# Chapter 1

# *The Purpose of Planning*

*"Planning is everything. Plans are nothing."*
—Field Marshal Helmuth Graf von Moltke

Estimating and planning are critical to the success of any software development project of any size or consequence. Plans guide our investment decisions: We might initiate a specific project if we estimate it to take six months and ¤1 million[1] but would reject the same project if we thought it would take two years and ¤4 million. Plans help us know who needs to be available to work on a project during a given period. Plans help us know if a project is on track to deliver the functionality that users need and expect. Without plans we open our projects to any number of problems.

Yet planning is difficult, and plans are often wrong. Teams often respond to this by going to one of two extremes: They either do no planning at all, or they put so much effort into their plans that they become convinced that the plans must be right. The team that does no planning cannot answer the most basic questions, such as "When will you be done?" and "Can we schedule the product release for June?" The team that overplans deludes themselves into thinking that any plan can be "right." Their plan may be more thorough, but that does not necessarily mean it will be more accurate or useful.

That estimating and planning are difficult is not news. We've known it for a long time. In 1981, Barry Boehm drew the first version of what Steve McConnell

1. Remember that ¤ is the universal, generic currency symbol.

(1998) later called the "cone of uncertainty." Figure 1.1 shows Boehm's initial ranges of uncertainty at different points in a sequential development ("waterfall") process. The cone of uncertainty shows that during the feasibility phase of a project a schedule estimate is typically as far off as 60% to 160%. That is, a project expected to take 20 weeks could take anywhere from 12 to 32 weeks. After the requirements are written, the estimate might still be off +/- 15% in either direction. So an estimate of 20 weeks means work that takes 17 to 23 weeks.

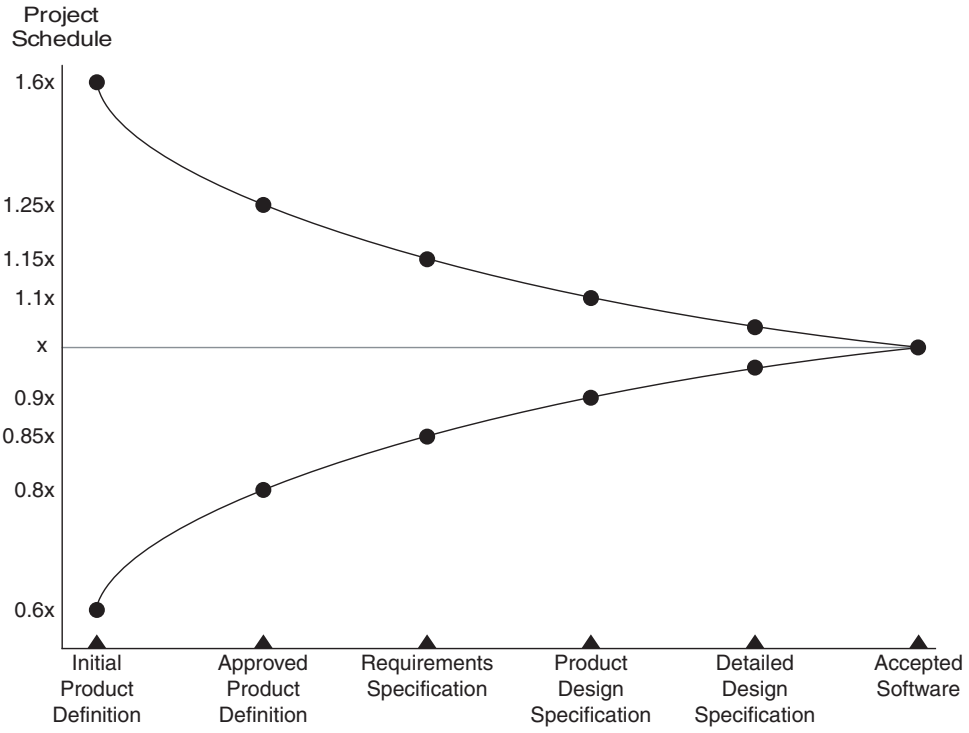


Figure 1.1 The cone of uncertainty narrows as the project progresses.

The Project Management Institute (PMI) presents a similar view on the progressive accuracy of estimates. However, rather than viewing the cone of uncertainty as symmetric, PMI views it as asymmetric. PMI suggests the creation of an initial *order of magnitude estimate*, which ranges from +75% to –25%. The next estimate to be created is the *budgetary estimate*, with a range of +25% to –10%, followed by the final *definitive estimate*, with a range of +10% to –5%.

## Why Do It?

If estimating and planning are difficult, and if it's impossible to get an accurate estimate until so late in a project, why do it at all? Clearly, there is the obvious reason that the organizations in which we work often demand that we provide estimates. Plans and schedules may be needed for a variety of legitimate reasons, such as planning marketing campaigns, scheduling product release activities, training internal users, and so on. These are important needs, and the difficulty of estimating a project does not excuse us from providing a plan or schedule that the organization can use for these purposes. However, beyond these perfunctory needs, there is a much more fundamental reason to take on the hard work of estimating and planning.

