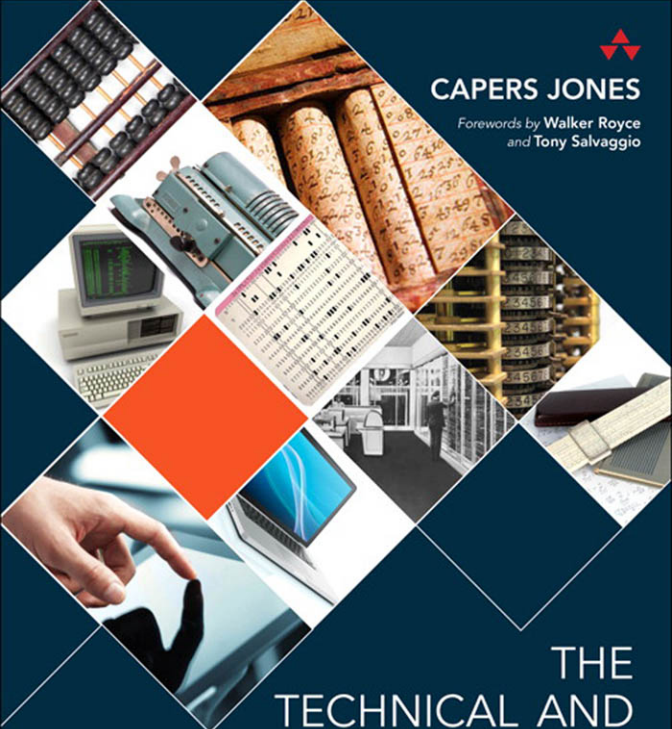




CAPERS JONES

Forewords by Walker Royce  
and Tony Salvaggio



# THE TECHNICAL AND SOCIAL HISTORY OF **SOFTWARE ENGINEERING**

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



# The Technical and Social History of Software Engineering

*This page intentionally left blank*

---

---

# The Technical and Social History of Software Engineering

Capers Jones

◆Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales  
(800) 382-3419  
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales  
international@pearson.com

Visit us on the Web: [informit.com/aw](http://informit.com/aw)

*Library of Congress Cataloging-in-Publication Data:*

Jones, Capers.

The technical and social history of software engineering / Capers Jones.  
pages cm

Includes bibliographical references and index.

ISBN 978-0-321-90342-6 (alk. paper)—ISBN 0-321-90342-0 (alk. paper)

1. Software engineering—History.
  2. Computer software—Development—Social aspects.
  3. Computer software industry—History.
  4. Computer software industry—Social aspects. I. Title.
- QA76.758.J644 201  
005.1--dc23

2013036017

Copyright © 2014 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to (201) 236-3290.

Cover design by Chuti Prasertsith

Cover images © Jim Richardson/National Geographic Image Collection/Alamy; © Graeme Dawes/Shutterstock; © Tim Jenner/Shutterstock; © Franck Boston/Shutterstock; © iFong/Shutterstock; © Hintau Aliaksei/Shutterstock; © dvande/Shutterstock; © Everett Collection/Shutterstock; © anaken2012/Shutterstock; © Peshkova/Shutterstock

ISBN-13: 978-0-321-90342-6

ISBN-10: 0-321-90342-0

Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.  
First printing, November 2013

# Contents

Foreword by Walker Royce . . . . .	vii
Foreword by Tony Salvaggio . . . . .	ix
Preface . . . . .	xix
Acknowledgments . . . . .	xxvii
About the Author. . . . .	xxix
<b>Chapter 1: Prelude: Computing from Ancient Times to the Modern Era . . .</b>	<b>1</b>
The Human Need to Compute . . . . .	1
Early Sequence of Numerical Knowledge . . . . .	3
Inventions for Improved Mathematics . . . . .	8
Mathematics and Calculating. . . . .	10
Recording Information. . . . .	14
Communicating Information . . . . .	15
Storing Information . . . . .	17
Enabling Computers and Software. . . . .	18
Key Inventions Relevant to Software . . . . .	21
Alphabetic Languages. . . . .	21
Binary and Decimal Numbers and Zero. . . . .	21
Digital Computers . . . . .	22
Higher-Level Programming Languages. . . . .	22
Random-Access Storage . . . . .	23
The Impact of Software on People and Society. . . . .	23
Beneficial Tools and Applications. . . . .	23
Harmful Inventions . . . . .	32
Weighing the Risks. . . . .	34
Summary. . . . .	35
<b>Chapter 2: 1930 to 1939: The Foundations of Digital Computing. . . . .</b>	<b>37</b>
The First Innovators of Modern Computing. . . . .	37
Small Mathematical Applications . . . . .	40
Summary. . . . .	42

<b>Chapter 3: 1940 to 1949: Computing During World War II and the Postwar Era</b> . . . . .	<b>43</b>
Global Conflict and Computing . . . . .	43
Wartime Innovations. . . . .	45
Analog Computers During World War II . . . . .	46
Computers in Germany During World War II . . . . .	47
Computers in Japan During World War II . . . . .	50
Computers in Poland During World War II . . . . .	51
Computers in France During World War II . . . . .	52
Computers in Australia During World War II . . . . .	52
Computers in Russia During World War II . . . . .	53
Computers in Great Britain During World War II . . . . .	54
Computers in the United States During World War II . . . . .	59
Computers in the Postwar Era . . . . .	68
The Cold War Begins . . . . .	69
Postwar Computer Development . . . . .	69
Historical Contributions of the Decade. . . . .	73
Building Software in 1945. . . . .	75
Summary. . . . .	76
<b>Chapter 4: 1950 to 1959: Starting the Ascent of Digital Computers and Software</b> . . . . .	<b>77</b>
Military and Defense Computers in the 1950s . . . . .	77
SAGE . . . . .	78
BOMARC . . . . .	80
Innovators of the 1950s . . . . .	81
Programming Languages of the 1950s . . . . .	84
The First Commercial Computers . . . . .	85
LEO . . . . .	85
IBM . . . . .	86
Other Computer Business Implementations . . . . .	88
Software Applications in the 1950s. . . . .	89
Function Points in 1955 . . . . .	90
Summary. . . . .	91
<b>Chapter 5: 1960 to 1969: The Rise of Business Computers and Business Software</b> . . . . .	<b>93</b>
An Evolving Workforce. . . . .	93

Early Specialized Outsourcing . . . . .	95
Computer Programmers in the 1960s . . . . .	96
Becoming a Programmer . . . . .	96
A High Demand for Programmers . . . . .	100
Emergence of the Software Engineer . . . . .	103
IBM System/360 . . . . .	104
The Turing Award . . . . .	106
The Invention of the Credit Card . . . . .	109
Automation and New Professions . . . . .	110
The DEC PDP-1 . . . . .	111
Programming Languages of the 1960s . . . . .	112
The Computer Business of the 1960s . . . . .	115
Litigation Changes the Computer World Forever . . . . .	118
Computers and Software in Space . . . . .	119
Computer and Software Growth in the 1960s . . . . .	120
Function Points Backfired for 1965 . . . . .	122
Summary . . . . .	123
<b>Chapter 6: 1970 to 1979: Computers and Software Begin</b>	
<b>Creating Wealth . . . . .</b>	<b>125</b>
Software Evolution in the 1970s . . . . .	125
Trends in Software . . . . .	126
Political Failures . . . . .	127
Rapid Rise of Computer Companies . . . . .	128
Major Companies Formed During the 1970s . . . . .	129
FedEx . . . . .	129
NASDAQ . . . . .	129
Southwest Airlines . . . . .	130
Computer and Software Companies Formed During the 1970s . .	130
Altair Computers . . . . .	131
Apple Computer . . . . .	132
Baan . . . . .	132
Computer Associates (CA) . . . . .	132
Cray Computers . . . . .	134
Cullinane . . . . .	134
Digital Research . . . . .	135
Galorath Incorporated . . . . .	136



IMSAI Computers . . . . .	137
InterSystems Corporation. . . . .	137
Lawson Software . . . . .	138
Microsoft . . . . .	138
Oracle . . . . .	139
Price Systems . . . . .	140
Prime Computers (Pr1me) . . . . .	141
Systems Applications Programs (SAP) . . . . .	141
Tandem Computers . . . . .	142
Yourdon, Inc. . . . .	144
The Impact of Companies Founded During the 1970s . . . . .	145
The Troublesome Growth of Software Applications . . . . .	146
Numerous Fragmented Software Subcategories. . . . .	147
Advanced and Experimental Software . . . . .	147
Business Software. . . . .	147
Communications Software . . . . .	148
Cybercrime and Hacking Software. . . . .	148
Database Software . . . . .	149
Education Software . . . . .	149
Embedded Software . . . . .	150
Gaming and Entertainment Software . . . . .	151
Manufacturing Software. . . . .	151
Middleware Software . . . . .	152
Military Software . . . . .	152
Open-Source Software . . . . .	153
Personal Software. . . . .	154
Programming Tool Software. . . . .	155
Project Management Software . . . . .	155
Scientific and Mathematical Software . . . . .	155
Security and Protective Software . . . . .	156
Social Network Software . . . . .	156
Survey Tools Software . . . . .	157
Utility Software . . . . .	157
A Lawsuit That Changed Computer History . . . . .	158
Background Enabling Inventions. . . . .	159
Function Points in 1975 . . . . .	162
Summary. . . . .	163

<b>Chapter 7: 1980 to 1989: The Rise of Personal Computers and Personal Software . . . . .</b>	<b>165</b>
Rapid Changes in Computing . . . . .	166
Companies Formed During the 1980s . . . . .	168
Accenture . . . . .	171
Adobe . . . . .	172
America Online (AOL) . . . . .	173
Avira . . . . .	174
BlackBerry (Research in Motion, or RIM) . . . . .	174
Borland . . . . .	175
Computer Aid, Inc. (CAI) . . . . .	177
Cisco Systems . . . . .	178
Digital Consulting Institute (DCI) . . . . .	179
Huawei . . . . .	180
Intuit . . . . .	183
KPMG . . . . .	184
Lotus . . . . .	186
NeXT . . . . .	187
PeopleSoft . . . . .	188
Rational . . . . .	189
SEI . . . . .	192
Software Productivity Research (SPR) . . . . .	194
Stepstone Corporation . . . . .	197
Symantec . . . . .	198
TechSoup Global . . . . .	200
Wolfram Research . . . . .	202
The Growth of Software During the 1980s . . . . .	204
Results for 1,000 Function Points Circa 1985 . . . . .	204
Summary . . . . .	205
<b>Chapter 8: 1990 to 1999: Expansion of the World Wide Web and the Rise of Dot-Coms . . . . .</b>	<b>207</b>
Emergence of the World Wide Web . . . . .	208
Other Innovations of the 1990s . . . . .	210
Companies Formed During the 1990s . . . . .	212
Akamai . . . . .	214
Amazon . . . . .	214

Apache . . . . .	216
Craigslist . . . . .	216
Digital Playground . . . . .	217
eBay . . . . .	218
GoDaddy . . . . .	220
Google . . . . .	221
Heartland Payment Systems . . . . .	223
Insight Venture Partners . . . . .	224
The ISBSG . . . . .	225
Monster.com . . . . .	227
Netscape Communications . . . . .	228
Priceline . . . . .	229
Red Hat Software . . . . .	230
Red Storm Entertainment . . . . .	231
Sirius Satellite Radio . . . . .	232
ThoughtWorks . . . . .	234
Visio . . . . .	235
VMware . . . . .	236
Mass Updates and Aging Legacy Software . . . . .	237
Incompatibilities of International Date Formats . . . . .	239
The Expansion of Outsourcing . . . . .	243
Growth of Software Applications During the 1990s . . . . .	243
Results for 1,000 Function Points Circa 1995 . . . . .	244
Summary . . . . .	245
<b>Chapter 9: 2000 to 2009: The Rise of Social Networks and Economic Crises . . . . .</b>	<b>247</b>
The Dot-Com Bubble . . . . .	248
The Great Recession . . . . .	250
Innovations of the 2000s . . . . .	254
Companies Formed During the 2000s . . . . .	257
AbsolutData Research . . . . .	260
Andreesen Horowitz . . . . .	261
Archon Information Systems . . . . .	262
Canonical, Ltd. . . . .	262
Facebook . . . . .	263

Freelanthropy . . . . .	265
Global Insight . . . . .	266
HIVE Group . . . . .	267
Intellectual Ventures . . . . .	268
Internet Marketing Association (IMA) . . . . .	270
Meeting Zone . . . . .	270
Moody's Analytics . . . . .	271
Mozilla Foundation . . . . .	272
Open Source Development Labs (OSDL) . . . . .	274
PerfectMatch . . . . .	275
RPX Corporation . . . . .	276
SolveIT Software . . . . .	277
Twitter . . . . .	278
YouTube . . . . .	279
Zillow . . . . .	281
Growth of Software from 2000 to 2010 . . . . .	283
Results for 1,000 Function Points Circa 2005 . . . . .	284
Summary . . . . .	285
<b>Chapter 10: 2010 to 2019: Clouds, Crowds, Blogs, Big Data, and Predictive Analytics . . . . .</b>	<b>287</b>
Predicting the Future . . . . .	288
Professional Status for Software Engineering . . . . .	289
Possible Software Engineering Improvements in the 2010s . . . . .	293
Companies Formed During the Early 2010s . . . . .	296
AngelPad . . . . .	298
Authr . . . . .	299
CloudVelocity . . . . .	299
CrowdCube . . . . .	300
Fiverr . . . . .	301
Flattr . . . . .	302
Geekli.St. . . . .	303
GoFundMe . . . . .	304
Namcook Analytics LLC . . . . .	304
Nest Labs . . . . .	308

- Peer Index . . . . . 309
- Unified Inbox . . . . . 310
- Yesware . . . . . 310
- Predicting New Companies and Products from
  - 2013 to 2019 . . . . . 311
  - Big Data . . . . . 312
  - Crowd Intelligence and Crowdfunding . . . . . 312
  - Cybercrime and Cybersecurity . . . . . 313
  - Education . . . . . 314
  - Intelligent Agents . . . . . 321
  - Medical Devices . . . . . 322
  - Predictive Analytics . . . . . 323
  - Wearable Computers . . . . . 325
- Projected Growth of Software from 2010 to 2019 . . . . . 327
- Results for 1,000 Function Points Circa 2015 . . . . . 328
- Summary . . . . . 329

**Chapter 11: Modern Software Problems . . . . . 331**

- Analysis of Major Software Failures . . . . . 331
  - 1962: Failure of the Mariner 1 Navigation Software . . . . . 333
  - 1978: Hartford Coliseum Collapse . . . . . 333
  - 1983: Soviet Early Warning System . . . . . 334
  - 1986: Therac 25 Radiation Poisoning . . . . . 334
  - 1987: Wall Street Crash . . . . . 335
  - 1990: AT&T Telephone Lines Shutdown . . . . . 336
  - 1991: Patriot Missile Target Error . . . . . 336
  - 1993: Intel Pentium Chip Division Problem . . . . . 337
  - 1993: Denver Airport Delays . . . . . 337
  - 1996: Ariane 5 Rocket Explosion . . . . . 338
  - 1998: Mars Climate Orbiter Crash . . . . . 339
  - 1999: Failure of the British Passport System . . . . . 339
  - 2000: The Y2K Problem . . . . . 340
  - 2004: Shutdown of Los Angeles Airport (LAX) Air-Traffic
    - Controls . . . . . 341
  - 2005: Failure to Complete the FBI Trilogy Project . . . . . 342
  - 2005: Secret Sony Copy Protection Software . . . . . 343

2006: Airbus A380 Wiring Problem . . . . .	344
2010: McAfee Antivirus Bug Shuts Down Computers . . . . .	345
2011: Failed Investment in Studio 38 in Rhode Island . . . . .	345
2012: Knight Capital Stock-Trading Software Problems . . . . .	347
2012: Automotive Safety Recalls Due to Software . . . . .	348
Summary . . . . .	349
<b>Chapter 12: A Brief History of Cybercrime and Cyberwarfare . . . . .</b>	<b>351</b>
A New Form of Crime . . . . .	351
The Hacker Invasion . . . . .	352
Preparing Defenses . . . . .	354
Increasing Seriousness of Cyberattacks . . . . .	356
A Growing Number of Victims . . . . .	358
Types of Cyberattacks . . . . .	362
Bluetooth Hijacking . . . . .	363
Botnets . . . . .	364
Browser Hijacking . . . . .	365
Computer Voting Fraud . . . . .	366
Cyberwarfare Against Civilian Targets . . . . .	371
Data Theft from Corporations . . . . .	372
Data Theft from Unsecured Networks . . . . .	373
Denial of Service Attacks . . . . .	374
Electromagnetic Pulses (EMPs) . . . . .	375
Email Address Harvesting . . . . .	376
Identity Thefts . . . . .	378
Java Vulnerability Attacks . . . . .	379
Keyboard Trackers . . . . .	380
Macro Attacks in Word and Excel Documents . . . . .	381
Pharming . . . . .	381
Phishing . . . . .	381
Rootkits . . . . .	383
Skimming . . . . .	384
Smart Card Hijacking . . . . .	385
Spam . . . . .	387
SQL Injections . . . . .	388
Trojans . . . . .	388

Viruses . . . . .	389
Worms . . . . .	390
Zero-Day Security Attacks . . . . .	391
The Odds of Being Attacked . . . . .	392
Improving Defenses Against Cyberattacks . . . . .	394
Raising Our Immunity to Cyberattacks . . . . .	396
Access Controls . . . . .	399
Authentication Controls . . . . .	400
EMP Protection . . . . .	401
Encryption . . . . .	401
Estimating Cyberattack Recovery Costs . . . . .	402
Insurance Against Cybertheft and Cyberattack Damages . .	403
Secure Programming Languages . . . . .	404
The Increasing Frequency and Costs of Cyberattacks in the United States . . . . .	404
Summary . . . . .	406
<b>Appendix A: Annotated Bibliography and References . . . . .</b>	<b>407</b>
<b>Index . . . . .</b>	<b>423</b>

# Foreword by Walker Royce

Over the last several decades, the software industry has advanced at a breakneck pace. Few of us have stopped to reflect on all the foundations, breakthroughs, and know-how that have made software the world's most dominant product.

Our world now depends on software. It is everywhere, in almost every man-made thing, and used by almost everybody. All that invisible stuff in our phones, cars, gadgets, banks, and hospitals was once considered to be magical. Now it is taken for granted as just another necessary part, service, or feature. Don't you wonder how it all evolved? And where it came from? And why it was built? If you are curious about the evolution of all this magical technology, this book provides an authoritative chronology of software's evolution. Moreover, if your profession depends on software—and there are very few that don't—you will find this book to be a valuable and educational history lesson. It is loaded with quantified benchmarks of performance that you won't find anywhere else in the literature.

As a young engineer, I was introduced to Capers Jones through his books and papers on software measurement. He was one of the go-to thought leaders on software economics, and he was bold enough to publish facts and figures that helped quantify the challenges and opportunities. He has written more than a dozen books since then. When I wrote my first book in the 1990s, Capers was one of my top choices for peer review. To some degree back then, we were competitors, and his review of my manuscript was . . . well . . . let's just say that it was brutal. However, his review was, by far, the most valuable, insightful, and constructive. He knows how to write. His strong convictions are credible because he compiles extensive data and statistics on software quality and productivity across diverse industries. The big lesson he taught me was this: *In the software world dominated by uncertainty, the person with the best data will be the most persuasive.* His critique was effective for steering me in a better direction, and we have maintained a great professional relationship ever since by frequently exchanging ideas, presentations, and provocative positions.

The measurements and forecasts of progress, quality, and business trends in most software organizations sound like the sleight-of-hand statistics quoted by politicians rather than the matter-of-fact measurements expressed by engineers



and scientists. Is this statement too harsh? No. Politicians have a well-deserved reputation and a track record similar to the software industry for under-delivering on committed forecasts and productivity improvements. The software marketplace is full of cynical customers because their experience with software productivity improvement—internally from their own people as well as externally from vendors—is plagued by hyperbole and spin. Software delivery endeavors have a high degree of uncertainty and complexity.

