On the impact of software

prof.dr.ir. Michiel van Genuchten

Mr Rector Magnificus,

Executive Board of Tilburg University

Committee of Deans

Ladies and Gentlemen

The amount of software within organizations and in all kinds of products as well as services has been growing for decades. This was summarized in the statement ‘every company is a software company’[1] [Humphrey 2001]. The reproduction costs of software are negligible which means the company that can sell software in the highest volume typically wins. The consequences are rather obvious: legacy companies disappear and new, powerful software companies start dominating. This was summarized as ‘Software is eating the world’[2]. How to avoid being eaten by software as an organization, industry, or economy is less clear. This inaugural lecture will first explain the underlying mechanisms that drive software growth and volume. Second, it will suggest possible responses for companies, industries, and economies to protect themselves from being next on the menu.

---

[1] https://www.satellitetoday.com/technology/2019/02/26/microsoft-ceo-every-company-is-now-a-software-company/ URL’s in the footnotes have been accessed on 28-08-2026, exceptions indicated

[2] https://www.wsj.com/articles/SB10001424053111903480904576512250915629460

Software grows

The growth of software started in the 1960's in the computer industry. In the next decades software ended up in products as computers, mobile phones, medical equipment and cars. In parallel, software played an increasing role in organizations as Information Systems, office applications and Enterprise Resource Planning.

In 2010, my colleague Les Hatton and I started a series of columns on the impact of software on different industries in IEEE Software. The goal was to better understand and quantify the impact of software on different products and organizations. Who better to ask than practitioners who have been with a product for a long time? We have published almost 60 columns since 2010, mostly written by practitioners. We have developed an obsession over software growth. One example is the software in the Mars Rovers as depicted in Figure 1. The software in the Mars Rover is famous for multiple reasons, not only it is rocket science, but it is also 3 million lines of code, built by a 35 people team and no defects have been found in the software so far [Holzmann 2013].

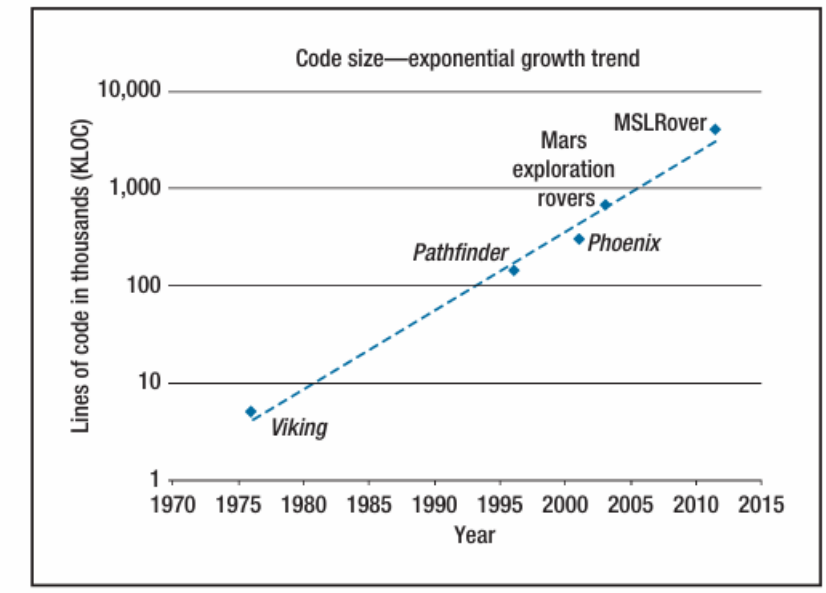


Figure 1 The growth rate of software in Mars Rovers over a 40-year period

The growth picture shows a straight line in a graph with the years on the horizontal axes and the size of the software on a logarithmic scale on the vertical axes. The growth rate is constant, in this case around 20 percent per year. The well-known compound annual growth rate (CAGR) formula is [Graham 1934]:

CAGR = (Ending Value / Beginning Value)^(1 / n) - 1,

where 'Ending Value' is the final amount, 'Beginning Value' is the starting amount, and 'n' is the number of years.

We observed the 20 percent growth in more products and decided to research it in a bigger data set: over 2000 data points in a set of 350 million lines of code [Spinellis 2017]. A few years later we could conclude that growth rate of software is 1.21 ± 0.01. As Popper taught us, a claim gains credibility by surviving attempts at falsification: the 20 percent growth rate was tested against a dataset comprising more than 350 million lines of code from products in use over several years, and the result held.

