你需要知道的 BVT(版本验证测试)

What you need to know about BVT (Build Verification Testing)

原作者：Vijay Kumar
译者：Keith,落日沉鱼

原文地址：
http://www.softwaretestinghelp.com/bvt-build-verification-testing-process/

日期：2009-12-22

【E 测中国翻译团队作品】

什么是 BVT?

版本验证测试是运行在每一个新版本上，用来验证该版本在发布给测试团队进行更深入的测试之前是否可测的测试集。这些测试用例关注核心功能，确保应用程序稳定、能够被彻底的测试。典型情况下，BVT 是自动运行的。如果 BVT 失败，版本会再一次分配到开发人员那里进行修正。

BVT 也被称为冒烟测试或是版本验收测试（BAT）

对于新版本主要检查两件事情

- 版本确认
- 版本验收

BVT 的一些基本原理：

- BVT 是验证主要功能测试的子集。
- BVT 一般在每日 build 上运行，如果 BVT 失败，build 会被拒绝并且一个新的 build 会在修复完成后发布。
- BVT 的优势是当应用软件主要功能有问题时，它节省了测试人员搭建和测试 build 所需要的大量精力。
- 仔细设计 BVT 使其覆盖基本的功能点。
- 一般来说 BVT 运行不应该超过 30 分钟。
- BVT 是一种回归测试，在每一个新 build 上执行。

BVT 主要检查项目的完整性及所有模块是否被正确的整合。当项目的各模块是由不同的团队开发时，模块的整合性测试是非常重要的。我听说过许多由于模块整合不当造成程序失败的例子。甚至在最坏的情况下，由于模块整合的失败使得整个项目陷入困境。

在版本发布中主要的任务是什么？ 很显然,文档”check in” 也就是包含所有与各 build 相关的新加的和修改的项目文档。BVT 主要用来检查初始版本健康度，也就是检查所有新加的和修改的文档是否被发布，所有文档格式是否正确，每个文档的版本、语言和标志是否和文档一一关联。
这些基本的检查在 build 发布给测试团队进行测试之前是有值得的。通过 BVT，你在一开始就能发现 build 的缺陷，从而节约时间和金钱。

BVT 应该包含哪些测试用例?

在自动执行 BVT 任务之前，这是一个非常棘手的决策。记住，BVT 的成功取决于它所包含的测试用例。

下面是为 BVT 自动化测试挑选用例的简单的贴士

- BVT 中仅包含重要的测试用例
- BVT 中的所有测试用例必须是稳定的
- 所有的测试用例应该有明确的期望结果
- 确保包含的所有重要功能性测试用例对应用程序测试做到充分的覆盖

另外，不要在 BVT 中包含还不稳定的模块。对于一些正在开发的特性你不能预知它们的行为，因为这些模块是不稳定的；或者在你测试这些不完整的模块之前你可能已经预知了一些已知错误。所以没有必要在 BVT 中使用这些模块或测试用例。

你可以通过了解相关的项目开发和测试生命周期，来使得为 BVT 挑选重要功能性测试用例的任务变得简单。这期间应该商定好最终可确保 BVT 成功的测试用例。设定一些 BVT 质量标准，并且这些标准只能通过分析主要的项目特征和方案才可以满足。

举例：关于文本编辑程序的 BVT 测试用例（一些简单测试）

1） 创建文本的测试用例
2） 向文字编辑器中输入一些文字的测试用例
3） 文字编辑器中复制、剪切和粘贴等功能的测试用例
4） 打开、保存和删除文本文件的测试用例

