BASIC CONCEPTS OF PERFORMANCE TESTING

1.1 PRINCIPLES OF PERFORMANCE TESTING

LEARNING OBJECTIVE

- **PTFL-1.1.1 (K2) Understand the principles of performance (15 mins)**

3

RICECONSULTING

PRINCIPLES OF PERFORMANCE TESTING

- **Performance is an essential part of the user experience.**
- **Performance testing plays a critical role in establishing acceptable quality levels for the end user.**
  - Often closely integrated with other disciplines such as usability engineering, performance engineering and DevOps.

4

RICECONSULTING

PRINCIPLES OF PERFORMANCE TESTING (2)

- **Evaluation of other quality characteristics such as functionality, usability under conditions of load, such as during a performance test, may reveal load-specific issues which impact those characteristics.**
  - Not just limited to the web-based domain where the end user is the focus.
  - It is also relevant to different application domains with a variety of system architectures, such as classic client-server, distributed and embedded.

5

RICECONSULTING

PERFORMANCE QUALITY CHARACTERISTICS

- **Technically, performance is categorized in the ISO 25010 [ISO25000] Product Quality Model as a non-functional quality characteristic with these three sub characteristics:**
  - Time Behavior
  - Resource Utilization
  - Capacity
- **Performance testing usually concentrates on one or more of these sub-characteristics.**
- **All three of the above quality sub-characteristics will impact the ability of the system under test (SUT) to scale.**

6

RICECONSULTING

PERFORMANCE QUALITY CHARACTERISTICS AND RISK

- **Proper focus and prioritization depends on risk assessment and the needs of the various stakeholders.**
- **Test results analysis may identify other areas of risk that need to be addressed.**

7

RICECONSULTING

TIME BEHAVIOR

- **Generally the evaluation of time behavior is the most common testing goal.**
  - This examines the ability of a component or system to respond to user or system inputs within a specified time and under specified conditions.
  - Measurements of time behavior may vary from the “end-to-end” time taken by the system to responding to user input, to the number of CPU cycles required by a code module to execute a particular task.

8

RICECONSULTING

RESOURCE UTILIZATION

- If the availability of system resources is identified as a risk, the utilization of those resources (e.g., the allocation of limited RAM) may be investigated by conducting specific performance tests.

9

RICECONSULTING

CAPACITY

- If the capacity of the system (e.g., numbers of users or volumes of data) is identified as a risk, performance tests may be conducted to evaluate the suitability of the system architecture.

10

RICECONSULTING

EXPERIMENTATION

- **Since performance testing must consider these different quality sub-characteristics, it often takes the form of experimentation, which enables measurement and analysis of specific system parameters to take place.**
  - These may be conducted iteratively in support of system analysis, design and implementation to enable architectural decisions to be made and to help shape stakeholder expectations.

11

RICECONSULTING

GENERAL TESTING PRINCIPLES FOR PERFORMANCE TESTING

- **Tests must be aligned to the defined expectations of different stakeholder groups**
  - In particular users, system designers and operations staff.
- **The tests must be reproducible**
  - Statistically identical results must be obtained by repeating the tests on an unchanged system.

12

RICECONSULTING

GENERAL TESTING PRINCIPLES FOR PERFORMANCE TESTING (2)

- The tests must yield results that are both understandable and can be readily compared to stakeholder expectations.
- The tests can be conducted, where resources allow, either on complete or partial systems that are representative of the production system.
- The tests must be practically affordable and executable within the timeframe set by the project.

13

RICECONSULTING

EXAMPLE

- A city water utility embarked on a project to modernize its work order system and utility billing system.
- The project included new software, new hardware, and a new database.
- No one on the project thought about performance testing until a new consultant suggested it.

