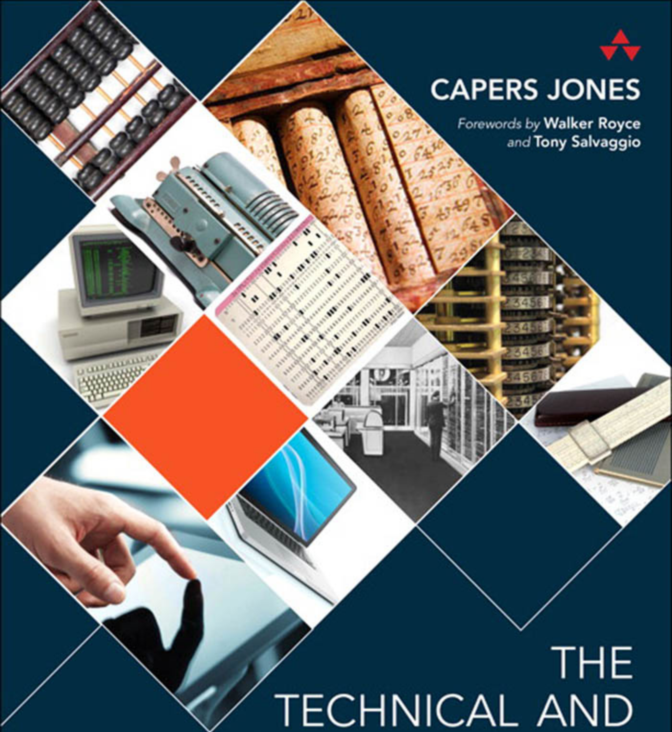

CAPERS JONES

Forewords by **Walker Royce** *and* **Tony Salvaggio**

THE TECHNICAL AND SOCIAL HISTORY OF SOFTWARE ENGINEERING

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

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

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

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 underdelivering 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 telecommunications 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 though 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. 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 Catul 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
Pebbles used for calculation	*20,000*
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
Analog computing devices	*4,000*
Algebra	4,000
Trigonometry	4,000
Fractions	4,000
Multiplication tables	3,900
Clocks: water	3,300
Decimal numbers	3,100
Abacus and mechanical calculations	*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
Antikythera mechanism	*2,200*
Astrolabe	*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
Slide rules	*575*
Measuring rates of change and acceleration	500
Mechanical calculators for addition and subtraction	*425*
Measuring power	400
Calculating trajectories	*400*
Mechanical calculators for multiplication and division	*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
Punch-card calculating devices	*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
Mechanical tabulating machines	*200*
Boolean algebra	*175*
Set theory	*150*
Fuzzy sets	*145*
Relativity	105
Measuring the strong and weak forces and gravity	100
Digital computers	*70*
Operations research	*65*
Programming languages	*65*
Sorting algorithms	*55*
Databases	*55*
Pocket calculators	*50*
Mathematical software applications	*50*
Scientific software applications	*50*
Financial software applications	*45*
Statistical software applications	*40*
Accounting software applications	*40*
Architectural and engineering applications	*40*
Graphics rendering engines for games	*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
On punched cards	*350*
Using tactile symbols such as Braille	250
On paper tape	*250*
Using cameras and film	*160*
Recording sounds	*130*
Magnetically on tape	*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
Recording in three dimensions	*75*
Magnetically on disks	*55*
Optically on disks	*50*
On solid-state devices	*35*
Using multimedia	*30*
Using digital cameras	*25*
On e-books	*25*
On smartphones	*10*
Using quantum bits	*5*
Heterogeneous databases (big data)	*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
Satellite	*60*
Subliminal signals	*50*

(Continued)

Table 1.3 *(Continued)*

Communication Channels	Approximate Number of Years Prior to 2013
Sleep learning	*40*
Ultra-low frequency sounds	*35*
Internet	*30*
Devices for the deaf (cochlear implants)	*30*
Electronic books (e-books)	*25*
Lasers	*25*
Automatic language translation	*25*
Intranet	*20*
Avatars in simulated worlds	*15*
Blogs	*12*
Webinars or podcasts	*12*
Wiki sites	*10*
Social networks	*10*
Animated multisensory methods	*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
Sequential databases of information	*65*
Random databases of information	*55*
Relational databases of information	*50*
Affinity recommendations based on past preferences	35
Web search engines for selection of keyword information	*25*
Intelligent agents for selection of relevant information	*15*
Big data analytical tools	*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
Boolean algebra	*175*
Plastics for computer cases, screens, connections, etc.	*125*
Vacuum tubes	*120*
Punched cards	*120*
CRT tubes	*80*
Von Neumann architecture	*75*
Paper tape	*75*
Integrated circuits	*70*
Transistors	*70*
Magnetic tape	*70*
High-level programming languages	*65*
Magnetic disks	*60*
Operating systems	*55*
Magnetic ink for bank checks	*55*
Magnetic stripes for credit cards	*50*
Graphics display adapters	*40*
Laser printers	*40*
Floppy disks	*40*
Dot matrix printers	*35*
Ethernet	*35*
LED displays	*30*
Ink-jet printers	*25*
Solid-state memory	*20*
Flash disks	*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 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