

# Issue 054: Google

Adrian Kosmaczewski

March 6<sup>th</sup>, 2023



Welcome to the fifty-fourth issue of *De Programmatica Ipsum*, about *Google*.

In this edition:

- We search for the source of the current challenges<sup>1</sup> Google is facing nowadays.
- In the Library section<sup>2</sup>, we review “Nine Algorithms That Changed the Future”<sup>3</sup> by John MacCormick.
- In our Vidéothèque section<sup>4</sup>, we recommend some classic Google TechTalks<sup>5</sup>.

We opened an account on Mastodon last year: follow us at [@deprogrammaticaipsum@mas.to](https://mas.to/@deprogrammaticaipsum)<sup>6</sup> to be notified of new releases!

We would also like to thank our patrons who generously contribute every month (or have contributed in the past) to our work and help us run this magazine. Thank you so much! In alphabetical order: Adam Guest, Adrian Tineo Cabello, Benjamin Sheldon, Christopher Nascone, Jean-Paul de Vooght, Moya, Patryk Matuszewski, Paul Hudson, Roger Turner, and Szymon Licau.

Enjoy this issue! Please subscribe to our free newsletter<sup>7</sup> to stay updated about new releases, share the articles on social media, or contribute<sup>8</sup> if you would like to support our work.

<sup>1</sup><https://deprogrammaticaipsum.com/feeling-lucky/>

<sup>2</sup><https://deprogrammaticaipsum.com/category/library/>

<sup>3</sup><https://deprogrammaticaipsum.com/john-maccormick/>

<sup>4</sup><https://deprogrammaticaipsum.com/category/videotheque/>

<sup>5</sup><https://deprogrammaticaipsum.com/google-techtalks/>

<sup>6</sup><https://mas.to/@deprogrammaticaipsum>

<sup>7</sup><https://deprogrammaticaipsum.com/newsletter/>

<sup>8</sup><https://deprogrammaticaipsum.com/contribute/>

Cover photo by Mitchell Luo<sup>9</sup> on Unsplash<sup>10</sup>.

---

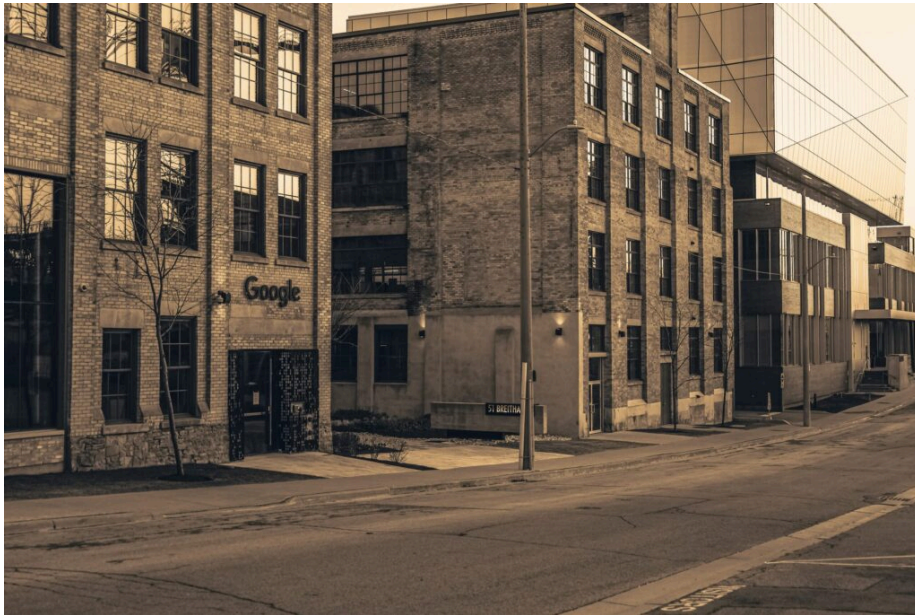
<sup>9</sup>[https://unsplash.com/@mitchel3uo?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/@mitchel3uo?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