Lines of code is not the whole story. Knowing that the distance to a destination is 1000 kilometres does not tell the time needed to drive there. One needs to know more such as such as the terrain, the weather, the traffic jams, your car and the willingness to risk speeding to estimate the time and investment needed.

What are the consequences? One is that it is difficult to catch up. One can grow your software with around 20 percent a year, but so can the competitor who is already ahead and will only extend its lead year after year. This holds an important lesson for companies and governments who think they can catch up and create their own cloud or other software products while they are years or decades behind the market leaders. For those executives that convince themselves they can catch up with one magic and very expensive effort, dream on. Or to say it more scientifically: the empirical data does not support your conviction.

A few thoughts for those who think Artificial Intelligence (AI) will change everything and therefore the discussion so far is not relevant. First, AI will likely accelerate the amount of software produced per year so the CAGR may increase. Yet, the consequences for the quality of the code are largely unknown. Gitclear[3] analysed 623 million lines of code between 2023 and 2026 and concluded that the amount of code duplicates increased by 41 percent. At the same time, the percentage of code that must be changed within two weeks has increased to 18 percent. Vendors of AI technology may state that this kind of empirical research is slowing down progress. They follow a tradition that served the software industry well over the past decades and can be summarized as: “go-go-go, just trust us, we know it all, this is all new, we will take care of it, you start paying today and we will see where it ends”.

Optimists are convinced that AI will help to catch up. And the entrance of young companies such as Anthropic with Claude is exciting. However, Bigtech companies such as Microsoft, Google and AWS own the development environment needed to build

[3] https://gitclear-public.s3.us-west-2.amazonaws.com/The_Maintainability_Gap_August_2026_GitClear_AI_Research.pdf

software. They will likely be more effective in using AI to improve software development than their customers, also because they also decide when they make the latest version of their development tools available.

Volume wins

The economics of software are unlike many other products because the reproduction cost is (almost) zero. Or as economists phrase it: “production of an information good involves high fixed costs but low marginal costs” [Shapiro 1999]. A viable strategy is therefore to achieve a higher volume and divide the investment by a growing number of users. If the software attracts a lot of new customers, the code is getting excellent mileage [Genuchten 2011]. If a lot of software must be written per new customer, one needs to have a high royalty per customer, or profitability is unlikely.

Software in products often starts as embedded software, which is software that is not on the invoice and given away with the hardware [Gal 1996, Genuchten 2007]. The next phase is often proprietary software: software starts appearing on the invoice, but the software only works on the hardware of the same supplier. Money starts changing hands, so it gets interesting from an economic perspective. That is how IBM started selling software on their mainframes more than 50 years ago [Campbell 2001]. It is the way that car manufacturers try to make some extra money today by selling users software updates and upgrades. In terms of volume: the supplier still cannot sell more copies of his software than the number of hardware products it sells.

The next phase is open systems software: software works on the hardware of multiple vendors and becomes an independent business. This happened in the 1990’s in the PC world with the arrival of operating systems and applications that worked on the computers of multiple vendors. The volume of Microsoft software on computers exceeded that of the largest computer manufacturers and Microsoft has dominated the computer industry since the mid-nineties. Large computing companies such as IBM lost its leadership position to software companies [Fisher 2000]. Something similar happened in mobile phones with the arrival of Android. Car manufacturers, banks and providers of medical equipment are facing a similar revolution as we speak.

Open-source software is a possible next stage. Source code is shared for free, and others are allowed to use it, under certain conditions. Open-source software is a serious movement that has led to some important products. For example, most internet servers run on Linux, which is arguably the most famous open-source product[4].

[4] https://w3techs.com

Notably, open source also shows up in the discussions on AI and large language models[5].

Let's look at volume from the point of view of the user of software. How about if a software application is unique and it must be built from scratch? Systems theorist Russell Ackoff [1972] famously noted, "We fail more often because we obtain the right solutions to the wrong problems than because we get the wrong solutions to the right problems". After five decades of large-scale software development, a lot of solutions are available in one shape or form. Concluding that only a tailor-made software solution can solve the problem at hand likely leads to solving the wrong problem.

As a user of unique piece of software, one does not monetize the fact that the reproduction costs are zero. This is denying a very important characteristic of software which will likely hurt the customer in the long run. Take the European ministries of defence, they have enough budget these days to build individually tailored and national solutions. Only 18 percent of defence investments in the EU took place collaboratively between EU countries[6]. This is worrying knowing that software is becoming a bigger part of defence investments. The Software Engineering Institute at Carnegie Mellon University predicted some years ago that by now software accounts for 80 percent of costs of weapon systems[7]. Building the software for weapon's systems multiple times in European countries is ineffective and inefficient. An important question whether your tailored solution can work in the first place. Defending the Dutch coastline with a Dutch solution likely is not. Aligning with the other countries around the North Sea is a necessity to know what is coming our way. Being able to divide the cost of such a solution between around 5 countries is an additional benefit. Another question is who will pay for the continued growth of the software after the initial budget is gone. Two things are sure: the software will keep growing and the supplier will not pay for continued development, the user will.