Reducing uncertainty through better measurement can increase trust among consumers, suppliers, and developers. Through his decades of work on measurement, Capers Jones has contributed immensely to the trust we put in today's software industry. Capers is a great writer and an authority on software history. He was a firsthand participant in the evolution of software from its infancy. This book synthesizes his research, knowledge, and quantified insights into a history lesson that every software professional will find useful and every software user will find enlightening. His writing is fluid, engaging, and crisp. Enjoy this measured story of software advances and learn from it.

—Walker Royce  
*Chief Software Economist, IBM*

# Foreword by Tony Salvaggio

When we founded Computer Aid, Inc., (CAI) in the early 1980s, we stated that our business mission would be to strive toward thought leadership in the areas of software engineering, software development productivity, and application support productivity. The founders of CAI had a strong belief that doing things the right way, repeatedly, would unlock tremendous business value for our future clients. Although our startup team had deep experience in software engineering, as well as large-project design and development, we knew that such experience alone would not be sufficient to qualify us as “thought leaders.”

There is a great quote from Isaac Newton about how scientists “stand on the shoulders of giants.” Newton meant that all scientific discovery and progress—in particular, his own historic breakthroughs—were built upon the hard work and insight of previous individuals. At CAI, our team members and associates have all stood solidly on the shoulders of Capers Jones. Over the course of CAI’s 30-year history, Capers has been the most learned, most knowledgeable, and most prolific discoverer in the software engineering industry, perhaps even in the entire history of computing. He has written more than eighteen books across the entire spectrum of IT management, and each one has unlocked and revealed new insights, both for engineers and managers. I have personally given countless executive-level presentations on productivity and process improvement while waving a copy of Capers’s original thick “yellow book” in one hand.

Capers has accumulated, without doubt, the most comprehensive data on every aspect of software engineering, and he has performed the most scientific analysis on this data. To say that he has forgotten more than the average top software professional ever learns would be an understatement. In his new book, Capers performs yet another invaluable service to our industry, and for each current and future IT professional, by documenting, for the first time, the long and fascinating history of information technology.

This profession, which was unheard of in the 1960s and ’70s, has evolved through so many dramatic changes in the course of my lifetime. I have seen the software industry lead the business reengineering revolution and watched how this, in turn, has revolutionized life on our planet for billions of people. History will repeat itself, whether we study it or not, and in this sense, Capers’s new

book is a must-read for every software engineering student and IT professional. In spite of our revolutionary successes, there has been a consistent record within the IT industry of not diligently putting to work the lessons of the past, the lessons first documented so well in Capers's original "yellow book."

Over these past three decades, Capers has become a treasured friend of mine who often starts my day with early morning emails in which we discuss quality management, removing early defects, and avoiding project failures. His words ring in my ear with their clarity and insight and for thirty years have helped me guide our business here at CAI. I highly recommend Capers's new book, as well as many of his earlier works. The messages are timeless in their value.

—Tony Salvaggio  
*CEO and President, Computer Aid, Inc. (CAI)*

# Preface

I was born prior to World War II and therefore just before the dawn of the computer era. From growing up, I have personal recollections of the announcement of the transistor being invented and of reading about integrated circuits. I also remember the arrival of television and later of color television.

When I attended the University of Florida, there were no on-campus computers, no computer science programs, and no software engineering programs. In fact, engineering students still used slide rules, and there was an active debate about whether new electronic calculators could be used for exams.

There were no personal computers, no personal music players, no social networks other than fraternities and sororities, and certainly no smartphones. There were no embedded software applications and no embedded medical devices such as cochlear implants; all of these things would come later.

Older readers have lived through the entire history of the computing and software industries from the very beginning. So many inventions have occurred so rapidly, and so many companies have sprung up, that they tend to blur together. We are living in remarkable times with technical advances occurring almost every month.

This is the fifteenth book I have written. Although I had been a professional programmer in the 1960s, my first eleven books, which included *Programming Productivity*; *Assessment and Control of Software Risks*; *Applied Software Measurement*; and *Estimating Software Costs*, were all about software management issues. I became interested in management topics while working at IBM when I was commissioned to develop IBM's first software estimating tool in 1973 with a colleague, Dr. Charles Turk.

My first book was published while I was at IBM. Later, I moved to ITT and then founded my own software company when ITT sold its telecommunication businesses. In general, I have written a book every two years.

As a lifelong reader of *Scientific American*, I like to stay current on scientific topics. (One of the highlights of my publication career was publishing an article about sizing software in *Scientific American* in December 1998. This article featured function point metrics.)

Having sold my first software company in 1998, my wife and I moved to Rhode Island, a state where I had never lived before but where my wife was

born and had many relatives. Soon after we arrived, the history of the state attracted my interest.

The economic history of Rhode Island was almost a microcosm of the U.S. economy, having started with ship building and commerce, then manufacturing, then moving toward services as labor costs drove out manufacturing. In 2006, I published *The History and Future of Narragansett Bay*, which was my first non-software-related book as well as my first history book.

The “future” part of the Rhode Island book dealt with modern problems that are also becoming endemic: rising taxes; unsustainable government pensions; pollution of the Bay and fresh-water aquifers; political corruption; new exotic diseases such as West Nile virus and Lyme disease; the dwindling numbers of professionals such as physicians and dentists; and an ever-expanding bureaucracy that primarily supports special interests rather than the general population. These are national problems as well as state and local problems.

In any case, having written a history of Rhode Island, it seemed useful to consider a history of the software engineering field, but at the time I had other book projects in mind. Two of these other books were *Software Engineering Best Practices* and *The Economics of Software Quality*. I was also busy starting a new software company, Namcook Analytics LLC, with my business partner, Ted Maroney.

The specific event that led to this book was a casual visit to a used book store adjacent to the University of Rhode Island. At the store, I happened to pick up a book with an interesting title: Paul Starr’s book *The Social Transformation of American Medicine*. This book won a Pulitzer Prize in 1984 and is highly recommended for software professionals. It shows the transformation of medicine from a craft with barely adequate training to the top tier of respected professions with perhaps the best training of any profession.

Starr’s book was the inspiration for this book. Software “engineering” is still also a craft and only approaching the status of being a true profession. For example, licensing is just getting started for software; formal specialization and board certifications are still in the future. Malpractice monitoring is in the future. Starr’s book contains a good road map for what software engineering needs to accomplish.

I had always had an interest in medical topics since my first programming job was in the Office of the Surgeon General at the U.S. Public Health Service in Washington, D.C. We were working on software for the National Institutes of Health.

In fact, one of my earlier books from 1994 was titled *Assessment and Control of Software Risks*. This book used the exact structure and format of a medical

textbook titled *Control of Communicable Diseases in Man*. The medical format has worked very well for discussing software problems.

There have been so many inventions and so many companies springing up in the computer and software domains that this new book needed a workable structure. What I decided was to look at software innovations, inventions, and companies decade by decade starting in 1930 and running through 2012 and beyond. Social and professional organizations such as the Institute of Electrical and Electronics Engineers, the Association for Computing Machinery, the Society for Information Management, SHARE, and so on would also be discussed.

The final chapter begins in 2010 and includes projections of potential future progress through 2019. This is reminiscent of the “future” chapter of my Rhode Island history, which also projected ten years from the completion of the book.

However, starting in 1930 was a bit too abrupt. Therefore, I decided to add a prelude chapter that would summarize the human drive toward faster computation from ancient times through the modern era. The overall structure of the book includes 12 chapters:

- Chapter 1 is a prelude on computing from ancient times to the current era. It deals with several interlinked topics, including the evolution of mathematics; the drive to speed up mathematical calculations using mechanical devices; methods for communicating mathematical results from person to person; and methods for storing or archiving mathematical results for historical purposes, including famous libraries from the ancient world.
- Chapter 2 deals with the 1930s and discusses the foundations of digital computing and software. The seminal works of Alan Turing, Konrad Zuse, and other pioneers are covered. The Great Depression was in force during this decade, and many companies failed. IBM came close to failing, but the arrival of social security in 1935 revived IBM earnings and led toward future growth for forty-five years in a row. Without social security, IBM might not have survived the decade, and computer and software history would be very different than it is today.
- Chapter 3 deals with the 1940s. This chapter covers computers and software among the belligerent countries during World War II and also the postwar era. The famous British code-breaking devices at Bletchley Park are discussed, as are Konrad Zuse’s computers in Germany. However, during World War II, analog computers were the real workhorses, so

the book also discusses ship-board gun controls, torpedo-aiming computers, bombsights, and other analog computing devices. The end of the chapter deals with the early electronic digital computers and the dawn of programming as we know it today.

- Chapter 4 deals with the 1950s. This decade witnessed computers and software moving from military and scientific purposes to business purposes. Two huge efforts bracketed this decade: The SAGE air-defense system at the beginning of the decade and the SABRE airline reservation system at the end were the two largest systems built up until that time. Many enabling inventions occurred, such as transistors and integrated circuits. High-level programming languages, like COBOL, began to appear.
- Chapter 5 deals with the 1960s. This decade saw computers and software becoming business tools for hundreds of corporations. Physical sizes of computers shrank as transistors and integrated circuits replaced tubes and discrete wiring. This decade also saw IBM growing rapidly due to computers such as the IBM 1401 and later the System 360. Minicomputers and special computers also emerged. Software expanded as operating systems and database applications made computers easier to use. Some universities began offering computer science and software engineering degree programs. Software jobs were exploding in numbers.
- Chapter 6 deals with the 1970s. This decade witnessed the birth of Apple and Microsoft and a push toward commercial packages. Several companies began to use computers to create new business models such as Southwest Airlines and Federal Express, with its unique hub-and-spoke arrangement to optimize shipping logistics. Software engineering became a common academic subject. Programming jobs expanded rapidly. Structured development emerged to control software chaos as applications got larger and harder to manage. Several companies founded in this decade would later grow and create wealth beyond imagination and become global powerhouses: Apple and Microsoft are two. Embedded medical devices, such as cochlear implants, appeared.
- Chapter 7 deals with the 1980s. This decade is clearly dominated by the IBM personal computer and the advent of the DOS and Windows operating systems. Hundreds of specialized software companies sprang up like mushrooms. Programming jobs continued to grow rapidly in numbers. Object-oriented development and object-oriented languages

began to appear. Programming languages expanded from dozens to hundreds for reasons that are hard to understand. Personal computers began to move toward portability, although the first of these weighed more than twenty-five pounds. The Software Engineering Institute (SEI) was founded to assist the military sector in achieving better and more reliable software.

- Chapter 8 deals with the 1990s. The big news during this decade was the development and rapid expansion of the internet and the World Wide Web. Toward the end of this decade, the famous dot-com bubble began to inflate as hundreds of companies tried to market products and services via the web. This bubble burst early within the next decade. Cybercrime began to expand as the internet made remote hacking of data centers fairly easy to accomplish. Outsourcing, in particular international outsourcing, expanded rapidly as companies decided that building their own software programs was not cost-effective.
- Chapter 9 deals with the 2000s. The start of this decade saw the bursting of the dot-com bubble. However, dot-coms that survived, such as Amazon, would grow to become giants. Social networks appeared, as did new search engines and new web browsers. The Agile development method began to expand in popularity, but so did others, such as the team software process (TSP) and the rational unified process (RUP). The number of programming languages topped 2,500 by the end of the decade and continues to grow, with new languages appearing almost every month. All of these programming languages and the aging of software make maintenance very expensive. During this decade, maintenance and support of legacy software applications moved past new software development as the dominant work of the industry. A new subindustry of “patent trolls” appeared, and patent litigation became endemic among computer, software, and telecommunication companies as they each tried to use patents to damage competitors and push ahead.
- Chapter 10 deals with the 2010s, with speculation about possible future inventions. Current trends that will expand include clouds, crowds, big data, and predictive analytics. Some possible future inventions may be wearable computers, virtual education, and significant advances in embedded medical devices. Quantum computing may occur, with another increase in speed and another reduction in physical size. Intelligent agents



will become increasingly powerful in extracting useful information from heterogeneous, big data sources. Cybercrime will certainly increase and cyberwarfare is already happening. The nations of the world now have formal cyberwarfare units, and attacks on industrial, financial, and military sectors are becoming common.

- Chapter 11 deals with topics that are difficult to pin down to a specific decade. This chapter revisits famous software failures and explains what happened and how they might have been avoided. It seemed better to show these in one place than to separate them by decade.
- Chapter 12 outlines the nature and forms of various cybercrime and cyberwarfare issues, which are becoming increasingly severe and increasingly common. Here, too, there are so many kinds of cyberattacks that it was best to put them in one chapter in order to emphasize their magnitude and seriousness.

History books are enjoyable for authors to write. Hopefully, this book will be enjoyable to read. It quickly became obvious while writing it that if the book attempted to include every company and every invention that appeared during this timeframe, it might top 1,000 pages, which no publisher would want and probably no reader would want either.

Therefore, quite a few companies are omitted in the interest of space. When a number of companies occupy a similar niche, only one or two are cited to explain the niche. There is no need, for example, to name fifty static analysis companies, fifty computer game companies, twenty-five webinar tool companies, or twenty-five antivirus companies.

---

**Note**

It is an interesting social characteristic of the software industry that as soon as a niche becomes hot, dozens of similar companies and products rush into it. It is sometimes hard for a new invention to get venture funding, but it is much easier for the next dozen companies within the same space.

---

When stringing together dates and timelines, some of the source information is inconsistent. One source might say a company was founded in 1982, while another might cite 1983 for the same company. Hopefully, this book is generally correct in timelines and dates, but it is easy to be off by a year in either direction.

The purpose of this book is to show the overall sweep of progress and the bubbles of inventions that keep occurring. The software engineering field has been one of the most innovative and exciting fields in human history, and I hope younger readers will enjoy learning about older inventions that might have occurred before they were born. I hope older readers will enjoy reading about the many new inventions such as social networks and (soon) wearable computers.

*This page intentionally left blank*

# Acknowledgments

As always, thanks to my wife, Eileen, for her support through fifteen books over a thirty-year period. Thanks also to my business partner, Ted Maroney, for his interest and support of my various patents and inventions.

Thanks to Bernard Goodwin, acquisitions editor at Addison-Wesley Professional, for his support of this book and several of my past books, too. Thanks to the capable editorial and production staff as well.

Many thanks to the reviewers of the drafts of this book and also of my older books, because often the same reviewers have seen more than one. Thanks to Rex Black, Gary Gack, Peter Hill, Leon Kappelman, Alex Pettit, Walker Royce, and Joe Schofield. Some unofficial reviewers, such as Tom DePetrillo, Pontus Johnson, Tony Salvaggio, Paul Strassmann, and Jerry Weinberg, also deserve thanks.

Thanks also to the editors of web journals who have published excerpts from this book and some of my older books: Andrew Binstock of *Dr. Dobb's Journal*; Greg Hutchins of the Certified Enterprise Risk Management Academy; Ben Linders of InfoQ; and Michael Milutis of the Information Technology Metrics and Productivity Institute.

All of us in the software field owe thanks to the pioneers and inventors who make this field so interesting: Al Albrecht, Barry Boehm, Fred Brooks, Ward Cunningham, Esther Dyson, Bill Gates, Grace Hopper, Watts Humphrey, Steve Jobs, Steve Kan, Mitch Kapor, Ken Olson, Alan Turing, An Wang, Jerry Weinberg, Stephen Wolfram, and hundreds more.

Over the years, I've had the good fortune of meeting several senior executives who understood the value of software to the world and to their companies. These executives funded research centers chartered to improve software methods and practices, and I was fortunate to work in some of them.

Among these top corporate executives have been Thomas J. Watson, Jr., of IBM, Harold Geneen and Rand Araskog of ITT, Mort Myerson of Electronic Data Systems, and Dr. Hishahi Tomino of Kozo Keikaku Engineering. Dr. Tomino's company has translated most of my older books into Japanese, and the translation teams did an excellent job. Hopefully, this new book will also find its way into Japanese and other languages.

Software and computers have changed human communications in profound ways. Today, many people have more virtual friends than real friends. Some young people spend more time texting and using social networks than speaking face to face. The internet and World Wide Web have opened up vast new collections of information, larger than the sum of every library in the world. Almost every complex device is now controlled by embedded software, including automobiles, aircraft, and even smart appliances. Computers and software have changed the world, and more changes are still in store for us.

# About the Author



**Capers Jones** is a cofounder, vice president, and chief technology officer of Namcook Analytics LLC. Namcook Analytics builds patent-pending advanced risk, quality, and cost-estimation tools. The website is [www.namcook.com](http://www.namcook.com). Capers Jones's blog is <http://namcookanalytics.com>.

Until cofounding Namcook Analytics LLC in 2011, he was the president of Capers Jones & Associates LLC from 2000 through 2011.

He is also the founder and former chairman of Software Productivity Research LLC (SPR). Capers Jones founded SPR in 1984.

Before founding SPR, Capers was assistant director of Programming Technology for the ITT Corporation at the Programming Technology Center in Stratford, Connecticut. He created the first software measurement program at ITT.

Capers Jones was also a manager and software researcher at IBM in California, where he designed IBM's first software cost-estimating tools in 1973 and 1974.

In total, Capers Jones has designed seven proprietary software estimation tools and four commercial software estimation tools.

Capers Jones is a well-known author and international public speaker. Some of his books have been translated into five languages. His four most recent books are *The Economics of Software Quality* (Addison-Wesley, 2011); *Software Engineering Best Practices* (McGraw-Hill, 2010); *Applied Software Measurement, Third Edition* (McGraw-Hill, 2008); and *Estimating Software Costs, Second Edition* (McGraw-Hill, 2007).

*The Technical and Social History of Software Engineering* is his second history book. This book was inspired by Paul Starr's book *The Social Transformation of American Medicine*, which won a Pulitzer Prize in 1984.

Among Capers's older book titles are *Patterns of Software Systems Failure and Success* (Prentice Hall, 1994); *Software Quality: Analysis and Guidelines for Success* (International Thomson, 1997); and *Software Assessments, Benchmarks, and Best Practices* (Addison-Wesley Longman, 2000).

# Chapter 1

---

---

## Prelude: Computing from Ancient Times to the Modern Era

The human need to compute probably originated in prehistory when humans began to accumulate physical possessions. It soon became desirable to keep track of how many specific possessions (e.g., cattle) were owned by a family or tribe. Once simple addition and subtraction became possible, a related need was to record the information so it could be kept for long time periods and could be shared with others. Early recording devices were pebbles or physical objects, but it was eventually found that these could be replaced with symbols.

As humans evolved and began to settle in communities, other calculating needs arose, such as measuring the dimensions of bricks or marking off fields. With leisure came curiosity and a need for more complex calculations of time, distance, and the positions of the stars.

Fairly soon, the labor involved with calculations was seen as burdensome and tedious, so mechanical devices that could speed up calculations (the abacus being among the first) were developed.

Tools for assisting with logical decisions were the last to be developed. The needs for rapid calculations, long-range data storage, and complex decision making were the critical factors that eventually came together to inspire the design of computers and software.

---

### The Human Need to Compute

A book on the history of software engineering and computers should not just start abruptly at a specific date such as 1930. It is true that digital computers



and the beginnings of software were first articulated between 1930 and 1939, but many prior inventions over thousands of years had set the stage.

From ancient times through today, there was a human need for various kinds of calculations. There has also been a human need to keep the results of those calculations in some kind of a permanent format.

Another human need that is harder to articulate is the need for logical analysis of alternative choices. An example of such a choice is whether to take a long flat road or a short hilly road when moving products to a marketplace. Another choice is what kind of crop is most suited to a particular piece of land.

More important alternatives are whether or not a community should go to war with another community. In today's world, some choices have life and death importance, such as what is the best therapy to treat a serious medical condition like antibiotic-resistant tuberculosis.

Other choices have economic importance. The Republicans and Democrats are examples of totally opposite views of what choices are best for the U.S. economy.

For choices with diametrically opposing alternatives, it is not possible for both sides to be right, but it is easily possible for both sides to be wrong. (It is also possible that some other choice and neither of the alternatives is the best.)

