Sakai Community Performance Testing Plan

The following is a proposed plan to create the base for a shared performance testing environment. The initial assumptions are that thorough performance testing of the Sakai code will improve overall quality of the Sakai "product" and that capacity testing of a local environment is very different from quality testing of the code base. Given these assumptions, the plan that follows focuses on designing a system for performance testing that can be shared centrally (ideally like our current functional QA servers, on hardware provided by a few campuses throughout our community), contributed to organizationally, and managed at the Foundation level. Michigan has volunteered to take the lead on this initiative, but by no means is working alone. All organizations interested in the pursuit of improved performance testing are encouraged to participate in this cause at any and all levels according to their ability and resources.

- **Goal**
  - Create a QA environment that enables performance testing by the community in a shared manner
  - Have working proof of concept by Paris Sakai Conference (6 months)
- **Evaluation of performance testing tools**
  - Call for community input on performance testing tools with goal of selecting best open source alternative or a community license purchased by the Sakai foundation.
  - Michigan evaluating WebLOAD open source load testing project, which appears to be a feature rich product akin to HP LoadRunner
- **Infrastructure**
  - Start by relying on the U-Michigan hardware infrastructure (leveraging the size of our investment)
  - Any campus with an infrastructure to contribute to community performance testing is encouraged to build on the same model
  - Domain name
    - Request load1-us.sakaiproject.org for Michigan system
    - Build on DN scheme as campuses contribute infrastructure to community testing
    - Follows model set by current QA servers
  - Sakai build
    - Work with Ian Boston and the CARET team to design auto-loading QA builds, and for running an initial base set of tests that the build works as advertised
    - To be used by all QA servers.
  - Data provisioning
    - Call for community input on data provisioning tools
    - Work with Alan Berg (and others) to produce provisioning scripts enabling auto-populating environment
      - Users
      - Sites
    - Need ability to create data files to parameterize load test scripts predictably based on auto-populated data
  - **Data Environment Design**
    - Call for use cases from community to determine varied needs for load testing scenarios covering tools critical for performance tests
    - Work with Stephan Marquard, Chris Kretler (and others) to design base environment (numbers of sites, numbers of participants, etc.) to fit known use cases
    - Allow for growth of environment in realistic ways
  - **Test scripts**
    - Create needed test scripts in selected tool
      - May start with Michigan LoadRunner only to meet target deadline
      - Expand to open source scripts as developers are available to contribute
    - Call for committed developers to create needed scripts with tool, if selected tool depends on coding (like jmeter and grinder)
  - **Test scenarios**
    - Create testing scenarios to address baseline stress tests of individual tools
    - Create scenarios for related tool clusters
    - Identify combination of "core" tools to become baseline load scenario
  - **Shared working space**
    - Develop shared working spaces in Confluence (and elsewhere) for sharing information on environment, testing models, scripts, results, etc.