The Cost of Poor Software Quality in the US: A 2020 Report

HERB KRASNER
MEMBER, ADVISORY BOARD
CONSORTIUM FOR INFORMATION & SOFTWARE QUALITY™ (CISQ™)
WWW.IT-CISQ.ORG
HKRASNER@UTEXAS.EDU
DATE: JANUARY 1, 2021

CISQ
Consortium for Information & Software Quality™

Contents

List of Figures and Tables

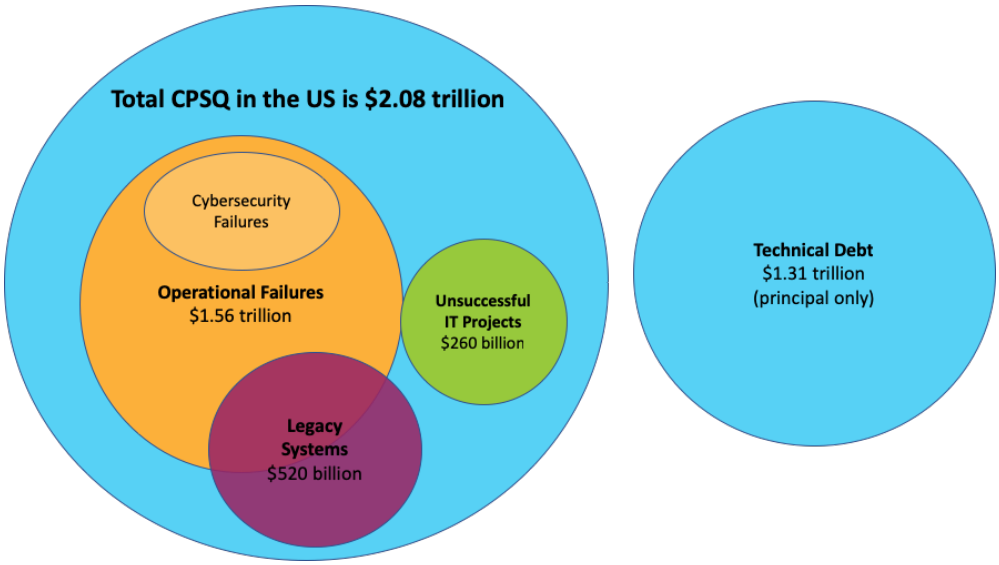
Copyright Notice

1. EXECUTIVE SUMMARY

This report was developed during especially turbulent times with the world battling a global pandemic. Yet, software continues to grow, proliferate, and enhance our digitally enabled lives. As organizations undertake major digital transformations, software-based innovation and development rapidly expands. The result is a balancing act trying to deliver value at high speed without sacrificing quality. Generally, however, we are not very good at balancing. Software quality lags behind other objectives in most organizations. That lack of primary attention to quality comes at a steep cost, which is revealed in this report. While organizations can monetize the business value of speed, they rarely measure the offsetting cost of poor quality.

For the year 2020, we determined **the total Cost of Poor Software Quality (CPSQ) in the US is $2.08 trillion (T)**. We also note that the 2020 US figure for the software technical debt residing in severe defects that need to be corrected would be $1.31 T (minus interest) but did not include technical debt in the total CPSQ since it represents a future cost which is increasing (14% rise since 2018). The graphical results are shown below.

Figure 1: CPSQ in 2020 in the US



Specifically, we determined that:

- The largest contributor to CPSQ is operational software failures. For 2020 we estimated that it is ~$1.56 T, a 22% growth over 2 years – but that could be underestimated given the meteoric rise in cybersecurity failures, and that many failures go unreported. The underlying cause is primarily unmitigated flaws in the software.
- The next largest contributor to CPSQ in unsuccessful development projects totaling $260 billion (B), which rose by 46% since 2018. The project failure rate has been steady at ~19% for over a decade. The underlying causes are varied, but one consistent theme has been the lack of attention to quality.
- Legacy system problems contributed $520 B to CPSQ (down from $635 B in 2018), mostly still due to non-value added "waste."

The detailed calculations leading to these totals are found in section 3 of this report.

Our general recommendations for 2020 continue to emphasize prevention. The next best approach is to address weaknesses and vulnerabilities in software by isolating, mitigating, and correcting them as closely as possible to where they were injected to limit the damage done. More specifically, we recommend that software shops:

- Avoid low quality development practices and adopt secure coding practices.
- Recognize the inherent difficulties of developing software and use effective tools to help deal with those difficulties.
- Ensure early and regular analysis of source code to detect violations, weaknesses, and vulnerabilities.
- Measure structural quality characteristics.
- Focus on the evaluation of included components (e.g., open source) and platforms which may have unknown weaknesses or vulnerabilities.
- Learn more about the typical vulnerabilities and exploitable weaknesses attributable to certain programming languages.
- Use best known practices for managing a legacy system – especially when it comes to overcoming the loss of understanding and knowledge of how the system works internally. Benchmarking health status is a good place to start.
- Avoid unsuccessful projects by not creating arbitrary schedules. Pay attention to defined quality objectives and measure against those objectives throughout the project lifecycle.
- Invest smartly in software quality improvements based on CPSQ numbers in hand.
- Focus on the different results of good vs. poor software quality in your shop and relevant benchmark organizations.

By attempting to improve CPSQ, other economic target areas will be impacted – e.g. cost of ownership, profitability, human performance levels, ability to innovate, and effectiveness of your mission critical IT systems.

In our conclusions we identify what specific actions you can take at the level of: 1) individual software professional, 2) team/project leader, and 3) management/executive level of an organization. We also reveal an important (but little known) study that explains the difference in practices between high performing vs. low performing software organizations. That study revealed a 5-10X difference in performance between the top 10% and the bottom 10% of organizations sampled. When you dig deeper into the data, the reason is clearly the adoption of certain quality and process best practices. The key enablers for achieving the highest levels of cost, schedule **AND** quality performance are:

- A well-defined, yet adaptable development process
- Excellent estimation methods
- Project management discipline
- Excellent staff skill levels
- Quality vision
- Customer satisfaction focus
- TQM management culture
- Defect prevention

These best practices and recommendations are then consolidated in our conceptual process model called DevQualOps – which represents the next evolutionary step beyond today's Agile plus DevOps and similar continuous evolution and delivery models.

In this report we quantified the negative economic value of poor quality in our software systems at a US national level. We did so, with the hope and expectation that the readers of this report will be inspired to do likewise within their own organizations.

2. THE ENVIRONMENT AFFECTING SOFTWARE AND ITS COSTS

2.1. The cost of Covid-19 in the US

This report is being developed during especially turbulent times with the world battling a global pandemic. In mid-October 2020, the International Monetary Fund (IMF) predicted a steep fall in current international growth as the global economy struggles to recover from the pandemic-induced recession, its worst collapse in nearly a century. After 2.8% growth in 2019, the IMF predicts a rebound to global growth in 2021, but the recovery will be long, uneven, and uncertain.

In the US, the IMF forecasts an economic contraction of 4.3% for 2020, compared to growth of 2.2% the previous year. The US budget deficit soared to a record $3.1 trillion in the 2020 fiscal year, underscoring the long-term economic challenges facing the US as it tries to emerge from its sharpest downturn since the Great Depression.

Based on current assumptions, the estimated cumulative financial costs to the US of the COVID-19 pandemic related to the lost output and health reduction is estimated at more than $16 trillion, or approximately 90% of the annual gross domestic product. For a family of four, the estimated loss is nearly $200,000.

Technology companies have led the way on a variety of strategies other industries are now using to cope in this crisis — from remote working to managing globally dispersed supply chains. This crisis might well spark further creativity and innovation. Remote work, online education, and social distancing create demand for products and services delivered by the IT industry.

2.2. Computing technology continues to move forward

The McKinsey & Co. October 2020 report found that nimble companies shifted to remote work more than 40 times faster than they expected possible. Interactions between customers and North American companies are now 65% digital, compared to 41% pre-pandemic. Similar trends are seen globally. Changes made to cope with the pandemic — like moving to cloud computing or online purchasing — are likely to stick in the long term. Computing technology has allowed people to work, shop, and study remotely, and many people will continue the habits they have acquired since March 2020 when the pandemic began. Rapid digitalization accelerates the global demand for software-based solutions.

On the other hand, privacy violations, data breaches, social media-enabled disinformation, market domination, and other controversies have thrust the technology industry, and many of its leading corporations, into the media spotlight and government watchdog lens. And these are only the problems that are visible. The vast majority of technology problems and failures are still well-hidden from public awareness, and therefore, so are the costs.

Software and digital content companies now make up 78% of the Deloitte Fast 500 list of winners, while hardware, device, and networking companies constitute just 10%. Over the last 25 years, *software ate the hardware.* In 2011, Marc Andreeson proclaimed that *"Software is eating the world"* - which is still the case, but more voraciously.

The US is the largest technology market in the world, representing 33% of the global total, or approximately $1.6 trillion for 2021. More detail on the economic trends for the IT market may be found in Appendix B.

2.3. Services are now trying to eat software

Products and services that were traditionally delivered through other means are now being run by software and delivered as online services with great financial success. In an age where billions of people have access to the internet, the economic shift to software and software-enabled services took shape, disrupting traditional industries. For example, consider the contrast between the fall of Borders and the rise of Amazon. New software platforms and tools are ironically making the development of yet more disruptive software easier and more accessible than any other time in history, spurring an accelerated cycle of development and growth.

Digital transformation is accelerating and being fueled by more software solutions. Software-as-a-Service (SaaS) is at the heart of digital transformation, transforming industry after industry, as companies substitute capital investments in software over labor and tangible assets. For more information, see Digital Transformation: The Definitive Guide (2021). Many digital transformation initiatives fail not because of a focus on technology and business, but rather poor software engineering and lack of attention to quality issues.

2.4. Key IT trends effecting software development in the next decade

Newer technologies to keep an eye on include virtual reality, autonomous vehicles, drones, 5G, augmented reality, 4-D printing, digital currencies, digital assistants, blockchain, quantum computing, microchip implants, volumetric displays/holograms, self-healing systems, digital twins, AI everywhere, smart dust, more mobile devices, and the spread of IoT. From a software quality cost perspective, the following five areas in particular bear further scrutiny over the next decade.

2.4.1. Cloud must integrate with the edge

Cloud adoption has now reached every industry, from government to manufacturing. Ninety-four percent of IT professionals say their company is using public, hybrid, or private cloud. Of those using cloud, 84% are taking a multi-cloud approach, using the best that large public clouds, including Amazon Web Services, Microsoft Azure, and Google Cloud, have to offer (Flexera). In the next decade, we will see diminishing returns on cloud investments in terms of differentiating from competition. Innovators are now focusing on improving capabilities at the edge of networks. The capability to capture data at the edge and automate decision making will create new competitive advantages for organizations. Some innovators are already collecting data inputs such as human biometrics and using AI to determine emotional reactions to make better decisions. IoT devices can collect a tremendous amount of data at the edge, and how best to store and transfer all that data leads to more decentralized solutions. Some of this data is sensitive, and secure access needs will force society to grapple with new ethical considerations.

2.4.2. Continued automation

Through robotic machinery and software, many tasks that were previously assigned to people are now fully automated in certain controlled settings. Factory workers are the primary example of people being replaced by robots, but many other professions will be impacted. A new paradigm of work will emerge that involves both robots and humans working side by side for mutual benefit. This activity will create 133 million new roles in the workplace, for a net gain of 58 million new jobs according to the World Economic Forum.

2.4.3. Data explosion continues

Through 2025, annual data generation is expected to increase from 33 zettabytes to 154 zettabytes. Estimated size of global datasphere in 2025 is 163 zettabytes; a 10x growth from 16.1 zettabytes in 2016 (Berthier). One zettabyte is equal to one trillion gigabytes. Ultimately, the value of data is becoming more literally equated to equity.

More data online means more opportunities exist for misuse, which has numerous consequences for our society and economy. Some online services are being misused by manipulative algorithmic systems to amplify the spread of disinformation and for other harmful purposes. These new challenges and the way platforms address them have a significant impact on fundamental rights online. "Gatekeeper" platforms now bear an additional burden to show responsibility for their content. The European Union (EU) is leading the charge in creating a modern legal framework to ensure the safety of users online, establish governance with the protection of fundamental rights at its forefront, and maintain fair and open online platform environments. 2021 will be a landmark year for this trend to continue.

