TAURUSEER

CODE ROT:
Healthcare's Growing Patient Threat.

A TauruSeer **Whitepaper** 2019

Everyday, my family puts our faith into the accuracy of healthcare software and the precision of medical devices. If they don't work as promised, my daughter's life is immediately at risk.

Jalen has been surviving Type 1 Diabetes since she was sixteen months old. Each day we rely on flawless communication between three pieces of technology: a continuous glucose monitor to report blood glucose levels, mobile apps on our devices that convert that data into precision dosing instructions and an electronic insulin pump that administers the exact amount insulin Jalen needs at exactly the right time. With Type 1 Diabetes the margin of error is too microscopic to make mistakes. If any of her machines lose accuracy (God forbid in the middle of the night), we could all face a life threatening emergency.

Parenting itself makes it hard to sleep at night.

Parenting at the mercy of machines that sometimes fail makes it nearly impossible.

Over the last eight years, our family has been on the receiving end of some unintended, but life-threatening consequences; all of which stem from a lack of governance, ineffective auditing and poor operational oversight of both software and hardware. This is why I've spent the last ten years debating with software engineers about moral coding philosophy, ethics in software, professional responsibility and well - the inconvenient truth about code rot.

Unfortunately, IT's targeted fixation on cybersecurity has most teams preoccupied with flagging symptoms and checking compliance boxes rather than solving systemic operational problems. It's a huge blindspot and one of healthcare's fastest growing threats.

This is why we founded TauruSeer. Our vision is to help build a culture of digital ethics, trust, and transparency by following six core values:

Stay Patient-Friendly: Manufacturers must stay accountable and respond faster by empowering patients to understand technical concepts, give them a place to articulate product experiences that bring quality issues to the surface and put in place collaboration policies with the FDA.

Surface Technical Insight for Business Leaders: Give leadership a clear and accurate understanding of the risk implications tied to tighter development timelines in the name of innovation.

Demand Transparency: Give security, IT, and developers a voice to articulate the value of investing in strategic initiatives, for example, Technical Debt and Code Rot.

Encourage Partnership: Allow policymakers to collaborate and develop guidelines that empower companies to securely operate at the pace of modern product development methodologies.

Slow Down to Speed Up: Evolve "Fail Fast" cultures that "Win Faster" by giving time back to engineering resources to implement best practices, automation, and continuous process and technology improvements.

Champion Visibility & Awareness: Truly care about and publicly address the real operational problems that put patients at risk.

Thank you,

CEO & Co-Founder, TauruSeer

Life with Type 1 is like living in a life threatening, never-ending math problem.

Diagnosed at 16 months old, Jalen Vaughan has spent the last 9 years navigating and surviving Type 1 Diabetes. A relentless disease, Type 1 Diabetes requires 24 hour vigilence to ensure blood glucose is controlled within tight safety parameters. A blood glucose spike or dip outside the tight parameters results in immediate and often life threatening symptoms that slowly compound and damage her long-term health.

The day of Jalen's diagnosis was the first time they almost lost her.

Initially, a mis-diagnosis of flu sent her home in a Benadryl induced sugar high with a Pedialyte chaser. Within 12 hours, Jalen was in the ER fighting a 900+ blood glucose level with her body in acute ketoacidosis. This trip to the ER signaled the beginning of the Vaughan's 'new normal': life as a constant math problem, where every meal, every activity and every reaction changes the formula to get the right answer. 'New normal' became the Vaughan's daily goal to keep Jalen's blood sugar between 80 and 120, 24 hours a day - even at night.

For the first five years, the Vaughan's had no tech. There were countless finger sticks, manual calculations, gut instincts and lots of guesswork. Overall, they managed her diabetes well, avoiding the ER and never using the dreaded glucagon rescue pen - but it was exhausting. Some welcome relief came in 2015, when Jalen got a much needed 'calculator' and the math got easier, fast.

For the last four years, Jalen has relied on two sophisticated products to keep her alive: an automated glucose monitor with corresponding mobile applications and an insulin pump. The future is a closed-loop system to continuously monitor her blood glucose, calculate needed insulin, and automatically administer it with better-than-human precision.

It could be a modern miracle.
Until it doesn't work.

Tech Failures Almost Cost Jalen Her Life.