From analysis of what passes for arguments between the Democrats and Republicans, both sides seem to be wrong and the end results will probably damage the U.S. economy, no matter which path is taken.

From the point of view of someone who works with computers and software on a daily basis, it would not be extremely difficult to create mathematical models of the comparative impacts on the economy of raising taxes (the Democratic goal), reducing spending (the Republican goal), or some combination of both.

But instead of rational discussions augmented by realistic financial models, both sides have merely poured out rhetoric with hardly any factual information or proof of either side's argument. It is astonishing to listen to the speeches of Republicans and Democrats. They both rail against each other, but neither side presents anything that looks like solid data.

The same kinds of problems occur at state and municipal levels. For example, before the 2012 elections, the General Assembly of Rhode Island passed unwise legislation that doubled the number of voters per voting station, which effectively reduced the places available for citizens to vote by half.

The inevitable results of this foolish decision were huge lines of annoyed voters, waits of up to four hours to vote, and having to keep some voting stations open almost until midnight to accommodate the voters waiting in line.

This was not a very complicated issue. The numbers of voters passing through voting stations per hour have been known for years. But the Rhode Island Assembly failed to perform even rudimentary calculations about what halving the number of voting stations would do to voter wait times.

As a result, in the 2012 elections, many Rhode Island citizens who could not afford to wait four hours or more simply left without voting. They were disfranchised by the folly of a foolish law passed by an inept general assembly. This law by the Rhode Island Assembly was incompetent and should never have been passed without mathematical modeling of the results of reducing polling places on voting wait times.

The point of carping about governments passing unwise laws and issuing foolish regulations is because in today's world, computers and software could easily provide impact assessments and perhaps even eliminate thoughts of passing such foolish laws and regulations.

The fact that humans have used mathematics, made logical choices, and kept records from prehistory through today brings up questions that are relevant to the history of software and computers:

- What kinds of calculations do we use?
- What kinds of information or data do we need to save?
- What are the best storage methods for long-range retention of information?
- What methods of analysis can help in making complicated choices or decisions?
- What are the best methods of communicating data and knowledge?

It is interesting to consider these five questions from ancient times through the modern era and see how computers and software gradually emerged to help in dealing with them.

---

## Early Sequence of Numerical Knowledge

Probably soon after humans could speak they could also count, at least up to ten, by using their fingers. It is possible that Neanderthals or Cro-Magnons could count as early as 35,000 years ago, based on parallel incised scratches on

both a wolf bone in Czechoslovakia from about 33,000 years ago and a baboon bone in Africa from about 35,000 years ago.

Whether the scratches recorded the passage of days, numbers of objects, or were just scratched as a way to pass time is not known. The wolf bone is the most interesting due to having 55 scratches grouped into sets of five. This raises the probability that the scratches were used to count either objects or time.

An even older mastodon tusk from about 50,000 years ago had 16 holes drilled into it, of unknown purpose. Because Neanderthals and Cro-Magnons overlapped from about 43,000 BCE to 30,000 BCE, these artifacts could have come from either group or from other contemporaneous groups that are now extinct.

It is interesting that the cranial capacity and brain sizes of both Neanderthals and Cro-Magnons appear to be slightly larger than modern homo sapiens, although modern frontal lobes are larger. Brain size does not translate directly into intelligence, but it does indicate that some form of abstract reasoning might have occurred very early. Cave paintings date back more than 40,000 years, so at least some form of abstraction did exist.

In addition to counting objects and possessions, it was also important to be able to keep at least approximate track of the passage of time. Probably the length of a year was known at least subjectively more than 10,000 years ago. With the arrival of agriculture, also about 10,000 years ago, knowing when to plant certain crops and when to harvest them would have aided in food production.

One of the first known settlements was Catal Huyuk in Turkey, dating from around 7,000 BCE. This village, constructed of mud bricks, probably held several hundred people. Archaeological findings indicate agriculture of wheat, barley, and peas. Meat came from cattle and wild animals.

Findings of arrowheads, mace heads, pottery, copper, and lead indicate that probably some forms of trading took place at Catal Huyuk. Trading is not easily accomplished without some method of keeping track of objects. There were also many images painted on walls and this may indicate artistic interests.

The probable early sequence of humans acquiring numerical knowledge may have started with several key topics:

- Prehistoric numeric and mathematical knowledge:
  - Counting objects to record ownership
  - Understanding the two basic operations of addition and subtraction
  - Measuring angles, such as due east or west, to keep from getting lost

- Counting the passage of time during a year to aid agriculture
- Counting the passage of daily time to coordinate group actions
- Numeric and mathematical knowledge from early civilizations:
  - Counting physical length, width, and height in order to build structures
  - Measuring weights and volumes for trade purposes
  - Measuring long distances such as those between cities
  - Measuring the heights of mountains and the position of the sun above the horizon
  - Understanding the mathematical operations of multiplication and division
- Numeric and mathematical knowledge probably derived from priests or shamans:
  - Counting astronomical time such as eclipses and positions of stars
  - Measuring the speed or velocity of moving objects
  - Measuring curves, circles, and irregular shapes
  - Measuring rates of change such as acceleration
  - Measuring invisible phenomena such as the speed of sound and light
- Numeric and mathematical knowledge developed by mathematicians:
  - Analyzing probabilities for games and gambling
  - Understanding abstract topics such as zero and negative numbers
  - Understanding complex topics such as compound interest
  - Understanding very complex topics such as infinity and uncertainty
  - Understanding abstract topics such as irrational numbers and quantum uncertainty

Prehistoric numeric and mathematical knowledge probably could have been handled with careful observation assisted by nothing more than tokens such as stones or scratches, plus sticks for measuring length. Addition and subtraction are clearly demonstrated by just adding or removing stones from a pile.

Numeric and mathematical knowledge from early civilizations would have needed a combination of abstract reasoning aided by physical devices. Obviously, some kind of balance scale is needed to measure weight. Some kind of angle calculator is needed to measure the heights of mountains. Some kind of recording method is needed to keep track of events, such as star positions over long time periods.

Numeric and mathematical knowledge probably derived from priests or shamans would need a combination of abstract reasoning; accurate time keeping; accurate physical measures; and awareness that mathematics could represent intangible topics that cannot be seen, touched, or measured directly. This probably required time devoted to intellectual studies rather than to farming or hunting.

Numeric and mathematical knowledge developed by mathematicians is perhaps among the main incentives leading to calculating devices and eventually to computers and software. This required sophisticated knowledge of the previous topics, combined with fairly accurate measurements and intellectual curiosity in minds that have a bent for mathematical reasoning. These probably originated with people who had been educated in mathematical concepts and were inventive enough to extend earlier mathematical concepts in new directions.

One of the earliest cities, Mohenjo-Daro, which was built in Northern India about 3,700 years ago, shows signs of sophisticated mathematics. In fact, balance scales and weights have been excavated from Mohenjo-Daro.

This city may have held a population of 35,000 at its peak. The streets are laid out in a careful grid pattern; bricks and construction showed signs of standard dimensions and reusable pieces. These things require measurements.

Both Mohenjo-Daro and another city in Northern India, Harappa, show signs of some kind of central authority because they are built in similar styles. Both cities produced large numbers of clay seals incised both with images of animals and with symbols thought to be writing, although these remain undeciphered. Some of these clay seals date as far back as 3,300 BCE.

Other ancient civilizations also developed counting, arithmetic, measures of length, and weights and scales. Egypt and Babylonia had arithmetic from before 2,000 BCE.

As cities became settled and larger, increased leisure time permitted occupations to begin that were not concerned with physical labor or hunting. These occupations did not depend on physical effort and no doubt included priests and shamans. With time freed from survival and food gathering, additional forms of mathematical understanding began to appear.

Keeping track of the positions of the stars over long periods, measuring longer distances such as property boundaries and distances between villages, and measuring the headings and distances traveled by boats required more complex forms of mathematics and also precise measurements of angles and time periods. The advent of boat building also required an increase in mathematical knowledge. Boat hulls are of necessity curved, so straight dimensional measurements were not enough.

Rowing or sailing a boat in fresh water or within sight of land can be done with little or no mathematical knowledge. But once boats began to venture onto the oceans, it became necessary to understand the positions of the stars to keep from getting lost.

Australia is remote from all other continents and was not connected by a land bridge to any other location since the continents broke up. Yet it was settled about 40,000 years ago, apparently by means of a long ocean voyage from one (or more) of the continents. The islands of Polynesia and Easter Island are also far from any mainland and yet were settled thousands of years ago. These things indicate early knowledge of star positions and some kind of math as well.

Many early civilizations in Egypt, Mesopotamia, China, India, and South America soon accumulated surprisingly sophisticated mathematical knowledge. This mathematical knowledge was often associated with specialists who received substantial training.

Many ancient civilizations, such as the ancient Chinese, Sumerians, Babylonians, Egyptians, and Greeks, invested substantial time and energy into providing training for children. Not so well known in the West are the similar efforts for training in India and among the people of Central and South America, such as the Olmecs, Mayans, Incans, and later the Aztecs.

Japan also had formal training. For the upper classes, Japanese training included both physical skills in weapons and also intellectual topics such as reading, writing, and mathematics. All of these ancient civilizations developed formal training for children and also methods of recording information.

The University of Nalanda in Northern India was founded circa 472 BC and lasted until about the 12th century, with a peak enrollment during around 500 AD. It was one of the largest in the ancient world, with more than 10,000 students from throughout Asia and more than 2,000 professors. It was among the first universities to provide training in mathematics, physics, medicine, astronomy, and foreign languages.

The University of Nalanda had an active group of translators who translated Sanskrit and Prakrit into a variety of other languages. In fact, much of the

information about the University of Nalanda comes from Chinese translations preserved in China since the University of Nalanda library was destroyed during the Moslem invasion of India in the 12th century. It was reported to be so large that it burned for almost six weeks.

Indian scholars were quite advanced even when compared to Greece and Rome. Concepts such as zero and the awareness of numerous star systems were known in India prior to being known in Europe. (The Olmecs of Central America also used zero prior to the Greeks.)

In ancient times, out of a population of perhaps 1,000 people in a Neolithic village, probably more than 950 were illiterate or could only do basic counting of objects and handle simple dimensional measures. But at least a few people were able to learn more complex calculations, including those associated with astronomy, construction of buildings and bridges, navigation, and boat building.

---

## Inventions for Improved Mathematics

From the earliest knowledge of counting and numerical concepts, those who used numerical information were troubled by the needs for greater speed in calculating and for greater reliability of results than the unaided human mind could provide. In order to explain the later importance of computers and software, it is useful to begin with some of the earliest attempts to improve mathematical performance.

It is also useful to think about what computers and software really do and why they are valuable. The services that are provided to the human mind by various calculating devices include, but are not limited to, the following:

- Basic arithmetic operations of addition, subtraction, multiplication, and division
- Scientific mathematics, including powers, sines, cosines, and others
- Financial mathematics, including simple and compound interest and rates of return
- Logical calculations, such as routing and choices between alternatives
- Calculations of time, distance, height, and speed

- Deriving useful inductive knowledge from large collections of disparate information
- Deductive logic, such as drawing conclusions from rules

In doing research for this chapter, a great many interesting and useful sources were found during my web searches. For example, IBM has a graphical history of mathematics that can even be downloaded onto iPhones. Wikipedia and other web sources have dozens of histories of computer hardware and some histories of software development, too. More than a dozen computer museums were noted in a number of countries, such as the London Science Museum, which has a working version of the Babbage analytical engine on display.

For this book, it seemed useful to combine six kinds of inventions that are all synergistic and ultimately related to each other as well as to modern software.

*Mathematics* is the first of these six forms of invention. Calculating devices, computers, and software were all first invented to speed up mathematical calculations. Mathematics probably started with addition and subtraction and were then followed later by multiplication and division. After that, many other and more abstract forms appeared: geometry, trigonometry, algebra, and calculus, for example.

The second form of invention is the *recording of ideas and information* so they can be shared and transmitted and also to keep the ideas available over long time periods. The inventions in this category include writing systems and physical storage of writing. Physical storage of writing includes stone tablets, clay tablets, papyrus, animal skins, paper, and eventually magnetic and optical storage. Storage also includes manuscripts, books, libraries, and eventually databases and cloud storage.

The third form of invention is that of physical *calculating devices* that could assist human scholars in faster and more accurate calculations than would be possible using only the human mind and the human body. Tables of useful values were perhaps the first method used to speed up calculations. Physical devices include the abacus, protractors, astrolabes, measuring devices, mechanical calculating devices, slide rules, analog computers, and eventually electronic digital computers.

A fourth form of invention involves the available *channels for distributing information* to many people. The first channel was no doubt word of mouth and passing information along to be memorized by students or apprentices. But soon information transmission started to include markings on stones and bones; markings on clay; and eventually pictographs, ideographs, and finally alphabets.



The fifth form of invention is that of *software* itself. This is the most recent form of invention; essentially all software used in 2013 is less than 55 years old, probably more than 50% of the software is less than 20 years old.

A sixth form of invention is indirect. These are *enabling inventions* that are not directly connected to computers and software but that helped in their development. One such enabling invention is the patent system. A second and very important enabling invention was plastic.

## Mathematics and Calculating

Table 1.1 shows the approximate evolution of mathematics, calculating devices, and software from prehistory through the modern era. It is intended to show the overall sweep of inventions and is not a precise timeline. The table focuses on the inventions themselves rather than providing the names of the inventors, such as Newton, Leibnitz, Turing, Mauchly, von Neumann, Hopper, and many others. The topics in Table 1.1 that eventually had an impact on computers and software are shown in *italic*.

**Table 1.1** *Evolution of Mathematics, Calculating Devices, and Software*

Mathematics, Calculating Devices, and Software	Approximate Number of Years Prior to 2013
Counting objects	35,000
Addition and subtraction	30,000
Measuring angles	25,000
Counting the annual passage of time	20,000
<i>Pebbles used for calculation</i>	<i>20,000</i>
Counting the daily passage of time	19,000
Quantifying physical length, width, and height	18,000
Measuring weights and volumes	15,000
Measuring long distances between towns	10,000
Measuring astronomical time	7,000
Geometry	5,500
Sundials	5,500
Measuring the height of the sun and mountains	5,000

**Table 1.1** (Continued)

Mathematics, Calculating Devices, and Software	Approximate Number of Years Prior to 2013
Multiplication and division	4,500
Measuring the speed of moving objects	4,000
<i>Analog computing devices</i>	4,000
Algebra	4,000
Trigonometry	4,000
Fractions	4,000
Multiplication tables	3,900
Clocks: water	3,300
Decimal numbers	3,100
<i>Abacus and mechanical calculations</i>	3,000
Clocks: mechanical	3,000
Binary numbers	2,700
Zero	2,600
Measuring curves, circles, and irregular objects	2,500
Measuring temperature	2,500
<i>Antikythera mechanism</i>	2,200
<i>Astrolabe</i>	2,100
Abstract topics such as zero and negative numbers	2,000
Hourglasses	1,500
Complex topics such as compound interest	1,400
Measuring probabilities for games of chance	1,000
Accounting	900
Graphs	800
<i>Slide rules</i>	575
Measuring rates of change and acceleration	500
<i>Mechanical calculators for addition and subtraction</i>	425
Measuring power	400
<i>Calculating trajectories</i>	400
<i>Mechanical calculators for multiplication and division</i>	375

(Continued)

**Table 1.1** (Continued)

Mathematics, Calculating Devices, and Software	Approximate Number of Years Prior to 2013
Measuring invisible phenomena such as sound	350
Abstract topics such as irrational numbers and uncertainty	350
<i>Punch-card calculating devices</i>	350
Calculus	350
Counting short passages of time (<1 second)	300
Large-scale statistical studies with millions of samples	250
Very complex topics such as infinity and uncertainty	250
Mathematical weather prediction	250
Measuring electrical and magnetic phenomena	200
<i>Mechanical tabulating machines</i>	200
<i>Boolean algebra</i>	175
<i>Set theory</i>	150
<i>Fuzzy sets</i>	145
Relativity	105
Measuring the strong and weak forces and gravity	100
<i>Digital computers</i>	70
<i>Operations research</i>	65
<i>Programming languages</i>	65
<i>Sorting algorithms</i>	55
<i>Databases</i>	55
<i>Pocket calculators</i>	50
<i>Mathematical software applications</i>	50
<i>Scientific software applications</i>	50
<i>Financial software applications</i>	45
<i>Statistical software applications</i>	40
<i>Accounting software applications</i>	40
<i>Architectural and engineering applications</i>	40
<i>Graphics rendering engines for games</i>	35

Table 1.1 illustrates the fact that the human use of mathematics is ancient and can be traced almost as far back as speech. The reason for this is that mathematical knowledge became a critical factor when human beings started to live in villages and trade with others.

Those who hunt and gather wild plants have little need for math and only rudimentary needs for sophisticated communications of any kind. But the advent of agriculture, living in communities, and trade with other communities brought the needs for weights, measures, awareness of seasonal changes, and at least basic arithmetic such as addition and subtraction.

**Table 1.2** *Evolution of Recording Methods and Media*

Recording Methods and Media	Approximate Number of Years Prior to 2013
On stone or bones	50,000
On clay	6,000
With pictographs such as hieroglyphics	4,500
On papyrus	4,000
With ideographs such as Chinese characters	4,000
Using encryption	2,500
With alphabetic information	2,500
On vellum	2,000
On paper	2,000
In full color	700
Graphically	400
<i>On punched cards</i>	350
Using tactile symbols such as Braille	250
<i>On paper tape</i>	250
<i>Using cameras and film</i>	160
<i>Recording sounds</i>	130
<i>Magnetically on tape</i>	125
On vinyl	125
Dynamically in full motion	100
On microfilm	80

(Continued)

**Table 1.2** (Continued)

Recording Methods and Media	Approximate Number of Years Prior to 2013
<i>Recording in three dimensions</i>	75
<i>Magnetically on disks</i>	55
<i>Optically on disks</i>	50
<i>On solid-state devices</i>	35
<i>Using multimedia</i>	30
<i>Using digital cameras</i>	25
<i>On e-books</i>	25
<i>On smartphones</i>	10
<i>Using quantum bits</i>	5
<i>Heterogeneous databases (big data)</i>	5

## Recording Information

Once calculations have been performed, there is also a need to keep the information in a permanent or at least long-lasting format so that the information can be shared with others or used later on as needed. Table 1.2 considers all of the various methods used from ancient times through the modern era for recording information in a permanent form.

As can be seen from Table 1.2, the recording of information is an ancient activity that dates back about as far as the invention of writing and numerals. Without a method of recording the information, calculations or ownership of articles could not be shared with others or used later to verify transactions.

A modern problem that will be discussed in later chapters is the fact that storage methods are not permanent and there is uncertainty about how long either paper records or computerized records might last.

Paper is flammable and also affected by insects, moisture, and other forms of destruction. Magnetic memory is long lasting but not permanent. What's worse is that any kind of stray magnetic field can damage or destroy magnetic records.

Optical records stored on plastic disks might last 100 years or more, but the plastic itself has an unknown life expectancy and the recording surfaces are easily damaged by abrasion, soot, fire, or mechanical stress.

The bottom line is that the earliest known forms of records, such as carvings on stone or clay, probably have the longest life expectancies of any form of recording yet invented.

## Communicating Information

Table 1.3 lists the inventions for how information can be transmitted or shared with other human beings once calculations have been performed and the results stored in some fashion. It is obvious that almost all information will be needed by more than one person, so communication and information sharing are almost as old as mathematics.