2.4.4. Harnessing AI

Artificial intelligence-driven technology is on the rise and is penetrating all manner of industries. The continued development of machine learning platforms is making it easier and convenient for businesses to utilize artificial intelligence. Some of the industries that are already adopting this technology include automotive, marketing, healthcare, finance, and so on. In the next decade, AI will also impact agriculture, aerospace, construction, logistics, robotics, connected mobility, and social media. Classifying and predicting data from the Internet of Things (IoT) with multilayer perceptron (MLP) intelligence will provide new findings and insights into connected devices. Intelligent data access, predictive analysis, enhanced customization, real-time marketing activities, virtual agents and AI-powered chatbots are coming soon. As AI-driven processes become more sophisticated and are specialized to solve more industry problems, there's no telling how much AI could influence existing business processes. On the home front we will continue to see more AI-driven smart devices/gadgets, media streaming gadgets, semi/autonomous vehicles, and home appliances. Advanced MLP and natural language processing (NLP) will lead to augmented analytics, augmented reality, and better speech recognition and interpretation. AI-software writing its own new software is not going to make a widespread impact any time soon, unless it can be taught to identify code patterns of vulnerabilities, weaknesses, and faults at the point of injection.

2.4.5. Cybersecurity

It is not surprising that 2019 saw over 3,800 publicly disclosed breaches as cyberattacks, and 2020 ended with one of the most publicized software supply chain attacks associated with the SolarWinds Orion compromise. Still, the number of cyberattacks continues to grow. It is estimated that by 2021, the global economy would bear the loss of $6 trillion due to cyberattacks. Ransomware damages alone are now predicted to cost the world $20 billion in 2021. The above estimates could be low since they may not include cybercrimes against individuals that don't get reported, such as identity theft. What makes most of these attacks successful by outside agents is vulnerable software systems – a problem we must vigorously address in the coming years.

Cybersecurity has become a chief concern and a business requirement among many business leaders. As businesses keep shifting and becoming more digital, they expose themselves more to the growing attack surface. With increasing awareness among businesses, 2021 will see an increase in spending on cybersecurity.

Recent reports from PricewaterhouseCoopers (PwC) claim that the cyber insurance market is expected to exceed $7.5 billion in premiums by the start of 2021. The PwC study also revealed over 60% of business owners recognize cyber as a significant threat to growing their business.

It's not surprising that privacy regulators have a tough time catching cybercriminals who breach cyber networks to siphon private data. Failed cybersecurity protocols form the crux of data breaches and ransomware attacks. Cybercriminals are evolving their tactics to outweigh the impact of cybersecurity investments by businesses. Even cybersecurity organizations are being hacked by well-organized nation states.

As this report is being released, cyberattacks against more than 18,000 government, consulting, technology, telecom and corporate entities in North America, Europe, Asia and the Middle East have been uncovered. Even Microsoft products may have been indirectly hacked. Additionally, certain nation-state troll farms are now franchising their tools and techniques to other troll farms.

Around 3.1 million professionals are needed worldwide to bridge the cybersecurity talent gap, according to the International Information System Security Certification Consortium (ISC2).

The top cybersecurity trends to be watching in 2021 are described below.

Critical infrastructure

Critical infrastructure is composed of the systems on which the societies heavily rely on. These include the electricity grid, water purification, traffic lights, shopping centers, and hospitals. To mitigate risk exposures attributable to exploitable software or reduce the impact of cyber-attacks, the organizations responsible for maintaining critical infrastructure must address software security. Organizations that utilize the critical infrastructure must also evaluate the amount of damage that is caused due to cyber-attacks. These organizations must have a contingency plan that would help their businesses to bear no brunt of the cyber-attacks.

Networks

Computer networks must be secured from intruders, targeted attackers, and opportunistic malware. As the Internet has an assortment of networks associated with various websites, it is often observed that organizations become targeted with unauthorized intrusion with malicious intent. Also, as many websites contain third party cookies, the users' activities are tracked. Sometimes this might prove helpful for organizations to grow their businesses, but often customers become prey to fraud and exploitation. Hence, to counter cyber-attacks and malware associated with the network, organizations must deploy a security program to monitor the internal network and infrastructure. Experts have suggested leveraging Machine Learning technology to alert the authorities in the case of abnormal traffic. Organizations must continue to upgrade their network security by implementing policies that can thwart cyber-attacks. Fixing vulnerable TCP/IP legacy code and network management platforms would be a good place to start.

Unfortunately, the global Covid-19 pandemic has created new incentives for cybercriminals. In July, hackers linked to a Russian intelligence service tried to steal information from researchers working to produce vaccines in the US, Britain, and Canada, according to security officials in those countries. Later, in December, with one or more vaccines being approved, sophisticated hackers, assumed to be state agents, have been carrying out a global phishing campaign targeting the vital "cold chain" that will protect coronavirus vaccines during storage

and transport, according to IBM security researchers and a US Cybersecurity and Infrastructure Security Agency (CISA) alert.

Cloud

With the plethora of data available, it becomes difficult for organizations to store this data in physical form. Also, it is observed that often this data is unstructured and derived from unknown sources, which can cause a risk to the organization's network. Hence, Amazon Web Services, Microsoft Azure, and Google Cloud present their customers with a cloud computing platform where users can store and monitor data by implementing security tools. Cloud services still rely on software to provide requisite capabilities.

Internet of Things (IoT) and Bring Your Own Device (BYOD)

IoT is being observed to be the next tool for the technological revolution. A report by Bain and Company has estimated the market size for IoT to expand by $520 billion by the year 2021. With the help of its security network, IoT provides the user with a variety of critical and non-critical appliances. The report suggests that one of the main obstacles for implementing IoT in any organization is the threat to security. By integrating the system with IoT security, organizations are provided with insightful analytics, legacy embedded systems, and a secure network.

Applications

Users rapidly get infatuated with different applications which include hardware, software, and devices. But an application becomes equally prone to cyber-attack or malware like the network. Application security thwarts cybersecurity infringement by adopting hardware and software assurance methods at the development phase of the project. With the help of an application security network, companies and organizations can detect sensitive datasets and secure them with specific applications. Some of the methods associated with application security are anti-virus, firewalls, encryption, and tools and practices focused on mitigating exploitable weaknesses and vulnerabilities in software architecture, design, and code.

Data

Data is software too. Data highways pose a rampant cybersecurity threat. The public concern about their data management has to be a top priority for businesses present online. The rising number of data breaches make it tough for businesses to ignore data security and privacy concerns. The usage of third-party data for business gains must fall under the following purviews:

- Individuals must know how their data will be used
- Data encryption cannot be ignored
- Individuals must have an option to forbid sharing their data
- Companies must report any public data breach within a stipulated time

Employees are increasingly getting involved in data leaks intentionally or unintentionally. 34% of cyber-attacks in 2019 were due to misdeeds of internal employees. Ubiquitous USB drives can transport massive information to aid in data exfiltration. Phishing expeditions are still quite successful, too. The source vectors for many data breeches continue to be unpatched software vulnerabilities.

Financial transactions

Blockchain is the go-to technology for enabling distributed trust, where intermediaries in a network rely on it to hold credible information about each participant's reputation, enabling transaction-level trust. This dynamic has enabled new governance models for systems securing IoT devices as well as systems designed to match problems to the best machine learning algorithms. It is also enabling new scenarios for data equity by allowing for digital scarcity. Blockchain solves the problem of data being infinitely copied without cost, meaning intellectual property owners can confidently enter digital markets where copyright is respected. In platform organizations, blockchain is being embraced as a tool that can effectively disintermediate relationships, enabling more efficient peer-to-peer connections where rules can be customized towards the end goals of the transactional parties.

2.5. The challenge to software engineering, software quality and costs

There are simply not enough good software developers around to create all the new and modified software that users need. Given the indirect as well as the direct contribution of software to the economic base of most industrialized countries, and considering the ways in which software can amplify the powers of the individual/teams/organizations, we cannot allow this situation to continue.

Just two percent of the worldwide population knows how to develop software, and the need is estimated to grow by 24% over the next seven years. There aren't enough educational programs available around the world to keep pace with the need. According to software.org, the number of jobs created directly by the US software industry increased 7.3% over 2016-2018, including a total of 14.4 million jobs (including indirect and induced impacts) and 3.1 million direct jobs, thus creating a Total Value-Added impact to the US GDP of $1.6 trillion (including indirect and induced impacts). The US Bureau of Labor Statistics (BLS) ranks IT-related employment as a field that is growing "much faster than average" and that software developers can expect to see ongoing robust growth through 2028. The BLS estimates that US software developer jobs will grow at a rate of 22% over the next decade.

To address this challenge and other pressures, software engineers are building new software out of available components to the extent possible. For example, currently popular open source software platforms include Fat Free CRM, InfluxDB, D3.js, R, TensorFlow, Keras, Serverless, Apache Airflow, Activiti, PrestaShop, and OpenCart. The quality of these software components is empirically unknown, and therefore might potentially introduce flaws that compromise success.

Software engineering is a highly challenging profession, made so because:

- Software size, complexity and technical debt is growing
- The nature of the problems to be solved is shifting
- Global disruptive forces are at play – e.g., cyberattacks, widespread disinformation, pandemic, etc.
- Rapid technology advancement
- End-user needs that constantly evolve
- The technology stack that a system is made of constantly evolves and changes
- The operating model constantly evolves to adapt to changing customer needs, changing technologies, and changing vendor environments
- The vendor ecosystem constantly evolves
- The necessary technical skills constantly evolve

- Data trustworthiness, protection and cybersecurity have become more important
- The pressures to deliver IT value at speed have not diminished
- Every line of code is a potential point of failure
- All software is affected by external factors (e.g., dependencies, constraints)

Software is inherently complex; the complexity of which often exceeds human intellectual capacity. Thus, the nature of software development is inherently difficult, and even more so when quality is not an explicit objective. Better tools are needed to automate development activities, especially in terms of preventing and mitigating flaws in software.

The definitions of software quality, good versus poor software quality, and the cost of software quality model that we advocate were described in Chapter 6 of our 2018 report. These concepts are assumed background knowledge for readers of the remaining sections of this year's report.

3. 2020 FINDINGS FOR COST OF POOR SOFTWARE QUALITY

The result of increased digitalization through software has created a balancing act trying to deliver value at high speed without sacrificing quality or security. As it turns out, we are not very good at balancing.

We have assumed in this year's findings that the cost of finding and fixing defects (although huge on its own) is fully represented in the first three categories below. Cybersecurity and Technical debt will be addressed separately. Cybersecurity has a measurable and growing impact on the cost of software quality. Technical debt has a growing impact on future cost.

The main categories representing different components of the cost of poor software quality (CPSQ) are:

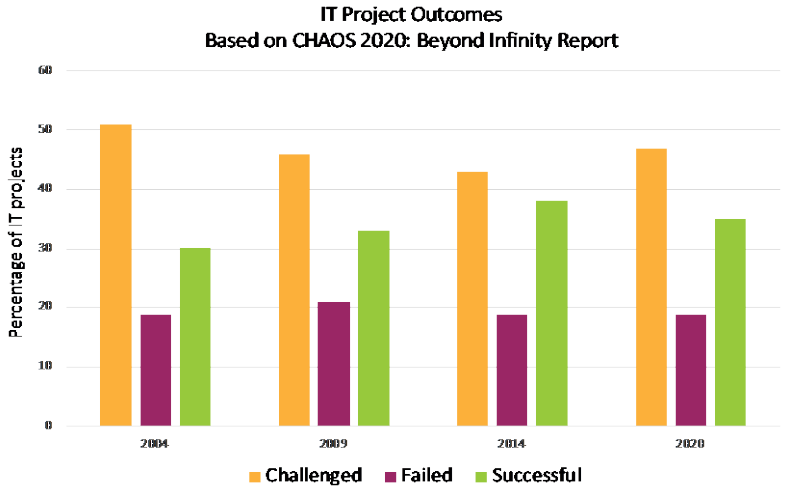
1. Cost of Unsuccessful IT/Software Projects
2. Cost of Poor Quality in Legacy Systems
3. Cost of Operational Software Failures
4. Cost of Cybersecurity and Technical Debt

In the following sections we present our CPSQ findings for 2020 followed by a brief discussion of technical debt and a summary of the total CPSQ in the US. Those readers who are interested in a retrospective summary of our 2018 report findings and conclusions are referred to Appendix A of this document.

3.1. Cost of Unsuccessful IT/Software Projects

Since 1994, The Standish Group has been reporting on IT project outcomes and building their database of 50,000 projects. At the end of March 2020, The Standish Group released its latest report CHAOS 2020: Beyond Infinity. In that report they stated that only 35% of projects were fully successful with respect to time and budget. Their data says nothing about the quality of the result; presumably those projects had successful outcomes as well. They show a staggering 19% of projects will be cancelled before they get completed, and 47% are challenged (over budget, behind schedule, and produce low quality deliverables). The cost of cancelled and challenged projects is usually hidden below the tip of the proverbial iceberg. The historical trend is seen in the summary figure below based on The Standish Group data. Project failure rates have been steady at ~19% for many years.

Figure 2: IT Project Outcome Rates: 2020



This year for the first time, The Standish Group tried to tease out why there might be such large variation in project outcomes. To do that, they created a new definition of project maturity, which broke away from the notions of organizational maturity (ala CMMI). Their new definition incorporates three factors of success, namely, having in place: a good sponsor, a good team, and a good place.

