

THE FUTURE OF EMBEDDED SOFTWARE

ARE WE READY?

FEATURING

PROFESSIONAL STANDARDS AND ETHICS

Who's responsible for software-related tragedies?

THE TRUE COST OF SOFTWARE

Paying for software decades after we've written it

DESPERATELY SEEKING TALENT

Why we need more than just educated engineers





Table of Contents

4 PROFESSIONAL STANDARDS AND ETHICS

Software engineering doesn't have professional certifications, liability, or a code of conduct. This needs to change. Now

14 DATA PRIVACY AND OWNERSHIP

The misuse of personal data is eroding our personal rights and our democracies — could the software industry have the solution?

8 THE TRUE COST OF SOFTWARE

From automation job losses to software quality ramifications, our experts explore the price society will pay for software beyond the price tag

17 DESPERATELY SEEKING TALENT

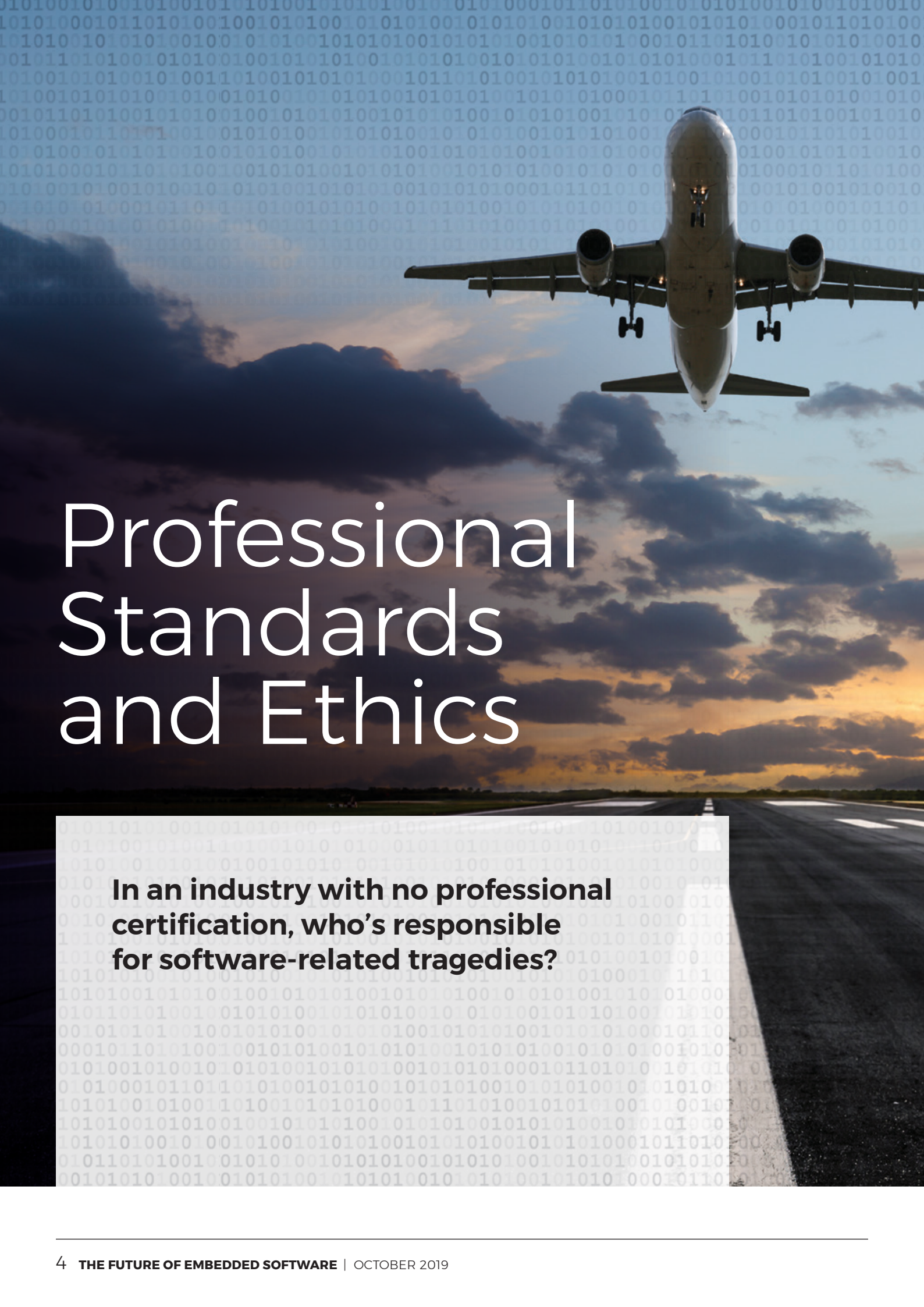
Software skills are in hot demand; we look at why we're not effectively finding the right people to get our products to market

11 CYBERSECURITY ARMAGEDDON

The short-term focus is on hacks and how to fix them, yet the problems exposed by security vulnerabilities go very deep

20 THE SEAMS OF INTEGRATION

Making software into reusable building blocks was last century's holy grail; now that they're here, they've introduced new unanticipated problems



Professional Standards and Ethics

In an industry with no professional certification, who's responsible for software-related tragedies?

Remember the days when electronic devices were novelties? Those days are ancient history. There has been an explosion in embedded software, and our lives are full of digital devices: from telephones, washing machines, and thermostats, to cars, medical devices, and airplanes. The technology has advanced to the point where we no longer marvel at the thought of a self-driving car, but pause to consider whether we may one day need a mechanic with a computer science degree to fix it when it breaks down.