There are two ways that customers can stimulate the increase in volume of software. The first is to increase the volume by teaming up with multiple customers. In the case of defence systems, they can be allies facing similar threats. A second way is to reuse parts of the software stack. A simplified model of the software stack may help. Gregor Hohpe proposed the Universal Architecture Diagram presented here as figure 2.

---

[5] https://github.com/eugeneyan/open-llms
[6] https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2026)785665
[7] https://www.sei.cmu.edu/news/sei-research-combats-mounting-acquisition-costs/

“Economies of scale dictate growing the base layer. … In contrast economies of speed and the drive for innovation favour independence and innovation in the vertical pillars. The relevance of this decision leads to virtually every IT strategy deck to have some version of the Universal Architecture Diagram.”[8]

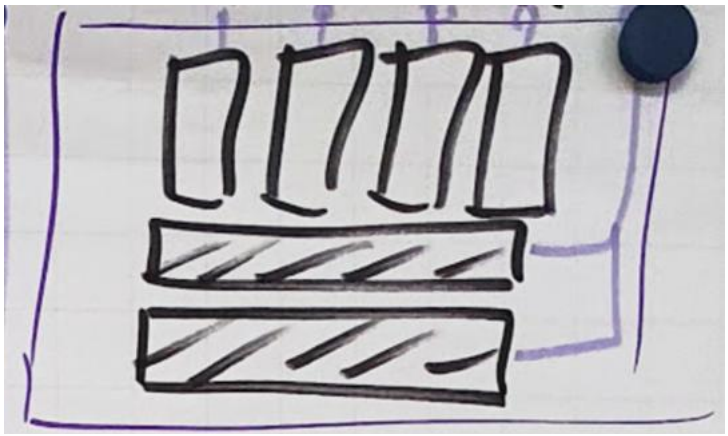


Figure 2 Universal architecture diagram © Gregor Hohpe

Togaf[9] proposed a similar solution architecture that models horizontal solutions (foundation and common systems) and vertical solutions (software products tailored towards a specific industry). The horizontal layers include Operating Systems in computers since the 1990’s, the development environments for software, the cloud (since 2010) and AI platforms (being developed for 10 years and marketed as we speak). The horizontal layers have large volumes since they accumulate their users across industries. Bigtech dominates the horizontal platforms [Cusumano 2020]. The vertical software layers represent the software needed to run ‘verticals’ such as banking, education, defence or healthcare.

Figure 2 may help to determine how a customer can utilize the fact that the reproduction cost of software is zero. First, find external suppliers who sell their software in high volume for the horizontal parts of the software stack. Next, try to find external suppliers who may be able to provide parts of the vertical. The only specific software that is needed to run an industry is likely much higher in the vertical software stack and a small percentage of the total amount of software needed.

---

[8] https://architectelevator.com/ref/universal-architecture/
[9] https://www.opengroup.org/architecture/0210can/togaf8/doc-review/togaf8cr/c/p3/ec/ec_ac.htm

‘How to grow a software business within a legacy company? You don’t’

Legacy companies are firms that rely on stable business models, established internal processes, and familiar technologies [Benassi 2024]. They often respond to the growth of software by solving the wrong problem and end up losing their markets.

Legacy companies view software often as an engineering problem. Everybody in software is busy to keep the ‘thing going’, whether it be a car, banking service or medical device. Typically, the market leaders in a legacy business are companies with many employees and large engineering teams. Large engineering teams tend to focus on big things like platforms. This is unfortunate since their core knowledge is more likely in applications, higher in the software stack. Typically, “A car company knows about driving experience” while “a TV company knows about video”. Legacy companies often only use their platform for their own products, so their volume will not exceed the number of legacy products they sell. An important question before engaging on a platform journey is how many of the current customers are interested to buy the platform from the same vendor from which they already acquire legacy products. Most B2B customers are by now aware of the value of their data and it is far from sure that they are interested to hand over their data via a platform to their legacy suppliers.

Achieving a higher volume for a legacy company would mean selling to their competitors or to another industry, which is likely not prioritized by the executive management. Building platforms for a legacy business often ends up in an engineering party: big budgets to solve interesting technical problems without much interaction with the market. There are many examples of these, among the best documented are Nokia with Symbian [Vuori 2016] and General Electric with Predix [Davenport 2018, Jacobides 2022]. Multiple examples of failed platforms that have not been documented can be found closer to Tilburg.

