The Key to Successful Tech Management

Learning to Metabolize Failure

Clay Shirky

Late last October, the management expert Jeffrey Zients was given a mandate to fix HealthCare.gov, the website at the forefront of U.S. President Barack Obama's health-care reform, after its disastrous launch. Refusing to engage in happy talk about how well things were going or how soon everything would be fixed, Zients established performance metrics for the site's responsiveness, insisted on improvements to the underlying hardware, postponed work on nonessential features, demanded rapid reporting of significant problems, and took management oversight away from the Centers for Medicare and Medicaid Services (CMS, a federal agency within the Department of Health and Human Services) and gave it instead to a single contractor reporting to him. The result was a newly productive work environment that helped the website progress from grave dysfunction in early October to passable effectiveness two months later.

Zients' efforts demonstrated the government's ability to tackle complex technological challenges and handle them both quickly and effectively. Unfortunately for the Obama administration, the transformation came too late to rescue its reputation for technical competence. Given that the people who hired Zients clearly understood what kind of management was required to create a working online insurance marketplace, why did they wait to put in place that sort of management until the project had become an object of public ridicule? And more important, is there any way to prevent other such debacles in the future? The answers to both questions lie in the generally tortured way that the government plans and oversees technology.

CLAY SHIRKY is Professor of New Media at New York University.

THE MANAGEMENT DILEMMA

On October 1, rolling out the public face of the Affordable Care Act (ACA), his signature domestic policy initiative, Obama said this:

> Just visit HealthCare.gov, and there you can compare insurance plans, side by side, the same way you'd shop for a plane ticket on Kayak or a TV on Amazon. You enter some basic information, you'll be presented with a list of quality, affordable plans that are available in your area, with clear descriptions of what each plan covers, and what it will cost. . . . Go on the website, HealthCare.gov, check it out for yourself. And then show it to your family and your friends and help them get covered.

Anyone taking this advice discovered how far the site actually was from working like Kayak or Amazon; almost none of the people trying to sign up were able to do so. On November 14, a chastened president tried to explain how things could have gone so wrong:

> We have a pretty good track record of working with folks on technology and IT [information technology] from our campaign where, both in 2008 and 2012, we did a pretty darn good job on that. So . . . the idea that somehow we didn't have access or [weren't] interested in people's ideas, I think isn't accurate. What is true is that . . . our IT systems, how we purchase technology in the federal government is cumbersome, complicated, and outdated. . . . On my campaign, I could simply say, who are the best folks out there; let's get them around a table, let's figure out what we're doing, and we're just going to continue to improve it and refine it and work on our goals. If you're doing it at the federal government level, you're going through 40 pages of specs and this and that and the other, and there are all kinds of laws involved, and it makes it more difficult. It's part of the reason why, chronically, federal IT programs are over budget, behind schedule.

Older citizens may have been willing to let Obama off the hook, since they may regard such difficulties as par for the course—the troubled launch of Medicare Part D in 2006 generated few long-term problems for President George W. Bush. And the poor routinely have to put up with atrocious government service. But younger and middle-class Americans—crucial components of both Obama's political base and the ACA's insurance market—are used to digital systems working properly. They view clunky technology as the product of incompetence or even contempt. And the legislation's bitter opponents were lying in wait, ready to pounce on any problems that might arise. So the HealthCare.gov rollout ended up being not just a technical catastrophe but also a self-inflicted political one, an experience that may actually drive a change in the way such projects are planned and executed.

Assuming basic technical competence, the essential management challenge for all large technology projects is the same: how best to balance features, quality, and deadline. When a project cannot meet all three goals simultaneously—a situation HealthCare.gov was in by the beginning of 2013, as the administration's internal memos show—something has to give, and management's job is to decide what. In such cases, if you want certain features at a certain level of quality, you have to move the deadline. If you want overall quality by a certain deadline, you have to simplify, delay, or drop features. And if both the feature list and the deadline are fixed, quality will suffer, and you have to launch and fix after the fact. This is the worst of the three options—and the one CMS, the overmatched agency in charge, mistakenly chose.

The silver lining in the HealthCare.gov fiasco is that its high visibility may create an appetite for real improvement.

As the president noted, such snafus are hardly limited to HealthCare.gov, which was actually far from the worst government IT disaster in recent memory. That honor probably goes to the Federal Aviation Administration's Advanced Automation System, an attempt at modernizing air traffic control in the 1980s and early 1990s that has been characterized by one participant as "the greatest failure in the history of organized work." The Advanced Automation System was so famously troubled that what was then the General Accounting Office began placing any significant technical work attempted by the FAA on its "high risk" list, simply because of the reputation of the agency in charge. In the end, the FAA determined that $1.5 billion of the total $2.6 billion spent on hardware and software for the system had simply been wasted—more than twice the total cost of HealthCare.gov.

