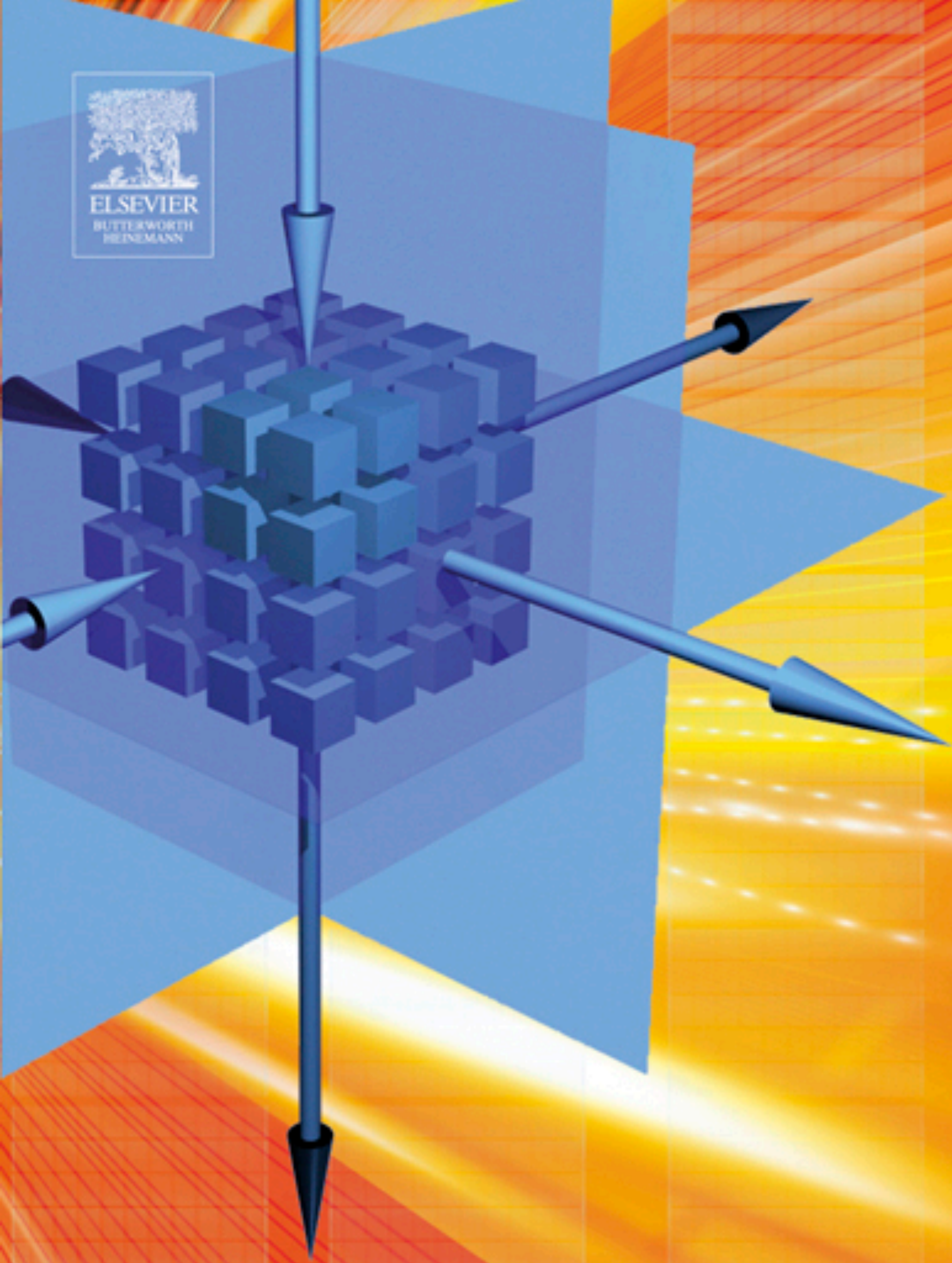
ELSEVIER
BUTTERWORTH
HEINEMANN



TOM GILB

COMPETITIVE ENGINEERING

A HANDBOOK FOR SYSTEMS ENGINEERING, REQUIREMENTS ENGINEERING, AND SOFTWARE ENGINEERING USING PLANGUAGE

Competitive Engineering

A Handbook for Systems Engineering Requirements Engineering, and Software Engineering Using Planguage

