Performance Success Criteria

How Test Results are Evaluated and Rated

Version 1.6

Xceptance Software Technologies, Inc.
One Broadway
Cambridge, MA 02142
United States

contact@xceptance.com
+1-617-225-4335
www.xceptance.com

Xceptance Software Technologies GmbH
Leutragraben 2-4
07743 Jena
Germany

kontakt@xceptance.de
+49 3641 376300
www.xceptance.com/de

Introduction

Motivation

You might remember headlines such as "Back to Black Friday: Performance Testing Lessons From Target"[1] or "Black Friday: Currys, Tesco, Argos, PC World and Boots websites crash in shopping frenzy"[2]. These are the extreme motivations for performance testing, but there are others, such as "Web Performance Optimization"[3] and "How Loading Time Affects Your Bottom Line"[4]. The last source cites several statistics, including the following:

- 73% of mobile internet users say that they've encountered a website that was too slow to load.
- 38% of mobile internet users say that they've encountered a website that wasn't available.
- 47% of consumers expect a web page to load in 2 seconds or less.
- 40% of people abandon a website that takes more than 3 seconds to load.
- A 1-second delay in page response can result in a 7% reduction in conversions.
- If an e-commerce site is making $100,000 per day, a 1-second page delay could potentially cost them $2.5 million in lost sales every year.

So what do we learn from all of that? Performance is a very important business driver, but often ignored in favor of features and content.

Two secondary factors are also important: Stability and Correctness. Fast is not enough when the response is not what is expected. A fast cart is useless if it does not do what it is supposed to do. Stability, on the other hand, means that performance and correctness do not change over time.

Communication Challenges

Load and performance testing is an essential part of the life cycle of any modern application. But sharing the state of the tests and talking about goals is extremely hard, partially because not everyone speaks the same language. On one side it is about the impact on the business, and on the other side about the technical implications of design decisions or what should be done to improve performance.

On many occasions the target goals are already vague and not unified in the sense that everyone understands them.

This document suggests general criteria for a load and performance test, explains them in detail and how a rating is finally applied to aid the decision process and support all participating organizations.

[1] https://blazemeter.com/blog/back-black-friday-performance-testing-lessons-target
[2] http://www.independent.co.uk/news/business/news/9889225.html
[3] http://www.stevesouders.com/blog/2010/05/07/wpo-web-performance-optimization/
[4] https://blog.kissmetrics.com/loading-time/

Performance Testing in a Nutshell

Performance can be seen from different angles and everyone has a slightly different understanding of the term at the same time. Therefore we want to show and describe the different performance views first. They are not complete, because there are many flavours and combinations possible, but these are the most important ones for our later discussion. To be able to read the suggested performance runtimes and results correctly, we have to make the differentiation between testing the server delivery time alone or a full download including rendering of all required resources.

Performance Tests and Load Tests

A common misunderstanding when starting to discuss the topic of performance, especially when talking about the ways to assess performance, are the terms surrounding the testing. We just want to explain the basics here and will later in the document no longer distinguish between load and performance testing.

What Is a Load Test?

When performing a load test, the future use of the tested system is simulated and assessed considering a certain number of users and transactions. The first goal is to detect functional failures that only appear under parallel and intense usage of the system. The second goal is to measure timing and resource behavior under load as particular values, such as response time, main memory usage, or general system utilization, are determined.

Load tests can focus on different system aspects. Different load profiles and test scenarios are needed for different aspects, often named with a special term like “stress test”, “endurance test”, or “scalability test”. “Load test” is the generic term for all of these tests.

What Is Performance Testing?

The term “performance test” is not always used in a uniform manner. It often describes a test to check the compliance of given load requirements by simulating the defined load and comparing the system’s behavior with the requirements regarding response time, throughput etc.

The term is also used if the test is meant to find bottlenecks without overloading the system.

Performance Areas

Performance tests can be executed against many targets hence it is important to align first on what performance areas exist for a typical commerce system.

Front End Performance

The front end is the part of the web that you can see and interact with. It mostly lives in a computer’s browser or in an app. It is mostly written in HTML, JavaScript, and CSS, spiced up with fonts and images and enriched with third-party services to provide additional functionality.

When talking about front end performance the technical performance and the perceived performance are to be distinguished. The technical performance measures the runtime of activities to build up a page or complete a user interaction. Some of the metrics are published by the browser as part of the Performance API, such as DomComplete and First Paint. The perceived performance on the other hand cannot be measured easily because it is the felt speed of the page. Or in other words, technical performance is a measure of how fast a site is while perceived performance is a measure of how fast a visitor thinks your site is.

Front ends can be app based as well, such as native apps for devices, and rather communicate using APIs with a backend. In this case, the front end performance is still shaped by responsiveness but cannot be measured with the means of browser metrics.

Back End Performance

The front end requires data and fetches that from back ends. The browser might just talk to one back end system, but this system in turn might contact other systems to fetch data from or process transactions.

However, increasingly the browser often talks to different backends at the same time. These reach from simple analytics tracking calls, to fetching dynamically created static content, such as resized images, to authentication providers authorizing the visitor.

