Where does "Change Management" end and "Project Failure" begin? [closed]

Asked 16 years, 3 months ago   Modified 12 years, 10 months ago

Viewed 2k times

1

I got into a mini-argument with my boss recently regarding "project failure." After three years, our project to migrate a codebase to a new platform (a project I was on for 1.5 years, but my team lead was on for only a few months) went live. He, along with senior management of both my company and the client (I'm one of those god-awful consultants you hear so much about. My engagement is an "Application Outsourcing") declared the project to be a success. I disagreed, stating that old presentations I had found showed that compared to the

original schedule, the delay in deployment was best measured in months and could potentially be measured in years. I explained what I know of project failure, and the studies and statistics behind failure rates. He responded that that was all academia, and that no project he led had failed, thanks to the wonders of change/risk management - what seems to come down to explaining delays and re-evaluating the schedule based on new data.

Maybe consulting like this differs from other projects, but it seems like this is just failure wrapped up in a prettier name to avoid the stigma of having failed to deliver on time, on budget, or with full functionality. The fact that he explained that my company gave away hours of work for free in order to finish the project within the maxed out budget says a lot.
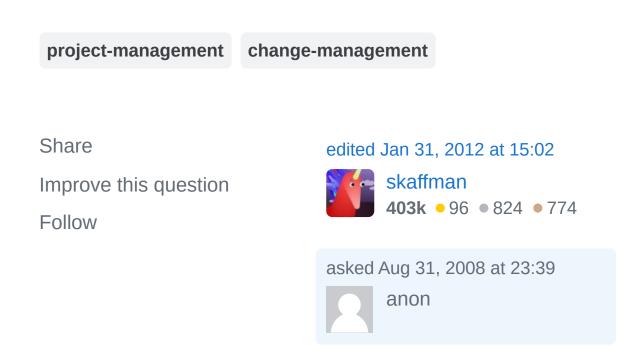
So I ask you this:

- What is change management, and how does it apply to a project?

- Where does "change management" end, and "project failure" begin?

@shog9:
I wasn't asking about a blame game with the consultants, especially since in this case I *represent* the consultants. I was looking for views on when a project should be considered "failed" regardless of if the needed functionality *was* finally implemented.

I'm looking for the difference between "this is actually a little more complex than we thought, and it's going to be another week" which I'd expect is somewhat typical, and "project failure" - however you want to define failure. Is there even a difference? Does this minor level of schedule slippage constitute statistical "project failure?"

project-management   change-management

5 Answers

Sorted by: Highest score (default) ⇕

I think, most of the time, we developers forget this we all do is, after all, about bussiness.

From that point of view a project is not a failure while the client is willing to pay for it. It all depends on the client, some clients have more patience and understand better the risks of software development, other just won't pay if there's a substantial delay.

Anyway, about your question. Whenever you evolve a project there are risks involved, maybe you schedule the end of the project in a certain date but it will take like six month longer than you expected. In that case you have to balance what you have already spent and what you have to gain against the risks you're taking. There's actually an entire science called "decision making" that studies it at software level, so your boss is not wrong at all.

Let's look at some questions, Is the client willing to wait for the project? Is he willing to assume certain overcosts? Even if he doesn't, Is worth completing the project assuming the extra costs instead of throwing away all the already done work? Can the company assume what's already lost?

The real answer to your problem lies behind that questions. You can't establish a point and say, here, if the project isn't done by this time then it's a failure. As for your specific situation, who knows? Your boss has probably more information that you have so your work is to tell him how is the project going, how much it will take and how much it will cost (in terms hours/man if you wish)

Share   Improve this answer

Follow

answered Sep 1, 2008 at 0:07

Jorge Córdoba
52.1k ● 11 ● 82 ● 130

Unless the goals were clearly stated in the beginning of the project, there are no clear lines between "success"

1

and "failure." Often, a project would have varying degree of success/failure.

For some, just getting some concepts in code would be a success, while other may measure success as recovering all investments and making profit. Two well-known modes of failures are schedule slip and quality deterioration, but in real-world, people do not seem to care much about them.

Simple ways to slip the schedule are to let the managers make request whenever they want (features creep) and let the programmers code whatever they feel is right (cowboy coding). Change management process such as sprint planning of scrum and planning game of XP are some of the examples. Theses are some of the attempts for the management and the developers to ship reliable products on time. If either party is not interested in reliable or on-time, then change management would not be useful.

Share   Improve this answer

Follow

answered Sep 1, 2008 at 0:09

Eugene Yokota
95.5k ● 45 ● 217 ● 320

0

I suppose how successful the project is depends on who the client is. If the client were the company directors and they are happy, then the project was successful regardless of the failures along the way.
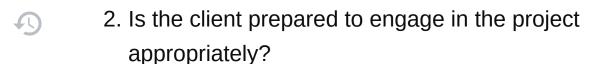
answered Aug 31, 2008 at 23:59

GateKiller
75.8k ● 75 ● 175 ● 204

[Andy Rutledge](#) has written a pretty interesting article on success. Though the title is [Pre-bid Discussions](#), the article defines *having a successful project*, which for Andy entails:

0

1. Will I or my team be allowed to bring our best work to the final result?

2. Is the client prepared to engage in the project appropriately?

3. Is the client prepared to begin this project?

4. Is the client prepared to invest trust in my or my team's ideas?

5. Am I or is my team prepared to fulfill or exceed the project requirements?

This article was pointed out by Obie Fernandez, a successful consultant, in his [Do the Hustle](#) conference about consulting.

answered Sep 1, 2008 at 7:25

Christian Lescuyer
19.2k ● 6 ● 50 ● 45

What is change management, and how does it apply to a project?

0

Change management is about approving and communicating changes to a project before they happen. If someone on your project (user, sponsor, team member.. whoever) wants to add a feature, the change needs to be documented and analysed for the effect. Any resulting changes to scope, budget and schedule must then be approved before the change is undertaken. These changes are typically approved by your sponsor, your steering committee or your client.

Once the changes have been approved and accepted that is your new plan. It doesn't matter what the original budget or schedule was.

Change Management on projects is all about the principle of "No Surprises". The right people (your Change Control Board) need to approve any changes to Scope, Schedule and Budget before they are acted upon.

One thing to remember is that there may be certain explicit or implicit constraints and tolerances for change. You may be have to deliver your project by a certain date to meet government regulatory requirements. Or your organisation may have a threshold that once a project budget is 30% over the original budget it must go to a "C" level or the project is killed. Investigating and explicitly stating these thresholds and tolerances up front are a good way of having better successful projects.

Where does "change management" end, and "project failure" begin?

If a project delivers on the approved scope, schedule and budget then it is successful.

However it may be still viewed as a failure. Post Implementation Reviews are a good tool to qualify this with your stakeholders (not just your boss). Also Benefit Realisation would be worthwhile looking into to see outside the blackbox of the project and the impact on the business as a whole.

Share Improve this answer

Follow

answered Sep 4, 2008 at 4:32

Mark Nold

5,698 ● 7 ● 33 ● 34