The Four Variables

Scope, Quality, Resources, Time

Projects are often given to developers in terms like these: "Take these four people, and get back here in three months with a perfect program". The developers ask "What does it have to do," and are given a huge list.

Often (it may seem like always), the developers kill themselves to produce something, but fall short. Either they don’t get done on time, or the program they produce is of low quality, i.e. it doesn’t make the customer happy.

Rarely, you get to add or remove people, but this doesn’t help much either.

Enlightened managers do it differently: they tell you the task and the time, and let you estimate resources. (Some even give you the resources you ask for.) Or they tell you the resources and the task, and let you estimate the time. (Some even give you the time you ask for.)

This turns out not to work either. You still kill yourself, and you’re still not done on time or else the customer isn’t happy.

What’s going wrong?

There are four variables you have to deal with: resources, time, quality, and scope. Here are examples of how they interact:

- Try to make more time using overtime. This never works. Quality suffers, and you wind up late anyway.
- Stick to your scope and quality goals. You wind up late.
- Try to make good decisions about what to leave out, but deliver on time. Customer isn’t happy with what you left out.
- Ask for more resources. You can't have them. You wind up late.
- Ask for more resources. You get them. Overhead buries you. You wind up late.

The end result is always the same: you deliver junk on time, or you deliver the expected program so late that no one remembers you. Either way, the customer isn't happy.

The reality is this: time and resources are probably fixed. Quality goals may be a little flexible, but only within narrow limits. The only thing that can be varied is scope. How can we vary scope to result in a perception of success rather than failure?

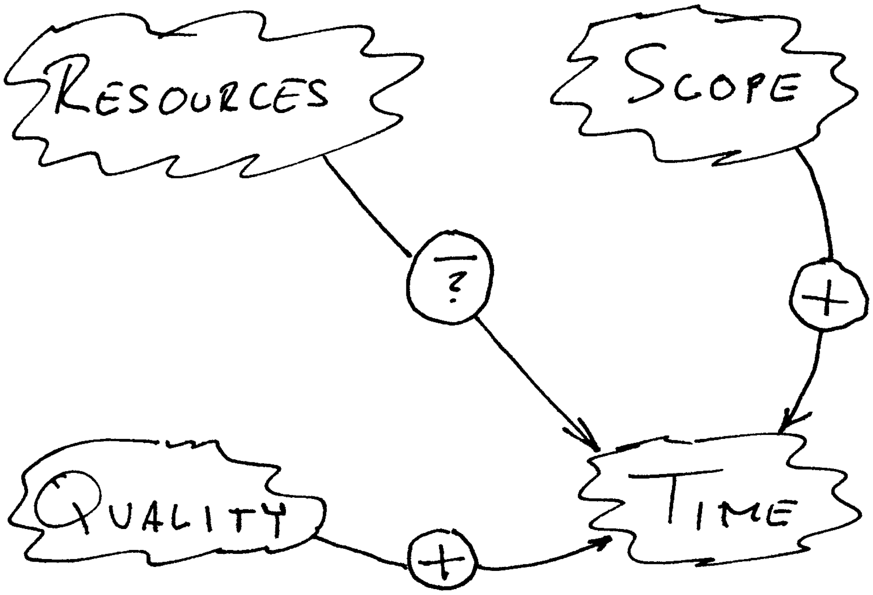
Review: the variables

There are four inter-related aspects to a software project:

- Scope - what is to be done; the features to be implemented;
- Quality - the requirements for correctness and other "good" things;
- Resources - the investment of personnel, equipment, and so on;

- Time - the duration of the project.

How are the variables related?



The primary relationships are shown in the picture above:

- Adding Scope (features) makes Time (duration) increase. Do more, take longer.
- Increasing Quality requirements makes Time increase. Do better, take longer.
- Increasing available Resources can sometimes make Time decrease. Rarely.

Reverse relationships hold. If you increase Quality requirements and hold Time constant, then you must add Resources (sometimes) or reduce Scope.

Similarly, if you want to decrease Time, you must do one or more of:

- Decrease Scope;
- Decrease Quality;
- Increase Resources.

Who owns the variables?

Finally, let’s examine who owns which variable:

- Scope is owned by the Customer, with some input from Management and Developer;
- Quality is owned by the Customer, perhaps again with input;
- Resources are owned by Management.

If Management and the Customer exercise control over Scope, Quality, and Resources, then Time is completely determined by the dynamics of the development process. It is a matter of measuring progress and predicting the final result. You hold your nose and do the math.

Conclusions

Learn how to measure and predict your progress. In a context of short iterations with defined goals, this is actually fairly easy. When you predict, measure, and report progress, Customer and Management can make good business decisions based on reliable predictions. And your credibility will be high, because everyone can see what is really happening.

Analyze results, determine what slows you down, and improve the process to help speed up. Don't make assumptions about the effect: measure results again and when there is a real effect, feed back into the predictions. This gives Customer and Management increasingly reliable information as time goes on.

The bottom line is that Customer and Management have direct control over all aspects of the project: Scope, Quality, Resources, and Time. You can adjust Time, for example, by tuning the other variables: and you can do it with high confidence in the results.

The customer can make the project go better by controlling Scope. Management can make the project go better by providing enough Resources and by protecting the team from Time-wasting activities. The developers can make the project go better by working as effectively as possible (the point of the rest of these pages), and by reporting accurate information on the Quality and Scope produced over Time.

You control your project. Make it be what you want, by controlling the variables.

[Home] [XP] [XP Practices Frame]

ronjeffries@acm.org
http://www.armaties.com