这些是简单的测试用例样本，是可以被标记为“重要的”的。对于程序中每个次要的和主要的改变，这些基本的关键测试用例应该被执行。使用 BVT 可以很容易完成这个任务。

BVT 自动化套件需要实时的维护和更新。例如，当有新的稳定项目模块可用时应增加测试用例。

当 BVT 套件运行时会发生什么：

版本验证自动化测试套件会在每个新 build 后执行

1） BVT 执行结果被发送至与项目有关的所有 email。
2） BVT 拥有者(执行并且维护 BVT 套件的人)需要检查 BVT 的结果。
3） 如果 BVT 失败那么 BVT 拥有者要诊断失败的原因。
4） 如果是由于版本中的缺陷而失败，所有与失败相关的 log 信息会被发送给相关开发人员。
5） 开发人员将其对失败原因的初步诊断回馈给团队。这是否是一个 bug？如果是 bug，修正 bug 的方案是什么？
6） Bug 被修正后，BVT 测试套件再次被执行，如果 build 通过了 BVT，它就被发送给测试人员进行进一步的详细功能、性能及其他的测试。

这些过程将在每个新 build 中反复运行。

为什么 BVT 或版本会失败？

BVT 有时会失败。这不总是意味着在 build 中存在 bug。有一些其他的原因使 build 失败，比如测试用例代码错误、自动化套件错误、基础结构错误、硬件失效等等。

你需要定位 BVT 失败的原因并且在诊断后采取适当的措施。。

BVT 成功贴士:

1） 花费足够的时间编写 BVT 测试用例脚本。
2） 尽可能详细的 Log 信息来诊断 BVT 通过或失败。这有助于开发人员进行调试，并

很容易知道失败的原因。

3） 为 BVT 选择稳定的测试用例。如果一些重要的测试用例在不同的配置中持续通过，那么把这些测试用例放在你的 BVT 中。这些将减少由于新的不稳定模块和测试用例所造成的 build 时常失败的可能性。

4） 尽可能自动化执行 BVT 流程。从版本释放过程到 BVT 运行出结果，一切都是自动的。

5） 对于破坏 build 的人员要予以惩罚要让那些破坏 build 的开发人员请吃巧克力或开咖啡派对。

总结：

BVT 只是一系列为每次新 build 所执行的回归测试用例。这也被称为冒烟测试。Build 不会被分配给测试人员，除非 BVT 通过。BVT 可以由开发人员或测试人员来执行，并且 BVT 的结果在整个团队中交流。如果 BVT 失败，将采取及时的行动去修正 bug。通过为测试用例编写脚本来实现 BVT 过程的自动化。只有关键的测试用例会被包含在 BVT 中。这些测试用例应该确保应用程序的测试覆盖率。BVT 对于 daily 版本和长期版本都非常有效。这些节省了大量的时间、成本和资源，测试人员也不会因为版本不完整而经受额外的挫折。

What you need to know about BVT

(Build Verification Testing)

What is BVT?

Build Verification test is a set of tests run on every new build to verify that build is testable before it is released to test team for further testing. These test cases are core functionality test cases that ensure application is stable and can be tested thoroughly. Typically BVT process is automated. If BVT fails that build is again get assigned to developer for fix.

BVT is also called smoke testing or build acceptance testing (BAT)

New Build is checked mainly for two things:

- Build validation
- Build acceptance

Some BVT basics:

- It is a subset of tests that verify main functionalities.
- The BVT's are typically run on daily builds and if the BVT fails the build is rejected and a new build is released after the fixes are done.
- The advantage of BVT is it saves the efforts of a test team to setup and test a build when major functionality is broken.
- Design BVTs carefully enough to cover basic functionality.
- Typically BVT should not run more than 30 minutes.
- BVT is a type of regression testing, done on each and every new build.

BVT primarily checks for the project integrity and checks whether all the modules are integrated properly or not. Module integration testing is very important when different teams develop project modules. I heard many cases of application failure due to improper module integration. Even in worst cases complete project gets scraped due to failure in module integration.