**Table 1.3** *Evolution of Communication Channels*

Communication Channels	Approximate Number of Years Prior to 2013
Word of mouth	50,000
Couriers	6,000
Flashing lights	5,000
Smoke signals	5,000
Music notation	4,500
Carrier pigeons	3,500
Codes and ciphers	2,500
Handwritten books	2,500
Mirrors or polished surfaces	2,000
Sign languages	1,750
Knotted strings	1,500
Printed books	1,000
Graphs for mathematical values	800
Newspapers	350
Magazines	300
Signal length (Morse code)	175
Touch for the blind (Braille)	175
Telegraph	175
Radio	150
Telephone	130
Television	70
<i>Satellite</i>	60
<i>Subliminal signals</i>	50

(Continued)

**Table 1.3** (Continued)

Communication Channels	Approximate Number of Years Prior to 2013
<i>Sleep learning</i>	40
<i>Ultra-low frequency sounds</i>	35
<i>Internet</i>	30
<i>Devices for the deaf (cochlear implants)</i>	30
<i>Electronic books (e-books)</i>	25
<i>Lasers</i>	25
<i>Automatic language translation</i>	25
<i>Intranet</i>	20
<i>Avatars in simulated worlds</i>	15
<i>Blogs</i>	12
<i>Webinars or podcasts</i>	12
<i>Wiki sites</i>	10
<i>Social networks</i>	10
<i>Animated multisensory methods</i>	5

Over the centuries, the human species has developed scores of interesting and useful methods for conveying information. Often, there is a need to transmit information over very long distances. Until recently, carrier pigeons were used for messages between distant locations.

However, military organizations have long recognized that visible hilltops or other high places could be used to send information over long distances by means of either polished surfaces during the day or fires at night. Recall the famous line from Paul Revere's ride that describes lighting lanterns in the North church tower to warn of the approach of British troops: ". . . one if by land, two if by sea."

Communication with undersea submarines was difficult until the advent of communication by ultra-low frequency sounds.

Codes and secret communications also have a long history of several thousand years. Later chapters of this book will deal with several forms of codes and secret communications during World War II, including the famous Native American "code talkers" who spoke in a code based on Navajo, Choctaw, and other Native American languages.

Awareness of the need to communicate is ancient knowledge. There is a curious passage in a Buddhist sutra dating from about the third century BC, in which the Buddha discussed how his teachings might be transmitted. He mentions casually that, on earth, teachings are transmitted with words, but on other worlds, teachings are transmitted by lights, by scents, or by other nonverbal means.

## Storing Information

Table 1.4 lists how information has been stored and accessed. As all scholars and researchers know, once the volume of information exceeds a few books or a few dozen written documents, there is an urgent need for some kind of taxonomy or catalog scheme to ensure that information can be found again when it is needed.

Information storage and access are critical features of modern computers, and modern software has played a huge part in improving information retrieval.

**Table 1.4** *Evolution of Information Storage and Access*

Information Storage and Access	Approximate Number of Years Prior to 2013
Personal collections of written information	6,000
Libraries or public collections of written information	4,500
Topical collections of laws and legal codes	2,000
Topical collections such as medical and law libraries	1,200
University curricula for information by topic	1,000
Taxonomy for biological and scientific organization	300
Dewey decimal system for book organization	135
<i>Sequential databases of information</i>	65
<i>Random databases of information</i>	55
<i>Relational databases of information</i>	50
Affinity recommendations based on past preferences	35
<i>Web search engines for selection of keyword information</i>	25
<i>Intelligent agents for selection of relevant information</i>	15
<i>Big data analytical tools</i>	10



Table 1.4 shows topics that have been difficult for large volumes of information for thousands of years and that in fact are becoming worse in the modern world. For most of human history, information collections seldom topped more than 10,000 volumes, even for large libraries. In today's world of almost instantaneous recording of all books, magazines, research papers, images, and other forms of intellectual content, there are now billions of documents. Every week that passes, more and more information is published, recorded, and added to cloud libraries and other forms of computer storage. There is no end in sight.

There is an urgent need for continuing study of the best ways of recording information for long-term survival and for developing better methods of sorting through billions of records and finding and then aggregating topics relevant to specific needs. The emerging topic of "big data" is beginning to address these issues, but the solution is not currently visible and is still over the horizon.

The first and most long-lasting method of storing and accessing data was by means of libraries. Throughout civilized history, many famous libraries have served scholars and researchers. The library of Alexandria, the library of the University of Nalanda, the library of Perganum, the five libraries of Ugarit, the Roman libraries of Trajan in the Forum, and the library of Constantinople were all famous throughout antiquity.

Modern libraries such as the Library of Congress, the Harvard Library, and in fact many large college libraries still serve as major repositories of information for students and researchers.

Books have been used for thousands of years to record and convey knowledge from human to human, especially from teachers to students. Personal libraries of reference books are the normal accoutrements of all professions, including engineering, law, medicine and, of course, software engineering.

More recently, e-books, web search engines, and intelligent agents are making it possible for individuals and scholars to access more data and information at greater orders of magnitude than was possible at any time in human history up until about 25 years ago.

## Enabling Computers and Software

Table 1.5 departs somewhat from the direct line of descent between inventions and computers and software. This table deals with some of the *enabling inventions* that later became important when computers and software also became important.

**Table 1.5** *Enabling Inventions for Computers and Software*

Enabling Inventions	Approximate Number of Years Prior to 2013
The modern patent systems	800
<i>Boolean algebra</i>	175
<i>Plastics for computer cases, screens, connections, etc.</i>	125
<i>Vacuum tubes</i>	120
<i>Punched cards</i>	120
<i>CRT tubes</i>	80
<i>Von Neumann architecture</i>	75
<i>Paper tape</i>	75
<i>Integrated circuits</i>	70
<i>Transistors</i>	70
<i>Magnetic tape</i>	70
<i>High-level programming languages</i>	65
<i>Magnetic disks</i>	60
<i>Operating systems</i>	55
<i>Magnetic ink for bank checks</i>	55
<i>Magnetic stripes for credit cards</i>	50
<i>Graphics display adapters</i>	40
<i>Laser printers</i>	40
<i>Floppy disks</i>	40
<i>Dot matrix printers</i>	35
<i>Ethernet</i>	35
<i>LED displays</i>	30
<i>Ink-jet printers</i>	25
<i>Solid-state memory</i>	20
<i>Flash disks</i>	15

One of the first enabling inventions is that of the patent system itself. The first known patent in English was granted in 1331 in England to a man named John Kemp. Later, an Italian patent was granted in Florence in 1421. Patents similar to modern patents and enforced by statute appeared in Venice in a law establishing patents in 1474.

The first patent issued in North America was issued by the Massachusetts General Court in 1641 to a man named Samuel Winslow for a method of making salt. The first federal patent law in the United States was passed on April 10, 1790, and had the title of “An Act to Promote the Progress of Useful Arts.”

---

**Note**

The name “patent” is derived from the phrases “letters patent” and “letters close.” The seal on letters close covered the fold and had to be broken in order to read the letter. The seal on letters patent was attached to the bottom of the document so it could be read with the seal intact.

---

Software patents have had a very convoluted path and were sometimes barred and more recently accepted. But there is no guarantee that software patents will always be accepted by the U.S. Patent Office. In the 1960s, software patents were barred and several lawsuits were filed, with the courts generally concurring that software was not patentable.

In 1981, the U.S. Supreme Court became involved in the case of *Diamond vs. Diehr* and decided that, at least in special cases, software was patentable. This forced a change of procedure in the Patent Office. But the situation remained murky and ambiguous and largely decided on by a case-by-case basis without any real guides or fixed rules.

In 1998 in the famous case of *State Street Bank vs. Signature Financial Group*, it was finally decided what forms of software could be patented. This case involved the hub-and-spoke method of processing mutual funds. The Supreme Court decided that business processes, including those embodied in software, were patentable.

A number of other precursor inventions were also important. For example, without transistors and integrated circuits, there would not be any portable computers, embedded computers, or any types of small electronic devices that today all use software.

The inventions that became integral parts of computers include plastic for cases and screens, integrated circuits, transistors, graphics boards, and LED displays.

Other inventions had a strong impact on the use of computers and hence on the software that was created to support those uses. For example, without the 1960 IBM patent on a magnetic stripe that could be applied to plastic, credit cards would not have been developed. Without the invention of magnetic ink, bank checks would still be sorted alphabetically instead of in numeric order and probably sorted by hand.

---

## Key Inventions Relevant to Software

The inventions listed in the previous tables are all important in one way or another. However, in thinking about the inventions that had the greatest impact on software, the inventions discussed in the following section are the most critical.

### Alphabetic Languages

Information recorded using pictograms such as Egyptian hieroglyphics is elegant and beautiful and has produced some wonderful calligraphy, but such systems do not lend themselves to rapid data entry and computerization. The same is true of information recorded using ideograms such as Chinese and Japanese kanji (which uses Chinese symbols). There are thousands of symbols, which makes typing extremely difficult.

During World War II, the text entered into the Japanese “Purple” coding machine actually used two American Underwood typewriters and plain text using English characters. Alphabetic languages have the greatest speed for typed entry.

### Binary and Decimal Numbers and Zero

Computers and software can process numbers using any base such as binary, octal, decimal, or hexadecimal. However, electronic circuits for performing mathematics are somewhat easier to design using binary arithmetic. Octal or base 8 numbering systems are easily convertible from binary. (Some Native American tribes used octal numbers since they counted by using the gaps between the fingers rather than the fingers themselves.) Several computers were based on octal numbers such as the DEC PDP line.

Hexadecimal or base 16 numbers are also used in computers and are convenient because they match byte capacities. However, the bulk of day-to-day calculations used by humans are based on decimal or base 10 numbers. Decimal numbers are somewhat analogous to the QWERTY keyboard: not optimal but so widely used that switching to something else would be too expensive to consider.

The decimal point seemed to have originated in India during the ninth century, but it was John Napier who made the concept important in Western mathematics around 1620. Napier also invented logarithms and an interesting manual

calculator called “Napier’s bones.” Logarithms were used in the first slide rules and hence are an important background topic for analog computation.

The concept of zero seemed to have several independent sources. It was used in Babylon with base 60 math, but apparently as a placeholder rather than actual calculations. This use was about 2,500 years ago.

The Olmecs and Mayans both used zero as a true number, and it was used for calendar calculations, which were quite complex. This use of zero seems to date to around 400 AD.

The use of zero in India dates to about 458 AD when it was found in a text on mathematics. Whether this was an indigenous invention or inherited from Babylon is not certain. Later in the 600s, the famous Indian mathematician Brahmagupta wrote a paper on the uses of zero, which moved past zero itself into negative numbers.

Decimal numbers, the decimal point, and zero were all important precursors leading to computers and software calculations.

## Digital Computers

Later chapters in this book will discuss the evolution of digital computers and associated software from the mid-1930s through 2010, with projections to 2019. Suffice it to say that software was created specifically to operate on digital computers. Without digital computers, there would be no software. Without software, digital computers would have no major purpose and would probably not have supplanted analog computers.

## Higher-Level Programming Languages

I started as a young programmer in the 1960s. Programming using both machine language (mainly for patches and bug repairs) and basic assembly language was how I first programmed IBM 1401 computers.

My firsthand experience was that machine language was very error prone and also rapidly fatiguing due to the high attention span needed to deal with it. Assembly language was a step in the right direction, but not a very big step. Having to use dozens of assembly instructions to handle calculations or format printed output was time consuming and actually boring. Higher-level languages, starting with ALGOL, COBOL, FORTRAN, PL/I, APL, and others, reduced coding time, significantly reduced coding errors, and converted programming into a viable occupation.

## Random-Access Storage

Sequential storage of data on paper tape, card decks, or magnetic tape had a fairly long and useful life. But it was very inefficient and required far too much movement of tapes to achieve high speeds. The invention of disk drives and random-access storage allowed faster processing, sophisticated search algorithms, and a path that eventually would lead to today's "big data" world with billions of records and millions of files being accessed for specific problems.

Without random access, modern computing and software could handle only a small fraction of important data analysis problems. Random access would also lead to the relational database concept, sorts, and a variety of powerful query languages in the Structured Query Language (SQL) family.

---

## The Impact of Software on People and Society

The time frame in which computers and software have developed has barely been more than 75 years. Yet their impact on individual humans and on societies has been as important as the printing press, airplanes, television, and automobiles.

### Beneficial Tools and Applications

The following is a summary of tools and applications that have transformed the way businesses operate; wars are fought; and individuals gather information, communicate, and use their leisure time. It is surprising that these have all originated within the past 50 years. Probably half of these tools and applications are less than 25 years old.

- Business tools
  - Accounting
  - Actuarial studies
  - Advertising via the web
  - Agricultural planning
  - Analytics
  - Bar-code scanners

- Big data
- Budget analysis
- Cloud computing
- Competitive analysis
- Cost and resource tracking
- Cost estimating
- Crowdsourcing
- Customer relationship management (CRM)
- Customer satisfaction analysis
- Customer support
- Distribution optimization analysis
- Electric power grid controls
- Enterprise resource planning (ERP) packages
- Finance
- Governance
- Human resource management
- Inventory
- Investments
- Just-in-time inventory controls
- Legal support
- Marketing
- Oil exploration
- Order entry
- Order tracking
- Planning and scheduling
- Process controls
- Reservation systems

- Risk estimation and analysis
- Robotic manufacturing
- Sales support
- Supply chain management
- Surveys and opinion analysis
- Telephone network controls
- Water purification
- Web retailing
- Databases
  - Graphics and images
  - Music
  - Signals and analog
  - Text and numeric
- Data warehouses
  - Mixed-data forms
- Education tools
  - Comparative education statistics
  - Curriculum planning
  - Customized e-learning for each student
  - Skills inventory analysis
  - Special tools for the handicapped
  - Student research via the web
  - Virtual classrooms
- Embedded devices
  - Automotive engines and brakes
  - Automotive security systems
  - Avionic



- GPS navigation
- Hearing aids
- Manufacturing
- Medical
- Signal processing
- Smart appliances
- Telecommunications
- Government tools
  - Air traffic control
  - Background verification
  - Budget analysis
  - Census
  - Court records
  - Disaster preparedness
  - Economic analysis
  - Employment statistics
  - Environmental monitoring
  - Financial controls
  - Health and longevity statistics
  - Highway siting, design, and construction
  - Identity verification
  - Land management
  - Law enforcement
  - Legislative records
  - Mandates and regulations
  - National defense
  - Patent analysis

- Political records
- Pollution monitoring
- Prisons
- Property assessments
- Redistricting
- Regulatory agencies
- Risk analysis
- Taxation
- Traffic analysis and controls
- Unemployment support
- Voter records
- Water supply controls
- Welfare
- Zoning
- Leisure
  - Blu-ray and digital video
  - Computer games
  - Digital music formats
  - Geocaching
  - Music playlists
  - Online magazines
  - Streaming video
  - Virtual reality worlds
- Medical
  - Coordination in real time among medical teams
  - External devices
  - Implanted devices

- Insurance record keeping
- Lab tests
- Patient hospital monitoring
- Patient records
- Robotic surgical devices
- Statistics: national, global
- National defense
  - Antimissile shields
  - Combat simulation
  - Command and control
  - Cybersecurity
  - Deep ocean monitoring
  - Early threat warnings
  - Encryption and decryption
  - Intelligence gathering and coordination
  - Logistics analysis
  - National Security Agency signal interception
  - Satellite monitoring
  - Secure communications
  - Threat analysis
- Personal tools
  - Blogs
  - Computers
  - Contact lists
  - Daily news feeds
  - Digital appliances
  - Digital cameras

- Digital image processing
- Digital watches
- E-books
- Email
- Graphics
- handheld full-function digital calculators
- Handicap support for the deaf, blind, etc.
- Home finances
- Instant computer chat
- Music
- Natural language translation
- Presentations
- Scheduling
- Search engines
- Smartphones
- Social networks
- Spreadsheets
- Statistics
- Tablet computers
- Text to speech
- Video processing
- Web browsers
- Word processing
- Professional tools
  - Accounting
  - Analytics
  - Animation and graphic arts

- Architecture
- Civil engineering
- Computer animation
- Data mining
- Drafting
- Economic analysis
- E-learning
- Encryption and decryption
- Engineering
- Intelligent agents for web scanning
- Law enforcement
- Legal support
- Math
- Medical support
- Music composition
- Music recording, playback, and mixing
- National security
- Patent analysis
- Pharmaceutical
- Project management
- Property management
- Publishing
- Real estate listings
- Spell checkers and grammar checkers
- Statistics
- Programming tools
  - Application sizing

- Automatic testing
- Complexity analysis
- Configuration controls
- Continuous integration
- Cost and schedule estimation
- Data mining of legacy applications
- Debugging
- Inspection support
- Maintenance and support estimation
- Measurements and benchmarks
- Programming language compilers
- Quality estimation
- Requirements and design analysis
- Requirements modeling
- Reusability analysis
- Risk estimation
- Static analysis
- Test tools (design and execution)
- Virtualization
- Website design and construction
- Protective tools
  - Antispam
  - Antispyware
  - Antivirus
  - Smart alarm systems
- Scientific tools
  - Archaeological analysis

- Astronomical analysis
- Biological analysis
- Chemical analysis
- Computer-enhanced image calibrations
- Computer-stabilized optical devices
- Deep ocean exploration
- DNA analysis
- Epidemiology analysis
- Forensic analysis
- Geological exploration (side-scan radar)
- Linguistic analysis
- Metallurgy
- Meteorology analysis and weather predictions
- Nanotechnologies
- Nuclear device controls
- Physics research equipment
- Self-aiming telescopes for the deaf, blind, etc.
- Simulations of physical phenomena
- Space vehicles, rovers, and satellites
- Visualization

As can be seen from this list, computers and software are making profound changes to every aspect of human life: education, work, warfare, entertainment, medicine, law, and everything else.

### **Harmful Inventions**

Computers and software have also introduced a number of harmful inventions that are listed below, some of which did not exist before. Among the harmful

inventions caused by computers and software are identity theft, hacking, and computer viruses. These are new and alarming criminal activities.

- Browser hijackers
- Computer botnets
- Computer keyboard tracking
- Computer spam
- Computer spyware
- Computer viruses
- Computer worms
- Computerized customer support
- Difficulty in correcting errors in computerized data
- Electronic voting machines without backup
- Hacking tools
- Identity theft
- Phishing
- Piracy
- Robotic telephone calls (robo-calls)
- Robotic weapons systems
- Smart weapons: bombs, drones, and missiles
- Spam
- Special viruses attacking industrial equipment
- Spyware
- Stock market software without anomaly shutoffs
- Unintelligible telephone voice menus
- Web pornography



These threats are comparatively new and all are increasingly hazardous in the modern world. Indeed, identify theft has become one of the largest and most pervasive crimes in human history. It is also an example of a new kind of crime where the criminal and the victim never see each other and can be separated by more than 12,000 miles when the crime takes place.

These harmful aspects of computers and software have triggered new laws and new subindustries that provide virus protection, hacking insurance, and other forms of protection.

These inventions have also led to the creation of new and special cybercrime units in all major police forces, the FBI, the CIA, the Secret Service, the Department of Defense and the uniformed services, Homeland Security, and other government organizations. The emergence of the Congressional Cyber Security Caucus is a sign that these new kinds of cybercrimes are attracting attention at the highest levels of government.

## **Weighing the Risks**

Computers and software are making profound changes to every aspect of human existence. Many readers have thousands of “friends” on social networks. Even more readers follow the daily lives and activities of countless celebrities and personal friends by using “tweets” or short messages. Text messages are beginning to outnumber live telephone calls (and also cost more due to new computerized billing algorithms).

Purchases of electronic e-books recently topped purchases of ordinary paper books. Banks now charge extra fees to provide paper bank statements as opposed to online electronic statements. All of our medical and education records are now computerized and stored in databases.

It would not be possible to book an airline flight or a hotel without computers and software. Indeed, after large snowstorms or hurricanes when power lines are down, many kinds of businesses cease operations because they are no longer equipped to handle manual transactions. Computerized games, including massively interactive games with thousands of simultaneous players, are now the preferred form of entertainment for millions of young people. Modern films use special effects with lifelike realism that are generated by computers. It is even possible to create new roles for actors and actresses who are no longer living by means of computers and software.

The impact of computers and software has been a mixture of good and bad. Certainly, the ability to send emails and text messages and to find information on the web are very useful additions to our daily lives. We use GPS maps on our smartphones almost every time we travel, particularly when we travel to new and unfamiliar locations.