At least parts of the Advanced Automation System eventually launched, however—something that cannot be said about the FBI's Virtual Case File, a wholesale upgrade of the agency's antiquated Automated Case Support system begun in 2000. The original project was a modest, practical effort to add a Web interface to the existing Automated Case Support database. But in the aftermath of 9/11, Congress expanded the objectives and moved up the deadlines (so as to "connect the dots" among various databases as soon as possible). Mandating competing imperatives of increased scope and reduced time was obviously a recipe for trouble, but the political urgency of doing something about counterterrorism overrode practical considerations. The expanded initiative was immediately plagued by "feature creep" and poor vendor oversight, the proposed upgrade failed outright, and by 2005, the entire $170 million project had to be written off. (It is sadly ironic that the need to be seen to be doing something often interferes with actually doing something.)

These are only two of many such examples one could choose from, all stemming from problems in at least one of three distinct arenas of government tech administration: hiring and procurement, planning, and management. The silver lining in the HealthCare.gov fiasco is that its high visibility, and the political pain it inflicted, may create an appetite for real improvement.

PEOPLE AND PLANNING

The U.S. government has perennial difficulties attracting and retaining technically skilled workers and getting competitive offerings for projects from outside firms (since the complexity of bidding for federal work often limits the number of vendors that can participate in the process). The likeliest short-term impact of the botched HealthCare.gov rollout will be efforts to remedy these problems.

One proposal being considered is the technology expert Clay Johnson's RFP-EZ project, an attempt to streamline the federal request-for-proposal process so that smaller vendors (with fewer lawyers) can more easily bid for federal work. Meanwhile, the Presidential Innovation Fellows program brings people with considerable technical and managerial insight into the White House for brief "tours of duty," and there is a program to embed government workers with outside tech companies in the works. Deeper changes being discussed include allowing government agencies to evaluate and hire job candidates directly (rather than going through the months-long process required by the centralized Office of Personnel Management) and having the General Services Administration assemble a department dedicated to working on large, public-facing websites.

These are all good ideas, and anyone who wants to see an improved return on the roughly $80 billion the federal government spends

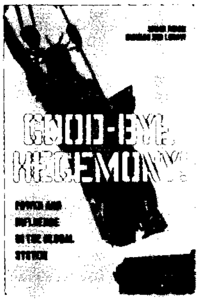
The Dollar Trap

How the U.S. Dollar Tightened Its Grip on Global Finance

Eswar S. Prasad

"At a time when the global repercussions of U.S. monetary policy are being closely examined, *The Dollar Trap* takes an authoritative look at the dollar's role in the international economy. The discussion of capital flows and the historical rise and fall of reserve currencies provides insights into the turbulent post-financial-crisis era and serves as a roadmap for thinking about the dollar's future.
A must-read for anyone interested in how the wheels of international finance spin."
—Carmen M. Reinhart, Harvard University

Cloth $35.00 978-0-691-16112-9



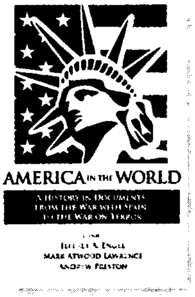
Good-Bye Hegemony!

Power and Influence in the Global System

Simon Reich & Richard Ned Lebow

"This beautifully provocative book turns the mainstream debate on the global power shift upside down. Reconceptualizing the relationship between power and influence, it refutes the widespread view that the United States must remain the world's hegemon and provides the blueprint for a more cooperative U.S. foreign policy.
A must-read for all who care about the world's future."
—Harald Mueller, executive director of the Peace Research Institute, Frankfurt

Paper $24.95 978-0-691-16043-6
Cloth $95.00 978-0-691-16042-9



America in the World

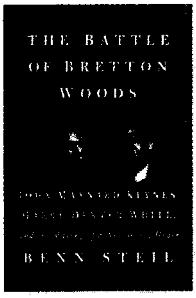
A History in Documents from the War with Spain to the War on Terror

Edited by Jeffrey A. Engel, Mark Atwood Lawrence & Andrew Preston

"A superb collection. The editors have researched hundreds of sources and selected some of the best. *America in the World* goes a long way in supplanting standard texts."
—Thomas W. Zeiler, author of *Annihilation*