In the summer of 2018 the Vaughans noticed the declining performance of Jalen's continuous glucose monitor (CGM). The CGM uses a mobile app provided by the medical device manufacturer. Features were disappearing, critical alerts weren't working, and notifications just stopped.

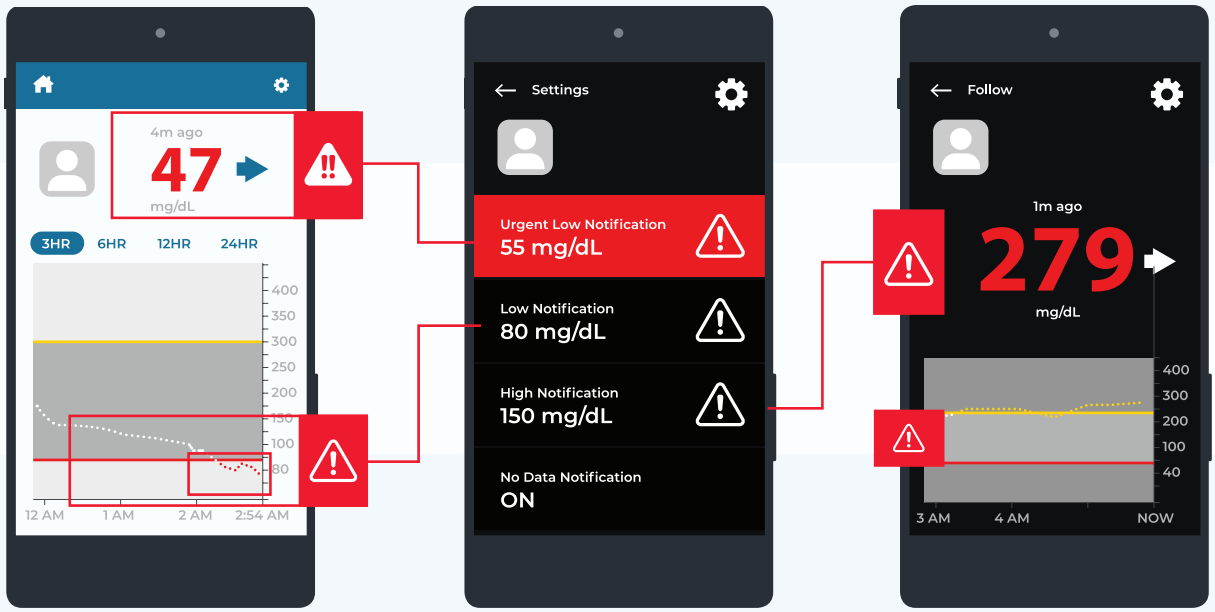
One night*, Jalen had to save her own life. On that particular night she awoke to her body shaking in response to a blood sugar of 47mg/dl. No alarm went off. Her parents weren't alerted. At nine years old, Jalen had to rely on instinct, immediately "juicing" herself with a lemonade Capri Sun.

Without blood glucose notifications of "lows" (Hypoglycemia) or "highs" (Hyperglycemia) the Vaughans were blindly trusting that the technology was simply working - but it wasn't.

By late October 2018 the CGM mobile apps were simply no longer dependable. Alerts didn't work, trend views were gone and there were way too many bugs to list. They found themselves flailing with outdated and unsupported software to track their daughter's glucose levels. This also led to challenges for Jalen's nurses and caregivers at school and her after hours programs.

The trust was gone.

By December 2018, the mobile apps were completely useless. The Vaughans felt alone, but suspected they weren't. They took to the reviews on Google Play and Apple App store and discovered hundreds of patients and caregivers complaining about similar issues.



*It happened overnight as we were forced to upgrade the Operating Systems on our mobile phones which completely made the app obsolete that evening. We didn't know until she had to save her own life.

Technically speaking, this is code rot.

As of 2019, the CGM mobile apps on the Apple iOS and Google Android stores hadn't been updated since October 2016, even though updates in both systems had occcurred within the same period of time. This was evidence of code rot.

For mobile technologies, this is eons!

What is code rot*? By definition, code rot refers to the slow degradation in the performance of computer software. Such software shows diminished responsiveness, lacks updates, and may become faulty over time owing to changes in the operating system it is running on, and thus may need upgrading.

