THE DARK SIDE OF SOFTWARE ENGINEERING
IEEE Computer Society Publications
The world-renowned IEEE Computer Society publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. These books are available from most retail outlets. Visit the CS Store at http://computer.org/store for a list of products.

IEEE Computer Society / Wiley Partnership
The IEEE Computer Society and Wiley partnership allows the CS Press authored book program to produce a number of exciting new titles in areas of computer science, computing and networking with a special focus on software engineering. IEEE Computer Society members continue to receive a 15% discount on these titles when purchased through Wiley or at wiley.com/ieeecs

To submit questions about the program or send proposals please e-mail kguillemette@computer.org or write to Books, IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314. Telephone +1-714-816-2169.

Additional information regarding the Computer Society authored book program can also be accessed from our web site at http://computer.org/cspress.
CONTENTS

FOREWORD ix
   Linda Rising

INTRODUCTION 1
I.1 What’s the Dark Side? 1
   I.1.1 Why the Dark Side? 2
   I.1.2 Who Cares About the Dark Side? 3
   I.1.3 How Dark is the Dark Side? 5
   I.1.4 What Else is on the Dark Side? 7
   I.1.5 Ethics and the Dark Side 8
   I.1.6 Personal Anecdotes About the Dark Side 11
Reference 14

PART 1
DARK SIDE ISSUES 15

CHAPTER 1 SUBVERSION 17
I.1 Introductory Case Studies and Anecdotes 17
   I.1.1 A Faculty Feedback System 18
   I.1.2 An Unusual Cooperative Effort 21
   I.1.3 Lack of Cooperation due to Self Interest 22
   I.1.4 An Evil Teammate 22
   I.1.5 Thwarting the Evil Union 24
I.2 The Survey: Impact of Subversive Stakeholders On Software Projects 24
   I.2.1 Introduction 25
   I.2.2 The Survey 26
   I.2.3 The Survey Findings 27
   I.2.4 Conclusions 34
   I.2.5 Impact on Practice 35
   I.2.6 Impact on Research 35
   I.2.7 Limitations 35
   I.2.8 Challenges 36
   I.2.9 Acknowledgments 37
I.3 Selected Responses 37
   I.3.1 Sample Answers to the Question: “What Were the Motivations and Goals of the Subversive Stakeholders?” 37
   I.3.2 Sample Answers to the Question “How Were the Subversive Attacks Discovered?” 45
   I.3.3 Sample Answers to the Question “How Can Projects be Defended Against Subversive Stakeholders?” 49
CONTENTS

1.4 A Follow-Up to the Survey: Some Hypotheses and Related Survey Findings 56
References 80

CHAPTER 2 LYING 81
2.1 Introductory Case Studies and Anecdotes 81
2.2 Incidents of Lying: The Survey 86
   2.2.1 The Survey Results 87
   2.2.2 General Scope 87
   2.2.3 An Overview of the Problem 88
   2.2.4 Clarification of Terms 89
   2.2.5 Discussion 93
   2.2.6 Conclusions 93
   2.2.7 Limitations 94
2.3 Qualitative Survey Responses on Lying 95
2.4 What Can Be Done About Lying? 96
2.5 The Questionnaire Used in the Survey 107
References 112

CHAPTER 3 HACKING 113
3.1 Case Studies of Attacks and Biographies of Hackers 113
3.2 Cyber Terrorism and Government-Sponsored Hacking 118
3.3 The Hacker Subculture 121
   3.3.1 Why They Are Called “Hackers” 121
   3.3.2 Motivation of Hackers 121
   3.3.3 Hacker Slang 122
   3.3.4 Hacker Ethics 123
   3.3.5 Public Opinion about Hackers 130
3.4 How a Hacker Is Identified 132
3.5 Time Line of a Typical Malware Attack 135
3.6 Hacker Economy: How Does a Hacker Make Money? 136
3.7 Social Engineering 142
   3.7.1 Social Engineering Examples and Case Studies 143
   3.7.2 Tactics of Social Engineering 151
3.8 A Lingering Question 153
3.9 Late-Breaking News 154