<sup>10</sup>[https://unsplash.com/images/apps/google?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/images/apps/google?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

# Feeling Lucky

Adrian Kosmaczewski

March 6<sup>th</sup>, 2023



The traditional Silicon Valley narrative revolves around certain archetypes: fabulously visionary leaders, a seemingly unbeatable streak of incredible products, and, very often, humble beginnings in a family garage. This benign description conveniently foregoes the later stages of such companies, where hubris, arrogance, and mismanagement bring those same behemoths to the brink of extinction. This story has happened dozens of times before and will happen again. It is a story narrated and driven by human nature.

Let us review some notable examples in history.

John Walker<sup>1</sup>, the founder of Autodesk and one of the original programmers of AutoCAD, wrote in April 1991 an inflammatory internal memo called “Information Letter 14,” today available<sup>2</sup> on Walker’s website. It is a long text, but it should constitute mandatory reading for all software CEOs. One particular phrase stands out, though:

One of the largest unappreciated factors in Autodesk’s success has been the poor strategy and half-hearted, incompetent execution that characterised most of our competitors in the past.

Let us keep this phrase in mind as we move forward.

Hewlett-Packard is another sorry example<sup>3</sup> of such a process. This company grabbed Compaq’s feet as it drowned in a naval war that old employees still remember as “the blues against

<sup>1</sup>[https://en.wikipedia.org/wiki/John\\_Walker\\_\(programmer\)](https://en.wikipedia.org/wiki/John_Walker_(programmer))

<sup>2</sup>[https://www.fourmilab.ch/autofile/e5/chapter2\\_86.html](https://www.fourmilab.ch/autofile/e5/chapter2_86.html)

<sup>3</sup><https://www.technewsworld.com/story/hp-vs-apple-the-contrast-is-in-the-turnaround-82116.html>

the reds.” The HP name brings today memories of overpriced toner or ink cartridges... and pretty much nothing else.

Let us not forget about IBM. After becoming synonymous with the computer industry, the company started in the 1990s a steady decline<sup>4</sup> that led to eroding its logo out of the memory of a whole generation of computer programmers. Only those with a particular affinity with computer history (or mainframes or quantum computing) can say anything about it, which is, no matter how you look at it, to say the least, unfortunate.

Some companies, like Dell, Apple, and Microsoft<sup>5</sup>, are famously known for having turned the tides in their favor, prompting dramatic changes in their companies, and directing them to lead new markets, sometimes at the antipodes of their founding ideas.

(IBM’s inflection point might have been the acquisition<sup>6</sup> of Red Hat in 2018, but it is still too early to tell. Or it could be quantum computing, whatever and however that turns out to be.)

Today it is Google’s turn. This company, which has undoubtedly changed the World Wide Web forever, is starting to face the same challenges as Autodesk, Hewlett-Packard, Apple, Dell, and Microsoft faced before; in particular, and paraphrasing John Walker, it is facing “the end of the incompetent execution that characterised” most of their competitors in the past.

In short, Google (the corporation) has been feeling lucky for the past 25 years, first by building a genuine ground-breaking product; then by shoveling cash via online advertising; and finally, by facing quite mediocre competition for years.

Many current Google engineers will have angrily stopped reading this article when reaching the “lucky” word above and will have canceled their monthly contributions<sup>7</sup> to this magazine if any. So be it. By stating the corporation as “lucky,” I do not mean to say individual engineers are not brilliant. Quite the opposite. Google has attracted the *crème de la crème* of software engineering over the past 25 years. The very best A-players at the top of their games. Google’s problem is not an engineering problem.

And this was true from the beginning. It is hard to realize how radically better<sup>8</sup> Google (the search engine) was in 1998. The best search engine available at that time, AltaVista<sup>9</sup>, was okay-ish at best. Google would regularly return *the* best result for your query on top of the list, and within milliseconds: the “I’m Feeling Lucky” button worked wonderfully well back in those days. It is no longer<sup>10</sup> the case.

