

# Familiar Metric Management: Leonardo da Vinci Knew Feasibility

When the King of France, Francis I, returned from his Italian expedition in 1516, the great painter, Leonardo da Vinci, came with him. Da Vinci is less well known as the engineer to the Duke of Milan and author of the earliest known drawing of a canal lock of a type still used today. The two considered a canal to link the Atlantic and the Mediterranean across southern France through Toulouse. Here there is a great natural corridor: the Garonne river flowing to Bordeaux on the Atlantic Ocean and the Aude river to the Mediterranean. Linking the two rivers meant digging a canal over the continental divide between them, at that point only 189 meters above sea level. That could be done by thousands of men with hand tools. But where would the water to operate the several dozen locks come from? The water in the Garonne and Aude rivers was well below this summit and steam power and pumps were still several hundred years in the future. The proposed canal was beyond the state of the art. The King and Leonardo recognized *feasibility* and deferred the vision.

During the ensuing 150 years local efforts all foundered on the issue of feasibility. It was not until 1662 that Pierre Paul Riquet saw the outline of the answer. In the Montagne Noire, the southern outcropping of the Massif Centrale, about 20 to 30 kilometers north of the likely canal route, there was water. Moreover, it was high-altitude water, averaging several hundred meters above the high point of the canal system. It could be dammed and it could flow downhill to the water input at the high point of the canal. This feed-water canal would have to follow an elevation contour line, falling only a meter or so per kilometer. The dams, the feed-water system, and the canal itself would be a difficult engineering accomplishment, as Riquet found out during the 20 years that followed, but it could be done. It was feasible. He and thousands of men and women did build the canal and the *Canal du Midi* still operates today. [fn]Ware just spent three weeks in southern France studying the canal from the initial water diversion point of the first stream down to the Mediterranean and then to the Atlantic. Larry.[/fn]

## Our ancestors were shrewd

Today we abandon many software development projects after several years of effort and sometimes the expenditure of tens of millions of dollars. There are many reasons for these failures, of course, but an important reason, is the failure to establish *feasibility* in the first place. Are we less shrewd than Francis I and Leonardo? Possibly, for they were exceptional men. Even then, however, they were heirs to several millennia of civil-engineering experience. Today, we in software development look back on only a single generation of experience with large systems. Forethought—the consideration of feasibility—seems not yet to have become accepted practice.

Why not? There have been several influences at work. One is that software came along after a period of remarkable engineering achievements: nuclear power, radar, jet aircraft, television— they all worked. We got in the mode of thinking that scientists and engineers could accomplish anything leaders could envision. Ergo, software people could make anything work, too. In fact, the results of their efforts often did! In the past 40 years software people have implemented billions of lines of software that have re-fashioned the worlds of business, industry, and government. But some of their efforts did not work. Now we have had enough failures to look back on the decision of Francis I and Leonardo with more appreciation.

## What feasibility means

In the broadest terms, feasibility means more than a sweeping vision of a system that would be “nice to have.” For instance, when President Ronald Reagan laid out his vision of the Strategic Defense Initiative in 1983, it would certainly have made the residents of the United States safe from intercontinental ballistic missiles. Unfortunately, the 10 million or so lines of source code estimated to be required were then beyond the state of the art at an error-free level of reliability. A not inconsiderable number of missiles—and one would have been a lot—would get through the shield.

The “Star Wars” episode illustrates the tendency of visionary leaders to confuse a “nice to have” vision with a *feasible* project. You can check out feasibility. A vision is more slippery. To move from vision to a project, an idea has to pass four checkpoints.