14

RICECONSULTING

EXAMPLE (2)

- **As a result of the performance tests it was learned that:**
  - The nightly billing system ran for 96 hours, instead of the 2 hours needed by the old system
  - The online application could only accommodate four users concurrently.
    - There were 40 people in the department.
- **The system was never implemented because the performance problems could never be resolved.**

15

RICECONSULTING

EXAMPLE (3)

- **In this real-life example:**
  - The principles of performance testing were not understood or followed.
  - Stakeholders expectations were not defined or managed.
  - Vendor management intentionally withheld knowledge of the potential performance problems.
  - The city sued the vendor and recouped all the fees (1.8 million dollars) paid to the vendor.
  - It took five more years before a new system was eventually implemented.

16

RICECONSULTING

1.2 TYPES OF PERFORMANCE TESTING

LEARNING OBJECTIVE

- **PTFL-1.2.1 (K2) Understand the different types of performance testing (15 mins)**

TYPES OF PERFORMANCE TESTING

- **Different types of performance testing can be defined.**
  - Each of these may be applicable to a given project, depending on the goals of the test.
- **Performance Testing**
  - An umbrella term including any kind of testing focused on performance (responsiveness) of the system under different volumes of load.
- **Load Testing**
  - Focuses on the ability of a system to handle increasing levels of anticipated realistic loads resulting from transaction requests generated by controlled numbers of concurrent users or processes.

TYPES OF PERFORMANCE TESTING

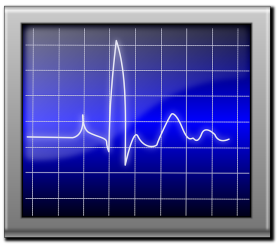
- **Stress Testing**
  - Focuses on the ability of a system or component to handle peak loads that are at or beyond the limits of its anticipated or specified workloads.
  - Also used to evaluate a system's ability to handle reduced availability of resources such as accessible computing capacity, available bandwidth, and memory.

TYPES OF PERFORMANCE TESTING

- **Scalability Testing**
  - Focuses on the ability of a system to meet future efficiency requirements which may be beyond those currently required.
  - The objective is to determine the system's ability to grow (e.g., with more users, larger amounts of data stored) without exceeding the currently specified performance requirements or failing.
    - Once the limits of scalability are known, threshold values can be set and monitored in production to provide a warning of impending problems.
    - The production environment may also be adjusted with appropriate amounts of hardware.

TYPES OF PERFORMANCE TESTING

- **Spike Testing**
  - Tests the ability of a system to recover from sudden bursts of peak loads and return afterward to a steady state.
- **Endurance Testing**
  - Tests the stability of the system over a time frame specific to the system's operational context.
  - Verifies there are no resource capacity problems that may eventually degrade performance and/or cause failures
    - e.g., memory leaks, database connections, thread pools

