1. 开发人员是否提交了测试申请？

2. 测试对象是否已经明确？

3. 测试范围是否已经明确？

4. 本次不被测试的范围是否已经明确？

5. 测试目标是否已经明确？

6. 何时开始性能测试？

7. 何时终止一轮性能测试？

8. 性能测试需要做几轮？

9. 所需的测试环境是什么？是否已经到位并配置完成？（包括硬件、软件、网络等）

10.所需的测试工具是什么？是否已经到位并保证可以正常使用？

11.被测系统的版本是否已经明确？是否已发布？从哪里可以获得？是否已经部署成功？

12.被测系统的相关功能是否已经正确实现？

13.压力点是否已经明确？响应时间的计算方式是否已经明确？

14.本次测试工作参考的文档有哪些？

15.场景是否已经设计完成并记录在场景管理文档中？

16.每个场景是否有明确的测试意图、前置条件和详细的设置？

17.脚本是否已经录制并调试通过？

18.是否已经明确了哪些地方需要参数化？

19.是否已经明确了各个参数的取值方式？

20.是否已经为参数化的部分准备了必须的数据？

21.是否已经准备了相应历史数据量？

22.是否已经准备了相应的数据恢复方法？（例如准备一个SQL语句用来恢复数据环境）

23.在Controller中对多VU、多次迭代的情况是否已经调试通过？

24.在Controller中Result的路径设置是否正确？

25.在Controller中检查脚本选择是否正确？

26.在Controller中检查VU数量设置是否正确？

27.在Controller中检查集合点是否禁用/启用?

28.在Controller中检查VU加载策略是否设置正确？

29.在Controller中检查迭代次数是否设置正确？

30.在Controller中检查迭代间隔设置是否正确？

31.在Controller中检查日志是否禁用/启用？

32.在Controller中检查Think\_Time是否回放？

33.在Controller中检查是否为UNIX服务器和Load Generator机添加了资源监视器并确认可以收到性能数据？

34.在Controller中检查是否为其他必要的资源添加了资源监视器，并确认可以收到性能数据（例如Oracle，WebSphere）？

35.在Controller中检查Load Generator机是否可以连上？

36.检查场景管理文档中是否添加了新的“场景执行情况”，并记录了运行前的数据？

37.在Controller中执行场景前，检查是否在Linux客户端中运行了vmstat和top，监视执行过程中的Linux服务器资源消耗情况？

38.在Controller中执行场景完毕后是否马上去系统中进行检查数据的一致性，并填写“场景执行情况”中的运行后情况？

39.场景完执行完后是否将vmstat和top的数据copy到记事本中，并保存到相应的结果目录下？

40.整个系统的测试工作执行完毕后，是否进行了性能图表的分析和测试报告的提交？

————————————————

版权声明：本文为CSDN博主「jackei」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/jackei/article/details/46094981. 开发人员是否提交了测试申请？

2. 测试对象是否已经明确？

3. 测试范围是否已经明确？

4. 本次不被测试的范围是否已经明确？

5. 测试目标是否已经明确？

6. 何时开始性能测试？

7. 何时终止一轮性能测试？

8. 性能测试需要做几轮？

9. 所需的测试环境是什么？是否已经到位并配置完成？（包括硬件、软件、网络等）

10.所需的测试工具是什么？是否已经到位并保证可以正常使用？

11.被测系统的版本是否已经明确？是否已发布？从哪里可以获得？是否已经部署成功？

12.被测系统的相关功能是否已经正确实现？

13.压力点是否已经明确？响应时间的计算方式是否已经明确？

14.本次测试工作参考的文档有哪些？

15.场景是否已经设计完成并记录在场景管理文档中？

16.每个场景是否有明确的测试意图、前置条件和详细的设置？

17.脚本是否已经录制并调试通过？

18.是否已经明确了哪些地方需要参数化？

19.是否已经明确了各个参数的取值方式？

20.是否已经为参数化的部分准备了必须的数据？

21.是否已经准备了相应历史数据量？

22.是否已经准备了相应的数据恢复方法？（例如准备一个SQL语句用来恢复数据环境）

23.在Controller中对多VU、多次迭代的情况是否已经调试通过？

24.在Controller中Result的路径设置是否正确？

25.在Controller中检查脚本选择是否正确？

26.在Controller中检查VU数量设置是否正确？

27.在Controller中检查集合点是否禁用/启用?