They also claim that success rates go up dramatically when using agile + DevOps in a highly mature way, thus leading to decision latency being minimized. They envision a seamless delivery pipeline between software development, implementation, and services delivered. Their pipeline concept is segmented into a series of cross-functional teams. Using this definition, they claim that highly mature projects are almost 90% successful, whereas immature projects are 10% successful. Failed projects are only 2% likely given high maturity. It is not clear how they came up with these likelihoods. Therefore, we will watch The Standish Group's maturity concept materialize in the coming years.

Recall from section 4.3 that the global computer and information technology industry was on pace to reach $5 trillion in 2020, with the US representing 32% of the total, or approximately $1.6 trillion for 2020. So, if the US spent $1.6 trillion in 2020 on IT, and 25% of that was spent on development projects, that equates to $400 billion. And if 65% of that was spent on failed and challenged projects, the CPSQ due to unsuccessful projects in the US in 2020 is $260 billion (up from $177.5 billion in 2018).

3.2. Cost of Poor Quality in Legacy Systems

Using the same approach as 2018, we created an estimate of the overall cost of poor-quality legacy software in operation and maintenance (O&M). Following that reasoning, we estimate that the US spent $1.6 trillion in 2020 on IT, and that ~75% of that was spent on legacy systems, or $1.2 trillion. If $1.2 trillion is being spent on legacy systems, and if as much as two-thirds of that could be classified as waste, that gives us an approximate upper bound of $800 billion on the cost of poor-quality software in O&M from a maintenance perspective. This waste does not include additional costs incurred outside of the IT organization. The lower bound using only corrective maintenance in the calculation would be $240 billion. The mid-point between the upper and lower bound would be $520 billion, which we take as our first order approximation of the CPSQ due to legacy systems in 2020.

Another way to approach this is to approximate the total IT wage base and estimate how much of that is being spent on legacy systems. Using the US Bureau of Labor Statistics (US BLS) data allowed us to calculate the US population of IT professionals and their approximate total salaries over all IT labor categories to come up with a total level of US IT salary expenditure for 2018 of $600 billion. Adding in the missing categories from the BLS database and gig workers, we concluded that the total US IT professional wage base in 2018 was ~$1 trillion, with about 6.6 million Americans working in computer or IT fields. It was hard to get any information on the number of gig workers. At that time, IT professionals were also experiencing annual salary growth of about 3.4-4.0%.

US BLS says there were 1.4692 million software developers in 2019 with a median pay rate of $107,510 per year. When you add up all of the other computer and IT occupations in the BLS, which are clearly involved with software, you get an additional 2.733 million jobs. BLS says the median annual wage for all computer and IT occupations was $88,240 in May 2019. Another report claims that the average annual wage for software developers in the US was $114,000 in 2018. They probably included categories not in the BLS database.

But the BLS database does not include an additional set of occupations involved with software. For example, BLS does not contain:

- IT managers – 461,000 of them at a pay rate of $146,360
- IT project managers/SQA/testers– 412,800 of them at a pay rate of about $88,550
- Computer systems engineers – number and pay rate unknown
- CIOs, CTOs and other software C-suite executives – pay rate unknown

Employment in computer and IT occupations was projected by BLS to grow 11% from 2019 to 2029, much faster than the average for all occupations. Other reports have claimed that employment growth in this field has been 7.3% over the last 2 years. For ease of computation, assume that the growth is ~10%. In 2018 we estimated 6.6 million IT workers in the US, so in 2020 there would be about 8 million workers in this field, and they are making anywhere between $88,500 and $150,000, plus additional C-suite software executives that are making much more. There are also many gig workers who are being paid hourly and have not been counted at all. When we factor in higher-level jobs missing from BLS data, we could conclude that the average IT worker salary is closer to $150K/year, which would give us an approximate wage base total for all US IT/software jobs in 2020 of $1.2 trillion**.** Which is not very different than the $1.6 trillion we calculated with our initial approach. Splitting the difference gives us an estimate of $1.4 trillion.

So, we will stick with $520 billion as our first order approximation of the CPSQ due to legacy systems in 2020. This is down somewhat from our 2018 estimate of $635 billion, which seems plausible as scarce resources may have been shifted and senior IT maintenance staff retired.

3.3. Cost of Operational Software Failures

2019-2020 was dominated by the biggest software failures in recent history including ransomware attacks, cybersecurity attacks, IT outages, and data leakages that affected some of the biggest companies and millions of customers around the world. Software failures wreaked havoc at banks, airlines, and in government systems, costing billions of dollars in damage and creating devastating disruption. Cybercrimes enabled by exploitable weaknesses and vulnerabilities in software are the largest growth area by far in the last 2 years.

This area of CPSQ is the most difficult to estimate due to the dearth of real empirical data on the subject and that many/most of these failures are not reported. It is also the area where rapid growth is most likely. In 2018 we estimated that the CPSQ due to failures was $1.275 trillion. Continued growth in failure reports tells us that in 2020 this should be much higher.

Over the last two years, we compiled a list of the most impactful software failures in the English news media, as anecdotal indicators of the larger problem. The table below lists some of the top 2018-2020 IT failures in the news (in no particular order), representing just the tip of the CPSQ iceberg.

Table 1: Software Failure Stories in the News

Home Depot (update)	On 11/21/2020, six years after its 2014 data breach, Home Depot agreed to pay US states $17.5 million for an incident that compromised payment card data of 56 million individuals.
Iowa Caucus Smartphone app	In preparation for the 2020 Iowa caucus, the Democratic Party wanted to update its reporting infrastructure, moving away from a system in which the state precincts would phone in results to a smartphone app used to simply upload the information. The party (as did Nevada) paid the new company, Shadow, $60,000 over a few months to develop an app called IowaReporterApp. The app sends the results from 1,700 precincts to a central office for tabulation. The caucus runners had to take and upload a picture of their results, which were then supposed to be captured by the app. But the app was not well-tested with regards to its intended usage environment and installation was not simple. This situation was clearly a system and process failure, and perhaps included coding errors. Shadow probably had almost no quality system (QA/QC) for this rushed effort. The damage done to the reputation of the Iowa caucus was unenumerable.
NASA/Boeing Starliner	NASA awarded Boeing nearly $5 billion to develop the Starliner spacecraft, which is built to carry as many as five people. Now more than three years behind schedule, the project is under heavy scrutiny from a NASA investigation, which is taking a closer look at problems found during the December 2019 test flight. The goal of that mission was to fly Starliner without crew to the International Space Station (ISS), deliver cargo, and return safely to demonstrate its capabilities and safety. But the spacecraft did not dock with ISS after a software issue during launch caused Starliner's autonomous flight-control system to misfire, putting Starliner in the wrong orbit. The mission was cut short, and the Starliner landed. Additionally, NASA confirmed that Boeing's Starliner suffered not one but two major software defects during that test flight. The latter software problem could have caused "catastrophic spacecraft failure," a panel of NASA safety experts said, if Boeing had not caught the issue during the mission. The second software issue was a piece of code that could have caused two pieces of Starliner – its "crew module" and "service module" – to collide in orbit before the spacecraft re-entered the atmosphere. "What we wish we'd done better was the software, so there's a lot of learning there," Boeing Senior VP Jim Chilton said. Boeing reported that it took a $410 million charge in 4Q2019 in case another uncrewed flight is determined to be necessary. Future commercial crew contracts will be up for grabs, as NASA would look to buy seats on SpaceX's Crew Dragon.
Heathrow Airport Disruption	On February 16, 2020 more than 100 flights were disrupted when their departure and check-in systems were hit by "glitches." This was fixed by the next day.
Google Plus Security Glitch	A vulnerability in the Google+ social network exposed the private information of nearly 500,000 people using the social network between 2015 and March 2018. The major part of the problem was a specific API that could be used to gain access to non-public information. The software glitch allowed outside developers to see the name, email address, employment status, gender, and age of network users. The error had been discovered in March 2018 and rectified immediately. The interesting part is — Google did not share information about the bug in Google+ at first, trying not to get into the limelight of the Cambridge Analytica scandal and become noticed by

	regulators. Google said they had no evidence of data misuse but also cannot say there was none. In any case, the tech backlash ended sadly for Google+ – the consumer version of the network was shut down shortly afterward.
Facebook's User Data-Leak	In late 2018, Facebook, whose ability to handle private information had already been questioned, confirmed that nearly 50 million accounts could be at risk. Hackers exploited a vulnerability in the system that allowed them to gain access to accounts and possibly to the personal information of Facebook users. There were three software flaws in the network's systems that allowed hackers to access user accounts, including founder Mark Zuckerberg's. The hackers likely exploited a vulnerability in the "View as" code, the function that allows checking how a profile looks as seen by other people. This, in turn, resulted in the acquiring of authentication tokens. 90 million users were logged out of their accounts the day the vulnerability was discovered. An additional 40 million accounts were logged out as a preventative measure. Over 540 million records on Facebook users were eventually exposed to Amazon cloud servers.
Sacramento Bee Records Breach	In February 2018, as many as 19.5 million records were breached from two databases owned and operated by The Sacramento Bee, a daily newspaper published in Sacramento, California. These IT assets comprised California voter registration data that was provided by California's Secretary of State. It also included the stored contact information of the subscriber base for the newspaper. The attacker demanded a ransom for releasing this data to the organization. Ultimately, the newspaper deleted the database to avoid any further risks and attacks. The costs to restore it were not revealed. Ransomware attacks have been one of the most prominent kinds of attacks on all segments of organizations in recent history. These risks are increasing, reinforcing the need for robust cybersecurity and data safeguarding measures.
Ticketfly Gets Vandalized	In May 2018, Ticketfly confronted an attack that vandalized its concert and sporting-event ticketing website. This resulted in total disruption almost for a week when the website had to be taken down. Apparently, the hacker had alerted Ticketfly about a vulnerability and had asked for a ransom to get it fixed. Eventually, when the company refused, the website was hijacked, and the homepage was replaced with customer and employee data. The sensitive data included names, addresses, email addresses, and even phone numbers of as many as 27 million Ticketfly account holders.
Equifax Data Breakdown	The Equifax breakdown has been one of the most prominent data breaches where sensitive information on millions of Americans was exposed. It included passport details, driver's license details, social security numbers, and much more. This affected the data of over 146 million consumers. The disclosure was made by the Securities and Exchange Commission and given to congressional committees who were investigating the breach. As noted in a CSO Online article on the biggest data breach fines, penalties and settlements, in 2020, Equifax was made to pay further settlements relating to the breach: $7.75 million (plus $2 million in legal fees) to financial institutions in the US plus $18.2 million and $19.5 million to the states of Massachusetts and Indiana, respectively.

British Airways glitch	On August 7, 2019, over one hundred British Airways flights were cancelled and near to three hundred flights delayed. During the busiest month for air travel, their computer system went down completely. Thousands of passengers had to stay behind and wait long hours in packed airports. The check-in procedures had to be switched to manual which made the queues start to resemble Dante's "Inferno." This was not the first time the system screwed up. The pattern of software failures over the last couple of years suggests poor computer management and calls for concern. Investors are worked up because the financial risk with such issues is high.
Ticketmaster data breach	The UK Information Commissioner's Office fined Ticketmaster UK Ltd. £1.25 million ($1.6 million) for a 2018 data breach that potentially exposed information from more than 9 million customers in Europe, BBC reports. Hackers accessed payment card details from Ticketmaster customers through a chatbot operated by a third-party supplier. Ticketmaster took nine weeks to start monitoring fraud on its website after it received alerts about possible fraud. Investigators said 60,000 payment cards from Barclays Bank customers were subject to fraud as a result of the breach. The regulator said Ticketmaster violated Europe's General Data Protection Regulation by failing to assess the risks of using the chatbot on its payment page and the company did not use appropriate security measures or identify the breach fast enough.
The Boeing 737-Max MCAS System	Although this situation was not exactly a classic software failure, it is still instructive to view how the software was misused within the larger context of overall system failure. According to a well-regarded 737 pilot, "everything about the design and manufacture of the Max was done to preserve the myth that 'it's just a 737.' Re-certifying it as a new aircraft would have taken years and millions of dollars." The problems with the 737 Max emerged after the Lion Air crash on October 29, 2018. Two planes crashed and 346 died, causing Boeing stock to lose $28 billion in one week. In the 737 Max, everything is monitored, if not directly controlled, by computers. From a technical perspective, Boeing produced a dynamically unstable airframe. Boeing then tried to mask the plane's dynamic instability with a new software system, MCAS. MCAS relied on other systems known for their propensity to fail (angle-of-attack indicators) and did not appear to include even rudimentary provisions to cross-check the outputs of the angle-of-attack sensor against other sensors, or even another angle-of-attack sensor. MCAS took complete control of the plane away from the pilots. Pilots were also not trained on how to disable the MCAS system. MCAS software was not well designed, implemented, or tested by knowledgeable system engineers. In reaction to the crashes and the FAA examining the plane's certification process, the 737 Max was grounded in the US and has not flown commercially since then. The company suspended production of the 737 Max in January 2020. Additional software flaws were found in February 2020. Since then, Boeing rolled out a set of software updates to the 737 Max flight control system. These updates include additional layers of protection and additional redundancy and safety capabilities. In August, the FAA said Boeing's software changes and other design changes are acceptable if instituted and inspected fully. As of March 2020, the cost of the Boeing 737 Max crisis was estimated at $18.7 billion and counting. This includes $8.3 billion in compensation to the company's airline customers, and $6.3 billion in increased costs to build the jet. That price tag could go significantly higher. Financial analysts estimate it will be about $23 billion, excluding liability for loss of life and interest expenses. Nor do the estimates include the cost of litigation - not just the suits filed

