Understanding the Next Production (Digitally Based) Revolution and its Ties to Software-Defined Infrastructure

Robert B. Cohen, Economic Strategy Institute, January 22, 2017

Paper draft – awaiting permission of authors to use several unpublished quotes

Summary

Software is the driving force behind the economy. Over the past five years, the creation of software has accelerated through innovations described here in detail, such as continuous service delivery, DevOps, and containers. In addition, the centrality of software in the management of data centers has facilitated the growth of Big Data and data analysis. Software has also been critical in the transformation of infrastructure, with many firms moving to data center and computing infrastructures that rely far less on hardware than previously. The new, software-defined infrastructure offers much more agility, scalability and interoperability. These changes are quite remarkable, particularly since a few years ago, only a few prominent firms, such as Facebook, Amazon, Netflix and Google, were among a small group that pioneered the extensive use of software in their core operations.

This paper makes five points:

1. Software innovations have become central to business operations. The cost of obtaining high performance software, such as Open Source Software, is nearly zero if a firm uses an online exchange such as GitHub and has access to a permissive license. This speeds up software development and reduces its cost because firms can often avoid paying licensing costs.

2. Software innovations are employed in such a wide range of industries that software should be a General-Purpose Technology.

3. Software behaves unlike other products. It is part of a continuing development process with developers adding code as needed. This ongoing process can be seen in new Internet-related software, such as Linux, Open Source, and Docker/containers.

4. The time it takes to complete software innovations is dropping sharply. This should change our perspective on how long innovation takes. It should also revise our ideas about how rapidly software changes can be deployed. Advances such as continuous service deployment, DevOps, containers/Docker, have created a situation where “software is eating the world.”

5. Complementary improvements in data analytics and software-defined infrastructure enabled businesses to take swift advantage of software innovations. They are using them to change processes and products. The result is more sophisticated control over design, testing and production processes, as well as supply chains.

Today, a wide range of enterprises depends software and data analytics to operate and create new strategies, a major change. Innovations in software, including “achieving higher IT and organizational performance is a team effort spanning development and operations”¹ They have resulted in a 6-fold increase in the number of times a high-performing firm deploys new code as compared to slow performing

¹ Puppet Labs and DevOps Research and Assessment, “2016 State of DevOps Report,” p.4
firms. Between 2015 and 2016, higher-performing firms increased code deploys per year from 200 to 1460 while slow-performing firms remained at 2 to 12 deploys per year.\(^2\)

This essay describes the fundamental changes in software development that have occurred. It also explores Big Data and data analytics. These advances are often facilitated by the move to software-defined infrastructure. This more agile infrastructure makes it easier to implement data analytics and more sophisticated predictive analytics. These applications help firms manage their operations more efficiently, increase their productivity and reduce costs. They have helped businesses change processes as well as production. As we argue below, these innovations are overlooked by some analysts who are trying to predict the future.

These innovations extend Internet technology. Internet principles and practices – interoperability, extensibility, and scalability – have been shaping how firms now create, deploy and utilize software. These principles provide firms with greater agility and rapidity. Analytics. Innovative programming processes, that build upon DevOps rely upon three processes: flow; feedback; and continual learning by experimentation.\(^3\) Containers are an important advance that supports DevOps. DevOps’ processes reconstruct compartmentalized approaches to how firms develop and use software and analyze data. This is discussed in greater detail below.

The second section of this essay addresses the argument made by several economists, particularly Robert Gordon, that there is very little likelihood that the Internet will provide much in the way of innovations after 2004. We review the basis for Gordon’s conclusion. We will also examine the views of other economists who don’t believe that the current software innovation process is different from previous instances of applications innovations in the information and communications technology industries (ICT).

A. Software as a General-Purpose Technology (GPT)

Economists who have studied the great rise in US productivity during the 20\(^{th}\) Century have identified several technologies as the key factors behind the upswing from the 1920s to 1970s. “GPT’s are characterized by pervasiveness (they are used as inputs by many downstream sectors), inherent potential for technical improvements, and innovational complementarities’, meaning that the productivity of R&D in downstream sectors increases because of innovation in the GPT. Thus, as GPT’s improve they spread throughout the economy, bringing about generalized productivity gains.”\(^4\)

\(^2\) Puppet Labs and DevOps Research and Assessment, pp. 15 and 18.
\(^3\) The author thanks Chris Swan for helping him refine the definition of DevOps.
General Purpose Technologies\textsuperscript{5} include the internal combustion engine and electricity. Because firms in a wide range of industries exploited these technologies in a myriad of ways, they were defined as “an invention that can lead to many sub-inventions.”\textsuperscript{6}

Most inventions that are considered GPTs unleashed the great ascent of US manufacturing and, thereby, the US economy in the 1920s, 1930s and 1940s. For these technologies, the largest economic impact occurred as industries adopted them and developed innovative ways to use them more efficiently.

While many economists did not find evidence to describe semiconductors or other inventions as GPTs, we believe that software should be considered as an emerging GPT. The following chart compares software to internal combustion engines and electricity, two well-established GPTs.


