

Non-functional requirements – Fault tolerance

Fault tolerance is the property that enables a system to continue operating properly in the event of one or more faults within some of its components.

(Wikipedia)

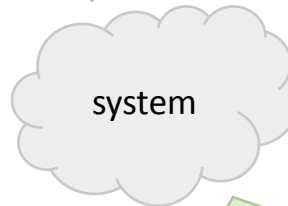
fault tolerance $\overset{?}{=}$ high availability

Sometimes I make mistakes
(errors)



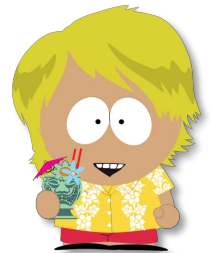
software
engineer

Oops... I have got a bug
(fault)



And it may cause a **failure** of
one of my components.

Because your system is
fault-tolerant, I do not see any
failures.



client

Non-functional requirements – Fault tolerance

fault tolerance

The system has the goal of zero downtime.

Achieved by using the same design principles and processes as for high availability.

Requires even more redundancy.

Resulting in a higher cost.

?

=

high availability

>

Downtime is possible and the system tries to minimize it.

Non-functional requirements - Resilience

Systems that in the face of faults can provide and maintain an acceptable level of service are called resilient systems.

resilience
resiliency


(almost)

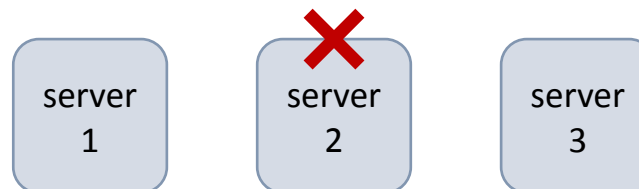
fault tolerance

I need faults to happen in my system periodically to test resilience (fault tolerance).

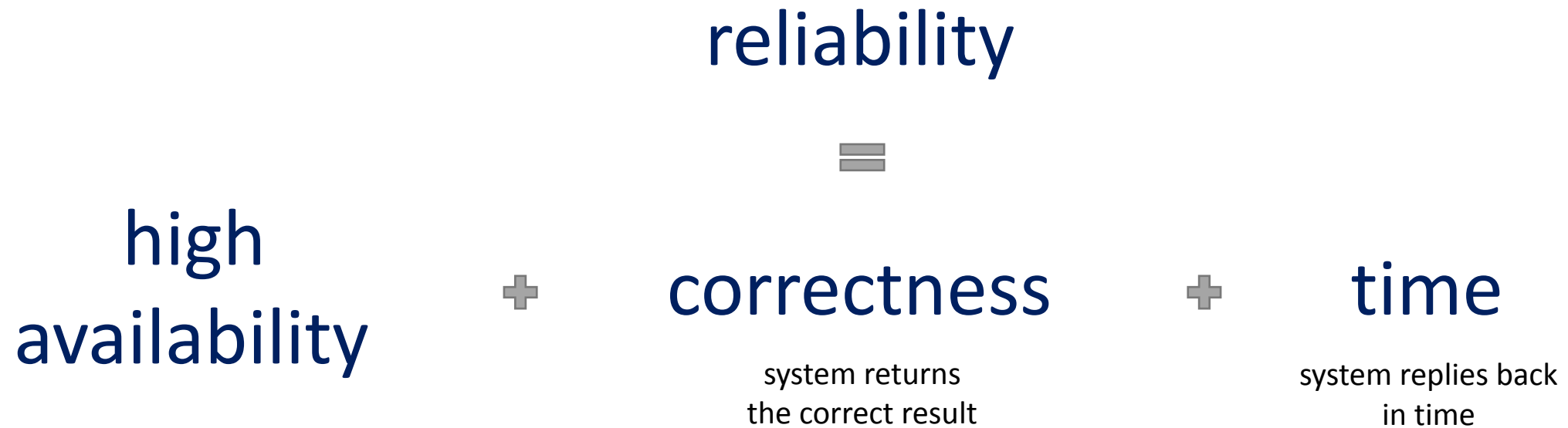


software
engineer

**Chaos engineering
rocks!**



Non-functional requirements - Reliability



Non-functional requirements - Reliability

My system is reliable and resilient.



software
engineer

system always performs its intended
functions correctly and in time

reliability

small downtime

high availability

close to zero downtime

fault tolerance

ability to quickly recover
from failures

resilience

system knows how to handle
expected failures

- server crash
- power outage in a data center
- network problems

system knows how to handle
unexpected failures

- load spikes
- dependency failures

reliability