Task	*Task Detail*
A. Database	
Schema	Design schema to hold all information which we want to store for.
Scripts and creation	SQL scripts to create the database, generate users, tables, indexes, and data.
Generate Test Data	Test data would include populating all tables as examples and so that system testing can be done.
B. Framework	
General Tools/Modules	Generic modules for database access, SOAP calls, logging and configuration.
SOAP Daemon	Find or write a daemon which has graceful failures(error checking), signals for administration (shutdown, re-load etc). (brt_server.pl.)
Forking server	Server forks for a new SOAP request. It would be nice if it could hold a connection to a database between queries (via a cookie) and timeout.
Password Authentication	If we want to run off site we need to be sure the correct users are accessing.
Admin group privileges	There are some tasks that a user must be an admin to do.
RPC modules	All DB access is activated through SOAP.
Client/Integrator	The Client scripts will handle DB information retrievals and updates(state, results) from SUT. (Other functionality is listed in the Client section below.)
Admin/User	We need to set up hosts, BOM's, tests, sub_tests, binaries manually. I want interactive admin scripts to do this rather than direct SQL.
Report	These would be scripts to access reports of the results.
Data Receive	The data should be sent in encypted and signed by the user using GPG. We will unpack, verify and update database.
Command Line Admin tools	Because It will be necessary for an admin to define complicated objects, such as a whole BOM or BRT, it is necessary to have tools to do this. Command line is simpler, faster than Web.
Test Client	This script communicates through the SOAP server, top get test info or update status or results.
User tools	Tools to enable viewing results. These could be dynamic, or perhaps may be through report generation.
PLM to run with configs/instances	Make plm able to run two instance on same host for in tests system. Also make sure supervisor can run on Gentoo. It needs an e-build and testing.
PLM as Gating Factor	Add syncing or querying mechanism which will prevent packages which do not pass prelimiary filters from being part of a BOM.
Portage Integration	Information needs to get from the database to a format used by the portage installer.
Packaging for distribution(rpm)	Scripts to package the application for installation on the server side, (an rpm for Suse?) And on the client (an ebuild) must be created.
System Testing	For the 'Lite' version: Set up complete database and set-up tests and store potentially meaningful results. There may be some manual steps, but none should be too painful and the whole cycle must be complete.

C. Client	**Initially we are working toward BRT-Lite**
Define BRT-Lite	This is a system that will run with some manual intervention, but will produce standard BRT results. A complete working definition is needed.
Research build systems	Since we want to build from source, It seems as though Gentoo would be a good option. Research it' s capabilities and determine whether it will do what we want.
Document Initial Setup	Setup required to get host ready for OS installation.
Define Host specifics('Mgr' boot partitions ...)	This would be things specific to the host, kconfig options, special lilo or environment settings, partitions, boot partition, devices.
Register host specifics	Be able to record all host specifics in DB.
Boot params, setup	Use the fix_lilo.pl if it works. Or make something else work.
Boot partition change scripts	For a dual boot system, have manager partition set-up and bot other partition.
Secondary install scripts	
Client script (brt_client.pl)	This is the script that will automate as much as possible the system set-up, system check, OS installation, tests execution, database queries and updates. If the system is an integration host, mark Bom as passed.
Automate 'Test' partition Install	log RESULT-DETAIL: $WARN_COUNT warnings, $ERROR_COUNT errors
Automate 'Test' Boot, networking, init scripts, etc.	
Automate Dep package Install	
Test Packaging Rules	Tests must be automated in set-up and execution. They must error check and give appropriate return values. Configuring and logging must be consistant.
Tools and API' s	Adapt the STP tool set to work for BRT. Ideally it should work in STP still too. Disk_handler, environment set up, console notes, logging, etc
Standard filenames, locations	Locations for log files. Configuration files, test results, file names, etc
Executing Tests	
wrap.sh-for BRT	Define and implement the main wrap.sh function for brt. This should execute the sub-tests which will have their own wrap.sh.
Get test/binary Info	Define test conventions, should be as close to STP conventions as possible.- Pass location ofd be configurable.tests and binaries. Where to get test from should be configurable.
Automate test/binary Install	Define test conventions, should be as close to STP conventions as possible. Binaries must be defined too.
Run tests	Define test conventions, should be as close to STP conventions as possible.

Cleanup Scripts	Cleanup if possible should be generic through standard directory and file names, but each test may have some custom clean-up. Host should be completely ready for set-up of another test. This may not work like STP, which only runs one test. Example items: sync disks, unmount file-systems, move files, logs, make sure system is quiet, with no leftover processes.
Upload results, data	Data and system information upload for each sub-test. After each test would be good to prevent data loss.
Recovery Scripts (from ' Mgr)	Recovery for if system stops working or reboots unexpectedly. Rebot on stable kernel and send data results.
System Classification Tool	This is a systems check to see if the system really looks like what we are expecting when the tests is about to be run.
Packaging for external use	Make sure someone at an external site can get all the documentation and source and information and access necessary to run BRT-Lite.

D. Test Instances

Choose Basic Acceptance Tests (BAT)	Basic acceptance tests which will be run in the integration host.
Set up BAT' s	Some of the initial tests will probably tests we are already running in STP. They will need to be examined and testsed for any changes that might be necessary.
Set up initial Tests	Beyond the BAT' s
Results	Define what results and system information to keep and where to store it.
Contact ISV's	Getting ISV' s involved and running their software/tests on our system is a major goal. This needs coherent planning, communication and follow-up.

E. Documentation

Use Cases	Cover uses by ISV' s, Distributions and Users.
Design Doc	
Base Install	Choose packages to install and set up an initial image. Which will be quick and easy to regenerate. Record this information for each test and so that we can build new systems on top of it.
Package Deltas	Define good way to track package deltas.
Portage	Document
Manual (sxw format)	Document
Overview	Document
Installation	Document
Framework	Document
SUT	Document
Administration	Document
Packaging Tests	Document
Internals	Document