But this is only the part that is obvious because it is needed to build the site and make it interactive. There are more back end activities that are often forgotten in the big picture. Data has to be imported from external sources, orders exported and transferred to other systems, and batch jobs clean up stale data or precalculate data that is needed later, which is too expensive to compute in real time.

By the way, API services are considered a backend and not a standalone type of area to be tested.

Tests and Areas

Server performance tests typically drive lot of load to stress the system and assess the behavior under these peak conditions. Client performance tests focus on the rendering and interaction performance on the client side (e.g. in browser) and don’t stress the servers much.

Server/Server-Side Performance Tests

When we group the activities by origin of the data, we can distinguish them by coming from the main origin, such as the initial host of the website or a third party domain, such as an analytics vendor.

As an example, your website is www.example.com. All data coming from www.example.com is main-origin. Data coming from images.example.com is an associated-origin. Data coming from google.com or liveperson.net is a third-party-origin.

You can also group this by the amount of influence you have on these origins. Because of the discussion we have here, you can probably heavily change the behavior of the main-origins either by code customization or uploading different content. On the other hand you probably cannot change anything on third-party-origin besides that you use it or you don't and maybe a few different parameters you send.

Client/Client-Side Performance Tests

Client performance is essentially an end-to-end view of the performance, including the full rendering in the browser, given that the performance test is web centric.

The end-to-end performance view is mainly driven by these factors:

- Back end performance (HTML and JSON creation and delivery), main-origin as well as third-party-origins
- Delivery performance for any static content, such as CSS, JavaScript, fonts, and images
- Execution speed of the JavaScript code, mainly defined by the quality of it as well as the browser type and computer system it is running on
- Complexity of the html
- CSS interpretation speed
- Rendering performance of the browser

Therefore the measurement won't load test the server or back ends, rather measure what the client will see and how it behaves.

Note: For some modern front-ends that use a lot of JavaScript for instance and handle state and logic in the client, it might be required to user client-side performance test concept when testing the server-side performance and applying a lot of load. This has test resource usage implications.

API Performance Testing

APIs in this document are seen as interfaces where machine readable information is produced rather than information that is directly served to a consumer such as HTML. In addition there are web services that provide information to other services and not directly to the end consumer device.

The modern web design right now transitions from serving server-side rendered information to pure machine readable information that will be rendered client-side. Therefore traditional metrics are not longer applicable easily.

If pure client-side rendering is used, API only metrics can be applied. If server-side rendering is the majority of the requests, the server-side metrics are suggested. A case by case decision has to be made if both technological concepts are used.

Recommendations

There are already recommendations on the market on how performance has to be shaped to satisfy the performance requirements of most users and create a positive business impact. It has to be distinguished between server-side, client-side, and API performance testing.

Please keep in mind that this document also includes error and stability information into the rating, hence this is more than just a performance goal, it is a behavioral goal for the implementation under test. Most criteria definition in existence just plainly ignore

Server-Side Performance

Based on our experience for most commerce platform to combine reachable and required goals to have a well performing and well converting implementation, Xceptance developed criteria that are listed in the Response Times Goals section of the server-side performance rating section.

We recommend a B rating or better to guarantee good performance for the consumer. A rating of A should be the long term goal of any organization to maximize customer satisfaction and business revenue.

Client-Side Performance

If the performance of the client side is important and not just the server performance to produce data, client-side metrics should be applied. The basic guidance is taken from the RAIL[5] guidelines:

“Optimize for fast loading performance relative to the device and network capabilities that your users use to access your site. Currently, a good target for first loads is to load the page and be interactive in 5 seconds or less …

For subsequent loads, a good target is to load the page in under 2 seconds.”

Find more information in the “Client-Side Performance section”. We also recommend B as the minimal rating.

API

When performance testing APIs, a concrete recommendation cannot be made because the purpose of the API and its functionality in part dictate what is possible and acceptable. Naturally the response time should be as low as possible, as stable as possible, and the response behavior should be error free.

The chapter API Performance Testing talks more in detail about goals and rating.

[5] RAIL: https://developers.google.com/web/fundamentals/performance/rail

Server Performance Tests

The Rating System

Introduction

The rating system uses a school like grade system to simplify the discussion and condense a lot of information into an easy to communicate results. Xceptance uses the grades of these three dimensions to compile a final grade:

- Response Times
- Errors
- Predictability

Response times symbolize the speed of the site, errors the overall stability, and predictability the consistency of response times and errors over time.

The worst grade of any dimension determines the overall grade.

Xceptance also provides a *Business Impact Value* based on errors and runtimes to make it easier to understand the results as well as relate the behavior to the business requirements. The predictability uses this value as one input.

Grades

These are the final grades assigned to a load and performance test result and symbolize a final easy to communicate verdict.

Grade	Description
A+	**Passed** without issues; exceptionally fast and stable response times
A	**Passed** with minor issues; fast and stable response times above average
B	**Passed** with minor issues; response times that matches the industry average
C	**Passed** with medium issues or response times slower than average; results are still acceptable, but improvements are recommended
D	**Failed** with major issues or request runtimes well below average; tuning, fixing, and validation runs are required
F	**Failed** with severe issues or unacceptable response times

The definition of issues stems from the dimensions used to compile this overall grade. Please refer to the Criteria: Errors section for more details.

