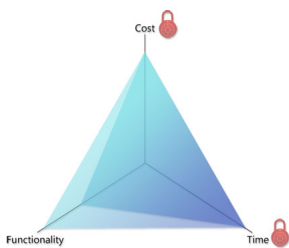


Great projects are all about meeting expectations. Our clients expect the best from us. Meeting these high expectations is a big part of why they choose us again and again on their projects. It also helps to explain how we have developed long term relationships with many organisations that feel more like partnerships than client/supplier transactions.

The trick is always about meeting each others expectations and I think the hardest part of any project is agreeing expectations that are realistic and acceptable to all parties. Anyone who has worked with Rubicon knows that we work really hard to **understand the constraints** affecting a project before it starts.

For me this is the key and I call it the "**Time-Cost-Functionality Triangle**".



You can always have a great project if you only lock two of the three constraints. In this scenario **great projects are delivered on time and to budget.** The trick is to realise that at least one of the three factors has to be allowed some freedom. In other words, **don't expect to fix time, cost and functionality.**

Before explaining how we aim to have "great projects", I'm going to explain how the Time-Cost-Functionality Triangle can help you to understand where things often start to go wrong.

What do you need to do?

If you think about any project you have been involved in, it will start with "We need to get this *thing*

done!". The "thing" is what I call functionality. It's the output of the project, the deliverable, the reason for the project, the goal and is normally the major factor in defining project expectations at the outset.

When do you want it?

What typical answers have you faced to this question?

- Now...(please!)
- Yesterday...(this is not helping!)
- As soon as possible (ok)
- As quickly as possible (that's not quite the same thing, is it?)
- By the end of the month (that's a drop-dead date, what happens if we slip?)

Each of these answers is subtly different in terms of expectations. For example, "As soon as possible" means the project should be delivered at the first possible date. This normally requires bringing in additional resources to maximise the rate at which the work can be completed. Of course there is a consequence in bringing these extra resources on to the project but that is part of really understanding what "As soon as possible" means.

Then there is "As quickly as possible" which could oddly mean you want to use the least number of resources. This is because "quick" can mean smallest amount of time recorded to the project (hence speed - doing the work in the smallest recorded time rather than "soon" - meaning smallest elapsed time).

It's at this point where you might think these semantics are just splitting hairs but the truth is, neither soon or quick are normally the constraints that end up determining the **real expectations** of the project.

Drop dead dates are very helpful for understanding constraints and setting expectations. If a great project means delivering before a drop dead date, then clearly **Time is the primary constraint**.

How much will it cost?

There are interesting dynamics around cost expectations too.

- I want your best price
- I want a fixed price
- I want to pay the least possible
- I want this to be delivered inside my budget

Of course, assuming high quality work, the "quickest" people are our most valuable people. This has an impact on cost. Loading a project with extra resources to meet a tight deadline can have significant cost implications which need to be identified before beginning. Fixing costs to meet a budget places constraints on what resources can be expended. Ultimately this has an impact on the complexity of the functionality and the timeframe in which the project will be delivered.

The difference between "Need" and "Want"

Once we start talking in these terms, it becomes clear that everything starts out as a "Want". These are the aspirations of the project. I am often involved in creative sessions where "blue sky thinking" is allowed to run riot. This can generate many "Wants"

- Oh yes, that would be excellent.
- We should have that.
- That's a good idea.
- We'd not thought of that before.
- We must have that. It's such a good idea.

The trouble is, these are *Wants*. When the project was outlined "This is the *thing* we need to do", many of the finer details and

want

aspirations had not be envisaged. The trick is to understand the

needs

; the things that enable you to have a great project and meet expectations without be drowned in scope creep and trailblazing.

So fix two!

In the Time-Cost-Functionality Triangle, I have always said to clients. You can fix two. I don't mind which two but you cannot fix all three. We end up with different projects depending on how this goes:

Functionality and Time - Fixed

This project has to work - price is a secondary concern. We have to be rigorous about the deliverables and the timeframe. We agree that there will be a contingency budget (Flexible Cost) to enable us to have a great project.

Time and Cost - Fixed

Our most common project ends up being about having real wins on budget and to time. We ensure that there is a contingency in the deliverable (Flexible Functionality) so that the *needs* are met and we squeeze in a few extra *wants* depending on how well the project has gone.

Functionality and Cost - Fixed

Everything at Rubicon is run as a project. Sometimes when we work on an internal R&D project, we will fix the deliverable and the investment necessary but, to enable us to do this work in our own time, we have contingency over the timetable (Flexible Time).

Where's the contingency?

Another way to think about setting expectations, is to decide where the contingency will be placed. The Time-Cost-Functionality Triangle dictates that you must have flexibility/contingency in at least one of the three areas.



I am reminded of the Bunco Booth 3-cups game. You have to have contingency somewhere in your project. However annoying it might be to find [the CEO may often say there can be no contingency] you must find a way to hide the contingency ball under one of the three cups.

If you do, you can ensure you have a great project.

{sharethis}