1. It has to be laid out, or “architected,” with sufficient thoroughness for those responsible for executing it as a project to know, in general, what they have to do.
2. Then, what has to be done has to be checked against the technical state of the art to establish that it is doable at this point in time. Not just doable by some ideal organization, but doable by the one you have at hand. And doable within the cost framework you can afford and the time limit your circumstances prescribe, such as getting to market as soon as the competition. Moreover, doable at the level of quality the application requirements impose.
3. In the case of a significant project in a new area, parts of the tentative architecture will usually be novel, that is, not previously worked out. These novel parts may be *major* risks, that is, failure to surmount them will make the system impossible to complete. These major risks must be reduced to manageable risks. Otherwise you cannot be sure the system is doable. Reducing major risks means establishing requirements in the risk area, analyzing possible solutions, and designing at least one valid solution (though it may not turn out to be the one eventually employed). Often the team trying to establish feasibility will find they need to code the core features of at least one design and try it out with live users.
4. There is one further issue to settle. Even if the vision can be carried out, is it worth doing? Will the business fundamentals pay off? We can do lots of things as a matter of technical capability that the market will not support, as businesses, large and small, demonstrate every day.

## Feasibility checks depend on metrics

Our Star Wars example may imply that feasibility checking is a high-level effort involving largescale thinkers. Of course, *thinking* never hurts, but it helps to have some numbers to think with. A major risk, for example, is not only a part of the vision that you don’t know for sure how to do; it is also a part you don’t know how to cost estimate, time schedule, and defect estimate. In consequence, major risk reduction means two things:

1. Laying out a potentially workable plan to do the work;
2. Carrying the plan far enough to make ballpark estimates of these basic metrics. (More precise estimates come later, when you know still more.) See Figure 1.