\textsuperscript{6} Gordon, p. 555.
In the case of the digital transformation, we see clear parallels with previous GPTs. The use of big data as well as advanced analytics that are included in the Internet of Things are fundamental to the transition to the digital world. These innovations have permitted firms like Facebook, Netflix, ETSY, Ford, Boeing, UPS, and John Deere to enhance how they produce products or services as well as manage their supply chains. To illustrate the scope of such benefits, we estimated that several firms already well on their way to adopting a wide range of software innovations already benefit from them. We examined public 10-K reports to estimate changes in output or revenues per employee, a very general estimate of productivity. We found that Facebook improved its productivity by 46 percent from 2010 to 2015 and ETSY improved its productivity by 17 percent from 2014 to 2015.
Figure 1. Estimates of Productivity Changes at Specific Firms, Selected Years

<table>
<thead>
<tr>
<th>Firm</th>
<th>Percentage Change</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Economy</td>
<td>10-20</td>
<td>10-K reports filed with the Securities and Exchange Commission, various years.</td>
</tr>
<tr>
<td>US Economy</td>
<td>10-20</td>
<td></td>
</tr>
<tr>
<td>Boeing</td>
<td>10-20</td>
<td></td>
</tr>
<tr>
<td>ETSY</td>
<td>20-40</td>
<td></td>
</tr>
<tr>
<td>Facebook</td>
<td>40-50</td>
<td></td>
</tr>
<tr>
<td>Netflix</td>
<td>40-50</td>
<td></td>
</tr>
</tbody>
</table>

Source: 10-K reports filed with the Securities and Exchange Commission, various years.

B. Software and the Digital Transformation

Firms have made software central to how they operate. The growing importance of software was highlighted by Marc Andreessen’s statement that “software is eating the world.” Andreessen believes that “six decades into the computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale.”  

This section explores Andreessen’s contention in greater detail. Our goal is to illustrate how innovations in software build upon fundamental enhancements to Internet technology. We argue that by becoming more innovation-focused, firms have shifted their approach to information technology. They now place much greater value on the benefits they obtain by deploying new software tools. Their success is facilitated by new software-defined infrastructure.

Many enterprises now depend upon software and data analytics to operate and create new strategies. Innovations in software, including “achieving higher IT and organizational performance is a team effort spanning development and operations” have resulted in a 6-fold increase in the number of times a high-performing firm deploys new code as compared to slow performing firms. Between 2015 and 2016, higher-performing firms increased deploys per year from 200 to 1460 while slow-performing firms remained at 2 to 12 deploys per year.

---

8 Puppet Labs and DevOps Research and Assessment, “2016 State of DevOps Report,” p.4
9 Puppet Labs and DevOps Research and Assessment, pp. 15 and 18.
This essay argues that a shift to data- and software-centrality is transforming business. Innovations linked to the Internet and data, rather than being “focused on the entertainment and information and communications technology”\(^\text{10}\) are transforming industries.

1. The New Process of Software Development

The following sections illustrate how firms have changed the process of software development. These innovations build upon enhancements represented in software-based infrastructure. In many cases, they employ technological innovations from the 1980s, such as containers, that have only been deployed with full features in the past few years.\(^\text{11}\)

We describe a series of software innovations in some detail. We do this because economic analysis of the digital economy has not advanced to the point where innovations in software are measured directly. Most economists rely upon indirect measurements of prices and performance for laptops, tablets and mobile phones to estimate the impact of the digital transformation. This often overlooks innovations in software and in our data and computing infrastructure.\(^\text{12}\)

Software development now proceeds in ways that are very different than was true in 2005 to 2010. Developers working on new software, such as Linux, have made continuous refinements to the original code. If we examine the changes that have been added to the code for the Linux operating system, we can see an ongoing pattern of improvements (see Figure 1)

\[\text{Figure 2. Commits per month in the Linux Source Code Management Repository, 1991-2011}\]

---


Facebook’s growth in the use of coding software, Figure 3, is characterized by a rapid growth in the number of software developers it has hired as well as rapid growth in the number of “commits,” committing code to a versioning system. At Facebook, as at many other “Internet Firms,” engineers write code in a “perpetual development mode, in which engineers continuously develop new features and make them available to users. Consequently, the system also grows continuously, possibly at a super-linear rate.”.  

At Facebook, “engineers commit code to the version control system up to 500 times a day, recording changes in some 3,000 files.” In addition, Facebook’s lines of code grew rapidly from 2005 to 2012, growing to 10.5 million lines of code. Facebook is known for its enormous live experimentation using A/B testing.

Figure 3. Increase in Coding Activity at Facebook through 2012

Source: Dror G. Feitelson, Eitan Frachtenberg, and Kent L. Beck, “Development and Deployment at Facebook,” p. 2. [https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf](https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf)

Figure 4. Timescales of Making new developments available. At Facebook, new code is deployed every day. It balances rapid development with foresight and monitoring

<table>
<thead>
<tr>
<th>Waterfall or Unified Process</th>
<th>Evolutionary Development</th>
<th>Agile Development</th>
<th>Facebook</th>
<th>Continuous Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Once</td>
<td>Months</td>
<td>Weeks</td>
<td>One day</td>
<td>&lt; Hour</td>
</tr>
</tbody>
</table>

Source: Dror G. Feitelson, Eitan Frachtenberg, and Kent L. Beck, “Development and Deployment at Facebook,” p. 3. [https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf](https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf)

---

13 Dror G. Feitelson, Eitan Frachtenberg, and Kent L. Beck, “Development and Deployment at Facebook,” p.2. [https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf](https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf)  
14 Dror G. Feitelson, Eitan Frachtenberg, and Kent L. Beck, “Development and Deployment at Facebook,” [https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf](https://pdfs.semanticscholar.org/566c/3ad271fcea439a4dfcc5b7aa388f6021d110.pdf)  
15 Feitelson, Frachtenberg, and Beck, p. 3.
During three years of surveys analyzing how firms were using DevOps, higher performing firms\textsuperscript{16} began to achieve much better performance than low performing ones. The ability to deploy new code is one indicator of this difference. Between 2014 and 2016, higher-performing firms increased their ability to deploy software from 200 deployments per year to 1460 deployments per year. This occurred while low-performing firms maintained their level of about 12 deployments per year.\textsuperscript{17}

**Figure 5. Deployment Frequency in Number of Deploys per Year**

![Deployment Frequency Graph](image)

Source: Puppet and Dora, p. 18.