Google (the company) and later Alphabet (the conglomerate) started claiming home run after home run: Google Images (thanks, Jennifer!<sup>11</sup>), MapReduce<sup>12</sup>, Gmail, “Don’t be evil,” GWT<sup>13</sup>, YouTube, Google Earth, Android, the purchase of DoubleClick, Testing on the

---

<sup>4</sup><https://www.cringely.com/2014/06/04/decline-fall-ibm/>

<sup>5</sup><https://deprogrammaticaipsum.com/where-does-microsoft-want-to-go-today/>

<sup>6</sup><https://www.ibm.com/support/pages/ibm-acquires-red-hat>

<sup>7</sup><https://deprogrammaticaipsum.com/contribute/>

<sup>8</sup><https://hachyderm.io/@thomasfuchs/109861355271027547>

<sup>9</sup><https://en.wikipedia.org/wiki/AltaVista>

<sup>10</sup><https://www.youtube.com/watch?v=48A0OynnmqU>

<sup>11</sup><https://www.thedigitalfix.com/jennifer-lopez/google-images>

<sup>12</sup><https://research.google/pubs/pub62/>

<sup>13</sup><https://www.gwtproject.org/>

Toilet<sup>14</sup>, Google App Engine, doodles<sup>15</sup>, the Chrome browser, the V8<sup>16</sup> JavaScript engine (largely responsible for the JavaScript craze<sup>17</sup> of the 2010s<sup>18</sup>), the Go programming language, Kubernetes, Flutter, Brotli<sup>19</sup>, Chromecast, Dart... and maybe Angular, too, even if Facebook's React ate its lunch.

And this is without counting a long list of discontinued<sup>20</sup> products and services. Google Reader, in particular, where are thou? Picasa? Google Code? Fastlane<sup>21</sup>? Oh, well, at least in this last case, Google still owns the intellectual property<sup>22</sup>, even though the “community” does the work. Phew. Nothing to see here; move along.

How did the competition, specifically Microsoft, react to Google? It did not. This is something Joel Spolsky famously observed<sup>23</sup>, and Steve Ballmer regretted<sup>24</sup> in front of Charlie Rose:

The very fact that Google invented MapReduce, and Microsoft didn't, says something about why Microsoft is still playing catch up trying to get basic search features to work, while Google has moved on to the next problem: building Skynet<sup>H<sup>H</sup>H<sup>H</sup>H<sup>H</sup></sup> the world's largest massively parallel supercomputer. I don't think Microsoft completely understands just how far behind they are on that wave.

In an ironic naming twist, IBM named a 2015 reorganization initiative “Project Chrome,”<sup>25</sup> whereby 26% of IBM employees were at risk of being laid off: a whopping 100'000 of them. Last January, Google followed its corporate peers in the new trend of massive layoffs with 12'000 job cuts, which shook<sup>26</sup> the otherwise calm Google offices next to Zürich Hauptbahnhof.

But the problem with Google today has one name: *hubris*. Its management and most devoted engineers cannot understand the threatening nature of what is coming upon them and, even worse, enjoy a twisted sense of exceptionalism<sup>27</sup> that no longer applies. And no, this is not (only) about GDPR or ChatGPT<sup>28</sup> but their crumbling from within, their self-destructing internal processes, and their incapacity to change.

IBM's and HP's downfall began at the end of the 1980s when sales became more important than engineering excellence, and layoffs became a fashionable way to increase shareholder value. The same happened to Microsoft in the early 2010s, to Apple in the mid-nineties, and so many other companies during their lifetime.

Google's downfall began the day advertising revenue became more important than anything else, including its employees.