CHAPTER 4 THEFT OF INFORMATION 157
4.1 Introduction 157
4.2 Case Studies 158
   4.2.1 Data Theft 158
   4.2.2 Source Code Theft 161
4.3 How Do the Victims Find Out That Their Secrets Are Stolen? 164
4.4 Intellectual Property Protection 166
   4.4.1 Trade Secret Protection 167
   4.4.2 Copyright Protection 169
   4.4.3 Patent Protection 169
   4.4.4 Steganography 170
4.5 Open Versus Closed Source 170
CHAPTER 5  ESPIONAGE  175
5.1 Introduction  175
5.2 What Is Espionage?  176
5.3 Case Studies  177
  5.3.1 Sweden Versus Russia  178
  5.3.2 Shekhar Verma  178
  5.3.3 Lineage III  179
  5.3.4 GM versus VW: Jose Ignacio Lopez  179
  5.3.5 British Midland Tools  179
  5.3.6 Solid Oak Software  180
  5.3.7 Proctor & Gamble versus Unilever  181
  5.3.8 News Corp Versus Vivendi  181
  5.3.9 Spying: Was A TI Chip Really Stolen by a French Spy?  181
  5.3.10 Conficker  183
5.4 Cyber Warfare  185
Reference  187

CHAPTER 6  DISGRUNTLED EMPLOYEES AND SABOTAGE  189
6.1 Introduction and Background  189
6.2 Disgruntled Employee Data Issues  192
  6.2.1 Data Tampering  192
  6.2.2 Data Destruction  194
  6.2.3 Data Made Public  196
  6.2.4 Theft Via Data  199
6.3 Disgruntled Employee Software Issues  199
  6.3.1 Software Destruction  199
6.4 Disgruntled Employee System Issues  200
6.5 What to Do About Disgruntled Employee Acts  203
6.6 Sabotage  206
References  212

CHAPTER 7  WHISTLE-BLOWING  213
7.1 A Hypothetical Scenario  215
7.2 Whistle-Blowing and Software Engineering  217
7.3 More Case Studies and Anecdotes  220
  7.3.1 Jeffrey Wigand and Brown and Williamson Tobacco  220
  7.3.2 A Longitudinal Study of Whistle-Blowing  221
  7.3.3 An Even More Pessimistic View  222
  7.3.4 Academic Whistle-Blowing  223
  7.3.5 The Sum Total of Whistle-Blowing  224
References  225

APPENDIX TO CHAPTER 7  PRACTICAL IMPLICATIONS OF THE RESEARCH INTO
WHISTLE-BLOWING  227
References  240
## CONTENTS

**PART 2**

**VIEWPOINTS ON DARK SIDE ISSUES** 243

**Introduction** 243

**CHAPTER 8**  **OPINIONS, PREDICTIONS, AND BELIEFS** 245

8.1 Automated Crime 246
   *Donn B. Parker*
   Information Sources 257

8.2 Let’s Play Make Believe 258
   *Karl E. Wiegers*
   Reference 260

8.3 Dark, Light, or Just Another Shade of Grey? 261
   *Les Hatton*

8.4 Rational Software Developers as Pathological Code Hackers 264
   *Norman Fenton*

**CHAPTER 9**  **PERSONAL ANECDOTES** 269

9.1 An Officer and a Gentleman Confronts the Dark Side 270
   *Grady Booch*

9.2 Less Carrot and More Stick 273
   *June Verner*
   References 275

9.3 “Them and Us”: Dispatches from the Virtual Software Team Trenches 276
   *Valentine Casey*

9.4 What is it to Lie on a Software Project? 281
   *Robert N. Britcher*

9.5 “Merciless Control Instrument” and the Mysterious Missing Fax 284
   *A. H. (anonymous)*

9.6 Forest of Arden 289
   *David Alan Grier*

9.7 Hard-Headed Hardware Hit Man 292
   *Will Tracz*

9.8 A Lighthearted Anecdote 294
   *Eugene Farmer*

**CONCLUSIONS** 299

**INDEX** 303
Robert Glass has always been one who “boldly goes” where the more cautious fear to tread. I have been a fan of his writing for, well, let’s just say, a long time. I remember when he started telling the truth as he saw it about software development and was forced to change the names of the companies and products that he was discussing—he even changed his own name to conceal authorship of published accounts. I remember teaching a course on structured design (using the green book by Yourdon and Constantine—that’s how long ago that was!) and if I finished a class early, I would say to my students, “You can go now or I can read another story by Robert Glass.” No one ever left before the story was finished. “Cornbelt Shakedown” (from Glass and DeNim [1980]) was a favorite. Many of these stories are the kind of humor that leads you to wonder, “Why am I laughing? To keep from crying?”