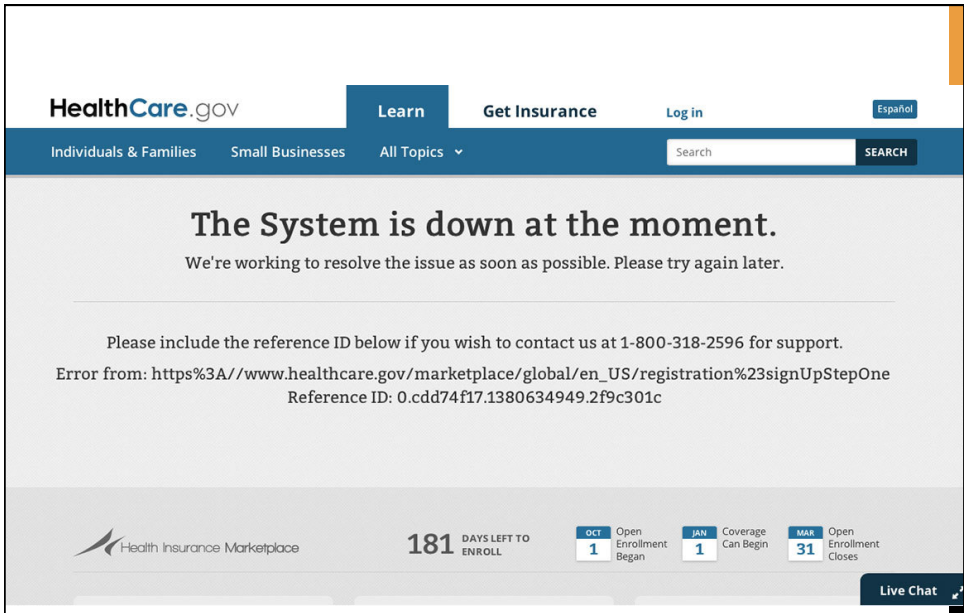


TYPES OF PERFORMANCE TESTING

- **Concurrency Testing**
  - Tests the impact of situations where specific actions occur simultaneously
    - For example, when large numbers of users log in at the same time.
  - Concurrency issues are notoriously difficult to find and reproduce, particularly when the problem occurs in an uncontrolled environment such as production.
- **Capacity Testing**
  - Determines how many users and/or transactions a given system will support and still meet the stated performance goals.

EXAMPLE

- **One of the most notable examples of system failure was the launch of the Affordable Healthcare Act website in October of 2013.**
- **"Problems with the website were apparent immediately. High website demand (250,000 users [5 times more than expected]) caused the website to go down within 2 hours of launch."**
- **"A total of 6 users completed and submitted their applications and selected a health insurance plan on the first day."**
  - https://rctom.hbs.org/submission/the-failed-launch-of-www-healthcare-gov/



25

RICECONSULTING

EXAMPLE (2)

- **In this example, there were *many* contributing causes beside system performance.**
  - https://oig.hhs.gov/oei/reports/oei-06-14-00350.pdf
- **However, the proper system and performance testing could have provided proof in advance that the website was not ready for launch.**
  - Even if such testing had been performed, the attitude was that the site was going to be launched regardless of readiness.

26

RICECONSULTING

EXAMPLE (3)

- **In the healthcare.org example, the following types of performance testing would have been needed:**
  - **Load testing** to simulate at least 250,000 users.
    - "Currently we are able to reach 1,100 users before response time gets too high," a bulletin from HHS stated the day before launch.
  - **Stress testing** to know how many users could be accommodated.
    - In this case, the number was 1,100.

27

RICECONSULTING

EXAMPLE (4)

- **Scalability testing to see if the website could handle increasing user loads.**
- **Spike testing to see if the site could handle the load on day 1.**
- **Endurance testing to see if the site could stay up and running for several days, at least.**
- **Concurrency testing to see how the system would process many people applying for healthcare policies at the same time.**
- **Capacity testing to see how many applications could be completed daily.**
  - In this case, the number was 6!

28

RICECONSULTING

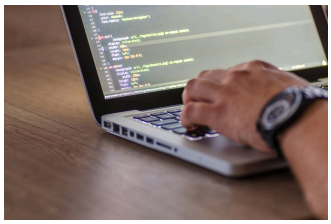
1.3 TESTING ACTIVITIES IN PERFORMANCE TESTING

LEARNING OBJECTIVE

- **PTFL-1.3.1 (K1) Recall activities in performance testing (10 mins)**

TESTING ACTIVITIES IN PERFORMANCE TESTING

- **Test types used in performance testing include static activities in performance engineering (planning and design) and dynamic testing.**
  - Static Testing
    - Testing a work product without code being executed.
      - ISTQB Glossary



31

RICECONSULTING

STATIC TESTING IN PERFORMANCE TESTING

- **In the case of performance testing, static activities are often more important than static activities for functional testing.**
  - This is because so many critical performance defects are introduced in the architecture and design of the system.
  - These defects can be introduced by misunderstandings or a lack of knowledge by the designers and architects.
  - These defects can also be introduced because the requirements did not adequately capture the response time, throughput, or resource utilization targets, the expected load and usage of the system, or the constraints.