---

<sup>14</sup><https://testing.googleblog.com/2007/01/introducing-testing-on-toilet.html>

<sup>15</sup><https://www.google.com/doodles>

<sup>16</sup><https://v8.dev/>

<sup>17</sup><https://deprogrammaticaipsum.com/innovationscript/>

<sup>18</sup><https://deprogrammaticaipsum.com/issue-17-a-2010s-retrospective/>

<sup>19</sup><https://github.com/google/brotli>

<sup>20</sup>[https://en.wikipedia.org/wiki/Category:Discontinued\\_Google\\_services](https://en.wikipedia.org/wiki/Category:Discontinued_Google_services)

<sup>21</sup><https://mastodon.social/@steipete/109891897156562936>

<sup>22</sup><https://mastodon.social/@joshdholtz/109892201251474243>

<sup>23</sup><https://www.joelonsoftware.com/2005/12/29/the-perils-of-javaschools-2/>

<sup>24</sup><https://www.youtube.com/watch?v=v9d3wp2sGPI>

<sup>25</sup><https://www.cringely.com/2015/01/22/ibms-reorg-hell-launches-next-week/>

<sup>26</sup><https://www.inside-it.ch/massentlassung-in-zuerich-googler-protestieren-20230215>

<sup>27</sup><https://medium.com/@pravse/the-maze-is-in-the-mouse-980c57cfd61a>

<sup>28</sup><https://deprogrammaticaipsum.com/open-letter-to-a-future-ai/>

These days... Google Cloud customers hope it will not be discontinued. Carbon will hardly become<sup>29</sup> a competitor to Rust. DuckDuckGo, Brave Search, and Swisscows<sup>30</sup> return excellent results without privacy concerns. Apple built strong tracking protection into iOS 15, cutting ad revenue. TypeScript has eaten JavaScript for breakfast. Angular is no longer relevant; these days, the whole concept of SPA is under attack<sup>31</sup>. Kubernetes is derided for its complexity. Thanks to Oracle lawyers, Google had to call JetBrains to the rescue<sup>32</sup>, again<sup>33</sup>. Flutter struggles<sup>34</sup> to become relevant. People hate receiving AMP<sup>35</sup> links. Pixel devices are a drop in an ocean of insecure Android smartphones. Chrome is the new Internet Explorer. And Bing is slowly becoming the Skynet<sup>36</sup> Joel was worried about.

And now, layoffs. What is next?

As Robert X. Cringely once explained<sup>37</sup>, the 5th stage of companies can either be rebirth... or death. Google is running out of luck.

Cover photo by Caleb Williams<sup>38</sup> on Unsplash<sup>39</sup>.

---

<sup>29</sup><https://accu.org/journals/overload/30/172/teodorescu/>

<sup>30</sup><https://swisscows.com/en>

<sup>31</sup><https://infrequently.org/2023/02/the-market-for-lemons/>

<sup>32</sup><https://kotlinlang.org/>

<sup>33</sup>[https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)

<sup>34</sup><https://deprogrammaticaipsum.com/the-law-of-diminishing-returns/>

<sup>35</sup><https://support.google.com/google-ads/answer/7496737?hl=en>

<sup>36</sup><https://www.independent.co.uk/tech/chatgpt-ai-messages-microsoft-bing-b2282491.html>

<sup>37</sup><https://www.cringely.com/2016/03/22/ginni-the-eagle-ibms-corporate-transformation/>

<sup>38</sup>[https://unsplash.com/es/@callys\\_corner?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/es/@callys_corner?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

<sup>39</sup>[https://unsplash.com/photos/vsQyYgz0t70?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/photos/vsQyYgz0t70?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

# Google TechTalks

Adrian Kosmaczewski

March 6<sup>th</sup>, 2023



Seventeen years is a long time in our industry. To give us an idea, Gmail appeared eighteen years ago, and in those days Google pledged not to be evil. Those were the days of AJAX web applications, of Prototype versus jQuery, of Ruby on Rails and script.aculo.us. Those were the times before Obama, before the MacBook Air, before the pandemic, before Google Chrome, before the Marvel Cinematic Universe, before Android devices, before Docker and Kubernetes, before the Go programming language, before the V8 JavaScript engine, before the 2008 stock market crash, before Brexit, before SPAs, before Node.js and npm, before Star Wars Episode 7, before HD video was widespread.

