



# Looking into the Future

Christof Ebert

No matter what business you're in, you're also in the software business. But where are we heading? This 50th installment of Software Technology tries to look into the future. To mitigate bias, I spoke with many software business leaders around the world. They pointed to five success factors that will advance the software business. They left unaddressed whether software will evolve humankind to Humanity 2.0—or a posthuman society. Read more in this column and let me know your opinion and own stimulus. I look forward to hearing from both readers and prospective column authors about this column and the technologies you want to know more about. —*Christof Ebert.*

**SOFTWARE MAKES** the world go round—at ever-increasing speeds. From the embedded software in cars, avionics, and automation to the ubiquitous computers in smart phones and consumer electronics, Vector Consulting Services' annual industry benchmark performed with different companies from various industries helped provide a sound quantitative baseline.<sup>1</sup> All the companies are in the software business, either directly

The companies include ABB, Bosch, GE, Google, Lufthansa, Philips, and Siemens, among others.

## The State of the Software Business

The software business is different from traditional industries.<sup>2,3</sup> Industry experts and leaders both agree that we are heavily confronted with the two faces of our industry—innovation and complexity:

- Software attracts more investment than other industries, which is making its market valuation grow faster than any other industry, thus continuously accelerating innovation.
- Creativity matters, whereas production and logistics are of limited relevance and incur marginal costs, which makes it easy to enter a market and sell globally.
- Software is flexible and easy to change, which results in a high frequency of products and

We are confronted with the two faces of our industry—innovation and complexity.

software provides features and functions in daily use. Our society depends on reliable software systems, which depend on innovative companies, which depend on competent software engineers.

In preparing this column, I spoke to business leaders and industry experts to get an impression of what they see as the future of software. Vec-

tor Consulting Services' annual industry benchmark performed with different companies from various industries helped provide a sound quantitative baseline.<sup>1</sup> All the companies are in the software business, either directly with IT solutions or indirectly with products in which software is critical for value creation, such as embedded systems, or value delivery, such as services. One-third of the companies have more than 100,000 employees worldwide, half have 10,000 to 100,000 employees, and the rest are small and medium-sized companies with approximately 1,000 employees.

