How to pick the right engagement model for remote software development

Picking the right engagement model is an almost absolute requirement for success when using outside resources in software development. And the model needs to be right for both client and provider.

With the right engagement model, expectations on both sides will be aligned giving the best possible background for mutually beneficial cooperation and creation of value.

With the wrong engagement model, commercial and technical cooperation will be difficult and is unlikely to get you closer to your business objectives.

Fundamentally, two things define an engagement model:

1. Expected duration of the cooperation
2. Distribution of management responsibility

Short- or long-term expectations?

Expected duration of cooperation decides whether relationship is focus on partnership or on the transaction itself. In a longer-term **partnership** model, the assumption is that we are creating value **together**. Both parties will take the interests of one another into consideration and show certain flexibility. You will only take the interests of the partner into consideration if you feel that you are not going to be cheated – so, **trust** is at the core of any partnership engagement. And a feeling that we have lots of common interests.

Building up trust takes time – which is why partnerships are more suitable for long-term product development than one-off tasks. **Transaction** based engagements are often **focused on price**. The underlying assumption is that your gain is my loss. A relationship structured as an argument – and to win the argument, we need to document obligations to be able to prove our rights. Our common interests are few.

The basic transaction based model is **project outsourcing**. The delivery of fixed functionality at fixed (lowest) price. The ability to freeze well-documented requirements allows the client to conduct tenders and select the cheapest bidder.

The danger is that from the moment of signing, interests diverge: the client will have an interest to demand requirements interpreted to the widest functionality possible while the provider will be interested in getting away with as narrow functionality as possible. Discussions may take place that would not happen if you are both planning to work together also next year.

Daily management by whom?

For obvious reasons, project outsourcing works better for smaller projects that do not pose questions during development. In project outsourcing, implementation is managed by the provider. The use of **freelancers** is another transaction based engagement model.

However one with less focus on upfront specifications. For this to work, **project management and coordination is the responsibility of the client**. Communication is much more frequent than in project outsourcing. In most cases, the spirit of a freelance engagement is transaction oriented and the relationship is temporary.

Team extension models work by expanding the existing organization of the client with extra resources and/or skills based at the provider. **Project management remains with the client.** Invoicing is on hourly or monthly basis. This is long-term cooperation that creates incentives for the client to transfer knowledge to individual team members. Knowledge that will increase innovation and productivity – especially in the long run. When enough knowledge has been transferred in a team extension model, the team will often be capable of working independently on functionality.

This is the **(self-)managed** or **dedicated team** model. The interface with the client is typically with the "product owner" rather than with a project manager. The managed team requires the biggest investment for transfer of knowledge but has the least client-side overhead for management, communications and control. It has the potential for creating the most value.

\---------

Jesper Lindholt, CEO at Livatek