What is the main task in build release? Obviously file 'check in' i.e. to include all the new and modified project files associated with respective builds. BVT was primarily introduced to check initial build health i.e. to check whether - all the new and modified files are included in release, all file formats are correct, every file version and language, flags associated with each file.
These basic checks are worth before build release to test team for testing. You will save time and money by discovering the build flaws at the very beginning using BVT.

Which test cases should be included in BVT?

This is very tricky decision to take before automating the BVT task. Keep in mind that success of BVT depends on which test cases you include in BVT.

Here are some simple tips to include test cases in your BVT automation suite:

- Include only critical test cases in BVT.
- All test cases included in BVT should be stable.
- All the test cases should have known expected result.
- Make sure all included critical functionality test cases are sufficient for application test coverage.

Also do not includes modules in BVT, which are not yet stable. For some under-development features you can't predict expected behavior as these modules are unstable and you might know some known failures before testing for these incomplete modules. There is no point using such modules or test cases in BVT.

You can make this critical functionality test cases inclusion task simple by communicating with all those involved in project development and testing life cycle. Such process should negotiate BVT test cases, which ultimately ensure BVT success. Set some BVT quality standards and these standards can be met only by analyzing major project features and scenarios.

Example: Test cases to be included in BVT for Text editor application (Some sample tests only):
1) Test case for creating text file.
2)Test cases for writing something into text editor
3) Test case for copy, cut, paste functionality of text editor
4) Test case for opening, saving, deleting text file.

These are some sample test cases, which can be marked as 'critical' and for every minor or major changes in application these basic critical test cases should be executed. This task can be easily accomplished by BVT.

BVT automation suits needs to be maintained and modified time-to-time. E.g. include test cases in BVT when there are new stable project modules available.

What happens when BVT suite run:
Say Build verification automation test suite executed after any new build.
1) The result of BVT execution is sent to all the email ID's associated with that project.
2) The BVT owner (person executing and maintaining the BVT suite) inspects the result of BVT.
3)If BVT fails then BVT owner diagnose the cause of failure.
4) If the failure cause is defect in build, all the relevant information with failure logs is sent to respective developers.
5) Developer on his initial diagnostic replies to team about the failure cause. Whether this is really

a bug? And if it's a bug then what will be his bug-fixing scenario.

6) On bug fix once again BVT test suite is executed and if build passes BVT, the build is passed to test team for further detail functionality, performance and other testes.

This process gets repeated for every new build.

Why BVT or build fails?

BVT breaks sometimes. This doesn't mean that there is always bug in the build. There are some other reasons to build fail like test case coding error, automation suite error, infrastructure error, hardware failures etc.

You need to troubleshoot the cause for the BVT break and need to take proper action after diagnosis.

Tips for BVT success:

1)Spend considerable time writing BVT test cases scripts.

2) Log as much detailed info as possible to diagnose the BVT pass or fail result. This will help developer team to debug and quickly know the failure cause.

3) Select stable test cases to include in BVT. For new features if new critical test case passes consistently on different configuration then promote this test case in your BVT suite. This will reduce the probability of frequent build failure due to new unstable modules and test cases.

4) Automate BVT process as much as possible. Right from build release process to BVT result - automate everything.

5) Have some penalties for breaking the build  Some chocolates or team coffee party from developer who breaks the build will do.

Conclusion:

BVT is nothing but a set of regression test cases that are executed each time for new build. This is also called as smoke test. Build is not assigned to test team unless and until the BVT passes. BVT can be run by developer or tester and BVT result is communicated throughout the team and immediate action is taken to fix the bug if BVT fails. BVT process is typically automated by writing scripts for test cases. Only critical test cases are included in BVT. These test cases should ensure application test coverage. BVT is very effective for daily as well as long term builds. This saves significant time, cost, resources and after all no frustration of test team for incomplete build.

E测中国翻译团队首页：https://sites.google.com/site/5etestingtranslating/home

E 测中国翻译团队发布点：
点击进入E测BBS发布点
点击进入E测下载发布点

www.5etesting.com
E 测中国