\textbf{a. Continuous Service Delivery and DevOps}

Continuous delivery or continuous deployment (CD) describes how teams produce software in short cycles. Since the software is tested continuously, it can be confidently released at any time in the cycle. This speeds up the building, testing and releasing of new software. Continuous integration is the process where developers integrate code they are writing into a shared repository many times a day. This lets developers check new code at each stage as it is written. This checking process permits teams of developers to detect problems in their code early. If they find the new code is error-free, their work can proceed to an automated build of a new software application. Many organizations achieve CD by connecting CI to an automated infrastructure, such as the cloud or software-defined infrastructure.

Consequently, firms develop software applications and employ them in their operations using different processes than the highly-segregated steps they used earlier (see Cockcroft’s chart in Figure 1). The new process eliminates time-consuming, step-by-step approvals. This process is still employed by many low-performing IT organizations; there, each stage in software development is isolated from every other. As noted above, the shift to CD is built upon changes in software-defined infrastructure and new process

\textsuperscript{16} For a definition of higher performing and low performing, see the Appendix.
\textsuperscript{17} Puppet and Dora, p. 18.
innovations built on top of it. The result is an enormous reduction in the time required to develop new software as well as to test, evaluate and deploy it in “real-life” situations. It also means that there is continuous learning about the new code, or software, that is being written, so that mistakes are identified before they are introduced to new designs.

The innovations illustrated in this section provide a sharp contrast to the concept of co-innovation of new ICT discoveries. Some economists employ this framework to assess changes in the ICT industries.

CD or continuous service delivery relies upon cross-disciplinary teams to program, deploy and test new software (see Cockcroft’s diagrams below, Figure 6). In continuous service delivery, integrated design teams replace siloed, or isolated, “skill areas,” like quality assurance, systems administration and development. This reduces the number of steps required to write and test software. Continuous service delivery lets firms respond to demands from markets and customers. By creating software in this way, firms not only improve their knowledge of customers and markets, but also quickly exploit new opportunities for sales.

Lori Beer, the CIO of JPMorganChase, has noted that while her bank achieved significant savings by moving to the cloud and creating a more software-based infrastructure, the real benefits of innovations in software were derived from the gain in productivity achieved by expanding business opportunities. By being able to analyze markets more rapidly and to create new software to address them, the bank expanded its revenues.

---


DevOps is an integral part of continuous service delivery (see Figure 7). Ops (operations) groups let developers manage the “operational characteristics” of applications they are building. In addition, developers must be mindful of operational considerations when they build applications. In DevOps, a “Shift Left” summarizes how the software development moves to developers but requires them to consider carefully how new software contributes to specific operations.
This process innovation has had a significant impact on business. A 2015 survey concluded that high-performing IT organizations deployed new software 30 times more frequently than low performing ones. In addition, high-performing IT organizations had 200 times shorter lead times, 60 times fewer failures and recovered 168 times faster.\textsuperscript{20} In addition, there has been an increase in the number of employees working on DevOps, with an increase from 16 percent in 2014 to 22 percent in 2016.\textsuperscript{21}

Figure 8. DevOps and the “Shift Left” in Building Applications-Aware Environments


Microservices are “set of practices, and tools that increases an organization’s ability to deliver applications and services at high velocity.”

They are also an important feature of continuous service delivery. Microservices divide “some types of applications” into smaller, composable pieces. These pieces work together very much like Lego blocks. So microservices’ components are “easier to build and maintain.” Each component, “is developed separately. An application is the sum of its “constituent components.”

This differs from how programmers previously developed software. In the past, programmers had to write software as a single, unified product.

Microservices can be easily “glued together” because they each contain “an [Application Protocol Interface or] API endpoint.” This is a programming interface that can be accessed like a “standard webpage.” This method for accessing microservices makes it easy for developers to consume them as required.

The infrastructure that supports this change in software development allows developers to focus on code, the basic instructions that software engineers and developers utilize. “As we move towards software-defined environments, we [can] build, version [or model] and manage complex environments, all as code.”

Open Source components are software whose original source code is freely available on locations such as GitHub. Free distribution and modification relies upon having a permissive license in place, such as Apache 2.0. By using these components, software developers, core developers who write the Open Source code and other programmers refine the code that is written and identify the flaws in it. The availability of Open Source permits many businesses avoid paying the licensing costs that vendors apply to software they have developed.

Open Source Software is often more reliable than vendor-developed software. Large numbers of programmers are involved in its creation. Open Source software frees firms from their previous dependence on vendors that built applications. Firms, such as JPMorganChase use a great deal of Open Source software. They also contribute to the software available through GitHub. Microservices can change this dependence on vendors for important applications that firms can now obtain as Open Source code.

Open Source components have achieved wide acceptance in the business community. They are being used by many large organizations, including major banks. They make it easier to create new software. This reduces costs.