"*America in the World* offers what may be the best collection of primary documents on the topic."
—Thomas Borstelmann, author of *The 1970s*

Paper $29.95 978-0-691-16175-4
Cloth $65.00 978-0-691-13335-5



The Battle of Bretton Woods

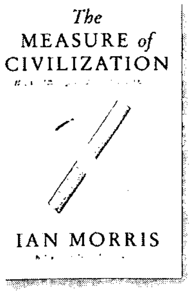
John Maynard Keynes, Harry Dexter White, and the Making of a New World Order

Benn Steil

"*The Battle of Bretton Woods* should become the gold standard on its topic. The details are addictive."
—Fred Andrews, *New York Times*

"Steil's book, engaging and entertaining, perceptive and instructive, is a triumph of economic and diplomatic history."
—Tony Barber, *Financial Times*

Paper $19.95 978-0-691-16237-9
A Council on Foreign Relations Book



The Measure of Civilization

How Social Development Decides the Fate of Nations

Ian Morris

"*The Measure of Civilization* is a terrific book—it will inform, stimulate, and challenge you. Beautifully summarizing and quantifying the major developments in energy capture, social organization, war technology, and categorization, storage, and communication of information over the last sixteen millennia, this book shows how far we have come and how this journey has been a cumulative process."
—Daron Acemoglu, coauthor of *Why Nations Fail*

Paper $19.95 978-0-691-16086-3

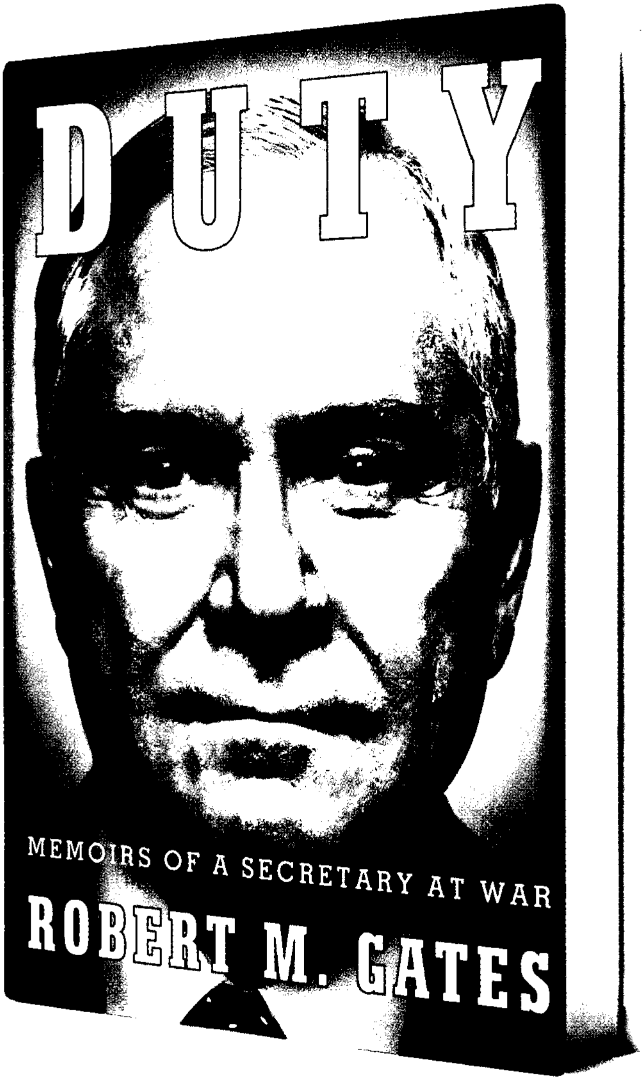
PRINCETON UNIVERSITY PRESS

See our E-Books at
press.princeton.edu

THE #1 NATIONWIDE BEST SELLER

"ONE OF THE BEST WASHINGTON MEMOIRS EVER,"

hails Thomas E. Ricks on the cover of *The New York Times Book Review.*



"Riveting...Illuminating... A breathtakingly comprehensive and ultimately unsparing examination of the modern ways of making politics, policy, and war."
—David M. Shribman, *The Boston Globe*

"Refreshingly honest and moving."
—Jack Keane, *The Wall Street Journal*

"Forthright, impassioned ...Highly revealing... Informed not only by a keen sense of historical context, but also by a long-time Washington veteran's understanding of how the levers of government work or fail to work."
—Michiko Kakutani, *The New York Times*

"A compelling memoir and a serious history... It should be read by every future defense secretary and executives of all stripes." —Fred Kaplan, *Slate*

KNOPF **RobertGatesAuthor.com**

DUTY

ROBERT M. GATES

MEMOIRS OF A SECRETARY AT WAR

annually on technology should hope they are implemented. But changes in staffing and procurement rules alone will not be enough to fix the problems. Talent is a necessary but not sufficient condition for success in tech projects; that talent also has to be deployed appropriately.