Estimating and planning are not just about determining an appropriate deadline or schedule. Planning—especially an ongoing iterative approach to planning—is a quest for value. Planning is an attempt to find an optimal solution to the overall product development question: What should we build? To answer this question, the team considers features, resources, and schedule. The question cannot be answered all at once. It must be answered iteratively and incrementally. At the start of a project we may decide that a product should contain a specific set of features and be released on August 31. But in June we may decide that a slightly later date with slightly more features will be better. Or we may decide that slightly sooner with slightly fewer features will be better.

A good planning process supports this by

- Reducing risk
- Reducing uncertainty
- Supporting better decision making
- Establishing trust
- Conveying information

## Reducing Risk

Planning increases the likelihood of project success by providing insights into the project's risks. Some projects are so risky that we may choose not to start once we've learned about the risks. Other projects may contain features whose risks can be contained by early attention.

The discussions that occur while estimating raise questions that expose potential dark corners of a project. Suppose you are asked to estimate how long it will take to integrate the new project with an existing mainframe legacy system

that you know nothing about. This will expose the integration features as a potential risk. The project team can opt to eliminate the risk right then by spending time learning about the legacy system. Or the risk can be noted and the estimate for the work either made larger or expressed as a range to account for the greater uncertainty and risk.

## Reducing Uncertainty

Throughout a project, the team is generating new capabilities in the product. They are also generating new knowledge—about the product, the technologies in use, and themselves as a team. It is critical that this new knowledge be acknowledged and factored into an iterative planning process that is designed to help a team refine their vision of the product. The most critical risk facing most projects is the risk of developing the wrong product. Yet this risk is entirely ignored on most projects. An agile approach to planning can dramatically reduce (and ideally eliminate) this risk.

The often-cited CHAOS studies (Standish 2001) define a successful project as on time, on budget, and with all features as initially specified. This is a dangerous definition because it fails to acknowledge that a feature that looked good when the project was started may not be worth its development cost once the team begins on the project. If I were to define a failed project, one of my criteria would certainly be "a project on which no one came up with any better ideas than what was on the initial list of requirements." We want to encourage projects on which investment, schedule, and feature decisions are periodically reassessed. A project that delivers all features on the initial plan is not necessarily a success. The product's users and customer would probably not be satisfied if wonderful new feature ideas had been rejected in favor of mediocre ones simply because the mediocre features were in the initial plan.

## Supporting Better Decision Making

Estimates and plans help us make decisions. How does an organization decide whether a particular project is worth doing if it does not have estimates of the value and the cost of the project? Beyond decisions about whether or not to start a project, estimates help us make sure we are working on the most valuable projects possible. Suppose an organization is considering two projects; one is estimated to make ¤1 million, and the second is estimated to make ¤2 million. First, the organization needs schedule and cost estimates to determine whether these projects are worth pursuing. Will the projects take so long that they miss a market window? Will the projects cost more than they'll make? Second, the

organization needs estimates and a plan so that it can decide which to pursue. The organization may be able to pursue one project, both projects, or neither if the costs are too high.

Organizations need estimates in order to make decisions beyond whether or not to start a project. Sometimes the staffing profile of a project can be more important than its schedule. For example, a project may not be worth starting if it will involve the time of the organization's chief architect, who is already fully committed on another project. However, if a plan can be developed that shows how to complete the new project without the involvement of this architect, the project may be worth starting.

Many of the decisions made while planning a project are tradeoff decisions. For example, on every project we make tradeoff decisions between development time and cost. Often the cheapest way to develop a system would be to hire one good programmer and allow her ten or twenty years to write the system, allowing her years of detouring to perhaps master the domain, become an expert in database administration, and so on. Obviously, though, we can rarely wait twenty years for a system, and so we engage teams. A team of thirty may spend a year (thirty person-years) developing what a lone programmer could have done in twenty. The development cost goes up, but the value of having the application nineteen years earlier justifies the increased cost.

We are constantly making similar tradeoff decisions between functionality and effort, cost, and time. Is a particular feature worth delaying the release? Should we hire one more developer so that a particular feature can be included in the upcoming release? Should we release in June or hold off until August and have more features? Should we buy this development tool? To make these decisions we need estimates of both the costs and benefits.

## Establishing Trust

Frequent reliable delivery of promised features builds trust between the developers of a product and the customers of that product. Reliable estimates enable reliable delivery. A customer needs estimates to make important prioritization and tradeoff decisions. Estimates also help a customer decide how much of a feature to develop. Rather than investing twenty days and getting everything, perhaps investing ten days of effort will yield 80% of the benefit. Customers are reluctant to make these types of tradeoff decisions early in a project unless the developers' estimates have proved trustworthy.