The ability of physicians to communicate instantly with colleagues helps medical practice. Computerized medical diagnostic machines such as CAT scans and MRI equipment are also beneficial. Cochlear implants have restored hearing to thousands of profoundly deaf patients. Robotic manufacturing is cheaper and sometimes more precise than the manual construction of many complex devices.

But the ever-increasing odds of identity theft and the constant need to keep our computers and electronic devices safe from hackers and data theft are a source of continuing worry and also a source of considerable expense.

In evaluating the advantages and disadvantages of computers and software, the weight of available evidence is that software and computers have provided more benefits to the human condition than they have caused harm. Of course, those who have been harmed probably disagree.

But statistically looking at all known uses of computer and software in the modern world, there have been significant benefits in the way we can communicate, transact business, and carry out scientific and engineering work. It is doubtful that any scientist or engineer would want to stop using computers and software. The same is true of many other kinds of work such as health care, law enforcement, accounting, and even real estate.

---

## Summary

This prelude showed the evolution and convergence of many fields that would come together to create modern computers and software. Mathematics, data storage and retrieval, communication methods, and software itself would come together to create the modern era of personal software and personal computing.

Later chapters in this book discuss the evolution of software engineering from the earliest dreams of visionaries in the 1930s through the growth of the largest and wealthiest companies in human history by the end of the 20th century.

*This page intentionally left blank*

*This page intentionally left blank*

# Index

- 3D printing, 288
- “15 Worst Data Security Breaches of the 21st Century” (Amerding), 358
- 901 computer, 84
- 1930 to 1939, 37
  - innovators, 37–40
  - small mathematical applications, 40–42
- 1940 to 1949, 43
  - global conflicts, 43–44
  - historical contributions, 73–75
  - postwar era, 68–72
  - software applications, 74–75
  - software building, 75–76
  - World War II. *See* World War II
- 1950 to 1959
  - cultural perceptions of computers, 80–81
  - first commercial computers, 85–89
  - function points, 90–91
  - innovators, 81–84
  - military and defense computers, 77–80
  - overview, 77
  - programming languages, 84
  - software applications, 89–90
- 1960 to 1969, 93
  - automation and new professions, 110–111
  - businesses, 115–118
  - computer and software growth, 120–122
  - credit cards, 109–110
  - early specialized outsourcing, 95–96
  - evolving workforce, 93–95
  - function points, 122–123
  - IBM System/360 computer, 104–106
  - litigation, 118–119
  - PDP-1 computer, 111–112
  - programmers, 96–104
  - programming languages, 112–115
  - space, 119–120
  - Turing award, 106–108
- 1970 to 1979
  - background enabling inventions, 159–162
  - companies formed, 129–145
  - function points in 1975, 162–163
  - Honeywell and Sperry-Rand lawsuit, 158–159
  - impact of companies, 145
  - overview, 125
  - software applications, 146–147
  - software categories, 147–158
  - software evolution, 125–129
- 1980 to 1989
  - companies formed, 168–204
  - function points, 204
  - overview, 165
  - rapid changes, 166–168
  - software growth, 204–205
- 1990 to 1999
  - companies formed, 212–237
  - function points, 244–245
  - innovations, 210–212
  - international date formats, 239–242
  - mass updates and legacy software, 237–239
  - outsourcing, 243
  - overview, 207–208
  - software growth, 243–244
  - World Wide Web emergence, 208–210
- 2000 presidential election, 368–370
- 2000 to 2009
  - companies formed, 257–283
  - dot-com bubble, 248–250
  - function points, 284–285
  - Great Recession, 250–254
  - innovations, 254–257
  - overview, 247–248
  - software growth, 283–284

- 2001: *A Space Odyssey* (Clarke), 81  
 2010 to 2019  
     companies and products predictions, 311–327  
     companies formed, 296–311  
     function points, 328–329  
     overview, 287–288  
     predictions, 288  
     software engineering improvements, 293–296  
     software engineering professional status, 289–293  
     software growth, 327–328  
 “Abbreviated Computer Instructions” report (Turing), 55  
 ABC computer, 40, 64–65  
 Abnow browser hijacker, 366  
 AbsolutData Research company, 260–261  
 Accenture company, 171–172  
 Access  
     controls, 399–400  
     evolution of, 17–18  
 Accord automobiles safety recall, 348  
 Accounting practices, 133  
 “Act to Promote the Progress of Useful Arts”, 20  
 Activity metric in social networks, 309  
 Ada programming language, 190  
 Adams, Charley, 72  
*Adaptive Business Intelligence* (Michalewicz, Michalewicz, Schmidt, and Chiriatic), 277  
 Addition in prehistoric knowledge, 4  
 Addresses  
     harvesting, 376–378  
     munging, 377  
 Adelman, Leonard, 390  
 Adhar, Azeem, 309  
 Administrators, 399  
 Adobe company, 172–173, 320  
 Adolphus, Gustavus, 413  
 Adult videos, 218  
 Advanced and experimental software, 147  
 Advanced Research Projects Agency (DARPA), 121  
 Agile Manifesto, 234, 255  
 Agile system, 190–192, 225, 256  
 Agnew, Spiro, 127  
 Aiken, Howard H., 40, 66–67  
 Air-traffic control systems  
     need for, 88–89  
     shutdown, 341–342  
 Airbus A380 wiring problem, 344–345  
 Airport luggage handling failure, 337–338  
 Akamai company, 214  
 Albrecht, Allan  
     backfiring, 226  
     function point metrics, 106, 161–162, 419  
     at SPR, 196–197  
 Alcatel company, 166  
 Aldrin, Buzz, 119  
 Alexander, Keith, 405  
 ALGOL language, 84  
 Algorithm, concept of, 39  
 Allen, Paul  
     entrepreneur of year, 180  
     Microsoft founding, 131, 138–139  
     wealth of, 145  
 Alphabetic languages, 21  
 Altair Basic language, 131  
 Altair computers, 131–132  
 ALTOPS architecture, 399  
 Amazon company, 214–216  
 Amdahl, Gene, 104, 106  
 Amdahl computers, 106  
 Amerding, Taylor, 358  
 America Invents Act (AIA), 269  
 America Online (AOL)  
     accidentally exposed records, 360  
     Netscape acquisition, 229  
     overview, 173–174  
     phishing on, 382–383  
 American Express credit cards, 109  
 American Institute of Electronic Engineers (AIEE), 68  
 American Medical Association (AMA), 414

- American National Standards Institute (ANSI) standard date format, 239
- American Programmer* magazine, 145
- American Standard for Information Interchange (ASCII), 95
- Amster, Geoffrey, 276
- AN/FSQ-7 Combat Direction Central computer, 79
- Analog computers
  - 1960s, 117
  - espionage, 48
  - gun control, 42
  - World War II, 46–47
- Ancient times to modern era, 1
  - beneficial tools, 23–32
  - harmful inventions, 32–34
  - human need to compute, 1–3
  - mathematics inventions, 8–20
  - numerical knowledge, 3–8
  - risks, 34–35
  - software inventions, 21–23
- Andersen Consulting company, 171
- Andresen, Marc, 228, 261
- Andresen Horowitz company, 261
- Android operating system, 175
- Andrus, Cashman, 310
- AngelPad company, 298–299
- Angles in prehistoric knowledge, 4
- Animation, 151
- Anonymous group, 356, 361, 373
- Anti-Phishing Working Group, 383
- Antiaircraft guns, 57
- Antitrust lawsuits
  - IBM, 136
  - Microsoft, 210, 249, 255
  - Sperry-Rand and Honeywell, 118–119
- Antivirus programs
  - bug, 345
  - development of, 199, 354
  - effectiveness of, 396–397
  - free, 174
- Apache company, 216
- Apache Software Foundation, 216
- APL programming language, 114
- Apollo Guidance Computer (AGC), 119
- Apollo spacecrafts, 119–120
- Apple Computer
  - Adobe dispute, 173
  - founding, 131
  - NeXT lawsuit, 187
  - Objective-C for, 166
  - overview, 132
  - wealth from, 145
- Appliances, smart, 309
- Application life cycle management (ALM), 176
- Applied Software Measurement* (Jones), 411
- Appraisals, real estate, 281–283
- Archon Information Systems, 262
- Argots, 209
- Ariane 5 rocket explosion, 338–339
- “Arms and Insecurity” (Richardson), 323
- Armstrong, Neil, 119
- ARPANET, 121
- Art of Software Testing* (Myers), 146
- Artemis company, 197
- Arthur Andersen company, 172
- Artificial intelligence (AI), 397–398
- Artzt, Russell, 132
- ASCII coding system, 109
- Assembly language, 22, 84
- Assessment and Control of Software Risks* (Jones), 410
- Association of Computing Machinery (ACM), 68
- Astronomical time, 241–242
- AT&T
  - Datanet modem, 109
  - telephone lines shutdown, 336
- Atanasoff, John Vincent, 38, 40, 64–65, 158
- Atomic bombs
  - EMP, 375–376
  - Soviet Union, 69
- Auction Bid company, 218
- Audience metric for social networks, 309
- Auerbach, Tjark, 174
- Australia in World War II, 52–53
- Australian Computer Society, 68

- Authentication controls, 400
- Authority metric in social networks, 309
- Authr company, 299
- Autoflow software, 114
- Automated function points, 226
- Automated Project Office (APO) tool, 178
- Automated teller machines (ATMs)
  - development of, 95
  - skimming, 384
- Automatic abacus machine, 50
- “Automatic Computing Engine” (ACE)
  - (Turing), 55, 65
- Automatic Sequence Controlled Calculator (ASCC), 40, 66
- Automation in 1960s, 110–111
- Automotive safety recalls, 348–349
- Avira antivirus company, 174
- Axon Corporation, 235
  
- Baan, Jan, 132
- Baan, Paul, 132
- Baan Corporation, 132
- Babbage, Charles, 40
- Babbage analytical engine, 9, 40
- Babylonia, prehistoric knowledge in, 6
- Backfiring, 122, 226
- Background enabling inventions,
  - 159–162
- Backus, John, 67
- Baez, Joan, 411
- Bagle botnet, 364
- Baidu company, 222
- Baker, Mitchel, 273
- Bales, Steve, 120
- Ballister, Andrew, 304
- Ballmer, Steve
  - spam report, 387
  - wealth of, 145
- Ballots, 367–371
- Bank of America credit cards, 109
- BankAmerica Services Corporation, 109
- Banks
  - automation in, 94
  - credit cards, 109–110
  - hacking targets, 371
  - mortgage foreclosures, 253
  - paper statement fees, 34
  - predictive analysis, 324
  - QIF format, 184
  - security offices, 396
  - vertical markets, 126, 141
- Bar-El, Hagai, 386
- Bardeen, William, 81–82
- Bargava, Rajat, 310
- Barker, John, 276
- Barrios, Brian P., 262
- Barron’s Magazine* dot-com article, 249
- Barton, Rich, 281
- Bartoo, Jim, 267
- BASIC programming language
  - development of, 114–115
  - Microsoft, 138–139
- Bauer, F. L., 103
- Bawdsey Research Station, 58
- Baxter, William, 119
- Bazaar service, 263
- Beard, Malcom, 53
- Beck, Kent, 255
- Beckman hybrid computer, 117
- Beedle, Mike, 255
- Bell, Alexander Graham, 182
- Bell, Gordon, 415
- Bell, Gwen, 415
- Bell Labs firewalls, 355
- Bellows, Matthew, 310
- Ben-Horin, Daniel, 200
- Benchmark data, 225–227
- Beneficial tools and applications, 23–32
- Berners-Lee, Tim
  - NeXT computer used by, 188
  - World Wide Web, 208
  - “World Wide Web” term, 166
- Bernoulli Box, 168
- Berry, Clifford, 38, 40, 64
- Bezos, Jeff
  - Amazon, 214–215
  - wealth of, 145
- Bibliography and references, 407
  - book sources, 407–415
  - web sources, 415–421



- Big data, 149, 312
- Bill and Melinda Gates Foundation, 219
- Binary Automatic Computer (BINAC), 72
- Binary numbers, 21–22
- Binary synchronous communication, 95
- Biometric identification, 385, 400
- Bitcoin method, 302–303
- BitDefender, 355
- Black Hat conference, 353
- Black hat hackers, 353
- Black Point Park, 63
- Black Point* ship, 63
- BlackBerry company, 174–175
- Bletchley Park, 54
- Blind people
  - future products, 322
  - virtual education, 316, 320
  - wearable computers for, 327
- Blogs, 296
- Blue boxes, 352
- Blue hat hackers, 353
- “Blue ox” bombsight, 59
- Blueprints tool, 263
- Bluetooth hijacking, 363–364
- Boat building and navigating, prehistoric
  - knowledge for, 7
- Boehm, Barry, 106, 407
- Bogardus, Claude, 119
- BOMARC supersonic guided missile, 80
- Bombe machine, 54
- Bombsights
  - Lotfernrohr 7, 48
  - Norden, 59–61
- Bonsignour, Olivier, 411
- Booch, Grady, 191
- Book sources, 407–415
- Books for storage, 18
- Boole, George, 39
- Borland company, 175–177
- Bosack, Leonard, 178
- Boston Computer Museum, 415–416
- Boston Consulting Group (BCG), 96
- Bot herders, 364
- Bot masters, 364
- Botnets, 364–365
- Bottlenecks in von Neumann architecture, 66
- Bougnion, Edouard, 236
- “Bouncing ball” game, 72
- Bownman, Eric, 299
- Brahe, Tycho, 324
- Brahmagupta (mathematician), 22
- Brain size, 4
- Brattan, Walter, 81–82
- Brazil, function points in, 255–256
- Bricklin, Dan, 161, 421
- Brin, Sergey
  - Google development, 210, 222
  - wealth of, 145
- Brindley, Doug, 197
- Briskman, Robert, 232
- Brisman, Julissa, 217
- British Computer Society, 68
- British passport system failure, 339–340
- “Bronze goddess”, 54
- Brook, Isaac S., 53, 82
- Brooks, Fred
  - IBM lab, 121
  - influence of, 145
  - on JCL, 114
  - Mythical Man-Month*, 104, 146, 408
  - Turing award, 106
- Broward County, Florida, electronic
  - voting problems in, 368
- Browser wars, 210, 229
- Browsers
  - competition, 273–274
  - development of, 208
  - hijackers, 365–366
- Brunner, John, 390
- Bryce, James, 67
- Buchanan, Pat, 369
- Buckmaster, Jim, 217
- Buddha
  - encryption, 402
  - teaching transmissions, 17
- Buick automobiles safety recall, 348
- Built-in cameras on portable computers, 399

- Bull, Fredrik Rosing, 52
- Bull company, 52
- Burroughs company, 88
- Bush, George H. W., 209
- Bush, George W., 369–370
- Bush, Vannevar, 37
- Busicom company, 50
- Business software, 147–148
- Business-to-business (B2B) software, 132
- Business tools, 23–25
- Butterfly ballots, 369
- Button, Beau L., 262
  
- C programming language, 112
- C&E Software company, 198
- CACHE database, 138
- Cadillac automobiles safety recall, 348
- Cagle, Stan, 131
- Calculating devices, evolution of, 10–14
- Caliber product, 176
- Cameras on portable computers, 399
- Camino browser, 273
- Camp Varnum, 63
- Campbell-Kelly, Martin, 408
- Campuses, virtual, 315–321
- CAN-SPAM Act, 387
- Canonical, Ltd., 262–263
- Capability Maturity Model (CMM), 192–193
- CAPTCHA method, 377
- Carmack, John, 188
- Carr, Robert O., 223
- Cartridges in APO, 178
- Cassandra, 325
- CAST Software company, 404
- Catal Huyuk settlement, 4
- Cathedral and the Bazaar* (Raymond), 154
- Cavity magnetrons, 61
- Celebrities on Twitter, 279
- Centre Electronique Horloger (CEH), 160
- Cerullo, John, 138
- CGI Informatique company, 145
- Chamberlain, Neville, 60
  
- Change control improvements, 294
- Channels for distributing information, 9
- Charitable activities, 291
  - Freelanthropy, 265–266
  - GoDaddy, 221
- Chawla, Rajeev, 300
- Chawla, Raman, 300
- Chen, Richard, 298
- Chen, Steve, 280
- Chess programs, 232
- Chief Information Security Officers (CISOs), 393
- Chief Security Officers (CSOs), 393
- Child training in ancient times, 7
- Chino, Kobun, 411
- Chiriak, Constantin, 277
- Cincom company, 119
- Cipher systems, 49
- Cisco Systems, 178–179
- CivicSource brand, 262
- Civilian cyberwarfare targets, 371–372
- Clancy, Tom, 231
- Clark, Jim, 228
- Clarke, Arthur C., 81
- Classrooms, virtual, 315–321
- Clerical work in 1960s, 94
- Climis, Ted, 121
- Cloud applications, 257
- CloudVelocity company, 299–300
- CMM levels, 212
- CMMI, 192–193, 403
- COBOL language, 84
- Cochlear implants
  - development of, 160
  - future products, 322
  - wearable computers, 326
- Cockburn, Alastir, 255
- CODASYL data model, 114, 135
- Codd, Ted
  - DCI seminars, 179
  - IBM lab, 121
  - relational databases, 139, 159
  - SECC, 67
- Code Breakers* (Kahn), 401–402
- Code-breaking programs, 55

- Code Complete: A Practical Guide for Software Construction* (McConnell), 413
- Code talkers, 16
- Cohen, Fred, 353, 390
- Cold War era, 68–69
  - beginning, 69
  - computer development, 69–72
  - military and defense computers, 77–80
- Coleman, Dennis, 198
- Collins, Susan, 356
- Colossus computer, 54–56
- Colt, Samuel, 412
- Commercial computers, first, 85–89
- COMMON association, 87
- Common Cause site, 368
- Communication channels, evolution of, 15–17
- “Communications, Computers, and Networks” (Gore), 209
- Communications software, 148
- Compact disk read-only memory (CD-ROM), 168
- Companies
  - 1970s, 130–145
  - 1980s, 168–204
  - 1990s, 212–239
  - 2000s, 257–283
  - 2010s, 296–311
  - predictions, 311–327
- Compaq company, 143–144
- Compilation, 99
- Computable Document Format (CDF), 202
- Computer Aid, Inc. (CAI), 177–178
- Computer Associates (CA), 132–133
- Computer Emergency Response Team (CERT), 194, 354
- Computer Fraud and Abuse Act, 391
- Computer-game companies, 260
- Computer History Museum, 416
- Computer programmers in 1960s, 96–104
- Computer Science Lab, 416
- Computer security engines, 397–398
- Computer Society of India, 68
- “Computer” term, 44
- “Computer virus” term, 353
- “Computer Virus: Theory and Experiment” (Cohen), 353
- Computer voting fraud, 366–371
- ComputerMentor Project, 200–201
- Computerspielemuseum, 416–417
- Conferences in virtual education, 318
- Conficker botnet, 364–365
- Congressional Cyber Caucus, 110, 355, 372, 417
- Consolidated Edison company, 128
- Constantine, Larry, 145–146
- Continuous operations in SAGE, 78
- Control Data Corporation (CDC), 88
- Control program for microprocessors (CP/M) operating system, 136
- Control Video Corporation, 173
- Cook, Scott, 183
- CoolWebSearch browser hijacker, 366
- Copy protection scheme, 343–344, 383
- Copyleft, 154, 275
- Copyright lawsuit of Borland and Lotus, 176
- Core memory, 70–72
- Core ropes, 119–120
- Corporate databases, 355
- Corporations
  - data theft from, 372–373
  - virtual education, 318
- Cost per defect metric, 225, 284
- Costs
  - cyberattacks, 402–406
  - identity theft, 395
  - software projects, 306–307
  - virtual education, 319–321
  - weapons and defense, 323
- Council for Scientific and Industrial Research Automatic Computer (CSIRAC), 52–53
- “Cowboy programming”, 98
- Cox, Brad
  - Objective-C, 166
  - Stepstone Corporation, 197