32

RICECONSULTING

STATIC ACTIVITIES FOR PERFORMANCE

- Reviews of requirements
- Reviews of database schemas, entity-relationship diagrams, metadata, stored procedures and queries
- Reviews of the system and network architecture
- Reviews of critical segments of the system code
- Modeling of system resource requirements and/or behavior using spreadsheets or capacity planning tools
- Analysis of other potential performance degradation factors

33

RICECONSULTING

BENEFITS OF PERFORMANCE ENGINEERING

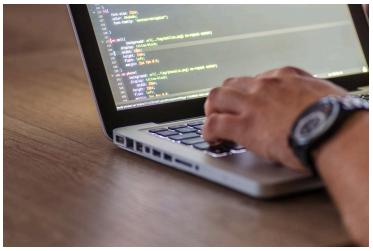
- Good performance engineering can help project teams avoid the late discovery of critical performance defects during higher levels of testing, such as system integration testing or user acceptance testing.
- Performance defects found late in the testing can be extremely costly and may even lead to the cancellation of entire projects.
- As the system is built, dynamic performance testing should start as soon as possible.

34

RICECONSULTING

DYNAMIC TESTING

- **Testing that involves the execution of the software of a component or system.**
  - ISTQB Glossary



35

RICECONSULTING

OPPORTUNITIES FOR DYNAMIC PERFORMANCE TESTING

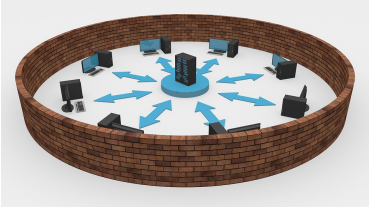
- **During unit testing, including using profiling information to determine potential bottlenecks and dynamic analysis to evaluate resource utilization**
  - **Component integration testing** - across key use cases and workflows, especially when feature integration or backbone integration methods are used
  - **System testing** - end-to-end behaviors under various load conditions
  - **System integration testing** - especially for data flows and workflows across key inter-system interfaces
  - **Acceptance testing** - to build user, customer, and operator confidence in the proper performance of the system and to fine tune the system under real world conditions (but not to find performance defects)

36

RICECONSULTING

THE ROLE OF TEST ENVIRONMENTS

- In higher levels of testing such as system testing and system integration testing, the use of realistic environments, data, and loads are critical for accurate results

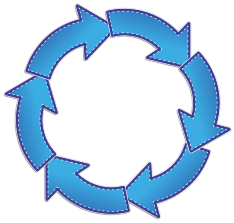


37

RICECONSULTING

THE ROLE OF AN SDLC

- In Agile and other iterative-incremental lifecycles, teams should incorporate static and dynamic performance testing into early iterations rather than waiting for final iterations to address performance risks.



38

RICECONSULTING

THE ROLE OF SIMULATORS

- **If custom or new hardware is part of the system, early dynamic performance tests can be performed using simulators.**
  - However, it is good practice to start testing on the actual hardware as soon as possible, as simulators often do not adequately capture resource constraints and performance-related behaviors.

39

RICE CONSULTING

1.4 THE CONCEPT OF LOAD GENERATION

40

RICE CONSULTING

LEARNING OBJECTIVE

- **PTFL-1.4.1 (K2) Understand the concept of load generation (10 mins)**

41

RICECONSULTING

THE CONCEPT OF LOAD GENERATION

- **In order to carry out the various types of performance tests, representative system loads must be modeled, generated and submitted to the system under test.**
- **The efficient and reliable generation of a specified load is a key success factor when conducting performance tests.**
  - There are different options for load generation.

42

RICECONSULTING

DIFFERENCES IN TEST LOADS

- **Loads are comparable to the data inputs used for functional test cases, but differ in the following principal ways:**
  - A performance test load must represent many user inputs, not just one
  - A performance test load may require dedicated hardware and tools for generation
  - Generation of a performance test load is dependent on a degree of functional stability in the system under test

LOAD GENERATION VIA THE USER INTERFACE

- **This may be an adequate approach if only a small numbers of users are to be represented and if the required numbers of software clients are available from which to enter required inputs.**
- **This approach may also be used in conjunction with functional test execution tools, but may rapidly become impractical as the numbers of users to be simulated increases.**