What type of standards exist to ensure these products are designed with quality in mind? Do embedded software developers have a responsibility to the greater good? What if one of these products doesn't function as advertised? How easily are they fixed — and who's responsible? A washing machine breaking down due to poorly designed code is one thing, but what if the same applied to a car — or an airplane? Who is responsible when an accident causing injury or death can be traced back to embedded software?

Till Adam, chief commercial officer at embedded software consultancy KDAB, says it's high time for a serious conversation on the state of computer science ethics and professional standards for developing computer software. "Doctors and lawyers can't practice their trade without minimum standards," says Adam. "Society has recognized that if they fail to do their job properly, people can lose huge sums of money, can have their health threatened, or even die."

Adam considers us to be at the very same inflection point with software right now — what we do as an industry has high risks and can impact entire nations. And it's not just medical software that has this problem. As a simple example, if you program a voting machine, how does anyone know you've followed established, documented standards, and that those machines are actually guaranteed to reflect the wishes of the electorate? Adam believes that we've massively underestimated how many embedded systems there



Safety-critical software needs certification, not just of its development processes, but of its developers.

are in the wild that impact the security and integrity of intricate systems which our society relies on.

Lars Knoll concurs. Knoll, chief technology officer at global software company The Qt Company, says we need a different approach to building software. "You can't just build things to get to market quickly by taking every shortcut," says Knoll. "Development for the long term requires different engineering. Maybe it will take a hit in terms of speed in the beginning as you develop a continuous integration architecture, test scaffolding, maintainable software, and documentation, but people have to recognize that starting with a disciplined approach gives you a longer-term advantage."

Matthias Kalle Dalheimer, KDAB president and CEO who simply goes by Kalle, laments about another "skill" not found frequently enough. Increasingly his company is looking for people who have been exposed to the consequences of technology. "Software engineers are taught how to build things but aren't shown what happens once those things are released into the world," explains Dalheimer. "There needs to be an ethical component to any computer science degree taught today."



We wouldn't consider building a bridge with advice from a forum, tools off the street, and free code from GitHub.

A move to regulate accountability may be inevitable and Dalheimer says we may need to take our cues from other industries. "Civil engineers have a code of conduct that enforces certain standards and commitment to quality," he adds. "There's a minimum bar in many industries that's lacking in software."

Dalheimer also points out that while mandated software development standards are a way to force accountability, they also protect engineers. Just like there are clear rules for malpractice lawsuits, regulations can shield engineers if they follow the rules properly.

Jared Spool, founder of User Interface Engineering (UIE) and co-founder of Center Centre design school, builds on this cross-discipline thought, reminding us that building codes were initially created because many people were dying in hotel fires. If people die in a fire today, the National Fire Protection Association does research, testing, investigation, simulation, and develops new guidelines, processes, and materials — like adding giant fans to pressurize the stairwells or using big doors that slam shut behind you. Each accident drives more improvements in the building code. The architect is in charge and if there are fatalities due to their negligence, they lose their license. Conversely if building owners try to cut costs, architects won't sign off on their plans, which puts a freeze on building permits. Unfortunately, Spool believes we're going to see some terrible consequences from software failures just as we

did from poorly constructed buildings before anything changes in computer science.

Adam agrees that the problem will get worse before it gets better: "Unless or until there is a catastrophic incident, software won't be regulated. Sadly, there will need to be death and destruction before anything changes because this lack of accountability is a hidden problem that the general public knows nothing about, and the cost to do anything about it is perceived to be too high."

Spool feels that certification bodies for embedded are coming. "We need to be at a place where things don't go to market without signoff from a licensed professional," he contends. "The models we've developed aren't working as things continue to proliferate — licensing will need come from within the industry and we're going to see this happen within the next 10 years."

Adam agrees that this is a self-policing problem. "Just like other industries have done, software people are going to have to institute licensing on themselves as a means of self-defense," he believes. "They're the only ones who understand software and what it's capable of — if they wait for governments to do it, nobody will be happy with the result."

Adam is also concerned by the level of professionalism in most commercial software: "Would you build a bridge with advice from a forum, tools off the street, and free code from GitHub?"

Probably not — the consequences from the bridge crumbling are too high. Yet people with no understanding of the impact of software are the ones building mission-critical products and this dynamic is being fueled by the rapid growth of VC money. “Nobody is building with longevity in mind,” adds Adam. “Meanwhile, the brightest minds of our generation are building apps so we can take pictures of food.”

Perhaps some of the challenge stems from engineers who know there are problems but don’t have enough clout at the organizations they work for. Dalheimer explains that experienced and competent engineers know how to do things right, however they’re often hampered with too little budget, too few resources, and timelines that are too aggressive. They aren’t able to build the quality, security, or reliability they know is needed because they can’t push back on their management. “If engineers had to swear an oath that they’ll always create good software and if the industry had a professional ethos behind it, they’d have more leverage to do what’s right,” he adds. “Professional integrity comes from the status and respect of the craft.”

As a recent example underscoring this point, Adam references the Volkswagen Dieselgate scandal, where software was written to recognize an ongoing emissions test to provide fraudulent results. “Automotive already has strict software standards and regulations in place,” he says. “So theoretically Dieselgate shouldn’t have happened.”

