
AWS
re:Invent
DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

ARC317-R1

Operational excellence: Best practices for resilient systems

Islam Ghanim
Principal Technical Account Manager
AWS

Will Laws
Senior Solutions Architect
AWS

How operations eat resilience for breakfast

Why systems fail

Hardware
Disk faults, clock skew

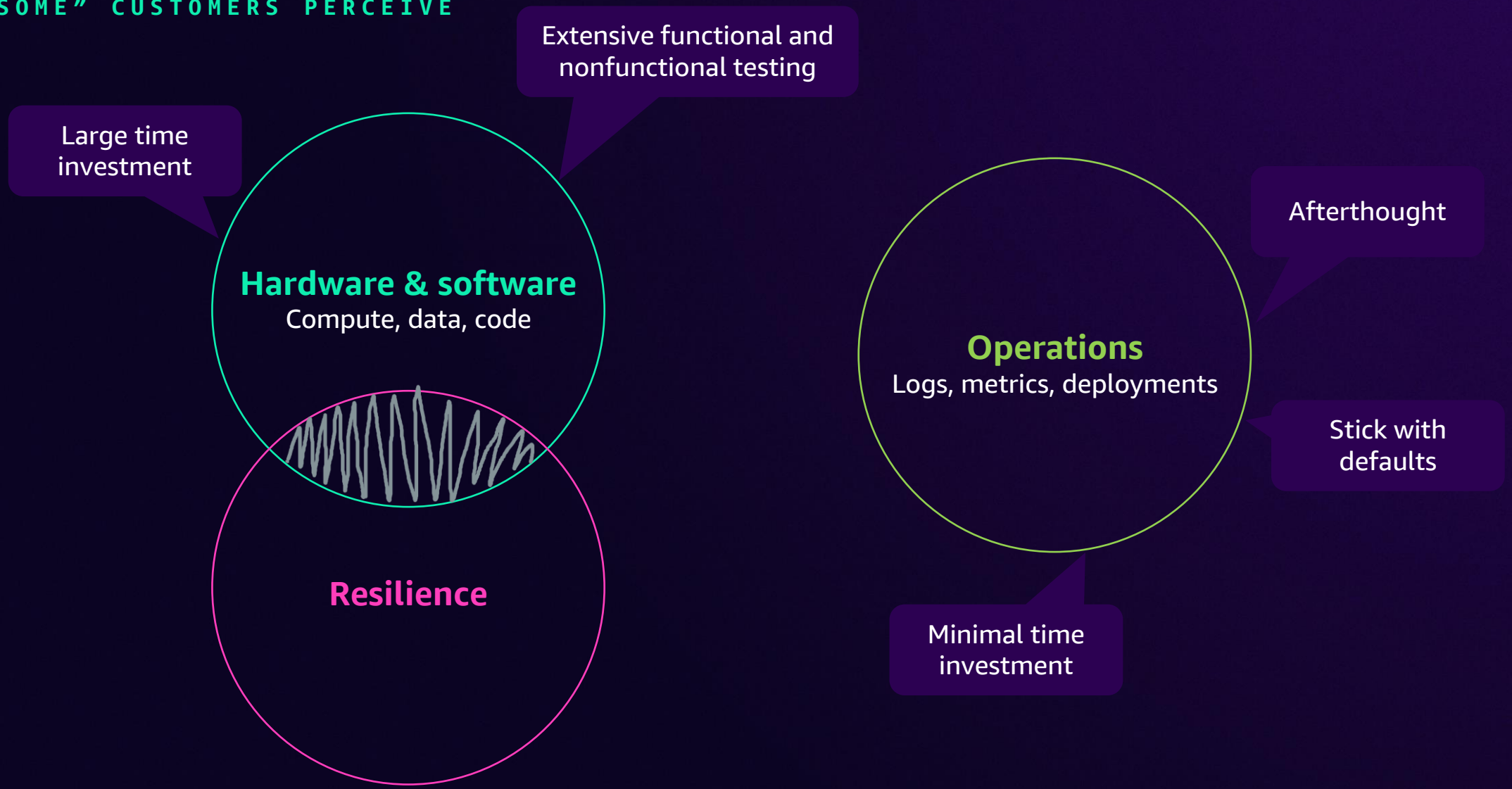
Software
Kernel, OS

Operations
Deployment, logs

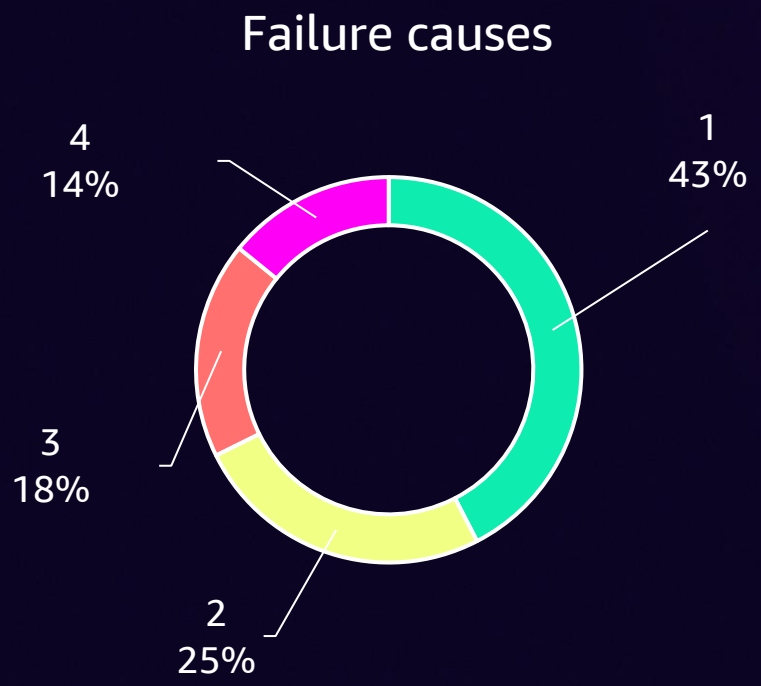