Later, as I was working in the industry, I led a study group on Software Runaways (Glass 1997) and experienced the serious side of Robert Glass. Very little of the wry and witty here, but, instead, a lot of lessons for serious consideration.

Robert Glass, joined in this book with Johann Rost, is still at it. He continues to be (I can’t resist) fearless! (The reference is to my own book, Manns and Rising [2005]). I don’t know Johann except through his work on this book, which is excellent, and from what I’ve been told—that he’s a German former IT consultant now living in beautiful Romania, the land of Transylvania, Dracula, and Ceaușescu … it’s no wonder the book has a “dark side” theme! This book is also full of stories about real projects at real companies. Names are named. The result is a compelling look at the dark side of computer programming. We are all hardwired to learn from stories, especially when we can identify with the protagonists.

Hacking, espionage, sabotage, theft, whistle-blowing, subversion, disgruntled employees who want to get even—and, of course, the dance of deception. We’ve all seen it—where we know and they know, in fact, everyone knows—but we all smile and keep dancing as long as we can. The authors cut in on this charade and force us to wake up and take stock.

Robert and Johann also include the results of their serious research. They have certainly done their homework. There’s an abundance of citations to back up their observations. The survey data on sabotage is fascinating!

This reporting is way out of the box; in fact, these authors are standing on the box and they share with us a good look at the terrain—something most of us just don’t take the time to do; we prefer to rush ahead and ignore the lessons of the past.

So, take a moment. We need a breather now and then. We need to step back and retrospect on the history of our industry and think about a better way of working.
within it. Robert Glass and Johann Rost are offering us a chance to do just that. Stop. Listen. Think. Is this the road that will serve us best for the next part of our journey?

REFERENCES

INTRODUCTION

I.1 WHAT’S THE DARK SIDE?

The dictionary doesn’t give a definition for “dark side.” Not even my heavyweight dictionary that I can barely lift. Oh, it defines words such as “dark” (“secret, mysterious, evil,” among other things), “darken” (“perplex, make foul, sully, cast a gloom upon”) and “darksome” (“dark, dismal”). So you get the idea—things that are on the dark side tend to be evil, gloomy, dismal.

That’s not a surprise to most of our readers, we suspect. The “dark side” has a sort of intuitive meaning that we all grasp and is (pretty much) in sync with those related dictionary definitions. Things that are on the dark side of the computing profession would be things that we wouldn’t necessarily want to be a part of or approve of.

I, Robert, remember an incident from my days of child-raising, when one of my sons played on a little league baseball team. There was a pitcher on that team whose father, like me, attended nearly all of the games. When his son was pitching, the father would shout to his son, from time to time, “Throw the dark one.” I never knew exactly what he meant by that cry. But I always assumed that it wasn’t so much about a particular pitch his son could throw but about intimidating the opposing batter, who might become convinced that the pitch to come was somehow evil and be less likely to make contact with it because of that.

In any case, even on the baseball diamond, the words “dark” and therefore “dark side” have an intuitively universal meaning.

It’s interesting that, if you know the software literature—be it the popular computing press, the academic journals, or even the general popular press—you would be aware that it doesn’t say very much about dark side issues. Oh, it says a lot about project success and project failure but that’s a different kettle of fish. Projects that fail may be in a sense “dark” but not in the sense of “evil.” We tend to assume, without ever saying so, that even projects that fail, do so largely because of some kind of ineptitude, not because of some kind of evil.

Let me be perfectly clear about what we are doing here. This is NOT a book about software project failure, or about prescriptive thinking about how to build software better. This is a book about the EVIL THINGS that happen on computing and software projects—what the kinds of evil are, how they manifest themselves, and what we good guys can do about them. I emphasize this point because a lot of folks we’ve asked to review the book’s material keep thinking that this is “Yet
Another Book About Project Failure” (YABAPF) or “Yet Another Book About Doing Software Engineering Right” (YABADSER)!