Although it would likely only have taken a very small team of engineers to write the software that illegally subverted regulations, anywhere from 10 to 100 engineers would have touched the system during integration, maintenance, testing, and validation. Although it is conceivable that many of those engineers might have known that something fishy was going on, there were no whistleblowers.

Why? Adam believes one reason is that those engineers have no legal standing, no union, and no organization to support them — the best they could expect was to lose their job. He argues that has to change if we want to establish self-accountability in the embedded industry.


Without software, the conveniences of our modern life would crumble. Without accountability, professional standards, and ethics, however, we might find ourselves on the horizon of a world where

“Theoretically Dieselgate shouldn’t have happened.”

TILL ADAM

software’s capabilities aren’t harnessed safely. Just as it’s hard to imagine buildings without fire safety codes and doctors who don’t take the Hippocratic oath, perhaps these times are ripe for the industry to mature into one that includes increased controls on the development process.

The will is there — and the introduction of regulation for minimum software quality is achievable if enough people in the industry work towards it. Creating software engineering trade groups, instituting professional ethics into training, and putting in place mandated certifications for developers will help raise the bar for software development. Such changes could even prevent a tragedy from occurring.

It’s difficult to remember how we functioned without embedded software — many people are young enough to have no memory of such a world. We may well be advancing towards an age where no one remembers a software industry that didn’t include high standards of professional ethics in order to safeguard the products so ubiquitous in everyday life. 



Till Adam



Lars Knoll



Kalle Dalheimer



Jared Spool

THE TRUE COST OF SOFTWARE

Why we'll be paying for software
decades after we've written it



Because software is embedded into so many devices, tools, and objects that are essential to our everyday lives — affecting our well-being, our finances, even our relationships — one would think we would spare no expense to build and maintain that software to the utmost degree of excellence. That, unfortunately, isn't always the case. In fact, the opposite may be true.

Taking a glimpse into the future of embedded software means taking a good hard look at the cost of software: the value of it, the labor involved, the quality of initial and ongoing investments, and the conditions required to create and maintain something of merit. We're not just talking about monetary expenditures here.

Take smart intelligent devices, for example. They will be impacting more than just the bottom line. **Adrien Leravat**, embedded software and IoT architect at embedded IoT software specialist Witekio believes many people in different industries are all focused on eliminating costs by applying more AI systems. "While those technologies will eventually drive down costs for the end user, there is a hidden cost to intelligent hospitals and self-driving cars," says Leravat. "AI will make a lot of existing jobs obsolete."

One insight that's not widely realized comes from embedded software consultancy KDAB. **Till Adam**, KDAB's chief commercial officer, feels that society doesn't understand the true cost of building software — that people think software just happens. "Unfortunately, programmers have enforced that mystique by deluding themselves that with enough all-nighters and caffeine they can always pull a rabbit out of the hat," says Adam. "Yet most software is shipped with not only user-visible bugs but created with a poorly designed internal architecture."

Adam argues that as a society, we've contributed to this software quality problem since no one wants to pay what it really costs to build solidly constructed software. "It's far too easy to slap something together and ship it, but behind most apps is a team of developers praying they don't fall apart next month," he adds. "Companies with employees that had absolutely no software expertise two to three years ago are shipping products today."



Technology will eventually drive down costs for the end user but there are hidden costs to intelligent institutions — like the obsolescence of existing jobs.

“Consumers value cost more than workmanship and that inevitable pressure is hurting software quality.”

TILL ADAM


Kalle Dalheimer, KDAB president and CEO, concurs that most software isn't built properly. “We see lots of software that's falling apart,” says Dalheimer. “It can give you sleepless nights when you realize that decades-old rotting code is running the banks, power plants, and other critical parts of our infrastructure, but unfortunately very few people want to spend the money needed to clean it up so it's a problem that's still yet to really hit us.” He adds: “In another five years or so I think that much of this decaying software will collapse under its own weight.”

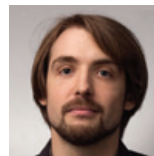
Adam further explains that part of the quality problem is due to developer culture. “Programmers that create shiny things are idolized by their peers,” says Adam. “Unfortunately, creating quality software isn't heroic — it's pretty boring.” If you've got unlimited resources, like Apple, or Google, or a 500-developer game company, you have a massive team that can divvy up the uninteresting work. Contrast that to your average embedded device created by two developers and their dog — they just don't have the time, energy, interest in, or capability to create lasting things of value. “In software, there's always someone who will do it cheaper and worse, than you,” Adam adds. “Consumers value cost more than workmanship and that inevitable pressure is hurting software quality.”

While the average consumer may not be willing to pay more for quality workmanship and while they may be oblivious to the reality of poorly designed and decaying software, there is no denying the impact

that embedded software has on almost every aspect of our lives. Would more awareness of the true cost of software be enough to drive up investment into excellence? Even industry experts who know that the current situation isn't sustainable find themselves continuing to work within a system that demands low-cost, high turnaround production.

What will it take to change the situation so that we all see the value of software?

Perhaps awareness is the first step — and that's no easy task. Software is hidden, invisible, intangible to the average Joe. But the people responsible for creating it — the developers, the engineers, and the coders — all know its value, and know it intimately; perhaps they are the best ones to lead the way to a world where quality and workmanship trump cost. 



Adrien Leravat



Till Adam



Kalle Dalheimer

CYBERSECURITY ARMAGEDDON



There's no silver bullet
for a problem that's been
decades in the making

Looking at the dazzling array of products we use that now have software built into them, it might be tempting to paint a picture of all this existing under the protective dome of a bright, sunny day. But there is a storm approaching. And ironically, it's a storm that threatens to get stronger the more sophisticated our embedded systems become.

