

## Interactive Development

Software development is challenging: many projects over-run in cost and especially in time, and the solutions developed may not match the requirements. Many larger companies and governmental organisations demand fixed price developments to protect against cost over-runs, but this approach demands a rigid specification, which seldom is exactly what is really needed. Also where the client passes the risks and uncertainties to the developer the client ensures that the price will have to contain a large safety margin and thus the client may well pay more than is necessary.

The method we call **Interactive Development** is designed to avoid these problems.

But first let us look more closely at the problems of the "traditional fixed price software development". These are some of the issues:

Stage 1: The starting point is the requirement. The client knows they want "something" but exactly what? So someone within the client organisation is given the task of writing a specification and they will commonly form a small committee of interested people to help them. This approach has the following weaknesses:

- The person writing the specification may not be skilled in writing specifications, which is a vital but uncommon skill;
- The others involved may not pay much attention to the draft specification if they are busy with other work;
- Requirements will be included with little idea of their cost implication; small changes that could be accommodated easily within the business might result in significant cost savings;
- At this stage no one knows what the best solution will be as **the best solution can only be discovered through a learning process with a degree of trial and error**. Overall there is a weakness in that the specification is being written by people whose main expertise may not be the development of business processes.

Stage 2: The specification is given to the potential suppliers and they are asked to bid a fixed price.

Now the following issues arise:

- The suppliers may see some of the cost implications and potential problems but not be given the opportunity to discuss them and have the specification improved;
- There may be misunderstandings because of weaknesses in the way in which the specification is written;
- The potential suppliers do not want to appear too argumentative and "difficult" as this stage by asking too many questions in case they deter the client from awarding them the contract;

- The only ways the potential supplier can handle these uncertainties is to build a substantial margin into the price to accommodate the uncertainties, or they agree only to produce exactly to the specification and require formal change notices and increments to the contract to handle the issues that have not been anticipated adequately. Thus the problems that are unavoidable in the specification lead to increased cost or delay or, commonly, both.

Stage 3: The system is delivered and staff at the client start to use it. Now:

- Several of the key users are forced to think about the issues and find that they want changes made. These changes might be difficult to implement given the way the system has been designed. If these aspects of the requirement had been known earlier they could have been accommodated easily.

So a supplementary specification is needed to cover the changes and another fixed price is requested - this time with only one bidder. The cost of the changes is likely to be high, not only for the reasons given earlier about building in a margin for uncertainties but also because the client is locked in to the supplier.

So in summary the traditional method is weak in the following areas:

- It does not provide a learning process to get the best solution;
- It results in significantly higher costs than necessary
- It results in a longer timescale to reach an adequate solution

How can these problems be avoided? In reality there is no perfect solution for software development or any other development - it is always a balance of different uncertainties and risks. But we believe that if there is a reasonable degree of trust between the parties based on the supplier's track record then there is a better way to do things. And this way is facilitated by the use of software development platforms such as RunMyProcess.

How does our Interactive development method differ?

- It is based on a higher level of trust between the parties but the client is safeguarded by being able to cancel the project at short notice and with minimal penalties;
- We discuss the client's needs in depth and then write the specification based on a combination of the client's needs and our knowledge of their cost implications and impact on system design;
- We demonstrate the system while it is being developed to obtain feedback and collect ideas for worthwhile improvements. We then update the specification immediately and build in the improvements. This is the learning process, which is critical to the real success of the project. We find that the client only finds out what they really need when they start to use a system - this leads to many "discoveries" in terms of additional valuable features and produces much added value for the client. After demonstrating the basic operation of the number portability system in Bermuda, the operators asked for many additions, which we

added relatively easily. The system grew incrementally to be the equivalent of a high functionality central database, but developed at a small fraction of the cost of a traditional solution.

How do we keep our cost slow and our flexibility high?

- We use only cloud based generic platforms such as Google Documents and RunMyProcess, thus we avoid high software licence fees;
- We operate with very low overheads using offices at home and modern communications such as Skype video conferencing and screen sharing.

We believe that this method, which we call **Interactive Development**, is by far the best method to develop customised process solutions for smaller companies and organisations.