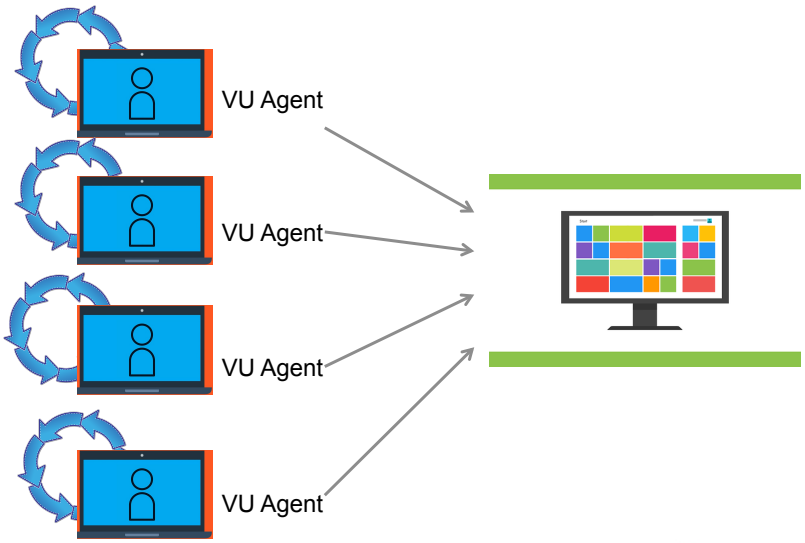
THE IMPACT OF UI STABILITY

- **The stability of the user interface (UI) also represents a critical dependency which can impact the repeatability of performance tests and may significantly affect the maintenance costs.**
- **Testing through the UI may be the most representative approach for end-to-end tests.**

45

RICECONSULTING

LOAD GENERATION VIA THE USER INTERFACE



46

RICECONSULTING

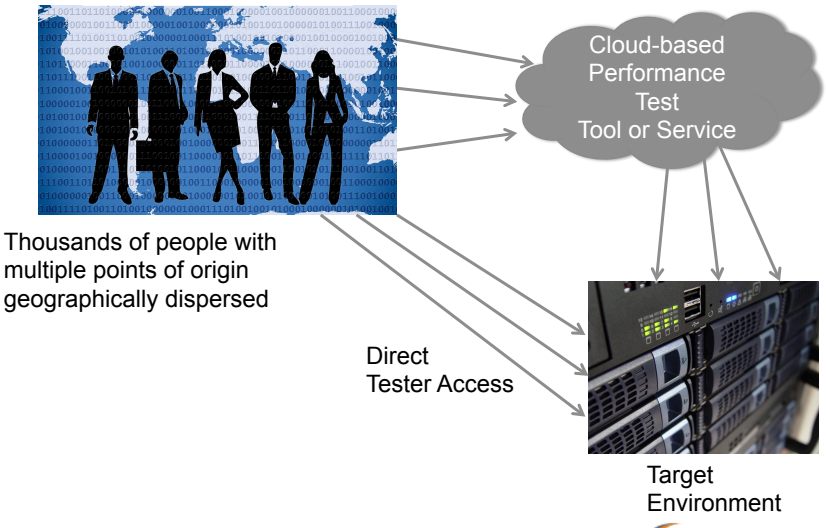
LOAD GENERATION USING CROWDS

- This "crowdsourcing" approach depends on the availability of a large number of testers who will represent real users.
- In crowd testing, the testers are organized such that the desired load can be generated.
- This may be a suitable method for testing web-based applications and may enable very large numbers of users to be utilized.
- The load generation will not be as reproducible and precise as other options and is organizationally more complex.