	by the families of the crash victims, but from shareholders and even some airline employee groups, such as one filed by Southwest Airlines pilots to compensate them for Boeing shares that dropped from the end of 2018 ($316) to November 2, 2020 ($144) – a loss of 54% of its market value and hitting a low of $89 in March 2020. The longer the crisis drags on, it becomes more likely that airlines will cancel or delay delivery of planes that they ordered, especially now that the coronavirus has decimated airline bookings and fares. Boeing's 737 MAX backlog stood at 3,408 planes as of August 31, 2020. Boeing has seen the cancellation of 433 orders of the 737 MAX jets and also removed 522 orders from the backlog for accounting purposes. Future deliveries are highly uncertain at this time but are expected to be considerably lower than planned. Boeing has ~450 planes in inventory waiting to be delivered to customers. Losses to Boeing and the airlines will probably be in the mega-billions. On November 18, 2020, the US FAA cleared the way for the MAX to return to operation, saying that changes in software, design, and training have made the plane now safe to operate. American Airlines will try to start flying passengers on the MAX by the end of this year. On December 4, 2020, Ryanair Holdings said it agreed to buy 75 of Boeing's 737 MAX jets. The European budget airline's deal is worth more than $7 billion at current list prices. Ryanair chose to exercise the option it had to buy additional aircraft on top of 135 it had previously agreed to buy from Boeing.

Most of these fiascos result from software flaws inside a system. Three trends magnify the impact of software flaws, driving business liabilities toward the billions:

1. **Digital Transformation:** a far greater slice of business operations from sales to delivery is integrated and controlled by software, thus rapidly spreading the effects of a malfunction across the value chain.
2. **Systems of Systems:** expanding complexity exponentially and concealing the triggers for huge failures in a thicket of cross-system interactions.
3. **Increased Competition:** especially online, has prioritized speed-to-business over operational risk and corrective maintenance costs, a huge gamble for systems not designed to expect and manage failures.

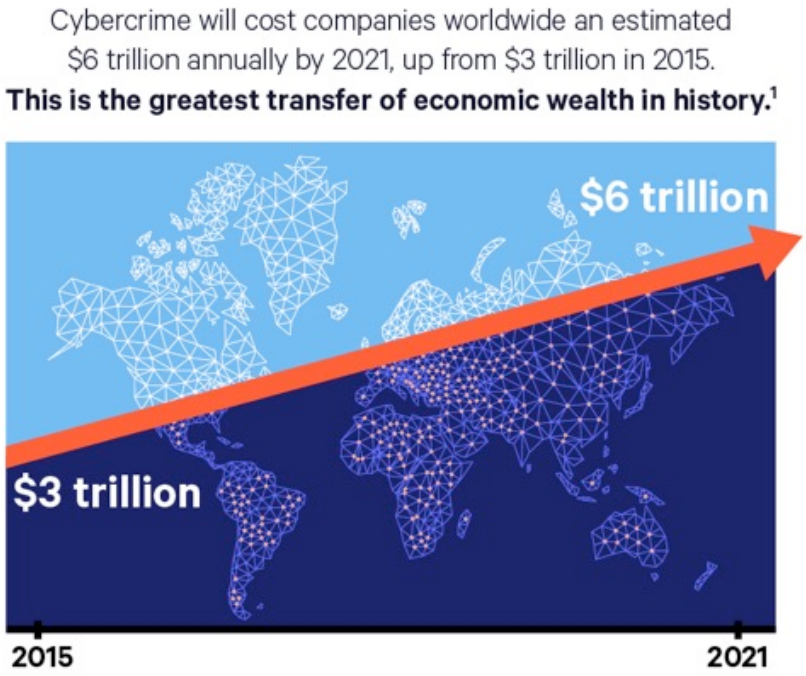
If the trend toward billion-dollar defects continues, or even accelerates, the status quo in IT will change, and not from inside the IT community. If the tipping point for greater regulation and centralized control has not been passed, avoiding it will require greater adherence to software best practices that move software development toward a real engineering discipline. For example, better architectural and coding practices can implement the internal system safeguards that limit the damage from potentially devastating defects long before they spiral out of control.

3.4. Cost of Cybersecurity and Technical Debt

3.4.1. Spotlight on Cybercrime

The projected growth of failures due to cybercrime is well-summarized in this infographic from Embroker and related facts.

Figure 3: Cybercrime $$ Growth Trend: 2015-2021



Since the US IT market represents 32% of the world's total, this might mean that the US share of 2021 cybercrime costs could be in proportion, and thus ~$1.92 trillion. This is much higher than other studies in the US which have estimated totals in the billions (~$100-200 billion).

The total cost of malicious cyber activity directed at US entities is difficult to estimate because many data breaches go undetected, and when they are detected, they are mostly under-reported, or the final cost is unknown. While no one has complete data on adverse cyber events experienced by firms, cyber insurance firms are arguably in the best position to collect reliable data. Firms that sell cyber insurance products need to use probability assessments and expected cost estimates for adverse cyber events to price their products. Moreover, insurance firms are best able to collect unbiased datasets because they track the same firms over time and observe otherwise undisclosed cyber-attacks and data breaches. A study performed by Advisor Smith Solution Inc. found that the average cost of a cyber liability policy in 2019 was $1,500 per year for $1 million in coverage, with a $10,000 deductible. A European consortium is investing $1.18 billion over the next five years to fund homegrown cybersecurity companies.

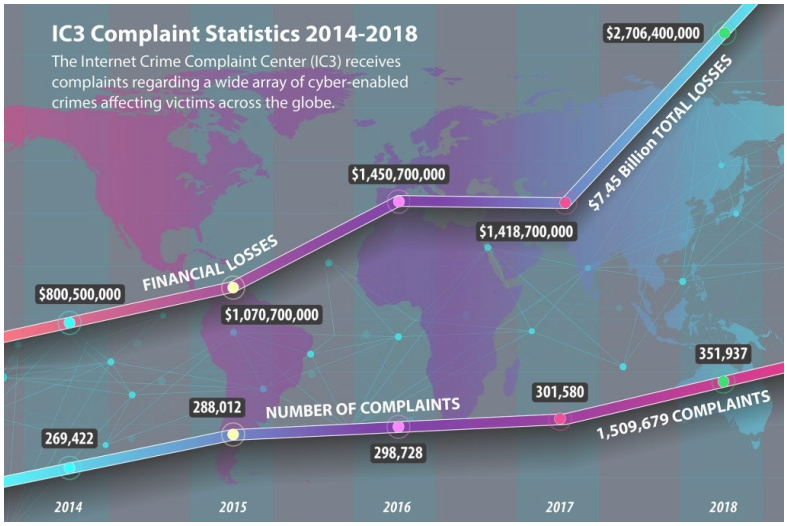
It is predicted that a business will fall victim to a ransomware attack every 11 seconds by 2021. Ransomware damages are now predicted to cost the world $20 billion in 2021. Ransomware attacks on hospitals are predicted to increase 5X by 2021. The average ransom paid out to hackers in the third quarter of 2020 was $233,817, according to cybersecurity firm Coveware. This fall, schools have increasingly been the target of ransomware attacks where hackers encrypt an organization's data and do not release access until they receive payment. Ransomware attacks on the manufacturing industry tripled in 2019, with at least 262 vulnerabilities found in industrial equipment. They paid the ransom because they require continuous uptime to keep production running.

Nearly half of all cyber-attacks are committed against small businesses, and the percentage is expected to continue rising. Recently Microsoft said that Russian and North Korean operatives were trying to hack

coronavirus research firms. The theft of personally identifiable information (e.g., identify theft) is widespread in the US.

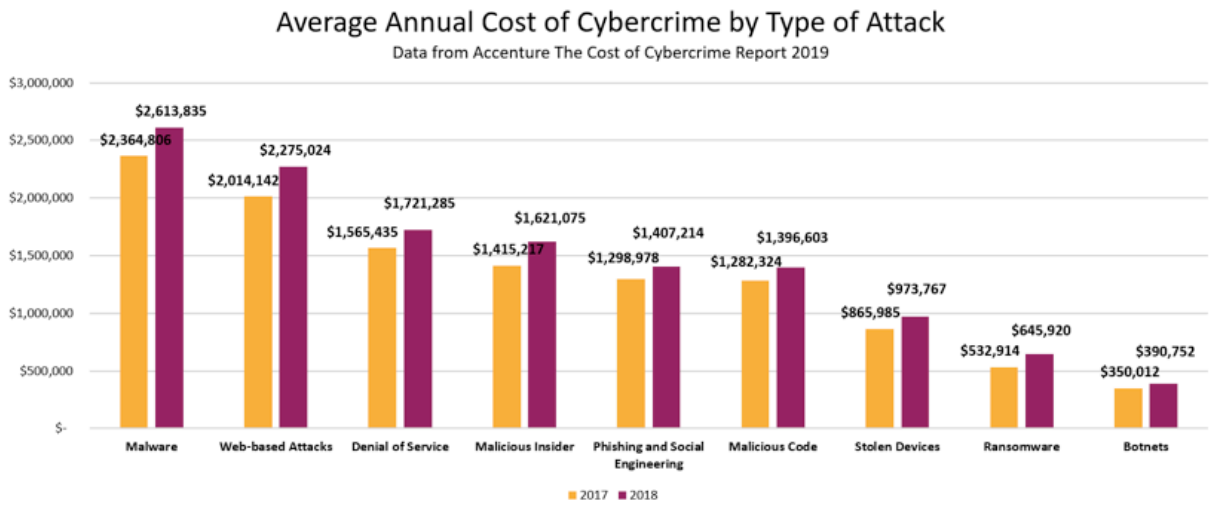
According to the FBI IC3, over the past five years cybercrime costs have increased to $10.2 billion, with $3.5 billion in 2019 alone. The trendline is seen in the graphic below. This is based on complaints generated from within the US.

Figure 4: Cybercrime Growth Trend in US



The United States topped the list of all countries with the average annual cost of cybercrime increasing by 29 percent in 2018. The total cost of cybercrime for each company was $13 million—a rise of 12 percent from 2018, according to Accenture, which also shows the cost growth by attack type over one year.

Figure 5: Cybercrime Cost by Attack Type



The least subjective method for estimating the impact of a cybersecurity events on a publicly traded firm is to quantify its stock price reaction to the news of such events. Firms on average lost about 1% of their market value in the seven days following news of an adverse cyber event. The most damaging to a business is IP theft.

According to a Computer Economics report, cybersecurity technology is the number one investment and adoption technology in 2020.

3.4.2. Attributing cybercrimes to software growth

Causing rapid growth in cybercrime is the rapid growth of the attack surface. According to Cisco VP, Susie Wee, in May 2017 there were more than 111 billion lines of new software code being produced each year — which introduces a massive number of vulnerabilities that can be exploited. Additionally, the world's digital data content was expected to grow from 4 billion terabytes (4 zettabytes) in 2016 to 96 zettabytes by 2020. It is also predicted that there will be 6 billion Internet users by 2022 (75% of the projected world population of 8 billion). Cisco estimates that the number of IoT devices will be three times as high as the global population by 2021.

In 2018, there were 500 million personal records stolen, according to the Identity Theft Resource Center. In 2019, 7.9 billion records were exposed by over 5,000 breaches, as reported by Risk Based Security.

Security vulnerabilities often stem from simple errors in software coding. In typical software code, there are an average of 25 errors per 1,000 lines of code (NIST 2016). Systems with near-zero errors are produced routinely today in the aerospace industry, but at several times the cost of ordinary software. Reducing this error rate will have substantial costs associated with its implementation, but ultimately will pay off through a sufficient reduction in software vulnerabilities.

3.4.3. Software technical debt

In 2018, we reported the amount of software technical debt in the US was approximately $1.145 trillion, which represented just the debt principal without accumulated interest. At that time, we assumed a code growth rate of 35 billion LOC per year, projecting that there would be 1.455 trillion LOC worldwide (US share of 31%). We have since seen that code growth is now up to ~100 billion new LOC per year, or ~7% growth per year. Projecting those figures to 2020, and assuming that very little code has since been retired, there would now be 1.655 trillion LOC worldwide and 513 billion in the US. The US figure for technical debt in 2020 would therefore be $1.31 trillion.