Competitive Engineering

A Handbook for Systems Engineering Requirements Engineering, and Software Engineering Using Planguage

Tom Gilb
Email: Tom@Gilb.com
URL: www.Gilb.com

Editor: Lindsey Brodie,
Middlesex University, UK
Email: lindseybrodie@btopenworld.com

NON SOLUS
ELSEVIER
BUTTERWORTH
HEINEMANN

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier Butterworth-Heinemann
Linacre House, Jordan Hill, Oxford OX2 8DP
30 Corporate Drive, Burlington, MA 01803

First published 2005

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication Data
A catalogue record for this book is available from the Library of Congress

ISBN 0 7506 6507 6

For information on all Elsevier Butterworth-Heinemann publications visit our website at http://www.books.elsevier.com

Typeset by Integra Software Services Pvt. Ltd, Pondicherry, India
www.integra-india.com
Printed and bound in Great Britain

CONTENTS

Foreword

Erik Simmons vii

Endorsements

Roger Pressman ix

Mark Maier x

Preface xii

Acknowledgements xv

Introduction xvii

Chapter 1 Planguage Basics and Process Control 1

Chapter 2 Introduction to Requirements 35

Chapter 3 Functions 81

Chapter 4 Performance 109

Chapter 5 Scales of Measure 137

Chapter 6 Resources, Budgets and Costs 165

Chapter 7 Design Ideas and Design Engineering 185

Chapter 8 Specification Quality Control 221

Chapter 9 Impact Estimation 261

Chapter 10 Evolutionary Project Management 291

Planguage Concept Glossary 321

Glossary Introduction 321

Bibliography 439

Further Reading 445

Index 449

To Jane

FOREWORD

Competition is the strongest force shaping today's product development landscape. The race to create customer and end user value is intense. In addition, companies large and small face ever-increasing product complexity, pressure to reduce time to market and increase productivity, and unprecedented challenges from globalization.

Competitive Engineering contains powerful tools to apply to these problems. At the same time, the tools are both practical and simple – a rare combination. Over the last decade, I personally have applied these tools in a variety of settings in software development and more general product development, on projects of various sizes. Thousands of students have been through training and workshops I have authored that contain Planguage, Evo, Specification Quality Control and other facets of Competitive Engineering. The vast majority of students immediately recognize their value and go on to use these beneficially on projects. *Competitive Engineering* is based on decades of practical experience, feedback and improvement, and it shows. This stuff works.

To be effective over a wide range of problems, a method or tool must possess many qualities: flexibility, scalability, portability and learnability, to name a few. The methods and tools found in *Competitive Engineering* are up to the challenge. They are designed to be tailored to local culture and practices. The central ideas are so fundamental that they apply to a huge range of project types and sizes. And, while they are rich enough that they require serious study to master, they can be learned and used effectively by almost anyone; I have taught them to people in product development, service delivery, manufacturing, site construction, information technology, eBusiness, quality, marketing and management.

You may encounter some resistance when first proposing or teaching these concepts; I have, and still do from time to time. Don't be discouraged. These are often revolutionary concepts in relation to a group's existing practices. Keep in mind that it is not necessary to use everything you find in *Competitive Engineering* right from the start. Instead, use Competitive Engineering to create Competitive Engineering. Start using Evo to improve things in small steps based on what is most valuable in your environment. Use Planguage to document stakeholder needs, success criteria and the like. Before you know it, you will have made significant progress. It all feels remarkably natural once you get going.

Erik Simmons
Intel Corporation
Requirements Engineering Practice Lead
Corporate Quality Network

Endorsements

When I was a young engineer, I thought all system problems were merely a matter of applying clever technological solutions. "If we had better technology," I thought, "we'd be able to get this problem solved quickly."