Cybersecurity is a concern for almost everyone today — from a teenager using Snapchat to entire governments arming their computer systems against hackers. In the embedded software industry, however, this coming storm represents a do-or-die situation. Call it an opportunity to rise to a great challenge. A call to battle. Call it cybersecurity Armageddon.

Where is the industry with regard to taking on this challenge? According to some industry experts, we are far from where we should be. **Adrien Leravat**, embedded software and IoT architect at embedded software specialist Witekio, believes that everything will continue to be interconnected, creating more potential for bridges and loopholes in the system. “The more intelligent the device, the more precarious it is,” says Leravat. “Smart devices have many more software components, which exponentially increases the attack surface.”

While some tools are bridging the gap to help find vulnerabilities, we need people who have a complete view of the system and what’s needed to make it secure. That’s not going to happen overnight though; the cybersecurity predicament we’re in has been

developing over the last 35 years. Adds Leravat: “The regular rhythm of new breaches being continually announced highlights how important security is.”

Till Adam, chief commercial officer at embedded software consultancy KDAB, agrees that the situation is serious. “Previously you could break into one system at a time,” he says. “But now it’s many at once, compounded by ‘normal’ developer practices.”

Adam has a point. How many hundreds of thousands of devices use a standard silicon evaluation firmware image — one that defaults to root access with no password and open demo ports? Those may not provide an interesting hacking challenge, but with no attempts made to protect themselves surely a great many of those systems are now running botnets or bitcoin miners.

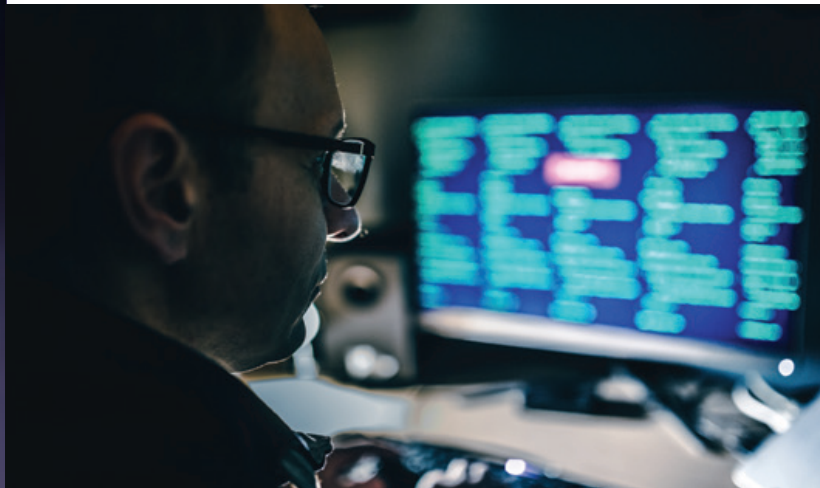
Cedric Vincent, director of technology at Witekio, concurs that in-field devices are a major weakness. “Once you have a lot of devices in the field, it’s already too late,” explains Vincent. “In most cases when those devices were built, nobody thought about security or how to update them.”

Cybersecurity regulations will eventually be the norm but it will take more than a white hat stunt to make it happen.



“The industry won’t self regulate cybersecurity without a disaster.”

LARS KNOLL



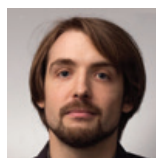
He believes the only way to protect those systems is to encircle them from the outside with an additional layer of security. Some solutions are trying to do that, using machine learning to analyze suspicious patterns and then remotely turn off a device, which is about the only remedy available. That means those legacy devices are doomed — a hacker can wreak havoc on your unsecure devices with the right probes in order to get parts of your factory or hospital turned off.

Many leaders in the embedded software industry believe that cybersecurity issues will eventually be regulated. Just as certification is needed for safety-critical systems, we need to do the same for security. **Lars Knoll**, chief technology officer of global software company The Qt Company, says this is going to come from regulatory or consumer pressure and unfortunately only after something really bad happens. “The industry won’t self regulate cybersecurity without a disaster,” says Knoll. “If it did, this would be the first time an industry preemptively changes course based on risk alone.”

As **David Faure**, senior software engineer for KDAB explains, cybersecurity problems often come back to insufficient education, as developers don’t learn about security in university or at their job since it’s not part of mandatory training. “New developers write code with a focus on features and stability, not security,” contends Faure. “Problems aren’t exposed in peer reviews since co-workers don’t know what to look for either.”

Thorough cybersecurity won’t come from on-the-job training; developers need experience to do it right. Either developers need to have this as part of their education, engineering departments have to institute training programs for their teams, or companies need to hire consultants to build and review the system — preferably all three.

All these strategies will be required for the coming cybersecurity Armageddon. But it’s not all doom and gloom — this battle is winnable. The cybersecurity provisions for embedded software outlined here present a clear path that, if followed, will help get us to a better future. It won’t be easy but instituting strong controls against cyber threats is achievable if enough people in the industry work towards them. This, in turn, will improve the state of software everywhere. 📌



Adrien Leravat



Till Adam



Cedric Vincent



Lars Knoll



David Faure



Data Privacy and Ownership

Can the public fight back against data monopolies?

With the power to solve so many modern problems, software is baked into everything from thermostats and cars to washing machines and calculators. In fact, it's getting more and more difficult to identify devices that don't have embedded software. One pitfall with this is the amount and kind of data that consumers of these products are sharing, sometimes unknowingly, and what's happening with that data.