Recommendation

The A+ and A grades should be the goals to go for. B can be seen as an industry average, while C still might be good enough but already places the result below the industry average. A business impact might be possible. With D and F, it is strongly discouraged to go live with the site or code version, and a business impact is very likely.

Note: B reflects the typical average but it does not stand for a best practise runtime goal, rather for a goal that is acceptable and comparable to others. If you strive for performance numbers better than your competition and a maximal conversion rate, you have to aim for A or better.

Criteria: Response Times

The first technical step for a good user impression and high responsiveness are fast responses. These are called request runtimes, response times, or just runtimes.

What is the Response Time?

A response time typically defines the time from initiating a request to a server until the full response has been returned to the caller. This includes DNS lookup, connection and TLS handshakes, sending of the request, waiting for the first byte, and downloading the response including processing of the header information. Often this will be referred to as TTLB - time to last byte.

Please keep in mind that server-side load testing does not measure page load times but rather communication times. Hence the measured time might later result in a physical page load or just in a page update. In case of mobile apps or API-first approaches, this is often just a status update.

Percentiles vs. Averages

Data representation with averages can be volatile and hide certain problems. Using percentiles[6] can improve the view and better reflect the later real user experience by more precisely reflecting what response time a certain amount of users will experience.

The influence of potential outliers is easier to judge by incorporating different percentiles. With averages it is easier to compare overall results, percentiles, on the other hand, allow to form specific (optimization) goals regarding particular performance problems.

Response Times Goals

When no other response time goals have been set as part of the test definition, the following data will be used to rate the overall outcome of the performance test.

The B rating symbolizes just about the average of most performance projects and might be a good indicator of the overall state of the market and to compare against it. Best practise response time recommendation on the other hand should match an A rating or better.

For all results, the easier reachable goal of P95 will be used. If one has more control over the infrastructure and the employed software components, we suggest to use the P99 values. Please keep in mind that when using P95, you leave about 5% of your visitors with response times worse than specified.

[6] Refer to the following link for a more detailed look at percentiles:
http://apmblog.dynatrace.com/2012/11/14/why-averages-suck-and-percentiles-are-great/

Goals and Associated Grades

The following table relates an interaction type to a response time and associates a grade with it. When there are several grades as the outcome, the worst (F) grade dictates the final response time grade.

Grade	Homepage		Catalog (Search, PLP, PDP, ...)		Cart/Account		Checkout		Place Order	
	P95	P99	P95	P99	P95	P99	P95	P99	P95	P99
A+	100	250	250	500	500	1,000	1,000	1,500	1,500	2,500
A	250	750	500	1,000	1,000	2,000	1,500	3,000	2,000	5,000
B	500	1,500	1,000	3,000	2,000	3,000	2,500	5,000	3,000	7,000
C	1,000	3,000	2,000	5,000	4,000	8,000	5,000	10,000	7,000	15,000
D	2,000	7,000	5,000	10,000	7,000	15,000	10,000	20,000	10,000	25,000
F	> 2,000	> 7,000	> 5,000	> 10,000	> 7,000	> 15,000	> 10,000	> 20,000	> 10,000	> 25,000

All times in milliseconds

Types of Pages or Interactions

The runtime goals are sorted by type of the page or interaction to differentiate them by the expected level of user patience. This also takes the presumed technical implementation into account, such as that checkout and place order are communicating to a third party system for data processing like address validation or payment gateways.

- **Homepage**: The entry page to the store. When the entry takes place via landing pages or advertised PLP or PDP pages, these numbers apply as well.
- **Catalog**: All pages that typically form the catalog of a store or are content pages. This includes search result pages and interactions, such as pagination and sorting.
- **Cart/Account**: Interactions with the cart, ranging from adding items to cart, removing items from cart or simply viewing the cart.
- **Checkout**: Any step in the checkout process.
- **Place Order**: The final order placement step, which typically includes payment system communication.

For pages that are outside this definition, depending on their basic functionality, they will be assigned to appropriate types. If required, custom criteria can be defined such as separating out landing pages from the homepage.

Criteria: Errors

What is an Error?

Two types of errors are distinguished, functional errors and technical errors. The outside view on these errors typically does not detail what was the cause neither if any error was actually caused by the performance test itself nor even indicate a load related failure.

Functional Errors

Functional errors which indicate that something unexpected happened, mostly on the logic side of things. Examples of functional error are incorrect pagination, checkouts that break because a certain product combinations do not sell together, failing checkouts due to not reaching the minimum order threshold or similar problems.

Even though it looks like a functional error, it can be a technical error on the server side and is not reflected correctly in the response code or the server handles the error and notifies the user (e.g. payment).

Technical Errors

Technical errors (failing requests) indicate that the response was either not received or the response code indicates a component problem. So a response code equal or greater 500 is a technical error, response codes such as 410 typically stand for functional or data issues (e.g. 400 Bad Request). Though this can also be a failure on the test side because the data sent was incorrectly formatted for instance.

Issues vs. Errors

Throughout this document, the terms issue and error are often used with the same meaning. But in many cases, the term error is used when the technical point of view is taken while the term issue is often used when communicating errors or related behavior.

