Table of Contents

Under what conditions is Full-Time Equivalent (FTE) resource planning effective?

Traditionally when we plan work for Portfolios of Projects we will pull together a detailed resource plan of our people with the idea being that we can predict how people will work to deliver Projects. We use this information to:

- Develop and work a plan: To understand how we might schedule the work of the effort out to ensure success
- Estimate cost: To help determine how much the effort will cost us, at least from the perspective of people cost
- People and skills: Understand whether we have the people and skills available to get the project done

In order to ensure that we have the best use of our people (that they are being well or fully utilized) we treat people as “full time equivalents” (or FTEs) and then allocate them to work on Projects so that, for example, a person is allocated 25% to one Project, 30% to another, and the remainder (to 100%) to a third Project.

Lean-agile implementations take a different approach. Rather than focus on the resources, they optimize the flow of business value by implementing a “pull” based system. We assume that we have a fixed capacity machine organized as stable teams to do work and that most of the time we have the right kind of people to get the work done (after all we have been getting work done up to now). The main issue we need to work is to ensure that our people are working on the most important things. We therefore prioritize the efforts on and pull work to the Teams on a regular cadence so we maximize value delivery for the capacity we have. We then make adjustments on this cadence as things change.

A lot of organizations still require the Portfolio planning approach to work FTE based resource planning as part of the approval process. For those organization we need to understand under what kind of circumstances this kind of planning is effective both from the perspective of the results it produces, and in terms of being able to provide answers to the questions. Lets start with:

Develop and work a plan

Let’s start with is this effective in terms of results - does the plan actually work? My experience is that this kind of planning is not effective except for the smallest of Projects, where there is a huge amount of existing experience, and where the Projects are independent from other Projects. For anything more than this it doesn’t work on a couple of levels:

- From an individual perspective, the people doing the work have a tough time working the way the plan implies
- From a systems perspective, it is impossible to deal the impact of changes that happen

From an individual perspective, I want you to put yourself in the position of the person that has been assigned to work 25% on one Project 1, 30% on Project 2, and the remainder (to 100%) on Project 3. The problem is that we are putting this person in a pretty difficult situation. Firstly, the reality is that this person doesn't actually have 100% of her capacity available to do work. You lose capacity as you switch from one type of work to another. For 3 Projects, the drop in capacity is typically around 40% leaving an actual capacity of the person to be 60% of the total (See What is the Impact of Context Switching on the Ability to Deliver? for more on this). If there were a fourth project another 20% would be lost. The problem is that there is always other work coming in - the production support issue, the meeting we need to attend, … The capacity plan we have developed is pretty much dead on arrival.

Even if the plan is actually executed as we planned there would still be problems. From the person's perspective, they are juggling 3 backlogs and need to decide which Project is to get their attention right now. They do this from their perspective and understanding, which typically does not have a wide enough context to do what in the end is best for the business. My experience is that people do not think in terms of allocating their time 25% to this project, 30% to another, even with the most detailed schedule for their work. They will work on a particular project for a time until they have completed something or are blocked by something and then move on. At the end of the week they will look back and guess how much they worked on each project, and then adjust those guesses based on expectations from management ("you can't charge to Project X …"). Bottom line is that my experience has been that our developers, QA, etc. do not really operate per the plan anyway - the plan will be impacted by local decision making.

From the system perspective, the issue is different - it's just too hard to plan out all the changes that need to be addressed as a result of change. It is possible to successfully use this approach to plan for our people's work for simple efforts and a small number of people. However it quickly becomes impossible to use this approach as the number of Projects increases and as the number of interactions that can occur between Projects between increases. The reason this happens is that, while we might start off with a plan that seems reasonable, as we execute the plan it will be impacted by random events (e.g. "system is down!") which change the interactions between people working the project, interactions between people and the technology and systems they are working with, and the overlapping relationships between this project and all the other projects, with their people, technology, and systems.

