Piacente Cristian 866020

Assignment A1.part1 – Impact of software failures

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

Summarise the main lessons that you learn from the Tricentis report: Software failure caused $1.7 trillion in financial losses in 2017.

Software testing has not experienced the same level of change with respect to **software development**: we're in a new era that requires us to rethink software testing strategies. Nowadays, software is so convenient that when there are software **failures** and **outages**, everyone notices, and our lives are disrupted. The modernization of software testing has lagged behind software development and deployment, and it's time for a change.

Software Fail Watch provides a "reality check": the number of news stories related to bugs continues to grow, and we can change it by approaching automation with the correct strategies.

Considering the year 2017, here are some numbers: 606 software fails, 1177 stories, 314 companies, 3,683,212,665 people affected, **$1,715,430,778,504 USD in losses**, and more than 268 years and 8 months of accumulated time lost.

In the previous years, Public Services and Healthcare got the most news coverage, while in this year (only 2017 was analyzed in this report), **Retail and Consumer Tech** has taken the lead.

While the overall total number of software failure stories has grown, Public Services and Healthcare got the biggest drop in media coverage.

If we have a look at the software fails by type, enterprise apps (software that requires installation in a specific location) has the lead, then the 2nd place belongs to mobile/cloud software, and finally embedded.

We can also consider the type of software fail: software bug (331) vs usability glitch (54) vs security vulnerability (136).

There's another important factor: taking into account the stock market, while it's not always possible to trace the effects of software fails, we can notice when the correlation between it and the company's stock prices is unmistakable (e.g., the crash of a new appointment scheduling application of a UK-based loan company).

In general, bugs surface during and shortly after implementation, but decision-makers are pressured to move ahead, and then the bugs eventually come to light when it affects the public.

One of the biggest stories in the Entertainment category is about the **arrest of a Palestinian man** when Facebook incorrectly translated his "good morning".

Speaking about the Transportation industry, the airlines are affected the most from outages: **British Airways** took one of the biggest hits with 6 outages within 2017. The biggest train bug took place in Singapore, where the software safety mechanism failed and caused **two metro trains to crash** into each other.

About the Tech industry, Apple's autocorrect bugs in 2017 were important because the global estimate of messages sent per year is 8.6 trillion. This is an example of demonstrating that **a simple glitch can become one of the most talked-about bugs**.

Software failures in finance rarely hit the headlines because of two reasons: one, major financial institutions often have powerful crisis control teams, and two, much of a financial institution's software lies behind the scenes.

If we consider **brand damage and erosion** caused by media coverage, the industry that came out on top is Consumer Tech. A factor is the speed at which companies are forced to innovate and release products: this can result in the quick release of **poorly tested software**.

The scariest fail is about the failure of automatic doors that trapped six people who **died** in an office fire.
One of the most shocking fails is about 709,000 pieces of medical mail undelivered (1,700 cases were identified where the missing information could have caused serious harm) because of a software glitch.

In conclusion, Software Fail Watch focuses on which industries had the most bugs and failures, but just because a bug doesn't make it to the front page of newspapers doesn't mean it never existed.
We can notice the **butterfly effect** of brand erosion as news coverage ramps up.

The author hopes that by publishing Software Fail Watch, we can realize how **impactful** software failures can be. The author used Google alerts and categorized all the stories into 9 categories: Public Services, Healthcare, Retail, Consumer Tech, Finance, Utilities, Services, Transportation, and Entertainment.