The Hidden Business Costs of Ignoring Performance Testing - The Silent Budget Killer

Author and Presenter: Sowmya Chintakindi
Independent Researcher and Sr. Performance Engineer
Email – sowmyar909@gmail.com

The Fifteenth International Conference on Performance, Safety and Robustness in Complex Systems and Applications
PESARO 2025
May 18, 2025 to May 22, 2025 - Nice, France

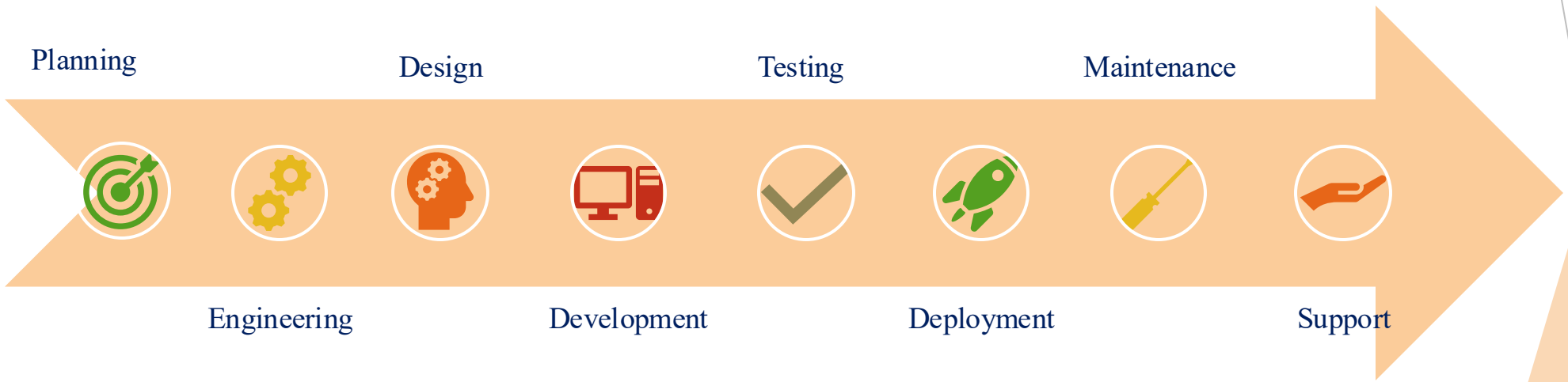
Sowmya Chintakindi

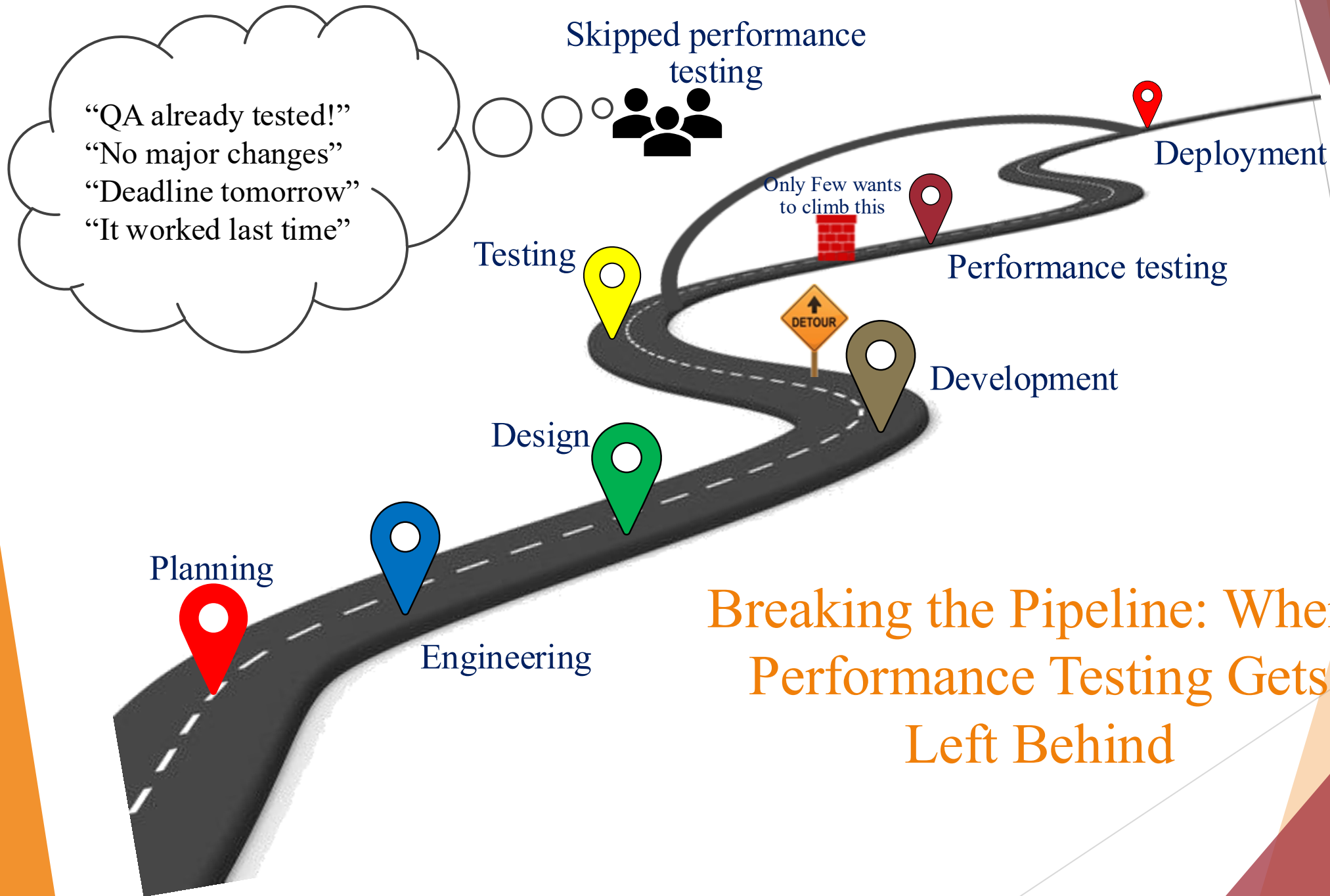
- With over 12 years of experience in the IT industry, Sowmya Chintakindi is a driven researcher at the forefront of performance testing, cloud computing, and sustainability.
- Passionate about pushing the boundaries of technology, she leads groundbreaking research that aims to revolutionize performance testing while promoting sustainable practices in tech.
- As an active OKC section professional activities chair and computer chapter chair, strongly advocates for diversity and inclusion.
- Additionally, she share insights on GreenOps — the intersection of sustainability and performance testing— through a popular blog on LinkedIn.

Agenda

- Introduction.
- What is SDLC?
- Performance testing is left behind.
- IT outages, its impact and strategies to avoid them.
- What is performance testing?
- Evolution of performance testing.
- Key performance metrics.
- Performance testing process.
- How performance testing is conducted.
- Types of performance testing.
- Case studies.
- Conclusion and future work.
- References.