28.在Controller中检查VU加载策略是否设置正确？

29.在Controller中检查迭代次数是否设置正确？

30.在Controller中检查迭代间隔设置是否正确？

31.在Controller中检查日志是否禁用/启用？

32.在Controller中检查Think\_Time是否回放？

33.在Controller中检查是否为UNIX服务器和Load Generator机添加了资源监视器并确认可以收到性能数据？

34.在Controller中检查是否为其他必要的资源添加了资源监视器，并确认可以收到性能数据（例如Oracle，WebSphere）？

35.在Controller中检查Load Generator机是否可以连上？

36.检查场景管理文档中是否添加了新的“场景执行情况”，并记录了运行前的数据？

37.在Controller中执行场景前，检查是否在Linux客户端中运行了vmstat和top，监视执行过程中的Linux服务器资源消耗情况？

38.在Controller中执行场景完毕后是否马上去系统中进行检查数据的一致性，并填写“场景执行情况”中的运行后情况？

39.场景完执行完后是否将vmstat和top的数据copy到记事本中，并保存到相应的结果目录下？

40.整个系统的测试工作执行完毕后，是否进行了性能图表的分析和测试报告的提交？

————————————————

版权声明：本文为CSDN博主「jackei」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/jackei/article/details/46094981. 开发人员是否提交了测试申请？

2. 测试对象是否已经明确？

3. 测试范围是否已经明确？

4. 本次不被测试的范围是否已经明确？

5. 测试目标是否已经明确？

6. 何时开始性能测试？

7. 何时终止一轮性能测试？

8. 性能测试需要做几轮？

9. 所需的测试环境是什么？是否已经到位并配置完成？（包括硬件、软件、网络等）

10.所需的测试工具是什么？是否已经到位并保证可以正常使用？

11.被测系统的版本是否已经明确？是否已发布？从哪里可以获得？是否已经部署成功？

12.被测系统的相关功能是否已经正确实现？

13.压力点是否已经明确？响应时间的计算方式是否已经明确？

14.本次测试工作参考的文档有哪些？

15.场景是否已经设计完成并记录在场景管理文档中？

16.每个场景是否有明确的测试意图、前置条件和详细的设置？

17.脚本是否已经录制并调试通过？

18.是否已经明确了哪些地方需要参数化？

19.是否已经明确了各个参数的取值方式？

20.是否已经为参数化的部分准备了必须的数据？

21.是否已经准备了相应历史数据量？

22.是否已经准备了相应的数据恢复方法？（例如准备一个SQL语句用来恢复数据环境）

23.在Controller中对多VU、多次迭代的情况是否已经调试通过？

24.在Controller中Result的路径设置是否正确？

25.在Controller中检查脚本选择是否正确？

26.在Controller中检查VU数量设置是否正确？

27.在Controller中检查集合点是否禁用/启用?

28.在Controller中检查VU加载策略是否设置正确？

29.在Controller中检查迭代次数是否设置正确？

30.在Controller中检查迭代间隔设置是否正确？

31.在Controller中检查日志是否禁用/启用？

32.在Controller中检查Think\_Time是否回放？

33.在Controller中检查是否为UNIX服务器和Load Generator机添加了资源监视器并确认可以收到性能数据？

34.在Controller中检查是否为其他必要的资源添加了资源监视器，并确认可以收到性能数据（例如Oracle，WebSphere）？

35.在Controller中检查Load Generator机是否可以连上？

36.检查场景管理文档中是否添加了新的“场景执行情况”，并记录了运行前的数据？

37.在Controller中执行场景前，检查是否在Linux客户端中运行了vmstat和top，监视执行过程中的Linux服务器资源消耗情况？

38.在Controller中执行场景完毕后是否马上去系统中进行检查数据的一致性，并填写“场景执行情况”中的运行后情况？

39.场景完执行完后是否将vmstat和top的数据copy到记事本中，并保存到相应的结果目录下？

40.整个系统的测试工作执行完毕后，是否进行了性能图表的分析和测试报告的提交？

————————————————

版权声明：本文为CSDN博主「jackei」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/jackei/article/details/46094981. 开发人员是否提交了测试申请？

2. 测试对象是否已经明确？

3. 测试范围是否已经明确？

4. 本次不被测试的范围是否已经明确？

5. 测试目标是否已经明确？

6. 何时开始性能测试？

7. 何时终止一轮性能测试？

8. 性能测试需要做几轮？

9. 所需的测试环境是什么？是否已经到位并配置完成？（包括硬件、软件、网络等）

10.所需的测试工具是什么？是否已经到位并保证可以正常使用？

11.被测系统的版本是否已经明确？是否已发布？从哪里可以获得？是否已经部署成功？

12.被测系统的相关功能是否已经正确实现？

13.压力点是否已经明确？响应时间的计算方式是否已经明确？

14.本次测试工作参考的文档有哪些？

15.场景是否已经设计完成并记录在场景管理文档中？

16.每个场景是否有明确的测试意图、前置条件和详细的设置？

17.脚本是否已经录制并调试通过？

18.是否已经明确了哪些地方需要参数化？

19.是否已经明确了各个参数的取值方式？

20.是否已经为参数化的部分准备了必须的数据？

21.是否已经准备了相应历史数据量？

22.是否已经准备了相应的数据恢复方法？（例如准备一个SQL语句用来恢复数据环境）

23.在Controller中对多VU、多次迭代的情况是否已经调试通过？

24.在Controller中Result的路径设置是否正确？

25.在Controller中检查脚本选择是否正确？

26.在Controller中检查VU数量设置是否正确？

27.在Controller中检查集合点是否禁用/启用?