* Software rot is also known as software erosion, code rot, software entropy, bit rot or software decay.

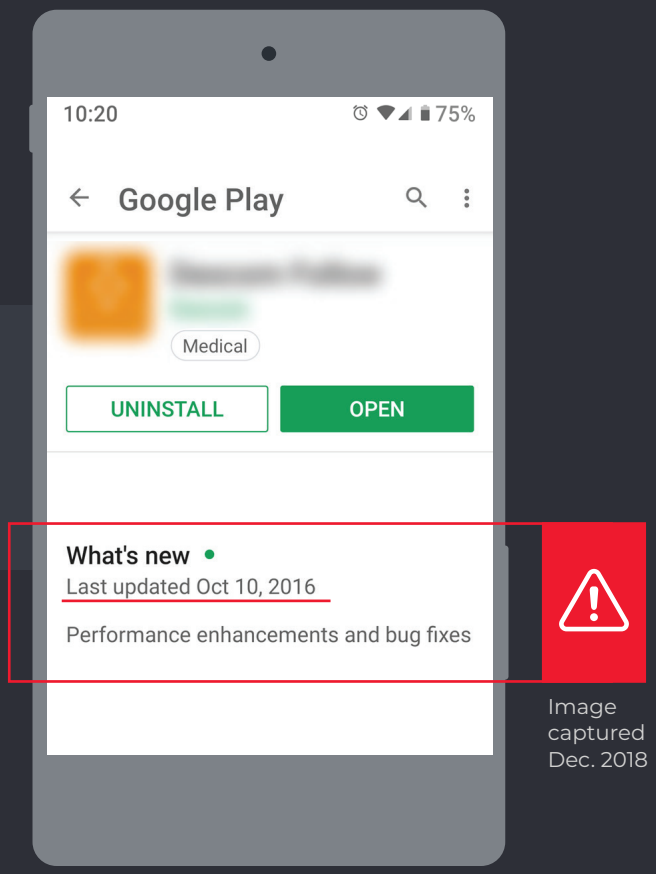
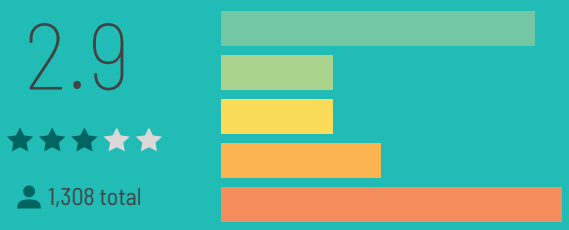


Image captured Dec. 2018

Code rot risks patient safety and destroys consumer trust:



"App won't launch. Got warning screen saying my version of Android (9.0 on a Pixel 2) wasn't supported. Pretty unacceptable for an app meant to notify me about my diabetic mother's health."

→ Connor de la Cruz

"After Android update, I stopped receiving alarms, completely rendering this app useless for alerting me to my child's highs and lows, especially at night. I currently have settings to notify at 90 however twice today my son was in the 80s and I never received a notification."

→ Amity Holland

Experienced risk management software engineer, Henrik Warne, published a post on "Thoughts on Programming" blog back in April 2017 which explains Code Rot beautifully. Similar to diabetes, Henrik explains it in two distinct ways:

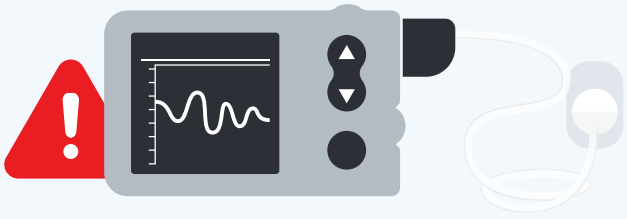
Type 1:
A Changed Environment.

Think of incompatibilities. The tech, systems, or other dependencies won't play nice together because of code and/or environment changes.

Type 2:
A Gradual Decay.

More common, this happens incrementally. Like driving a brand new car off the dealer's lot. At launch, it starts depreciating & deteriorating over time.

To compound the risk inherent in the industry's push into a closed-loop system, we have to consider the other half of the 'calculator': the insulin pump. If we can't guarantee that the glucose monitoring mobile app is working, are we really going to trust machine intelligence via code to make precise dosing recommendations to the device that essentially functions as a pancreas?



