Functional Requirements
vs.
Non-Functional Requirements

Functional Requirements

- We've mostly discussed functional requirements so far
  - Describes a function of the system
  - Potentially includes its inputs, behaviors and output
  - Feature
- Familiar Examples
  - As a potential user, I need to login in order to access my account data
  - As a new user, I need to create a new account that will be associated with my credentials

Non-Functional Requirements

- Constraints on the product's services or functions
- Often apply to the system as a whole, not to a specific feature
  - Reliability
  - Recovery (from failure)
  - Availability
  - Usability
  - Performance
  - Regulatory
  - Supportability (bug fixing)

Non-Functional Requirements Not Important?!

- How much $$$ does Amazon lose for every second their services are not running?

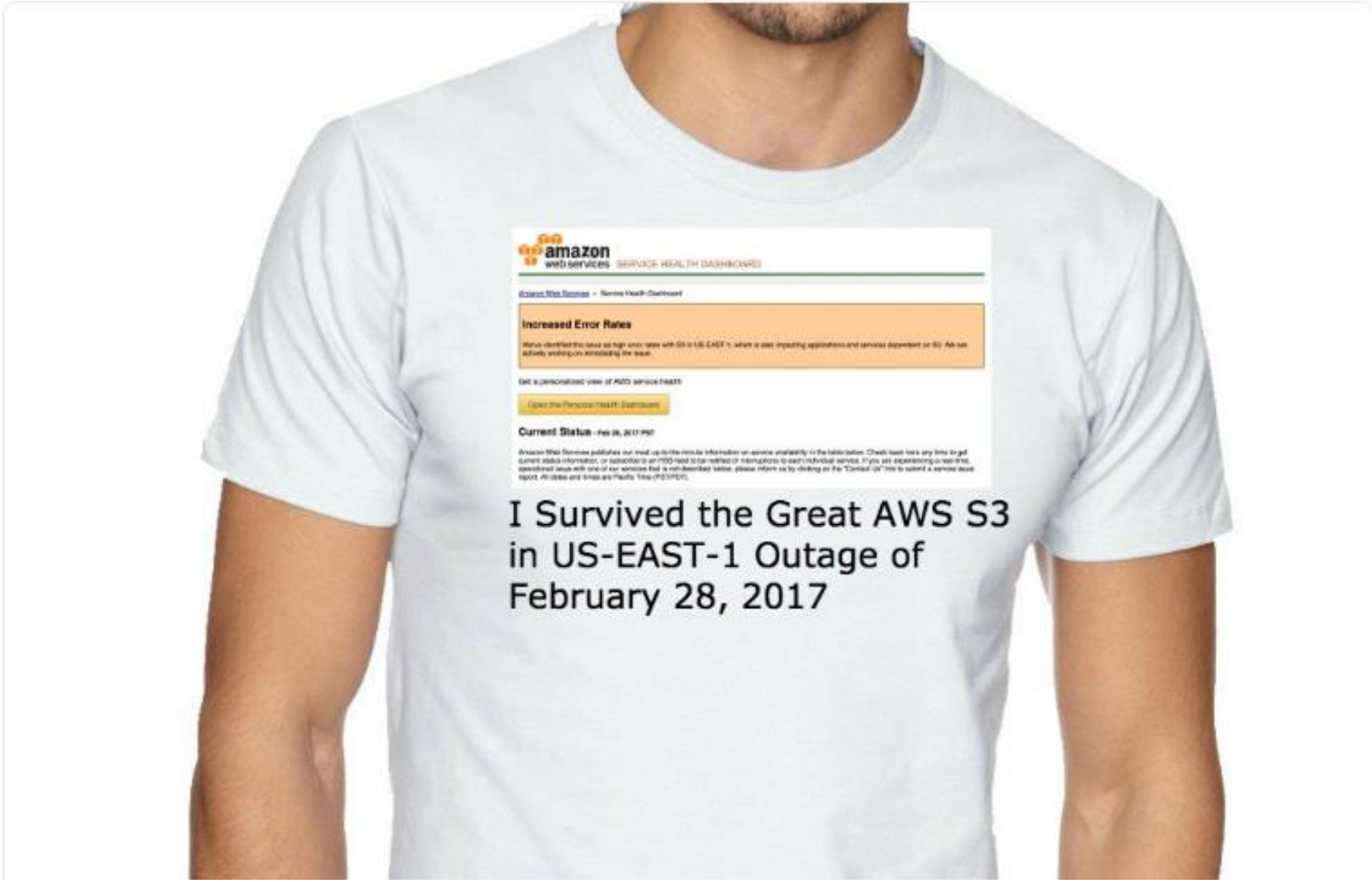
Non-Functional Requirements Not Important?!