Environment
Power, networking

The perception

WHAT "SOME" CUSTOMERS PERCEIVE



Why systems “actually” fail



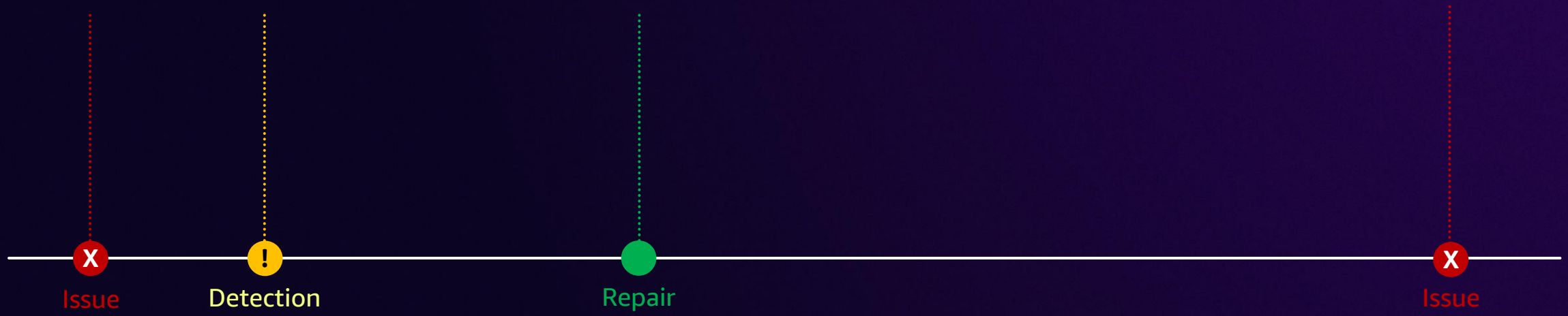
Failure mode	Probability	MTBF
Operations	**42% ***	31 years
Software	**25%**	50 years
Hardware	**18%**	73 years
Environment	**15%**	87 years

The techniques for fault-tolerant hardware are well documented and quite successful. Dealing with system configuration, operations, and maintenance remains an unsolved problem.

