

How many people should I put on this (Agile) project?

Perhaps one of the most commonly asked questions about Agile projects is simply: How do I know how many people to staff a project with?

Its actually very difficult to answer this questions for Agile or non-Agile projects but those working Agile have to face up to this difficulty right from the off. Non-Agile projects have a ceremony which provides an answer.

The right answer to this question is really: Staff your project strategically and inform your decisions with empirical data. Before I explain that in detail lets look at what actually happens with staffing.

Classic ritual

There is a ritual that most software developers will have been involved with at sometime. A project manager is assigned a new project or section of a project. He visits selected individuals and asks for "estimates" of how long it will take to do some piece of work. He takes these estimates, lies them end-to-end, and announces a date they reach.

There is an underlying assumption that there is a fixed lump of work to do, this can be measured in days and when the number of days is divided by the number of developers on the project then you have an end date. If the end date isn't satisfactory you can add developers - increase the denominator - and reduce the end date.

According to legend this method allows companies to decide how many people they should assign to a project, how long the project will take and how much it will cost. However just about everything here is myth.

Projects are never really staffed on the basis of "how many people do we need" because no project is ever staffed in isolation. Suppose you are kicking off a new project today, sometime in mid-2011. Chances are you will staff the project with more people than you would have in mid-2009 when the world economy was looking a lot less secure. Still you will probably have few people than you would if you did the same thing in min-2008, prior to the Lehman's collapsed when the economy was booming.

Neither are estimate given in isolation. Stories abound of Developers and Project Managers who massage estimates upwards or downwards depending on their desire to do the work or their desire to impress their managers.

In fact the starting assuming that there is a lump of work and it is constant is itself a myth. Parkinson's Law - " Work expands so as to fill the time available for its completion" and Conways Law - " organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations" - tell us that changing the denominator will change the numerator.

The equation is not so much:

$$\text{Elapsed days} = \text{Lump of Work} / \text{Number of People}$$

as

Elapsed days = F(Lump of Work, Number of People)/Number of People

So lets not pretend that this "how many people should I put on this project" problem is somehow unique to Agile, it isn't. Its common to all work.

Use the data

Agilists and traditionalist would probably agree that the best way of estimating the manpower to be deployed is to look at the data. This approach works fine when a project, a team, is already at work. Teams can estimate upcoming work using their favourite units - story points, hours, nebulous units of work, ideal days, or my favourite abstract points; add up these units and you have the denominator.

Deduce a denominator from the teams current velocity in the same units - I like to use a rolling average of the last three iterations, many teams just use the last iteration value; divide one by the other and you know how many iterations you need.

Unfortunately if you don't like the answer you are a little stuck. You can add more people to increase the velocity but Brooks Law - "adding manpower to a late software project makes it later" - tells us that this is going to take time to have an effect. The Agile answer is to start scope cutting.

Actually, scope cutting is the traditional answer too, Brooks Law holds for all methodologies. The difference is Agile starts cutting scope from the off, traditionalists wait until they have once again proved Brooks Law.

Of course this approach is utterly useless when a project is starting. Until a team have formed and learned to work together you have no velocity, nor do the team have a shared understanding of their unit of estimation is so you have no numerator either.

Again, this isn't a problem unique to Agile; it affects traditional methods too but is hidden. Setting a team size or suggesting a completion date without any data to back it up borders on professional incompetence, it's little better than guessing.

Historical data offers one answer: what have similar teams done in similar circumstances? Many companies track the effort expended on projects for just this reason. However, this approach too is flawed too.

Capers Jones has studied this data over decades and is damning: "frequent project accounting errors can cause the true cost of a software project to be up to 100% greater than the apparent cost." Jones suggests a many reasons for including the omission of unpaid overtime, user effort and managerial effort.

Jones goes on to say that to poor quality of effort tracking is a major contributor to project failure. Because the data is so poor any estimates based on it are likely to be wrong. Consider a project which had 10 man months logged against it, suppose unpaid overtime and other omissions led to

a 25% under. If the company used this project as benchmark for the next it would be under priced and behind schedule on day one.

Strategic staffing

So it looks like which ever way we turn we are stuck. What is a poor project manager to do? The answer lies in taking a strategic view of project staffing and arranging the team to adopt.

First there needs to be a recognition that without some accurate data there can be no realistic estimates. This is one of the risks we face at the start of a project. There are some things we can do to minimise this risk.

Avoid creating new teams, reuse the teams you have, don't disband a team when its work is completed, roll them into a new project.

If you must create a new team then start small. Two is the real minimum for a team - probably a software engineer and requirements specialist, or two engineer. This team is then locked in a race to produce something that works and some data about which can be used to reason with.

Starting small minimise the downside if things go badly. If things go well the team will still have to contend with Conway's Law as they expand but at least they won't be behind schedule

Accepting these risks, appointing a bootstrap team and delaying further staffing decisions means these decisions need to be take at a strategic level. Since all resourcing decisions are really made in relation to how else the resources could be deployed all resourcing decisions are really strategic.

All resourcing decisions are made relative to other work, this fact is overlooked completely in the classical model. Every time a resource is assign to a project it is a decision not to assign them to another project, or not to invest the wage costs in risk-free bonds.

Assigning resources is not a one off decision. It is one that needs to be reviewed regularly as part of portfolio process and resources rebalanced. Projects which are demonstrating success - high velocity, value delivered, business benefits - need to be supported and promoted; those that don't need to be shrunk, closed or reset.

The Rules

There are four rules for staffing project:

- 1: Without data you can't forecast anything so don't pretend you can.
- 2: Start small, get data, grow later.
- 3: Staff work strategically relative to other work to be done, corporate priorities and risk. Rebalance only occasionally
- 4: Winning teams stay together, avoid disbanding performing teams.

Without data telling your client "We think we need six people for six months" is little better than lying. Without data you need to conduct experiments to get the data, and some experiments will go.

Better to tell the client: “We propose to start with three people for three months, at the end of that time we will review progress with you and decide with you, when we have data, whether to increase or decrease staffing. If you want to move faster than we can allocate four people and review after one month but that will mean more risk”

It might not be what your client wants to hear but it sure beats guessing, or lying. Involve your customer, give them choices. It might not be a great sales technique but if the client won't engage in this conversation there are probably other conversations they won't have either.