- Amazon's revenue is ≈$177,860,000,000/year
- That's >$5,600/second
- While Amazon is an extreme example, reliability, supportability and availability are important to all eBusiness

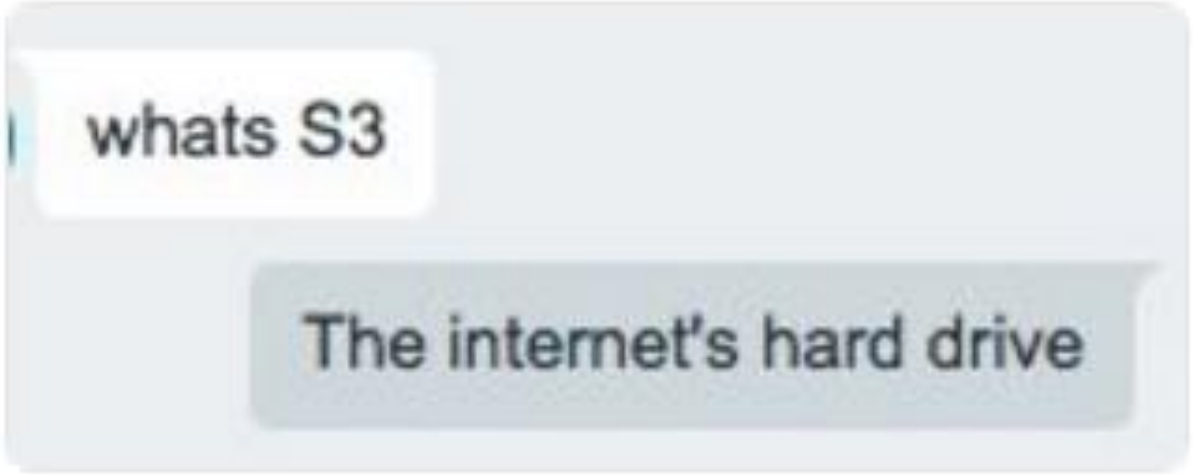
February 28, 2017

Amazon AWS support team offices right now. #awsoutage

I Survived the Great ASW S3 in US-EAST-1 Outage of February 28, 2017 - #awsoutage #awscloud #awsdown