Errors and Grades

The following tables list the grade and the requirements to achieve it for the general criteria *Errors*.

Grade	Short		Description	Examples
A+	No Issues	Visits	0.0% failed	no errors at all
		Clusters	None	
		Features	All working	
A	Minor Functional Issues	Visits	< 0.1% failed	a few missing products, an empty category
		Clusters	None	
		Features	No feature under test is broken	
B	Minor Issues	Visits	< 0.5% failed	a sorting rule is broken, a shipping method is broken

		Clusters	Transaction error rates are below 1% No recognizable repeating error patterns	
		Features	No main feature under test is broken Errors are sporadic	
C	Major Issues	Visits	< 5% failed	registration did not work for a longer time period; checkout was broken for some time
		Clusters	Transaction error rates exceeded 3% once A repeated error pattern might be visible	
		Features	Main features might be broken temporarily	
D	Blocking Issues	Visits	>= 5% failed	broken feature, such as but not limited to search, registration, checkout, or add-to-cart
		Clusters	Transaction error rates exceed 3% more than once Error patterns might be visible	
		Features	A main feature is broken for a longer period of time or entirely	
F	Fatal Errors	Visits	>= 15% failed	
		Clusters	Transaction error rates exceed 10%	
		Features	Test aborted due to too many errors, broken main features or complete unresponsiveness of the system under test	

Terminology

Visits

This document uses the common definition of a visit:

> *In web analytics, a session, or visit is a unit of measurement of a user's actions taken within a period of time or with regard to completion of a task. Sessions are also used in operational analytics and provision of user-specific recommendations. There are two primary methods used to define a session: time-oriented approaches based on continuity in user activity and navigation-based approaches based on continuity in a chain of requested pages.*[7]

For tests that don’t use visit in the classical sense such as API to API tests, a request is seen as a visit and the metrics will be applied in the same way.

Clusters

A cluster is a small time period were the same problem happens. This problem has not happened in this frequency before or after this period. It can be seen as a sudden behavior change.

Behavior changes typical indicate state changes of systems and hence often indicate error conditions.

Patterns

A pattern consists of recurring clusters of the same problem which are equidistantly distributed.

[7] https://en.wikipedia.org/wiki/Session_(web_analytics)

Features

Feature describe important functionality of the system under test. Typically features are the main drivers of business hence a broken feature will typically cause a drop in customer satisfaction and revenue.

Error Examples

Request Errors

- Response codes 5xx
- Read timeouts (runtime of 30 sec and more)
- Connection errors such as connection reset or SSL handshake failures

Validation Errors

- Failed registration, failed login, failed payment, failed order
- For the rating we will skip functional issues or false validation issues (e.g. unexpected page for the scripts but page is still valid, product set failed because scripts were not prepared for sets, no results after paging because we got sold out over time, missing prices or inventory for some products or similar etc.).

Response Codes 4xx

- Missing products, missing categories that are responded to with a code such as 410 or 404 for instance
- Missing pages or incorrectly configured redirects

Criteria: Predictability

In a perfect world, response times would be the same all day long. But reality shows that cache states, what is requested (a product with a large number of variations, for instance) and how many things are requested at once influence response times heavily. Stability is defined by the amount of responses which heavily deviate in runtime or show a permanent increase in runtime over time as well as patterns that indicate some other incorrect behavior.

Predictability is hard to measure. In a best case scenario, the performance curve would not contain spikes and no requests would fail. But in reality there is always some noise and it is important to identify patterns or trends in that noise.

Because the evaluation of noise is a complex process, Xceptance decided to retract to a metric anyone can grasp - the business impact on the end user of said noise.

Predictability and Business Impact

Predictability is a metric defined by the business impact on the end user. It uses the response times exceeding a certain criteria as well as technical errors (mainly non-recoverable errors such as response codes 500 or no response at all). It should give the merchant an idea how many visits might have been affected overall. Furthermore, affected means that a visit carries the risk of a loss in revenue or reputation.

The following formula represents the *Business Impact Value*. The business impact value is measured in percent.

$$\textit{Business Impact Value} \;=\; 100 \cdot \frac{ev + rv}{tv}$$

ev: Visits broken by response errors
rv: Visits with at least one request over 10 sec
tv: Total number of visits

10 seconds is chosen based on the user perception model published in Google RAIL Model[8]. This 10 second limit applies to a request in this metric and not to the overall page load. Hence, it is even softer than the value Google suggests.

For customers with a really low visit volume, a case by case decision has to be made, because 2% out of 1000 visits are 20 visits that can be easily affected, while the same 20 visits out of 10,000 visits are less of a business problem. But because it is a percentage of the user base that sees this subpar behavior, the business has to decide based on that number.

The load test report lists this value separately for the business to make the call.

A Business Impact Example

This example explains quickly what a typical Business Impact Value calculation might look like, and how to read it. In this document we will also assign a revenue to the Business Impact Value by assuming an average order value of $100.