47

RICECONSULTING

CROWDSOURCED PERFORMANCE TESTING



48

RICECONSULTING

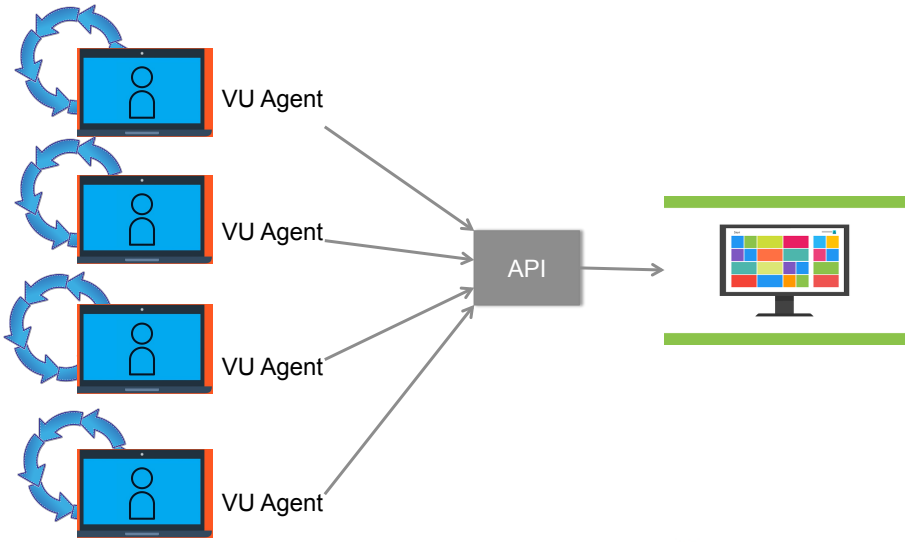
GENERATION VIA THE APPLICATION PROGRAMMING INTERFACE (API)

- This approach is similar to using the UI for data entry, but uses the application's API instead of the UI to simulate user interaction with the system under test.
- The approach is therefore less sensitive to changes (e.g., delays) in the UI and allows the transactions to be processed in the same way as they would if entered directly by a user via the UI.
- Dedicated scripts may be created which repeatedly call specific API routines and enable more users to be simulated compared to using UI inputs.

49

RICECONSULTING

GENERATION VIA THE APPLICATION PROGRAMMING INTERFACE (API)



50

RICECONSULTING

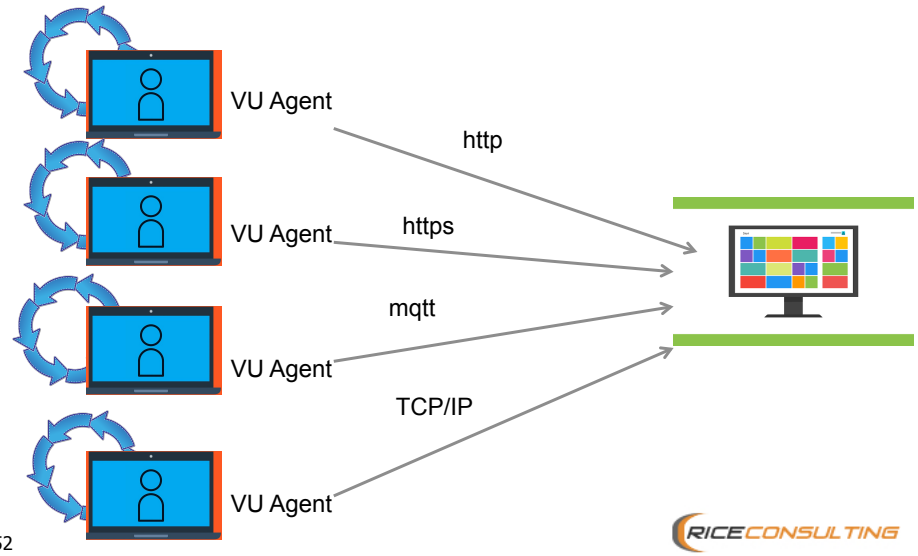
GENERATION USING CAPTURED COMMUNICATION PROTOCOLS

- This approach involves capturing user interaction with the system under test at the communications protocol level and then replaying these scripts to simulate potentially very large numbers of users in a repeatable and reliable manner.

51

RICECONSULTING

GENERATION USING CAPTURED COMMUNICATION PROTOCOLS - EXAMPLE



52

RICECONSULTING

EXAMPLE

- **A large e-commerce web site is planning the performance testing of a new architecture that will have a new user interface and will use new APIs to access a variety of services in the cloud.**
- **Some performance tests will directly access the UI to simulate actual user activity.**
  - These will be performed across an https:// protocol.
  - The activity will be based on user scenarios such as searching, ordering, etc.