---

24 “What are microservices?”
b. Containers, Docker and Facilitating the Creation and Deployment of New Software

Containers (see Figures 3 and 4) began as an effort to create finer-grain controls to “develop software faster, cheaper, and operate at a scale never seen before.” Containers represent a fundamental change in how workloads and applications can be virtualized. Containers can scale more efficiently, operate faster and offer greater portability than hardware virtualization. Eventually, they are expected to replace most instances where virtual machines are involved.28

Google’s container was “a kind of virtualized, simplified OS [Operating System] which we used to power all of Google’s applications.” Initially, Google developed cgroups,29 “a framework pattern that provides encapsulation and separation of concerns for the components that use them …. the container will provide mechanisms to address cross-cutting concerns like security or transaction management….a container wraps the component.”30

Containers, in contrast to virtual machines, offer:

- “Simple deployment: By packaging your application as a singularly addressable, registry-stored, one-command-line deployable component, a container radically simplifies the deployment of your app no matter where you’re deploying it.
- Rapid availability: By abstracting just the OS [operating system] rather than the whole physical computer, this package can “boot” in ~1/20th of a second compared to a minute or so for a modern VM.
- Leverage microservices: Containers allow developers and operators to further subdivide compute resources.”

---

Figure 9. Comparing Containers to Virtual Machines (VMs)

Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries


Figure 10. Applications on Virtual Machines and Containers

Why are Docker containers lightweight?

Original App
(No OS to take up space, resources, or require restart)
Copy of App
No OS. Can Share bins/libs
Modified App
Union file system allows us to only save the diffs between container A and container A'
Based on these benefits,

- “a developer has in their laptop plenty of compute power to run multiple containers, making for easier and faster development”
- “a single command” can push out “new version of a container,”
- “with applications using open source software,” the composability of application systems” is greatly enhanced. This means that developers can bring together many tools that might be complicated to set up individually, such as the big data tools, Hadoop and MongoDB® numerous tools can be deployed on a single computer and used to improve the quality of the software that can be programmed.\(^\text{31}\)

An April-May 2016 DevOps.com and ClusterHQ survey\(^\text{32}\) found that 79 percent of respondents’ organizations were using container technologies. Of this group, 76% of the deployments were running in production environments, not experimental ones. This was a significant increase over 2015, when only 38 percent of respondents had containers in production ecosystems. The report concluded that container adoption was driven by a desire to “increase developer efficiency (39 percent) and support microservices (36 percent).” Over two thirds of the survey’s respondents said their firms are realizing the results they expected from using containers.

Bloomberg Inc. has adopted containers and software-defined networking over the last four years to add simplicity and high volume to its development of new applications and products. It has assembled a staff of 2500 developers and embraced the use of Open Stack. It notes that modern applications (software) has become ephemeral in nature with developers using templated images and automated images to write software. This has moved Bloomberg away from a model where applications development required complex policies. The move to software-defined networking has also let developers use microservices and micro-segmentation of applications.\(^\text{33}\)

Docker is an interoperable format for containerized applications that builds upon operating system level virtualization. Operating System “containers wrap-up an application in a self-contained filesystem and that includes everything the [application] needs to run independently: binaries, runtime libraries, system tools, system packages, etc. This level of simplification and compartmentalization allows applications to be spun up [or launched] much faster than before.”\(^\text{34}\) while ensuring consistent and predictive up time.


\(^{33}\) Truman Boyes, talk on Bloomberg Inc. and Software-Defined Networking, Open Network Users Group, Columbia University, May 2015.

c. Big Data and Data Analytics

McKinsey describes Big Data as “large pools of data that can be captured, communicated, aggregated, stored, and analyzed.” In 2011, McKinsey found that Big Data is “part of every sector and function of the global economy.”35 In 2013, ABI Research estimated that spending on Big Data was $31 billion and that this spending would increase to $114 billion in 2018, a compound growth rate of 29.6 percent.36 The Eckerson Group has concluded that if firms are going to capitalize on Big Data, they need to “fundamentally rethink the way they capture, store, govern, transform and analyze” it.37

Two different McKinsey Global Institute studies illustrate how much firms have increased their dependence on data analytics. In its first Big Data study,38 McKinsey found that firms in nearly all U.S. sectors had at least 200 terabytes of stored data per company; that is in firms having more than 10,000 employees in 2009, a level where McKinsey believed firms could capture real value from data analytics.39 This report also estimated that the number of employees in data analytics positions was 300,000. It forecast that by 2018, demand for these jobs would rise to between 440,000 to 480,000.40

McKinsey estimated that firms could obtain significant savings when they implemented data analytics. It estimated that when retailing uses data analytics “marketing levers can affect 10 to 30 percent of [the] operating margin; merchandising levers can affect 10 to 40 percent; and supply chain levers can have a 5 to 35 percent impact.”41

To provide one illustration of the impact of data analytics, nearly four-fifths of firms in one survey reported that it helped them institute new business processes, such as creating an Internet of things. In the same survey, nearly one-third or more of firms in the energy and utilities, automotive and retailing industries had adopted machine-to-machine (M2M) communications. More than a quarter of firms in healthcare and consumer electronics as well as one-sixth or more of firms in manufacturing and transport and logistics had also adopted these communications.

McKinsey noted that Big Data could contribute real value very differently depending upon the type of retail sector that used it. The retailing sectors that were early adopters of data analytics usually obtained the greatest benefits. General merchandise stores, building material and garden, electronics and appliances and health and personal care stores were forecast to have the greatest big data value potential.

Figure 11. The Difference in Big Data Value Potential in Retailing Subsectors


43 Brenneis, p. 19.
McKinsey’s later Big Data report concluded that “data is now a critical corporate asset.”\textsuperscript{44} While it found that data is doubling every three years, it also noted that since its earlier study, many firms have not taken advantage of the gains that it had forecast in 2011. Some observers\textsuperscript{45} believe that many firms probably had a difficult time figuring out how to improve the flow of analysis. They also did not create a framework of continuous learning by experimentation, so that the use of Big Data could be refined. Nonetheless, McKinsey finds that a group of “analytics leaders are changing the nature of competition and consolidating big advantages,”\textsuperscript{46} by deploying and using Big Data. These firms include “Apple, Alphabet/Google, Amazon, Facebook, Microsoft, GE, and Alibaba Group.”\textsuperscript{47} One reason for this shift is that these firms are exploiting a wide difference between their level of performance and that of other firms. In many cases, these firms are using data to provide better situational awareness and then employing analytics to improve the situations they experience.