In those days there was not only YouTube, but also a similar thing called Google Video, a service that was promptly phased out, as Google often does. And in that old service we could dream to be a Google employee in Mountain View attending one of the many Google TechTalks<sup>1</sup> of that era.

Thankfully, Google transferred most of those videos to YouTube, and we can (and should) watch them today. The audio is sometimes jaggy; the image is far from HD quality; it has the grainy texture of early web videos, and it looks certainly amateur by today standards. Some were filmed pointing directly to the screen where the slides were being projected. HDMI and splitters weren't as widespread back then as they are today.

Let us enumerate some jewels in the Google TechTalks collection, beginning with program-

---

<sup>1</sup><https://www.youtube.com/user/GoogleTalksArchive/videos>

ming language-related subjects. In July 2007, Lennart Öhman celebrated 20 years of Erlang<sup>2</sup> with a Google TechTalk. In 2006 Jack Herrington talked about code generation in Ruby<sup>3</sup>. Lawrence Crowl showed the upcoming new features of C++0x<sup>4</sup>. Raj Bandyopadhyay about the compilation of Python<sup>5</sup>. Philippe Mouglin about the FScript<sup>6</sup> scripting language for Mac OS X.

There were lots of celebrities, too. Guido Van Rossum, Google employee at the time, explained not only how the Mondrian code review tool<sup>7</sup> worked, but also gave another talk about Python 3000<sup>8</sup> (remember?) Gregor Kiczales talked about Aspect-Oriented Programming<sup>9</sup>. Jacob Kaplan-Moss gave a speech about Django<sup>10</sup>. Brian Foote described the Big Ball of Mud<sup>11</sup>. John Resig talked about JavaScript library design<sup>12</sup>, a topic he knew a thing or two about. Bjarne Stroustrup talked about C++0x initialization lists<sup>13</sup>. Marissa Meyer talked about scalability<sup>14</sup>. Scott Berkun about the browser wars<sup>15</sup> in the 1990s. Mark Shuttleworth about Ubuntu<sup>16</sup>. Ola Rosling about Gapminder.org<sup>17</sup>. Richard Hipp about SQLite<sup>18</sup>. And Ron Avitzur told the story of how the Graphing Calculator<sup>19</sup> became part of Mac OS despite Apple's best efforts.

This is what I call being starstruck.

Many presentations dealt with project management, teams, and their dynamics. Robert Watson explained how the FreeBSD<sup>20</sup> project worked. In a premonitory talk 13 years before the pandemic, Hubert Smits showed how to plan and work with distributed teams<sup>21</sup>. Viral Shah and Vikram Aggarwal talked about the measurement of programmer productivity<sup>22</sup>. Wilco Jansen and Louis Landry about the Joomla!<sup>23</sup> project. Adam Connors and Joe Walnes about testing AJAX applications<sup>24</sup>.

Among those Google TechTalks there were not only talks about programming subjects: there were also sessions about physics<sup>25</sup>, economics<sup>26</sup>, creativity<sup>27</sup>, mathematics<sup>28</sup>,