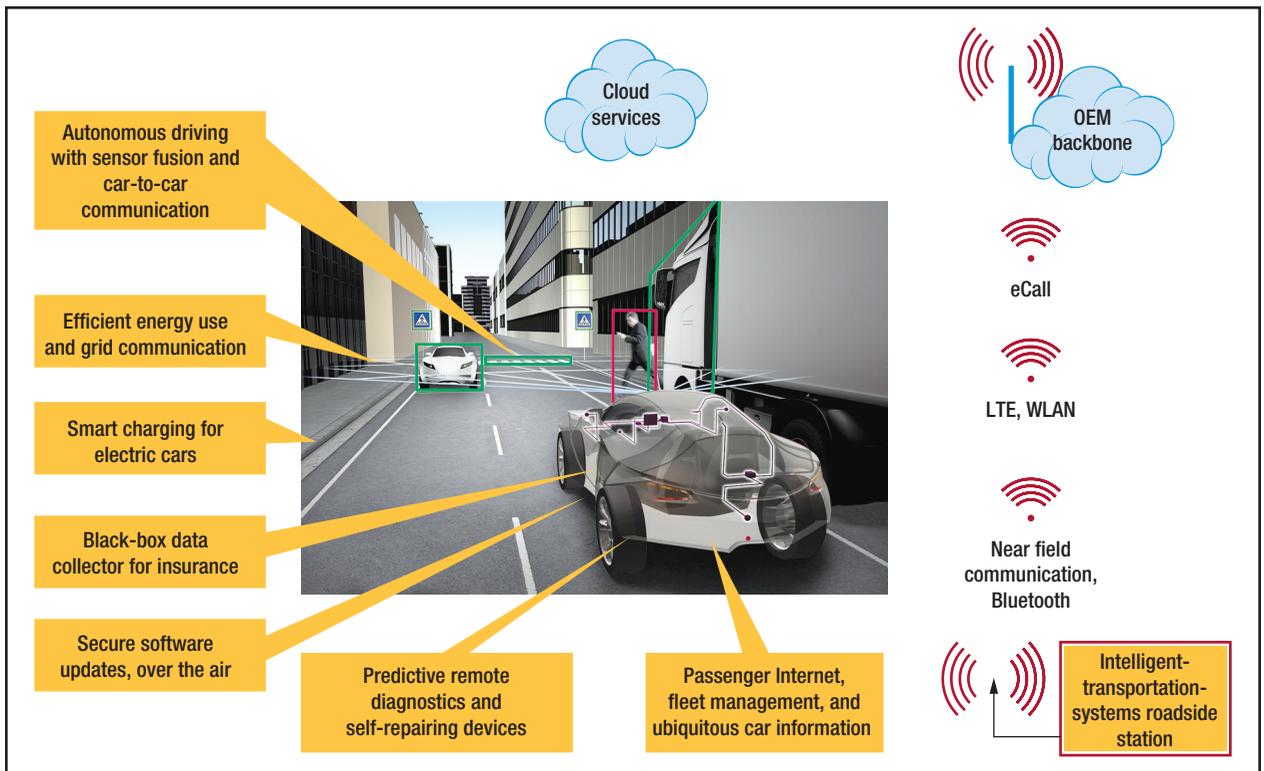


FIGURE 1. Software innovation fuels automotive advances, from smart energy efficiency to autonomous driving.

releases that don't always create real value.

- The overwhelming complexity of technology and products, combined with insufficient competences, severely cannibalizes the quality of software-driven products.

One impression clearly dominates. Software is changing practically all industries and is the major driver of innovation across all industries. While we used to distinguish components, systems, and services, we today see flexible boundaries driven entirely by business cases to determine what we should package, at which level, and in which component, whether it's software or silicon.

Take consumer and communication systems. A TV in the 1970s had no software, whereas today its com-

petitive advantages are software-driven. Or consider aerospace systems. In 1960, only 8 percent of the F-4 fighter's functionality was implemented in software; by 2000, software provided 80 percent of the F-22's features.<sup>2,3</sup> Today we see an increasing number of military aircraft operating as autonomous drones.

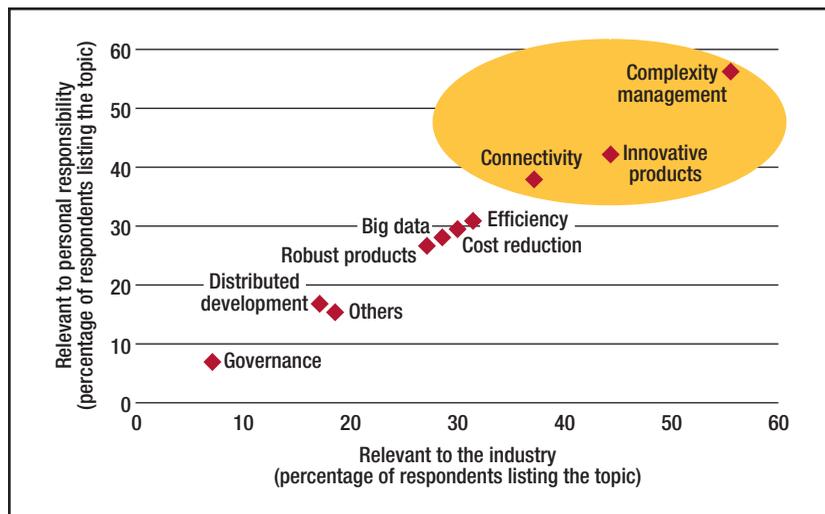
The automotive industry is following the same pattern. Modern cars contain more than 100 embedded computers, with more software than any airplane. Cars sell primarily because of their software-based functions, whether those functions involve the power train, energy efficiency, an electric-car platform, or communication facilities for driver assistance, safety, diagnosis, and so on. Autonomous driving is being introduced, thus moving the industry to a primarily software business.

Not surprisingly, the North American International Auto Show recently featured the slogan that a car is a microprocessor with wheels and an engine (see Figure 1).