After more than 30 years in the IT and systems world, I now know better. Creating, building, and delivering high quality computer-based systems, on-time and within budget not only requires solid technology, but also a meaningful process, effective project management, comprehensive risk control, and broad-based communication between all constituencies at all levels of the project. In fact, it is the last item in this list that is probably most important (and the most difficult to achieve). Projects fail because communication fails.

System and software engineers and their customers need a consistent mechanism for communicating the purpose of the system, the constraints that must be addressed, the design and implementation strategies that are to be applied, the risks to be managed, and the measures of quality that are relevant and meaningful. But how do we achieve a consistent mechanism for communication and how do we use this to better manage and implement complex systems?

In this book, Tom Gilb provides us with answers that are both elegant and comprehensive. He describes the Planguage method – "a practical set of ideas, to help you get better control over all forms of planning, design, engineering, and project management."

To be honest, those of us who are industry veterans have heard this before. Literally dozens of methods (and books) have made similar claims and then failed to deliver. Why then, should we believe that Planguage is any different? There are many reasons.

Gilb has designed an evolutionary approach that will allow you to define system requirements clearly and unambiguously. More importantly, he provides a bridge that enables you to describe the resultant solution, design a high quality implementation, and then analyze how the proposed solution impacts business objectives. In addition, he stresses quantitative evaluation, so that progress toward the competitive goals of your system can be evaluated competitively. Planguage enhances communication – at the specification level, at the design level, at the project level, and at the process level. Projects succeed when communication succeeds, and Planguage leads to successful communication.

As Tom himself admits, this book is not a light read. Competitive Engineering provides thought-provoking ideas on almost every page, is rich in detail, and comprehensive in scope. It provides guidance for every system engineering activity and a thorough description of every aspect of Planguage. You'll have to spend time with Gilb's ideas, but once you understand and apply them, your ability to engineer complex systems will be greatly enhanced.

Roger S. Pressman, Ph.D.
President, R.S. Pressman & Associates, Inc.
e-mail: pressman@rspa.com

Competitive Engineering stakes out unusual ground in engineering literature. The ground it takes is doubly unusual for a systems engineering book. *Competitive Engineering* works to provide pragmatic and directly applicable methods to fundamental design problems that cross application domains. Where many books provide procedural methods that apply only to particular problems, *Competitive Engineering* seeks to be general and to be applicable to many forms of complex systems. Where many books take on abstract or theoretical aspects of system development, *Competitive Engineering* provides specific, directly applicable techniques. And where many books address only narrow business models, such as contracted development, *Competitive Engineering* applies to a range of evolutionary and competitive developments.

Tom Gilb clearly intends for the book to be taken as a whole. Planguage provides a documented and uniform set of concepts and terms that a team can use to organize development efforts from simple to complex. As Tom explains, the book can be viewed as a comprehensive handbook to managing the development of complex systems. Its background is particularly strong for information systems, but it also applies to many other types of systems.

However, systems engineers should rightly be interested in this book even if they have no intention of adopting Planguage in its entirety. *Competitive Engineering* contains many other important nuggets, in consequence of the unique ground it has staked out, that are important to systems engineering even removed from full adoption of Planguage. A few examples are the sections on Scales of Measure, Impact Estimation, and Evolutionary Project Management.

One of the intellectual foundations of systems engineering is decision theory. Decision theory rests on our ability to discover a proper set of attributes on which to measure the goodness or worth of a system, to quantify relative to those attributes, to evaluate the consequences of design choices relative to those attributes and to make decisions based on those attributes. *Competitive Engineering* adds a rich set of heuristics and methods to the notoriously difficult problem of discovering good sets of attributes, quantifying performance on complex attributes and communicating the consequences of design choices relative to those attributes. The approach in *Competitive Engineering* is strongly pragmatic, while also being theoretically grounded. A user who has committed to Planguage can take the approach as a whole and find strong guidance for fully implementing it. A user who has not adopted Planguage as a whole can also benefit, because he or she will discover that the Scales of Measure and Impact Estimation concepts are solidly grounded and can be lifted and transferred into other development ontologies and other decision theory methods. The 'lift-and-carry' into other methods will benefit from the strong heuristics and well-thought-out communication methods presented in the book.

