



Performance Testing Checklist Packet

Everything you need to trigger thoughts, discussions and actions in the projects you are working on



Performance Testing Checklist:

People & Processes within Performance Testing

These checklists are designed to trigger thoughts, discussions and actions. They are not a learning tool, more they are designed to help you ask questions about the projects you are working on. We hope you can take this checklist concept and create adapted versions which will be more relevant to the projects you work on.

Considerations	Analysis	Resources
<ul style="list-style-type: none"><input type="checkbox"/> Why do you want to do performance testing?<input type="checkbox"/> What needs to be measured?<ul style="list-style-type: none"><input type="checkbox"/> Users<input type="checkbox"/> Volume<input type="checkbox"/> Response time<input type="checkbox"/> Throughput<input type="checkbox"/> Physical<input type="checkbox"/> Resource consumption: CPU, Memory, Disk I/O<input type="checkbox"/> What exactly are you performance testing?<input type="checkbox"/> What type of performance testing are you doing?<ul style="list-style-type: none"><input type="checkbox"/> Performance<input type="checkbox"/> Load<input type="checkbox"/> Stress<input type="checkbox"/> Is there room for exploratory performance testing?<input type="checkbox"/> How will tests be made to be realistic?	<ul style="list-style-type: none"><input type="checkbox"/> Does the software and performance testing data recover from after long periods, crashes, (un)scheduled downtime?<input type="checkbox"/> What failures do you anticipate?<input type="checkbox"/> How often will performance testing be performed? [sporadic, weekly, nightly]<input type="checkbox"/> Will you be benchmarking your performance against competitors, SLA's, previous versions or other standards?<input type="checkbox"/> What is the pass / fail / goals criteria for frame rate, load time, etc?<input type="checkbox"/> How does the software plan to change in the future and what effect will this have?	<ul style="list-style-type: none"><input type="checkbox"/> What budget is available?<input type="checkbox"/> Who is available to support the performance testing (including issues that may arise)?<input type="checkbox"/> How will issues be reported?<input type="checkbox"/> Who is responsible for replicating issues?<input type="checkbox"/> Who will maintain the performance tests?<input type="checkbox"/> How flexible are the tools being used?<input type="checkbox"/> How costly are the tools and load traffic?<input type="checkbox"/> How can costs be reduced?

Performance Testing Checklist:

Questions to Ask About Your Performance Test Data

Considering your performance testing data, have you thought about the following:

- Do you have a baseline of current performance to measure against?
- How much data can the test system hold?
- What are the accepted figures for peak and maximum users?
- How do user roles and hierarchies come into play?
- Does data need to be unique? (e.g. user ids)
- Does additional data need to be setup so a user can complete their user journey?
- Does additional data need to be generated to test future volumes and scalability?
- Does the application require interaction between users? Do they share data?
- How many times and how much data do you need to run the tests?
- Where will you get the data from? [Production / legacy / off the shelf data tool / bespoke]
- What's the acceptable time between transactions / clicks / actions?
- How and to what should future tests be compared against?
- Do you have access to (Google) analytics to identify high volume user journeys and load conditions?
- Check your ability to reset the database back to baseline state between each run?
- Do you have permissions in the test database to perform backups/restores?
- Do you have enough spare storage in the test database to have backups?

Performance Testing Checklists:

Tips for setting up a performance testing environment



Setup the performance test environment configuration as similar to production as possible:

- What tools and resources can be made available for performance testing?
- Load generation and coordination tools
- System and application monitoring tools
- Application and database profiling/diagnostic tools
- Service and network virtualization tools
- Results graphing and reporting tools
- Collaboration tools for chatting, file sharing, screen sharing
- Consider using the same tools in test as the other engineers (ops, db, dev, network)

What are your resources going to cost?

- Are the tools expensive? Consider the mix of open-source and commercial tools
- Are the tools supported? You need help when the tool isn't working right
- Hardware costs - servers, network, storage, devices, etc.
- System setup and configuration - who can help you setup and configure the system

Configure monitoring for the system under test:

- Physical Resources: CPU, DISK, MEMORY, NETWORK
- Application: .NET, JVM, Web Server, MQ/JMS, etc.
- Database: Connections, Blocking, Locking, Scans, Cache, Storage, I/O

Schedule a cross-functional team to support the process:

- Business Analysts
- System Administrators
- Database Administrators
- Functional Testers
- Database Developers
- Application Developers
- Network Engineers
- Managers

Performance Testing Checklists:

Tips for setting up a performance testing environment

Do all network components in production exist in test?	Is the architecture of the test site the same as production?	Check the size of the test environment when compared to production:	Are your test environment resources being used or shared by other processes?	Do you collect performance stats automatically?
<ul style="list-style-type: none"><input type="checkbox"/> If yes, then testing more accurately with load balancing and latency will be possible<input type="checkbox"/> If no, then your testing scope will be limited to only testing systems<input type="checkbox"/> You may discover network bottlenecks that limit your testing	<ul style="list-style-type: none"><input type="checkbox"/> Are the tools expensive? Consider the mix of open-source and commercial tools<input type="checkbox"/> Are the tools supported? You need help when the tool isn't working right<input type="checkbox"/> Hardware costs - servers, network, storage, devices, etc.<input type="checkbox"/> System setup and configuration - who can help you setup and configure the system	<ul style="list-style-type: none"><input type="checkbox"/> If the test systems are smaller, it is acceptable for tuning specific components<input type="checkbox"/> If the test systems are the same, it is acceptable for testing realistic loads<input type="checkbox"/> If the test systems are bigger, it is acceptable for testing future capacity and scale	<ul style="list-style-type: none"><input type="checkbox"/> Coordinate schedules to give you an exclusive time to run tests<input type="checkbox"/> Turn off any auto-scheduled jobs that might disrupt your testing (cron, batch, etc.)	<ul style="list-style-type: none"><input type="checkbox"/> Hardware, SO, dbms, virtual machines, runtime (java, .net, etc), application server and application itself (caches, queries, services stats, etc))

These checklists have been created as part of a crowdsourced community effort in collaboration with [Software Testing Club](#) & [SmartBear](#). We'd like to say a special thank you to everyone who contributed David Thompson, Thomas Lattner, Matias Reina, Dan Caseley, Simon Knight, Shane Dennis, Daniel Wiersma, Lindsey Thomas, Maciej Szatanik, Chris Gallant, Skip Marler, Sandeep Maher and Mark Tomlinson.