Data is collected and used without our knowledge by the services on which we depend.

Now that services such as Facebook, Twitter, and Google are part of the daily lives of billions of people, these services are often accused of either using our data against us or failing to protect private information from falling into the wrong hands. The use — and misuse — of customer data is now a commonplace story in the nightly news, and when software issues have entered public awareness, you know it's becoming serious.

Unfortunately it's a problem that's more prevalent than we may be aware. **Kalle Dalheimer**, president and CEO at embedded software consultancy KDAB, believes that the public isn't concerned enough with their personal data and where it's being made available. And because people aren't aware of the data they're generating, they can't take measures to protect it.

"Programmers need to be more aware of data privacy — it's not just a problem for technology pessimists," says Dalheimer. "If programmers became accountable for data leaks, they would probably take them more seriously and their organizations would prioritize them before they're out of hand."

Lars Knoll, chief technology officer at global software company The Qt Company agrees. "Consumers have given up caring about their data privacy and blindly

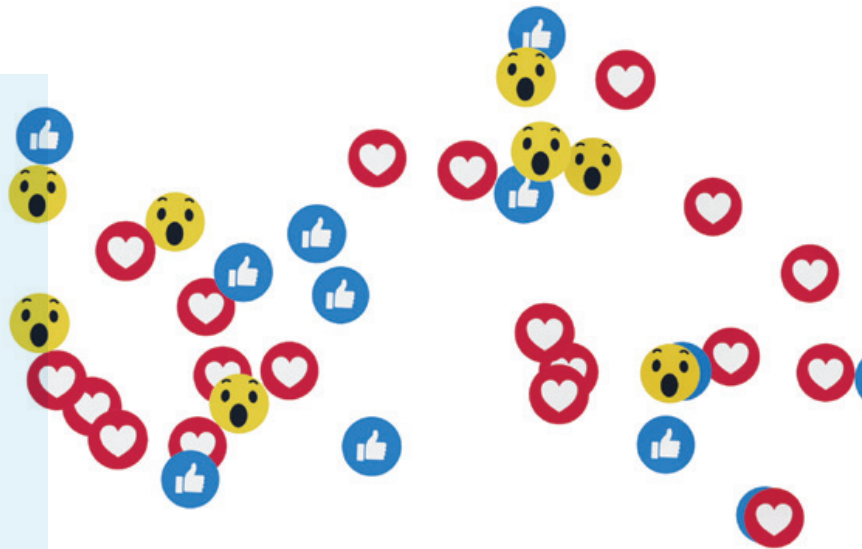
click every EULA," says Knoll. "Consumer-driven market pressure won't change the industry without some form of regulation that creates consequences for unsafe data practices."

Because access to massive amounts of data is so important to machine learning, data acquisition can quickly turn a business into a monopoly. A company with a data-collecting edge can produce better services and attract more people, creating a self-reinforcing cycle. "Big data monopolies are a big problem — once they reach critical mass, they can acquire other companies or force them out of business," explains Knoll. "A couple of huge players owning everyone's data isn't a healthy long-term trend, but they're going to fight hard to keep it."

Jared Spool, founder of UIE and Center Centre, agrees with this sentiment. He believes big vendors live off their data and that they're going to push back on any effort to regulate it. "When corporations have their business model threatened, they employ lobbyists and lawyers," says Spool. "We currently don't have interest in federal regulation in the US, so the states will have to own this problem."

“A couple of huge players owning everyone’s data isn’t a healthy long-term trend, but they’re going to fight hard to keep it.”

LARS KNOLL



What’s even more frightening is it’s not just in politics where this is happening. Spool contends that lives are now at stake — even though people mistakenly think that data privacy is an issue only for credit card companies and cheating spouses. “What if someone maliciously logs into your smart home interface and locks you out in winter?” he asks. “What if a hate group doesn’t like your tweets and finds your private address so they can mail you a bomb?”

Data privacy, it appears, can literally be a life and death matter. And while consumers may have a fuzzy awareness of what’s at stake, programmers, by contrast, are seeing the picture in sharp focus. They understand how critical it is to defend data privacy right now. And they are the ones who are uniquely positioned with the knowledge of what it takes to avert — and prevent — tragedies from striking. This challenge could allow programmers to evolve into a more responsible role — from one of mere creator to guardian. Programming professionals, armed with accountability, integrity, and skills, may one day prove to be the true protectors of data privacy. ▀

Data collection has now become an invisible “feature” of some of our services. Dalheimer knows of instances where people have had their TVs turn on when they’re talking with someone about it. “Your devices are always listening so they can offer digital assistance,” says Dalheimer. “And that’s an awful lot of data that’s being processed — with no controls over who’s seeing or listening to it.” He has a good point: Can Google, Apple, and Amazon prove that they’re not misusing your data? And is it too late to pull back the reins on data privacy?

Spool thinks data privacy issues have already left the barn: “We’ve seen privacy gone wrong with Facebook and Cambridge Analytica. The former made data available with no controls or oversight, resulting in unforeseen influences on world politics and an election that was apparently influenced by foreign actors.” He goes on to say that partisan bias aside, the vast majority of people would say using people’s data against them to overthrow a government is unacceptable.