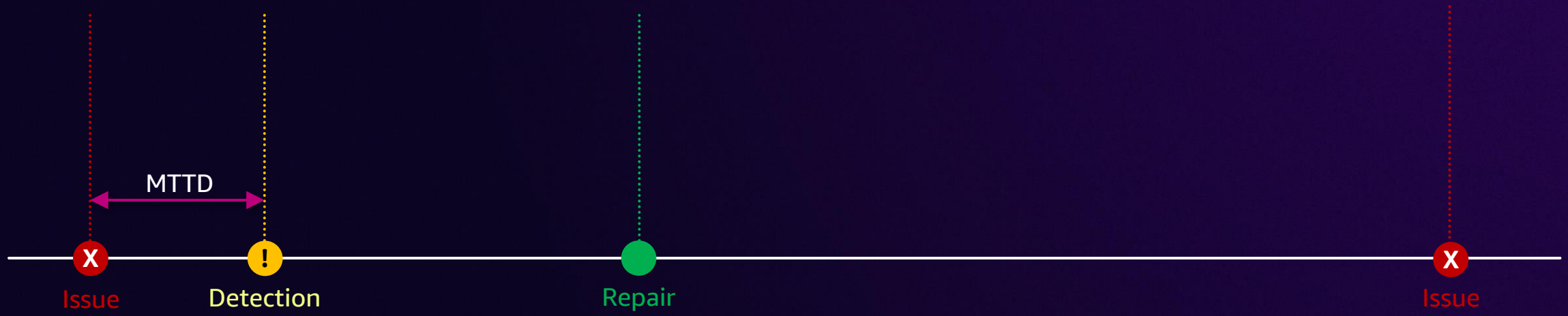
Jim Gray, 1985

"Why Do Computers Stop and What Can Be Done About It?"