According to the FDA, hackers can exploit vulnerabilities to change the pump's settings to either over-deliver insulin, which leads to low blood sugar, or stop insulin delivery altogether, which leads to high blood sugar, or ketoacidosis. That is scary by itself, but even scarier is the fact that the FDA only seems concerned with external threat. Nowhere does the FDA address the internal systemic resiliency and the risks of code rot.

What if the insulin pump's code is buggy or the algorithms calculate incorrectly because of poor architecture or inefficient code? The outcomes could be life-threatening. The margin for error of a lethal amount of insulin is very low. To put this microscopic margin of error into perspective, take a look at the two spoons. The spoon on the top is TWO units (enough insulin for Jalen to eat 20 grapes). The spoon on the bottom contains TEN units. If only two units were needed, the extra eight units could kill her.

Why does code rot happen?

- ZERO measurement or tracking
- Limited resources and people
- Growing code and technology complexity
- Delivery demands and time pressures
- Inexperienced or indifferent developers
- Lack of FDA Regulations

Is regulation helping or hurting?

Due to the growing risk of Internet of Things (iOT)-powered medical devices, newly sponsored legislation such as HR 3985--Internet of Medical Things Resilience Partnership Act of 2017 is gaining momentum. In addition collaboration between the US Food & Drug Administration (FDA), American Hospital Association (AHA), and the US Department of Commerce's National Telecom & Information Administration (NTIA) is pushing forward a "Software Bill of Materials (SBOM)."

> "Modern software systems involve increasingly complex and dynamic supply chains. Lack of systemic transparency into the composition and functionality of these systems contributes substantially to cybersecurity risk as well as the costs of development, procurement, and maintenance. In our increasingly interconnected world, risk and cost impact not only individuals and organizations directly but also collective goods like public safety and national security. "
>
> → NTIA Working Group

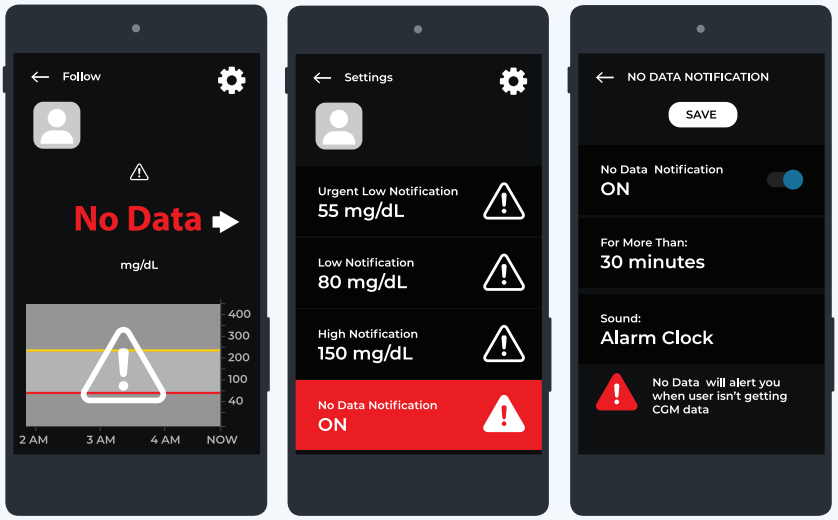
Progress in research & standards does not equal patient safety

Though there is progress in standardization around cybersecurity risk management, it's not translating into patient safety.