Kalle Dalheimer



Lars Knoll



Jared Spool



Desperately Seeking Talent

Why the industry needs more than just highly educated engineers

The rapid proliferation of embedded software is solving a lot of problems for society but it's also opening a Pandora's box filled with a whole new set of issues — one of them being the need for a substantial talent pool. The skills needed to put together today's technology are intense and finding the people with those skills can be a big bottleneck for getting products to market. In the world of embedded software, we're talking about a full-on talent shortage. Talk about pressure.

Software is being developed at breakneck speed and getting new products, or the latest versions of those products, into customers' hands is creating a demand that is causing the people who desperately want to onboard new talent to start pulling their hair out. Education, skills, and experience are three pieces of the puzzle. But providers of embedded systems are also looking for that extra something — can your unique perspective make a positive contribution to the team?

According to **Lars Knoll**, chief technology officer at global software company The Qt Company, good software engineers aren't just hard to find, they're also expensive. "When you're trying to hire, it's very difficult to assess people's skill level, understand if they'll complement your team, or predict if they'll be an active contributor," says Knoll. "People who understand the entire software stack are even more rare than regular engineers, which makes them really in demand."

Kalle Dalheimer, president and CEO of embedded software consultancy KDAB, is having the same problem. He finds there's a huge talent shortage —

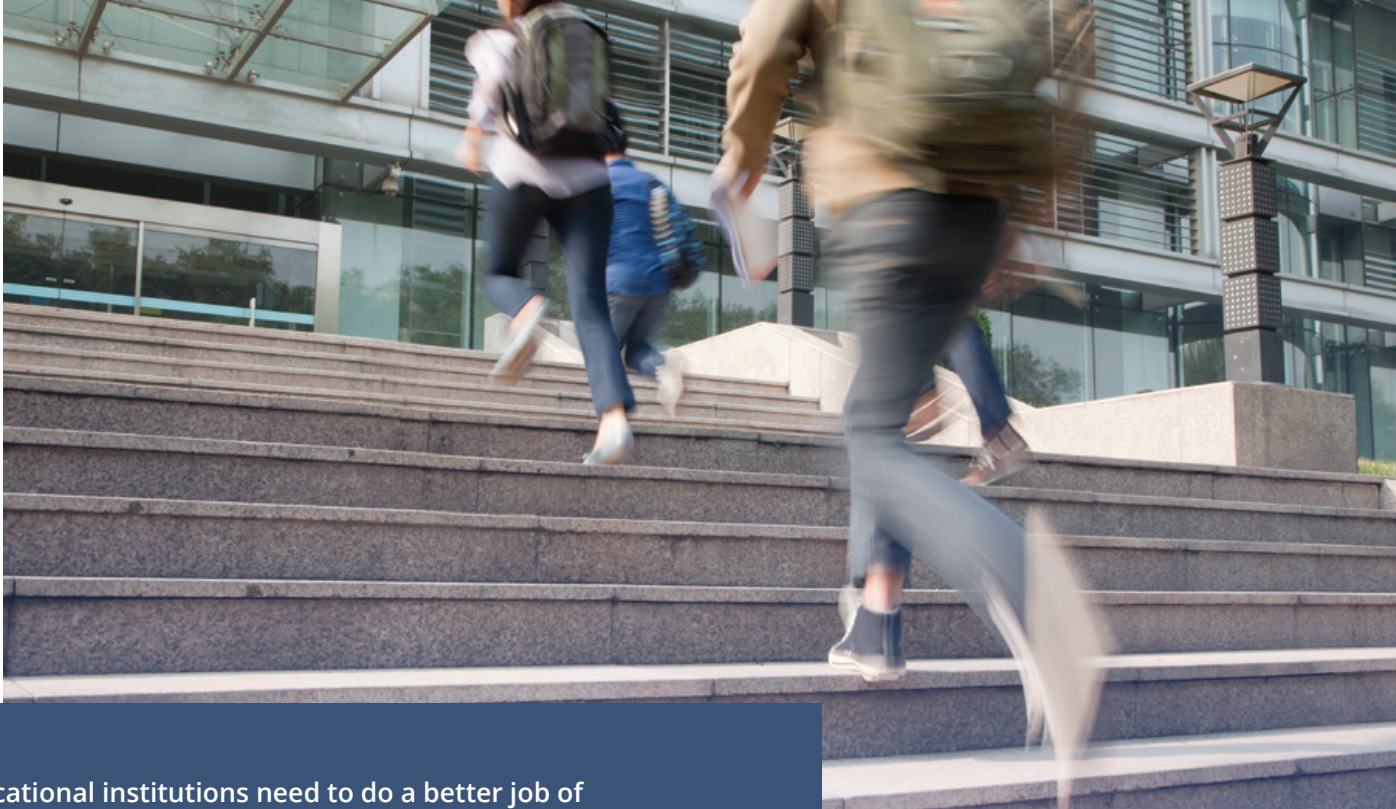
even worse today than in the mid-nineties — and that trained engineers, especially those with a good theoretical grounding, are very difficult to find. "We require people who understand algorithm complexity and who understand the difference between unsolvable and solvable problems," says Dalheimer. "Those skills don't come intuitively — they need to be taught. But educated people also need experience."

Education makes a difference and solves some problems — but not all. Knoll believes that a large problem in acquiring talent from universities is that people are being taught how to create software but not how to design and develop products. "People fresh out of school often have a throwaway approach to their code because that's how things have worked throughout their education," he says. "Ship and start over from scratch isn't sustainable in the working world; software needs to be maintained, evolved, and reused, and that's a very different set of skills."