Where might we find discussions of dark side matters in the traditional software engineering literature? Look at the topics that literature on computing and software tend to be divided into. They are usually organized into these topics:

- Problem-solving
- Computer hardware
- Systems/software
- Data/information
- Application problem domains
- Systems/software management
- Organizations
- Society
- Disciplinary issues

This list is derived from the computing research topics explored in the series of papers culminating in Glass, Ramesh, and Vessey (2004).

Where in that list of topics would you look to find “dark side” topics? Perhaps in “systems/software management.” Perhaps in “disciplinary issues.” It doesn’t fit comfortably into either of those topics, but it could be forced to fit—inconveniently—into them. But the fact of the matter is, any taxonomy of computing topics you choose is unlikely to provide a convenient home for this issue of the dark side. It is, in other words, a topic that people writing about computing have not only avoided over time; they have avoided it because it doesn’t fit nicely into any list of topics that describe the field.

And that brings us to the topic of the next section.

I.1.1 Why the Dark Side?

Both authors of this book have been intrigued by the lack of discussion of dark side issues in computing literature. We were both aware, from personal experience, that dark side things happened in the field. But hardly anyone seemed to be talking about them. Perhaps more importantly, hardly anyone was researching them. For example, how often did dark side matters affect computing and software projects?

I, Johann, had initially thought about exploring this issue. I knew from personal experience the effect of dark side behavior: For example, subversion on software projects, while it does not occur often, has serious repercussions when it does. Because of that, and because of the lack of any appearance whatsoever of “subversion” in computing literature, I conducted a study to determine its prevalence, its effects, and ways of overcoming it. That survey is presented as a chapter later in this book. It is a pioneering study in the software field; to this date, no one else has explored this topic.
My co-author, Robert, came at the subject from a different direction. He was surprised while presenting a topic at a software seminar; the seminar attendees hijacked the session and diverted it to talking about lying as a problem in the software project world. The attendees were vehement—lying was a big-time problem in the projects on which they had been involved. Because of that, and because—once again—of the lack of any significant appearance of the topic of “lying” in the computing literature—he began to explore that topic in more depth.

It was about then that we met one another. (It is interesting to note, in this day of electronic communication, that we have only met on the Web, never in person!) I was having trouble finding a leading journal willing to publish my subversion paper, that is, the one that resulted from his survey. I asked Robert for help, and—to make a long story shorter—the result became a co-authored paper that eventually was published in a leading journal.

Intrigued by the subversion study, Robert suggested that we conduct a similar study about lying. As we have said, neither topic was discussed much in any of the literatures surrounding the field. So the two of us, together with another contributor named Matthias Matook, performed a study in the form of a survey about the prevalence of lying, its effects, and ways of overcoming it. Eventually, to make this long story also shorter, that too was published. Variations and enhancements of the two published papers are presented later in this book.

By then, we had become thoroughly intrigued by these topics, and we began to see them as part of a broader issue: “dark side” issues on computing projects. We expanded the topic into more and more sub-topics, eventually identifying seven dark side matters that affected these projects: subversion, lying, hacking, theft of information, espionage, disgruntled employees and sabotage, and whistle-blowing. There is a chapter of this book devoted to each of those topics.

We considered doing thorough research into the latter five topics, but decided that there was sufficient material in the literature of those more-often covered topics; so we relied on already published case studies, not the survey research that we conducted about subversion and lying, to cover them. (To be honest, that research was extremely laborious and time-consuming, and we were reluctant to engage in it beyond what we had already done!)

And then there is another final fact that brought the interest in dark side matters to a head: Robert has published a number of books and articles on the subject of failed computing projects. (As we said earlier, there is not a direct link between failure and dark side matters, but the two are similar enough to draw the same kind of interest.) He had been intrigued by failure and became equally intrigued by dark side matters!

### I.1.2 Who Cares About the Dark Side?

The short answer to this question, of course, is that we hope YOU do! We chose to write about the dark side because we were interested in the subject and because we felt we had some contributions to make on the subject. Our fervent hope is that you, our intended reader, will also be interested in what we have to say.