Considering the above, how to become a market leader in a world dominated by software? In a legacy company, a global market share of around 10% often puts you among the leaders. For example, Toyota is the market leader in the car industry, with a 12% global market share[10]. Market leaders in a software dominated world typically have

[10] https://www.focus2move.com/world-car-group-ranking-2025

much larger shares of the market, so the question is how to get from a 10% market share in a legacy product to a market share of 40 o 50 percent or more in the emerging software market in your business. The good news is that legacy companies can become successful in the software business. One of the best documented successful examples is Dassault Systems. It grew out of an aerospace company and became the world market leader in Computer Aided Design and Manufacturing (CAD/CAM) software. At the end of the 1960s, some engineers in Dassault Aerospace started using computers to smooth the curves on planes and wings like mechanical engineers could do on paper. By 1977, all Dassault planes were designed and manufactured with the help of computers using a software product called CATIA [Bernard 2024].

In the 1980s, the company faced a decision: "Should the company keep the product only internally, taking advantage of its benefits to maintain an edge over the competition, or should it create a new business to sell the software outside of the company? The first option turned out to be a dead end. There were only 15 CATIA developers on the team. How could Dassault afford to fund the hundreds of developers who would be needed in the future to address all of the necessary functionalities, even for one company? Eventually, another company would replace CATIA with a new product that Dassault Aviation would have to buy." [Bernard 2024]. Marcel Dassault, the owner of Dassault summarized it all in one sentence: "you should not build that software in my company because you will never sell it to Boeing"[11]

And the rest is history: Dassault Systems became independent and the world market leader in CAD/CAM. They were able to sell their software to Boeing and other aerospace companies, but they also ventured into other industries, such as the car industry. Both the aerospace and the software company are still public companies. The value of Dassault Systems exceeds that of Dassault Aerospace[12].

More companies these days are following a similar trajectory as Dassault Systems did. There are three economic lessons for legacy companies who want to succeed in software:

1. Make software invoiceable and get a profit and loss responsibility for the software business as a result.

---

[11] September 2025 discussion with Francis Bernard, first CEO of dassault Systems
[12] Data from August, 14, 2026

2. Create some form of separation between the legacy business and the software business. A separate go-to-market and the right to sell software to competitors of the legacy business is a necessary condition to increase volume and market share.
3. Focus on go-to-market and applications rather than on technology and platforms.

How about AI?

Will AI solve it all? Again, there is a lot to be excited about. The Nobel prizes for physics[13] and chemistry[14] in 2024 were awarded for AI research. Krugman, a Nobel prize winner in economics, warned that it will take years before we understand the economic and social consequences of AI[15]. So, I will not join those who claim they know the future of AI. As I stated before: "those who claim they can predict the future of AI, likely want to sell you something today"[16]. My statements about AI will be focused to my field of research, the impact of software.

AI is an interesting technology but the economical rules governing it makes it act like another form of software. A quote from Shapiro and Varian from 1999 that is very valid today: "durable economic principles can guide you in today's frenetic business environment. Technology changes. Economic laws do not." [Shapiro 1999]. The cost of an additional user is (close to) zero so volume wins and generic platform players (Bigtech) likely will grab a larger part of the software stack. AI As A Service (AIAAS) also comes in multiple layers [Tsaih 2023]. Focus is required to have a chance to compete in one of the layers. Again, define a software market and create a competitive layer of software instead of going after the whole software stack. This can be done starting from the Netherlands, as the Dutch start-up Weaviate has shown [Luijt 2020].

The European Union has the power of a large market but does not have much power as a supplier of software. An analogy with the automobile market: Bigtech is not only owning the new engine technology, but they also own all the roads (the cloud), the toll booths (to be paid in tokens), and the IP (proof in Nobel prize winners, patents, billions of lines of copyrighted software and very good engineering teams) and the go-to-market.