Dalheimer believes part of our industry's talent problem is due to a lack of diversity in the educational system. He feels that neither our education system nor society at large have been particularly successful

"Software thrives on creativity and diverse perspectives — we desperately need more women and more people of color to enter the field."

KALLE DALHEIMER



Educational institutions need to do a better job of attracting a wider diversity of people to computer science in order to reenergize the talent pool.

at attracting a wide diversity of computer science candidates — and that’s a problem. “Software thrives on creativity and diverse perspectives,” he says. “We desperately need more women and more people of color to enter the field.”

Till Adam, chief commercial officer at KDAB, agrees that’s it’s been a selective club for far too long. His belief is that diversity will come. “As the middle class continues to grow in China, India, and Africa, people are finally able to access technology by using relatively inexpensive boards like Raspberry Pi or Arduino,” says Adam. “They’re seeing that software is not just for gear heads; it can be an enabler for other disciplines, letting people solve problems in their communities or create works of art.”

This, he says, makes software a desirable career path that can bring monetary, creative, and humanitarian rewards. He adds: “I think this will create an influx of new enthusiastic people, reenergizing the global computer software talent pool.”

While hiring gaps in the industry do not create a comfortable situation, they are compelling those in power to rethink their approach and this could be a springboard for transformation. New opportunities

may open up the industry to different perspectives and approaches that could revitalize the embedded software industry as a whole. These times also give a chance for those in the industry to think hard about exactly what kind of employees they want, and give the educational system a chance to update and upgrade its offerings. There is much work to be done and much further to go but with a clear vision and strong intent, this phase in the industry’s evolution could eventually lead to raising the bar of software development. This, we can all agree, is a worthy goal. ■




Lars Knoll



Kalle Dalheimer



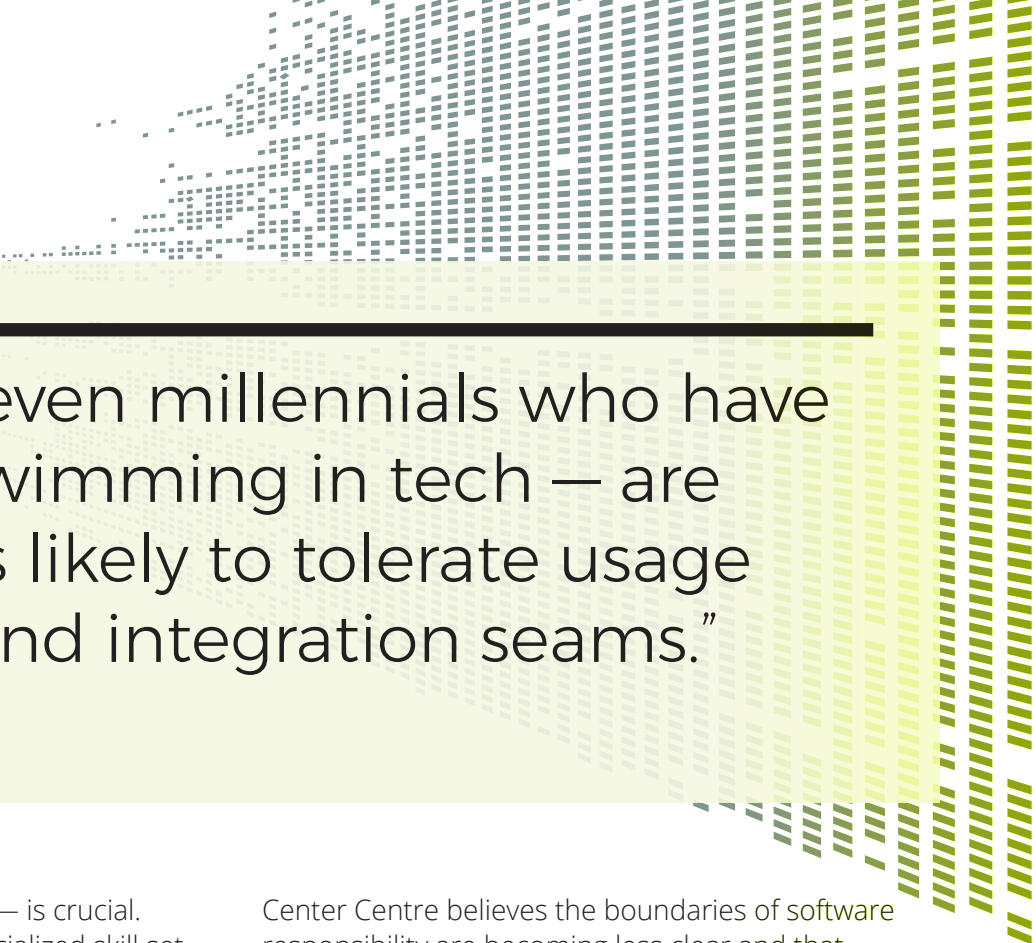
Till Adam



THE SEAMS OF INTEGRATION

Now that software building blocks are here, is our new problem the mortar?

Software is being built faster than ever before and much of that development speed comes from the ability to assemble applications and services from many preexisting blocks of code. However, this ease of development comes with some challenges.



“All users — even millennials who have grown up swimming in tech — are less and less likely to tolerate usage ambiguity and integration seams.”

JARED SPOOL

Integration — seamless integration — is crucial. Achieving it requires a wide and specialized skill set. Interoperability and compatibility are also critical — in fact they may never have been so important as right now, with different, competing companies all creating software at lightning speed that may eventually need to work together.