<sup>2</sup><https://www.youtube.com/watch?v=apZYqHPqzYo>

<sup>3</sup><https://www.youtube.com/watch?v=68hYME8t93I>

<sup>4</sup><https://www.youtube.com/watch?v=nLX8xaysUMo>

<sup>5</sup><https://www.youtube.com/watch?v=8D2EFbY5csg>

<sup>6</sup><https://www.youtube.com/watch?v=SNVnoulAHbg>

<sup>7</sup><https://www.youtube.com/watch?v=CKJRt48rZGk>

<sup>8</sup><https://www.youtube.com/watch?v=UIDdgeISLUI>

<sup>9</sup><https://www.youtube.com/watch?v=40Q16Ix-src>

<sup>10</sup><https://www.youtube.com/watch?v=n8KnFywpXOE>

<sup>11</sup><https://www.youtube.com/watch?v=h6Y9aJhqO78>

<sup>12</sup><https://www.youtube.com/watch?v=012mt05yzjc>

<sup>13</sup><https://www.youtube.com/watch?v=-OioZ78jnlM>

<sup>14</sup><https://www.youtube.com/watch?v=Syc3axgRsBw>

<sup>15</sup><https://www.youtube.com/watch?v=sInCa-Adrco>

<sup>16</sup><https://www.youtube.com/watch?v=4yox180r3ME>

<sup>17</sup>[https://www.youtube.com/watch?v=rZ1\\_XbbJMZE](https://www.youtube.com/watch?v=rZ1_XbbJMZE)

<sup>18</sup>[https://www.youtube.com/watch?v=jN\\_YdMdjVpU](https://www.youtube.com/watch?v=jN_YdMdjVpU)

<sup>19</sup><https://www.youtube.com/watch?v=GMyg5ohTsVY>

<sup>20</sup><https://www.youtube.com/watch?v=Y0av1OOMKOA>

<sup>21</sup><https://www.youtube.com/watch?v=33ppmkhQCz8>

<sup>22</sup><https://www.youtube.com/watch?v=iDO3AWbBLNw>

<sup>23</sup><https://www.youtube.com/watch?v=8a5hGkjleXg>

<sup>24</sup><https://www.youtube.com/watch?v=liliUhHIRjw>

<sup>25</sup><https://www.youtube.com/watch?v=ogeACgnXyuE>

<sup>26</sup><https://www.youtube.com/watch?v=jeCi93dBrio>

<sup>27</sup><https://www.youtube.com/watch?v=3fSB6ut-cT0>

<sup>28</sup><https://www.youtube.com/watch?v=lvLxRnSrhck>



education<sup>29</sup>, and of course at least one about the much hyped semantic web<sup>30</sup>.

Google used to stand for bigger things. The Google TechTalks<sup>31</sup> series were one of those things that inspired a whole generation of young engineers to work at a company that openly claimed not to be evil.

Cover snapshot by the author.

---

<sup>29</sup><https://www.youtube.com/watch?v=rXmBoicw1FA>

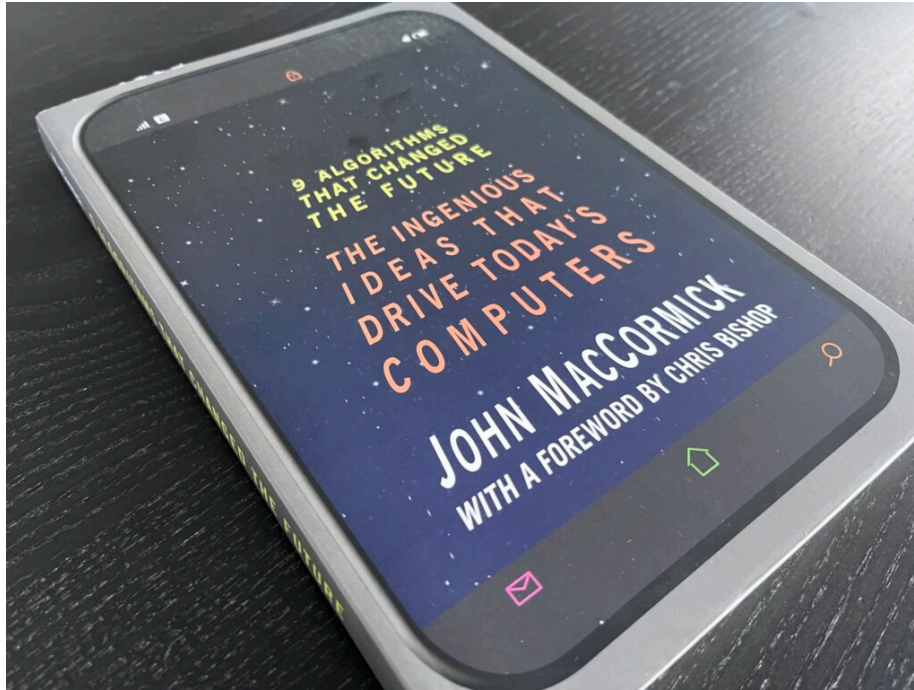
<sup>30</sup><https://www.youtube.com/watch?v=mW12yS1sxfI>

