

James M. Kaplan

# Ten books to make you a better business technologist

Business Technology September 2015

What books does McKinsey's James Kaplan put on his must-read list for business technologists? You may be surprised.

**After going through** the slightly traumatic process last year of writing a book about the intersection of business and technology,<sup>1</sup> I started to think about which books had an impact on me as a business technologist—especially beyond the obvious ones like *The Soul of a New Machine* and *The Mythical Man-Month*.<sup>2</sup> None of the books that came to mind related to the latest technology trends (social! mobile! machine learning!)—those are important, but they change quickly. The books that really shaped my thinking provided perspectives, often historical, on the organizational, strategic, and human dimensions of business technology. I hope you enjoy them as much as I have.

<sup>1</sup> James M. Kaplan, Tucker Bailey, Derek O'Halloran, Alan Marcus, and Chris Rezek, *Beyond Cybersecurity: Protecting Your Digital Business*, Hoboken, NJ: Wiley & Sons, 2015.

<sup>2</sup> *The Soul of a New Machine*, by Tracy Kidder, describes the development of a new minicomputer in fascinating detail. Published by Little, Brown and Company in 1981, it won a Pulitzer Prize for the author in 1982. *The Mythical Man-Month*, by Frederick P. Brooks Jr. (Addison-Wesley, 1975), is a collection of essays reflecting on software engineering and project management, by the man who managed the development of IBM's System/360 computer line and OS/360 software package.

## **1. Code: The Hidden Language of Computer Hardware and Software, Charles Petzold, (Microsoft Press, 1999)**

I don't have a background in electrical engineering, and this book helped me get beneath the logical to the physical layers in the stack. Starting from first principles, and including a lot of history, Petzold explains how simple on-off switches can be combined into the mightiest of computational machines. Having read this book, you won't be able to design circuits, but you'll be able to understand how circuits get designed.

## **2. Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date, Robert X. Cringely (Addison-Wesley, 1992)**

Microsoft's acquisition of Q-DOS. IBM's decision to build the PC. Apple's growing pains. Many of the seminal events of the early personal-computer industry have been told again and again. Yes, Cringely touches on some many-told tales, but he also delves into aspects of the technology industry that few talk about. Just one example: how Microsoft applied Charles Simonyi's concept of the "metaprogrammer" to build a "software factory" that hired thousands of inexperienced computer-science majors to build the world-conquering applications of the 1980s and 1990s.

## **3. The Reckoning, David Halberstam (William Morrow, 1986)**

Is any business story more fascinating or more terrifying than the decline of US auto manufacturers in the 1960s and 1970s? By my lights, this is the best of Halberstam's many

books, far better than, say, *The Best and the Brightest*, which is mostly about people writing memos to each other. *The Reckoning* documents the rise of Nissan and the declining market share of US automakers. Why is this an important technology book? Because it provides a cautionary tale of many of the pitfalls business technologists must avoid: suspicion of new approaches, short-term decision making, managerial distance from frontline operations and distortive managerial accounting.

#### **4. *The Visual Display of Quantitative Information*, Edward Tufte (Graphics Press, 1982)**

Design know-nothings will tell you not to simplify your slides and not put too much information on any of them. Tufte points out that communicating information is the purpose of written communications, that far too many charts don't have much information in them at all, and that well-designed graphics allow us to absorb tremendous amounts of information quickly. For example, what Tufte describes as "layering" allows you to drill into the information you need. Think of a newspaper front page. You can scan all the headlines, then look at the sub-headlines for interesting articles, and then decide whether to read the first few paragraphs. Fortunately, the same type of layered structure can be used to communicate the business case for new initiative or project.