MTBF, MTTR, and MTTD



MTBF, MTTR, and MTTD

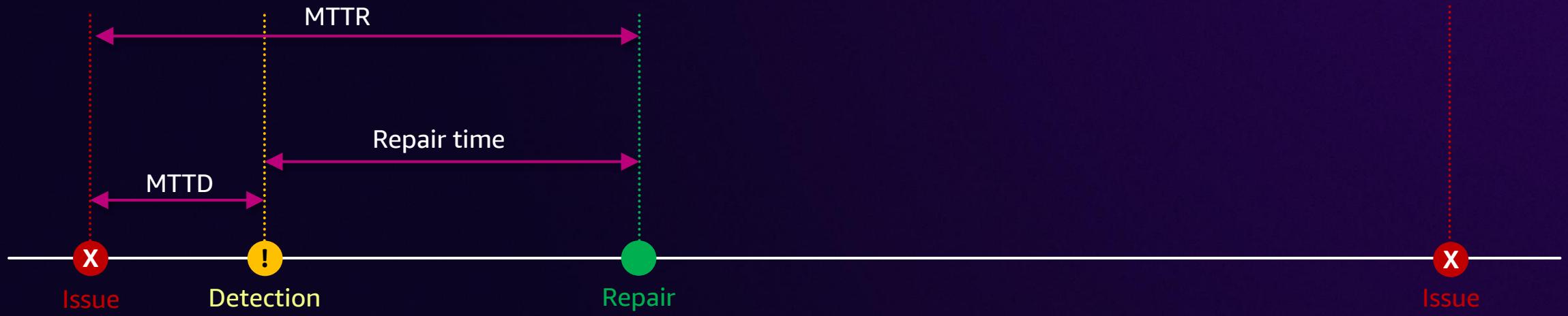


Goals

Lower MTTD

Detect faster

MTBF, MTTR, and MTTD

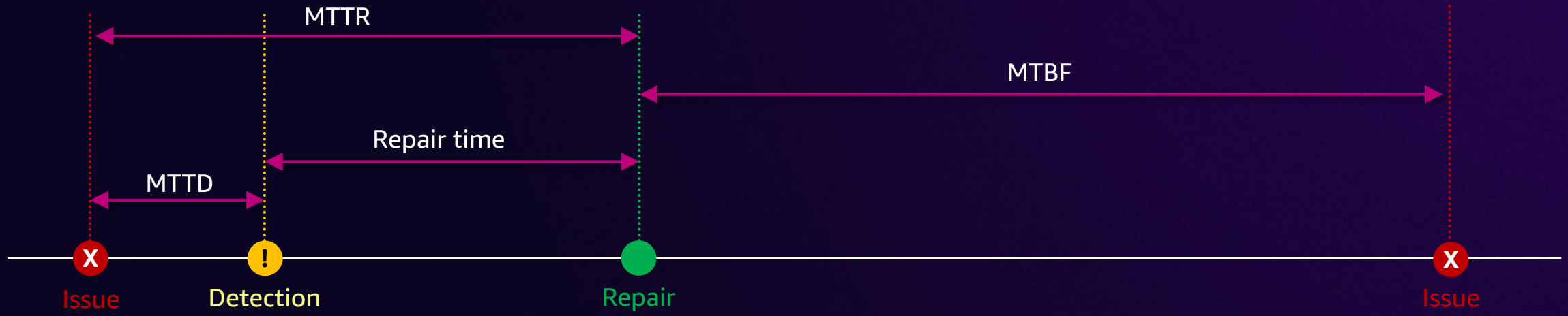


Goals

Lower MTTD
Detect faster

Lower MTTR
Repair/mitigate faster

MTBF, MTTR, and MTTD



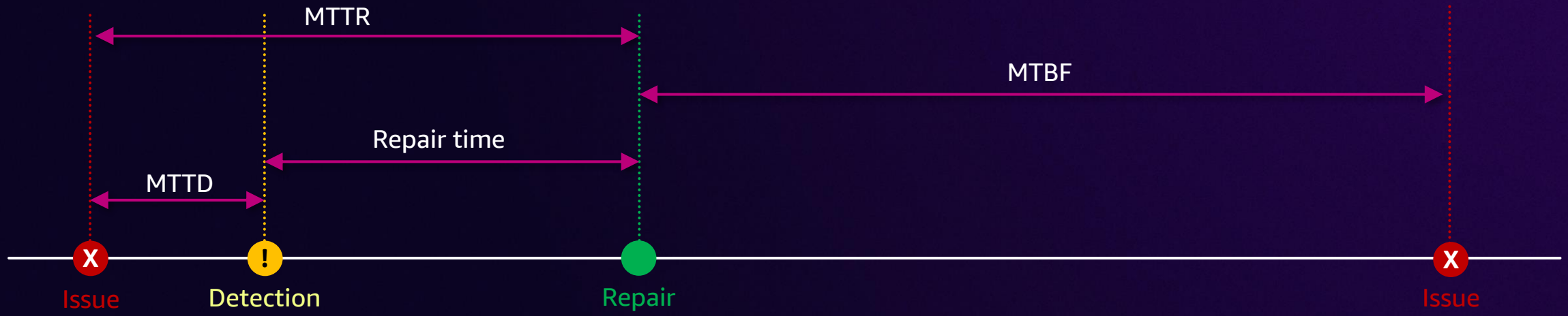
Goals

Lower MTTD
Detect faster

Lower MTTR
Repair/mitigate faster

Higher MTBF
Fail less frequently

MTBF, MTTR, and MTTD



$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

Goals

Lower MTTD
Detect faster

Lower MTTR
Repair/mitigate faster

Higher MTBF
Fail less frequently

The fact

WHAT CUSTOMERS SHOULD PERCEIVE

Story one: Logging

Between 12:30 PM and 12:45 PM, traffic to our e-commerce website dropped to zero. Visitors received an HTTP 504 error code due to a sudden loss of capacity of our core compute node, Amazon ECS. The root cause is a bad deployment that denied the cluster access to the Amazon CloudWatch API.

AnyCompany

A public statement on AnyCompany's e-commerce website availability incident

AWS Well-Architected

REL04-BP02 Implement loosely coupled dependencies

AWS Well-Architected guidance

Story two: Health checks

Between 9:05 AM and 09:10 AM users could not submit orders on the marketplace. At 09:10 AM, some orders were able to be submitted, and by 9:20 AM the order system was operating nominally.

This event resulted in a near 100% outage for 5 minutes and degraded capacity for 10 minutes. The root cause was a transient networking issue from 09:04 AM to 09:05 AM.

AnyCompany
Public statement on an incident impacting AnyCompany's two-sided marketplace system availability

AWS Well-Architected

REL11-BP03 Automate healing on all layers

Story three: Safe deployments

Between 8:00 AM and 9:00 AM our nationwide point of sale (POS) systems experienced 100% request failures. The root cause was a scheduled maintenance activity to upgrade a core database cluster that took longer than planned. The maintenance activity was expected to complete within 15 minutes, 2 hours before store opening time.

AnyCompany

Public statement on an incident impacting POS system availability

AWS Well-Architected

OPS06-BP01 Plan for unsuccessful changes

AWS Well-Architected guidance

Conclusion

Key takeaways

Resilience is not a silo

- Design for operations, don’t just stick with defaults
- Use resilience testing to validate your design and assumptions
- Consider failure modes
- Design for unsuccessful deployments

Thank you!

Please complete the session survey in the mobile app

Islam Ghanim

ghanimig@amazon.com
linkedin.com/in/islamghanim/

Will Laws

lwsw@amazon.com
https://laws.rocks