# Staffing Profile

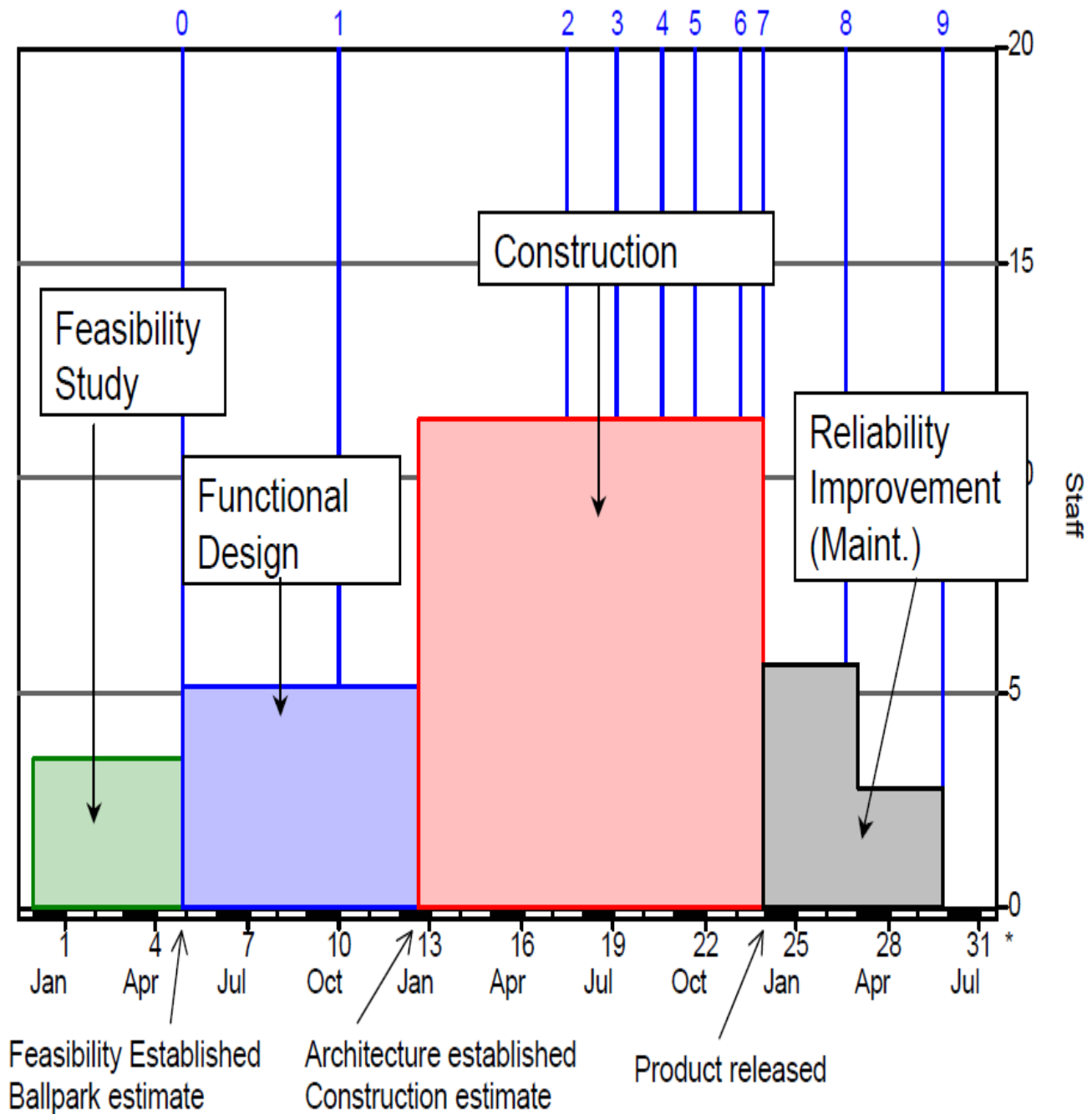


Figure 1. Until feasibility is established, there is no point to expending all that effort (money) in the later stages.

If you can't see a path through the major risks at a bearable cost in effort, schedule, and quality, the vision is *not*

feasible. Until you can find and deliver the water, you can't operate a system of canal locks.

In addition to reducing major risks singly, the feasibility team must also lay out a path through the entire vision. That means carrying the overall architecture far enough to make ballpark estimates for the entire project. This estimate is not the one with which you eventually start the main build. It is far too rough. It does tell you, however, that you can accomplish the vision within broad limits—limits you can afford. The project is *feasible*.

Beyond project metrics lies the idea of the business case. The business case fundamentally balances two sets of metrics. On the one scale is the cost of carrying out the project. On the other scale are the benefits expected. The benefits have to outweigh the costs—usually by a healthy margin, because these numbers are still broadband estimates.

The final application of metrics in the consideration of feasibility is estimating the cost and time parameters of the feasibility study itself. Because we cannot precisely define this stage, we cannot make a firm estimate of how much it will cost and how long it will take. Because a feasibility study usually involves only a few people (compared to the large number eventually employed on the construction phase) and they are often research or overhead people anyway, a close estimate is not essential. However, metric records of past feasibility studies can provide management with an idea of what the cost and time are likely to be.

## **Establishing feasibility takes schedule time**

Few will disagree that executive management ought to insist on a finding of feasibility before it commits funds for the main build. Yet this finding is often overlooked. The principal reason seems to be that studying feasibility takes time. When the vision is hot, we want to strike. Establishing feasibility delays the main build by months, sometimes years. In the case of software development, many managers like to see countable code piling up, even though no one has established that the final pile will work.

Actually, if the feasibility study is properly handled, these managers can see code in the feasibility stage. But that early code is not just a growing pile of ultimately nonfeasible code. It is the prototype code that comes out of the major-risk reduction efforts. It is the “proof of principle” demonstration code that establishes that the new and difficult parts of the vision are indeed feasible. There may not be much of this risk-reduction code, but it is critically important code. Creating a huge pile of easy-to-make code, as is the practice of “look the other way” projects, is no substitute. Under this pattern of project execution, the hard parts pile up at the end of the schedule in system test. At best the schedule slips. At worst, the responsible parties then find out, after the dribbling away of much time and money, that the vision was not feasible.

So, as Leonardo da Vinci might have said, “Managers, do the hard stuff first. Find the water. Establish feasibility.”