Software Development Life Cycle



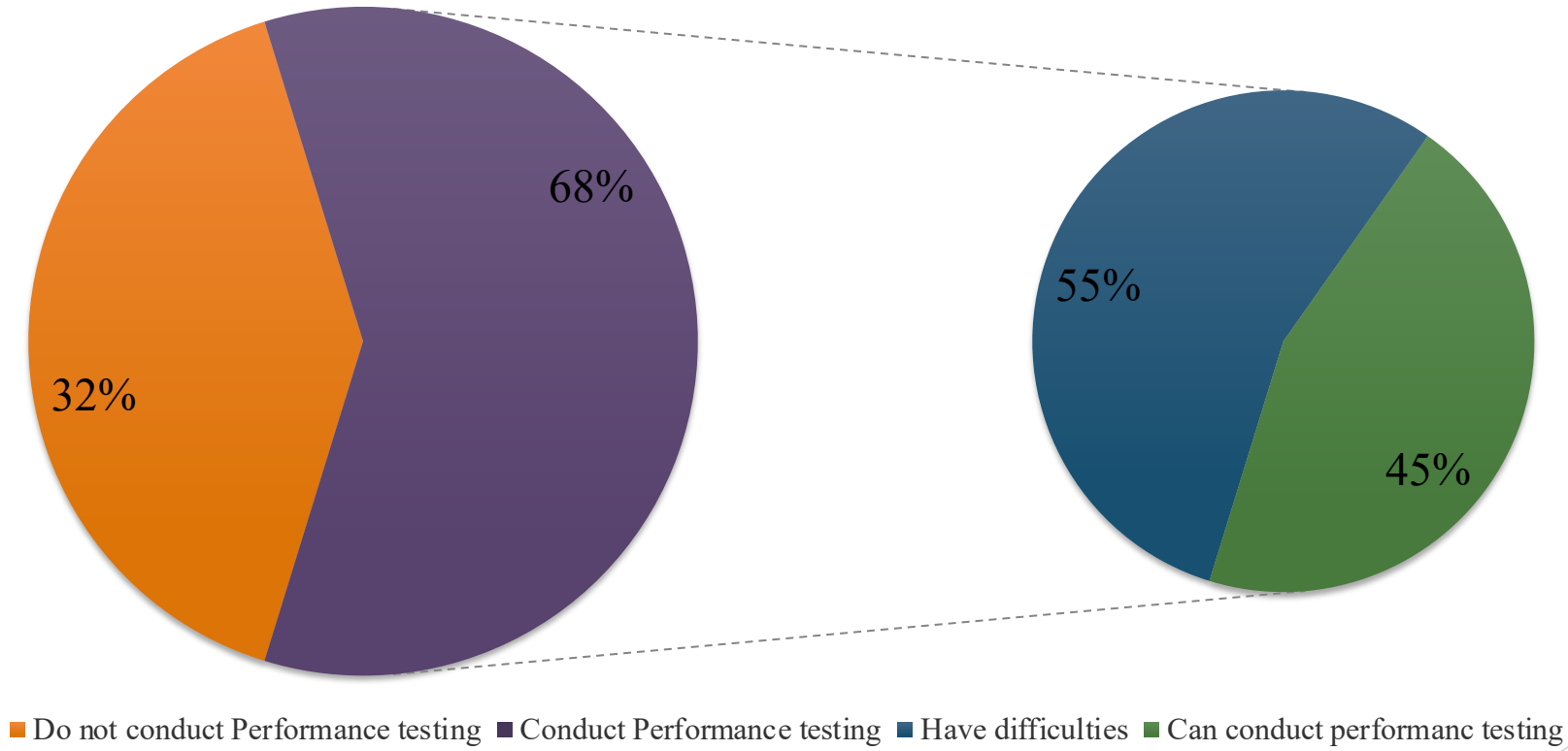


Breaking the Pipeline: Where Performance Testing Gets Left Behind

- ✗ No performance related changes done.
- ✗ It’s a minor release.
- ✗ QA covered the testing for the changes made.
- ✗ We’ve never had issues before.
- ✗ Our monitoring will catch it.
- ✗ Our infra scales automatically

- ✓ But interaction with upstream code changed.
- ✓ Minor changes cause major issues.
- ✓ Functional ≠ Performance tests.
- ✓ That was then — today's context is different.
- ✓ By the time it does, users are already impacted.
- ✓ Auto-scaling ≠ efficient or optimized under load

Understanding IT outages: causes, effects and preventive measures



Effects of these IT outages

25% of organizations report outages costing over $1 million.

Global 2000 companies lose $400B annually.

70% of the mobile page size is over 1MB.

