Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

For further volumes:
http://www.springer.com/series/7592

Gerard O'Regan

Introduction to Software Process Improvement

Springer

Dr. Gerard O'Regan
11 White Oaks
Mallow, Co. Cork
Ireland
oregang@yahoo.com

Series editor
Ian Mackie

Advisory board
Samson Abramsky, University of Oxford, UK
Chris Hankin, Imperial College London, UK
Dexter Kozen, Cornell University, USA
Andrew Pitts, University of Cambridge, UK
Hanne Riis Nielson, Technical University of Denmark, Denmark
Steven Skiena, Stony Brook University, USA
Iain Stewart, University of Durham, UK

Undergraduate Topics in Computer Science ISSN 1863-7310
ISBN 978-0-85729-171-4 e-ISBN 978-0-85729-172-1
DOI 10.1007/978-0-85729-172-1
Springer London Dordrecht Heidelberg New York

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

© Springer-Verlag London Limited 2011
Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.
The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.
The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my two wonderful godchildren
Cian and Niamh

Preface

Overview

The objective of this book is to give an introduction to the software process improvement field to computer science students. The well-known Capability Maturity Model Integration (CMMI[1]) is used as the vehicle for software process improvement as it has become the de facto standard[2] for implementing best practice in software engineering. The reader is guided through the steps in setting up a CMMI improvement initiative; implementing the various CMMI process areas and maturity levels; and conducting appraisals. Appraisals are an essential part of the continuous improvement cycle as they allow the organization to understand its current software engineering process maturity, to prioritize future process improvements, and to confirm that the process improvements made have been effective.

The Software Engineering Institute (SEI) and many other quality experts believe that there is a close relationship between the quality and maturity of the underlying software processes and the quality of the delivered software product. The implementation of the CMMI brings best practice in software and systems engineering into the organization, thereby enhancing its software engineering effectiveness and enabling it to achieve more consistent results.

The implementation of the CMMI enables the organization to design and develop software following sound software engineering principles, i.e. the requirements are clearly defined and documented; a software design is produced that is valid with respect to the requirements. The software code implements the design and peer reviews and testing are employed to verify that the implementation is valid with respect to the requirements and the design. Sound project management practices are employed during the project, and this enables a high-quality software solution to be delivered on time and on budget to the customer.

[1] Specifically, the CMMI Development Model, CMMI-Dev, V1.2 is used as the vehicle for software process improvement in this book.

[2] Other approaches that are popular include the ISO/IEC 15504 standard (SPICE) and ISO 9001.

The steps involved in setting up a CMMI improvement initiative in an organization are described. This includes a discussion of the typical teams that will need to be involved as well as their roles and responsibilities. Guidance is provided to assist in the implementation of processes to satisfy the CMMI requirements, and this includes examples of typical deliverables produced to implement specific and generic practices in the various process areas. Tools to support the organization in enhancing its software engineering effectiveness are also described.

The software engineering maturity of an organization is determined objectively by an appraisal, and the organization is typically assigned a maturity level to reflect the extent of its implementation of the CMMI. Large organizations will often be interested in achieving a particular CMMI maturity level (e.g. CMMI level 2 or level 3) to allow benchmarking against competitor organizations, and this is determined by a formal SCAMPI Class A appraisal. Small organizations will often have a limited budget for process improvement initiatives, and formal appraisals may not be a priority. Instead, the focus may be on improvements to provide a tangible return on investment and to make a difference to software quality, on-time delivery, and so on.

Software process improvement should provide a tangible return on investment. These initiatives typically take place in order to achieve certain business goals such as improving software quality, reducing the time to market, improving productivity, reducing the cost of poor quality. Companies will measure the return on investment achieved to judge its effectiveness.

Organization and Features

The first chapter provides motivation for software process improvement. It discusses the benefits that may be gained and models that are available to support software process improvement.

Chapter 2 provides a broad overview of software engineering and includes a discussion on software life cycles and the phases in software development. It includes discussion on requirements elicitation, software design, implementation, testing, and maintenance.

Chapter 3 gives an overview of the CMMI model and discusses the five maturity levels and their constituent process areas.

Chapter 4 describes the activities and teams required to set up a CMMI improvement initiative for an organization. These include the CMMI Steering Group, the SEPG team, and process-specific teams. A continuous improvement cycle is described, as software process improvement is continuous.

Chapter 5 focuses on the implementation of CMMI level 2 in a typical organization. It includes an overview of each of the level 2 process areas as well as typical deliverables produced in a level 2 implementation.[3]

[3] These are typical deliverables produced and the CMMI needs to be interpreted appropriately to meet the needs of the organization.

Chapter 6 focuses on the implementation of CMMI level 3 in an organization. It includes an overview of each of the level 3 process areas as well as typical deliverables produced.

Chapter 7 gives an overview on the implementation of CMMI level 4 and 5 in an organization. Many organizations are more focused on CMMI level 2 and 3 implementation rather than on the higher maturity levels.[4]

Chapter 8 discusses various tools to support the organizations in improving their software engineering maturity. The focus is first to define the process and then to find tools to support the process.