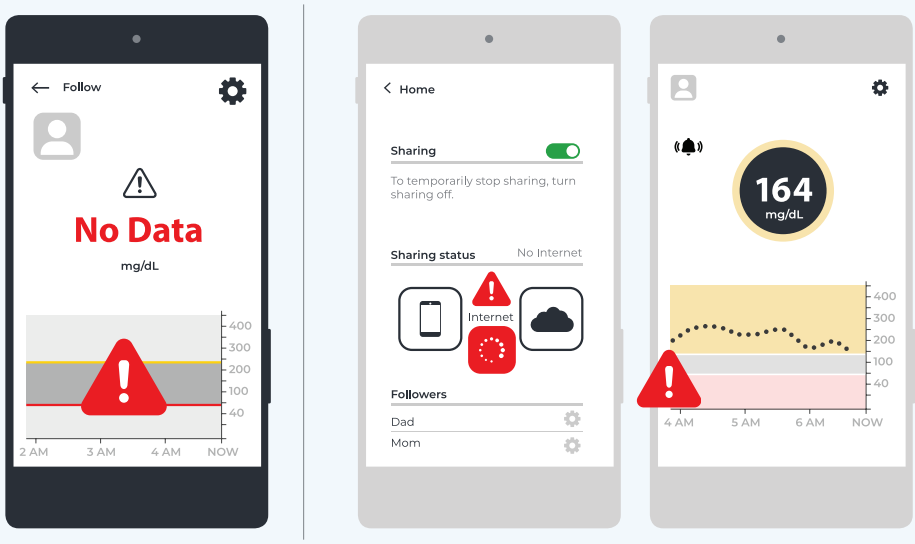
Software and hardware manufacturers must still account for and address mounting technical debt. Technical debt (also known as design debt or code debt) is a concept in software development that reflects the implied cost of additional rework. Technical debt can be compared to monetary debt. If technical debt is not repaid, it can accumulate 'interest', making it harder to implement changes later on. Unaddressed technical debt increases code rot.

> "Not paying down technical debt in a timely fashion can bankrupt your product."
>
> → Andrew Dallas, president & CTO at Full Spectrum Software

Dad's Google Pixel



Mom's Google Pixel



Daughter's Apple iPhone

Here is an example of Technical Debt "bankrupting a product" while simultaneously impacting patient safety. After two years, we blindly trusted that the app had been rebuilt. Unfortunately, the only improvements were quick fixes and patchwork. Technical debt wasn't addressed. We validated errors on all devices and still found reviews sharing similar stories.

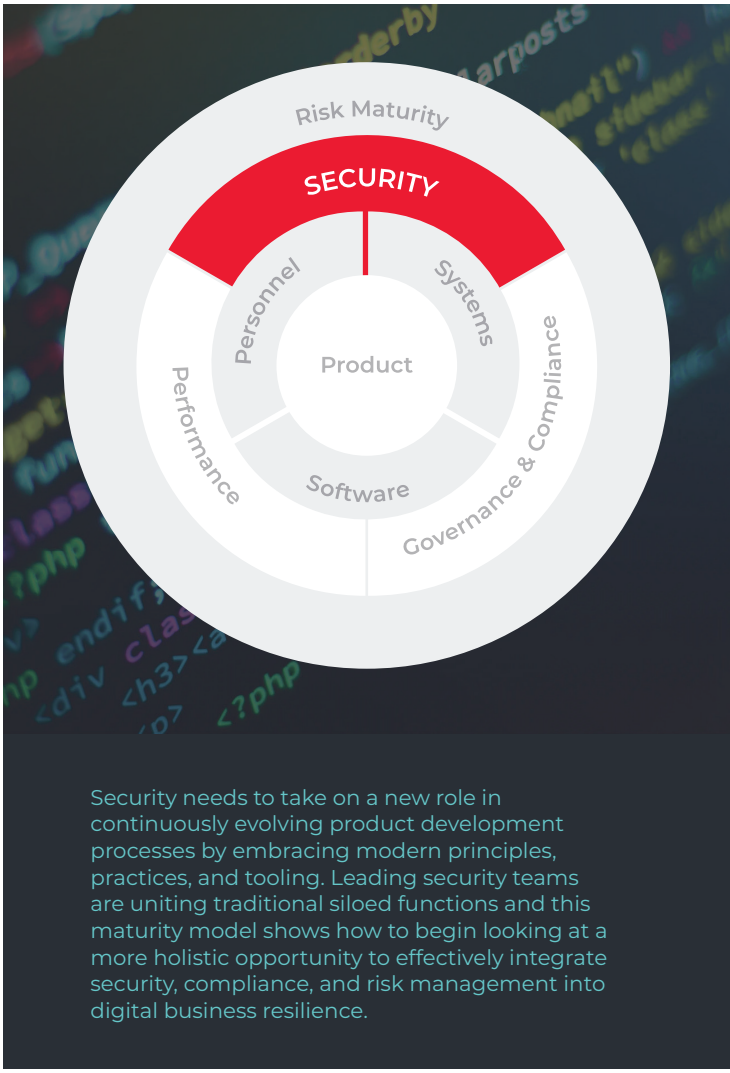
- Device Manufacturer "fixes" app - March 2019
- Follows FDA Post-Market Process/Guidelines
- Found this via NTIA: The US Food and Drug Administration (FDA) has published draft Pre-Market Guidance for medical device manufacturers seeking FDA certification. This guidance maintains that: "The device design should provide a CBOM in a machine readable, electronic format to be consumed automatically" where Cybersecurity Bill of Materials (CBOM) is defined as " a list that includes but is not limited to commercial, open source, and off-the-shelf software and hardware components that are or could become susceptible to vulnerabilities." https://www.fda.gov/medical-devices/digital-health/cybersecurity#guidance
- Again, "vulnerabilities" ....missing the point of tech debt/code rot in continuous risk monitoring.
- App still not working as expected AND technical debt seems to be creeping back into creating problems -- August 2019