C. Economists and Software Innovations

Several economists have described their vision for the future of economic growth and its relationship to innovations in information and communications technologies (ICT). We review of several of these visions here to illustrate how economists have not invested an effort in understanding the role of software innovation in the economy. This review draws links between the previous section where we identify the structure of some of the more fundamental software innovations and the views presented in several recent economic publications.

One of the more notable points of view is Robert Gordon’s. The main summary of Gordon’s recent book\textsuperscript{48} is that important innovations, such as electrification, the airplane, and the refrigerator, were already widespread by the 1970s and were invented during the Second Industrial Revolution. Therefore, Gordon concludes, there are likely to be fewer innovations in the future, particularly since the chances that a widely used, revolutionary technology would be found, is unlikely. While carefully identifying General Purpose Technologies – technologies that were adopted widely and adapted to the requirements in a broad range of industries during the Second Industrial Revolution -- Gordon does not attempt to consider whether software might play a similar role in the Third Industrial Revolution (as we do below). Consequently, for Gordon, there is likely to be lower US productivity growth and lower levels of US GDP growth.

Gordon concludes that innovations based on the Internet largely ended in 2004. He carefully details how the US experienced a special century, or Second Industrial Revolution, from 1870 to 1970. The key identifying characteristic of the special century was that firms adopted important new technologies and refined them so that they could achieve significant increases in productivity growth. These advances in productivity and GDP peaked during the 1940s and continued through the early 1970s. Gordon dismisses

\textsuperscript{44} Nicolaus Henke, “The Age of Analytics: Competing in a data-driven world,” McKinsey Global Institute, December 2016, p. vi.

\textsuperscript{45} The author thanks Chris Swan of CSC for his comments on why more firms did not take advantage of Big Data.

\textsuperscript{46} Henke, p. 5.

\textsuperscript{47} Henke, p.

the notion that the Third Industrial Revolution (IR#3), the Internet Revolution, associated with computers and digitization, will have a similar effect. He finds that “business practices in the office, the retail sector, and in the banking and financial sector ... current methods of production had been largely achieved by 2004.”

In discussing Gordon’s book below, we identify ways that IR#3 is beginning to have real impacts in businesses rather than just consumer markets. We connect many of these impacts to software innovations. These indicate there are likely to be opportunities for improvements in productivity and cost savings in IR#3.

Although Gordon finds that innovation has continued since 1970, he concludes that it has not achieved the same rate of growth that it did earlier. This would slight economic growth. Gordon argues that recent innovations are more limited in the scope of industries where they are adopted. He finds that the main advances have “focused on entertainment and information and communications technology (ICT).” Gordon’s review finds that the growth of total factor productivity (TFP) in the US declined after 2004, with productivity growth growing half as fast from 2004 to 2014 compared to 1994 and 2004. In part, he finds that this is due to the slow transformation of business processes; by 2004, the productivity of office workers attained the level where it would be in 2014.

Gordon also buttresses his argument by asserting that IR#3 has impacted only a few key industries. He notes that “the main impact on retail productivity growth of big-box [retailing] stores ... largely occurred a decade ago.” He cites the decline in stock trading after the financial crisis as evidence of a “plateau of activity in finance and banking.” In reviewing the home and consumer electronics industries, Gordon concludes that “within the past decade ... computer hardware, software, and business methods ossified into a slowly changing set of routines.”

Intel’s work on artificial intelligence (AI) contradicts Gordon’s claim that the business models of the hardware, software and home and consumer electronics industries have “ossified.” Nidhi Chappell, Director of AI Strategy, at Intel notes that Intel has built upon Moore’s Law, data availability and innovation in algorithms to drive greater use of AI. One way Intel broadens the use of AI is by compressing the innovation cycle (a major change in routines), democratizing access to AI and guiding the development of AI in service of humankind (by solving cancer, decoding the function of the human brain, etc.).

50 Gordon, p. 567.
51 Gordon, p. 580.
52 Gordon, p. 581.
53 Gordon, p. 582.
54 Gordon, p. 583.
55 Nidhi Chappell, Intel’s Director of AI Strategy, “Under the Hood: Intel Accelerating the Future of Artificial Intelligence | Intel IT Center,” [https://www.youtube.com/watch?v=MKFlvNTre2I](https://www.youtube.com/watch?v=MKFlvNTre2I)
Intel has also used Big Data and software engineering to solve high-value problems. It has assembled teams of 5 people who focus on solving production problems for 6 months. They use historical data as well as unstructured data to predict outcomes. Each team is expected to save Intel at least $10 million in six months. The Intel teams harness new, Big Data skills plus software-defined, cloud computing infrastructure to analyze large databases. They reduce manufacturing costs by enhancing the initial testing of new semiconductor manufacturing processes. When Intel saves 1 second of test time in production, it saves $5 million to $10 million. This is a breakthrough in solving defects and problems in production by identifying the root cause. Again, this evidence defies Gordon’s contention that the industry has ossified.