For example, say one person, a developer, has to take a day off because they are sick. The developer was going to work on Project 1 with her 25% allocation to that project. In the simplest case this will impact the next person that is waiting on their Project 1 output so they can do their work, say a QA person who was expecting to use some of their 10% capacity allocated to this Project 1. The QA person realizes that the developer is not going get that work done, and so decides to work on Project 2 instead, another part of their allocation. However this decision has at least two impacts 1) anyone waiting on Project 1 work from the QA person 2) anyone how is now expected to respond to the change as a result of the QA's decision to work on Project 2. And so on. And so on. And so on. The result is that this one change in the plan has a potentially huge ripple effect not just through one Project, but potentially impacting multiple Projects.

And this is just one random event. By building a plan with FTE's we have actually created a whole series of dependencies between Projects whether there are other more direct dependencies between those Projects or not. In this network of people doing projects, how many random events are there - from the large such as "The competitor has released a product that makes this work obsolete" to smaller events such as “Having seen the result, that’s not what I need…”, “You need to work this production issue!”, “Management has scheduled a meeting with …”, “We have a new person we need to onboard.”, “The test environment is not available”, etc. Change is in fact the norm.

There are a lot of potential interactions that could result from a change. As a result the impact on your future plan is simply unknowable. To give you a flavor of the number of potential interactions between people:

Number of people (Dots - N)	**Number of direct interactions (Links - L)**	**Number of direct and related interactions (Patterns - P)**
N = 2	L = 1	P = 1
N = 4	L = 6	P = 64
N = 10	L = 45	P = 35 trillion
In general	L = N*(N-1) / 2	P = 2 ** L

Source: Max Boisot

The chart says that if we have 10 people, then the number of potential direct and related interactions is 35 trillion. Wow.

In reality, the situation is probably not as extreme as this level of interdependency indicates. We do not expect to have everyone working with everyone else. Rather we work to isolate dependencies by having people focus in particular business areas or technologies for example. Further, not all people will have the same impact. Some people will see a change, and manage it themselves, and that change will have limited “blast radius”.

To me the real impact of this chart is how big the numbers of potential interactions become even for small numbers of people. In my experience, the burden of FTE type thinking typically falls on the people that we regard as experts, as critical. We have a tendency to put more onto these people which creates a huge dependency for all projects (the “Brent” problem from the book the “Phoenix Project”). These are also the people we want to leverage when we have a problem and so they are more likely to be impacted by random events. Let’s say there are 10 of these folks so the number of potential interactions, the ripple effect, is around 35 trillion. So when something impacts those key people the ripple effects are potentially dramatic. And with these numbers you can see why it is impossible to predict what might happen as part of a re-planning exercise.

While it is comfortable to think that we can plan this upfront, this turns out to be wishful thinking. There are just too many potential changes and ripple effects to really ensure we will be able to deal with changes in our system. Sadly, looking back on the project we will be able to pinpoint the problem and even have a discussion about what we should have done differently. Unfortunately, this will only reinforce the idea that we can be deterministic in our planning process - “If we only had analyzed this more!”. In fact we know something now only with the benefit of hindsight because, rather than dealing with 35 trillion possibilities, we can see the actual possibility that happened - this would have been impossible to determine this at the time of the change. This is the result of the kind of system we are operating in - it is a complex system where it is not possible to have accurate upfront planning.

Estimate cost

There is no doubt that using the FTE type approach will allow us to develop an estimate for the effort. In many ways this is seen as a requirement for portfolio planning. We need to know the cost in order to ensure that when we work on the item we have a positive business outcome - a return on this investment if you like.

Like all estimates, the use of FTE type thinking is a guess. We might think that we have a “good” guess because in going through the process of determining the FTE plan we’ve probably also thought a little about how we are going to get it done (architecture and design work) as well as the steps required to get something done (FTE scheduling).

From the planning discussion above we can see that this level of “additional” detail is actually not going to improve the estimate much as the bigger impact on the plan is change. In fact it is even worse than that in that the process gives us a false sense of security. Let’s face it, the time when we are pulling these estimates together is the time when we know the least about whatever it is we are doing. We generate additional documentation, but the reality is that we have not delivered anything the customer, or the business, cares about - (part of) a working system - and so don’t actually know what the rate of progress will be. It is only when we start delivering real functionality that we can understand what the likely progress of the work is.