Look at the way that Bigtech is now sneaking AI on to our desktops. We are not asked whether we want AI in our solutions, it just shows up and we use it because it is convenient. Try to remove Copilot (by Microsoft) when it decides to get involved when opening a pdf (by Adobe) on your phone powered by Android (by Google). How many of us have gradually moved to the AI desktop of Google while we know that each AI question takes up to 10 times the electricity compared to an 'old' query[17]? And those

---

[13] https://www.nobelprize.org/prizes/physics/2024/hinton/facts/
[14] https://www.nobelprize.org/prizes/chemistry/2024/hassabis/facts/
[15] https://paulkrugman.substack.com/p/ai-in-an-age-of-oligarchy
[16] 'De toekomst van AI is onvoorspelbaar', interview in Financieel dagblad, 31 maart 2026
[17] https://sustainabilitymag.com/news/google-ai-search-uses-10x-more-energy-than-standard-searches

who now think ‘that is why I have an Apple phone’, again underestimate the importance of software and volume. Apple is a volume player with its iPhone in the smartphone market. Their market share in the US is 60 percent and in Europe it is 40 percent[18]. However, for AI the addressable market is not ‘high-end phones’ but ‘all devices we use to collect and organize information’. Suddenly the market share of Apple is a lot smaller. Apple has been struggling in AI for years and now gets its AI from Google[19]. Again, software volume wins, even against Apple.

AI is likely another weapon that allows Bigtech to grow further into the verticals. Software is executable knowledge [Humphrey 2001]. A lot of application knowledge in the verticals is tacit knowledge that has proven cumbersome to translate in software. AI may help. Bigtech stayed in the horizontal layers so far because there they could accumulate volume. AI may make it worthwhile for Bigtech to get into the verticals because the development effort becomes less and it would allow them to dominate a profitable niche. So legacy companies must watch their front – the space between them and their customers where AI start-ups likely enter. And they must watch their back where platform vendors may make the 90 degrees turn and enter their vertical.

[18] https://gs.statcounter.com/os-market-share/mobile data from August, 14

[19] https://www.nytimes.com/2026/01/12/technology/apple-google-ai-partnership.html

Impact of software on economies

Digitally delivered services represented 5.8% of world exports in 2024[20]. It is not trivial to assign software or digital services revenue to a country or a region in the world because digital firms have a preference to book their revenue in economies with a favourable tax regime [Stojkoski 2024]. Unlike companies providing physical products, they are not bound to the tax regime in the country where physical goods are produced. The digital product exports of the United States and Europe are comparable when one attributes the exports to the fiscal residency of the subsidiaries. However, the United States dwarves all other regions when one attributes the digital product exports to the headquarters location. The WTO[21] indicates that Europe is the only region that consumes more digitally delivered services than it exports to other regions in the world.

The impact of software on economies will keep growing for the foreseeable future. It is a profitable and competitive industry and Europe better grows its software industry. Having more competitive software products will be key. The fact that Europe is lagging has, for instance, also been extensively documented in the Draghi report[22] on the future of European competitiveness. The mobile phone industry can serve as an example what may happen to Europe if the software is provided by players from other regions in the world. Android took over from Symbian around 2010 and the European mobile phones disappeared. Japan saw their electronics industry shrink after ignoring software as a business for decades [Cole 2014]. Applying these historical lessons to AI solutions suggests future challenges for Europe. For example, Deepmind Alphafold (Nobel prize winning software, owned by Google) will have a significant impact on the pharmaceutical industry.

Software economics results in huge companies that only seem to want to grow faster, which may be good for them and their shareholders but not necessarily for their users. Take a hypothetical company with a market cap of US$1 trillion. To give a 5% return on investment to its shareholders means they must create US$50 billion worth of value

---

[20] https://www.oecd.org/en/about/news/press-releases/2024/05/growth-of-digital-economy-outperforms-overall-growth-across-oecd.html
[21] https://www.wto.org/english/res_e/booksp_e/trade_outlook25_e.pdf
[22] https://commission.europa.eu/topics/competitiveness/draghi-report_en

within one year [Genuchten 2019]. That is likely not possible within their own market because their market share may already be 70% and factored into their market capitalization. It is tempting to export that dominance to a vertical, which may be automotive, finance, defence, education or space. For comparison, the market cap of famous car companies and banks is typically tens of billions, less than a software giant must gain in market value in a year. Knowing that market leaders within vertical markets are already using their software in one shape or form (such as office systems, AI agents, cloud services, databases, or data mining), it is not that difficult to expand their dominance into a new market.

Market leaders are allowed to benefit from being the best in their market but antitrust law states they should not be able to use their size and impact to dominate a new market as well. That is why antitrust laws were created a century ago. Tim Wu [Wu 2025] shows that regular antitrust laws can play a role to limit the power of monopolists, like it did in the 1950 when AT&T invented the transistor and was at the time the monopolist for long distance telephony. They were forbidden to enter the semiconductor industry and the computer industry, which contributed to the rise of many new companies and the birth of Silicon Valley. It may take a few elections here and there, but applying antitrust laws to software may be needed to prevent the concentration of too much power against the interests of society.