[8] https://developers.google.com/web/fundamentals/performance/rail

Total Visits	**10,011**	A total of 10,011 visits were simulated during the load test. 55 of these visits saw a technical error, such as a response code 500 or higher or experienced a timeout or failing request of any kind.
Affected by Failure	55	
Affected by Response Time	98	98 visits saw a response time of 10 s or more.
Total Affected Visits	**153**	When real users browse the site and these response times happen or similar failures occur, users might or might not abandon their visit. Hence about 1.53% of visits are at risk which could lead to total revenue loss of $15,300.
Business Impact Value	**1.53%**	
Revenue at Risk	$15,300	

Please note, the Business Impact Value is calculated for the runtime of the entire test and not normalized.

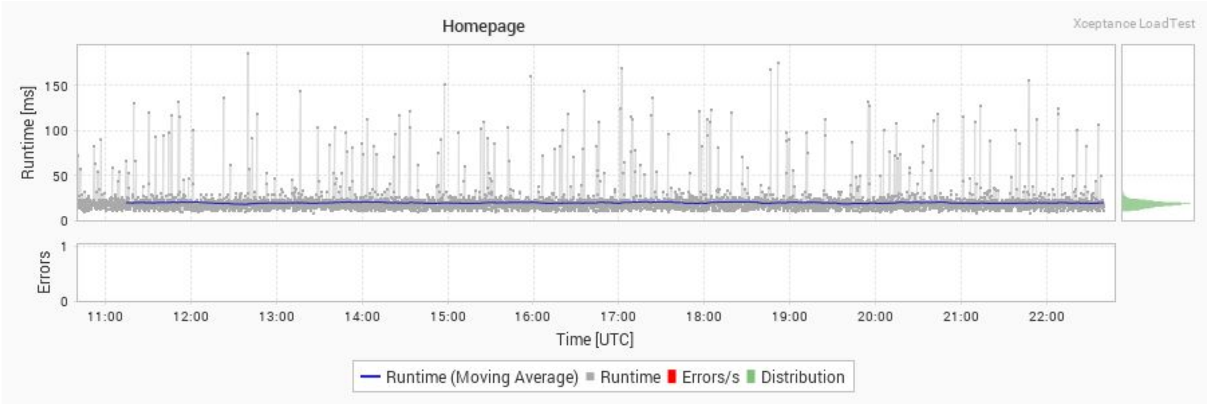
Predictability and Grades

The following tables list the grades for predictability and the criteria how they are determined.

Grade	Impact		Description
A+	None	Visits	0.00% affected
		Patterns	No response time patterns
A	Very Low	Visits	<0.1% of visits are affected
		Patterns	No patterns
B	Low	Visits	<0.5% of visits are affected
		Patterns	Minor response time patterns visible, but under 10 sec; No increase in response times over time except for order and registration
C	Noticeable	Visits	<1.0% are affected
		Patterns	Repeating patterns visible, but under 10 sec; No increase in response times except for order and registration
D	Large	Visits	<2.0% are affected
		Patterns	Timeout clusters occurred or response time patterns occurred and are over 10 sec; Sudden changes in response time outside cache expiration; Response times increase outside of order and registration
F	Heavy	Visits	>=2.0% are affected
		Patterns	Frequent sudden runtime changes; Timeout occurred in clusters or frequently; Strong increase of response times over test duration outside of order and registration