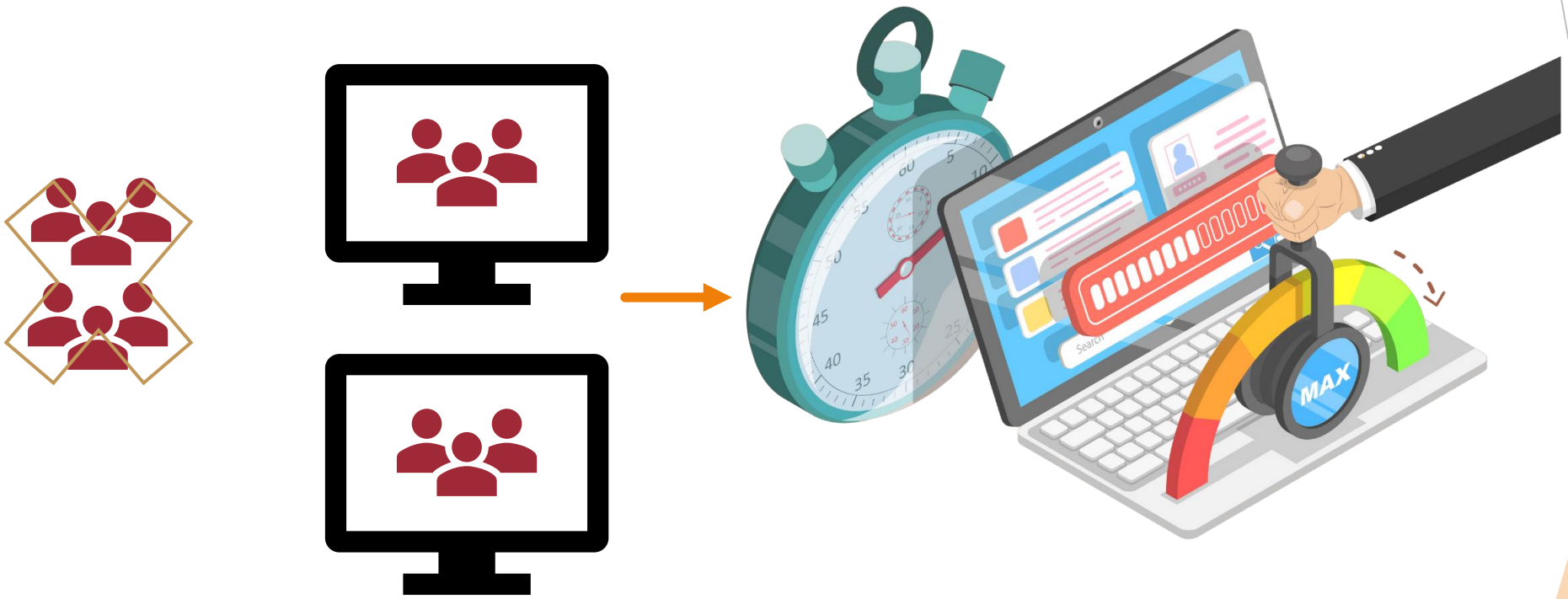
45% face costs between $100,000 and $1 million per outage.

70% of the mobile pages took 10secs to load full content.

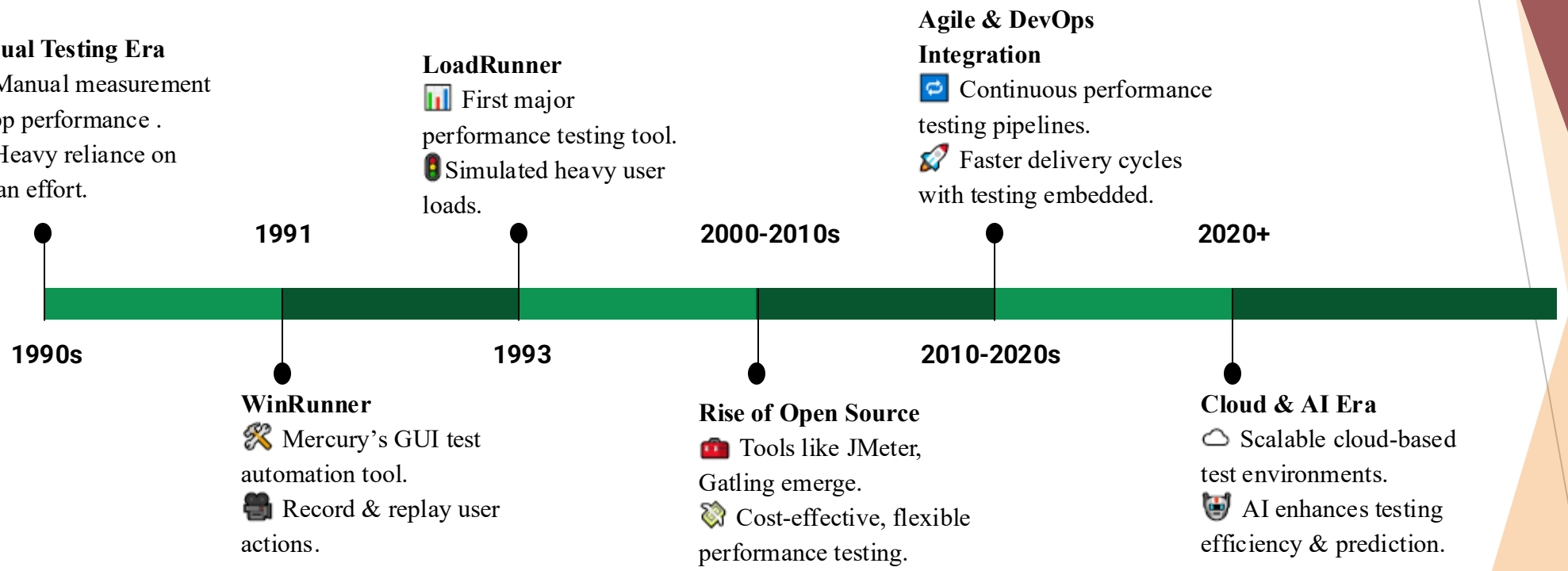
How we can avoid outages with performance testing