Why treating cybersecurity issues won't solve software development issues.

After digging into regulatory requirements to understand medical software and device security standards, the Vaughans found plenty of regulations around cybersecurity, but nothing about continuous risk measurement and monitoring for patient safety. Why? Does the FDA process make this too hard or too expensive for organizations to manage? Is there too much emphasis on treating cybersecurity symptoms rather than solving software problems?

Treating cybersecurity symptoms does nothing to maintain healthy code and therefore leaves patients at risk. They are built to combat external threats, leaving internal issues festering unnoticed and unchecked:

Cybersecruity Treatment



Security needs to take on a new role in continuously evolving product development processes by embracing modern principles, practices, and tooling. Leading security teams are uniting traditional siloed functions and this maturity model shows how to begin looking at a more holistic opportunity to effectively integrate security, compliance, and risk management into digital business resilience.

01

Web Application Firewall.

Web Application Firewalls (WAF) was everyone's friend, so long as they were tested, configured correctly, and maintained. But, WAFs were purpose-built for monitoring the perimeter, not application based protection. With the move to the perimeter disappearing with cloud-native SecDevOps, WAFs struggle to stay relevant. Finally, given the historically contentious relationship between engineering and security teams, leaders need a better way to unite Secure DevOps teams to secure the environment.

02

Cybersecruity Treatment

Penetration Testing.

Penetration testing, also known as "ethical hacking" is most often deployed post-production making it largely reactive. If it's integrated into the software development cycle it can be preventive. However, Ilja van Sprundel, director of penetration testing at security company IOActive, admits he detects a significant amount of [code] rot in the foundation of a wide swath of commonly used software code.

Cybersecurity Treatment

Application Security Testing.

SAST and DAST solutions move us into more proactive approaches, but these security tools still fall short in addressing operational problems which create the symptoms in the first place. These tools are great as niche solutions and "point-in-time" scans, however, they don't fully address risk management as intended by the FDA, AHA, and NTIA.

Cybersecurity Treatment

Vulnerability Assessment.

The point-in-time vulnerability assessments and the newer orchestration and automation tools offer great promise for developer efficiency but could add to the "more noise" problems as they still lack the context to inform overall strategic risk of an organization. These approaches will remain incremental security tools unless they can integrate to solve the unified, integrated risk management platform problem.

Cybersecurity Treatment

Audits for Regulatory Compliance.

This is perhaps the biggest treatment illusion of them all. Unfortunately, solving for compliance "checkboxes" does not help the expanded team in modern product delivery. It creates a convenient "wall of compliance" between security, IT, and developers. The result is minimal collaboration that compounds risk and leads to ineffective management and security practices.