Gordon’s conclusions overlook the sizable recent changes in business practices and business structure. The Wall Street Journal has cited Equifax Inc., insurer Liberty Mutual, and consumer-products giant Procter & Gamble Co., as firms that are adopting a mobile, cloud and data technology from Silicon Valley. By doing this, they are changing the way they operate, taking on many of the characteristics of tech and Internet firms, such as Facebook, Google, and Amazon. Many of these larger firms are also shifting to shorter development cycles. This helps them be more agile and responsive to changes in the marketplace. This is an enhancement in corporate behavior that Gordon does not discuss.

Gordon emphasizes that price declines for ICT equipment relative to performance has slowed. He cites data showing that by 2014, there were almost no price declines at all, as compared to rapid price declines in the late 1990s. Gordon expects that this slowdown to continue at the same pace, repeating the slow rate of TFP growth from 2004-2014. He then asks whether the next wave of innovations prove to be revolutionary, as they were during the dot-com revolution of the late 1990s, or result in a slower growth of productivity, essentially at the same pace as during 2004-2014. It is fair to note in rebuttal that some scholars have recently developed a performance indicator that they have merged with the producer price index (PPI) for servers. When they do this, the price changes, i.e., the blended indicator and PPI fall 11 percent faster than the BEA Investment Index.

Gordon then asks whether the next wave of innovations will prove to be revolutionary, as they were during the dot-com revolution of the late 1990s, or result in productivity a growth of at the same pace as during 2004-2014? The theme of his chapter on IR#3 is that “the main benefits for productivity growth provided by the Third Industrial Revolution were centered on the decade between 1994 and 2004. Since

---

58 Gordon, p. 593.
59 Gordon, p. 593.
61 Gordon, p. 593.
2004, the pace of innovation has been slower, but it has certainly not been zero.\textsuperscript{62} To assess the future advances that are widely forecast by Brynjolfson and McAfee as well as others, he divides them into four main categories, medical, small robots and 3D printing, dig data and driverless cars. Gordon then examines whether these categories of innovation might bring TFP growth back to the levels in the late 1990s.

In examining medical and pharmaceutical advances, Gordon finds that medical technology has continued to advance since 1980 but at a “slower and measured pace.” He finds that pharmaceutical research has hit a “brick wall of rapidly increasing costs and declining benefits.”\textsuperscript{63}

It is disconcerting that Gordon has overlooked examining some of the cornerstone achievements of medical science. One change he overlooks is the rapidly declining cost of decoding the human genome. As the following chart notes, the cost per genome has fallen much faster than Moore’s Law.\textsuperscript{64}

\textsuperscript{62} Gordon, p. 593.
\textsuperscript{63} Gordon, p. 594.
\textsuperscript{64} National Institutes of Health, National Human Genome Research Institute “The Cost of Sequencing a Human Genome” \url{https://www.genome.gov/sequencingcosts/} As this site notes, “The underlying costs associated with different methods and strategies for sequencing genomes are of great interest because they influence the scope and scale of almost all genomics research projects.”
Gordon maintains that pharmaceutical research has hit a wall of rapidly increasing costs. This point of view is true if one accepts the findings of the Tufts Center for the Study of Drug Development. A cogent criticism of these results is that they only consider new molecular entities, drugs with chemical compounds that have never been approved for individual use or combination therapies. This is, however, only a small part of the population of new drugs each year and does not include many of the drugs developed with funding from the National Institutes of Health and other government entities.

---

In discussing the limitations of innovations such as small robots, Gordon reasons that it is difficult for them to “match a human’s dexterity and problem-solving abilities.” He notes that it is difficult for these robots to distinguish between picking up lace and crumpled jeans.

Research completed after Gordon’s book was published notes that advances in machine learning have made it possible for robots to enhance their dexterity by combining vision and touch. The lab at the University of California, Berkeley that enhanced robotic dexterity taught a robot to fold clothes.

In 3-D printing, Gordon notes that it is “not expected to have much effect on mass production and thus on how most U.S. consumer goods are produced.”

Recently, engineers at Renault Truck have boosted performance of truck engines by using 3-D printing to replace 25 percent of these engines’ parts. This results in much better performance and the ability to carry greater workloads with lower fuel consumption.

Timothy Bresnahan and Pai-Ling Yin offer a second interpretation of the impact of IR#3. They indicate that “the invention of new applications based on information and communications technologies (ICTs) has had two economic effects up to now.” It has “transformed production” and shifted the demand for skilled labor in the workforce.

One of the assertions that Bresnahan and Yin make is that ICTs are enablers of the invention of new applications and that most of the innovation occurs in firms outside of the ICT industries. They focus on co-invention, “the product and process improvements created by industries as they apply new ICT,” noting that:

“ICT co-invention is defined as the product and process improvements created by industries as they apply new ICT. One driver of co-invention is the ICT advances themselves (supply), such as cheaper storage, faster networks, or more capable software. ICT advances produce a large scope of feasible opportunities. The other driver is the industry circumstances (demand) of firms trying to use ICT: competition, customer demand, and the production processes already in place.” They argue that even with the rise of Big Data, analytics, Mobile, and The Cloud, co-invention today is very like previous stages in the growth of ICT advances and that “while there has been terrific technical progress in ICT, there has been little change in the ICT co-invention process. Co-invention still requires considerable brainpower and experimentation. Co-invention still looks for ways to change whole organizations. Indeed, modern co-invention often looks for ways to change whole supply chains.”

---

66 Gordon, p. 596.
68 Gordon, p. 597.
We would agree that the invention of new applications, particularly new software, takes place largely in firms outside the ICT industries.

