Some of our Annals issues coalesce around a theme, sometimes intentionally, sometimes by chance. Other issues simply contain whatever smart and well-researched topics that our community of historians happen to be working on lately. This is one of the latter type of issues.

The history of the IBM Corporation has been well-documented by historians of computing, at least in terms of its operations in the United States. One might imagine that there are few unexplored paths to be explored within this well-traveled historical territory. But in fact the global history of IBM, by which I mean both the history of IBM operations in other parts of the world and the history of IBM as an international and transnational institution, is only just beginning to be developed. In the past decade, the Annals has published novel studies of IBM in the context of global history, from its savvy cultivation of a European-wide manufacturing and research network (including a two-piece series on the IBM La Gaude Laboratory in France), to the establishment of IBM Chile, to its role in the appropriation and “Americanization” of the history of computing itself. 1

In this issue, Jim Cortada describes in his article “IBM Branch Offices: What They Were, How They Worked, 1920s–1980s,” how IBM developed and maintained its global sales and service network, which included more than 800 of these offices in 170 countries. These offices served as a home away from home for a widely distributed workforce, replicating in foreign territories a familiar and supportive environment as well as the provider of practical support and services. Together these branch offices comprised a transnational national network across which information, resources, and personnel could freely flow. These offices also served as the public face of IBM, providing the local knowledge and maintaining the personal relationships essential for a far-flung global sales and service operation. For nearly a century, Cortada argues, “branch offices were the key channels for engaging with customers”; in fact, for many of these customers, these branch offices were the only institutional face of IBM that they ever encountered.

In “Increasing the Yield: Nuclear Testing, Weapons Strategy, and Supercomputer Selection at Los Alamos in the 1960s,” Nicholas Lewis explores two of the most critical decisions facing any organization reliant on computing resources, namely, how and when to upgrade. Lewis describes the combination of changes in weapons strategy, weapons design, and weapons testing policies that created significant pressures for change at Los Alamos, leading to a massive expansion of its computing facilities. The process of selecting a new supercomputer “provides a window onto a variety of forces and drives that influenced the form and use of computing at a weapons laboratory during the Cold War,” argues Lewis: “Far from a simple matter of selecting machines based upon performance or price, committee members brought to the process interests and concerns for the consequences their decisions might have for the future of the Lab and its mission.”

Finally, in his article on “Sixty Years of Software Development Life Cycle Models,” Ralf Kneuper explores the history of software development practices from the ENIAC to the present. Of particular significance is the introduction in 1956 by Herbert Benington of the notion of the “software lifecycle.” This lifecycle consisted of a series of discrete phases, according to Benington, beginning with the development of an operational plan and concluding with a process of system evaluation. Although the specific elements of Benington’s life cycle would be altered and disputed by later adopters, the general principle of the lifecycle proved durable. In the 1970s work by Royce, Bell, Thayer and most famously Barry Boehm, established the “waterfall model” as one of the most influential (and later derided) versions of the software lifecycle. These “heavy weight” models were particularly appealing to organizations developing large, complex systems. By the early 1990s, increased demand for smaller, more flexible software systems encouraged the development of more iterative processes that could respond quickly to changing market and user requirements. These newer, more agile methods (including, of course, the method called simply “Agile”), adopted a model of the software lifecycle that bore only a scant relationship to the sequential models outlined by Benington. Nevertheless, these earlier models continue to persist in the development of large-scale software systems.

References and Notes

Nathan Ensmenger is editor in chief of *IEEE Annals* and an associate professor in the School of Informatics and Computing at Indiana University. Contact him at nensmeng@indiana.edu.