Chapter 9 discusses the SCAMPI appraisal methodology. This includes the formal SCAMPI Class A appraisal often employed by large organizations to obtain a CMMI rating that allows them to benchmark itself against other organizations, and SCAMPI Class B and C appraisals that are less expensive and time consuming but may not be used for benchmarking.

Audience

This main audience of this book are computer science students who are interested in learning about software process improvement, and in how it can assist software companies in achieving high-quality and reliable software on-time and on budget. It will also be of interest to software engineers, quality professionals, and software managers who are involved in software process improvement programmes as well as the motivated general reader.

[4]The exception to this may be companies in India where the marketing benefit of CMMI level 4 and level 5 maturity is significant. Many software companies in India need to be appraised at CMMI level 5 to be taken seriously as well as to attract and retain high-calibre staff.

Acknowledgements

I am deeply indebted to family and friends who supported my efforts in this endeavour. I wish to express a special thanks to former colleagues in industry and academia. I thank Dr. Micheal Mac An Airchinnigh, Dr. Andrew Butterfield, and Dr. Hugh Gibbons for a wonderful experience at Trinity College Dublin; Dr. Richard Messnarz for an interesting CMM assessment at Friedrichshafen, Germany; Dr. David Parnas (a grandfather of computer science) for sharing his interesting theoretical world of software quality (I do not see it solving the problem of delivering high-quality software on-time and on-budget); Rohit Dave of Motorola who was superb at the corporate quality game; John Murphy of DeCare Systems Ireland; John Wall of Lecan; Linda Booth, Charlie Smith, Mike Bew, and Neil Hawthorn of Allianz in the UK for a wonderful cultural experience in England; Uinsionn O'Connor and the team in EirGrid for an interesting CMMI level 3 implementation in Dublin; and Sarah Farrell of the Centre of Software Engineering, Dublin.

Cork, Ireland Gerard O'Regan

Contents

1 Motivation for Software Process Improvement 1
1.1 Introduction 1
1.2 What Is a Software Process? 2
1.3 What Is Software Process Improvement? 4
1.4 What Are the Benefits of Software Process Improvement? 5
1.5 What Models Are Used in Software Process Improvement? 6
1.6 Process Mapping 8
1.7 Process Improvement Initiatives 9
1.8 Barriers to Success 10
1.9 Review Questions 11
1.10 Summary 11

2 Software Engineering 13
2.1 Introduction 13
2.2 What Is Software Engineering? 16
2.3 Challenges in Software Engineering 18
2.4 Software Processes and Life Cycles 20
2.4.1 Waterfall Life Cycles 21
2.4.2 Spiral Life Cycles 22
2.4.3 Rational Unified Process 23
2.4.4 Agile Development 25
2.5 Life Cycle Phases 27
2.5.1 Business Requirements Definition 27
2.5.2 Specification of System Requirements 30
2.5.3 Design 30
2.5.4 Implementation 31
2.5.5 Testing 32
2.5.6 Maintenance 33
2.6 Software Inspections 35

2.7 Software Testing . . . 36
2.8 Software Project Management . . . 37
2.9 Process Maturity Models . . . 39
2.10 Review Questions . . . 40
2.11 Summary . . . 40

3 Capability Maturity Model Integration . . . 43
3.1 Introduction . . . 43
3.2 The CMMI . . . 46
3.3 CMMI Maturity Levels . . . 49
3.3.1 CMMI Representations . . . 52
3.4 Categories of CMMI Processes . . . 55
3.5 CMMI Process Areas . . . 56
3.6 Components of CMMI Process Areas . . . 58
3.7 SCAMPI Appraisals . . . 63
3.8 Review Questions . . . 64
3.9 Summary . . . 65

4 Setting Up a CMMI Initiative . . . 67
4.1 Introduction . . . 67
4.2 Approach to Continuous Improvement . . . 68
4.3 CMMI Improvement Structure and Teams . . . 70
4.3.1 Setting Up the SEPG Team . . . 74
4.3.2 Setting Up the Steering Group . . . 74
4.3.3 Setting Up Dedicated Improvement Sub-teams . . . 75
4.3.4 Role of the CMMI Project Manager . . . 76
4.3.5 Risks to Success . . . 76
4.4 Planning the Improvement Cycle . . . 77
4.4.1 Appraisals . . . 78
4.4.2 CMMI Project Plan . . . 80
4.4.3 CMMI Project Schedule . . . 80
4.4.4 CMMI Kick-Off Session . . . 81
4.5 Implementation of Improvements . . . 82
4.5.1 Process Mapping . . . 83
4.5.2 Layout of Templates . . . 84
4.5.3 Layout of Procedures and Guidelines . . . 85
4.6 Piloting the Process . . . 85
4.7 Rolling Out Process . . . 86
4.8 Review Questions . . . 86
4.9 Summary . . . 87

5 CMMI Level 2 Implementation . . . 89
5.1 Introduction . . . 89
5.2 Project Management . . . 91
5.2.1 Project Management Processes . . . 93