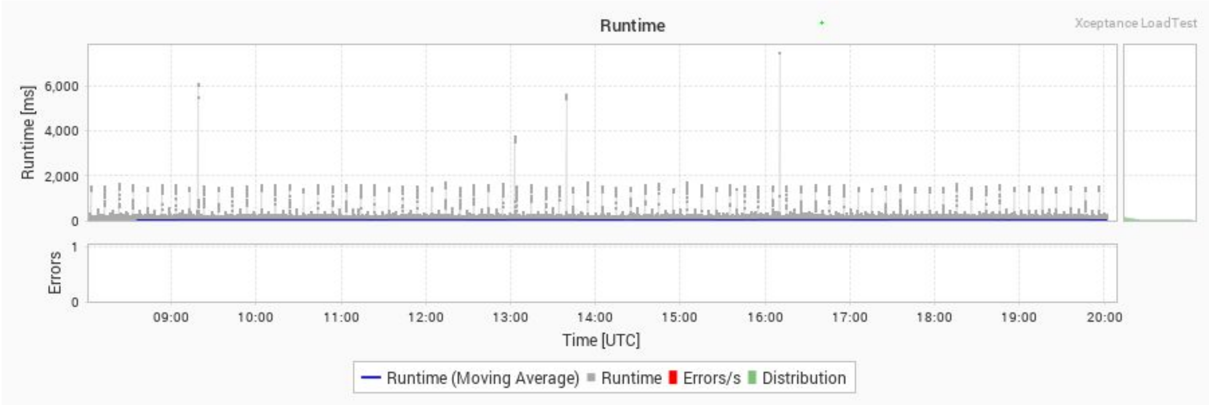
Examples

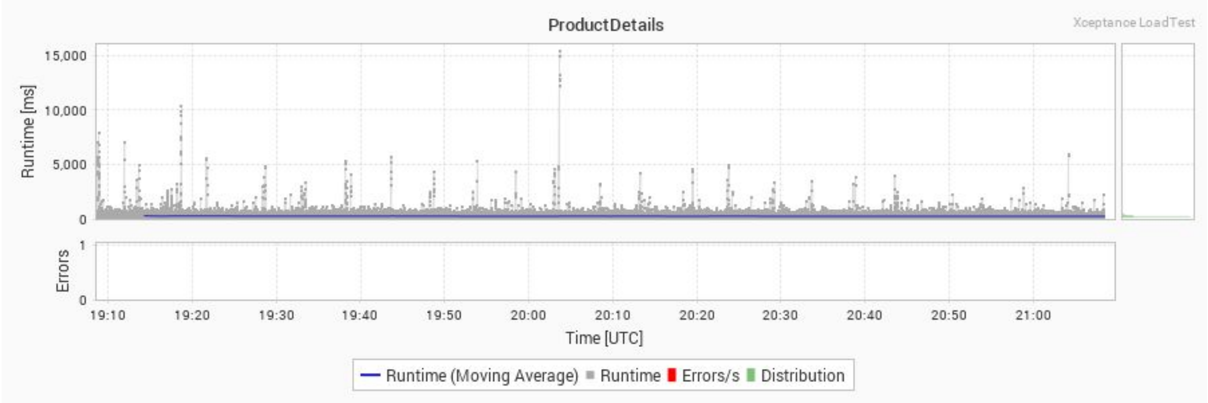
A Rating



Stable response times in a range of 25 to 200 ms, no clusters of high runtimes, no recurring patterns of events.

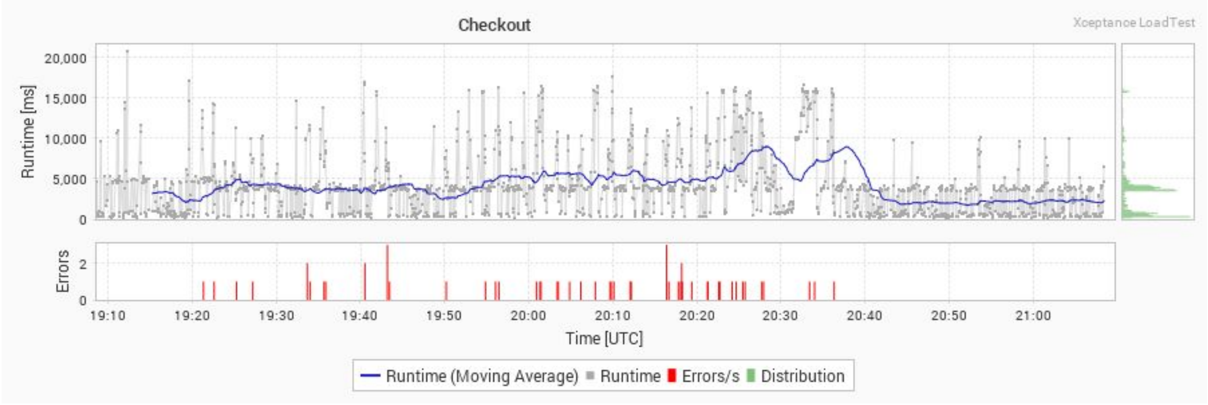
B / C Rating





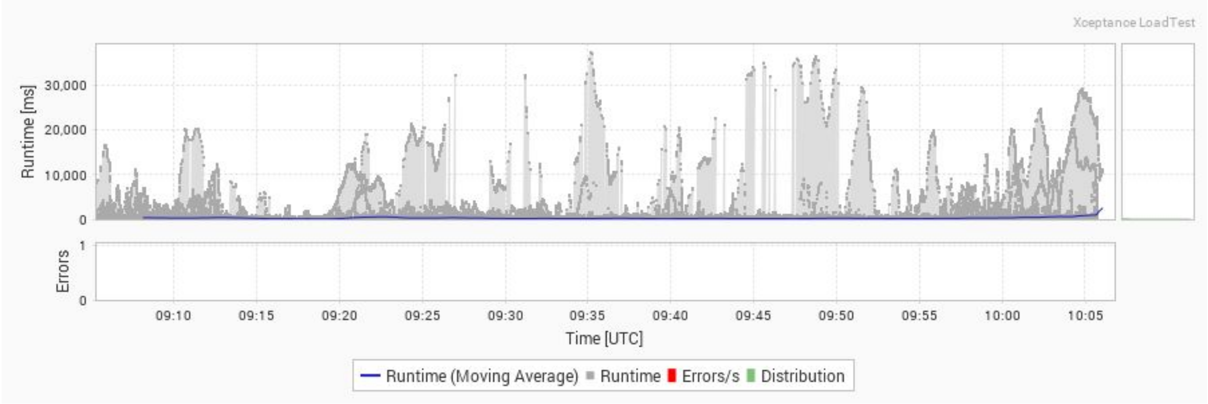
B rating: Recurring pattern of response time spikes every 5 min only on Product Detail Pages
C rating: Recurring pattern of response time spikes for a number of request types