3.5. CPSQ 2020 Conclusions

The 5th Edition of the Tricentis Software Fail Watch reported 606 major software failures from 2017, causing a total loss of $1.7 trillion in assets at 314 companies. This averages out to $2.8 billion per failure. These were only those reported by English language news media. We do not know the relative percentage of the reported failures by country, but they are very likely dominated by US, UK, and Australia. With software growing at ~7% per year, it might be expected that failures would grow at about the same rate, or by about 22.5% through 2020. Extrapolating from the 2017 base, we could say that ~742 failures would likely occur in 2021 in English language media sources, for a total of $2.08 trillion. If we conservatively attribute 1/3 of that to the US, then $686 billion would be our share. The ones that make it into the media reports are likely the most impactful. We also know

that most failures are not reported for a variety of reasons - largely because firms face a strong disincentive to self-report negative news.

In 2018, we estimated that the CPSQ due to failures was $1.275 trillion. For 2020, we estimate that it is ~$1.56 trillion, but that could be low given the meteoric rise in cybersecurity failures.

As we approach the end of 2020, several factors are merging to cause a potential perfect storm for helping cybercriminals. The pandemic surge has led to health care systems in the US overloaded and under-supported, and thus susceptible. International supply chains of critical products (e.g., PPE, test kits, vaccine elements) and software components are huge targets. The holiday season also creates many opportunities. For example, the amount of money stolen last Christmas through online shopping fraud in just the U.K. was £13.5 million, according to that country's National Fraud Intelligence Bureau.

In 2018 we also reported that $500 billion was being spent on finding and fixing software bugs in the US. With the US code base growing at ~7% per year, and IT wages growing at ~3% year, then the amount spent in 2020 finding and fixing software bugs would be ~$607 billion, which is felt across all three categories below, but by itself represents a significant target for CPSQ.

The CPSQ in 2020 determined in our three main categories is:

- $260 billion due to unsuccessful projects (up from $177.5 billion in 2018)
- $520 billion due to legacy system problems (down from $635 billion in 2018)
- $1.56 trillion due to software failures in operational systems (way up from $1.275 trillion in 2018)

There could be some overlap between legacy system problems and operational failures, so we will discount legacy system costs by 50%. Therefore, the estimated total of US CPSQ for 2020 is $2.08 trillion.

4. RECOMMENDATIONS

Given the CPSQ results from the previous section, here we turn to the question "what can we do about it?" We will examine this question by category in the order of highest impact.

4.1. Operational software failures

The biggest bucket of CPSQ identified is in operational software failures ($1.56 trillion in the US in 2020), which is ~10 times costlier than finding and fixing the defects causing it before releasing software into operation. The biggest bang for our CPSQ investment is in preventing those defects from occurring as early as possible when they are relatively cheap to fix. The second line of attack is isolating, mitigating, and correcting those failures as quickly as possible to limit the damage done.

The CAST Crash 2020 report gives us insight into where to look for potential CPSQ improvements. Their current benchmark on the structural quality of IT applications was developed from their database of 2,505 applications consisting of 1.549 BLOC (billions of lines of code), distributed across 533 organizations and 26 countries. The five software quality characteristics analyzed in their report are Robustness, Security, Performance Efficiency (aka Performance), Changeability, and Transferability. These are four out of the eight major software quality characteristics found in the ISO/IEC 25010 software product quality model standard (Changeability and Transferability are sub characteristics under Maintainability in ISO/IEC 25010).

CAST's findings were based on calculating densities of weaknesses in applications and included:

- The size of an application had negligible to no relation to its structural quality.
- Densities of critical weaknesses for Security were higher than those for Robustness and Changeability.
- The lowest densities were observed for Transferability (equivalent to Understandability).
- Industry segment is of lesser importance than other factors – but this only applies to Java-EE applications, for which Telecom, Software ISVs, and IT consulting had the highest densities of critical Robustness, Security, and Changeability weaknesses.
- Most industries showed wide variability in critical weakness densities and numerous extreme outlier scores.
- Security was the area where the mean densities of critical weaknesses and variability of scores were the highest.
- The factors that most affect quality attributes like Robustness, Security, or Changeability appear most likely to be specific to the application, the development team, and the specific conditions in the development environment.

Selected recommendations from the report were:

- Greater attention must be given to secure coding practices as many applications had densities of critical Security weaknesses that were unacceptably high. Security scores displayed wider variation than those of any other quality characteristic.

- Analyze source code on a regular basis prior to release to detect violations of quality rules that put operations or costs at risk. System-level violations are the most critical since they cost far more to fix and may take several release cycles to fully eliminate.
- Treat structural quality improvement as an iterative process pursued over numerous releases to achieve the optimal quality thresholds.

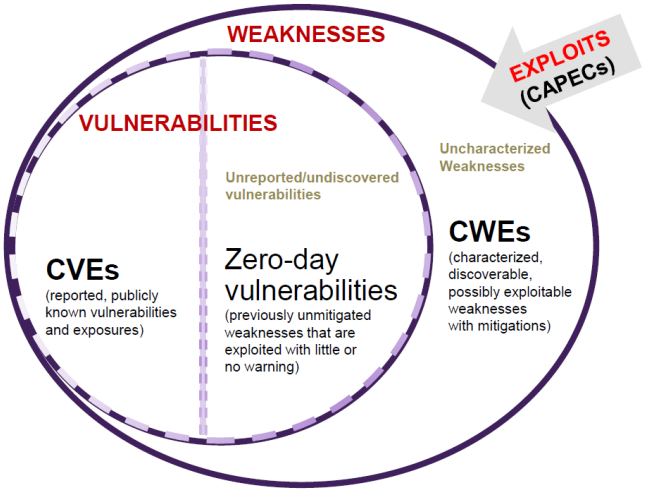
While adopting these evidence-based recommendations cannot guarantee high structural quality, they have been shown empirically to be associated with lower risk applications.

Many often consider "vulnerabilities" and "weaknesses" in software as interchangeable words. While they are related, they are different. These terms are defined in international standards. Standardized definitions for weaknesses and vulnerabilities are part of the ITU-T CYBEX 1500 series (CVE ITU-T X.1520, CWE ITU-T X.1524, CAPEC ITU-T X.1544) as outlined below.

- **Weakness:** Mistake or flaw condition in architecture, design, code, or process that if left unaddressed could under the proper conditions contribute to a cyber-enabled capability being vulnerable to exploitation; represents potential source vectors for zero-day exploits.
- **Vulnerability:** Mistake in software that can be directly used by a hacker to gain access to a system or network, or **Exposure:** Configuration issue or a mistake in logic that allows unauthorized access or exploitation.
- **Exploit:** Action that takes advantage of weakness(es) to achieve a negative technical impact.

The existence of an exploit designed to take advantage of a weakness (or multiple weaknesses) and achieve a negative technical impact is what makes a weakness a vulnerability. Weaknesses are listed in the Common Weakness Enumeration (CWE) Repository (cwe.mitre.org). Vulnerabilities (CVEs) are published in both the Common Vulnerabilities and Exposures dictionary (CVE.mitre.org) and the National Vulnerability Database (nvd.nist.gov). The methods bad actors use to exploit weaknesses and vulnerabilities are enumerated in the Common Attack Pattern Enumeration and Classification (capec.mitre.org). The figure below summarizes the relationships between these concepts.

Figure 6: Vulnerabilities, Weaknesses & Exploits



Because the number, size, and complexity of software systems increases every day, so do the number of weaknesses and vulnerabilities. Cyber attackers use known vulnerabilities and weaknesses to exploit systems. Eliminating known vulnerabilities (CVEs) and the most egregious weaknesses (CWEs) would substantially reduce the impact from cyberattacks and data leakages. See the latest version of the CWE, the top 25 CWEs, and 100+ CWEs in the OMG Automated Source Code Quality Measure standards developed by CISQ. If all new software (111 billion LOC per year globally) was created without these known vulnerabilities and exploitable weaknesses, the CPSQ would plummet.

As indicated by the growth in data breaches, data protection and privacy are at the top of many organizational priorities. Many organizations will be undergoing process assessments associated with regulations to protect data, including General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), Health Insurance Portability and Accountability Act (HIPAA), and Cybersecurity Maturity Model Certification (CMMC).

Scanning code that will run or is running in enterprise network-connected assets that process or transmit data would determine if the systems or devices enable data leakage or lack adequate protections to mitigate unauthorized access to read or modify data. If so, then such a scan would reveal if the data protection/privacy controls associated with the process assessment were inadequately implemented.

To address this, CISQ developed an Automated Source Code Data Protection Measure (ASCDPM) that can be used in application security testing and software development to provide independent verification of processes revealing source vectors for data leakage or data corruption; providing indicators for non-compliance with respective data protection and privacy guidelines. Based on the CWE, the measure elements (weaknesses violating software quality rules) that compose the CISQ ASCDPM contain 36 parent weaknesses and 53 contributing weaknesses.

4.1.1. Low quality development practices

Traditionally, software development does not include quality concerns or robust testing until the end of the development cycle, meaning that code is often produced to be functional but not necessarily of high quality. Often, the amount of time and effort needed to retroactively make code of high quality, in combination with tight deadlines, means that weaknesses and vulnerabilities are more likely to be ignored or insufficiently corrected. As development teams take on agile methodologies and the speed of development and deployment increases, this problem can be compounded. So, “shift-left” quality becomes important. Automated tools have begun to positively impact this area (e.g., Jenkins, Git, Jira, Undo LiveRecorder, OverOps, etc.). Manual techniques such as software inspections have seemingly fallen out of favor. Static and dynamic analysis techniques are often used when supported by tools. More advanced capabilities, like software failure replay, are increasingly being used to accelerate software defect resolution in development, test or production. Unfortunately, software quality measurement and tracking practices are not as robust as they could be. Defined software quality objectives are often an afterthought, when they should become a forethought.

No-code/low-code development platforms are relatively new on the scene and are being overhyped as expected. Essentially, when you build with a no-code platform you are building with code, it is just hidden (so you don’t have to write it yourself). These platforms typically contain a GUI builder, a visual modeling tool (e.g., UML), reusable components, and an integration technology. As such, the application domains to which these can be applied is quite limited. And, do you really trust all of the components that you can’t verify (see the next section)? Over my six decades in this industry, I have seen similar concepts many times before (e.g., higher level programming languages, 4GL, CASE, RAD, OOP, application generators, “automagical” programming, high quality

reusability, AI, agile, CI/CD), none of which have been or will be a silver bullet solution to our software quality problems.

4.1.2. Inclusion of flawed components

Most applications make use of one or more open source code modules or libraries, many of which have weaknesses and vulnerabilities. Although CWEs and CVEs are known, and resources exist explaining how to correct them, developers are not always aware of this information or different versions might be used inconsistently. If only snippets of code are used or libraries are used in a "black box" fashion, it is difficult to know whether the components are vulnerable without analyzing them directly. Third-party integrations and deployment environments can also introduce unforeseen risks or create conflicts that can be exploited. This is happening more frequently than CI/CD developers would like to admit. One of the major such components are cloud systems (e.g., cloudhopper attacks) which are presumed (only) to be secure. In fact, any vendor supplied platform or subsystem should be evaluated for vulnerabilities.

Open source software (OSS) plays a critical role in today's IT ecosystem. The overwhelming majority of modern codebases contain open source components, with open source often comprising 70% or more of the overall code.

There is useful data on the extent of the OSS problem from a recently published Synopsys study. The Synopsys Black Duck Audit database represents open source activity from over 20,000 sources worldwide. Their 2020 report (the 5th of their series) described their 2019 study of the audit findings from 1,253 commercial codebases in 17 industries. By codebase they mean the source code and libraries that underlie an application, service, or library. Their findings included the following:

- 82% of the open source components found were out of date (i.e., unpatched or not well supported)
- 99% of codebases audited contained open source components
- Open source made up 70% of the audited codebases (doubled in 5 years)
- 75% of codebases contained vulnerabilities (up from 60% in 2018), and 49% contained high risk vulnerabilities (e.g. Heartbleed)
- An average of 82 vulnerabilities were identified per codebase
- The most frequent languages found were: JavaScript (74%), C++ (57%), shell (54%), C (50%), Python (46%), Java (40%), TypeScript (36%), C# (36%), Perl (30%), Ruby (25%)
- The top 10 open source components found (in order of occurrences) were: jQuery, Bootstrap, Font Awesome, Lodash, jQuery UI, Underscore-stay, Inherits, isArray, Visionmedia and Minimatch

A second Synopsys survey report explores the strategies that organizations around the world are using to address open source vulnerability management as well as the growing problem of outdated or abandoned open source components in commercial code.