- Execute performance tests early in development.
- Monitor system utilization.
- Conduct different types of testing based on the load.
- Execute Chaos testing.
- Monitor system utilization.
- Conduct different types of testing based on the load.
- Execute Chaos testing.
- Introduce disaster recovery testing.
- Cloud auto scaling.
- Automate testing process in continuous delivery.

What is performance testing?



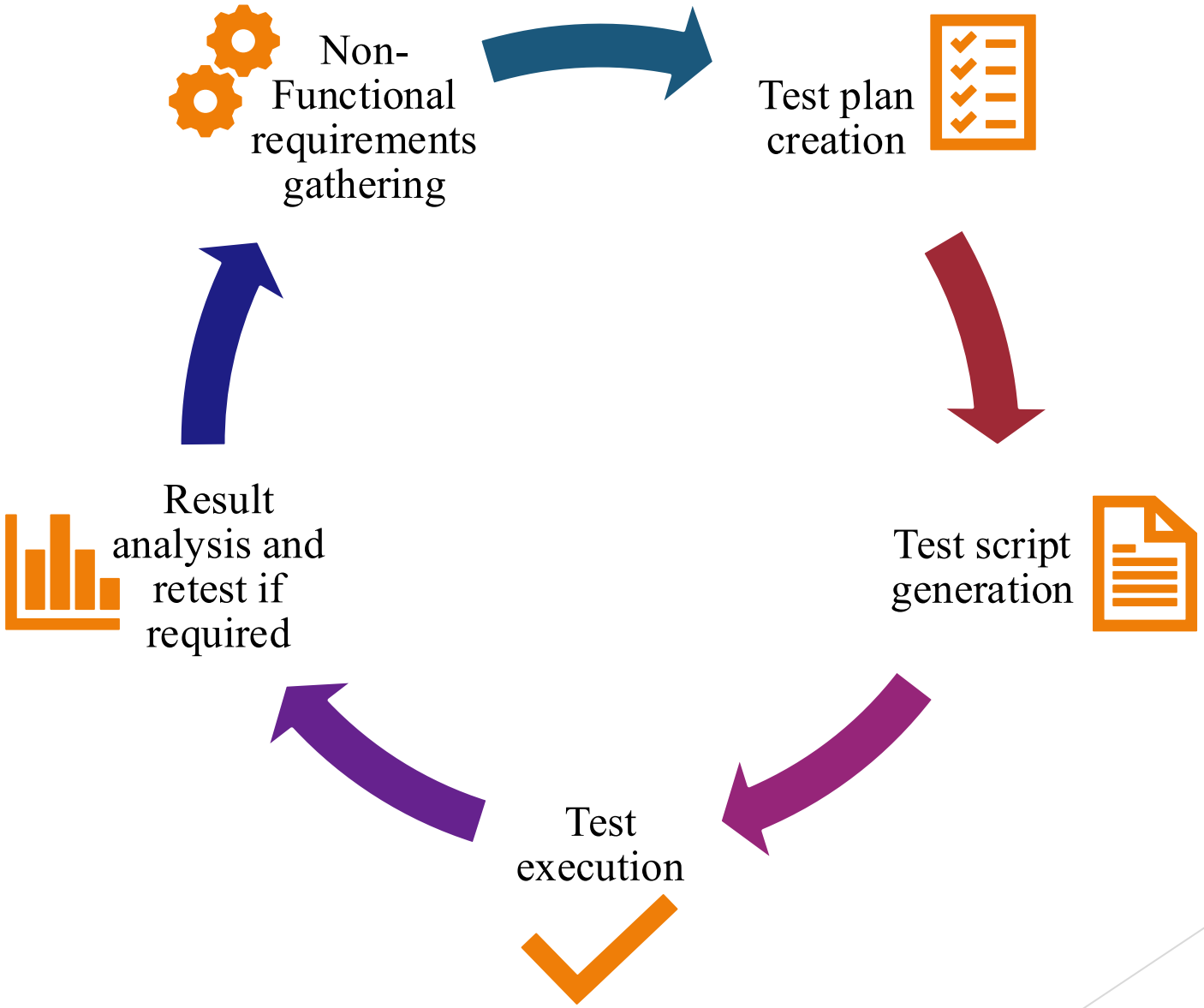
Evolution of performance testing



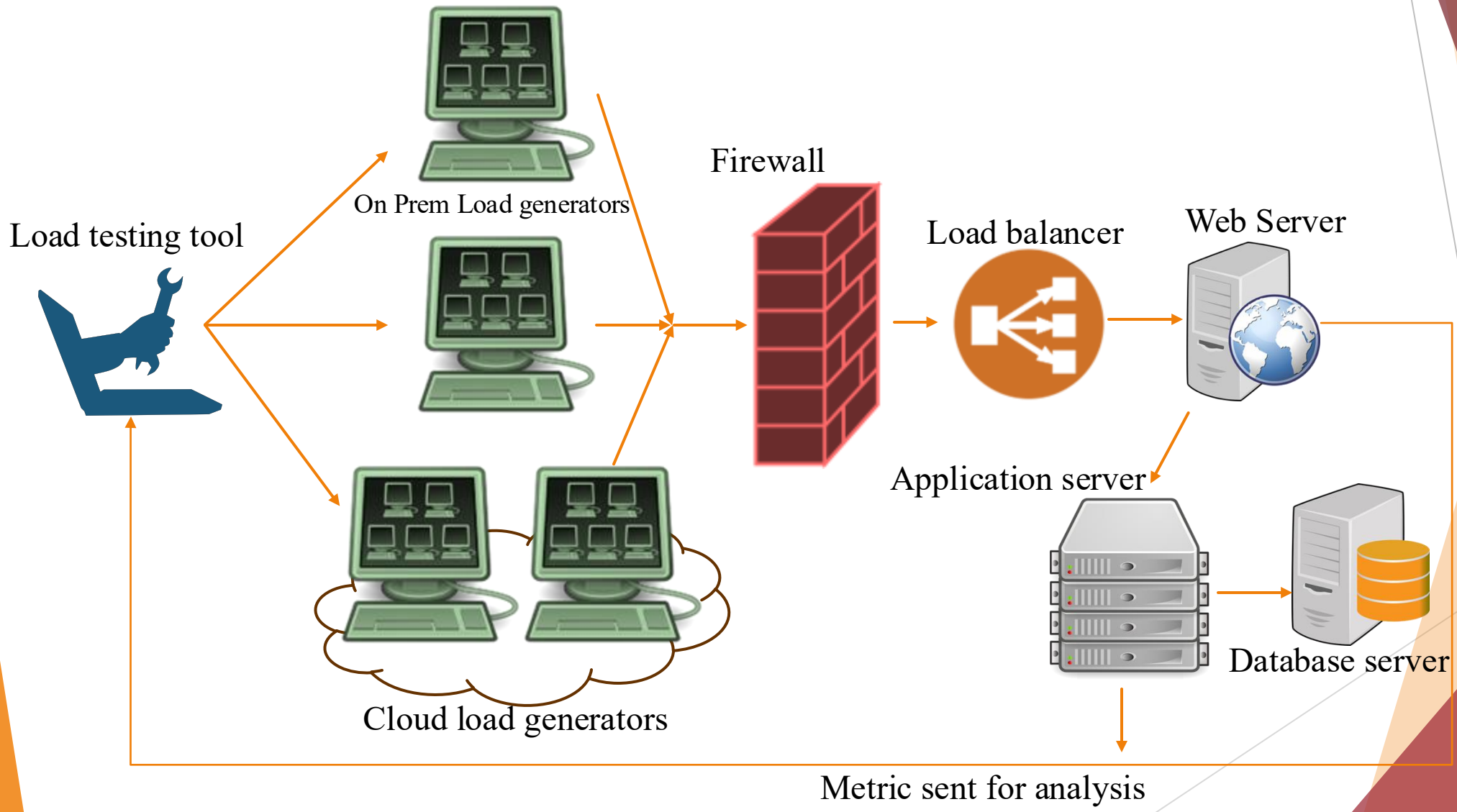
Key performance metrics

- Response times.
- User load
- System utilization of the server
- Latency
- Error rate
- Error rate
- Page loading time
- Page size
- Database metrics.