<sup>31</sup><https://www.youtube.com/user/GoogleTalksArchive/videos>

# John MacCormick

Adrian Kosmaczewski

March 6<sup>th</sup>, 2023



As a member of Generation X, this author has had the distinctive privilege of trying to explain computer topics to family members born in the early 1900s. In particular, my grandmother would, around 1999, ask what I did for a living. As a Polish immigrant who arrived in Argentina months before World War II, she could not have been further away from the likes of the World Wide Web, the Netscape browser, or the VBScript programming language. I tried as hard as I could, but of course, I failed miserably. For most of her life, she must have thought, just like my mother, that I was into some dodgy business.

Explaining computer concepts to people outside the field is easier now than it used to be, but many aspects remain obscure to the public. The computer, arguably the most significant invention of the 20th century, is not a single object but rather an utterly novel field of knowledge and work, the result of a mere century of work in mathematical analysis and electronics. There are things you can see and touch (the “hard” ware) and those things you just cannot, the “soft” ware. While the former quite straightforwardly stems from science fiction serials, the latter belongs to an occult realm. The “ghost in the shell” that animates every little thing that defines our modern world is ridden with mystery.

Very few scientists have dared to explain the inner workings of these new machines to the “non-initiated” crowd. Among those who tried, few have had the success that John MacCormick<sup>1</sup> had while writing his 2011 masterpiece, “Nine Algorithms That Changed the Future: The Ingenious Ideas That Drive Today’s Computers.” MacCormick knows a thing or

<sup>1</sup><https://www.dickinson.edu/johnmaccormick>

two about the subject: he is a professor of computer science at Dickinson College in Carlisle, Pennsylvania, specializing in computability.

The book's first two chapters are entirely dedicated to web search engines, granting its faithful inclusion in this edition of the Library section. How does a search engine, say, Google or AltaVista, return meaningful results? And most importantly, how is it possible to do that so quickly? After a first chapter dealing with simple approaches to the problem, conveniently sprinkled with diagrams and simple examples, the second chapter tackles the core of Google's search engine: the PageRank algorithm<sup>2</sup> designed by Larry Page and Sergey Brin that kickstarted a revolution 25 years ago.

Later chapters deal with public key cryptography, Hamming codes, neural networks, and databases. The last chapter, "What is Computable?" describes the concepts behind Alan Turing's work. In all of these cases, the examples are simple, profusely illustrated with diagrams and simple analogies, and written with a language easily understood by anyone with at least a high-school level of knowledge.

Complement this volume with his 2018 introduction to the theory of computation, "What Can Be Computed? A Practical Guide to the Theory of Computation"<sup>3</sup>. However, be aware this volume targets undergraduate students and not the general public per se.

"Nine Algorithms That Changed the Future," updated in 2020<sup>4</sup>, is a joy to read, if not for programming experts, but at least for their family members. If my grandmother were still alive, I would probably try again to explain what I do, this time taking a few cues from Professor MacCormick's playbook. Well, maybe.

Cover photo by the author.

---

<sup>2</sup><https://en.wikipedia.org/wiki/PageRank>

<sup>3</sup><https://whatcanbecomputed.com/>

<sup>4</sup><https://press.princeton.edu/books/paperback/9780691209067/nine-algorithms-that-changed-the-future>