- CR-Z automobiles safety recall, 348
- Craigslist company, 216–217
- Crawford, Perry, 71
- Cray, Seymour, 134
- Cray Computers, 134
- CreateAFund organization, 304
- Credit cards and credit card numbers
  - cyberattacks, 223–224
  - cybercrime response unit, 355
  - invention of, 109–110
  - skimming, 384
  - smart, 385
  - theft of, 358, 373, 396
- Credit ratings, 272
- CreditReport company, 379
- Creep virus, 390
- Crime. *See* Cybercrime and cyberwarfare
- Cro-Magnons, 3–4
- Crowd intelligence, 312–313
- CrowdCube company, 300–301
- Crowdfunding
  - future, 312–313
  - GoFundMe organization, 304
  - overview, 300–301
- Crowdsourcing, 156, 247
- Cryptographic machines, 51–52
- Cryptologic bombs, 51
- CSI Global Education company, 272
- Cullinane, John, 134
  - DCI seminars, 179
  - oral recordings, 421
- Cullinane company, 134–135
- Cultural perceptions of computers, 80–81
- Cunningham, Ward
  - Agile Manifesto, 255
  - technical debt metric, 284
  - Wikipedia concept, 254
- Curriculum planning engine, 315–316
- Curtis, Bill, 194
- Cutter IT Journal* magazine, 145
- Cyber arms race, 394
- Cyber Liabilities insurance conference, 403
- Cyber Security Act, 372
- Cyber Security Agency, 403
- Cybercrime and cyberwarfare, 357
  - access controls, 399–400
  - attack types overview, 362–363
  - authentication controls, 400
  - Bluetooth hijacking, 363–364
  - Botnets, 364–365
  - browser hijackers, 365–366
  - civilian targets, 371–372
  - corporate theft, 372–373
  - credit cards, 109, 223–224
  - criminal activities, 33–34
  - defenses, 354–356, 394–396
  - denial of service attacks, 374–375
  - email address harvesting, 376–378
  - EMP, 375–376
  - EMP protection, 401
  - encryption, 401–402
  - frequency and cost, 404–406
  - future, 288, 313–314
  - growth, 358–362
  - hacker invasion, 352–354
  - history, 351–352
  - identity thefts, 378–379
  - immunity to, 396–399
  - insurance, 403
  - Java attacks, 379–380
  - keyboard trackers, 380–381
  - macro attacks, 381
  - odds of being attacked, 392–394
  - overview, 351
  - pharming, 381
  - phishing, 381–383
  - recovery costs, 402–403
  - rootkits, 383–384
  - secure programming languages, 404
  - seriousness, 356–358
  - skimming, 384–385
  - smart card hijacking, 385–386
  - software, 148–149
  - spam, 387–388
  - SQL injection attacks, 388
  - start of, 256
  - Trojans, 388–389
  - unsecured networks, 373–374

- viruses, 389–390
- voting fraud, 366–371
- worms, 390–391
- zero-day security attacks, 391–392
- Cybersecurity in virtual education, 319
- Cyclomatic complexity metric, 160, 182
- Dahl, Duane, 275
- Daimler automobiles safety recall, 348
- Damphousee, Brad, 304
- Dasai, Keval, 298
- Data encryption, 398, 401–402
- Data theft
  - from corporations, 372–373
  - from unsecured networks, 373–374
- Data warehouses, 25
- Database management systems (DBMS), 95
- Database software, 149
- Databases, 25, 139, 159, 355
- Datanet modem, 109
- Date formats, 239–242
- Date-related software updates, 238
- Datta, Sudeshna, 260
- Dave and Buster's company, 224
- David's Consulting Group, 197
- Davidson, Mary Ann, 357
- Deaf people
  - virtual education, 320
  - wearable computers for, 326
- Debit card information
  - skimming, 384
  - theft, 373
- Decimal numbers, 21–22
- Decimal point, 21
- Decline and Fall of the American Programmer* (Yourdon), 144, 414
- Deep Blue supercomputer, 232
- Defect removal, 290
- Defense Advanced Research Projects Agency (DARPA)
  - cybercrime response unit, 354
  - for SEI creation, 192
- Defense computers in 1950s, 77–80
- Defenses for cyberattacks, 354–356, 394–396
- Degrees from virtual education, 319
- Delphi product, 176
- Demand for programmers in 1960s, 100–103
- DeMarco, Tom, 408
- DeMarco, Tony, 140
- Deming, W. Edwards, 50, 83
- Deming Prize, 83
- Democratic National Committee (DNC)
  - headquarters break-in, 127
- Denali Systems, 300
- Denial of service attacks, 364, 374–375
- Dennard, Robert, 70
- Denver airport delays, 337–338
- Depth charges, 58, 63
- Derby, Jack, 278
- Desk checking, 99
- Devine, Scott, 236
- Devlin, Mike, 189
- Diamond v. Diehr*, 20
- DIANA metalanguage, 190
- Differential calculator, 37–38
- Digital computers for software, 22
- Digital Consulting Institute (DCI), 179–180
- Digital Domain company, 249
- Digital Equipment Corporation (DEC) company
  - Compaq acquisition of, 144
  - creation, 88
  - firewalls, 355
  - PDP-1, 110–112
- Digital Playground company, 217–218
- Digital Research company, 135–136
- Digital video disks (DVDs), 168, 211
- Digital watches, 160
- Dijkstra, Edsger, 103
- Diners Club credit cards, 109
- Disabled people, virtual education for, 316
- Disk drives, 159
- Disk operating systems (DOS), 136, 166
- Distributing information, channels for, 9

- DNS analysis tool, 389
- DNS Changer Trojan, 388–389
- Doenitz, Karl, 63
- Domain Name System (DNS), 221
- Domain names, 264
- Dorsey, Jack, 255, 279
- Dot-com bubble, 248–250
- Douglis, Charles, 196
- “Dr. Dobb’s Journal of Tiny Basic Calisthenics and Orthodontia” journal, 135
- Dr. Dobb’s site, 417
- DR DOS operating system, 136
- DRAM, invention of, 70
- Draper, Charles Stuart, 120
- Duffield, Dave, 189
- Duffy, Tim, 271
- Dummer, Geoffrey, 82
- Duquesne, Fredrick, 48
- Duval County, Florida, election problems in, 369
- Dylan, Bob, 411
- Dynabook notebook computer, 117
- Dynamic modeling, 191
- E-books
  - readers, 215
  - virtual education, 317
- E-learning tools, 320
- E programming language, 404
- EAI 680 scientific hybrid computer, 117
- Eames, Ray, 418
- Early Office Museum, 94
- Early-warning system failure, 334
- eBay company
  - craigslist ownership, 217
  - overview, 218–220
- Eckert, J. Presper
  - EMCC, 72
  - ENIAC computer, 65
  - ENIAC lawsuit, 158
  - UNIVAC computer, 82
- Eckert, Wallace John, 67
- Eckert-Mauchly Computer Company (EMCC), 72
- Economic importance, 2
- Economics of Software Quality* (Jones and Bonsignour), 411
- Education
  - future, 314–321
  - software, 149–150
  - tools, 25
- EDVAC computer, 65
- Edwards, Nathan, 399
- Egypt, prehistoric knowledge in, 6
- Einstein, Albert, 49
- Electrical Numerical Integrator and Computer (ENIAC)
  - Honeywell and Sperry-Rand lawsuit, 158–159
  - overview, 64–65
  - Sperry-Rand against Honeywell case, 118–119
  - women programmers, 44
- Electromagnetic pulses (EMP), 375–376, 401
- Electronic Data Systems (EDS) company, 96, 133
- Electronic Delay Storage Automatic Calculator (EDSAC), 70–71
- Electronic Discrete Variable Automatic Computer (EDVAC), 55
- Electronic Systems and Software company, 368
- Electronic voting ballots, 367–371
- Ellison, Larry
  - DCI seminars, 179
  - in *Jobs*, 411
  - Oracle founding, 139
  - wealth of, 145
- Email addresses
  - harvesting, 376–378
  - phishing, 382
- Embedded devices, 25–26, 78
- Embedded software, 150–151
- EMC corporation, 236
- Emmott, Bill, 309
- Employment websites, 227
- Enabling inventions, 10, 18–20
- Encryption, 398, 401–402

- English, Larry, 134
- ENIAC computer, 64–65
  - Honeywell and Sperry-Rand lawsuit, 158–159
  - overview, 64–65
  - Sperry-Rand against Honeywell case, 118–119
  - women programmers, 44
- Enigma code machine, 44, 49, 51
- Entertainment software, 151
- Epsilon company, 359
- Equal rights and equal pay, 38, 44
- Equifax company, 379
- Equivalency determination, 201
- Ericsson, John, 412
- Ericsson AXE telephone switching system, 384
- Espionage, analog, 48
- Estimating Software Costs* (Jones), 410
- Estonia, internet shutdown in, 359–360
- ESTSoft company, 360
- Eubanks, Gordon, 198
- Euro currency, 237
- European Network, 403
- Everett, Robert, 71
- Evolving workforce in 1960s, 93–95
- Ewing, Mark, 230
- Excel documents, macro attacks in, 381
- Expensive Typewriters, 112
- Experian company
  - identity theft support, 379
  - portfolio management, 272
- Extended Binary Coded Decimal Interchange Code (EBCDIC), 95
- Extreme Programming (XP), 256
- Facebook, 157
  - introduction, 255
  - overview, 263–265
  - virtual education, 317
- FaceSmash site, 263
- Fad, Bruce, 140
- Fadell, Tony, 308
- Fagan, Michael, 162, 408
- Falkoff, Adin, 114
- Families of computers, 105
- Faraday boxes, 401
- FBI Trilogy project failure, 342–343
- Federal Aviation Administration (FAA), 89
- FedEx
  - overview, 129
  - reliance on computers, 145
- Fermi, Enrico, 49, 375
- Festinger, Leon, 412
- Fidelity Growth Partners, 261
- Field-length problems, 340–341
- Fink, Lloyd, 281
- Fire-control computers, 56
- Firefox browser, 229, 273
- Firewalls, 355, 396–397
- First commercial computers, 85–89
- “First Draft of a Report on the EDVAC” (von Neumann), 65
- Fischer, Thomas, 137
- FISH machines, 49
- Fitch’s company, 272
- Fiverr company, 301–302
- Flame attacks, 361
- Flash drives, 254
- Flashback Trojan malware, 379
- Flattr company, 302–303
- Floppy disks, 159
- Florida, electronic voting problems in, 368
- Flowers, T. H., 54
- “Fly-by-wire” systems, 120
- Follett tool, 320
- Ford, Gerald, 127
- Ford automobiles safety recall, 348
- Foreclosures, 252–253
- Forrester, Jay, 70–72
- FORTRAN programming language, 84, 88
- Foster, Peter, 403
- Foundry Group, 310
- Fowler, Martin, 234, 255
- Frame, Jim, 121
- France in World War II, 52
- Frankston, Robert, 161, 421

- FreeX operating system, 275
- Free Standards Group, 274
- Free wireless networks, 373–374
- Freelanthropy group, 201, 265–266
- Freemium business model, 311
- Freiman, Frank, 140
- Freitas, Nancy, 137
- Frequency and costs of U.S. cyberattacks, 404–406
- Friden Flexowriters, 111
- Fuel hedging, 130
- Function points
  - 1955, 90–91
  - 1965, 122–123
  - 1975, 162–163
  - 1985, 204–205
  - 1995, 244–245
  - 2005, 284–285
  - 2015, 328–329
  - overview, 225–226
  - use of, 167
- Future System operating system, 146
- Gack, Gary, 408
- Galorath, Dan, 136
- Galorath Incorporated, 136–137
- Games
  - bouncing ball, 72
  - companies, 260
  - software, 151
  - Spacewar, 112
  - success of, 231–232
- Gandy, Steve, 271
- Gates, Bill
  - browser suit, 210
  - on education, 149
  - entrepreneur of year, 180
  - in *Jobs*, 411
  - Microsoft founding, 131, 138–139
  - referred IBM to Kildall, 136
  - social issues, 174
  - spam to, 387
  - wealth of, 145, 166
- Gates, Mary Maxwell, 105, 139
- Gates, Rick, 254
- Gauss virus attacks, 372
- Geekli.St organization, 303
- Gender discrimination, 38, 44
- Geological time periods, 241
- GeoStar system, 232
- Germany in World War II, 47–49
- Gerrold, David, 353, 390
- Geschke, Charles, 172
- Gibson, William, 353
- Gift of Hearing Foundation (GOHF), 200
- Gilb, Tom, 408
- Glad, Mole, 175
- Glass, R. L., 409
- Global conflicts in 1940s, 43–44
- Global Insight company, 266–267
- Global Payments company, 373
- Global System for Mobile Systems Communications (GSM) standards, 278
- Global warming reversal method, 268
- Glushkov, V. M., 53
- Gmail, spam blocking by, 388
- GNU General Public License (GPL), 153–154, 275
- GNU project, 275
- GoDaddy Bowl, 221
- GoDaddy company, 220–221
- Goetz, Martin, 114
- GoFundMe organization, 304
- Goldsmith, Thomas, 69
- Golf courses mapping, 211
- Gonzalez, Albert, 224, 359
- Google company
  - cloud, 300
  - hacked, 360
  - overview, 221–223
  - search engine, 210
- Google Glasses, 223, 325–327
- Google Maps product, 223
- Google Ventures, 310
- Googolplexes, 222
- Gore, Al, 209, 369–370
- Government programs, 324–325
- Government-sponsored botnets, 364
- Government tools, 26–27



- Gow, Brad, 403
- GPS satellites
  - golf courses, 211
  - notebooks and tablets tracking, 398
- Graham, Dorothy, 408
- Graphical history of mathematics, 9
- Gray, Elisha, 182
- Gray hat hackers, 353
- Great Britain in World War II, 54–59
- Great Recession, 250–254
- Green, Diane, 236
- Green, Michael, 142
- Grenning, James, 255
- Grosch, Herb, 67
- Ground Observation Corps, 79
- Groupe Bull company, 52
- Groves, Ken, 411
- Guest speakers for virtual education, 318
- GUIDE association, 87
- Guided missile, 80
- Gulf War, patriot missile target error in, 336–337
- Gun control computer, 42
  
- Hacker invasion, 352–354
- Hacking, 148–149
- Hactivism, 356
- Hambrecht and Quist company, 135
- Hamilton, Francis, 67
- Hamilton company, 160
- Harappa city, 6
- Harmful inventions, 32–34
- Hartford Coliseum collapse, 333
- Harvard architecture, 66
- Harvard Connection project, 264
- Harvard Mark I computer, 40, 56, 66–67
- Harvesting bots, 377
- Health condition monitoring, 326
- Healy, Jim, 121
- Hearing aids
  - future products, 322
  - wearable computers for, 326–327
- Heartland Payment Systems
  - credit card theft from, 358–359, 373
  - overview, 223–224
  
- Hector, Hans-Werner, 142
- Heil, Oskar, 82
- Hemy, Derek, 86
- Henderson Carriage building, 195
- Hendrix, Gary, 198
- Henriksen, Ole, 175
- Henry-Dahl, Cindy, 275
- Hewett, William, 40
- Hewlett-Packard company, 40, 404
- Hexadecimal numbers, 21
- High Performance Computing and Communications Act, 209
- Higher-level programming languages
  - evolution of, 161
  - overview, 22
- Highsmith, Jim, 255
- Hijackers, browser, 365–366
- Hill, Peter, 225
- Hill, Winslow, 177
- History of the Software Industry: From Airline Reservations to Sonic the Hedgehog* (Campbell-Kelly), 408
- History Timelines site, 417–418
- HIVE Group, 267–268
- Hoff, Ted, 50
- Home heating and cooling control
  - product, 308
- Home networks, 373
- Honda automobiles safety recall, 348
- Honeypots, 377
- Honeywell company
  - Nest Labs lawsuit, 309
  - Sperry-Rand lawsuit, 158–159
- Hoover's Business Guides site, 418
- Hopp, Detmar, 142
- Hopper, Grace
  - Mark I design, 40, 66
  - oral recordings, 421
- Horing, Jeff, 224
- Horowitz, Ben, 261
- Huawei company, 180–182
- Huawei Symantec company, 199
- Hub-and-spoke model, 129
- Hughes, Chris, 263
- Human need to compute, 1–3

- Human resource management systems (HRMS), 188–189
- Humphrey, Watts, 409
  - software assessments, 166
  - TSP, 256
- Hunt, Andrew, 255
- Hurley, Chad, 280
- HYCOMP desktop hybrid computer, 117
- HYDAC 2400 hybrid computer, 117
- Hypertext markup language (HTML), 208
- Hypertext transfer protocol (HTTP), 208
- IBM
  - antitrust lawsuit, 119, 136
  - Apollo program, 120
  - Automatic Sequence Controlled Calculator, 66
  - Bull lawsuit, 52
  - first commercial computers, 86–88
  - function points, 162–163
  - graphical history of mathematics, 9
  - longevity of, 128
  - Lotus acquisition, 186
  - magnetic core memory, 72
  - magnetic tape, 109
  - personal computer release, 166
- IBM 650 computer, 87–88, 98
- IBM 701 computer, 86–87
- IBM 1401 computer, 98–100, 110–111
- IBM 1403 printer, 99–100
- IBM 5100 computer, 160
- IBM 7090 computer, 87
- “IBM and the Seven Dwarves”, 115
- IBM Corporate Archives site, 418
- IBM Math Timeline site, 418
- IBM Rational company, 191–192
- IBM System/360 computer, 104–106
- Ichitaro, Kawaguchi, 50
- Identity Theft Resource Center, 379
- Identity thefts
  - costs, 395
  - highest incidence states, 396
  - overview, 378–379
  - pervasiveness, 34
- Ideograms, 21
- IEEE Computer Society site, 418–419
- “Illustrated History of Computers”, 416
- Immunity to cyberattacks, 396–399
- Impact of software on people and society
  - beneficial tools and applications, 23–32
  - harmful inventions, 32–34
  - risks, 34–35
- IMSAI computers, 131–132, 137
- IN04 directive, 255
- India
  - encryption in, 402
  - scholars, 8
- Industrial sites, cyberattacks on, 372
- Information management system (IMS), 95, 135, 137
- Information storage and access
  - distribution channels, 9
  - evolution of, 17–18
  - recording information, 14
- “Information superhighway” term, 209
- Information Technology Metrics and Productivity Institute (ITMPI)
  - description, 177, 255
  - website, 419
- Infrastructure costs for virtual education, 320
- Injection attacks, 358–359, 388
- Innovators and innovations
  - 1930s, 37–40
  - 1950s, 81–84
  - 2000s, 254–257
- Inprise company, 176
- Insight Venture Partners, 224
- Instant messaging for phishing, 382
- Instant translation, wearable computers for, 326
- Institute for Numerical Analysis, 81
- Institute of Electrical and Electronic Engineers (IEEE), 68
- Institute of Radio Engineers (IRE), 68
- Instructional staff for virtual education, 315–317, 320
- Insurance companies in 1960s, 94

- Insurance for cybertheft and cyberattack damages, 403
- Integrated circuits
  - development of, 81–82
  - as enabling invention, 20
  - patent, 83
- Integrated Data Base Management System (IDMS), 135
- Intel Pentium chip division problem, 337
- Intellectual Ventures, 268–269
- Intellectual Ventures Labs, 268
- Intelligent agents
  - description, 202
  - future, 321–322
- Intelligent Assistant tool, 198
- Interactive Productivity and Quality model (IPQ), 114
- InterBank Card Association (ICA), 110
- International Computers Limited (ICL), 86
- International cyberwarfare, 372
- International date formats, 239–242
- International Function Point Users Group (IFPUG)
  - in Brazil, 255–256
  - ISBSG functions points, 225
  - move to U.S., 167
  - website, 419–420
- International Organization of Standards (ISO) standard date format, 239–242
- International Software Benchmark Standards Group (ISBSG), 225
  - formation, 211
  - function point projects, 256
  - website, 420
- Internet Crime Complaint Center, 396
- Internet Crime Report, 396
- Internet Explorer browser, 273
- Internet Impact Awards, 270
- Internet Marketing Association (IMA), 270
- Internet pornography, 218
- Internet protocol (IP), 352
- INTERPROGRAM programming language, 53
- InterSystems Corporation, 137–138
- Intuit company, 183–184
- Intuit Quicken product, 183–184
- Inventions for improved mathematics
  - calculating, 10–14
  - communication channels, 15–17
  - enabling, 18–20
  - overview, 8–10
  - recording information, 14
  - storing information, 17–18
- Inventions overview
  - background enabling, 159–162
  - harmful, 32–34
  - for software, 21–23
- Investigation on The Laws of Thought* (Boole), 39
- Iomega Zip Disk, 168
- iPads, 256
- iPhones, 256
- Iran, cyberattacks on, 360–361, 372
- Isaacson, Walter, 132, 149, 409
- ISBSG company, 225–227
- IStation tool, 320
- IT History Society site, 420
- Italy, function points in, 256
- Iverson, Ken, 114, 121
- Iyengar, Amand, 300
- J. Lyons Company, 85
- Jackson, Thomas Penfield, 210
- Jacobsen, Ivar, 190–191
- Jaech, Jeremy, 235
- Jaguar automobiles safety recall, 348
- Japan
  - training in ancient times, 7
  - World War II, 50–51
- Java programming language attacks, 379–380
- Jazz product, 186
- JBuilder product, 176
- JD Edwards company, 189
- Jeffries, Ron, 255
- Jensen, Niels, 175