Bresnahan and Yin note that “The three key features of ICT co-invention have important implications for value creation and for labor demand. (1) The need for brainpower and experimentation to translate ICT into new and valuable products and services provides a very simple explanation of why ICT co-invention has raised the demand for smart managers and professionals. Despite all the advances in artificial intelligence (AI), there does not yet seem to be a program that can figure out how to change the incentives of many workers, customers, and suppliers to make a new organization that can supply improved product quality as defined by market demand. (2) The organizational change involved in ICT co-invention increases demand for workers with “organizational participation skills (OPS)” in all wage brackets. (3) The complexity of ICT co-invention renders it a long and sustained process, so these labor demand implications will sustain many of important disparities in the labor market over the last 50 years (changes which many observers have found alarming).”

We believe that there are important changes in the software innovation process that change points 2 and 3. First, there is a great deal of independent development of innovative software, such as Linux, Open Software, and Docker/containers. This means that organizations can obtain important software at very low cost and without having the internal skills to develop it. Second, these new efforts to develop software...

---

71 Bresnahan and Yin, pp. 4 and 5.
as well as new points to access and share software, such as GitHub, have shortened the time it takes for co-invention. With today’s software, many firms can change the way they manage supply chains and organize production processes in a very short amount of time.

Underlying these trends, as we noted at the beginning of this paper, firms are changing the processes they use to develop new software. The software development process is now in “perpetual development mode.” This means that by using DevOps, continuous service delivery and microservices, firms are deploying and benefitting from the latest software innovations. They have also streamlined what once was a very cumbersome and time consuming process for co-invention.

We would argue that economists should examine software innovation to discover how it is part of perpetual process of improvement that benefits from groups working on Open Source software. These innovations no longer rely upon the cumbersome process described by Bresnahan and Ying. There are patterns of rapid sharing between firms that rapidly change the benefits that firms can achieve in deploying software for key business functions.

Yale Garten\textsuperscript{72}, the Director of Data Science at LinkedIn reported at the Strata-Hadoop conference that her group had developed a Unified Metrics Platform (UMP) that lets analysts and engineers write the logic for new metrics more easily. In designing the new LinkedIn mobile app, the data science group took nearly 9 months to complete the upgrade. Once analysts had more experience with the UMP, they created the laptop app in a month’s time. This great gain in efficiency was tied to the use of the UMP. Garten spoke of the improvement as one tied directly to innovation in the use of software.

D. Recent Changes in the Way Software is Created and Deployed

1. Business Gains from Software Innovation

Businesses have reported that they are using cloud computing environments to change the way they use software.

This anecdotal evidence supports several trends that many have recognized. First, there is an overall shift to new functions within many firms as part of the move to become more software-based and data-oriented.\textsuperscript{73} Second, this shift brings about a change in the workforce within organizations.

First, as firms become more digitally-oriented, they adopt many new functions. In some cases, these are cost-saving and related to virtualization of computing and data storage. Many firms also begin to use software and software-defined infrastructure to add new business functions, as mentioned in the cases

\textsuperscript{72} Shirshanka Das and Yael Garten “Architecting for change: LinkedIn’s new data ecosystem” O’Reilly Strata-Hadoop Conference, Sept 28, 2016

\textsuperscript{73} The author has studied several social media and incumbent firms across six industries in detail and found that this change is common across a range of industries.
provided above. When these functions are combined, the result is not only cost savings and restructuring, but also gains in revenue and efficiencies.\textsuperscript{74}

Second, many firms invest in specific occupations to obtain the skills needed to focus on new digital functions. McKinsey’s recent study of Big Data\textsuperscript{75} finds that demand for these positions is growing so rapidly that there is likely to be a shortfall of 250,000 people to fill data analysis jobs in 2024. This is although many students have been attracted to programs that offer data analysis degrees.

Figure 14. The Expected Number of Trained Data Scientists is Insufficient to meet Demand

![Figure 14](source)

Source: Nicolaus Henke and others, “Big data: The next frontier for innovation, competition, and productivity,” McKinsey Global Institute, June 2011, p. 11.

2. Where will technology increase productivity?

There are three ways that new infrastructure technology will increase productivity:

1. Through reductions in the cost of software-defined infrastructure

   a. Since software takes over the operation and management of many parts of infrastructure, hardware costs that had been quite high are avoided.

   b. Software costs can also be reduced because much of the software used in software-defined infrastructure, such as Software Defined Data Centers, comes from sharing. Open Source software and software provided by GitHub, a sharing hub on the Internet, reduce spending on programming. They also mean that enterprises avoid paying vendors a licensing fee for software.

\textsuperscript{74} My talk, “IoT, New Business Models and Digital Agility – Ties to Security,” explores these efficiencies in greater detail. \url{http://www.slideshare.net/bcohen777/iot-new-business-models-and-digital-agility-102116}

\textsuperscript{75} Nicolaus Henke and others, “The Age of Analytics: Competing in a Data-Driven World,” McKinsey Global Institute, December 2016.
2. Through direct spending on new infrastructure.

   a. As firms make the transition to software-defined infrastructure, they will invest substantial amounts to take advantage of cost savings and the ability to do analyses more rapidly. The new infrastructure, such as software-defined data centers, will also support more refined data analysis as well as more rapid development and testing of new software and new applications.

   b. This renewal of infrastructure is essential for moving firms to a New Production Ecosystem where products can be produced more cheaply and services can be created more rapidly and at lower cost.

   c. It is also important to note that there have been major changes in the way software is developed and applications are created will change. Software development now relies upon the rapid creation of new applications and swift modification of existing applications. This requires software-defined storage and computing that support continuous software delivery (drawing upon DevOps, i.e., shortening the software and services development cycle, microservices where software is assembled in Lego-like fashion, and containers, where developers can create a single application and easily run it using a wide range of operating systems without significant modifications). Using such techniques, some firms have been able to deploy new applications or services in less than an hour.

   d. The result is that speeding up analytics as well as software development makes a firm more productive. This is because analysis that is central to a firm’s operations can be supported in ways that move more to real-time analytics. In addition, important processes and software development can be speeded up, lowering costs.