When **Amazon S3** is down. #awscloud #awss3



S3 size

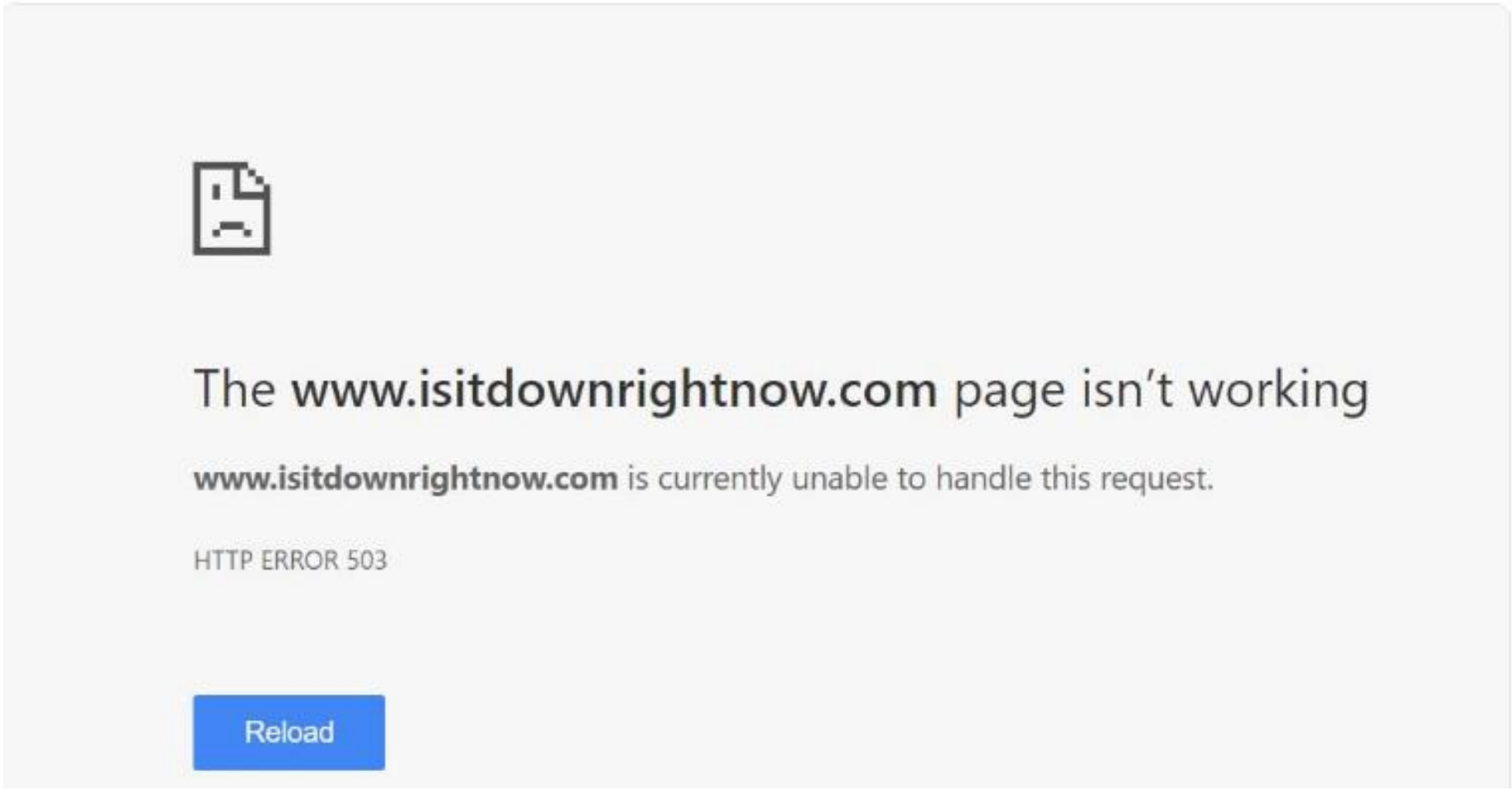
- "The S3 system is used by 148,213 sites according to market research firm SimilarTech"
  - https://www.similartech.com/technologies/amazon-s3

- "It has "north of three to four trillion pieces of data stored in it," Bartoletti said."

https://www.usatoday.com/story/tech/news/2017/02/28/amazons-cloud-service-goes-down-sites-scramble/98530914/

Amazon S3 according to the #AWS status page.

Irony, defined. #awsoutage



AWS Outage – What caused it?

"[…] The Amazon Simple Storage Service (S3) team was debugging an issue causing the S3 billing system to progress more slowly than expected. At 9:37AM PST, an authorized S3 team member using an established playbook executed a command which was intended to remove a small number of servers for one of the S3 subsystems that is used by the S3 billing process. Unfortunately, one of the inputs to the command was entered incorrectly and a larger set of servers was removed than intended. The servers that were inadvertently removed supported two other S3 subsystems. One of these subsystems, the index subsystem, manages the metadata and location information of all S3 objects in the region. This subsystem is necessary to serve all GET, LIST, PUT, and DELETE requests. The second subsystem, the placement subsystem, manages allocation of new storage and requires the index subsystem to be functioning properly to correctly operate. The placement subsystem is used during PUT requests to allocate storage for new objects. Removing a significant portion of the capacity caused each of these systems to require a full restart. While these subsystems were being restarted, S3 was unable to service requests. Other AWS services in the US-EAST-1 Region that rely on S3 for storage, including the S3 console, Amazon Elastic Compute Cloud (EC2) new instance launches, Amazon Elastic Block Store (EBS) volumes (when data was needed from a S3 snapshot), and AWS Lambda were also impacted while the S3 APIs were unavailable. " - https://aws.amazon.com/message/41926/