- Jesse, H. William, 267  
Jet aircraft, 88  
Job control language (JCL), 113–114  
Jobs, Steve  
    Apple founding, 131  
    biography, 132, 149, 409  
    DCI seminars, 179  
    on education, 149  
    entrepreneur of year, 180  
    NeXT company, 187  
    Objective-C, 198  
    Turing award, 106  
    wealth of, 145  
    Xerox influence, 117  
Johnson, Ted, 235  
Joint Application Design (JAD), 212  
Jolt awards, 417  
Jomax Technologies, 220  
Jones, Capers  
    *Applied Software Measurement*, 411  
    *Assessment and Control of Software Risks*, 410  
    *Economics of Software Quality*, 411  
    *Estimating Software Costs*, 410–411  
    Namcook Analytics founding, 304  
    *Program Quality and Programmer Productivity*, 146  
    *Software Engineering Best Practices*, 411  
    SPR founding, 194  
Jones, Eileen, 194, 200  
“Joone” film maker, 217  
Joule programming language, 404  
Juran, Joseph, W., 50, 83  
Just-in-time manufacturing, 152  
  
Kahn, David, 401–402  
Kahn, Philippe, 175  
*Kama Sutra*, 402  
Kanatsiz, Sinan, 270  
Kapoor, Suhale, 260  
Kapor, Mitch  
    Lotus founding, 186  
    Mozilla Foundation funding, 273  
Karim, Jawed, 280  
  
Kasner, Edward, 222  
Kasparov, Gary, 232  
Kaspersky company, 355  
Kasputys, Joseph, 267  
Katz, Reuben, 303  
Katzman, James, 142  
Kaufman, Micha, 301  
Kaul, Anil, 260  
Kay, Alan, 117  
Kelleher, Herb, 130  
Kemeny, John, 115  
Kemp, John, 19  
Keppler, Johannes, 324  
Kern, Jon, 255  
Key inventions for software, 21–23  
Key process areas (KPAAs), 192  
Keyboard trackers, 380–381  
Kilburn, Tom, 69  
Kilby, Jack, 83  
Kildall, Dorothy, 135  
Kildall, Gary, 135–136  
Kindle e-book readers, 215  
King, Earl, 140  
King, Ernest, 62  
*King George V* battleship, 42, 56  
Knight Capitol stock-trading software problems, 347  
KnowledgePLAN tool, 196  
“Known Attacks Against Smart Cards” (Bar-El), 386  
Kongo Gumi company, 128  
Korte, Thomas, 298  
KPMG company, 184–185  
Kubrick, Stanley, 81  
Kuhn, Thomas, 412  
Kumar, Sanjay, 133  
Kurtz, Thomas, 115  
Kurtzman, Clifford, 238  
  
LaCrosse automobiles safety recall, 348  
Lang, Luke, 300  
Langevin, James  
    Congressional Cyber Caucus, 110, 355  
    Cyber Security Act, 372  
Language levels, 123

- Languages
  - alphabetic, 21
  - programming. *See* Programming languages
- Larson, Earl R., 158
- Las Zetas drug cartel, 361
- Late millennium bug, 386
- Latent security flaws, 403
- Law, future of, 319
- Lawson, Richard, 138
- Lawson, William, 138
- Lawson Software, 138
- Lawsuits
  - Apple and NeXT, 187
  - Borland and Lotus, 176
  - Bull and IBM, 52
  - CA and Electronic Data Systems, 133
  - embedded software, 151
  - Facebook, 265
  - Honeywell and Sperry-Rand, 118–119, 158–159
  - IBM and Digital Research, 136
  - Microsoft and Intersystems, 138
  - Microsoft antitrust, 210, 249, 255
  - Nest Labs and Honeywell, 309
  - Oracle and SAP, 139
  - patent, 20, 269
  - Sony copy protection, 344
  - thumb drives, 254
- Lazaridis, Mike, 174
- Leaf automobiles safety recall, 348
- Learning and education
  - future, 314–321
  - software, 149–150
  - tools, 25
- Lebanon, cyberattacks on, 361
- Lebedev, S. A., 53, 82
- Legacy applications
  - 1990s, 237–239
  - 2000s, 257
  - maintenance of, 295
- Lehman Brothers company, 251
- Lehnert, Markus, 310
- Leighton, Tom, 214
- Leisure tools, 27
- LEO computer, 85–86
- LEO Computers Limited company, 86
- Leon County, Florida, electronic voting problems in, 368
- Lerner, Sandy, 178
- Levy, Paul, 189
- Lewin, Daniel, 214
- Leyden, John, 386
- Li, Robin, 222
- Libraries
  - for storage, 18
  - virtual universities, 316
- Licenses
  - GNU, 153–154, 275
  - software engineering, 289
  - virtual education, 317
- Lieberman, Joe, 355–356
- Lieberman Collins Cyber Security Bill, 356
- Life expectancy of American corporations, 128
- Lillienfeld, Julius, 82
- Lines of code (LOC) metric, 167, 225
- LINK trainer, 71
- Linked oscillating systems, 252
- LinkedIn network
  - data thefts from, 373
  - lengthy messages, 279
  - virtual education, 317
- Linux Foundation, 274
- Linux operating system
  - development of, 210
  - emergence of, 274–275
  - kernel, 230
  - Ubuntu version, 262
- Linux Trademark Institute, 275
- LISP language, 84
- Lister, Tim, 408
- Litigation. *See* Lawsuits
- Littlejohns, Doug, 231
- Local information in virtual education, 317
- Logarithms, 21–22
- London Science Museum, 9
- Long lance torpedoes, 50

- Los Angeles Airport (LAX) air-traffic controls shutdown, 341–342
- Lotfernrohr 7 bombsight, 48
- Lotus company
  - Borland lawsuit, 176
  - overview, 186–187
- Loustaunou, Jack, 142
- Love, Tom
  - Object Lessons*, 412–413
  - Stepstone Corporation, 166, 197
- Luggage handling failure, 337–338
- Lulzsec group, 373
- Lyons Electronic Office computer, 86
  
- M-1 computer, 82
- Machine language
  - limitations, 22, 84
  - small mathematical applications, 41
- Macro attacks, 381
- Macro instructions, 84
- Magnetic core memory, 70–72
- Magnetic tape, 109
- Magneto-optical drives, 188
- Magnetrons, 61
- Maintenance of legacy applications, 295
- Malone program, 263
- Malware term, 381
- Malwarebytes tool, 313
- Man-in-the-middle attacks, 400
- Management information systems (MIS), 149
- Managing the Black Hole—The Executive's Guide to Project Risk (Gack)*, 408
- Managing the Software Process (Humphrey), 409
- Manchester Automatic Digital Machine (MADM), 69
- Manchester Mark I computer, 69
- Mann, Estie Ray, 69
- Manufacturing software, 151–152
- Manwin company, 218
- Manzi, Jim, 186
- Marconi, Guglielmo, 352
- Marick, Bran, 255
- Mariner 1 navigation software failure, 333
- Mariposa botnet, 364
- Mark I computer (CSIRAC), 53
- Mark I computer (Harvard), 40, 56, 66–67
- Mark I computer (Manchester), 69
- Mark II computer, 67
- Mark III computer, 47, 62, 67
- Mark IV computer, 62, 67
- Mark VI detonators, 62
- Mark 14 torpedoes, 62
- Mark 18 torpedoes, 62
- Markoff, Philip, 217
- Maroney, Ted, 304
- Mars climate orbiter crash, 339
- Marshall stores, 359
- Martin, James, 145
- Martin, Robert C., 255
- Mass updates in 1990s, 237–239
- Massively multiplayer online role-playing games (MMORPG), 231
- MasterCard credit cards, 110
- Match-making services, 275–276
- Mathematica product, 202–203
- Mathematical applications, 40–42
- Mathematical software, 155–156
- Mathematics
  - graphical history, 9
  - inventions. *See* Inventions for improved mathematics
- Mauchly, John W.
  - EMCC, 72
  - ENIAC computer, 65
  - ENIAC lawsuit, 158
  - UNIVAC computer, 82
- Mayans, 22
- McAfee Antivirus bug, 345
- McAfee company, 256
- McCabe, Tom, 160, 182
- McCaul, Mike, 355
- McClure, Carma, 145
- McCollum, Andrew, 263
- McConnell, Steve, 413
- McCool, Robert, 216

- McDonald, Stuart, 388
- McPherson, John, 67
- McVey, Jason, 275
- Mean time to failure (MTTF), 143
- Medical degrees, 414
- Medical devices, 322–323
- Medical records, 138
- Medical tools, 27–28
- Medicine, future, 319
- Meeting Zone company, 270–271
- Meister, Bill, 173
- Melbourne Computer Museum site, 420
- Mellor, Steve, 255
- Memory
  - magnetic core, 70–72
  - Williams-Kilburn tube, 69–70
- Mercury delay lines, 71
- MESM computer, 53, 82
- Metal cases for EMP, 401
- Metrics
  - overview, 225–226
  - social-network effectiveness, 309
- Meucci, Antonio, 182
- Michalewicz, Matthew, 277
- Michalewicz, Zbigniew, 277
- Micro Focus company, 176
- Micro Instrumentation and Telemetry Systems Corporation (MITS), 131
- Microblogging, 278
- Microdonations, 302–303
- Microsoft Basic language, 138–139
- Microsoft Corporation
  - antitrust lawsuit, 210, 249, 255
  - founding, 131, 138–139
  - Google competition, 223
  - Intersystem lawsuit, 138
  - Visio acquisition, 235
  - wealth from, 145
- Microsoft DOS (MS-DOS) operating system, 166
- Microsoft Money product, 183–184
- Middleware software, 152
- Military computers in 1950s, 77–80
- Military software, 152–153
- Millard, William, 137
- Mills, Harlan, 121
- Mims, Forrest, 131
- Mindplay tool, 320
- Miner, Bob, 139
- Minicomputers, 111
- Ministry of International Trade and Industry (MITI), 83
- Missiles
  - BOMARC, 80
  - target errors, 336–337
  - World War II, 48
- MITRE vulnerabilities list, 394
- Mittal, Vibuh, 298
- Mockapetris, Paul, 221
- Model 1 computing machine, 39
- Modeling, dynamic, 191
- Modems, 109
- Mohenjo-Daro city, 6
- Molitano, David, 403
- Monster.com, 227–228
- Moody, John, 271
- Moody's Analytics company, 271–272
- Moody's Foundation, 271
- Morgan Stanley company, 265
- Morris, Ken, 189
- Morris, Robert, 391
- Morris worm, 391
- Mortgages, 251–253
- MOSAIC computer, 55
- Mosaic web browser, 209
- Moskowitz, Dustin, 263
- Mozilla Foundation, 229, 272–273
- Multitasking operating systems, 120
- MUMPS programming language, 138
- Murdock, Jerry, 224
- Mutually assured destruction, 69
- MyDoom worms, 391
- Myers, Glenford, 146
- Myhrvold, Nathan, 268
- MySearch browser hijacker, 366
- Mythical Man-Month* (Brooks), 104, 146, 408
- Nalanda University, 7–8
- Namcook Analytics LLC, 226, 304–307

- Nano devices, 323
- Napier, John, 21–22
- Napier’s bones, 22
- Narenda, Divya, 264
- NASDAQ stock exchange, 129, 145
- Nash Bargaining Solution, 269
- National Center for Supercomputing Applications (NCSA), 209
- National defense
  - cybercrime, 357
  - tools, 28
- National information in virtual education, 317
- National Institute of Standards and Technology (NIST)
  - analysis tool study, 395
  - standard date format, 239
- National Institutes of Health (NIH), 98
- National Museum of Computing site, 421
- National Research and Education Network, 209
- National Science Foundation (NSF), 221
- National Security Agency (NSA), 82–83
- Natural language translation, 327
- Naur, Peter, 103
- Navajo code talkers, 16
- Naval gunnery and battles, 56–57
- Navigation, prehistoric knowledge for, 7
- Neanderthals, 3–4
- Nelson* ship, 56
- Nest Labs company, 308–309
- Nest Learning Thermostat, 308–309
- Netscape browser, 273
- Netscape Communications, 228
- New economy, 248–250
- Newmark, Craig, 216–217
- NeXT company, 187–188
- NeXT computer for browser design, 208
- NeXTStep operating system, 188
- Nies, Thomas, 119
- Night vision, 326
- Nimoy, Leonard, 229
- Nippon Calculating Machine company, 50
- Nishar, Deep, 298
- Nissan automobiles safety recall, 348
- Nixdorf company, 88
- Nixon, Richard, 127
- Nonprofits
  - Bill and Melinda Gates Foundation, 219
  - in computing and software, 200
  - Google philanthropic wing, 223
- Norden, Carl, 60
- Norden bombsight, 59–61
- North Korea, 323–324
- Norton Utilities, 199
- Notebooks, GPS tracking of, 398
- “Notes on Structured Programming” (Dijkstra), 103
- NoteTaker computer, 160
- Number systems, 21–22
- Numeric-related software updates, 238
- Numerical knowledge, early sequence of, 3–8
- Oates, Ed, 139
- Obamacare, 324
- Object Lessons* (Love), 412–413
- Object Management Group (OMG), 226
- “Object-Oriented Precompiler: Programming Smalltalk—80 Methods in C Language” (Cox), 197
- Objective-C programming language
  - Apple computers, 132, 166
  - NeXT computers, 188, 198
  - Stepstone Corporation, 197
- Objectory tool, 190
- Octal numbers, 21
- Odeo company, 278
- Olmecs, 8, 22
- Olsson, Linus, 302
- Omidyar, Pierre, 218
- “On Computable Numbers” (Turing), 38, 54
- On-Line Career Center (OLC), 227–228
- Online web-based education information, 314–321
- Online “webinars”, 180



- Opel, John, 104–105, 139
- Open-source business model, 263
- Open Source Development Labs (OSDL), 274
- Open-source software, 153–154
- OpenStep operating system, 188
- Operations research, 58
- Oracle company
  - overview, 139–140
  - wealth from, 145
- Oshima, Hiroshi, 51
- Outsourcing
  - 1960s, 95–96
  - 1990s, 243
- Packard, David, 40
- Page, Larry, 210, 222
- Page-description languages, 172
- Palm Beach County “butterfly” ballots, 369
- PanAmSat group, 232
- Pandora music service, 151
- Paper voting ballots, 367–368
- Pappalardo, Neil, 138
- Parametric estimation, 137, 140
- Park, Robert, 140
- Parsons, Bob, 220
- Parsons Technology, 220
- Passport readers, 386
- Patent litigation
  - Honeywell and Sperry-Rand, 158–159
  - Nest Labs and Honeywell, 309
  - recent years, 269
- Patent trolls, 82, 268–269, 276
- Patent wars, 288
- Patents and patent system
  - first U.S. law, 20
  - IBM and Bull lawsuit, 52
  - integrated circuits, 83
  - invention of, 10, 19
  - openness with, 82
  - pattern matching, 305
  - Sperry-Rand and Honeywell lawsuit, 118–119
- Patrick, Danica, 221
- Patriot missile target error, 336–337
- Pattern matching, patents, 305
- Pay-per-click model, 266
- PayPal company, 219
- Pearcey, Trevor, 53
- Peat Marwick company, 185
- Peer Index company, 309
- People’s Liberation Army of China, 372
- PeopleSoft company, 188–189
- PeopleTools methodology, 189
- Peopleware* (DeMarco and Lister), 408
- PerfectMatch company, 275–276
- Perot, H. Ross, 96, 187
- Personal computers, 131
- Personal data theft, 362
- Personal software, 154–155
- Personal Software Process (PSP), 256
- Personal tools, 28–29
- Peter Norton Computing company, 199
- Pfister, Douglas, 81
- Pharming, 381
- Phishing, 381–383
- Phone phreakers, 352
- Pictograms, 21
- Pierce, John R., 82
- Piggybacking, 373
- Pilot Model ACE computer, 55
- Pinkerton, John, 86
- Pirate Bay, 302
- PL/I programming language, 114–115
- Plan Calculus language, 47
- Planet Oasis site, 266
- Plastic
  - as enabling invention, 20
  - importance of, 10
- Plattner, Hasso, 142
- Plaxo for virtual education, 317
- Poduska, William, 141
- Point Judith battle, 63
- Poland in World War II, 51–52
- Poley, Stan, 88
- Political activists, 356
- Political failures, 127–128
- Polly, Jean Armour, 209

- Pornography, 218
- Port St. Lucie, Florida, office failures in, 249
- Portable computers
  - cameras on, 399
  - early, 160
- Portable document format (PDF), 172
- Postwar era, 68–72
- Predictions
  - 2010s, 288
  - companies and products, 311–327
  - software project outcomes, 305
- Predictive analytics
  - future, 323–325
  - working with, 307–308
- Premier company, 368
- Presidential election of 2000, 368–370
- Pressman, Roger, 413
- Pretty Good Privacy, 402
- Price Systems, 140
- Priceline company, 229–230
- Priests in prehistoric knowledge, 5–6
- Primary education, virtual, 320
- Prime Computers (Pr1me), 141
- Printers, 99–100
- Printing, 3D, 288
- Prius automobiles safety recall, 348–349
- Productivity and quality measures
  - improvements, 293
- Productivity Products, 197
- Professional associations for virtual education, 318
- Professional status for software engineering, 289–293
- Professional tools, 29–30
- Program Quality and Programmer Productivity* (Jones), 146
- Programmed Data Processor-1 (PDP-1) computer, 110–112
- Programming aptitude test (PAT), 97
- Programming Language/Systems (PL/S), 123
- Programming languages
  - 1950s, 84
  - 1960s, 112–115
  - higher-level, 22
  - number of, 212
  - secure, 404
- Programming tools, 30–31, 155
- Project estimation improvements, 294
- Project Management Institute (PMI), 168
- Project management software, 155
- Protective software, 156
- Protective tools, 31
- Proult, Tom, 183
- Psychology of Computer Programming* (Weinberg), 121, 146, 408, 414
- Public networks, 373–374
- Punched cards, 52, 99
- Purple coding machine, 21, 51
- Puskas, Tivadar, 183
- QNX operating system, 175
- Quality control
  - improvements, 294
  - telecommunications, 182
- Quality/Productivity Management Group, 197
- Quartz movement, 160
- Quattro spreadsheet, 176
- Quick response (QR) codes, 303
- Quicken Interface Format (QIF), 184
- Quicken product, 183–184
- Radar, 61
- Radiation poisoning, 334–335
- Radice, Ronald, 162, 408
- Radio, satellite, 232–233
- Ragon, Philip T., 137–138
- Ragon, Terry, 145
- Rajaram, Gokul, 298
- Rameev, B. I., 53
- Randall, Brian, 103
- Random-access storage, 23, 70
- Ranking systems on Amazon, 215
- Rational company, 189–192
- Rational Environment, 190
- Rational Unified Process (RUP), 190, 212
- Raymond, Eric, 154
- RCA Corporation, 140