4.1.3. Language-specific vulnerabilities

Each programming language has unique structural flaws which might lead a developer into creating a flaw. For example, low-level languages like Assembly, C, or C++ are vulnerable to buffer overflow which hackers can exploit to write malicious code to adjacent memory once buffer capacity is full. Another common vulnerability found in languages like SQL, JavaScript, and PHP is code injection, where hackers exploit flaws in data processing that cause user input to be interpreted as system commands or include malicious script in uploaded files.

This ISO working group technical report includes a taxonomic hierarchy of weaknesses that applies to all languages. It also contains specific suggestions to avoid weaknesses that arise from constructs that are incompletely specified, exhibit undefined behavior, are implementation-dependent, or are difficult to use correctly.

Even if an issue is not present when code is deployed, cyber criminals are constantly innovating new ways to infiltrate systems and software and can discover flaws that were previously unknown or not possible to exploit.

4.2. Legacy systems CPSQ

The CPSQ due to legacy systems in 2020 is $520 billion. CPSQ in legacy systems is harder to address because such systems automate core business functions and modernization is not always straightforward. After decades of operation, they may have become less efficient, less secure, unstable, incompatible with newer technologies and systems, and more difficult to support due to loss of knowledge and/or increased complexity or loss of vendor support. In many cases, they represent a single point of failure risk to the business.

Faced with a legacy system challenge, the best approach depends on the priority of problems to be solved – functionality, performance, obsolete technology, inflexible architecture. Several strategies are available to improve quality and longevity and reduce the total cost of ownership going forward. These are:

1. Encapsulate - to hide/isolate details (create APIs and containerization)
2. Rehost – to move systems off the mainframe and migrate them to new hardware or the cloud
3. Replatform - to speed up with new hardware
4. Repair – to fix the bugs, maintain
5. Refactor - to reduce technical debt and future failures
6. Rearchitect - to adapt to a new platform or technology
7. Rebuild – to fine tune it
8. Replace – with new or SaaS solution – the most difficult choice

The common theme underlying all these strategies is often overcoming the lack of understanding and knowledge of how the system works internally. Therefore, any tool which helps the system engineer identify weaknesses, vulnerabilities, failure symptoms, defects and improvement targets is going to be useful in reducing the CPSQ. Benchmarking the health status of the system would be a useful starting point. Getting detailed blueprints of system connectivity can be especially useful for modernizing architectures that have degraded over time.

4.3. Cost of Unsuccessful IT/Software Projects

The CPSQ due to unsuccessful projects in the US in 2020 is $260 billion. The Standish Group CHAOS reports show a steady 19% of projects will be cancelled before they ever get completed and that 47% are challenged in execution. Their new definition incorporates three factors of success, namely, having in place: a good sponsor, a good team, and a good place. In other words, they are saying that most projects fail due to non-technical reasons. Many approaches to solving this view of the problem have been tried over the years, and still only one third of software projects are fully successful. And that is because this view fails to recognize the inherent complexity and difficulty of engineering good software.

Jones and Bonsignour (page 490) have provided us with interesting data on IT project cancellation rates as a function of quality. For projects of large size (10^4 FPs) and above, low-quality projects are 5-6X more likely to be cancelled then high-quality projects.

Table 2: Probability of Project Cancellation by Size and Quality Level

Function Points	High Quality	Low Quality	X-factor
100	.02	.07	3.3
1000	.05	.16	3.2
10^4	.07	.45	6.4
10^5	.12	.65	5.4

Focusing on high quality is a good way to avoid project cancellations and will help challenged projects that routinely produce low quality deliverables that cause timelines to stretch and costs to go up.

It is amazing how many IT projects just assume that "quality happens." The best way to focus a project on quality is to properly define what quality means for that specific project and then focus on achieving measurable results against stated quality objectives. Standards like ISO/IEC 25010 and OMG Automated Source Code Quality Measures from CISQ provide a model to guide that process.

From a more practical point of view, this means employing such practices as:

- prioritizing needs and requirements
- controlling scope changes
- minimizing complexity
- using systems engineering discipline
- assigning quality champions on the team
- planning for bug fixing and refactoring
- establishing rigorous quality gates
- testing early and often
- rigorous analysis of included components
- investing in quality engineering tools
- as well as all the softer skills of good teamwork and effective communication

If we can reduce the number of failed and challenged projects, the CPSQ in this area will drop quickly.

5. CONCLUSIONS

The CPSQ is a trillion-dollar problem in today's digital world. Good software quality has many benefits. Not only does it facilitate innovation by increasing predictability, lowering risk, and reducing rework, it serves as a differentiator to set an organization apart from its competitors. Most importantly, continuously ensuring quality will always cost less than ignoring it. Quality is more than free, it saves.

In 2018 we concluded that the key strategy for reducing CPSQ is to find and fix problems and deficiencies as close to the source as possible, or better yet, prevent them from happening in the first place. This advice still applies in today's world of rapid change. There are many ways to achieve better software quality. Not necessarily with "big bang" corporate initiatives, but with small incremental steps – including, if necessary, adjusting the organization's attitude towards quality. Definition and measurement are good second steps.

Successful approaches to improve software quality must address three levels of an organization: individuals, teams/projects, and organizational leadership.

5.1. At the individual level

As Smokey the Bear once said "only you can prevent forest fires" - and software bugs, if you are a software engineer reading this paper!

Software development is perhaps the most intellectually challenging profession ever. The inherent properties of software are described as complexity, changeability, conformity, and invisibility. And performance at the individual level varies widely. The best software developers are 10X better than the worst, and 5X better than the average. So individuals should strive to be the best software engineering professional they can be – become superior at estimating and planning, engage in continuous learning, use the right tools for the circumstances, become excellent at architecture/design, apply engineering discipline, be a leader on your team, adopt quality standards, apply defect prevention practices, insist on the participation of those who know what is needed, make sure you have management support for doing high quality work, and do not be afraid of factual metrics. Recognize that software quality engineering should not be an afterthought or a second-class citizen in your organization.

This paper provides a better explanation of the economics of software quality and how excellent software engineering impacts the bottom line in terms that CEOs can understand (e.g., ROI, cost of ownership, cost of quality, reputation, etc.). That is not to say there is no business pressure to move fast and agile, but good quality must counterbalance pressure to quickly release marginal software. The agility of the organization is strictly limited by the ease of modifying or enhancing its most critical business applications.

5.2. At the project/team level

Imagine your software team was like the New England Patriots of the last twenty years. The best teams create dynasties after many repeated successful performances. We also know from previous studies that the best teams are 5X better than the worst teams, and ~3X better than the average team. This kind of repeated success does not happen by accident – consistency is created by great teamwork, great role players, great leadership, and organizational culture.

To achieve repeated success, the team must know what success is. In football, it's easy, because we keep score and we know what a touchdown is. The same is true for software development. We must have a concrete understanding of what success means to achieve it. On our software projects, we can easily determine time and cost (e.g., effort). For quality, it needs to be defined upfront. We all know that it is difficult and situation-dependent to define quality, but it must be done for shared team understanding. Make quality a defined objective with criteria. Higher quality software and shorter development cycles are possible when quality is the entire team's responsibility.

There are some simple well-known techniques for accomplishing that:

1. Developers should follow established coding guidelines. This ensures the code is written in a consistent style, enhances its readability, facilitates its maintenance, and leads to fewer defects.
2. Check that your requirements, use cases, user scenarios and/or user stories are SMART—specific, measurable, acceptable, realistic, and time-bound—and make necessary updates prior to software and test case development. Review twice and code once.
3. Track problem and defect metrics - If you don't measure it, you can't improve it—that is, you won't know if you are successful unless success is defined and tracked.
4. Review work products, revise your process, and remember what worked and didn't - continuously improve over a series of projects.
5. Confront problems as early as possible, when they are cheap to fix.
6. Use tools that check for quality positioned early in the development lifecycle.
7. Avoid arbitrary and unrealistic schedules.

5.3. At the organization level

Producing quality products and systems makes good business sense, but what that means must be well-known in your organization. Quality results in positive ROI and reduced cost of ownership (COO). Routinely producing quality software is not by coincidence. Excellent quality enables cost effectiveness and superior performance when delivering software projects, products, systems, and/or services.

Variation in software performance is common across organizations. Back in the 1990's, IBM sponsored a benchmarking survey of 363 European software organizations which covered 15 countries (Germany and UK represent over half the sample) and over a dozen industries (banking, insurance, manufacturing, IT and distribution most heavily represented). The results are shown in the table below, exhibiting the striking difference in performance between the top 10% and the bottom 10% of organizations sampled.

Table 3: Best vs. Worst IT Performance Factors

Performance Factor	**Top 10%**	**Bottom 10%**
Productivity (Function Points/month)	25	5
Delivered quality (% defects removed)	>95%	<50%
Cost/Schedule Performance	<=10%	>40% over
Post-delivery maintenance costs (within first year)	<1% (of total development effort)	>10%

As presented in Figure 7, the huge differences between organizations appear to be caused in part by the adoption of certain quality and process best practices. The key enablers for top levels of cost, schedule **AND** quality performance are:

- A well-defined, yet adaptable development process
- Excellent estimation methods
- Project management discipline
- Excellent staff skill levels
- Quality vision
- Customer satisfaction focus
- Total Quality Management (TQM) culture
- Defect prevention

Figure 7: Best Practices vs. Project Performance

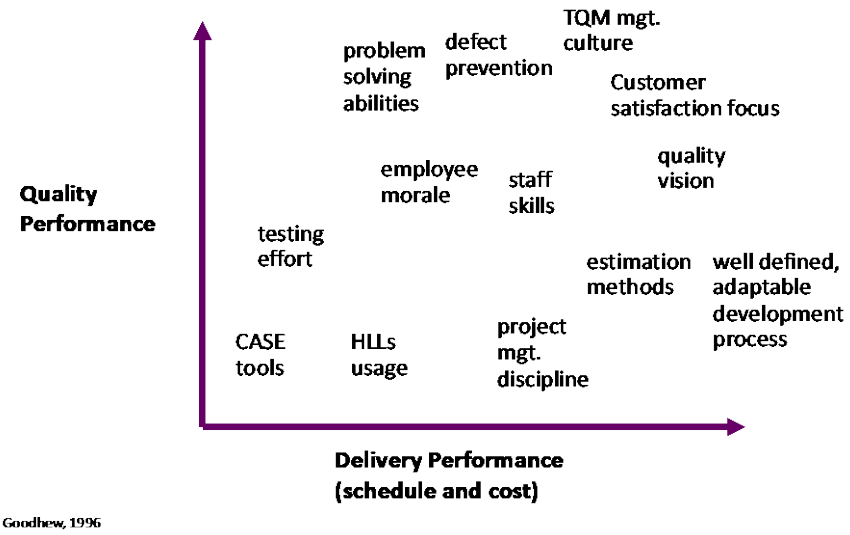


Table 3 and Figure 7 from Herb Krasner, 2018 derived from: Goodhew, 1996, *Achieving Real Improvements In Performance From SPI Initiatives*, the European SEPG Conference, June 26, 1996.

Management is responsible for providing leadership that will create, provide, and sustain a positive work environment in which all employees are continually seeking and acting on answers to the following questions:

- Who are my customers?
- What are their real needs?
- How am I (we) doing?
- What can I (we) do better?

To that basic list we now add the following internally focused questions:

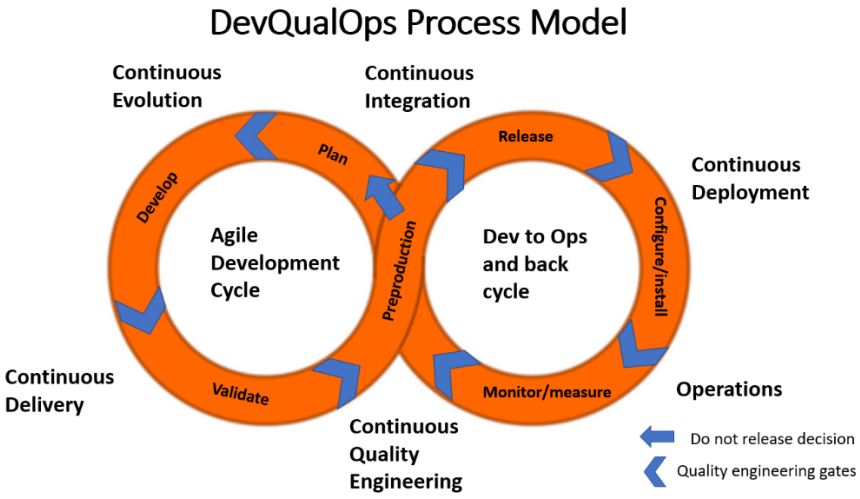
- How good is the quality of my software products/services/systems and what are the costs/benefits involved?
- How do we establish quality goals and then track them to achievement?
- What level of quality commitment should be made on each project/product/system/service?
- How do we work more effectively together and be proud of what we produce?
- How do we prevent flaws, defects, weaknesses, and vulnerabilities from being introduced into our software systems?
- What tools could be used early in the development lifecycle to check for quality?
- What do I need to learn now and in the future to remain competitive?

