Guidelines:Testing

Overview

This guidelide do not intend to explain a fully developed testing procedure, only a first and basic (but functional) approach to testing development.

Testing can never completely identify all the defects within software. Instead, it furnishes a criticism or comparison that compares the state and behavior of the product against oracles—principles or mechanisms by which someone might recognize a problem. These oracles may include (but are not limited to) specifications, comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria.

The separation of debugging from testing was initially introduced by Glenford J. Myers in 1979. Although his attention was on breakage testing ("a successful test is one that finds a bug"4[5]), it illustrated the desire of the software engineering community to separate fundamental development activities, such as debugging, from that of verification.

So, seems to be a good idea to have a testing team working parallel to development team. This is sometimes impossible due lack of resources, but the way it works its the same even the same person performs the development and the test: he should separate both enviroments and the way testing and debugging is done.

The **primary purpose for testing is to detect software failures** so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do.

The real world

Three levels of testing has been identified:

- Developer level: Local checks do on the developer enviroment.
- Development integration testing: Done for a small team of different developers, testing units not developed by theirselves.
- External (Q/A) global testing: Done for non-development staff, and after the validation of the development team.

Development and testing, are a cycle, and after founding a bug, identifyng it and writing in the tracker, development for that issue will start to the beginning, and all the process will repeated, until no bugs are identified by the external Team.

This three phases are identified in a classical nomenclature:

- Development.
- Pre-Alpha phase.
- Alpha phase.

Packages will be delivered for open-testing to the public on Beta phase, is coming after a successful testing for Q/A team.

Some considerations for the testing done by developers

These are general considerations to be taken in mind by developers:

- Testing should be done always in an external system, DIFFERENT from the development enviroments. That means, if you develop in a Ubuntu, testing should be done in the “most similar enviroment to a production site”, in Pandora FMS that means OpenSUSE 11.x. Use a Virtual Machine for that.
- Use *ALWAYS* real data, or data which seems to be real, that means “weityweruyewr” is not a good name for anything, use **real names** and try to imitate **real data** as much as possible. We have tools for creating data, similar to real enviroments, and data to export / import configurations just for helping have a “real” system with “real” data: **USE IT**.

References

Wikipedia: Software Testing http://en.wikipedia.org/wiki/Software_testing