**Other questions to answer about testing and testability, from your point of view**

* How do you review code?
* How do you enforce coding standards?
* How do you plan what kind of approach you take for test automation - what libraries to use, how does it work in couple of years, how to make it easy to maintain, etc? What are the main points to consider?
* Code testability, how do you enforce it?
* How do you make sure that the product is testable?

1. I review the code in TFS in a pull requests or sometimes pulling the branch to have the code in my IDE. I try not to spend more than 30 minutes for one review session. If I have any questions on any part of the code, I add comments on the pull request page in TFS. After my comments are resolved either with a comment back or a code fix, I resolve my comment.
2. Code standards should be applicable for all the code base, it eases code readability especially for the juniors, it also improves the reliability of the tests. We enforce coding standards explaining and showing examples on common sessions.
3. It firstly depends on an application under test. So sometimes modular testing is enough, or keyword testing is more sufficient. If an application requires a lot of data to be populated, then Data testing is the best. Sometimes many testing approaches can be combined. It is better to use libraries and tools that are not out of date, that are supported well and have new releases with new features from time to time. Of course, the tools that suit more the testing approach and the type of testing. The code should be written in a readable and reliable way to maintain it in the future.
4. Actually, in my projects I never test the tests or write unit tests for the code, so it is hard to answer although I understand that the code should be readable not chaotic and should be implemented against common principles.
5. The product is testable enough if it has clear requirements and can be tested with any existing tool. IF UI is needed to test that the elements should have stable locator anchors and so on.