



Scope Creep Management

Babu Suresh

Scope Creep is not only Inevitable – It's Natural

Every IT project is executed with a set of deliverables, and has an expected closure time. Prior to this closure period, there are a predetermined set of tasks and activities to complete the project successfully. These tasks constitute the **scope** of a project. Since a project schedule is closely tied to the delivery time line and the scope, a little variation in the scope can affect delivery and in turn affect the success of the project.

This inching forward of scope to introduce more requirements that are not included in the initial planning of the project whilst maintaining the same time frame for project delivery, is called **Scope Creep**. Scope Creep is the pejorative name given to the natural process by which clients discover what they really want.

The scope creep can be classified based on the users who creates these changes:

- Business Scope Creep
- Technology Scope Creep

Business Scope Creep

Systems are designed to solve the business needs for a company. Due to continual changes in market trends, the requirements that are defined before, might change. The common reasons for these changes are:

1. Insufficient Requirements Analysis Definition resulting in business requirements that are not well defined.
2. Underestimating the complexity of the problem in an unknown industry.
3. Management failure in managing user expectations.
4. Involving the users only in later stages of project life cycle such as programming and testing.

In an IT project, regardless of whether it is outsourced or built in-house, the project team works with the client to gather the requirements. This requirement definition analysis phase involves meetings, interviews, and questionnaires with the client about the current system and their future needs. In most cases, clients are unable to specify exactly what they want in the beginning until they see the product. It is also often difficult for business users to visualize how the new system will be until they see it.

When the users do see the new system for the first time, changes may be needed because any new applications will be initially unfamiliar to users. Most of the times, the user perspective is to always look for things that won't work, rather than the things that do work in the system. The approach the business users have in mind is that,

“We're spending so much time and money anyway, so let's add this during the testing phase”.

This expands the scope way beyond what you can accomplish or really need.

Solution to Business Scope Creep

1. Define the business requirements as “**must-haves**” and “**nice to haves**” and prioritize them. Identify the risks for each “must-have” requirement and get the stakeholders approval. Plan these prioritized requirements in the form of phased deliverables during the project life cycle.
2. Set project expectations with the customer stakeholders and get the buy in from the customer.
3. Decide and document the agreed project deliverables in the **Statement Of Work (SOW)** document and requirement areas that are NOT included.
4. Document requirements and review with the customers before any sign off.
5. Decide and document how the users will use the system in the form of test cases during the requirement analysis phase.
6. Make a flexible project plan allowing users to participate at the design phase and incorporate their suggestions. In case scope creep cannot be avoided, participate in rescoping.
7. Introduce a formal change management process that would allow the users to define the requests as “**Your Enhancement Submission**” (YES) form. It is surprising how effectively this cuts out low priority demands, when users have to initiate a change requisition. Follow the six steps for any changes or deviations from the initial set of requirements
 - Record
 - Assess
 - Plan
 - Build
 - Implement
 - Close.
8. Do an impact analysis and attach a cost and time for the new requirements. This is effective in getting the sponsor to revalidate the new requirements.

Technology Scope Creep

The scope creep created by the technologists can be broadly classified into two categories –

“Customer Pleasing”

The project team or an individual who wants to please the customer and is reluctant to say “no” to a change in the requirements.

Solution

1. The project team’s responsibility is to let the business know that the requested change is considerably different from the requirements approved during the project scoping process. The team should provide the business sponsors with the options and explain to them how these

changes could impact the budget, timelines and resources. The options are

- i. Integrating the new set of requirements in a different phase
 - ii. Stop the project so that new additional requirements can be properly scoped and integrated rather than tacked on.
 - iii. Continue the project without rescoping.
2. Since the user can visualise the system, perform a visual walkthrough session during the requirements phase to define what the client wants before the system is built. This iterative **Prototyping Approach** or **Joint Application Development (JAD)** session with the client can help the team to identify the features and can deliver a final product close to the client's needs and will result in project success.

“Technical Gold-Plating”

The programmers/developers who adds features and functionality that have not been specified in the approved requirements definition. The reason for these changes could be the business requirements are lacking the details or the programmer is a perfectionist.

Solution:

1. Specify the “must-have” requirements in the form of a checklist and track them through the development process. This process would help to check the deliverables from the developer.
2. Because they play a crucial role in the process, involve the developers during the Requirements Management stage to prevent the team from starting with incomplete or ambiguous requirements. To decrease the risk of destabilizing the project, the developer needs to incorporate the changes that have been approved by the team so that he knows that he is working on agreed upon functionality.
3. Introduce a **Change Control Board (CCB)** team that would evaluate the risk of implementing the changes done by the developers. The team would categorize the risk as high, medium, low and define a process that could capture this kind of requirements in the early stages of the project. The team should suggest a reward mechanism for the developer if the feature introduced by developer gets implemented.

Conclusion

The conclusion from this article is that “scope creep” is always a change or growth of project scope. Instead of preventing the changes completely, we should work as a team to effectively manage the changes by not affecting the project timelines and budget.



Suresh Babu is an architect specializing in SOA (Service Oriented Architecture), Web Services and FDLC based (Feature Development Life Cycle) project management. He has more than seven years of experience in all phases of software development life cycle and provided various J2EE based solutions in consumer banking business. His area of expertise includes Object Oriented Analysis & Design, UML, J2EE/J2ME, XML/XSLT, WebSphere, EAI and FileNet. You can reach him at babus@citifinancial.com.

Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell “Project Administrator” software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called “Method H”™, and sell software to support the technique. For more information on Project tools or Project Management visit www.projectperfect.com.au