



BLOG

# Worst of the Worst—The Biggest Software Fails in Recent Memory

July 24, 2018 - Ankur Verma



CREATE



MAINTAIN



SCALE

View Demo

## Technology

Architect

ALP™

AEA™

Data Science

Visual Testing

Cross Browser

Mobile Testing

## Company

About

Manifesto

Partners

Blog

Resources

In the media

Contact Us



Friends don't let friends release bad software. Most teams know too well the sting of learning from their own mistakes, but it's much less painful to learn lessons from the experience of others. With that in mind, this article reviews a number of very public catastrophic software failures in recent years. No, we're not trying to depress you. Rather, these are good reminders that can incentivize all software professionals to remember the importance of testing.

The [Cap Gemini World Quality Report](#) makes it clear that the last few years have been very challenging for software companies as they grapple with new technologies. Also, there have been too many major software failures, each of which has a deleterious impact on public reputation, net profit, and customer satisfaction. The aftermath of each major failure disperses far and wide over the Internet and the evening news, and the black cloud can hover over a business for years.

Before outlining an approach to avoiding such failures, let's review a sampling of the major software failures that became bad news in recent years.

## GPS satellite failure affects many critical systems

The [catastrophic failure of a 25-year-old GPS satellite](#) this past in January 2017 activated a software bug which that only manifested itself for only a mere 13 microseconds. Though the fault was momentary, the result was an enormous impact to global positioning systems (GPS), the US Air Force, and telecom networks. For every nanosecond of

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

It's fine



several hours later.

## The Equifax mega-breach

This is one of the biggest exposures, mostly due to poor quality assurance culture and practices. The personal information that was stolen in [the Equifax database breach](#) potentially compromised the identities of 145 million people. That is more than a third of the US population. Many of these people had given no consented for Equifax to retain their personal information in their databases. Worse yet, Equifax did not apply any patches to its website for many months after major flaws in its software were disclosed publicly.

hissing and clicking noises that had to be fixed with an update. While Google is relatively new at hardware products, we have a humble suggestion for all product developers everywhere: test your products before manufacturing them in large volume.

## Blue Cross/Blue Shield system failure

The Blue Cross Blue Shield Association of North Carolina endured a major new-system failure in January 2016, the result of which was that nearly 25,000 participants were enrolled in the wrong health insurance plan. BC/BS of North Carolina was eventually made to pay [a fine of \\$3.6 million](#). The deluge of complaints made it clear that there was a system problem. The problems became much worse when it became news that an internal source made it know that BCBS management had been aware of the problem yet continued with the implementation of the system. Clearly, that was an abysmally poor decision. Software bugs and shoddy features can be very challenging for any organization. Failure to remediate and fix all serious problems is a sure-fire way to lose customers, incur penalties, and damage the company image.

## GameStop fails on the launchpad




GameStop has been struggling to stay viable as digital games continue to increase in popularity. The company launched a rental service known as PowerPass in October 2017. For a fee of \$60, a member could check out a used game from a local store one at a time over the course of six months. At the end of that period, the member could keep one of those used game titles. One month after it launched, GameStop brought an end to PowerPass. [Kotaku reported](#) that company computers had not been set up to manage the PowerPass program. Surely this is a lesson on the value of good implementation and integration planning.



the touch controls were initially configured such that the software was receiving signals for a continuous depression of those buttons. The result was that [Google Assistant was continuously recording every word that was said](#). This was a major fail because even though Google made an attempt to fix this problem by publishing an update to the firmware, it didn't solve the problem at all. Google wiped the collective egg off its face and disabled this feature entirely. This feature launch gone wrong is even worse when we realize that it was entirely preventable.

## Software malfunction drops thousands of 911 callers

In 2014, a major 7th-party notification-wide emergency call center built that directs and assigns

			<b>Technology</b>	<b>Company</b>
<a href="#">CREATE</a>	<a href="#">MAINTAIN</a>	<a href="#">SCALE</a>	Architect	About
			ALP™	Manifesto
			AEA™	Partners
			Data Science	Blog
	<a href="#">View Demo</a>		Visual Testing	Resources
			Cross Browser	In the media
			Mobile Testing	Contact Us

product changes should never put your company at risk. It's quite feasible for teams to deploy a continuous stream of changes and simultaneously ensure that each and every build functions properly.