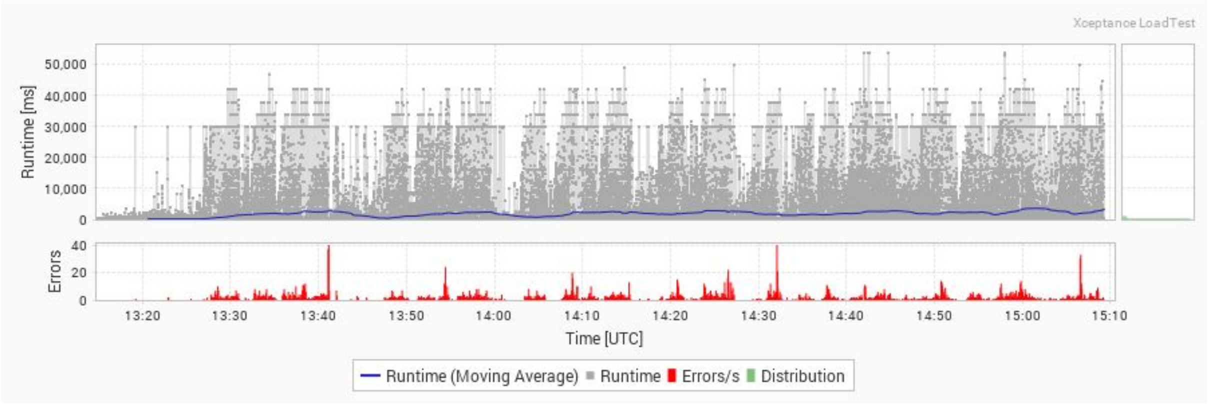
D Rating

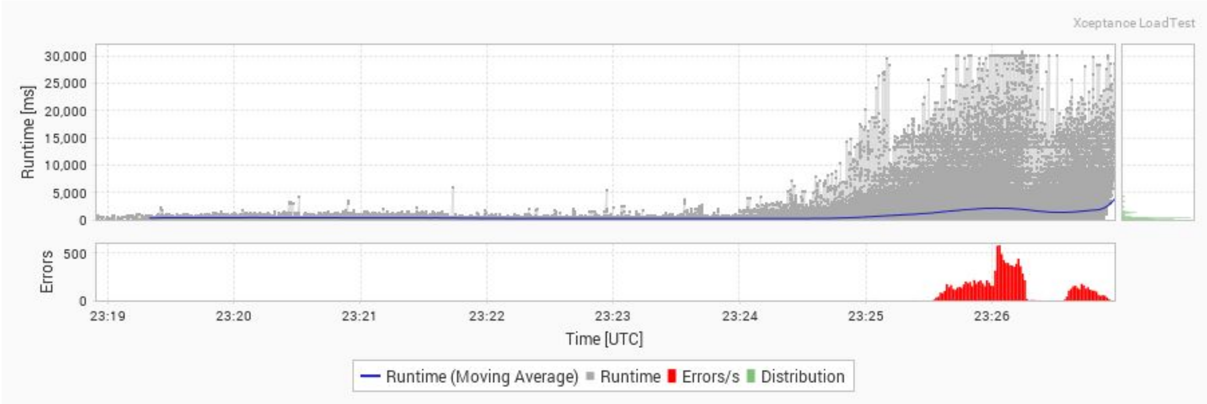


Sudden response time changes with high deviation and spikes over 10 s.

F Rating







Unpredictable runtime behavior with the chance of losing the entire site or a lot of functionality at the same time.

Executive Summary

Example

<table>
<tr><td><b>Quick Verdict</b></td><td colspan="2">Implementation shows average performance that aligns with target numbers and exhibits no functional errors under load.</td></tr>
<tr><td><b>Rating</b></td><td>A <b>B</b> C D F
A: Passed
B: Passed with minor issues
C: Passed, performance or stability subpar
D: Partially failed
F: Failed</td><td><table><tr><td>B</td><td>Response Times</td></tr><tr><td>A</td><td>Errors</td></tr><tr><td>B</td><td>Predictability</td></tr></table></td></tr>
<tr><td><b>Business Impact</b>
For entire test duration</td><td>0.20%
Potentially Lost Revenue
$19,500
Assuming an order value of $100</td><td><table><tr><td>Total Visits</td><td>96,981</td></tr><tr><td>Visits with Response Failures</td><td>70</td></tr><tr><td>Visits with Response Times 10 s or larger</td><td>125</td></tr><tr><td>Total Affected Visits</td><td>195</td></tr></table></td></tr>
<tr><td><b>Test Time</b></td><td>2017-03-15 from 07:10 UTC to 15:25 UTC</td><td><b>Duration</b> 8 h 15 min</td></tr>
<tr><td><b>Target Load</b></td><td colspan="2"><table><tr><th>Overall</th><th>Target</th><th>Tested</th></tr><tr><td><b>Visits / hour</b></td><td>95,000</td><td>96,981</td></tr><tr><td><b>Page Views / hour</b></td><td>500,000</td><td>521,751</td></tr><tr><td><b>Orders / hour</b></td><td>500</td><td>511</td></tr></table></td></tr>
<tr><td><b>Measurements</b></td><td colspan="2"><table><tr><th></th><th colspan="2">P95 (ms)</th><th colspan="2">P99 (ms)</th></tr><tr><th></th><th>Target
(B Rating)</th><th>Achieved</th><th>Target
(B Rating)</th><th>Achieved</th></tr><tr><td><b>Homepage/Landing Pages</b></td><td>500</td><td><b>459</b></td><td>1,500</td><td>1,250</td></tr><tr><td><b>Catalog/Search</b></td><td>1,000</td><td><b>780</b></td><td>3,000</td><td>2,583</td></tr><tr><td><b>Account/Cart</b></td><td>2,000</td><td><b>1,034</b></td><td>3,000</td><td>2,872</td></tr><tr><td><b>Checkout</b></td><td>2,500</td><td><b>2,041</b></td><td>5,000</td><td>4,098</td></tr><tr><td><b>Place Order</b></td><td>3,000</td><td><b>2,543</b></td><td>7,000</td><td>4,531</td></tr></table></td></tr>
<tr><td><b>System Under Test</b></td><td colspan="2">http://the.system.un.de/r/test</td></tr>
<tr><td><b>XLT Report</b></td><td colspan="2">https://report.server.domain/download/results/project/reportDir</td></tr>
</table>