### The Future of Software

Five dimensions characterize the future of software:<sup>3</sup>

- *collaboration*—for example, consumer Internet, social networks, single-customer segmentation, product and service configurators, digital money, computer-assisted collaboration, and crowdsourcing;
- *comprehension*—for example, semantic search, big data handling, smart data, data analytics, data economy, online data validation, and data quality;
- *connectivity*—for example,



**FIGURE 2.** Industry trends.<sup>1</sup> Innovation, complexity, and connectivity are the major issues for business and technology leaders. The sums are greater than 100 percent because the survey allowed three answers per question.

ubiquitous mobile computing, mobile services, cyber-physical systems, Industry 4.0, machine-to-machine communication, sensor networks, and multisensor fusion;

- *cloud*—for example, applications and services in the cloud, location-based networks, new license models for software and applications, sustainability, and energy efficiency; and
- *convergence*—for example, mobile enterprises, bioinformatics, the Internet of Things, pervasive sensing, and autonomous systems.

These dimensions, coupled with the underlying complexity and scale, demand new software solutions based on new computing paradigms and infrastructure. Examples include IT architectures that facilitate seamless connectivity, robust infrastructures for cyber-physical systems in safety-critical environments, and data analytics to predict choices and

behaviors to improve the customer experience. Such software-driven solutions can create nontraditional market entry points and consequently new mechanisms to provide each customer time-specific and location-specific services.

What are the dominating expectations and needs from a software business perspective? From our benchmarks with companies, we found they'll continue to invest in growth through innovation by developing new products and solutions because this determines their market position. At the same time, they are aware of the volatile market and are thus trimming their development teams worldwide to be as lean and as efficient as possible.

Figure 2 provides a recent aggregation of industry trends from our annual business survey.<sup>1</sup> In that anonymous survey, we asked managers from different companies, industries, and hierarchy levels what will be relevant in the near future for their own industry and for their

own responsibility. They could select from different topics and add their own topics.

Most industry leaders push for more innovation—balanced with a clear need for not only more efficiency and cost reduction but also product robustness. Obviously, the complexity caused by continuous innovation, specifically in the software market, must be balanced with more rigid instruments to keep business processes lean and products robust. The evolution from our preceding surveys is interesting. In previous years, software companies focused on cost reduction and efficiency. Now, with full order books and a positive outlook, they're focusing on managing homegrown complexity.

### Software Risks

Success in the software business isn't a given. Products and solutions must meet increasing quality requirements but must be designed to be inexpensive and easily adaptable and to exploit modern platforms' advantages. New competitors are entering markets with new solutions, without being hampered by legacy systems and traditional business models. The technology landscape has become increasingly complex.

In general, the interviewees revealed four key obstacles to success:

- a dysfunctional organization with unclear responsibilities and silo work, which results in continuously changing focus and schedules;
- lack of strategy or an unclear strategy and roadmaps with unclear dependencies and fuzzy technical requirements and impacts;
- no standardized business processes across the company, with

slow, cumbersome decision making and many individual ad hoc agreements; and

- insufficient requirements that are often just collections of what was heard at customer visits and other such events and that haven't been mapped to value creation and business cases.

Innovation, globalization, and complexity are fueling the software business worldwide but also creating many risks that can easily endanger or kill a business. So, the software business has many challenges, ranging from the creation process and its inherent risks to direct balance sheet impacts.

Software is getting more complex, more connected, and more life-critical. This complexity's sources are hidden in the nature of software, which often consists of many components from different vendors and runs on hardware manufactured by different vendors. Also, software teams frequently are multifunctional, and team members are responsible for many activities such as planning, developing, and executing plans, roadmaps, and strategies—without adequate training. This differs from other industries in which, for instance, production and logistics are strictly separated from development. Even within development teams, the people who design a product or component are often those who test it. No wonder that quality is often below expectations and that specifications are inconsistently implemented.

### Stimulus for the Leading Practitioner

How do software companies find appropriate starting points to manage complexity? How do they bring innovations to the market faster and

improve connectivity? Which techniques have been proven in practice by successful companies?

Here are some concrete experiences and advice from Vector Consulting Group's projects around the world. Looking at the articulated challenges on one hand and the successful companies on the other hand, we found five success factors that will help companies master future challenges. These factors aren't comprehensive but can help companies focus on what really matters.

#### Focus Where the Money Is

Products are pushed to the extreme to be ever more efficient and low cost, but they don't always sell as expected. Customers demand many changes, thus reducing margins dramatically from the initial targets. Many managers with whom we talked tend to concentrate on what they know best, which too often in the software industry are technical topics, rather than basic business principles. Targets on each level

used. Each company and software team can introduce these two techniques to focus on the right things and balance short-term survival needs with medium- to long-term strategy evolution, investments, and improvements.