What else can be done? Supporting European software companies that play in the world market makes sense. And there are promising European software companies. We [Berghout 2025] see two additional strategies.

Invest in high volume Regional Solutions

There will be increasing opportunities for high volume, regional solutions in the coming years. Well-integrated local applications may be more useful in a specific market than global solutions, examples are WeChat[23] in China and Grab[24] in Singapore. They have become market leaders in sizable markets at the expense of global players, likely with some help in their home market. An interesting recent example closer to home is an initiative by Amsterdam taxi drivers called NLCabs that charges significantly less for a

[23] www.wechat.com
[24] www.grab.com

ride than the global market leader[25]. Regulations around new digital technologies may also open opportunities for regional players. Take, for example, the AI and privacy regulations put forward by the EU that will make it more challenging to compete with a global solution in the EU market. Regional solutions for defence and cybersecurity may have a better chance in the current political climate. Some governments will not trust the current market leaders in cloud computing with their safety critical solutions.

Exploit the Fact There Is a Software Company Hiding in many Companies

Another way to grow a software business in a region is rooted in the fact that 'every company is a software company' and there are many software companies hiding in legacy companies. Europe had 33.5 million enterprises in 2024[26]. Fifty-five thousand enterprises with more than 250 employees generated over half of the net turnover of the EU business economy. The 250,000 companies with between 50 and 250 employees represent another 17 percent of the EU business economy. The proposal is therefore to release software businesses from their legacy parent companies [Berghout 2025], allow them to take care of their own business, and let 100,000 flowers blossom, hoping that this leads to tens of thousands of successful companies that strengthen the software industry in a region. Not all will become market leaders but being able to focus on software and the markets gives them a chance. Some standalone software companies may be sold a few years later to the market leader in the emerging software market. That is still a better way to protect the interests of the stakeholders in the legacy company than pumping money into proprietary software stacks until it is too late.

This is no quick fix, but we must realize it took Europe some decades to dig the software hole we are in. So, fixing it will take time. To give an idea of the magnitude of the challenge: in terms of market capitalization, the EU would need around 100 Dassault Systems or seven SAPs to compensate for Microsoft or Google. Seven ASML's would also do.

For those who think this proposal is too radical it is worthwhile to consider the alternative: hope that European legacy companies will figure out a way to become competitive in the software business while trying to defend their legacy business at the same time. In the words of Deming: "Learning is not compulsory; it's voluntary. Improvement is not compulsory; it's voluntary. But to survive, we must learn"[27].

---

[25] https://www.dutchnews.nl/2026/09/amsterdam-taxi-drivers-launch-app-to-rival-uber/ accessed on September, 19, 2026

[26] https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20251209-2

[27] https://deming.org/quotes/learning-is-not-compulsory-its-voluntary-improvement-is-not-compulsory-its-voluntary-but-to-survive-we-must-learn-3/

There is no reason to let software eat our world

What can we do to prevent software from eating the world. Some modest pointers for scientists, educators, government and policy makers.

For scientists interested in economics and impact on society

The Tilburg Law and Economics Center[28] [Prufer 2026] brings together law and economists working on governance in the digital age. Some suggestions for research. First, the growth rate of software seems to be independent of experience of the engineers, technologies or programming languages used and application area. We have a hunch that teams cannot model more than an additional 20 percent of the reality around them per year. AI may change that but quantifying the growth rate using AI would be useful to predict the growth rates in different domains and help organizations to earmark software domains where they have a chance to catch up.

Second, more empirical studies like Dassault should be documented. We were able to publish a few in IEEE Software [Avery 2011, Schaminee 2011] but we need more studies from different organizations and industries.

Third, more empirical research is needed into successful regions in terms of software economics. An interesting region are the Nordics that are net exporters of software. The software companies now comprise 6% of the gross domestic product, and the software supply is about double the software demand[29]. On the other side of the world, New Zealand managed to export $NZ 1.7 billion worth of software with a population of five million inhabitants[30]. What do they do different and what can we learn from them?

For educators

At TIAS we educate digital leaders of the future. Suppliers and customer of software learn together in an executive masters' study. A growing number of participants come from different backgrounds, disciplines and very different organizations such as

---

[28] https://www.tilburguniversity.edu/research/institutes-and-research-groups/tilec

[29] https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/whats-driving-the-nordic-countries-software-export-surge

[30] https://technewzealand.org.nz/wp-content/uploads/sites/8/2023/08/Tech-Sector-Key-Metrics-2022.pdf

defence, software companies, hospitals, industry, banks and ASML, to name a few. Digital economics are an important part of the education of the digital leaders of the future.