Reliable estimates benefit developers by allowing them to work at a sustainable pace. This leads to higher-quality code and fewer bugs. These, in turn, lead

back to more reliable estimates because less time is spent on highly unpredictable work such as bug fixing.

### Conveying Information

A plan conveys expectations and describes one possibility of what may come to pass over the course of a project. A plan does not guarantee an exact set of features on an exact date at a specified cost. A plan does, however, communicate and establish a set of baseline expectations. Far too often a plan is reduced to a single date, and all of the assumptions and expectations that led to that date are forgotten.

Suppose you ask me when a project will be done. I tell you seven months but provide no explanation of how I arrived at that duration. You should be skeptical of my estimate. Without additional information you have no way of determining whether I've thought about the question sufficiently or whether my estimate is realistic.

Suppose, instead, that I provide you a plan that estimates completion in seven to nine months, shows what work will be completed in the first one or two months, documents key assumptions, and establishes an approach for how we'll collaboratively measure progress. In this case you can look at my plan and draw conclusions about the confidence you should have in it.

## What Makes a Good Plan?

A good plan is one that stakeholders find sufficiently reliable that they can use it as the basis for making decisions. Early in a project, this may mean that the plan says that the product can be released in the third quarter, rather than the second, and that it will contain approximately a described set of features. Later in the project, to remain useful for decision making, this plan will need to be more precise.

Suppose you are estimating and planning a new release of the company's flagship product. You determine that the new version will be ready for release in six months. You create a plan that describes a set of features that are certain to be in the new version and another set of features that may or may not be included, depending on how well things progress.

Others in the company can use this plan to make decisions. They can prepare marketing materials, schedule an advertising campaign, allocate resources to assist with upgrading key customers, and so on. This plan is useful—as long

as it is somewhat predictive of what actually happens on the project. If development takes twelve months instead of the planned six, this was not a good plan.

However, if the project takes seven months instead of six, the plan was probably still useful. Yes, the plan was incorrect, and yes, it may have led to some slightly mistimed decisions. But a seven-month delivery of an estimated six-month project is generally not the end of the world and is certainly within the PMI's margin of error for a budgetary estimate. The plan, although inaccurate, was even more likely useful if we consider that it should have been updated regularly throughout the course of the project. In that case, the one-month late delivery should not have been a last-minute surprise to anyone.

## What Makes Planning Agile?

This book is about agile planning, not agile plans. Plans are documents or figures; they are snapshots of how we believe a project might unfold over an uncertain future. Planning is an activity. Agile planning shifts the emphasis from the plan to the planning.

Agile planning balances the effort and investment in planning with the knowledge that we will revise the plan through the course of the project. An agile plan is one that we are not only willing, but also eager to change. We don't want to change the plan just for the sake of changing, but we want to change because change means we've learned something or that we've avoided a mistake. We may have learned that users want more of this feature or that they want less of that feature or that usability is more important than we'd believed or that programming in this new language takes longer than we'd expected. The financial impact of each of these changes can be assessed and, if worthy, can alter the plan and schedule.

As we discover these things, they affect our plans. This means we need plans that are easily changed. This is why the planning becomes more important than the plan. The knowledge and insight we gain from planning persists long after one plan is torn up and a revised one put in its place. So an agile plan is one that is easy to change.

Just because we're changing the plan does not mean we change the dates. We may or may not do that. But if we learn we were wrong about some aspect of the target product and need to do something about it, the plan needs to change. There are many ways we can change the plan without changing the date. We can drop a feature, we can reduce the scope of a feature, we can possibly add people to the project, and so on.

Because we acknowledge that we cannot totally define a project at its outset, it is important that we do not perform all of a project's planning at the outset. Agile planning is spread more or less evenly across the duration of a project. Release planning sets the stage and is followed by a number of rounds of iteration planning, after which the entire process is repeated perhaps a handful of times on a project.

So in defining agile planning we find that it

- Is focused more on the planning than on the plan
- Encourages change
- Results in plans that are easily changed
- Is spread throughout the project

## Summary

Estimating and planning are critical, yet are difficult and error prone. We cannot excuse ourselves from these activities just because they are hard. Estimates given early in a project are far less accurate than those given later. This progressive refinement is shown in the *cone of uncertainty*.

The purpose of planning is to find an optimal answer to the overall product development question of what to build. The answer incorporates features, resources, and schedule. Answering this question is supported by a planning process that reduces risk, reduces uncertainty, supports reliable decision making, establishes trust, and conveys information.

A good plan is one that is sufficiently reliable that it can be used as the basis for making decisions about the product and the project. Agile planning is focused more on the planning than on the creation of a plan, encourages change, results in plans that are easily changed, and is spread throughout the project.

## Discussion Questions

1. This chapter started by making the claim that overplanning and doing no planning are equally dangerous. What is the right amount of planning on your current project?
2. What other reasons can you think of for planning?
3. Think of one or two of the most successful projects in which you have been involved. What role did planning play in those projects?