In a similar way, *Competitive Engineering*'s take on Evolutionary Project Management is highly useful on its own. Virtually all software-intensive systems today are developed in an evolutionary way, even when we don't plan to. While the importance and basic mechanisms of evolutionary development are well known, the community is in need of wider sets of guidelines and alternatives for actual implementation. It is in this area that *Competitive Engineering* delivers important new material.

Systems engineers should find *Competitive Engineering* widely useful, with or without the additional framework provided by Planguage. Even without adopting Planguage as a whole there are numerous important principles and techniques that can benefit any system project. And those who dip in looking for solutions to one problem or another may come to appreciate the full framework of Planguage.

Dr Mark W. Maier
Distinguished Engineer at The Aerospace Corporation and Chair of the INCOSE Systems Architecture Working Group. Co-author of *The Art of Systems Architecting*, Second Edition (CRC Press).

PREFACE

Background to writing *Competitive Engineering*

It has been sixteen years since *Principles of Software Engineering Management* (Tom Gilb, 1988, Addison-Wesley) was published. Since then I have continued to develop Planguage and many changes have been introduced. So, my main intention in writing *Competitive Engineering* (CE) is to document the current basic definition of Planguage (that is, the language and its methods).

In practice, the discipline of writing this book has also caused considerable improvement in the consistency of ideas and the quality of the explanation. Hopefully, readers will forgive me that the style of this book is deliberately 'less chatty' than *Principles of Software Engineering Management*. The aim is to provide a fundamental systems engineering handbook, which is more directly concerned with providing practical guidance on how to use Planguage.

In large part, CE is based around the Glossary of Planguage. The Glossary gives additional rigor to the book, as it has been applied to the entire text, including the Glossary itself.

Major influences on Planguage

The dominating influences behind the creation of Planguage include:

- The works of Deming, Juran, Crosby, Jevons (*The Principles of Science*, Dover Edition, 1960, originally published 1875), Boehm, Weinberg, Lord Kelvin, Keeney, Koen and Peters. See also the Bibliography and the citations in this book.
- My work with *real* engineers and managers in the industrial world; who want and need these ideas. It is amazing to me to see how each new piece of work, or consulting, directly results in evolutionary improvements to the theory and practice of Planguage.

- My many professional friends, clients and students, who appreciate, encourage, comment and discuss Planguage and share with me their ideas, case studies, papers and books.

> See http://zapatopi.net/Kelvin/quotes.html, which reads: ''In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be.''
>
> *Lord Kelvin*

How to use this book

I do not expect readers to adopt everything in this book. Adoption must evolve based on feedback from real use.

Only *generalized* processes and rules are given in this book, and so it is quite likely that they will need to be *tailored* to *your* organization. In any case, some of the rules and processes in this book are rather too long for everyday use. This is because explanatory text has been included. If you intend to use any of these rule sets, then some editing is required to produce a shorter 'working' version for Specification Quality Control (SQC) purposes. (You can always make reference from your shortened version to other text for more background information.)

You are at liberty to adopt and adapt any of the CE ideas to your needs.

Some book conventions

Terminology: I decided to use the following as the main terms[1] in Planguage:

- 'Requirement specification' and 'requirement engineering' rather than 'requirements specification' and 'requirements engineering', respectively.

[1] The alternative terms are declared as synonyms.

- 'Function' as an adjective, rather than 'functional' (so Planguage uses 'function requirement' and 'function specification').
- I retained 'systems engineering' as it is such a widely used term (even though I do feel it needs to be 'systems' engineering' with an apostrophe!).
- 'Resource' has been used as a main collective attribute term, rather than 'cost'. This was a difficult decision because 'resource' in the USA tends to only mean 'staff'. Planguage usage is wider: 'resource' includes all committed money, staff, time and any other assets.

