Master Test Plan – the strategic side of testing

by [**Joel Montvelisky**](http://qablog.practitest.com/author/joel-montvelisky/) on March 4, 2008 in [**Test Process**](http://qablog.practitest.com/category/test-process/)

There are many software organizations where the term QA refers to a group of testers who randomly receive builds from the developers and proceed to “play” with the system in order fish out the bugs. In some cases these engineers use some informal *flows or scripts*to base their tests but these documents are not even close to a testing strategy or test plan of any sort.

We all agree that one of the main objectives of the QA team is to rid the product of the most important and disruptive bugs, but doing it completely Ad-Hoc and without a good preparation is the most ineffective and inefficient way to go about this process.

The common testing project is composed of 4 phases:  
a. Planning  
b. Preparation  
c. Execution  
d. Post-analysis  
In this post I will focus on the planning stage, since I believe this is the most important phase and the one that we take for granted the most.

So now we can start with 2 big questions:  
– What do we need to plan?  
AND  
– How do we go about planning it?

The answer to both questions is the MTP or *Master Test Plan* (also known as *Software Test Plan*, *Testing Strategy*, etc). The MTP is both a document and a process; by this I mean that at the end of the day you will have a document you can look at and admire (you may even hang it on the wall!), but not less important is the process you need to follow to create and communicate all the aspects that conform the document with all its sections.  
Also see: [Test Plan Recipe for a Mixed Formal and Informal Testing](http://qablog.practitest.com/2008/01/test-plan-recipe-for-a-mixed-formal-informal-testing-approach/)

**What makes a good MTP?**

Each company has different needs and thus each will require a different MTP template. The important thing is to understand that this document will represent your Scope of Work (SOW) for the specific project. It should be the place where you and your external stakeholders (Product Management, Development, Support, etc) turn to in order to understand what your team is testing and how are they approaching each testing task.  
To look at it in a simple way, imagine your company decides to outsource all its testing tasks to an external group (your group) and you need to put together a contract explaining what will your team do and what will it need in order to do it. Like all contracts, the idea is to review all the details and agree on them before signing the deal (or starting the project).

Following are the usual sections I include in my MTPs (remember to take them only as a suggestion and to modify them based on your needs!):

**1. Objectives of the testing process:**  
The objectives of the testing process depend on the nature of the development project. Examples of testing objectives are: new feature validation, additional configuration certification, translation validations, installation and/or upgrade testing, etc.  
Just make sure you don’t write trivial stuff like: *to find all the bugs in the system* or *to assure we release a quality product*.  
This section is to communicate what will YOU be thinking when planning and running your tests.

**2. Testing scope:**For many people the testing scope is the heart of the Master Test Plan. It describes the things you will focus in each of the application areas and/or features to test. I tend to make this section a nested list, and for each item I describe:  
– The main aspects to tests  
– The product risks or potential bugs I foresee  
– Concrete faults or main scenarios to validate  
– Assumptions or requirements (documented API, stable GUI, etc)  
– And any other aspects worth mentioning regarding the specific area under test.  
This is the place where you provide your stakeholders with the information about what will you be testing on each section of the product, and here is also where you should look for comments and suggestions from developers, product architects, support engineers, fellow testers, etc.

Some MTPs go the extra mile and provide a list of areas and features that are *Out of the Scope* of the testing process.

**3. Testing configuration matrix:**Your application surely needs to support a defined number of configurations and platforms, here is where you should list these configurations together with the testing matrix you will run in order to validate it.  
Keep in mind that by configurations we may refer to different things for different projects; on one project it may be Operating Systems and Browsers, while on another project it is additional product-components and specific versions required by your product to function correctly.  
In any case, by configuration we mean the environmental (and thus external) parameters required by our system to work.

In the cases when the list of officially supported configurations is more extensive than the systems you plan to test you should provide 2 separate lists:  
(a) The list of theoretically supported configurations, as specified by your customer or product management team.  
(b) The list of actual configurations you will test in order to achieve the above level of support, together with the distribution or percentage of tests that will be run on each.  
In order to do this I suggest a presentation format and planning method similar to what I described in my [last post](http://qablog.practitest.com/2008/02/a-par-wise-solution-for-your-planning-your-testing-environment-matrix/).

**4. Required Hardware / Software**Based on the testing scope and the required configurations you need to create a list of all the hardware and software resources you will need to complete your tests.  
In addition to special machinery and/or licenses, this is the place to ask for specific stubs or simulators you may require during your tests.  
If you plan to use automation of any sort include the number of software licenses as well as the amount of virtual users you will need for load and performance testing.

**5. Testing preparations**By now it should be clear what you want to test, now you need to understand what preparations to do in order to test it.  
For this point should make a high level review of your test plan inventory and compare it to your testing scope. You should end with tree lists of tests:  
(1) Tests that are ready to be run *as is*  
(2) Tests that need to be reviewed and/or updated to match the changes to the functionality  
(3) Tests you need to write from scratch  
For each list assign the amount of time you will require to work on the tests; you can also include the information and/or help you will need from other teams.

If you also need time to prepare testing environment or create testing data this is the section where to add this additional preparation costs

**6. Testing schedule:**(The favorite section of all our Project Managers!)  
Our operations are usually divided into stages and cycles. For example a project may have a preparation stage, an execution stage composed of 3 to 5 testing cycles, and a final product release stage/cycle.  
For each one we should provide at lease the following information:  
(a) Expected time lines  
(b) Entry and exit criteria  
(c) Testing scope & objectives  
(d) Testing resources (peopleware, software and hardware)  
and any additional information pertaining to the specific cycle.

**7. Testers & schedules**Following on the project side of the MTP you should list all your testers and the dates when they will be available for your project. List holidays, vacations, training and any other activities that may have an impact on the availability of a resource.  
If part of your tester-resources will be new make sure to account for the training and ramp-up time they will require at the start of their work.

**8. Risks**Like every project you should list the risk you may encounter. Examples of risks are difficulties in recruiting resources, instability of the product that may delay your schedule, high attrition rates, etc.  
Each risk should include the following information:  
(a) Person in charge of the risk  
(b) Severity and likelihood of the risk materializing  
(c) Dates of relevancy when the risk may materialize  
(d) Consequence of the risk materializing  
(e) Prevention & contingency plans

**9. References & attachments**Add links to any additional documents and/or information referent to the project.  
Add also a list of all the contact persons as well as their areas of responsibility.

**A final word of advice –** Some companies and processes are not ready to handle a full blown MTP, specially start-ups and/or companies working under less structured development process (e.g. [Agile Development](http://www.practitest.com/agile-testing/)).  
If you work on one of these companies it doesn’t mean that you can or should work without a strategic QA plan, it simply means that you need to mold it in order for the plan to meet your company culture and way of life