28.在Controller中检查VU加载策略是否设置正确？

29.在Controller中检查迭代次数是否设置正确？

30.在Controller中检查迭代间隔设置是否正确？

31.在Controller中检查日志是否禁用/启用？

32.在Controller中检查Think\_Time是否回放？

33.在Controller中检查是否为UNIX服务器和Load Generator机添加了资源监视器并确认可以收到性能数据？

34.在Controller中检查是否为其他必要的资源添加了资源监视器，并确认可以收到性能数据（例如Oracle，WebSphere）？

35.在Controller中检查Load Generator机是否可以连上？

36.检查场景管理文档中是否添加了新的“场景执行情况”，并记录了运行前的数据？

37.在Controller中执行场景前，检查是否在Linux客户端中运行了vmstat和top，监视执行过程中的Linux服务器资源消耗情况？

38.在Controller中执行场景完毕后是否马上去系统中进行检查数据的一致性，并填写“场景执行情况”中的运行后情况？

39.场景完执行完后是否将vmstat和top的数据copy到记事本中，并保存到相应的结果目录下？

40.整个系统的测试工作执行完毕后，是否进行了性能图表的分析和测试报告的提交？

————————————————

版权声明：本文为CSDN博主「jackei」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/jackei/article/details/46094981. 开发人员是否提交了测试申请？

2. 测试对象是否已经明确？

3. 测试范围是否已经明确？

4. 本次不被测试的范围是否已经明确？

5. 测试目标是否已经明确？

6. 何时开始性能测试？

7. 何时终止一轮性能测试？

8. 性能测试需要做几轮？

9. 所需的测试环境是什么？是否已经到位并配置完成？（包括硬件、软件、网络等）

10.所需的测试工具是什么？是否已经到位并保证可以正常使用？

11.被测系统的版本是否已经明确？是否已发布？从哪里可以获得？是否已经部署成功？

12.被测系统的相关功能是否已经正确实现？

13.压力点是否已经明确？响应时间的计算方式是否已经明确？

14.本次测试工作参考的文档有哪些？

15.场景是否已经设计完成并记录在场景管理文档中？

16.每个场景是否有明确的测试意图、前置条件和详细的设置？

17.脚本是否已经录制并调试通过？

18.是否已经明确了哪些地方需要参数化？

19.是否已经明确了各个参数的取值方式？

20.是否已经为参数化的部分准备了必须的数据？

21.是否已经准备了相应历史数据量？

22.是否已经准备了相应的数据恢复方法？（例如准备一个SQL语句用来恢复数据环境）

23.在Controller中对多VU、多次迭代的情况是否已经调试通过？

24.在Controller中Result的路径设置是否正确？

25.在Controller中检查脚本选择是否正确？

26.在Controller中检查VU数量设置是否正确？

27.在Controller中检查集合点是否禁用/启用?

28.在Controller中检查VU加载策略是否设置正确？

29.在Controller中检查迭代次数是否设置正确？

30.在Controller中检查迭代间隔设置是否正确？

31.在Controller中检查日志是否禁用/启用？

32.在Controller中检查Think\_Time是否回放？

33.在Controller中检查是否为UNIX服务器和Load Generator机添加了资源监视器并确认可以收到性能数据？

34.在Controller中检查是否为其他必要的资源添加了资源监视器，并确认可以收到性能数据（例如Oracle，WebSphere）？

35.在Controller中检查Load Generator机是否可以连上？

36.检查场景管理文档中是否添加了新的“场景执行情况”，并记录了运行前的数据？

37.在Controller中执行场景前，检查是否在Linux客户端中运行了vmstat和top，监视执行过程中的Linux服务器资源消耗情况？

38.在Controller中执行场景完毕后是否马上去系统中进行检查数据的一致性，并填写“场景执行情况”中的运行后情况？

39.场景完执行完后是否将vmstat和top的数据copy到记事本中，并保存到相应的结果目录下？

40.整个系统的测试工作执行完毕后，是否进行了性能图表的分析和测试报告的提交？