Massive, complicated undertakings are always fraught with uncertainty, and proper planning is crucial to keeping potential problems at bay. In some fields, it is possible to generate extremely detailed specifications and carefully thought-through timelines in advance, flagging known difficulties and making the project as predictable as possible. When it comes to tech projects that require the creation of novel infrastructure, however, this approach often creates more problems than it solves. The hardest challenge in creating new technology is not eliminating uncertainty in advance but adapting to it as the work uncovers it.

To understand why, it helps to visualize a tech project as two lines crossing, one representing flexibility and the other completion. On the first day of work, flexibility is at 100 percent and completion is at zero percent; on the last day, the percentages are reversed. With every decision that gets made and executed, flexibility is reduced and completion advances. The art of tech management is trading the right amounts of flexibility for the right amounts of progress at the right times. One might think that detailed advance planning would be extremely helpful in this regard, but in fact, what overly meticulous planning actually does is trade away flexibility long before it is necessary, making it harder, rather than easier, to handle unforeseen problems as they inevitably arise.

On a major new tech project, you can't really understand the challenges involved until you start trying to build it. Rigid adherence to detailed advance planning amounts to a commitment by everyone involved not to learn anything useful or surprising while doing the actual work. Worse, the illusion that an advance plan can proceed according to schedule can make it harder to catch and fixed errors as early as possible, so as to limit the damage they cause. The need to prevent errors from compounding before they are fixed puts a premium on breaking a project down into small, testable chunks, with progress and plans continuously reviewed and updated. Such a working method, often described as "agile development," is now standard in large swaths of the commercial tech industry.

The larger a tech project is and the more users it will have, the likelier it is that unexpected bugs will surface. And the longer term a technological prediction is, the likelier that it is wrong. A technology plan that tells you what will be happening next week is plausible. One that tells you what will happen next year is far less so. One that tells you what will happen in five years is largely fiction. So thinking of a tech project as something that can be implemented according to a single, fixed plan, with a product that can be delivered in a package at some fixed date long down the road, can be a recipe for disaster.

Each step of a tech project's implementation thus serves three functions. The obvious function is bringing the project further toward completion. But two other functions are also essential: any step in the implementation tests the assumptions that went into the design, and it produces new information that can and should be used to inform planning for the rest of the project. The people who want to be able to procure technology the way they would procure pencils often ignore both of those informative functions.

You can't really understand the challenges involved with any major tech project until you start trying to build it.

Unfortunately, decades of nine- and ten-figure failures have not sufficed to teach the federal government and its contractors such basic lessons. One reason is that the notion that good advance planning leads to good outcomes has deep, intuitive appeal. The program that put a man on the moon, for example, is often cited as a model for how the government can engage in a long burst of technically excellent work, have that work progress in a straightforward way for years on end, and then see it culminate in a stunning success.

In fact, however, the moon landings succeeded because they followed a far more circuitous path. NASA worked on the project in careful iterations, conducting a huge number of tests along the way—many of which failed and forced changes in engineering. The tower of the rocket called Little Joe 1 ignited prematurely, taking the spacecraft with it. Little Joe 5 suffered the same problem. Mercury-Atlas 1 collapsed and exploded during launch. Mercury-Atlas 3 did not go into orbit, and its mission was aborted. The guidance system of Mercury-Scout 1 malfunctioned, and its mission, too, was aborted. And so on. And those were just the failures of unmanned spacecraft. In 1967, a capsule fire in Apollo 1 killed three astronauts, the worst disaster in NASA history up to that point. A congressional investigation into the accident found "deficiencies existed in Command Module design, workmanship, and quality control." People were fired, processes were revamped, and later work took that failure into account.

NASA didn't figure out how to put a man on the moon in one long, early burst of brilliant planning; it did so by working in discrete, testable steps. Many of those steps were partial or total failures, which informed later work. In digital technology, such an incremental, experimental approach is called "test-driven development." It has become standard practice in the field, but it was not used for HealthCare.gov. Tests on that site were late and desultory, and even when they revealed problems, little was changed.