This is not to say we should not provide estimates. Prudent business decisions are based on understanding some view of the potential cost and potential outcome. But there are quicker and cheaper ways to get at cost (such as using relative sizing with the historical throughput) without the overhead of building a plan that won’t be followed. In addition, with lean-agile approaches we will review the information we have on a cadence and, based on what we are seeing, make a call for the best use of our scarce capacity right now, updating the financial data as we go based on what we are seeing - incremental funding.

In other words we replace upfront approval process and plan with an incremental funding approach which optimizes our delivery of value.

People and skills

The FTE process usually incorporates a view of the skills we need for a project. It might say things like “we need a JDE developer, a QA person, etc.” The thinking is that using this information we can start having discussions about whether we have the skills we need and whether we have enough people to get the work done.

One downside of this approach is that the natural discussion will always be “we need more people”. The reality is that there is always more demand on the system (more requests) than can be possibly handled. Typically this results in a constant feeling that we are understaffed. But it should be noted that we are only understaffed in comparison to the budget we currently have. The real discussion we need to have is whether it is worthwhile to bring in additional people in the light of the amount we have budgeted from an overall business perspective.

Lean-agile takes a slightly different approach. At the broad level of portfolio planning, we can be pretty sure we have people with the skills that we need to do the work we need. We know this as we have a history of delivering the work. The approach lean-agile takes that work and prioritize it, force ranking every effort (usually within specific domains). Based on historical data we know how much we have been traditionally been able to complete per unit time. We can leverage this to determine the cut-line of the work, above which it will probably be done, below which we cannot start right now. Looking at the Initiatives below the cut-line will allow us to have a focused discussion about whether it is a good business decision to bring more people in to address some of these initiatives.

There is an additional discussion about skills. In general the organization has the skills it needs to get the work done in the organization. If we find ourselves with too few of a particular skills we can leverage the organization to teach others within the organization the skills that are required. In general the 80-20 rule applies here - you can teach someone to support 80% what they deliver very quickly with the remaining 20% requiring more experienced involvement. For example, one place I worked, we did not have enough QA for the amount of work. We put a plan in place to bring in more QA, but realizing that this would take some time (and be potentially disruptive) we had our QA folks work with our development folks to better test their efforts - think like a QA person. Sure, the first use of people in this skill area might be slower, but this approach is general less disruptive than bringing in new people to get the work done, and you end up with a more resilient organization, able to take anything on.

The place where we could anticipate a skills mismatch is when we are contemplating an Initiative that is in a totally new domain or uses a totally new technology. If you are unable to leverage internal people to grown their skills, learning, then we might chose to bring in additional experts. From a lean-agile perspective the mechanism exists to address this. Since we are planning and implementing on a cadence (say quarterly PI), if we discover this need we raise the need as part of the process to get ready for the next PI and work to bring in the required skills on the cadence. This has the additional benefit of reducing disruption.

In answer ...

There we have it. In answer to the question “Under what conditions is Full-Time Equivalent (FTE) resource planning effective?” and related questions about planning, estimating costs, and people and skills, for any organization of sufficient size the answer is “you aren’t going to need it” at best and “you are doing activities that are wasteful” at worst.

That said, many organizations do not trust that this is possible and so set up an experiment to run both approaches in parallel for a while. While this is wasteful (you are using two different approaches to the same questions), and will slow down decision making (building the FTE plan takes time) and hence time to market, this might be the easiest way to get to comfortable with the lean-agile approach.

Want to Know More?

- What is the Impact of Context Switching on the Ability to Deliver? for effect of context switching on our capacity.
- What is the Best Approach to Making Decisions in Our Context? for a systems understanding how predictable we can be.

FAQ, FTE, Planning, Estimating, Portfolio

From:
https://www.hanssamios.com/dokuwiki/ - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:
https://www.hanssamios.com/dokuwiki/under_what_conditions_is_full-time_equivalent_fte_resource_planning_effective

Last update: **2024/10/10 06:50**