Explained

This chapter explains the executive summary table which is presented as overview as part of the runtime reporting.

<table>
<tr><td><b>Quick Verdict</b></td><td colspan="3">A short description of the overall outcome and, when applicable, possible next steps. It is not a detailed assessment but rather a short summary of the entire test that might include a recommendation if applicable.</td></tr>
<tr><td><b>Rating</b></td><td colspan="3">The overall rating of the test including the individual category ratings, which the overall rating is based on.</td></tr>
<tr><td><b>Business Impact</b>
For entire test duration</td><td colspan="3">A quantification of the business impact of high response times and intermittent errors. In percentage of affected visits as well as a revenue value based on an average cart worth of $100.</td></tr>
<tr><td><b>Test Time</b></td><td>When was the test executed. This might include the ramp-up and shutdown phases if this is relevant for the result interpretation.</td><td><b>Duration</b></td><td>The duration of the test. This might include the ramp-up and shutdown phases if this is relevant for the result interpretation.</td></tr>
<tr><td><b>Target Load</b></td><td colspan="3">What target load was specified and what load target was achieved.</td></tr>
<tr><td><b>Measurements</b></td><td colspan="3">What response times have been measured in comparison to the set baseline for a B rating which corresponds to about an industry average.</td></tr>
<tr><td><b>System Under Test</b></td><td colspan="3">Which system has been tested.</td></tr>
<tr><td><b>XLT Report</b></td><td colspan="3">A link to the technical report of the load test tool XLT for more in-depth details.</td></tr>
</table>

Terminology

This section explains commonly used terms and abbreviations.

Term	Description
Action	A term used by XLT to mark a single user interaction (e.g. add to cart, view the homepage, submit a billing form) or an indirect action that should be measured separately (e.g. search suggestions). An indirect action is issued by the implementation based on user interaction, but the user does not necessarily know about or expects that response.
Session	Identical to a "visit" but often used as technical implementation term, while "visit" is used as a business term
Load Profile	Combination of user and load numbers including a time aspect
Test Run	The execution of a test with a given load profile
Test Scenario	A single test case that emulates a user visit and includes typical interactions
Test Script	The implementation of a test scenario
Transaction	A term used by XLT to express a single execution of a test scenario as part of a test run
User Journey	See Test Scenario
User Mix	Combination of users of certain scenarios
User Setup	See User Mix
Probability	Way of controlling user behavior and guaranteeing that important activities (interactions) are executed. This helps to reduce the number of required user scenarios and makes them more realistic overall. An example is a search scenario which contains a probability of 15% for search actions that yield no result.
PDP	Product Detail Page
PLP	Product Listing Page
Visit	A visit or session is defined as a series of page requests […] from the same uniquely identified client. A unique client is commonly identified by an IP address or a unique ID that is placed in the browser cookie. A visit is considered ended when no requests have been recorded in some number of elapsed minutes. [...] Note that a visit can consist of one page view, or thousands. […] (Wikipedia[9])
XLT	Xceptance LoadTest, performance and test automation tool https://www.xceptance.com/xlt/

[9] https://en.wikipedia.org/wiki/Web_analytics

About Xceptance

Xceptance is a quality assurance and testing company founded in 2004. We serve international companies from various business sectors, such as e-commerce, telecommunications, optical industries, and finance. Xceptance` clients are located in North America, Europe, and APAC.

We decided in 2007 to create our own dedicated load and performance test tool XLT to gain more flexibility, a cost advantage, increase scalability as well as we can fix and address findings through additional debugging or verification points more easily. In 2018, we made XLT available to all users free of charge and support the usage with optional training and support offers.

Contact

Load Test Team

Email	loadtest@xceptance.com

USA

Xceptance Software Technologies, Inc.
One Broadway, 14th Floor
Cambridge, MA 02142
United States

Email	contact@xceptance.com
Phone	+1-617-225-4335
WWW	https://www.xceptance.com

Germany

Xceptance Software Technologies GmbH
Leutragraben 2-4
07743 Jena
Germany

Email	kontakt@xceptance.de
Telefon	+49 3641 376300
WWW	https://www.xceptance.com/de/