53

RICECONSULTING

EXAMPLE (2)

- **Other performance tests will directly access APIs to simulate high transaction load to the services.**
  - These will be performed across an https:// protocol to access REST services.
  - Examples include currency conversion, inventory queries with vendors, and shipping rates with carriers.

54

RICECONSULTING

1.5 COMMON FAILURES IN PERFORMANCE TESTING AND THEIR CAUSES

RICE CONSULTING

LEARNING OBJECTIVE

- **PTFL-1.5.1 (K2) Give examples of common failures of performance testing and their causes (10 mins)**

RICE CONSULTING

COMMON FAILURES IN PERFORMANCE TESTING AND THEIR CAUSES

- **The following are some examples of common failures (including system crashes), along with typical causes, as shown on the following slides.**

57

RICECONSULTING

SLOW RESPONSE UNDER ALL LOAD LEVELS

- **In some cases, response is unacceptable regardless of load.**
  - This may be caused by underlying performance issues, including, but not limited to, bad database design or implementation, network latency, and other background loads.
- **These problems can be detected during functional and usability testing, not just performance testing, so test analysts should know to look for them and report them.**

58

RICECONSULTING

SLOW RESPONSE UNDER MODERATE-TO-HEAVY LOAD LEVELS

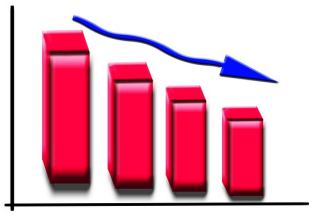
- In some cases, response degrades unacceptably with moderate-to-heavy load, even when such loads are entirely within normal, expected, allowed ranges.
- Underlying defects include saturation of one or more resources and varying background loads.

59

RICECONSULTING

DEGRADED RESPONSE OVER TIME

- In some cases, response degrades gradually or severely over time.
- Underlying defects include memory leaks, disk fragmentation, increasing network load over time, and database growth.



60

RICECONSULTING

INADEQUATE OR GRACELESS ERROR HANDLING UNDER HEAVY LOAD

- **In some cases, response time is acceptable but error handling degrades at high and beyond-limit load levels.**
- **Underlying defects include insufficient resource pools, undersized queues and stacks, and too rapid time-out settings.**

61

RICECONSULTING

FAILURE EXAMPLES

- **A web-based application that provides information about a company's services does not respond to user requests within seven seconds.**
  - The user goes to another company that provides similar services.
- **A system crashes or is unable to respond to user inputs when subjected to a sudden large number of user requests (e.g., ticket sales for a major sporting event).**
  - The capacity of the system to handle this number of users is inadequate.

62

RICECONSULTING

FAILURE EXAMPLES (2)

- **System response is significantly degraded when users submit requests for large amounts of data (e.g., a large and important report is posted on a web site for download).**
  - The capacity of the system to handle the generated data volumes is insufficient.
- **Batch processing is unable to complete before online processing is needed.**
  - The execution time of the batch processes is insufficient for the time period allowed.

63

RICECONSULTING

FAILURE EXAMPLES (3)

- **A real-time system runs out of RAM when parallel processes generate large demands for dynamic memory which cannot be released in time.**
  - The RAM is not dimensioned adequately, or requests for RAM are not adequately prioritized.
- **A real-time system component A which supplies inputs to real-time system component B is unable to calculate updates at the required rate.**
  - The overall system fails to respond in time and may fail.
  - Code modules in component A must be evaluated and modified ("performance profiling") to ensure that the required update rates can be achieved.

64

RICECONSULTING

SUMMARY

- **High performance is one of the most important quality attributes of a system or web site.**
  - Correctness can be overshadowed by poor performance.
- **Performance testing involves more than just running a tool.**
- **There are a variety of sub-types under performance testing.**
- **Static testing has an important role in performance testing.**
- **Tools are needed to achieve sustained load generation that can be reproduced.**
- **Performance failures can occur in a variety of ways.**

65

RICE CONSULTING