**Performance Testing Questionnaire**

1. **Testing Strategy**
   1. Please provide an overview on your application, and insight into your performance testing requirements.
   2. What type of performance testing should be performed?
      1. Load test – simulate typical user traffic (load) of the system
      2. Stress test – pushing system beyond typical load to determine when/how the system breaks
      3. Duration test – long running test to determine system performance over time
      4. Component test – testing a specific functionality of the system
2. **Application Architecture** 
   1. **Please provide architecture diagram**
   2. What application is being performance tested?
      1. What are the hardware specifications?
         1. What is the current server (web, application, database) configuration? (capacity, number of processes, number of threads, etc.)
      2. What software is being used?
   3. What type of application is being performance tested? (web application, mobile application, web service, etc.)
   4. What is the URL of the web service or web application being performance tested?
   5. Is the application externally facing?
      1. Does the application need firewall rules to access?
   6. Is the application stable and functionally tested?
   7. What other systems does this application interface with? (database, MQ, ESB, web services, etc.)
   8. Are there any calls to external services that should not be included in testing? (i.e. Western Union, eADMS)
      1. Do these external calls need to be stubbed or mocked?
   9. What environment is being utilized for testing? (QA, Staging, Perf, New Environment)
      1. Will this environment be shared across multiple teams?
      2. If so, what is your approach for environment scheduling / sharing across teams?
   10. Is the application already in production?
       1. For applications currently in development (not yet in PROD) –
          1. Has functional testing been completed?
          2. How will performance testing fit into your development cycle?
       2. For applications currently in production -
          1. Does this test environment infrastructure mirror production?
             1. If no, what are the differences between the test environment and production?
          2. Are there any known performance issues in production?
          3. Can you provide production volume metrics for each functionality being tested?
3. **Nonfunctional Requirements (NFRs) /Service Level Objectives (SLOs)**
   1. Have you established a target for maximum concurrent users that should be reached for each script/test scenario?
   2. Have you established any standard expectations for transaction response times? (i.e. 3 seconds)
   3. Have you established an accepted threshold for percentage error rate?
4. **Test Cases**
   1. What types of users are accessing the application? (external users, employees, customers, etc.)
   2. What (if any) authentication is required for users?
   3. What are the key scenarios that users are performing that will be performance tested? (e.g. log in, report outage, pay bill, etc.)
      1. Are there any pieces of the key scenarios that will be considered out-of-scope? (e.g. hitting the pay bill page is in-scope, but paying the bill is out-of-scope)
5. **Test Volume Metrics**
6. Can you provide production volume metrics for each functionality being tested?
7. Based on production volume, do you have target user volumes per hour for each script/test scenario.
8. What is the traffic trend of users that access the application? (e.g. all at once, certain numbers during peak hours, sporadically, etc.)
9. **Test Data Requirements**
   1. What test data is needed for performance testing?
      1. What are the specific requirements for test data? (e.g. logging in, users that have not reported outages, users that have not paid their bills, etc.)
   2. How will the team be providing the data to the performance engineering team? (i.e. CSV, Excel, etc.)
   3. Can data be reused for each test, or will it be newly generated for each test?
   4. Are there any anticipated changes to the data structure being utilized for testing?
10. **Performance Testing Tools**
    1. NOTE: The current tooling set includes SOASTA for web application scripting, LoadUI/SoapUI for web service scripting, and ServiceV Pro for virtual services.
11. **Monitoring Approach/APM Tools**
    1. What native monitoring tools are currently in place? (e.g. tools, log files, etc.)
12. **Testing Schedule**
    1. Is there a timeline/schedule that performance testing needs to align to?
    2. Is there a specific time window for running performance tests? (e.g. after business hours, etc.)
13. **Defect Management Approach**
    1. Who should be the point of contact on your team for any defects/issues found? (NOTE: It would be beneficial for this person to have access to server logs to analyze any defects/issues)