#### **5. *Military Power: Explaining Victory and Defeat in Modern Battle*, Stephen Biddle (Princeton University Press, 2005)**

Using compelling case examples, Biddle shows that force deployment—a tightly interrelated complex of cover, concealment, dispersion, suppression, small-unit independent maneuvers, combined arms, depth, reserves, and differential concentration—has been winning battles since World War I, not technological innovation or sheer mass. You can make the same case about enterprise IT. I don't think anyone would argue that IT shops with the largest budgets are the most efficient or the most effective. Likewise, any of us can list examples of companies that introduced exciting new technologies to little effect. Instead, breakthrough value in business technology seems to depend on how organizations can employ people and technology in a coherent and consistent fashion. One could even ask if there's a "modern system" for enterprise IT that combines agile, lean, service orientation, and other practices.

#### **6. *A Fiery Peace in a Cold War: Bernard Schriever and the Ultimate Weapon*, Neil Sheehan (Random House, 2009)**

I should start by saying that I think Sheehan botched the framing of this book. General Bernard Schriever may have been responsible for developing the first Intercontinental Ballistic Missile (ICBM), but he's also a cold and remote figure. It would have been more interesting to focus on the Hall brothers—Edward, the architect of America's first ICBMs, and Theodore, who betrayed secrets from the Manhattan project to the Soviet Union. That said, *A Fiery Peace in a Cold War* provides fascinating insights into how Curtis LeMay's Strategic Air Command deteriorated from

an innovative, nonhierarchical organization into a dogmatic bureaucracy and how Schriever and his team convinced the Eisenhower administration to bet on a generational leap from bombers to missiles.

#### **7. *Why the Allies Won*, Richard Overy (W. W. Norton, 1996)**

People who work with me know that I can't stand the word *governance*. It means something different to pretty much everyone, and I suspect that talking about decision-making processes provides an easy excuse for people who don't want to engage on strategic, operational, and technical content so as to avoid making decisions. Overy demonstrates that the Allies didn't just out-produce the Axis—they made better and more rational decisions using the resources they had. The British proved especially adept at running a war on a shoestring, but even the Soviet Union succeeded in getting a workable set of management processes in place. Any IT executive who has argued that his or her servers must be configured just so should read the passages comparing the how the Red Army deployed a few types of trucks and tanks at scale, even as Wehrmacht infighting resulted in a hard-to-maintain array of vehicles.

#### **8. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*, Cliff Stoll (Doubleday, 1989)**

Stoll's account of tracing a 75-cent accounting anomaly back to a KGB-funded, German spy ring introduced many of us who grew up in the 1980s to the idea of information security. It provides an invaluable reminder that protecting sensitive information depends far more on the ability to ask intelligent questions than on the latest and most loudly promoted security tools. It also provides a fascinating introduction to the process of security forensics.

#### **9. *Show Stopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft*, G. Pascal Zachary (Free Press, 1994)**

Developing a new commercial operating system is a monumental undertaking, requiring not only big technology bets, but also the coordinated effort of thousands of business analysts, technical architects, developers, testers, product managers, marketers, and others. Zachary's book provides an all-but-unique window into the mechanics of a complicated, expensive, multiyear development effort. It also recognizes that such efforts are as much human events as technological ones, with developers and managers anxious to advance their careers, terrified about making the wrong architectural choices, terrified that the code just won't work, and exhausted by hours required to shipping deadlines.

#### **10. *Racing the Beam: The Atari Video Computer System*, Nick Montfort and Ian Bogost (MIT Press, 2009)**

In today's multicore world, developing for a platform with the processing and memory limitations of an Atari 2600 is all but unimaginable. Heck, it was hard to fathom back in the 1990s when I was developing for 286s with 640Kb of main memory. Montfort and Bogost's book gets into the details

of the 2600 platform and explains the coding techniques the era's developers used to deliver exciting, compelling games. Their creativity and problem solving is an inspiration to technologists struggling to deliver compelling user experiences with today's far more advanced platforms.



Business technology is a demanding profession. Getting value from technology investments at scale requires integrating insights across business strategy, engineering, information theory, communications, operations, group psychology, and other areas. Personally, I find insights from previous generations of technology and other disciplines such as military history to be invaluable. Whether you're on the beach in the northern hemisphere or sitting by a log fire in the southern, I hope this list provides interesting reading. □

**James Kaplan** is a principal in McKinsey's New York office.