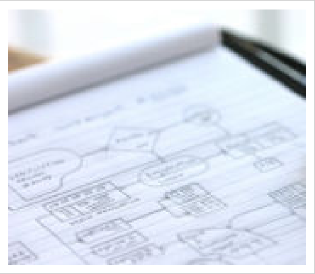
PROJECTSMART
.CO.UK

Dealing with "Scope Creep" in Software Development Projects

By Linda Russell

Scope creep is a significant risk in software development projects. We discuss why this is so, and how to avoid or at least mitigate the risk.



What is Scope Creep?

New software is usually developed as a result of a customer (which may be an internal or an external organisation) identifying a need. The next step is to specify how the software will meet that need; specifically, what functionality will be developed. This is the "scope" of the project. The project plans are drawn up, based on the estimates for developing and delivering the specified functionality, and an end date is agreed.

Development starts and the project seems to be progressing well. But then the customer realises that there are additional requirements they forgot to mention, or extra elements of functionality that they need. Often, adding these extras will cause the project duration to be extended, resulting in missed deadlines and increased costs, leading to erosion of margin on the project and potentially customer dissatisfaction and loss of credibility due to late delivery.

How to Deal With Scope Creep

It is important that a functional specification is produced at the outset, written in terms that the customer can understand. For example, a walk-through of the process that the software will support, perhaps illustrated with mocked-up screen shots, will help to clarify how the new system will work from the user's point of view.

The functional specification must be agreed and signed by the customer, and should include a Scope Statement. This states that only the functionality which is explicitly described in the specification is included in the project scope, and that anything not described is "outside scope."

When the customer subsequently identifies additional elements, reference is made to the specification: is the required functionality described or alluded to? If it is not, then the new development is outside scope.

The next step is to work out the impact of developing the new functionality: what extra effort will be required? What effect will this have on the overall project duration? What additional costs will be incurred and how will this affect the project margin?

If the impact is trivial, it may be agreed to include the new functionality in the existing project, but ideally this should be stated in writing by issuing a revised specification. The danger here is that the customer believes that a precedent has been set and that further revisions will be made in a similar way: it is important to communicate the reasons for allowing the revision in this instance.

It is more likely that the additional development will cause delay and/or extra cost. The customer needs to be made aware of the implications of the revision in terms of its impact on timescales and costs, and a specification of the additions and changes should be written (with its own Scope Statement). It is then up to the customer to decide whether they are willing to pay more, and if they can accept the revised end date for the project. If they agree, the new specification should be signed as before.

Do we Really Need a Scope Statement?

You may think that writing a sufficiently detailed specification to be able to make the Scope Statement would involve more time (and cost) than is warranted by the value of the project as a whole. For instance, if the whole project is expected to only take a few weeks and it would take 5 days to write a detailed specification, a cost/benefit analysis would show that it is not worth doing.

If this is the case, assess the likelihood of the risk (based on your knowledge of the customer and how confident you are that all the requirements have been identified) and the possible impact, and build in sufficient contingency in your estimates of time and cost to cover all but the most major revisions to the specification.

Linda Russell has an M.A. (with Distinction) in Technical Authorship, and over 25 years' experience in software implementation and consultancy. She has worked on the 4c programme, project and resource management software product for nearly 10 years. For details of the 4c application, visit http://www.4csys.com