3. Through the expansion of business opportunities.

   a. New ways of developing software reduce the time to market for new applications and services. This lets enterprises expand the services and products they offer with very short development times. The result is an increase in firm revenues and productivity, the output produced per employee.

   b. Big data analysis, the analysis of large amounts of data (data lakes) gives enterprises insights into markets. This type of analysis was not easy to perform with highly distributed data bases. This provides opportunities to address these markets in ways that create new value. It also means that businesses can refine the analysis of designs to develop products and services that compete better in the marketplace. Big data analysis also provides better understanding of the markets businesses are trying to serve.
We are developing an argument that innovation is taking place in software itself. This has resulted in changes in what firms are doing with software. In moving to software-defined infrastructure, not only have costs been reduced, but businesses have been able to exploit new software to expand operations and refined products.

The improvements in productivity noted above:

a. Reduce the cost of developing and refining new products and services.

b. Create software innovations that operate very much like Moore’s Law, sparking continuous expansion in the amount of work that firms can perform in a broad range of operations, not merely in information technology viewed in a restricted context.

c. Enhance the ability of firms to analyze data at much lower cost than had been the case previously. This adds considerable value to product and services development. It also contributes to the Moore’s Law-like cost reductions in a range of business operations that can exploit the software innovations mentioned above.

Case Examples:

3. Why does technology not appear to have improved productivity in recent years?

   a. Most economists continue to rely upon very traditional analytic approaches to capture the impact of substantial changes in software development and deployment.

   For the most part, the analytic approaches continue to focus on changes in hardware prices and performance. These metrics are the main indicators of the shift to the digital economy. Some economists have explored pricing adjustments for software, but evaluating software changes through declines in the prices of hardware is difficult.

   For “cloud and related ICT services, Byrne and Corrado76 (2016) … imply… these prices should fall no slower than the rate of decline in ICT asset prices.

   Press reports suggest that cloud computing and storage services are falling very fast (in the 20 to 30 percent per year range). The cost of cloud services (from purchasers’ perspective) includes high-speed broadband (WAN and LAN) services.”77

---


77 Carol Corrado, “Discussion of: Improving ICT Deflators in the National Accounts,” papers were prepared for the meeting of the Bureau of Economic Analysis Advisory Committee, November 18, 2016. p. 3.

http://bea.gov/about/advisory.htm
b. The contribution from the growth of output from software and cloud services is not estimated directly in national income and product accounts. “ICT capital continues to grow and penetrate the economy -- **increasingly via cloud services which are not fully accounted for in the standard narrative on ICT’s contribution to economic growth** -- the contribution of ICT to growth in output per hour going forward is calibrated to be substantially larger than it has been in the past.”

1. It is possible that many economists studying the impact of cloud services and software are not capturing the real changes in output by estimating hardware price changes and assuming they mirror the trend for ICT (information and communication technologies) in the economy. Changes in hardware prices may not accurately capture software’s impact on the contribution to growth in output per hour.

2. Economists need a better understanding of how software is developed. They need to understand how programmers have created new generations of software; i.e., that software developers have created substantial innovation in the ICT sector.

1. As noted above, many firms have adopted continuous service delivery, with microservices, containers and DevOps to speed software development.

2. Many firms no longer optimize software development for cost, as economists assume. As Cockcroft notes, “Nordstrom is no longer optimizing for software cost but for delivery speed.”

3. The rise of continuous service delivery, microservices, containers and DevOps, has changed the way that developers create, test and deploy software and applications has changed.

4. Economists have not yet taken these new processes into account.

5. Economists also have not developed ways to measure innovations in software delivery. This hampers our understanding of how to incorporate software into the national income accounts.

6. At the firm level, the move to more rapid creation of software is having a clear impact. The 2016 State of DevOps Report, high performing firms speed up the

---

78 Byrne and Corrado, “ICT Prices and ICT Services.”
80 Cockcroft, p. 12.
81 Cockcroft, pp. 52-112. Robert Cohen, The Internet of Things, Productivity, and Employment, presentation for Internet of Things Summit, Boston, Sept. 8-9, 2015 offers a summary of the main points Cockcroft makes.
deployment of new software 200 times more frequently than low performing firms. Nonetheless, many economists are using price changes in some of the key devices used in the digital economy, such as tablets, desktops and laptops to estimate changes in the software industry.\textsuperscript{84}

Appendix: Comparing High Performance Firms to Low Performing Ones

<table>
<thead>
<tr>
<th></th>
<th>High IT Performers</th>
<th>Medium IT Performers</th>
<th>Low IT Performers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment frequency</strong></td>
<td>On demand (multiple deploys per day)</td>
<td>Between once per week and once per month</td>
<td>Between once per month and once every 6 months</td>
</tr>
<tr>
<td><strong>Lead time for changes</strong></td>
<td>Less than one hour</td>
<td>Between one week and one month</td>
<td>Between one month and 6 months</td>
</tr>
<tr>
<td><strong>Mean time to recover (MTTR)</strong></td>
<td>Less than one hour</td>
<td>Less than one day</td>
<td>Less than one day*</td>
</tr>
<tr>
<td><strong>Change failure rate</strong></td>
<td>0-15%</td>
<td>31-45%</td>
<td>16-30%</td>
</tr>
</tbody>
</table>

* Low performers were lower on average (at a statistically significant level), but had the same median as the medium performers.