Cybersecruity Treatment

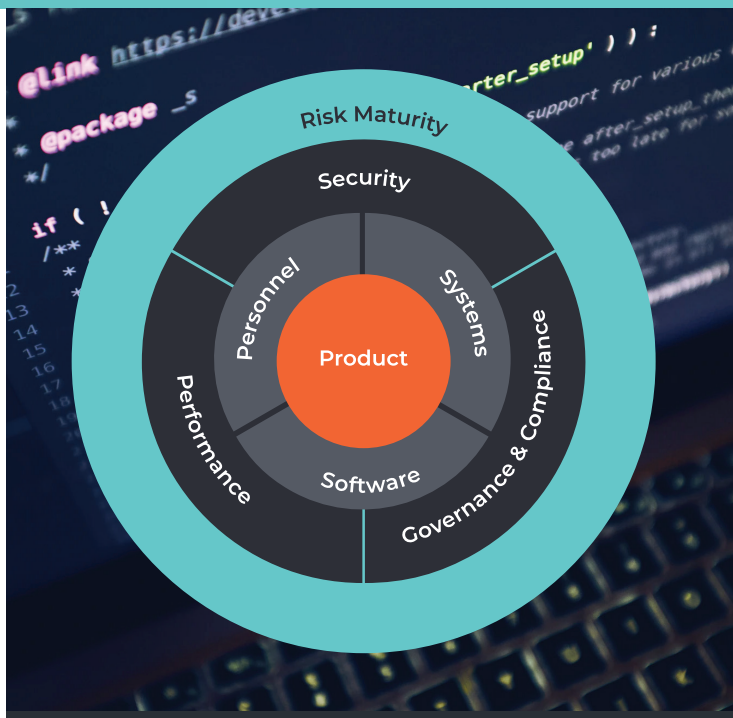
Software/Cybersecurity Bill of Materials (SBOM/CBOM).

To be sure, SBOM/CBOM creates a solid foundation for meeting the responsibilities of secure software and devices. However, just like vulnerability management, SBOM/CBOM must be a continuous process embedded throughout the entire lifecycle of the product. It cannot be a "snapshot," but rather needs to be a continuous monitoring with automated risk assessments.

> Good physicians treat the disease. Great physicians treat the patient who has the disease. The solution to solving patient safety risks needs more context.

Addressing systemic operational health effectively.

In order to 'treat the patient,' healthcare software and device manufacturers need to define what systemic operational health looks like for all stakeholders and then proactively work toward achieving it. We believe there are four pillars to a successful framework for operational health:



Cyber and IT Operational Resilience focuses on a comprehensive risk management approach through People, Processes, Applications and Systems.

01 Pillar **Collaboration**

Systemic operational health takes collaboration between security, IT, and engineering teams. It also takes treating internal risk with the same rigor and proactive discipline as the external threat.

Recommendations for getting started:

- Unite cross-functional teams in a product-centric approach.
- Champion visibility, awareness, and accountability.
- Promote digital ethics, trust, and transparency.
- Run IT and product development with mission discipline.
- Invest in automation and long-term approaches to adaptability.

02 Pillar **Balance**

Systemic operational health takes a commitment to balancing the pace of innovation with the ethical responsibility to monitor and maintain existing products that impact patients.

Recommendations for getting started:

- Align governance, risk and compliance (GRC) and security operations with modern product development.
- Make it easier for developers to understand security and regulatory compliance requirements in order to build and deploy trusted products from the start.
- Provide engineering and DevOps teams with in-the-moment security, risk, and compliance training to drive independent mitigations.
- Integrate software to streamline risk-based management practices to efficiently and proactively manage security, performance, and/or noncompliance problems.

03 Pillar **Education**

Systemic operational health takes educating policymakers and calling for real regulatory support from FDA.

Recommendations for getting started:

- Modernize approaches from legacy GRC and evolve from single point-in-time security assessments and compliance certifications to continuous risk monitoring.
- Build a shared understanding and common language that translates risks and aligns business leaders, technologists, and policymakers.
- Enhance premarket guidelines for and postmarket surveillance requirements of deployed software and systems.
- Implement better technology management practices and accelerate process improvements by requiring software providers that incorporate open source components, external APIs, or microservices to document and monitor their source and related data in an integrated Software Bill of Materials (SBOM).
- Require software providers, buyers, and operators to track operations inventory of applications and require continuous compliance against cybersecurity frameworks.
- Evolve certified auditor training requirements so that they can understand software, systems, and data for more targeted investigations; hold them responsible for sourcing/software vendor verification.

04 Pillar **Automation**

Finally, systemic operational health requires the right tools to keep up. We've discussed why continuously auditing and monitoring software, mobile apps, and device code is critical. Unfortunately, humans alone cannot keep pace. Systemic operational health requires smart automation tools that can surveil and detect issues in millions of lines of code -- instantly.