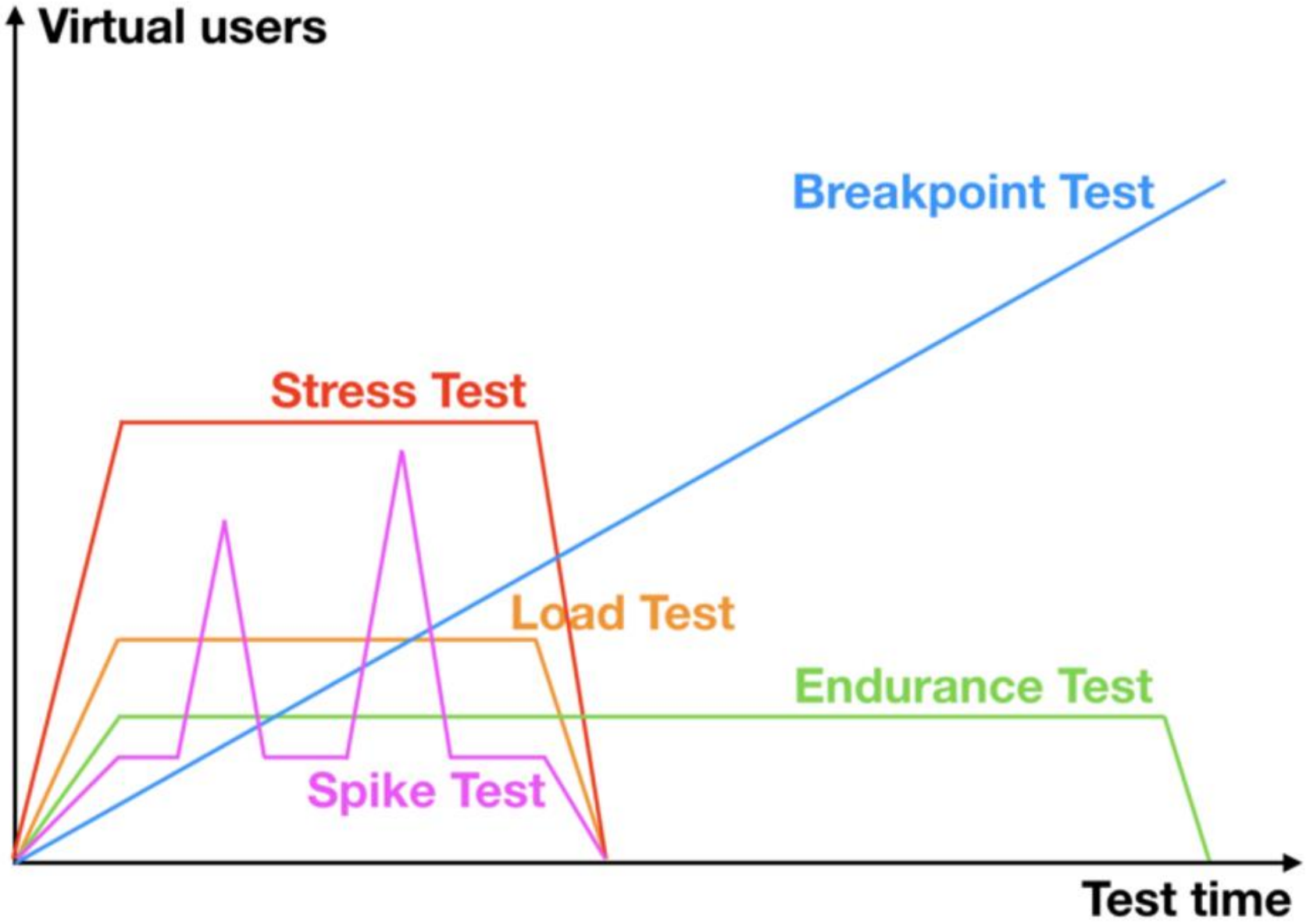
Performance testing process



Architecture of typical load testing environment



Types of performance testing?



Case Study 1: Azure

- Issue: More resources consumed by the failed ARM nodes.
- Root cause: A configuration change that has a code defect.
- Effect: Impacted Azure services.
- Downtime: 7 hours.
- Implications: Even small configuration changes can impact performance.
- Strategy to avoid this issue: Implementing negative performance tests.