Education about software and IT does not stop with the future digital leaders of organizations. The economic motivation of software companies to keep us glued to screens is successful for them but we need to educate ourselves and the next generation to balance our IT diet [Rutkowski 2013]. Education is in the long run the way to do so given the power of algorithms and economic interests involved[31].

For government and policy makers

Do not get distracted by chasing political desires that do not match engineering realities. Europe has been trying to build European clouds since 2010 and the sobering result is that AWS, Google and Microsoft now have 72 percent of the cloud market in Europe [Calcara 2026]. Also, the 'all or nothing' discussion about sovereign cloud is not very useful. The software stack is too big, and we are too far behind. An analogy: an athlete may want to run a half-marathon within two hours but will fail if he/she did not train for 20 years and shows up at the start an hour late. European defence officials and defence contractors estimate that "that the US had an eight- to 10-year lead in cloud and AI defence platforms" [32].

Focus on the software applications where we can achieve a dominant market share and use the market power to make it happen. Europe is the biggest market so Bigtech wants to be here and will adjust to our rules. An example of a more realistic approach towards a European cloud is Thales that now offers the Google cloud under their governance for major European markets[33]. Thales is a major defence and security player so they must be able to provide a workable solution in an area where Europe needs more independence.

The concentration of power and money in software companies will continue. More sectors of the economy will be eaten by software. There are few limits to growth for big software companies, as the current market capitalization of the Bigtech shows. Governments and policy makers should find ways to redistribute some of the power that is accumulated in too few companies and individuals [Wu 2025]. For Europe, it should however start by providing more credible competitors in the global software market.

Software and its companies may grow beyond what is healthy for society. Aristotle[34] already saw the unchecked concentration of power as a threat to justice and stability.

---

[31] https://www.tristanharris.com
[32] https://www.ft.com/content/0e9d714b-f367-4dd7-8ef4-6d4b2c08b79c?syn-25a6b1a6=1
[33] https://www.reuters.com/business/aerospace-defense/thales-inks-deal-with-google-homegrown-german-cloud-provider-2026-05-20/
[34] https://classics.mit.edu/Aristotle/politics.html

Continued growth of major software companies can be considered unchecked concentration of power. Fast forward from Aristotle to the New York Times of August 2026: “Jobs could be lost. Industries could be destroyed. Natural resources could be drained. Entire communities could be reshaped by AI, with little guarantee that the public would be protected.”[35]

Democracy and independent press are important pillars of the Western world. Antitrust legislation is in place to fight concentration of power within the hands of too few companies and individuals. Let’s use them.

There is no reason to let software eat our world.

[35] https://www.nytimes.com/2026/08/16/business/ai-data-centers.html

Epilogue

I would like to thank the executive board of Tilburg University and the TIAS business school for naming me in this position.

My students at TIAS have experience working in practice and every organization is now a software organization. The fact that many of you have been able to apply the theory within your organization is rewarding for all involved. You have experienced what Einstein once stated: “In theory, theory and practice are the same. In practice, they are not.” I also want to thank the lecturers in my courses that they make the time in their busy agenda to make their knowledge and experience available.

Piet Ribbers started the MIM executive masters 40 years ago at TIAS. We worked together a lot in 2025. Piet kept contributing to his executive masters by lectures and master’s supervision until a few weeks before he passed away in March. Piet, I enjoyed working with you and I will take care of the masters.

Theo Bemelmans, my PhD advisor in a previous century. Theo, it is interesting that I end up at the university where you wrote your PhD. In the mean-time you supervised my masters and my PhD and we were colleagues in Eindhoven. I am glad we still regularly meet and I keep learning.

Les Hatton, we started 18 years ago with the “Impact of software” in IEEE Software. We knew it would take time to better understand it but would not have predicted we would keep it going for so many years. I appreciate our collaboration and our discussions about other topics than software.

Family and friends. Thanks for being there. For some of you it is the second time you take the time to listen to an inaugural speech of mine. It is appreciated, as is your love and friendship. For my four children: you will find your way in the world and do not get too distracted by IT, nor by me.

Dear Anne-Francoise, we met at the TU in Eindhoven and now we end up here in Tilburg together. Let’s be clear; this is your university and you are the true scientist. I am proud of you. In one word that works in all our languages: you are formidable.

I have spoken. Ik heb gezegd.

REFERENCES

Ackoff, R. A revolution in organizational concepts, Naval War College Review, Vol. 24, No. 5, second international seapower symposium (January 1972), pp. 4-14.