Formatting of dates: To avoid the text 'ageing', most of the dates have been declared as user-defined terms, such as 'Next Year.' In practice, users should use more precise dates, such as 'December 15, 2004', or 'Initial Delivery plus 3 Months'.

In many examples within the book, the use of the 'Version' parameter, and other administrative parameters, have simply been omitted to reduce complexity of the examples.

Planguage Glossary: Approximately 180 concepts have been formally defined, exemplified and annotated in the CE Glossary. The other 75 per cent of defined Planguage concepts are and will be in the complete glossary on the web. They are selected from over 640 defined Planguage concepts. The complete and updated Planguage concept glossary will be found at www.gilb.com. At the beginning of each chapter are listed the key glossary concepts.

The reader is well advised to consult the Glossary when trying to make sense of the text. In fact, it might be a good idea for the reader to scan through the entire Glossary, stopping at interesting concepts and getting a sense of the types of concepts I have defined.

I often think that the main lasting contribution of this book lies in the Glossary itself. It is not that I imagine the entire world standardizing on these terms! Rather, that I hope this Glossary might be quite useful in helping to develop improved standard systems engineering concepts. I do not doubt that we can all make these concept definitions even better. However, the reader can be assured that all the concept definitions were arrived at after considerable struggle and due consideration of many points of view and needs! Not least was the struggle to make them internally consistent.

At the very least, I hope that the Glossary helps the reader to be sure of what the text is saying. I also hope that a study of the Glossary will give the student an excellent grounding in systems engineering concepts.

Tom Gilb, Kolbotn, Norway
Email: Tom@Gilb.com

Acknowledgements

I would especially like to thank my clients too numerous to mention, both the companies and the enthusiastic individuals, and classroom students. They created the need, and validated the practice, for so many of the Planguage ideas.

Specific credit must also go to the following:

My personal editor, Lindsey Brodie: she helped edit *Principles of Software Engineering Management* (Tom Gilb, 1988, Addison-Wesley) and has worked with me as a consultant. She has edited all my papers for many years. She has done a heroic and wonderfully conscientious task of editing this book, undoubtedly changing unintelligible, well-intended prose into consistent, intelligible text and diagrams. The majority of the diagrams are hers alone in concept and execution. If the book is readable, it is Lindsey who deserves the credit. If it is not, then I must have insisted on something she disagreed with!

The following have been our core team of advisors for this book. We could always trust them to have interesting, useful opinions and to do some useful research for us. They saved us from many a fatal error! They all practice, and teach, what we preach here too!

Don Mills (New Zealand), especially for his discussion on implicit direction of numeric values for levels on scales of measure.

Erik Simmons (Portland, Oregon), Intel Corporation for permission to use two extracts from his teaching material on 'Landing Zone', and who always has a sensible, practical answer to everything.

Stuart Woodward (Ipswich, England) who decided to teach himself Planguage, method by method, and provided the most interesting feedback and insights on what happened when he successfully applied the methods.

My son, Kai Thomas Gilb has been a consistent influence in keeping the Planguage ideas simple and straightforward. He has an absolutely independent mind: we constantly teach and consult together, and constantly argue the technical points. He does not give up until I see the light. He has written his own book manuscript (in a more popular style than CE) and developed two automated tools for Planguage.

Thanks too must go to numerous others for pointing out various issues and problems in Planguage and in the text. Too many to recall all of them.

Of course, any remaining errors in this book are entirely the author's responsibility!

Planguage continues to evolve, and it never fails to amaze me how new Planguage ideas continue to emerge. However, Planguage has reached a stage where publication is timely. I hope that readers will enjoy the book, and find something within it that they can apply – tomorrow, if not today!

Further acknowledgements

- Clarkson N. Potter for the Alice and The Cheshire Cat and the Humpty Dumpty figures.
- Gerrit Muller, Embedded Systems Institute, Eindhoven for permission to use his Stakeholder diagram.