Otherwise, it becomes a case of "anybody, somebody, nobody" that is responsible for quality.

5.4. Pulling it all together - DevQualOps

Agile and DevOps have taken hold in many IT shops, where developers apply enhancements as small, incremental changes that are tested and committed daily, hourly, or even moment-by-moment into the production system. The result is faster and more responsive development cycles, but not necessarily better quality. As you have seen in this report, software quality is more important today than ever – it is a business imperative for success. The natural evolution of this trend is called DevQualOps, which encompasses activities that assure an appropriate level of quality across the Agile+DevOps lifecycle. Our concept of DevQualOps is seen in the figure below. DevSecOps is therefore a sub model of DevQualOps since security is a subfactor of quality in the ISO 25010 model.

Figure 8: The New DevQualOps Process Model



In this new model, we have added the following features:

- Defined quality objectives, many of which are measurable
- The traditional QA/testing role is shifted left to become a broader and more strategic team role in software quality engineering (SQE)
- Identified quality characteristic champions are on the team

- Quality level measurement and trend analysis
- Defect prevention practices as a part of continuous process improvement
- Planned refactoring to control technical debt and reduce complexity
- Bug finding and fixing planned into every iteration
- Quality gates at key points in the process
- Automated tools for every aspect of SQE – not just testing
- Static and dynamic analysis of all included components (especially open source software)

A "start, define, measure, manage" approach enables the organization to break free from "anything-goes" approaches to quality and starts the journey towards DevQualOps maturity. The ability to measure quality is the ultimate test for gauging how well quality investments are benefiting the business and its customers.

6. ACKNOWLEDGEMENTS

About the author. Herb Krasner is a retired Professor of Software Engineering at the University of Texas at Austin. He has been an industry consultant for six decades; helping organizations baseline and improve their software development capabilities. He is an active member of the CISQ Advisory Board and is well-known for his previous research in the empirical studies of software professionals, the ROI of software process improvement, and the cost of software quality. He can be reached at hkrasner@utexas.edu.

Grateful thanks to our technical review team for their most helpful review comments on the drafts of this report: Don Shafer, Eric Mizell, Greg Law, Laila Lofti, Joe Jarzombek, and Tracie Berardi. Special thanks to my long-time friend and collaborator, Don Shafer, Athens Group founder and Technical Fellow – with whom I have worked on many important initiatives since the mid-1980's. Much thanks to our project administrator, Tracie Berardi; and to my buddy Bill "Tex" Curtis, Executive Director, CISQ, for his five decades of friendship, professional collaboration, and support. Also, thanks to my wife Judy for her professional editing assistance, plus almost 50 years of love, support, and putting up with me.

About CISQ. The Consortium for Information & Software Quality™ (CISQ™) develops international standards to automate software quality measurement and to promote the development and sustainment of secure, reliable, and trustworthy software. Through their work, industry-supported standards have been developed to measure software size, structural quality, and technical debt from source code. These standards are used by many IT organizations, IT service providers, and software vendors in contracting, developing, testing, accepting, and deploying software applications.

6.1. 2020 Report Sponsors

Synopsys
Synopsys is the #1 electronic design automation company that focuses on silicon design and verification, silicon intellectual property and software security and quality. Their technology is present in self-driving cars, artificial intelligence, and internet of things consumer products. They have invested over $1 billion into building the ultimate software security solution. The Synopsys Software Integrity Group focuses on software security and quality. They provide static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix weaknesses, vulnerabilities and defects in proprietary code, open source components, and application behavior. Ranked as the leader in Gartner's Magic Quadrant for Application Security Testing, with a combination of industry-leading tools, services, and expertise, Synopsys SIG helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

Undo
Undo is the leading software failure replay platform provider. *LiveRecorder* provides engineers with full visibility and data-driven insight into what their software did before it failed - eliminating the need to reproduce issues and making bug fixing predictable. It is used by engineering teams building complex systems to resolve their software bugs faster, accelerate software delivery, and reduce engineering costs.

OverOps
OverOps was founded in 2011 with one goal in mind: to provide a better way to analyze, understand and act on issues within servers and applications, even in real-time. As engineers their mission is to help fellow developer and operation teams ship the best products they can, faster than before. OverOps Continuous Reliability

solution helps address this by analyzing code as it executes to enable organizations to detect critical code issues in real-time at any point in the software delivery pipeline and resolve them before customers are impacted.

7. APPENDIX A – 2018 CPSQ REPORT REVISITED

Our CPSQ 2018 report was used by many different organizations for purposes ranging from justifying the value of products, strategies, processes, and/or services to obtaining academic research grants to advancing the state of the art in software engineering technology. Several podcasts, blogs, and webinars referenced the report. For example, here are some of the online articles that directly referenced it:

TechHQ: AI could be the answer to a software developer shortage
DZone: Why Software Testing is Necessary for Delivering Superior Customer Experiences
DZone: Will Our Software Bankrupt Us?
Software Testing News: What is the Real Impact of Software bugs?
OverOps: Calculating the Cost of Software Quality in Your Organization
Emtec: 5 Reasons Why Continuous Testing is Important
BitBar: How Much Did Poor Quality Software Cost in 2018?
EnjoyHQ: The Cost of Product Failures
Reseller Club: 7 Software Testing Strategies Every Tester Should Know About
Information Age: Software quality issues: Not just for Boeing CIOs
CloudQA: 5 Software Testing Strategies to Uplift Business Growth
Modus: We Need More QA, Not Less
Soft Tech Group: Why Software Quality Control is Important?
Team International: 7 Tips to Achieve Strategic QA and Testing Operations
International Journal of Scientific and Engineering Research: Software Engineering Process Model: A cost and quality enhancive technique
iLab: When to Accept a Defect
QCon: How to Convince Your Manager to Address Technical Debt

More recent references are found, for example, at: Counsulting.us and OverOps.

The 2018 report laid basic concepts and definitions which we refer to in the 2020 update. In that sense, this report is not standalone. These definitions and concepts are:

- Common abbreviations used: IT, US, LOC, CoSQ, CPSQ (pg. 3)
- What is software (pg. 6)
- How much was being spent on IT/software at that time (pg. 7)
- The iceberg model of hidden software quality costs (pg. 10)
- What are legacy software maintenance costs (pg. 12)
- Summary of the major software failure stories in the news (pg. 16)
- What is software technical debt (pg. 19)
- The impact of available talent on software quality and its costs (pg. 21)
- What is software quality (pg. 28)
- The definition of the cost of software quality model (pg. 30)
- Our conclusions about the total CPSQ in analyzed categories (pg. 36)

The following section includes a brief summary of the CPSQ findings, conclusions, and recommendations from 2018. These have been updated slightly to reflect new thoughts since then. The bottom line, for those of you wishing to skip the detailed explanation of this section is: **the revised total for US CPSQ in 2018 would be**

approximately $2.1 trillion. The Conclusions and Recommendations summary subsection contains information that we still consider valid in 2020.

In 2018, we approached the task of creating a first-order estimate of CPSQ in the US by examining known information in several reported categories that we identified from a broad search of references. These categories were: 1) legacy system problems, 2) losses from software failures, 3) troubled/cancelled projects, 4) finding and fixing defects, and 5) software technical debt.

7.1. Looking at legacy systems

In most organizations, the sustainment or lifecycle support of existing IT systems consumes most of the IT budget, roughly 75% of total IT spend per year. Legacy systems can become unwieldy due to aging, varying by the type of system.

Determining the cost of poor-quality software in legacy systems requires an analysis of the activities that consume the most effort during the sustainment phase of an IT systems lifetime. Pigoski explained the classic view of software maintenance in which costs fall into the following basic categories:

- Corrective maintenance –modifying software to correct issues discovered after initial deployment (generally 20%)
- Adaptive maintenance –modifying a software solution to allow it to remain effective in a changing business environment (generally up to 50%)
- Perfective maintenance –improving or enhancing a software solution to improve overall performance or maintainability (generally 25%)
- Preventive maintenance –modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults (generally 5%)

Agile development and DevOps teams rarely use the term 'maintenance' for software updates. Operation and Maintenance (O&M) has become an outdated term used for the lifecycle support phase of software that has released for operation; more often it is in 'sustainment' because the software continues to be changed. However, for consistency with the previous report's analysis, the term 'O&M' is still used here.

According to Curtis, correcting defects frequently accounts for as much as one third of all post-release work, and time spent understanding the code has been shown to account for as much as half of all the effort expended by maintenance staff. According to Jones, about 60% of US software engineering work time centers on finding and fixing errors, which could be avoided with greater development discipline. When the overlap between these two activities is removed, as much as two thirds of all maintenance effort can be classified as waste.

Using the factors above allowed us to estimate the overall cost of poor-quality legacy software in O&M in the US in 2018. If ~75% of all IT dollars are spent on O&M, and if as much as two thirds of that could be classified as "waste," that gives us an approximate upper bound of $980 billion on the cost of poor-quality software in O&M. Waste is a lean term that means all non-value-added activities. This waste does not include additional costs incurred outside of the IT organization. The lower bound using only corrective maintenance in the calculation would be $290 billion. The mid-point between the upper and lower bound would be $635 billion, which we took as our first order approximation of the CPSQ due to legacy systems in 2018.

7.2. Software system failures in operation

The table on pages 17-20 of this report lists top IT failures in the news for 2018-2020, representing just the tip of the CPSQ iceberg. In a 2018 report, software testing company, Tricentis, analyzed 606 software fails from 314 companies to understand the business and financial impact of software failures. Their report revealed that software failures affected 3.6 billion people and caused $1.7 trillion in financial losses and a cumulative total of 268 years of downtime. Software defects were the most common reason behind these failures. Their failure events came from English language news outlets. The report did not allocate failures to specific countries, but in terms of relative GDP the US dominates the group of English language speakers (75% of the total). We therefore assume that 75% of these failure totals are attributable to the US. So, we estimated the US total for software failures was $1.275 trillion for 2018. What about all those software failure stories that don't make the news?

In the area of software management failures, a study by IBM and Ponemon Institute looked at compromises to business continuity or security due to IT risks. They estimated the cost of business disruptions to be:

Table 4: Cost of Business Disruptions Due to Software Management Failures

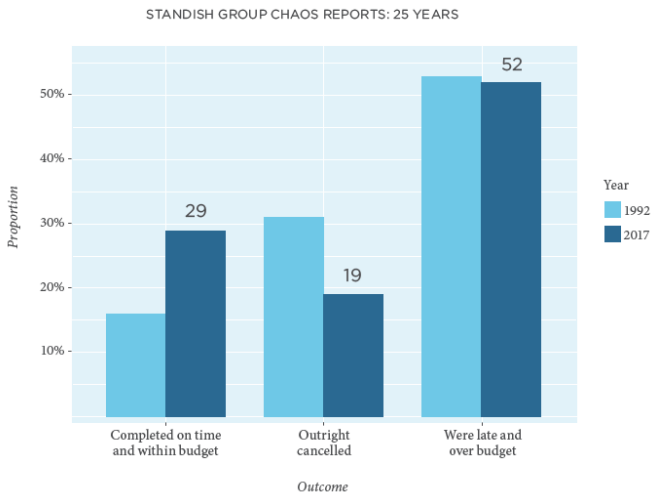
Type of event	Length of disruption (minutes)	Cost (per minute)	Likelihood of event over next 2 years
Minor	20 min.	$53,210	69%
Moderate	112 min.	$38,065	37%
Substantial	442 min.	$32,229	23%

The most telling statistic was that 75% of event costs go to reputational damage and the bottom line. Since we have no simple way of directly converting this into a cost of poor-quality software, we left it out of our totals.

7.3. Troubled/Challenged Projects

Based on reviewing 25 years of historical software projects, The Standish Group reported that only 29% of projects successfully met scheduled delivery and budget. Their data says nothing about the quality of the result. Presumably, those projects had successful outcomes. The Standish Group research shows a staggering 19% of projects are cancelled before they ever get completed. Further results indicate 52% of projects will cost 189% of their original estimates. The number and percentage of challenged projects (over budget, behind schedule, or low-quality deliverables) has barely changed over 25 years. The cost of cancellations and overruns are usually hidden below the tip of our CPSQ iceberg. There are also thousands of troubled projects that rarely make the news.

Figure 9: Software Project Outcome Rates: 2018 Report



Looking at total IT spend for labor, with the assumption that 25% applies to development projects, gives us an estimate of the total dollar amount impacted. Using the percentages above to gauge the cost impact - assuming that all projects are equally funded (on average) - provides a dollar amount. The US software labor base in 2018 was about $1 trillion, with $250 billion in development projects. Therefore, we estimated that $130 billion was lost in troubled projects, and $47.5 billion lost in cancelled projects in 2018.