Non-Functional Requirements

- Constraints on the product's services or functions
- Often apply to the system as a whole, not to a specific feature
  - Reliability
  - Recovery (from failure)
  - Availability
  - Usability
  - Performance
  - Regulatory
  - Supportability (bug fixing)

Usability Example

- The average first-time user will successfully create a new account in less than 60 seconds

- The user should be able to go from X to Y in less than 3 clicks

Reliability Example

- The Mean Time To Failure (MTTF) for the server will exceed 720 hours (~1 month)

Recovery Example

- The Mean Time To Recover (MTTR) the server following a failure will be less than 1 hour

Availability Example

- The server will be available more than 99.86% of the time
  - Note: Is preventive maintenance included?

- How to compute the 99.86%?

Availability Example

- The server will be available more than 99.86% of the time
  - Note: Is preventive maintenance included?
- A_I = Inherent Availability (excludes preventive maintenance)

$$A_I =$$

Availability Example

- The server will be available more than 99.86% of the time
  - Note: Is preventive maintenance included?
- A_I = Inherent Availability (excludes preventive maintenance)
- MTTF = Mean Time To Failure
- MTTR = Mean Time To Recover
- MTBF = Mean Time Between Failures (MTTF+MTTR)

$$A_I = \frac{MTTF}{MTTF + MTTR}$$

Availability Example

- The server will be available more than 99.86% of the time
  - Note: Is preventive maintenance included?
- A_I = Inherent Availability (excludes preventive maintenance)
- MTTF = Mean Time To Failure
- MTTR = Mean Time To Recover
- MTBF = Mean Time Between Failures (MTTF+MTTR)

$$A_I = \frac{MTTF}{MTTF + MTTR} = \frac{720}{720 + 1} = 99.86\%$$

Amazon AWS Outage – February 28, 2017

- On the day of the outage, it was reported that AWS up-time was ~99.59% (~178 minutes/month downtime)

- which is under their 99.95% (~21 minutes/month downtime) target

https://www.forbes.com/sites/patrickmoorhead/2017/03/01/with-its-latest-outage-amazon-web-services-is-helping-to-sell-hybrid-it/#18cfa05b5c0d

Amazon AWS Service Commitment

"AWS will use commercially reasonable efforts to make Amazon EC2 and Amazon EBS each available with a Monthly Uptime Percentage (defined below) of at least 99.95%, in each case during any monthly billing cycle (the "Service Commitment"). In the event Amazon EC2 or Amazon EBS does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below."

https://aws.amazon.com/ec2/sla/

Performance Examples

- We are considering both timing and load here (some authors separate them)

- The server will provide an average response in 1 second

- The server will support 100 users
  - Requirements may have interdependencies
  - And still provide a 1 second average response?
  - And still provide 99.86% availability? :)

Regulatory Examples

- The server will comply with all **H**ealth **I**nsurance **P**ortability and **A**ccountability **A**ct (HIPAA) requirements
- **F**amily **E**ducational **R**ights and **P**rivacy **A**ct (FERPA)
- The server will be ISO 9001:2015 certified
- The system will comply with DO-178B, Software Considerations in Airborne Systems and Equipment Certification
- MIL-STD
- IEEE 754 Standard for Floating-Point Arithmetic
- European Union regulations

Supportability Examples

- Uncommon but important to certain IT circles (e.g., Facebook)
- The average time required to repair a defect in the server will be less than 1 day
  - How can it be "enforced"?

Supportability Examples

- Uncommon but important to certain IT circles (e.g., Facebook)
- The average time required to repair a defect in the server will be less than 1 day
  - How will your project demonstrate compliance?
  - Does the time to repair include the time to run all the tests?
  - Does rolling back to a previous (working) version "repair" the defect?
  - If so, can we keep a previous version of the server in hot-standby ready to go online if the new version fails?