Case Study 2: Jira

- Issue: Users saw 503 service unavailable errors.
- Root cause: A scheduled database upgrade.
- Effect: increased back pressure made requests to timeout.
- Downtime: 3.5 hours.
- Implications: Upgrades can break performance.
- Strategy to avoid this issue: rigorous performance testing with proper test plan.

Case Study 3: Microsoft 365

- Issue: Users saw 503 service unavailable errors.
- Root cause: A change that surged number of requests.
- Effect: Impacted processing capabilities of the infrastructure.
- Downtime: affected services for 7 hours.
- Implications: even partial outages can significantly affect user experience.
- Strategy to avoid this issue: Executing Spike testing.

Case Study 4: Netflix

- Issue: service was not available
- Root cause: Loading Netflix OCAs during off peak hours
- Effect: Received 500,000 reports about streaming problems.
- Downtime: 6 hours.
- Implications: This disruption emphasizes the importance of performance testing.
- Strategy to avoid this issue: testing with 20% more than peak production volume

Case Study 5: JCREW

- Issue: Shoppers frequently bumped with "hang on a sec" message.
- Root cause: servers couldn’t keep up with the load.
- Effect: J.Crew lost $775,000 due to unsold inventory
- Downtime: 5 hours.
- Implications: emphasizes the importance to prepare for peakseason
- Strategy to avoid this issue: performance tests before peak season.

Conclusion and future work

Outages can happen anytime — testing reduces the risk.

Early performance testing catches bottlenecks sooner.

Even with issues, testing speeds up resolution.

It ensures systems stay fast under heavy load.

Future work will explore real cases and reliability strategies.

References

1. E. Klotins, T. Gorschek, K. Sundelin, and R. Berntsson Svensson, “Towards cost-benefit evaluation for continuous software engineering activities.,” Empirical Software Engineering, vol. 27, p. 157, 2022. DOI: 10.1007/s10664-022-10191-w.
2. X. Han and T. Yu, “An empirical study on performance bugsnfor highly configurable software systems,” ser. ESEM ’16, New York, NY, USA: Association for Computing Machinery, 2016, ISBN: 9781450344272. DOI: 10.1145/2961111.2962602.
3. M. R. Woodward and M. A. Hennell, “Strategic benefits of software test management: A case study,” Journal of Engineering and Technology Management, vol. 22, no. 1, pp. 113–140, 2005, Research on Social Networks and the Organization of Research and Development, ISSN: 0923-4748. DOI: https : / / doi.org/10.1016/j.jengtecman.2004.11.006.
4. S. Zaman, B. Adams, and A. E. Hassan, “A qualitative study on performance bugs,” in 2012 9th IEEE Working Conference on Mining Software Repositories (MSR), 2012, pp. 199–208. DOI: 10.1109/MSR.2012.6224281.
5. U. institute, “Annual outages analysis 2023,” Last accessed: February, 2025, 2023, [Online]. Available: https : / / datacenter . uptimeinstitute . com / rs / 711 - RIA - 145 / images /AnnualOutageAnalysis2023.03092023.pdf.
6. Magnitia, “Software testing statistics – 2023,” Last accessed: February 20,2025, 2023, [Online]. Available: https://magnitia.\com/blog/software-testing-statistics-2023.
7. A. Petrosyan, “Most common root causes of it system and software-related outages worldwide,” Last accessed: February, 2025, 2023, [Online]. Available: https : / /www. statista .com/statistics/1482105/it- system- software- related- outages- rootcause/.
8. Splunk, “The hidden costs of downtime strike below the surface,” Last accessed: February, 2025, 2024, [Online]. Available: https://www.splunk.com/en_us/campaigns/the-hiddencosts-of-downtime.html.
9. Google, “Find out how you stack up to new industry benchmarks for mobile page speed,” Last accessed: February, 2025, 2017, [Online]. Available: https : / / think . storage . googleapis .com/docs/mobile-page-speed-new-industry-benchmarks.pdf.
10. C. thousand eyes - internt research team, “Internet and cloud telligence blog,” Last accessed: February, 2025, 2024, [Online]. Available: https://www.thousandeyes.com/blog/?cat=outage-analyses.
11. J. Yoon, “Thousands report netflix livestream crashes during mike tyson-jake paul fight,” Last accessed: February, 2025, 2024, [Online]. Available: https://www.nytimes.com/2024/11/16/business/media/netflix-outage-crash-boxing.html.