Various suggestions have been made in the past with the objective of better balancing the interests of clients and contractors in agile software projects. We are going to present some of these as examples in this chapter, briefly discussing them based on the three dimensions that we introduced earlier (variability of price and scope as well as risk sharing). One should keep in mind that an unlimited number of mixed forms is not merely conceivable but actually occurs in practice. The examples that follow are sufficient for our purpose, which is to determine the criteria that a fair model should meet.

13.1 Fixed Price per Iteration

When one thinks about agile software development and the challenging basic conditions of the business environment, one might think that nothing is fixed. Naturally this is particularly true of the scope. If we were able to describe it with sufficient accuracy in advance, then we probably would not consider anything other than a fixed price contract anyway. That is why compensation models that appear to take this initial situation into account—by making the price variable as well; T&M models in other words—seem to suggest themselves as an initial approach. But there are problems. Not only is the distribution of risk unfair, as illustrated above (because there is no risk distribution; the full risk is borne by the client)—another imbalance is built in as well: The contractor has an incentive to expend as much effort as possible, regardless of the result that is achieved, because it is paid for every person-day expended. It may be very late in the game before the client suspects that the supplier’s team is rather inefficient. But by that time, it is usually too late to engage in discussions about efficiency. Such a contract model trains the supplier to develop an interest in bloated software.
Fortunately, one of the characteristics of agile software projects is that they work with iterations, i.e. sub-projects providing results within defined time periods. This forms the basis of a rather simple contract design model—contracts according to the principle of a fixed price per iteration. The idea is simple: A fixed price is paid per iteration, the basic assumption being that every iteration takes the same amount of time and the development team is kept constant. Let us assume the team consists of five developers and a project manager. A rate of 800 € per person-day was agreed. This means a two-week iteration (or sprint) costs 10 working days × 5 persons × 800 €, which equals 48,000 €. Based on this calculation, one might think this is no different from a simple T&M model. Superficially and from a purely contractual perspective, this may be true, since the supplier makes no commitment about the attainable scope with this model, so both the value risk and development risk are borne by the client.

However, commercial reality often lies outside the legal construct. In this case, the simple fact is that the existence of iterations with defined time limits permits the frequent verification of success and promotes quick reviews and evaluations. The shorter the iteration and the smaller the team, the clearer the objective of the iteration is going to be before the start of the sprint. This means the client will be that much more disappointed if this objective is not achieved. Maybe not after the first, but most likely after the second and definitely after the third iteration, the discussion will not only turn to underestimated complexity in this scenario but also to the efficiency of the team. While the risks formally remain on the client side, the contractor will develop a vital interest in measures that improve efficiency and quality—especially when the client not only looks disappointed but angry as well. Certainly it becomes more difficult for the client to change suppliers the further the project has progressed (which is the actual threat and therefore the motivator for the contractor). However, the contract model with a fixed price per iteration [also known as “progressive contracts” (Larman and Vodde 2010)] ensures that it is unlikely to be “too late” to change suppliers, because early escalation is built into the model. Insofar as this mechanism lying outside the legal framework works, it will ensure that at least some of the development risk is shifted to the contractor side, and perhaps even some of the value risk.

13.2 **Fixed Price per (Whatever) Point**

Under the fixed price per iteration model, we have a discrepancy between the formal contract provisions and the resulting commercial behavior of the parties to the contract. Ultimately this can lead to the desired effect of shifting part of the risk to the contractor. Therefore, would it not make sense to find a more direct way of accomplishing this, putting into place contract provisions to ensure that the contractor earns less if it builds software of lesser quality, and more if it delivers better software? (Or, since “worse quality” and “better quality” are difficult to define clearly in a contract, if the supplier achieves a greater or lesser scope per iteration?)