#### Sell Value Rather Than Features

When we work with clients on product strategy and requirements engineering, we first ask, what market, what user, and what usage? The overwhelming number of responses can be reduced to, "The spec says so." Now this might be formally true, but it will lead to a downward spiral of complexity and cost.

Although complexity sells, it must be balanced with other factors. Complexity creates much extra cost for service, evolution, variance management, and regressions along the life cycle. We often face clients who don't really know how to effectively control complexity. They trap themselves with slogans such as "We're too expensive," rather than nailing

Industry leaders push for more innovation—balanced with efficiency and product robustness.

must be set up consistently and monitored continuously against actual performance. Changes in the business climate must be managed, or they ripple through uncontrolled.

On the senior-management level, we recommend a balanced-scorecard approach, whereas on the operational level, value-driven steering, such as earned value management, should be

it down to concrete levers they can directly address. Around half of all requirements don't create marginal value but contribute to cost.<sup>2</sup> Variants and specific features for individual customers further increase complexity. Steve Jobs was one of the few taking concrete lessons from this principle, demanding simplicity in his products.

Our Vector RACE (Reduce Accidents, Control Essence) method provides the tools to effectively manage complexity. Reducing accidents means cutting overheads such as gold-plating and rework due to late defect removal or too many requirements changes. Controlling essence enforces looking to what customers really pay for. Each requirement must be justified to support the

represent different external perspectives. The success factor is to give this core team a clear mandate to own the project.

Apply adequate risk management techniques to make your portfolio and commitments dependable. Projects might need more resources, suppliers could deliver late, or technology won't work as expected. Replace traditional labor-cost-based location

decisions are made without considering business case and downstream impacts.

Automated product life-cycle management (PLM) and application life-cycle management (ALM) are the primary mechanisms for efficiently integrating engineering processes, tools, and people across the domains of system, software, hardware, and mechanical engineering. Process shouldn't mean a burden and documentation, but efficiency in repetitive activities. Life-cycle management ensures governance and thus provides the safety net that almost all businesses need, not only vis-à-vis their owners and investors but also in the market for legal reasons.

It's up to today's software engineers how to embed software in humans and how to embed humans in software.

business case and to allow managing changes and priorities. Ask a tester to write a test case before processing the requirement. Ask the team's marketer to check whether he or she can sell the feature as described. Determine what's good enough, and ensure that any further insight is adequately considered. With all that focus on value, ensure that agile and lean don't lead to arbitrariness.

### Manage Relevant Stakeholders

Stakeholders matter in both small and big companies. Often they follow their own agendas, optimize locally in their silos, and don't actively own and drive the company and its products.

So, we strongly recommend creating a core team with the product, marketing, project, and operations managers for each product (release) and making it fully accountable for the product's success. These persons not only represent the major internal stakeholders in product or solution development but also sufficiently

decisions with systematic improvement of business processes in a distributed context. The benefits will be tangible, as our clients emphasize: multisite collaboration, clean variant management, and transparent workflows are the most common reported benefits. Replace looking merely at labor cost with a holistic strategy taking into account onsite presence, customer ecosystems, and reduction of friction losses.

### Master the Life Cycle

Software products have the same liability requirements as any other product, but their suppliers often fail to establish the necessary life-cycle processes. All product releases and projects should follow a standardized product life cycle. Most companies have defined such a life cycle but rarely use it as the pivotal tool to derive and implement decisions. Requirements changes are often agreed on in sales meetings without anyone checking feasibility, and technical

### Continuously Improve

Nearly 90 percent of our interviewees claimed that their companies want to improve performance, but only a third of them were satisfied with the results. The same held for improving their own competences and behaviors. Insufficient change management is a common reason for failing.

Cost and cycle times can be reduced when you're working with continually optimized development processes. Lean development and agile approaches are useful for streamlining interfaces and reducing rework and inefficiencies. Earned value management, value stream mapping, and Scrum are proven techniques that can easily be tailored to organizations. Streamline workflows and related tools stepwise, with an overarching strategy, incremental goals, and a future-oriented IT architecture. Set concrete improvement targets on a quarterly basis.