- Real estate bubbles, 250
- Real estate property values, 281–283, 305
- Real-time computation, 78
- Real-time translation services, 316
- Recalls, automotive, 348–349
- Recession in 2000s, 250–254
- Recording of ideas and information
  - invention of, 9
  - methods and media evolution, 13–14
- Recovery costs from cyberattack, 402–403
- Recruited software personnel, 94
- Red Hat Software, 230
- Red Storm Entertainment company, 231
- Redistricting packages, 367
- Redundancy, 143
- Reis, Johan Philip, 182
- Relational database model, 159
- Relay-based computers, 38–39
- Remington-Rand company, 82, 88
- Remote computers
  - input and output, 39–40
  - security for, 352
- Requirements creep, 307
- Research in Motion (RIM) company, 174–175
- Reusable components, 294
- Rhode Island assembly, 2–3
- Rhode Island Computer Museum site, 421
- Richardson, Lewis F., 323
- Richley, Tom, 119
- Rio Tinto Iron Ore company, 277
- Risak, Veith, 390
- Risks
  - assessing, 185
  - weighing, 34–35
- Ritchie, Dennis, 112, 274
- Riverdeep tool, 320
- Roberts, Ed, 131
- Robo calling, 388
- Rodney* ship, 56
- Rogers, Matt, 308
- Roosevelt, Franklin Delano, 61
- Rootkits, 383–384
- Rosen, Ben, 186
- Rosenblum, Mendel, 236
- Rosetta program, 263
- Rothblatt, Martine, 232
- Royce, Walker E., 413–414
- RPX Corporation, 276–277
- RSA Security company, 359
- Ruckert, Toby, 310
- Rumbaugh, James, 191
- Russia in World War II, 53
- Russian Business Bureau group, 382
- Russian Mafia, 357
- Ryoichi, Yazu, 50
- S-100 bus, 131
- SABRE airline reservation system, 79–80, 89
- Sachs, Jonathan, 186
- SAGE air-defense system
  - Cold War, 69
  - overview, 78–80
  - Whirlwind computer for, 72
- Sales taxes, 215
- Salvaggio, Tony
  - CAI founding, 177
  - social issues, 174
- SAMATE study, 395
- Sandberg, Sheryl, 265
- Sands, Christian, 303
- Sanger, Larry, 254
- SANS Institute
  - purpose, 355
  - vulnerabilities list, 394–395
- Satellite radio, 232–233
- Satellites, GPS
  - golf courses, 211
  - notebooks and tablets tracking, 398
- Saudi Aramco company, 372
- Savarin, Eduardo, 263
- Scacco, David, 298
- Scandals, Watergate, 127
- Schilling, Curt, 250, 345
- Schmidt, Martin, 277
- Schussel, George, 179–180

- Schwaber, Ken, 255  
Schwanenflugel, Ditlev, 309  
Schwartz, Matthew, 404  
Schwartz, Pepper, 276  
Scientific purposes  
  dates, 241  
  software, 155–156  
  tools, 31–32  
Script kiddies, 360  
Scrum programs, 256  
SDS/SAPE package, 142  
SeaMonkey browser, 273  
search.conduit browser hijacker, 366  
Search-daily browser hijacker, 366  
Secondary education, virtual, 320  
Secret copy protection software, 343–344  
Secure programming languages, 404  
Secure Sockets Layer (SSL) protocol, 228  
Security. *See also* Cybercrime and cyberwarfare  
  future, 313  
  improvements, 294–295  
  software, 156  
  virtual education, 319  
Security Agency, 403  
Security staffs, 393  
Seeber, Robert, 67  
SEER product, 136  
Segue product, 176  
SEI assessment approach, 193–194  
SEI company, 192–194, 404  
Selective Sequence Electronic Calculator (SSEC), 67  
Seminars  
  DCI, 179  
  virtual education, 319  
*Sentinel of Eternity* (Clarke), 81  
Sequence controlled calculators, 40  
Sequoia Capital company, 280  
Shamans, 5–6  
Shannon, Claude, 39  
SHARE association, 87  
Shatner, William, 229  
Sheehy, Dan, 266  
Shestakov, Victor, 39  
Shneiderman, Ben, 267  
Shockley, William, 81–82  
Short message service (SMS), 278  
Shuttleworth, Mark, 262  
SideKick product, 175  
Siemens computers, 360  
SIGCOM groups, 68  
SIGGRAPH groups, 68  
Silicon Valley, 130–131  
Simula programming language, 114  
Singham, Roy, 234  
Singham Business Services, 234  
Sirius Satellite Radio, 232–233  
Sirota, Milton, 222  
Six pack sales program, 199  
Six Sigma quality program, 182  
Skimming, 384–385  
Sky Map product, 223  
Slide rules, 22, 97  
Small mathematical applications, 40–42  
Small-Scale Experimental Machine (SSEM), 69  
Smart appliances, 309  
Smart cards  
  biometric information in, 400  
  hijacking, 385–386  
Smart thermostats, 308  
Smartphones as hearing aids, 322  
Smith, C. R., 89  
Smith, Frederick, 129  
Smith, R. Blair, 89  
SNAP metrics, 226, 295  
Sniffers, 374  
Snow, Gordon, 384  
Snowbird Lodge meeting, 255  
*Social Network* film, 255, 263  
Social networks  
  software, 156–157  
  virtual education, 317  
Social security stolen files, 361  
*Social Transformation of American Medicine* (Starr), 414  
“Software”, first use of word, 83  
Software  
  1940s, 75–76

- for education. *See* Education
- invention of, 10
- key inventions for, 21–23
- patents, 20
- security improvements, 294–295
- Software applications
  - 1930s, 41
  - 1940s, 74–75
  - 1950s, 89–90
  - 1960s, 120–122
  - 1970s, 146–147
  - 1980s, 204–205
  - 1990s, 243–244
  - 2000s, 283–284
  - 2010s, 327–328
- Software as a service (SaaS) applications, 257
- Software categories in 1970s, 147–158
- Software Development Laboratories company, 139
- Software Engineering—A Practitioner’s Approach* (Pressman), 413
- Software Engineering Best Practices* (Jones), 411
- Software Engineering conference, 103
- Software Engineering Economics* (Barry), 407
- Software Engineering Institute (SEI), 166
- Software Engineering Methods and Theory (SEMAT), 296
- Software engineers
  - emergence of, 103–104
  - Geekli.St organization, 303
  - improvements in, 293–296
  - professional status for, 289–293
- Software evolution in 1970s, 125–129
- Software failure analysis
  - Airbus A380 wiring, 344–345
  - Ariane 5 rocket, 338–339
  - AT&T telephone lines, 336
  - automotive safety recalls, 348–349
  - British passport system, 339–340
  - Denver airport, 337–338
  - FBI Trilogy project, 342–343
  - Hartford Coliseum collapse, 333
  - Intel Pentium chip division, 337
  - Knight Capitol stock-trading software, 347
  - Los Angeles Airport air-traffic controls, 341–342
  - Mariner 1, 333
  - Mars climate orbiter, 339
  - McAfee Antivirus, 345
  - overview, 331–332
  - patriot missiles, 336–337
  - Sony copy protection, 343–344
  - Soviet early-warning system, 334
  - Studio 38 investment, 345–346
  - Therac 25, 334–335
  - Wall Street crash, 335
  - Y2K problem, 340–341
- Software Industry Special Interest Group site, 421
- Software Inspections* (Gilb and Graham), 408
- Software Process Improvement Network (SPIN), 270
- Software Project Management—A Unified Approach* (Royce), 413–414
- Software Query Language (SQL) injection attacks, 358–359, 388
- Software Risk Master (SRM) tool
  - cyberattack recovery costs, 403
  - overview, 305–307
- Software Runaways: Lessons Learned from Massive Software Project Failures* (Glass), 409
- Solar flares, 376
- SolveIT Software, 277–278
- Sony Corporation
  - copy protection software, 343–344, 383
  - hacked accounts, 359
- Sony TR-63 transistor radios, 88
- Sossamon, William D., 262
- SourceForge.net site, 154
- South Korea
  - function points, 256
  - North Korea threat, 324
- Southwest Airlines, 130, 145

- Soviet Union
  - Cold War era, 68–72
  - dissolution, 357
  - early-warning system failure, 334
  - military and defense computers
    - against, 77–80
    - in World War II, 53
- Space, computers in, 119–120
- Space Travel game, 112
- Spacewar game, 112
- Spam, 387–388
- Spear phishing, 382
- Special interest groups in virtual education, 317
- Specialized outsourcing in 1960s, 95–96
- Speculative bubbles, 248
- Sperry-Rand company, 65, 118–119, 158–159
- Spider traps, 377
- Spiders, 377
- Sports, 121, 211
- SPQR/20 estimating tool, 167, 195
- SPR company, 194–197
- Spreadsheets, 186
- Spy rings, 48
- SQL injection attacks, 358–359, 388
- Square company, 279
- SRX automobiles safety recall, 348
- St. Petersburg, Florida, electronic voting problems in, 368
- Stabilized Bombing Approach Equipment (SBAE), 60
- Stallman, Richard
  - GNU project, 274
  - open-source software, 153
  - Wikipedia concept, 254
- Standard and Poors company, 272
- Standingford, Oliver, 85–86
- Stanford Digital Library Project, 222
- Star positions, prehistoric knowledge of, 7
- Starfish Prime test, 375
- Starr, Paul, 414
- StarTeam product, 176
- State Street Bank v. Signature Financial Group*, 20
- Static analysis tools, 395–3496
- “Statistics of Deadly Quarrels” (Richardson), 323
- Stepping switches, 183
- Stepstone Corporation, 197–198
- Steve Jobs: The Exclusive Biography* (Isaacson), 132, 149, 409
- Stevens, John Paul, 176
- Stibitz, George, 39
- Stibitz relay machine, 39
- Stock trading
  - computers for, 117
  - software problems, 347
- Stone, Biz, 278
- Storage
  - evolution of, 17–18
  - permanence of, 14
- Stowger, Almon Brown, 183
- Stowger Automatic Telephone Exchange Company, 183
- Strauss, Richard, 168
- Structure of Scientific Revolutions* (Kuhn), 412
- Structured Design* (Yourdon and Constantine), 146
- Student centers in virtual education, 317
- Studio 38 company, 249, 345–346
- Stuxnet worm, 360, 372
- Submicron nano devices, 323
- Subprime mortgages, 251–252
- Subroutines, 70–71
- Subtraction in prehistoric knowledge, 4
- Sudan, cyberattacks on, 361
- Sunde, Peter, 302
- Super Bowl ads, 248
- Supercomputers, 134
- Supersonic guided missile, 80
- Supreme Court
  - 2000 presidential election, 370
  - Borland and Lotus case, 176, 186
  - patent cases, 20, 288
- “Surfing the Internet” (Polly), 209
- Survey tools software, 157

- Sutherland, Ivan, 114
- Sutherland, Jeff, 255
- Sykipot Trojan, 386
- Symantec company
  - cybercrime research, 355
  - hacking attacks report, 358
  - overview, 198–200
- Symbolic logic, 39, 49
- Symbolic Optimal Assembly Program (SOAP), 88
- Symphony product, 186
- Syria, cyberattacks on, 361
- Systems Applications Programs (SAP)
  - Oracle lawsuit, 139
  - overview, 141–142
  - wealth from, 145
- Szilard, Leo, 49
  
- T. J. Maxx company, 224
- Tables of useful values, 9
- Tablets, GPS tracking of, 398
- Tandem Computers, 142–144
- Tarkenton, Fran, 179
- Tax Almanac application, 184
- Taylor, Jeff, 227
- TCP/IP protocol, 166, 352
- Teachers in virtual education, 315–317, 320
- Team Software Process (TSP), 256
- Technical debt metric, 284, 290
- TechSoup Global organization, 200–202
- Telecommunications, quality control in, 182
- Telephone
  - hackers, 352
  - history of, 182–183
- Telephone lines shutdown, 336
- Telephone Marketing Programs (TMP), 227–228
- Teletype keyboard, 39–40
- Teller, Edward, 49
- Teoglou, George, 140
- The Monster Board (TMB), 227
- thefacebook project, 264
- Therac 25 radiation poisoning, 334–335
  
- Thermostats, 308–309
- Thomas, Dave, 255
- Thompson, Ken, 112, 274
- Thomson, Raymond, 85–86
- ThoughtWorks company, 234–235
- “Three amigos”, 191
- Thumb drives, 254
- Thunderbird email package, 273
- Tillman, John, 49
- Time sharing, 126
- TimeLine tool, 199
- Tizard, Henry, 61
- Tizard mission, 61
- TJX Companies, 359
- Torpedo-aiming devices, 57, 62
- Torpedo Data Computer (TDC), 47, 62
- Torpedo test areas, 63
- Torpedoes, 50
- Torvalds, Linus, 210, 274–275
- TOTAL database package, 119
- Toyota automobiles safety recall, 348–349
- Training for children in ancient times, 7
- Transistor radios, 88
- Transistors
  - development of, 81–82
  - as enabling invention, 20
- Translation
  - in virtual education, 316
  - wearable computers for, 326–327
- Transmission control protocol (TCP), 352
- Transunion company, 379
- Tree-mapping, 267–268
- Trends in software, 126–127
- Treybig, Jimmy, 142
- Trilogy project failure, 342–343
- Trojans, 388–389
- Troubled Asset Relief Program (TARP), 251
- Truman, Harry S., 82
- Tschira, Klaus, 142
- Tsirigotas, Panagiotis, 300
- Tuition for virtual education, 319–320
- Tukey, John Wilder, 83

- TurboTax product, 184
- Turing, Alan
  - ACE computer, 55, 65
  - algorithm concept, 39
  - Bletchley Park, 54
  - computer description, 38
  - decryption computers, 51
- Turing award, 106–108
- Turing-complete computers, 54
- Turing machines, 38
- Turk, Charles, 114, 121
- Turner Hall company, 199
- Tutte, Bill, 49
- Twitter, 255, 278–279
- TX-0 computer, 112
  
- U-boats
  - attacks by, 58
  - U-853, 63–64
- Ubisoft company, 231
- Ubuntu operating system, 262
- Ultra program, 55
- Unified Inbox company, 310
- Unified modeling language (UML), 190–191, 212
- United States
  - cyberattack frequency and costs, 404–406
  - World War II, 59–67
- Univac company, 72, 88
- UNIVAC computer, 82, 85
- Universities. *See* Education
- University Alliance Program, 142
- University of Florida programming courses, 96–97
- University of Nalanda, 7–8
- Unix clock, 238, 341
- UNIX operating system
  - development of, 112, 121
  - start of, 160
- Unsecured networks, data theft from, 373–374
- Use-case points, 225
- Usenet services, 296
- User associations, 87
- utility software, 157–158
  
- V-1 cruise missile, 48
- V-2 rocket, 48
- Vacuum tube failures, 65
- van Benekum, Arie, 255
- Vasa* ship, 412–413
- Venture capital community, 248–250
- VeriSign company, 200, 360
- Veritas company, 199
- Verizon study on corporate theft, 372
- Vermam, Gilbert, 49
- Vertical markets, 126, 137–138, 141
- Veterans Administration hacking, 359
- Virtual bulletin boards, 317
- Virtual Education Software (VESi), 320
- Virtual machines, 236–237
- Virtual reality for education, 314–321
- Virus, term, 353
- Viruses
  - Antivirus programs. *See* Antivirus programs
  - overview, 389–390
- Visa credit cards, 110
- VisiCalc application, 161
- Visio company, 235–236
- Vision Appraisal company, 281
- Visual Basic for Applications (VBA), 381
- Visual modeling, 191
- VMware company, 236–237
- VodaPhone network, 383
- Voice-to-text tools, 316
- Volvo automobiles safety recall, 349
- von Neumann, John, 49, 55, 65
- von Neumann architecture
  - bottlenecks, 66
  - weaknesses, 399
- Voting fraud, 366–371
  
- Wales, Jimmy, 254
- Walker, Dave, 235
- Walker, J. S., 229
- Wall Street crash, 335
- Wang, An, 70
- Wang, Charles
  - bonuses, 133
  - CA founding, 132
  - wealth of, 145



- Wang, Edward, 236
- War. *See* World War II
- Ware, Willis, 81
- Warnock, John, 172
- Watches, 160
- Waterfall projects, 290
- Watergate scandal, 127
- Watson, Thomas J., Jr.
  - IBM System/360 computer, 104
  - quality improvements, 121
  - Turing award, 106
- Watson, Thomas J., Sr.
  - equal rights and equal pay, 38, 44
  - Mark I computer, 67
- Wearable computers, 325–327
- Weather prediction, 323
- “Weather Prediction by Numerical Process” (Richardson), 323
- Web browsers
  - competition, 273–274
  - development of, 208
  - hijackers, 365–366
- Web logs, 296
- Web sources, 415–421
- Webinars
  - popularity of, 180
  - virtual education, 317
- Weighing risks, 34–35
- Weinberg, Gerald M.
  - IBM lab, 121
  - influence of, 145
  - Psychology of Computer Programming*, 121, 146, 408, 414
  - structured development, 144
- Wellenreuther, Claus, 142
- Westlake, Darren, 300
- Whale phishing, 382
- Wheeler, David, 70
- Whirlwind computer, 71–72
- White hat hackers, 353
- Whitman, Meg, 219
- WikiLeaks group, 356–357
- WikiLeaks site, 302
- Wikipedia encyclopedia
  - contributors, 247
  - introduction, 254–255
  - overview, 421
  - success of, 312–313
- Wilkes, Maurice, 70
- Williams, Evan, 278
- Williams, Frederic, 69
- Williams-Kilburn tube, 69–70
- Windows operating systems
  - first versions, 166
  - Windows 3.1, 210
  - Windows 95, 211
- Wininger, Shai, 301
- Winklevoss, Cameron, 264
- Winklevoss, Tyler, 264
- Winslow, Samuel, 20
- Wireless networks, 373–374
- Wolfram, Stephen, 202–203
- Wolfram Alpha product, 202–204, 321
- Wolfram Research company, 202–204
- Wolfram System Modeler product, 202
- Women
  - Bletchley Park, 54
  - ENIAC programmers, 44
  - IBM, 38
- Wood, David, 173–174
- Word documents, macro attacks in, 381
- Word processing, 160
- Workforce in 1960s, 93–95
- World War II
  - analog computers, 46–47
  - Australia, 52–53
  - France, 52
  - Germany, 47–49
  - Great Britain, 54–59
  - innovations overview, 45–46
  - Japan, 50–51
  - overview, 43–44
  - Poland, 51–52
  - Russia, 53
  - United States, 59–67
- World-Wide Military Command and Control System (WWMCCS), 153
- World Wide Web, 208–210
- World Wide Web term, 166
- WorldCom company, 249
- Worms, 390–391

- Wozniak, Steve
  - Apple founding, 131
  - blue boxes, 352
  - entrepreneur of year, 180
  - on Prius acceleration problem, 349
- Wright, Terry, 225
- Wright amendment, 130
  
- Xerox
  - Jobs influenced by, 117
  - SAP influence, 142
- XM Radio, 232–233
  
- Y2K problem
  - legacy applications, 237–238
  - overview, 340–341
- Yahoo Real Estate service, 281
- Yamhill County, Oregon, electronic voting problems in, 368
- Yesware company, 310–311
- Young, Bob, 230
- Young, Edward, 268
- Yourdon, Edward, 146
  - DCI seminars, 179
  - Decline and Fall of the American Programmer*, 144, 414
  - structured development, 144–146
- Yourdon, Inc., 144–145
- YouTube company, 279–281
  
- Z1 computer, 38, 47
- Z2 computer, 38, 47
- Z3 computer, 47
- Zaller, Bob, 131
- Zero, 21–22
- Zero-day attacks, 380, 391–392
- Zestimate tool, 282
- Zeus botnet, 364
- Zhenfei, Ren, 181
- Zillow company, 281–283, 305
- Zimmerman, Phil, 402
- Zombie computers, 364, 391
- Zuckerberg, Mark, 255, 263
- Zuse, Konrad, 38, 47–48