Scope Management

Yegor Bugayenko

Lecture #2 out of 10
80 minutes

The slidedeck was presented by the author in this YouTube Video

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as web sites. Copyright belongs to their respected authors.

Management Triangle

Your responsibility, as a project manager, is to *control* that all three elements of the triangle are *in sync*: when one of them changes, you have to change the other two. If you don't know how much the project will cost, how long it will take, or how much functionality it will implement—you are not a project manager. Maybe a leader, a mentor, or … a scapegoat.

1. Even though it was not required by the customer, a programmer suggests implementing an additional feature in the product, because it is obvious that users will love it; what do you say, as a PM?

1. “Definitely not!”
2. “Maybe we should discuss with the customer first?”
3. “Only if it doesn’t delay all other features”
4. “Sure, customer first!”

#gold-platting

2. Which definition of a task for a programmer sounds perfect for you?

1. “Make sure `FooTest#test15()` passes in the `master` branch!”
2. “Fix the build, make sure it’s green!”
3. “Together with the team, resolve the problem ASAP!”
4. “Make sure the customer is happy!”

#d-o-d

3. Which one is the right formulation of a functionality in a Use Case?

1. User *can* download a picture
2. User *downloads* a picture
3. User *will* download a picture
4. User *should* download a picture

#use-case

4. A customer asks you how much work is left to be done. Where do you find this information?

1. You ask your team
2. Use Cases
3. Backlog
4. Traceability Matrix

#scope-control

5. Which one *isn't* a Non-Functional Requirement (NFR)?

1. When "Send" is clicked, email must be sent in less than 500ms
2. The "Settings" page must be intuitively easy to use
3. A picture of 100Kb downloads in less than 1.5s
4. In case of a security breach at any web server, user passwords won't leak

#use-case

6. After six months of hard work, your team releases the product to customer's servers. The customer says: "This is not what I wanted :(" Whose fault is it?

1. The *project manager* didn't validate it earlier
2. The *customer* didn't explain what they wanted
3. *Testers* didn't verify it earlier
4. It's *nobody's* fault! We don't blame! We learn and improve!

#V&V

7. A project of *one year* and *five programmers* can be decomposed into how many Work Packages?

1. Nobody knows
2. Seven
3. 25 sprints, 5 coders, 1 WP per week → 250 WPs
4. Hundreds

#use-case

8. Which estimate of project scope is the most reasonable?

1. 100 *Function Points* implemented
2. 500 *Pull Requests* merged
3. 50,000 *Lines of Code* written
4. 10,000 *hours* of hard work, including weekends

#estimate

Homework:

“A *Vision* document defines the high-level scope and purpose of a program, product, or project. A clear statement of the problem, proposed solution, and the high-level features of a product helps establish expectations and reduce risks.” — IBM Engineering Lifecycle Management

Read this:

SSD 2/16: Lecture about Requirements Engineering (2022)

Wikipedia: COSMIC, Function Point, MoSCoW method

Karl Wiegers et al., *Software Requirements* (1999)

Alistair Cockburn, *Writing Effective Use Cases* (1999)

Incremental Requirements With Requs (2014)

10 Typical Mistakes in Specs (2015)

How We Write a Product Vision (2014)

Bibliography