EMBRACING FAILURE

The toughest nut to crack is project management. Given that the administration didn't put competent management in place early on, it is no surprise that the HealthCare.gov launch failed. What is surprising is that as late as the launch day, people at the highest levels of government seem to have been deluded into thinking it would be successful. The president discussed this failure in November: "I was not informed directly that the website would not be working the way it was supposed to. Had I been informed, I wouldn't be going out saying, 'Boy, this is going to be great.' I'm accused of a lot of things, but I don't think I'm stupid enough to go around saying, 'This is going to be like shopping on Amazon or Travelocity,' a week before the website opens if I thought that it wasn't going to work." The president's staff, in other words, not only allowed Obama to embarrass himself by making unsupportable claims; they also helped him make the situation worse, by driving extra traffic and attention to a barely functioning site.

Although many Obama supporters dispute the comparison, the HealthCare.gov launch resembles the performance of the Federal Emergency Management Agency, or FEMA, in New Orleans during Hurricane Katrina in one key respect: the failures in oversight and communication were presided over by senior administration officials. The inability of these political appointees to know or admit that the launch was doomed indicates that the managerial failure was worse than the technical failure. (And indeed, as the progress under Zients demonstrated, the core problems involved not the competence of the programmers but the competence of their bosses.)

Managers cannot manage when they don't understand what is happening and are not willing to hear bad news and make unpleasant choices. There has been much speculation about just who hid the truth about the website's problems, but reading the communications trail from the month before the site launched, the answer seems to be almost everyone. Because the government has not regarded the development of new technology as a primary function, technical managers tend to answer to nontechnical managers at every level of the bureaucracy, which in this case obscured the technical bad news without there being any one person who decided to do so.

The technical leadership on HealthCare.gov did not answer to the chief information officer of CMS, the chief information officer of CMS did not answer to the chief information officer of the Department of Health and Human Services, and the chief information officer of Health and Human Services did not answer to the chief information officer of the federal government. Instead, each reported to a nontechnical bureaucrat or political appointee, and during the long game of telephone, key details of the story kept getting stripped out or distorted. (When you are trying to describe the performance of a database under various sorts of load, you need both speaker and listener to understand database engineering.)

Who hid the truth about the website's problems? The answer seems to be almost everyone.

Given this sort of organizational chart, it hardly required outright deception or malicious withholding of information to keep accurate information from moving up the chain of command in a timely fashion. Without improvements to transparency and communication, all the procurement reform and agile development in the world will have only a small impact on improving future government IT projects.

The biggest challenge in raising the level of federal management of technical work will be changing managers' incentive structure. Creating financial and career penalties for failure seems like the obvious approach, but this would actually make things worse. All major technological work involves trying new things, trying new things always involves failures, and those failures can often be extremely useful learning opportunities. So creating penalties for failure would actually create penalties for learning and would ensure that workers never tried anything new or interesting.

Instead of failure, what should be penalized are opacity and information hoarding, which are far greater sins. The way to deal with failure is to break it up into small, rapidly metabolizable doses, none of which would be fatal to the project as a whole. But in order for that to happen, managers need to know about problems in great detail and in real time—something that the government's work environment rarely encourages. As one observer of the government's technical culture put it to me, "There are two ways to answer the question, 'How is it going?' One way is to offer an honest assessment of the overall project. The other is to say, 'Everyone is doing what they said they would do.' Everyone in government wants to offer the latter answer and pretend it's the former." Until that changes, government tech failures will be routine.

BEYOND CRISIS MANAGEMENT

The most depressing aspect of the post-launch turnaround of HealthCare.gov is that the management methods Zients used—establishing clear chains of responsibility; demanding rapid, honest reporting of problems; and being willing to make difficult but necessary choices about cutting or delaying features—were highly unlikely to have been adopted by the government until after the project had already visibly and publicly failed. At that point, having chosen not to learn early, in private, the administration ended up learning late, in public.

In October, the site was up but not really running. Only a tiny fraction of potential users could try the service, and those users generated concrete errors. Those errors, in turn, were handed to a team whose job was to fix things. Improvements were incremental, put in place over a period of months. Bug reports were attacked in order of importance, rather than time of discovery. Features were prioritized, and some were dropped. The result has been what is known in the tech world as a phased rollout—just one conducted in the most visible and politically damaging way conceivable.

Substitute David Petraeus and the Iraq war for Zients and HealthCare.gov, and the story is the same, and other examples are easy to find. So the real question is not how to fix a website, even a big, complicated one. It is whether Washington will ever allow good management to become part of its standard operating procedures, rather than something that it turns to only when its regular routines fail badly enough to produce a crisis.