Recommendations for getting started:

- Automate everything possible, including workflows, business processes, decisions, approvals, documentation, vulnerability assessment/management, configuration monitoring and related policy/procedure documentation, and evidence required by the Risk Management Framework (RMF)
- Adopt TauruSeer to enable modern product delivery methodologies and deliver actionable insights for smarter, business-aligned decisions.
- Design an explicit, product-centric risk management framework aligned with HITRUST CSF, defining ownership and alignment among stakeholders.
- Further invest in TauruSeer's platform by integrating compensating controls, data management systems, end-to-end workflow and analytics to increase efficiency.
- Schedule alerts about software end-of-life (EOL) situations, make code easier to review, get patched code into production faster, provide a blacklist of banned components and a whitelist of preferred components, and provide a continuously updated SBOM to customers to verify claims.

As the promise of technology continues to enhance and even extend patient lives we will find ourselves even more dependent on the software and devices that keep us well.

When they work properly, we won't even know they're there. However, as that dependency grows, so must our vigilance around creating and maintaining operational oversight and governance for both software and hardware.

With a clear understanding of the operational health challenges we all face, and with a shared framework to help us move forward, we can (and should) build a culture of digital ethics, trust, and transparency.

CONTACT

Email us at:
hello@tauruseer.com

Visit us online at:
www.tauruseer.com

The Barnett
112 W. Adams St., 4th Floor
Jacksonville, FL 32202

Future Proof Your Healthcare Technology

TauruSeer's Product-Centric Risk Management platform unites risk leaders, SecDevOps teams, engineering, and executives to collaborate and proactively take the right actions to build a **culture of systemic cyber and IT resilience.**

REFERENCES

This whitepaper has been peer-reviewed by:

- A Medical Doctor and Scientific Researcher at the US's #1 HDO
- Executives at one of the largest Healthcare Payers (Insurance Company)
- A Product Manager and Software Engineer at the largest device manufacturer
- A Leading Diabetes Clinical Study Research Firm
- Healthcare Software and Security Experts (CTOs and CISOs)

Recommended Reading

- Kevin Fu, Associate Professor, University of Michigan. 2015. On the Technical Debt of Medical Device Security.
- Andén, P., Gnanasambandam, C., Strålin, T. McKinsey & Company. February 2015. The Perils of Ignoring Software Development.
- Marks, J. The Washington Post. 2019. The Cybersecurity 202: Hackers are going after medical devices — and manufacturers are helping them.
- Dallas, A. Medical Device and Diagnostic Industry. March 2017. What Technical Debt Means for Medical Device Developers Not paying down technical debt in a timely fashion can bankrupt your product.
- Fatemi, F. Forbes Magazine. May 2016. Technical Debt: The Silent Company Killer.
- Rosenblatt, S. The Parallax. 2019. 'Memsad' software rot threatens to leak your digital secrets.
- Donovan, F. Health IT Security. Medical Device Security Requires Collaborative Action from Industry.
- Donovan, F. Health IT Security. Reducing Cybersecurity Vulnerabilities Part of FDA Action Plan.
- Donovan, F. Health IT Security. How Healthcare Providers, Vendors Can Collaborate on IT Security.
- Snell, E. Health IT Security. How FDA Medical Device Cybersecurity Guidance Affects Providers.
- Dieterich, E. Health IT Security. 2016. Assessing Vendor Risk for Stronger Health Data Security.
- Westman, N. The Verge. April 2019. Healthcare's Huge Cybersecurity Problem: Cyberattacks aren't just going after your data.
- Kelly. S. MedTech Dive. July 2019. FDA sets final guidance on combo product safety reporting.
- Kouns, J., Corman, J. RSA Conference 2014. Software Liability?: The Worst Possible Idea (Except For All Others).
- Warne, H. Thoughts on Programming Blog. April 2017. Code Rot.
- United States Department of Commerce. National Telecommunications and Information Administration. Software Component Transparency.
- Health Industry Cybersecurity Practices: Managing Threats and Protecting Patients
- FDA pushes back aganst criticism of third party 510(k) review
- 510(k) Third Party Review Program
- Amazon News: 510(k) Third-Party Review Program
- The Challenges and Need for a Cost-Effective Risk Management Program
- Culture Change, Processes Crucial for Effective Risk Management
- When Compliance Isn't Enough: A Case for Integrated Risk Management
- Gartner Blogger John Wheeler
- CDRH Proposed Guidances for Fiscal Year 2020 (FY 2020)