Introduction

Competitive Engineering (CE) is about technological management, risk control, and breakthrough improvement in complex business systems, projects and processes. It is systems engineering, with application to all forms of planning, requirement specification, design and project management. It also applies to management of organizations, both top management and technical management. 'Competitive Engineering' is hopefully the end result of using this book's ideas.

What is in *Competitive Engineering*?

CE is taught using 'Planguage.'[1] Planguage consists of a new industrial systems engineering language for communicating systems engineering and management specifications, and a set of methods providing advice on best practices. 'Planguage' is central to CE and permeates all themes of this book.

The **Planguage Specification Language** is used to describe all the requirements, designs and plans for a system.

The main **Planguage Methods** are as follows:

- **Requirement Specification**: used to capture all the different requirement types. Emphasis is placed on specifying competitive performance and resource attributes quantitatively.
- **Impact Estimation**: used to evaluate designs against the requirements. It is also used during project implementation to track progress towards meeting the requirements.
- **Specification Quality Control**: used at any stage of a project to check the adherence of any plan, contract, bid or technical specification to best practice specification standards.
- **Evolutionary Project Management**: used to plan and monitor implementation of the selected designs.

The reader will, hopefully, find that these are all very practical and innovative methods, compared with current practice and literature.

[1] The word 'Planguage' is derived from a combination of 'plan' and 'language.' It is pronounced like 'language' with the initial 'p' pronounced as in 'plan.'

How was *Competitive Engineering* developed?

The Planguage language and methods have gradually emerged from my practical experience, since 1960, as a teacher and consultant to industry. Since the early 1980s, earlier versions of the central Planguage elements have been adopted both by pilot projects and/or corporate-wide in several of my client multinationals including HP, IBM, Intel, Philips, Nokia, Ericsson and Douglas Aircraft (now Boeing). The interest and practical acceptance has encouraged me to expand and refine the definition of the language well beyond my earlier books and papers.

In the past, my public courses, books and articles have explained Planguage, under a variety of names and in terms of its various subsets, exclusively for software engineering purposes. However, since the early 1980s, Planguage has been used in top management and various systems engineering disciplines. This book has been written to reflect that fact. The aim is that this book is useful for any 'systems engineering' or 'systems engineering management' purpose.

What is special about Planguage?

Planguage integrates all the basic systems engineering disciplines from requirement specification to product delivery. It has at its core the aim of delivering the stakeholder-critical values. It makes the technical strategies clearly subordinate to the required results. It gives us a detailed set of ways to express our ideas for a system, including very many useful glossary-defined concepts, such as: objective, strategy, design and risk.

Planguage provides a common language for all the different disciplines to communicate with each other. It enables interdisciplinary groups to work as real teams towards well-documented common purposes.

Planguage is designed for adaptation and tailoring to your own specific projects, organizations and cultures.

Planguage lays the groundwork for systematic learning organizations and continuous work process improvement. It is based on the fundamental principles of feedback, of 'Plan-Do-Study-Act' cycles of activity as taught by W. Edwards Deming (Deming 1993) and Joseph Juran (Juran 1974). These are probably the most powerful weapons we have for improving productivity and economics. It is also oriented towards the ideas of Philip B. Crosby on defect prevention and 'clear measurable requirements' (Crosby 1996).

Planguage gives us tools to tackle large and complex systems in a more systematic way than current common practice. Used properly, it should reduce the high risk of waste, delay and failure, which has plagued all large-scale modern projects and 'high-tech' disciplines for decades. We have plenty of good, intelligent and well-intentioned people; now they, hopefully, have a useful tool for better understanding and management of the fast-moving, complex and 'not-yet-experienced' world, which seems to be our current daily working environment.

CE's practical ideas are already proven in practice internationally in electronics engineering, telecommunications engineering, aircraft engineering, top management, marketing, information technology and industrial software engineering projects. Once they are imbedded in corporate practice, they stay there. Once the individual knows and uses these ideas, they are irreversibly sold on the practicality and usefulness of these methods.

> Planguage should be viewed as a powerful way to develop and implement strategies that will help your projects to deliver the required competitive results.