5.2.2 Project Management Procedure or Guidelines 96
5.2.3 Project Management Templates 98
5.2.4 Project Management Checklist 99
5.2.5 Institutionalization 99
5.3 Supplier Agreement Management 100
5.3.1 Supplier Agreement Management Process 102
5.3.2 Supplier Agreement Management Procedure 102
5.3.3 Supplier Agreement Management Templates 104
5.3.4 Supplier Agreement Management Checklist 105
5.3.5 Institutionalization 105
5.4 Requirements Development and Management 106
5.4.1 Requirements Process Map 110
5.4.2 Requirements Procedure 111
5.4.3 Requirements Templates 113
5.4.4 Requirements Checklist 114
5.4.5 Institutionalization 116
5.5 Configuration Management 116
5.5.1 Configuration Management Process Map 120
5.5.2 Configuration Management Procedure 121
5.5.3 Configuration Management Templates 122
5.5.4 Configuration Management Checklist 122
5.5.5 Institutionalization 124
5.6 Process and Product Quality Assurance 125
5.6.1 Audit Process 126
5.6.2 Audit Procedure 126
5.6.3 Audit Templates 128
5.6.4 Audit Checklist 128
5.6.5 Institutionalization 129
5.7 Measurement and Analysis 131
5.7.1 Measurement Process 132
5.7.2 Measurement Procedure 134
5.7.3 Measurement Templates 134
5.7.4 Measurement Checklist 134
5.7.5 Institutionalization 134
5.8 Review Questions 137
5.9 Summary 137

6 CMMI Level 3 Implementation 139
6.1 Introduction 139
6.2 Organization Process Focus 142
6.2.1 Improvement Process 143
6.2.2 Improvement Procedure 144
6.2.3 Improvement Templates 144
6.2.4 Improvement Checklist 145
6.2.5 Institutionalization 145

6.3 Organization Process Definition 148
6.3.1 Standard Process 149
6.3.2 Procedure and Guidelines 151
6.3.3 Templates 152
6.3.4 Checklist 152
6.3.5 Institutionalization 152
6.4 Organization Training 153
6.4.1 Organization Training Process 155
6.4.2 Training Procedure 155
6.4.3 Training Templates 156
6.4.4 Training Checklist 157
6.4.5 Institutionalization 158
6.5 Requirements, Design, and Development 158
6.5.1 Design Process 162
6.5.2 Design and Development Procedure 162
6.5.3 Design Templates 164
6.5.4 Design Checklist 164
6.5.5 Institutionalization 165
6.6 Verification and Validation 166
6.6.1 Testing and Peer Review Processes 169
6.6.2 Testing and Peer Review Procedures 169
6.6.3 Testing and Peer Review Templates 175
6.6.4 Testing and Peer Review Checklists 175
6.6.5 Institutionalization 177
6.7 Integrated Project Management 178
6.7.1 Tailoring Process 180
6.7.2 Tailoring Procedure 180
6.7.3 Tailoring Templates 181
6.7.4 Tailoring Checklist 182
6.7.5 Institutionalization 182
6.8 Risk Management 182
6.8.1 Risk Management Process 185
6.8.2 Risk Management Guidelines 185
6.8.3 Risk Management Templates 186
6.8.4 Risk Management Checklist 186
6.8.5 Institutionalization 187
6.9 Decision Analysis and Resolution 187
6.9.1 Decision-Making Process 189
6.9.2 Decision-Making Guidelines 191
6.9.3 Decision-Making Templates 191
6.9.4 Checklist 192
6.9.5 Institutionalization 192
6.10 Review Questions 193
6.11 Summary 193

7 CMMI Level 4 and 5 Implementation 195
7.1 Introduction 195
7.2 Organization Process Performance 196
7.3 Quantitative Project Management 197
7.4 Organization Innovation and Deployment 199
7.5 Causal Analysis and Resolution 200
7.6 Review Questions 201
7.7 Summary 201

8 Software Engineering Tools 203
8.1 Introduction 203
8.2 Tools for Project Management 204
8.3 Tools for Requirements 208
8.4 Tools for Design and Development 211
8.5 Tools for Configuration Management and Change Control 213
8.6 Tools for Code Analysis and Code Inspections 215
8.7 Tools for Testing 216
8.8 Review Questions 218
8.9 Summary 218

9 SCAMPI Appraisals 221
9.1 Introduction 221
9.2 Planning and Requirements for the Appraisal 224
9.2.1 Analyse Requirements 225
9.2.2 Develop Appraisal Plan 226
9.2.3 Select and Prepare Team 227
9.2.4 Obtain and Analyse Initial Evidence 228
9.2.5 Prepare for Conducting Appraisal 229
9.3 Conducting the Appraisal 230
9.3.1 Prepare Participants 230
9.3.2 Examine Objective Evidence 231
9.3.3 Document Objective Evidence 233
9.3.4 Verify Objective Evidence 234
9.3.5 Validate Preliminary Findings 235
9.3.6 Generate Appraisal Results 235
9.4 Reporting the Results 236
9.4.1 Deliver Appraisal Results 236
9.4.2 Archive Appraisal Results 237
9.5 Review Questions 237
9.6 Summary 238

Glossary 239

References 243

Index 245