Comprehensive testing combines conventional testing excellence with effective automation technology to achieve bug feature validation and broader test coverage—while updates and changes continue to aggregate in the pre-production build. Technology failures can spawn interminable public relations problems—as we have seen in this article. Though it can sometimes be difficult to precisely quantify cost savings that directly result from preventing defects, there is enormous value in making the effort to build bug-free applications and protecting product/brand reputation.

Ensuring that all features will be thoroughly tested while new updates continue through the product pipeline simply cannot be done effectively with manual testing. This is why automation frameworks such as Functionize are critically important for testing non-trivial applications. With the Functionize AI toolset, teams can reliably and extensively automate testing—at virtually any frequency or scale.

One way to think about improvements to your test automation toolset is to view it as insurance. Not only will your team build better quality software, many have found that implementing test automation is a hedge against customer disappointment, negative publicity, and employee frustration. This is quite important if you are an Agile DevOps development team. The rate of change in large, complex application development pipelines makes it impracticable to ensure high quality can be managed using conventional, non-automated approaches.

## Protect your brand, and delight your customers

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

[It's fine](#)



customers with best-in-class tools that will help ensure that the inevitable bugs in your software products will be found by your testers—and never be seen by your customers.

## Similar posts



[CREATE](#)



[MAINTAIN](#)



[SCALE](#)

[View Demo](#)

### Technology

[Architect](#)

[ALP™](#)

[AEA™](#)

[Data Science](#)

[Visual Testing](#)

[Cross Browser](#)

[Mobile Testing](#)

### Company

[About](#)

[Manifesto](#)

[Partners](#)

[Blog](#)

[Resources](#)

[In the media](#)

[Contact Us](#)



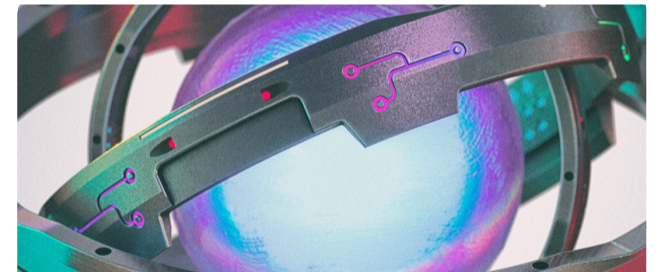
**[What's the difference between a good QA director and a great one?](#)**

[Management](#) [Software quality](#)



**[A handy software defect tracking checklist](#)**

[Software quality](#) [Test Creation](#) [Testing](#)



**[What is hyper-automation?](#)**

[Digital Transformation](#) [General](#) [Management](#)

## Categories

[Agile QA](#) (42)

[AI Visual Testing](#) (4)

[Artificial intelligence](#) (26)

[Autonomous Testing](#) (17)

[Canary Testing](#) (5)

[Career](#) (9)

[case study](#) (2)

[CI/CD](#) (13)

[Cloud Testing](#) (13)

[Continuous Testing](#) (7)

[Cross Browser Testing](#) (2)

[DevOps](#) (16)

[Digital Transformation](#) (6)

[Functionize](#) (31)

[General](#) (34)

[Load Testing](#) (4)

[Management](#) (4)

[Mobile Testing](#) (5)

[Performance Testing](#) (7)

[PR](#) (17)

[Product Releases](#) (4)

[Selenium](#) (9)

[Software quality](#) (7)

[Test Creation](#) (18)

[Test maintenance](#) (8)

[Testing](#) (13)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

[It's fine](#)



in a cloud-based solution.

First name\*

Last name\*

Email\*

Phone number\*

Company name\*

I'm interested in learning more about\*

- AI Cloud Testing
- ALP test creation
- Autonomous testing
- Cross Browser Testing
- Functional Testing
- Load and Performance Testing
- Model-based testing
- Self-healing maintenance
- Service Virtualization
- Test cloud orchestration

Submit

1-800-826-5051  
 156 2nd Street  
 San Francisco, CA 94105

Get Started

**SOLUTION**

- Test Creation
- Test Maintenance
- Test Cloud
- Demo

**TECHNOLOGY**

- Architect
- Adaptive Language Processing
- Adaptive Event Analysis
- Data Science
- Performance & Load
- Visual Testing
- Cross Browser Testing
- Mobile Testing

**RESOURCES**

- Blog
- Webinars
- White Papers
- Case Studies
- Videos
- Backgrounder

**COMPANY**

- About
- Partners
- Contact Us
- Manifesto
- In the media



© 2020 Functionize, Inc. All rights reserved.

[Privacy Policy](#) | [Terms of Service](#) | [Support Policies](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

It's fine