Train employees in lean development. Have each team develop an action plan for reducing waste, rework, and interface conflicts—with

reference to your company-wide efficiency targets. Train people to grow technical and—even more relevant—soft skills. Evaluate performance—for example, by sales per developer, lead time, fault detection rates, cost drivers, competences, and innovations. Apply professional change management.

**O**ur business climate will remain volatile. At the same time, complexity and technology will quickly grow. The resulting competence gap will lead to an even stronger fight for skills. As we mentioned before, companies will continue to invest in growth through innovation and will require development teams to be lean and innovative. Let's look briefly at what the next 30 years might hold.

Virtual environments and augmented reality have set the path toward software directly creating our own reality. Many kids today can't imagine life without software-driven games and embedded AI. Many people see games and social networks with many so-called friends as more realistic and more appealing than our classic Humanity release 1.0. Direct interaction with our nervous system will advance such virtual reality toward real reality, or Humanity 2.0. The impact on society will be manifold.

For example, consider cars, transportation, and their negative impact on the environment. If we can meet people at any place without actually travelling, pollution and accidents will be very much reduced. With blood-cell-sized devices in our bodies fighting against diseases and improving our intellectual and cognitive abilities, life will fundamentally change.<sup>4</sup> Software will extend the brain the same

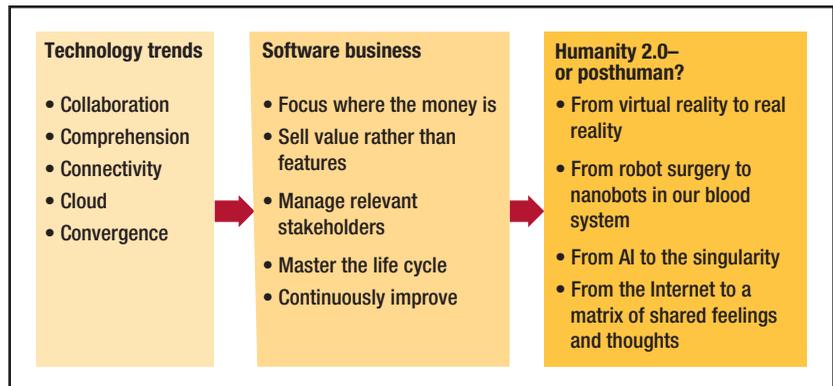


FIGURE 3. The way ahead—Humanity 2.0 or a posthuman society?

way it has extended our behaviors today. Our computers will successfully pass the Turing test and soon after achieve the singularity.<sup>5</sup>

What used to be civilization in the classic sense of humanitarianism will become increasingly dominated by nonbiological and posthuman intelligence. The Internet will evolve to a global network connecting billions of people and machines to share data, information, and eventually thoughts and feelings. All this will dramatically affect how we live, behave, and evolve not just as individual humans but also as societies and as humanity. Software will be at the center of this change, so we'd better shape today the necessary competences to do it the right way, which is more about values and soft skills than technology.

Humanity 2.0 will be fueled by the software technology we build today, as we innovate processes and products, while their development, production, operation, and service are increasingly done by machines. In doing so, we'll advance humanity—hopefully for the better. Figure 3 illustrates these major trends but leaves one key question unanswered. Will it be a Humanity 2.0 with better quality of life, or will it be a posthuman

environment? It's up to today's software engineers how to embed software in humans or how to embed humans in software, while preserving a society of humans who act human. 

## References

1. "Industry Trends 2015," Vector Consulting Services, 2015; [www.vector.com/trends](http://www.vector.com/trends).
2. C. Ebert and S. Brinkkemper, "Software Product Management—an Industry Evaluation," *J. Systems and Software*, vol. 95, 2014, pp. 10–18.
3. C. Ebert, G. Hoefner, and V.S. Mani, "What Next? Advances in Software-Driven Industries," *IEEE Software*, vol. 32, no. 1, Jan. 2015, pp. 22–28.
4. E. Brynjolfsson and A. McAfee, *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*, Norton, 2014.
5. R. Kurzweil, *How to Create a Mind: The Secret of Human Thought Revealed*, Penguin, 2013.

**CHRISTOF EBERT** is the managing director of Vector Consulting Services. He is on the *IEEE Software* editorial board and teaches at the Universities of Stuttgart and Paris. Contact him at [christof.ebert@vector.com](mailto:christof.ebert@vector.com).



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.