Avery, D. "The Evolution of Flight Management Systems," in *IEEE Software*, vol. 28, no. 1, pp. 11-13, Jan.-Feb. 2011

Benassi, Mario, and Riccardo Rialti. "Exploring the metaverse from a legacy company perspective: a capabilities-based view." *California Management Review* 66.4 (2024): 51-79.

F. Bernard, "Dassault Systèmes: A CAD Success Story," in IEEE Annals of the History of Computing, vol. 46, no. 4, pp. 124-135, Oct.-Dec. 2024.

Bouchard, L., M. van Genuchten, “How to grow a software business within a legacy company: You don’t,” IEEE Software, vol. 42, no. 4, pp. 79–83, Jul./Aug. 2025,

Calcara, A, “European cloud computing policy: Failing in Europe to succeed nationally?” West Eur. Politics, 2026.

Campbell-Kelly, Martin. From Airline Reservations to Sonic the Hedgehog — A History of the Software Industry. Cambridge: MIT Press, 2003.

Cole, R. and Y. Nakata, “The Japanese software industry: what went wrong and what can we learn from it?” California Manage. Rev., vol. 57, no. 1, pp. 16–43, 2014.

Cusumano, M & Yoffe, D. & Gawer, A.,. (2020). “The Future of Platforms”. MIT Sloan Management Review, January 2020.

Davenport, T., George Westerman, Why So Many High-Profile Digital Transformations Fail, Harvard Business Review, March 9, 2018

Fisher, F.. "The IBM and Microsoft cases: What's the difference?", *American Economic Review* 90.2 (2000): 180-183.

Gal, R, van Genuchten, “Release the Embedded Software; the Electronics Industry in Transition,” Int’l J. Technology Management,June 1996, pp. 33-44

Genuchten, M. “The impact of software growth on the electronics industry,” Computer, vol. 40, pp. 106–108, Jan. 2007,

Genuchten, M., and L. Hatton, “Software: What’s in it and what’s it in?” IEEE Softw., vol. 27, no. 1, pp. 14–16, 2010.

Genuchten, M., and L. Hatton, “Software mileage,” IEEE Softw., vol. 28, no. 5, pp. 24–26, 2011.

Genuchten, M., and L. Hatton, "Ten Years of "Impact" Columns—The Good, the Bad, and the Ugly," in *IEEE Software*, vol. 36, no. 6, pp. 57-60, Nov.-Dec. 2019

Graham, B., Dodd, D. Security Analysis, Whittlesey House, McGraw-Hill Book Co., 1934

Holzmann, G., “Published, landing a spacecraft on Mars,” IEEE Softw., vol. 30, no. 2, pp. 83–86, 2013.

Hatton, L., Rutkowski, A., “Lessons must be learned—But are they?”, IEEE Software vol. 36, no. 4, pp. 91–95, 2019.

Hatton, L., Spinellis, D., and van Genuchten, D., “The long-term growth rate of evolving software: Empirical results and implications,” J. Softw., vol. 29, no. 5, 2017.

Humphrey, W.S., Winning with software: An executive strategy. Pearson Education, 2001

Jacobides, M.G., How to Compete When Industries Digitize and Collide: An Ecosystem Development Framework, California Management Review, 2022, Vol. 64(3) 99 –123

Luijt, B. Van, Verhagen M., "Bringing Semantic Knowledge Graph Technology to Your Data," in *IEEE Software*, vol. 37, no. 2, pp. 89-94, March-April 2020

OECD, “Handbook on measuring digital trade,” 2nd ed., OECD, Paris/Int. Monetary Fund/UNCTAD, Geneva, Switzerland, 2023.

Prufer. J. We are living in interesting times, Inaugeral speech at Tilburg university, June, 2026.

Schaminée, H., and Aerts, H., "Short and Winding Road: Software in Car Navigation Systems," in *IEEE Software*, vol. 28, no. 4, pp. 19-21, July-Aug. 2011

Shapiro, C., and Varian. H. *Information rules*. Vol. 14. Boston: Harvard business school press, 1999.

Stojkoski, V., Koch, P., and Coll, E., “Estimating digital product trade through corporate revenue data,” Nature Commun., vol. 15, no. 1, 2024, Art. no. 5262.

Tsaih, Rua-Huan, et al. "The AI tech-stack model." *Communications of the ACM* 66.3 (2023).

Vuori, T, and Quy N. "Distributed attention and shared emotions in the innovation process: How Nokia lost the smartphone battle." Administrative science quarterly 61.1 (2016): 9-51.

Wu, T, The Age of Extraction: How Tech Platforms Conquered the Economy and Threaten Our Future Prosperity. New York: Alfred A. Knopf, 2025