7.4. Finding and fixing defects

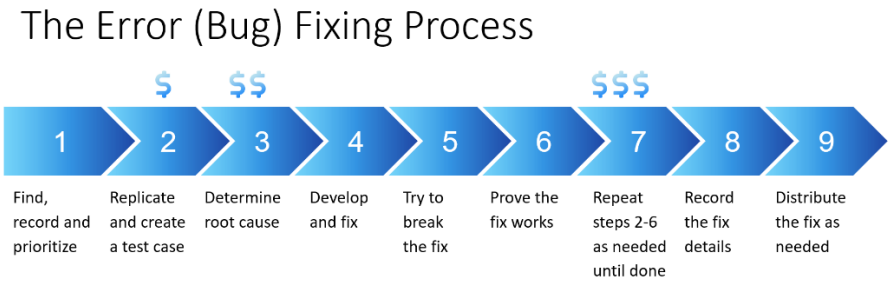
The cost of finding and fixing bugs or defects is the largest single expense element in the software lifecycle. Over a 25-year life expectancy of a large software system, almost fifty cents out of every dollar will go to finding and fixing bugs. Large systems have much higher defect potentials that are more difficult to remove than for small systems due to size and complexity. The earlier in the development lifecycle defects are found, the more economical the overall delivery will be. A good resource is the book, The Economics of Software Quality.

To understand the costs involved here, we must look at the processes involved. First, since there is an order of magnitude cost difference between internal and external defects, we needed to understand those categories.

- Internal Defect Costs - Costs associated with software deficiencies discovered before the system leaves the development organization and is deployed into the operational environment. These deficiencies occur when a system fails to meet a certain requirement, resulting in waste, scrap and/or rework. The deficiencies could be in the work products of development, the development process, and/or components if they fail to meet quality standards and requirements. Unfortunately, very few organizations track this category prior to the commencement of system testing.
- External Defect Costs - Costs occurring when the failure of software to reach quality standards is not detected until after it is transferred into operation or to the customer. External failure/deficiency costs are incurred during customer use. The largest category of cost is professional effort to replicate, find, and fix all of the fielded defects and re-appraisals to verify fixes.

Second, it is instructive to look at the engineering process of finding and fixing defects so that we may observe where the effort is expended. A simple model (Krasner, 2018) is shown in the figure below. The dollar signs indicate where most of the cost is concentrated.

Figure 10: The Process of Finding and Fixing Software Defects



In the 2018 report section on "Computer and IT occupations in the US today" (pgs. 23-25), we determined that the total US IT professional wage base in 2018 was ~$1 trillion. Asking all of those IT professionals to write down what activities they spend their time on provides a surprising answer. Only a few empirical studies have studied time allocation by developers. The common assumption is that developer activities must be in line with the project plan work breakdown structure or the project's defined software development lifecycle process. Most cost tracking systems for software development projects omit as much as 50% of the total effort. Given this wage base, we conservatively estimated that $500 billion was being spent finding and fixing defects. In 2018, published empirical evidence was lacking in this area (and still is).

7.5. Technical Debt

We tried to approach technical debt in two ways. First, assuming that this debt is generally true of all software applications and systems, and, second, estimating the lines of code (LOC) that existed in the US helped identify the contribution of technical debt to the total CPSQ. Grady Booch estimated in 2005 that about 35 billion LOC are written globally each year. This is based on about 15 million or so software professionals worldwide, of which about 30% actually cut code. Each developer contributes about 7,000 LOC per year. Based on this estimate, there would have been about a trillion lines of code written up to 2005. If we conservatively assume that the worldwide code growth rate is steady at 35 billion new LOC per year, then in 2018 there would be 1.455 trillion LOC worldwide. And assuming that there is $4 of technical debt per LOC, then the debt in 2018 would be $5.82 trillion. If the US share of that debt is roughly 31% (according to the section on IT spending), the US technical debt figure would be $1.8 trillion.

Second, in 2011, CAST and CISQ estimated that the global IT technical debt was $500 billion and would rise to $1 trillion by 2015. If the doubling period is in fact four years, then the debt in 2018 would be $1.75 trillion. The US share of that would be $.54 trillion – a lower figure than our previous calculations would indicate.

If we split the difference between these two approaches, the amount of software technical debt in the US in 2018 was approximately $ 1.145 trillion. That represents just the debt principal without accumulated interest.

7.6. CPSQ in Summary

To summarize the estimates in five categories in the 2018 report, we estimated that the CPSQ included:

1. $635 billion due to legacy systems issues
2. $1.275 trillion due to software failures
3. $130 billion due to troubled projects, and $47.5 billion due to cancelled projects
4. $500 billion due to finding and fixing defects.
5. $1.145 trillion in technical debt

In 2018 we made the simplifying (knowingly incorrect) assumption that the above categories were mutually exclusive. A straightforward addition, therefore, gave us a rough order for the total CPSQ covering the landscape in 2018 to be about $3.73 trillion.

At that time, there was no data on how the above categories intersected, implying the need for deeper research and analysis. We also knew that IT and software organizations do not collect cost or effort data in these ways.

In retrospect, had we attempted to tease out the likely relationships between the five categories analyzed, we could have come up with a more conservative estimate. For example, if we make the following intersection analysis assumptions:

- The impact of internal defect finding and fixing is also felt in cancelled and delayed projects
- The impact of external defect finding and fixing is also felt (perhaps) in the software failures category, and also in the legacy systems category (~75%)
- Since technical debt is actually a future cost to be addressed, it should not show up in this summary at all, except perhaps for the actual cost of refactoring not reflected in defects or failures.

There may also be some overlap between legacy system CPSQ and the costs of software failures to the extent that some failures are caused by a few legacy system maintenance “fixes” introducing new catastrophic defects. But the “bad fix” rate is relatively small (2-4%) compared to all maintenance defect fixes, and most of these do not introduce severe defects. The older a legacy system is, the more likely that all the severe defects have been removed. Therefore, we shall not consider this small overlap in our 2018 estimate of CPSQ.

With those assumptions in place, our revised total for US CPSQ for 2018 was $2.1 trillion. Still a huge number.

7.7. Conclusions and recommendations summary - 2018

The 2018 report was an aggregation of publicly available source material pertinent to the first order approximation of CPSQ in the US and a discussion of how that estimate could be leveraged to stimulate software quality improvement programs across industry and government.

We knew that the raw data we needed was not there. Without a defined model of Cost of Software Quality (CoSQ), most IT leaders do not have a basis for estimating the answers to questions. E.g.,

- How much are you spending today on the cost of poor-quality software in your organization?
- How are your investments in good software quality affecting your overall costs of quality and cost of ownership for software assets?

Therefore, we defined such a model for CoSQ measurement in the report (pages 30-35).

We observed the term "quality costs" means different things to different organizations. In any case, the difference can be significant – e.g., nearly 20 to 40 percent of a company's sales, according to Juran's Quality Control Handbook. In software, the difference between poor-quality and good-quality is felt differently based on software usage, system types, and sizes.

Key to reducing the cost of poor software quality is to find and fix problems and deficiencies as close to the source as possible. This strategy implies that wise investments in software engineering discipline and good software quality will dramatically reduce the cost of poor-quality. The concepts of “shifting-left” of quality, early and continuous testing, and advanced defect detection and resolution capabilities are gaining traction in the industry for exactly this reason. There are many ways to achieve better software quality, but all of them start with a well-conceived measurement program, fully encouraged by top management.

8. APPENDIX B - IT SPENDING TRENDS

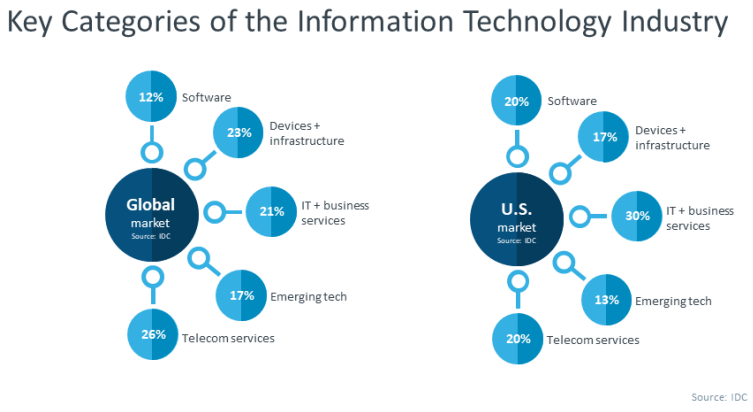
8.1. IT market trends in 2020 and beyond

In 2020, the global information technology industry took a small step back in terms of overall revenue. As of August 2020, the research consultancy IDC was projecting global revenue of $4.8 trillion for the year, compared to their original estimate of $5.2 trillion. While the tech sector fared better than many other industries during the pandemic, it was not immune to cutbacks in spending and deferment of major investments. Moving forward, IDC projects that the technology industry is on pace to reach $5 trillion in 2021. If this number holds, it will represent 4.2% growth, signaling a return to the trend line the industry was on prior to the pandemic. Looking further into the future, IDC expects the pattern to continue, estimating a 5% compound annual growth rate (CAGR) for the industry through 2024.

The US is the largest tech market in the world, representing 33% of the total, or approximately $1.6 trillion for 2021. Despite the size of the US market, the majority of technology spending (67%) occurs beyond its borders.

Using the conventional approach, the IT market can be categorized into five top level buckets: 1) software, 2) devices and infrastructure, 3) IT and business services, 4) emerging technology and 5) telecom services. The traditional categories of hardware, software and services account for 56% of the global total. The other core category, telecom services (that rely on software), accounts for 26%. The remaining 17% covers various emerging technologies that either don't fit into one of the traditional buckets or span multiple categories, which is the case for many emerging 'as-a-service' solutions that include elements of hardware, software, and service, such as Internet of Things (IoT), drones, and many automating technologies.

Figure 11: Categories of IT Spending



While emerging technologies currently account for only 17% of the overall global revenue, they are expected to drive nearly half of the growth in new revenue in the coming years. Software development is the area where most companies expect to place focus in the upcoming years, but there is also strong demand for cybersecurity, data, and infrastructure.

8.2. IT Spending Trends in US and Canada

The *Computer Economics IT Spending and Staffing Benchmarks* 2020/2021 study was a survey of 233 IT organizations in the US and Canada conducted in the first half of 2020. Respondents had at least $50 million in annual revenue or IT spending in excess of $1 million and maintained at least some operations in the US or Canada. The size of sampled companies was broken down based on IT operational budget into small (<$5M), medium ($5-20M), and large ($20M+) at equal percentages. Companies represented the following business sectors in order of number participating: manufacturing, government/non-profit, financial services, professional/technical services, healthcare, retail/wholesale, construction/trade services, utilities, and other. For reference, for the 2021 fiscal year, the U.S. federal government has budgeted $92.17 billion for IT spending.

Their outlook for the coming year was looking strong for IT organizations—until the global pandemic struck. Based on follow-ups, at the time of report publication, the effect to IT organizations was mixed. Some companies were dialing back new IT projects. Some companies were accelerating projects for unified communications, remote work capabilities, business intelligence, and analytics to weather the storm. It seemed unlikely that even a deep global recession would stop the cloud migration.

Some companies were impacted negatively by the pandemic, especially entertainment, tourism, hospitality, and travel. Other companies, such as online retail, logistics, some types of manufacturing, and those companies born in a digital world were less affected, and a few were even thriving. Regardless, some of their key findings were interesting, including:

1. As a percentage of revenue, IT spending remained stable at ~2.6%
2. IT operational spending growth was still increasing at a modest 3.0% rate (median), as it had over the past 4 years, and professional/technical services sector led the way at 5.0%
3. IT spending per user declined to $7,602 (by 2%) due to cloud, virtualization, and IT automation efficiencies, while user counts rose at many companies
4. The top spending priorities reported were (in order of percentage reporting): cloud applications, cloud infrastructure, data analytics, digital transformation, systems/data integration, business continuity, legacy systems, and shared services
5. Information security/privacy and business applications are the top areas for increased spending again this year as last
6. IT capital spending has been trending down at ~18% per year for the last three years (2018-2020)

Some other useful references are found here:

- Wall Street Journal: Corporate Tech Spending Helps Lift U.S. Economy, *September 24, 2019.*
- CIODive: Software eats the world, jobs double US employment growth rate, *September 20, 2019.*
- InfoWorld: Report: Software jobs pay twice the national average, *September 20, 2019.*
- TechRepublic: Report: Software jobs expected to grow twice as fast as overall US jobs, *September 19, 2019.*
- Press Release: US Software Jobs Grow Twice as Fast as Overall US Jobs, *September 19, 2019.*
- Connecting the Dots: Software: Growing US Jobs and the GDP, *September 19, 2019.*