How to use *Competitive Engineering*

This is not a book to be read quickly and forgotten. Some CE chapters could expand to book length to fully explain their concepts. Indeed, this is the first in a series of related books, some of which are already written in draft form.[2]

This book is more like a dictionary or a handbook. It is intended to be a unifying standard. It should serve, for years, as the basis of your professional development (as it has for me and for my clients). Study it as needed. Try out the ideas in practice. Study more detailed literature. Translate it into your organization's local dialect. Use it to make rapid progress towards putting in place additional teaching and improved standards for your engineering and management methods.

[2] See www.Gilb.com for initial draft samples.

Structure of *Competitive Engineering*

The key contents of *Competitive Engineering* are as follows:

- **Chapter 1: Planguage Basics and Process Control**: This chapter explains why Planguage is necessary and introduces the structure of the Planguage language and methods.
- **Chapter 2: Introduction to Requirement**: This chapter outlines the fundamentals of requirement specification: the framework for setting targets and constraints.
- **Chapter 3: Functions**: This chapter describes functions and function requirements.
- **Chapter 4: Performance**: This chapter describes the basics of how to specify performance attributes quantitatively.
- **Chapter 5: Scales of Measure**: This chapter discusses finding and defining appropriate scales of measure.
- **Chapter 6: Resources, Budgets and Costs**: This chapter outlines how to specify the resource requirements.
- **Chapter 7: Design Ideas and Design Engineering**: This chapter describes how to find and specify design ideas. It also outlines the Design Engineering process.
- **Chapter 8: Specification Quality Control**: This chapter gives an overview of the Specification Quality Control method (also known as 'Inspection'), which measures specification quality against your own tailored specification standards.
- **Chapter 9: Impact Estimation**: This chapter describes the Impact Estimation method, which is used to evaluate quantitatively the impact of design ideas on your performance and resource requirements, and can also be used to track quantitatively project progress towards critical objectives.
- **Chapter 10: Evolutionary Project Management**: This chapter gives an overview of Evolutionary Project Management. It gives you the basic concepts of evolutionary delivery and discusses how you identify evolutionary steps.
- **Glossary**: The glossary in the book has over 180 concepts. It provides detailed reference definitions, supporting the text in the chapters (only the main concepts could be fitted into this book, the additional concepts can be found in the complete glossary at http://www.Gilb.com/).

Format of *Competitive Engineering*

The format of each chapter is the same:

- **Section 1: Introduction**: Each chapter has an introduction that puts the chapter in context.

- **Section 2: Practical Example**: Each chapter has a simple example that aims to introduce the subject area of the chapter.
- **Section 3: Language Core**: Each chapter has the basic new Planguage concepts described.
- **Section 4: Rules**: Each chapter (apart from Chapter 8, Specification Quality Control) includes some rules, which can be used as a specification standard for the chapter's subject.
- **Section 5: Process Description**: Each chapter has a process definition that corresponds to the subject of the chapter.
- **Section 6: Principle**: Each chapter has 10 principles, which are intended (in a light-hearted manner) to highlight and remind you of the key ideas discussed within the chapter.
- **Section 7: Additional Ideas**: Each chapter discusses some advanced ideas to try to give you some insights beyond the basic ideas described in the chapter.
- **Section 8: Further Example/Case Study**: Each chapter contains a more detailed example. Real case studies or practical examples from the author's first-hand experience are used.
- **Section 9: Diagrams/Icons**: Each chapter has diagrams providing graphical ideas for presenting the chapter content. Specifically, the icon language supporting Planguage is shown.
- **Section 10: Summary**: Each chapter is summarized in the last section.

A friendly warning

This book is intentionally written in a very condensed style. Don't get discouraged if you have to slow down to understand it, or if you have to reread parts. It is 'useful ideas per hour' which count, not 'pages turned per hour'.

Chapter
1

Planguage Basics and Process Control

The Purpose of Planguage

Glossary Concepts

Planguage
Standards
Rule
Process
Procedure
Task