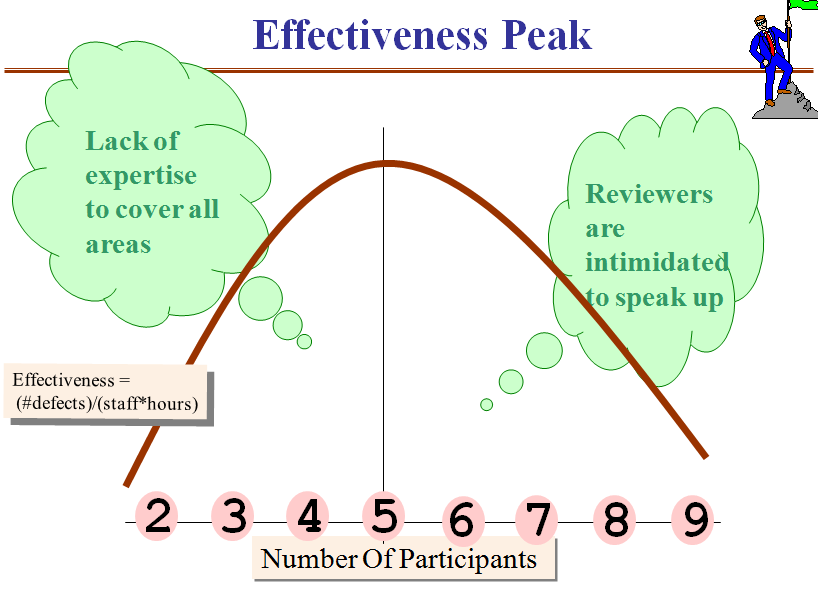
1. Relate the pictorial below to your environment. What is the optimal number of participants in a code review? What would be a result of an insufficient participation? Or over-abundant participation?

**Answer:** Relating the figure given below the optimum number of participation is 5. When it is less than 5 the productivity is low whereas excess leads to drop in the effectiveness peak .The result of insufficient participation leads to lack of expertise to cover all the areas as less reviewers means less contributed area and from different perspective as well. The over-abundant participation leads to reviewers feeling intimidated in front of many reviewers, which by human nature is not so surprising.



1. Describe circumstances when a F2F review meeting - could be skipped ? (see activity #5 of the diagram Module 4 Section 5 "Swim Lanes").

**Answer:** When there is no or less preparation before the review process begins it is beneficial for all the participants to postpone the meeting because to arrive at a peer meeting without through examination of artifact is not productive at all and it just waste to time, money and frustration for the well prepared participants as well.

1. The dynamics of an active peer review meeting involving multiple participants - is quite different from a situation when a reviewer is sitting quietly in his/her office while trying to find hidden defects. An experienced Author is able to leverage both of these very different environments.

**Please provide three examples of types of defects that are found**

* + - during preparation, versus
    - during F2F meeting

Focus on an approach to finding defects instead of focusing on defects themselves.

An Author sends a checklist included with an initial distribution of an asset. Such checklist would leverage the individual preparation of a reviewer and would say something like...

* please examine the document I am attaching; and make sure it does not conflict with documents you are currently working on.
* Or it could say something like....I am not sure about this specific piece, hope you could give me a hand here.  
    
  An Author sends a very different note toward F2F meeting so to leverage the convergence, clarification, synergy building environment. It could sounds something like....
* during the upcoming meeting, while we examine my document page-by-page.... please remind each other about the most visible issues we have had in the past and make sure my document is clean of such issues.
* Or such note could say...please prepare a list of your assumptions, so during the meeting, we shall resolve the differences in these assumptions and will also make sure my document is based on our common assumptions.
* Or it could say...here is the key use-case scenario, when a user is pressing these five buttons, right in the middle of the download; and I would like for all of us to go through this scenario step-by-step and note any possible inconsistencies in system's behavior that my document presents

**Please provide three examples of types of defects that are found**

* + - during preparation, versus
    - during F2F meeting

**Answer:**

During preparation:

1. Omission of code

2. Incorrect coding

3. Ambiguity of code in the context

During F2F meeting:

1. Defect origin (where the defect is originated)

2. Defect severity (minor, major, unknown, not harmful)

3 .Defect location (repository, storage device, shared resource)

1. Provide your assessment of the relationship between the unit test and code reviews.

* State several advantages of facilitating a code review prior to executing a unit test

**Answer:** Reviewers can often spot the bugs that’s are missed. The types of bugs found happen to be fairly obvious and common ones.

Another important benefit of code reviews is the training that automatically results as part of the process. Training during code reviews is bidirectional. The reviewer is trained on the new code developed by the implementer, while the implementer is trained by the recommendations made by the reviewer. Code reviews are also an excellent opportunity for senior engineers to mentor less experienced engineers. During the code review process documentation of the review is maintained. Added documentation improves code quality by making it more understandable to others, and thus of greater value to the software organization

* State several advantages of bringing results of a unit test to a code review

**Answer:**

Increase productivity and software quality

Identifying runtime bug without executing software

Automate code analysis for compliance

Effective and comprehensive team code review

1. List several underlying reasons hindering adoption of peer reviews as a consistent practice across a large organization

**Answer:**

1. The author must trust and respect the reviewers enough to be receptive to their comments. Similarly, the reviewers must show respect for the author’s talent and hard work. Reviewers should thoughtfully select the words they use to raise an issue, focusing on what they observed about the product.
2. The attitude and behavior that managers exhibit toward reviews affect how well the reviews will work in an organization. Although managers want to deliver quality products, they also feel pressure to release products quickly. They don’t always understand what peer reviews or inspections are or the contribution they make to ship quality products on time. Managers also must be sensitive to the interpersonal aspects of peer reviews. Watch out for known culture killers, such as managers singling out certain developers for the humiliating “punishment” of having their work reviewed.
3. Many people don’t understand what peer reviews are, why they are valuable, the differences between informal reviews and inspections, or when and how to perform reviews. Education can solve this problem. Some developers and project managers don’t think their projects are large enough or critical enough to need reviews
4. The misperception that testing is always superior to manual examination also leads some practitioners to shun reviews. Testing has long been recognized as a critical activity in developing software.
5. Developers need a robust enough ego to trust and defend their work, but not so much ego that they reject suggestions for better solutions. Software professionals take pride in the things they create.