Lars Knoll, chief technology officer at global software company The Qt Company, believes that integration is often referred to as glue but that this is a terrible metaphor: “The integration process is far more complex than just tacking things together.”

Kalle Dalheimer, president and CEO of embedded software consultancy KDAB, agrees. “Building an embedded system requires a high level of software engineering competence in multiple areas,” he says. “A system integrator needs to be very familiar with all of the components and their APIs, understand module behavior under varied circumstances, configure an OS kernel and device drivers, integrate the user interface layer and multiple middleware stacks, and build smooth OTA and IoT frameworks.”

The list of necessary skills is huge and the number of people with this knowledge is low. Adds Knoll: “Often a lack of good integration can cause a project to fail, if it’s even able to make it to market.”

The thing is, integration problems don’t just affect developers. The biggest impact may in fact be with the user. **Jared Spool**, founder of UIC and

Center Centre believes the boundaries of software responsibility are becoming less clear and that ambiguity shows up in the consumers’ mental model of a product. “Take Apple CarPlay as an example,” says Spool. “From the perspective of the user it’s a single thing, but it’s really a pastiche of software and hardware components from Apple, the automaker, and automotive suppliers.”

Spool believes that this Frankenstein nature becomes obvious when you start seriously using it: Handoff problems between the phone and the car, unanticipated interactions, Bluetooth and Wi-Fi issues — these all create confusion for the user and an overall poor experience.

Moreover, as the user solves one set of problems, a new underlying set of issues reveals itself. This can be incredibly frustrating since only a few technically savvy people are capable of debugging their own devices.

Cedric Vincent, director of technology at embedded IoT software specialist Witekio, also feels that integration problems are becoming problematic, especially between disparate systems. “Standards make interoperability possible but technology is moving faster than the standards,” says Vincent. “Take IoT as a prime example — many companies are working on it but since they all want differentiated capabilities we’re going to end up with lots of cool technology that won’t talk to each other.”



Part of the problem is that standards are usually regulated regionally but technology is increasingly deployed worldwide. “There are global consortiums trying to address interoperability and some companies like Google may set a de facto standard,” explains Vincent. “However, different countries — especially in regions like the US, EU, and Asia — all have different ideas of what’s important, how technology should evolve, and how products should adhere to local laws.”

Vincent doesn’t see much hope that we’ll avoid incompatibility and expects a lot of gateways will be needed to translate between systems. Of course, that’s a problem since things always get lost in translation.

When he looks through the lens of the user, Spool agrees that IoT is heading into troublesome territory. He predicts that user acceptance of IoT systems will become harder to achieve, even among the technically literate. “We know that the burden of technology is too much for many consumers but all users — even millennials who have grown up swimming in tech — are less and less likely to tolerate usage ambiguity and integration seams,” says Spool. “TiVO decimated the VCR because customers refused to use products they can’t understand.”

Dalheimer concurs: “Since very few homeowners are willing to become information architects in order to outfit their smart homes, companies are going to have to start working better together instead of waiting for governance and regulations to help them out.”

Dalheimer’s vision makes sense: In order for software A and software B to work better together, the software developers from company A and company B need to start working together. It doesn’t have to be a competition. It could — and should — be a win-win-win situation for company A, company B, and the end user. Seamless integration is definitely a large challenge. When it doesn’t work, it presents frustrating problems for the end user that aren’t easy to solve. The solution lies in developers thinking outside of the box, crossing boundaries, collaborating — doing almost exactly what the software they’re creating will do. Developers need to think of themselves not as working for an organization but as working for the end user. They need to start thinking of their creations as not existing in one solid form but as living creations whose nature is to contact, collaborate, communicate, and expand. ■



Lars Knoll



Kalle Dalheimer



Jared Spool



Cedric Vincent



AUTHORS

Andy Gryn, Nancy Young | CX3 Marketing

SPONSOR

The KDAB Group

CONTRIBUTORS

Till Adam, Chief Commercial Officer | KDAB

Matthias Kalle Dalheimer, President and CEO | KDAB

David Faure, Senior Software Engineer | KDAB

Lars Knoll, Chief Technology Officer | The Qt Company

Adrien Leravat, Embedded Software and IoT architect | Witekio

Jared Spool, Founder | UIE and Center Centre

Cedric Vincent, Director of Technology | Witekio

ACKNOWLEDGEMENT

We would like to thank the industry executives who donated valuable time to share their views on the future of the embedded software industry and appreciate their passion for this field.



ABOUT THE KDAB GROUP

The KDAB Group is the world's leading software consultancy for architecture, development and design of Qt, C++ and OpenGL applications across desktop, embedded and mobile platforms and is the biggest independent contributors to Qt. Our experts build run-times, mix native and web technologies, and solve hardware stack performance issues and porting problems for hundreds of customers, many among the Fortune 500. KDAB's tools and extensive experience in creating, debugging, profiling and porting complex applications help developers worldwide to deliver successful projects.

KDAB's trainers, all full-time developers, provide market leading, hands-on, training for Qt, OpenGL and modern C++ in multiple languages. Founded in 1999, KDAB has offices throughout North America, Europe, and Asia.

www.kdab.com

This document is current as of the initial date of publication.

The information in this document is provided "as is" without any warranty, express or implied, including without any warranties of merchantability, fitness for a particular purpose, and any warranty or condition of non-infringement.

This report is intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. KDAB shall not be responsible for any loss whatsoever sustained by any organization or person who relies on this publication.