



March 6, 2001

Sound Requirements Management Makes Validation Easier

Margo Visitacion

Catalyst

A client inquiry

Question

Which recognized methods/best practices exist in gathering and specifying requirements for the targeted IT solution and for making a decision for the best solution (cost, risk, potential). We understand that there are methods for requirements management that are deployed in an object-oriented development context, but we feel that the level of detail of the results is too fine for later use than simply buying a software package.

Answer

While some techniques of requirements management for internally developed applications, such as modeling, may be too detail oriented for implementing commercial off-the-shelf (COTS) software, companies should not take acquisition decisions lightly. As with any project, practicing requirements management determines user and system requirements, integration issues as well as long-term support issues. Companies considering COTS implementation projects should not skimp on requirements management and instead should use it as an opportunity to validate that any selection will fully support business processes and enhance overall system value.

In gathering requirements for COTS projects, the project team should adopt a two-pronged approach, considering the business requirements, including usability and cost feasibility, and technical requirements, including integration or customization requirements and compatibility or support requirements. This information, gathered via several different processes, benefits the project team by allowing them to see the risks and benefits to acquiring a COTS application, as detailed in the table below.

Requirements Methods: Benefits and Risks

Requirement Attribute	Process	Benefit	Risks to Consider
Use Cases – High Level	Interviews	Gets user perspective on what they want to do and how they want to do it	<ul style="list-style-type: none"> • Keep it at the user perspective • Must be understood and validated by users • Mapped to testing • Focus on real vs. perceived
Use Case – Low Level	Interviews and System Analysis	<ul style="list-style-type: none"> • Understand integration issues within system • Determine reuse abilities • Determine change 	Understand how user interacts with the system

management procedures			
Checklists – Business Oriented	Interviews and Prototypes	Determines value of acquisition	<ul style="list-style-type: none"> • Does product have wide audience base? Will it be around for a while? • Support/service quality? • Standards based?
Checklists – Technically Oriented	Prototypes/ System Analysis	Determines technical feasibility of acquisition	<ul style="list-style-type: none"> • Are the interfaces open or closed? • Ease of integration – does it affect ongoing support costs? • No. of platforms supported and frequency of upgrades? • Are upgrades released w/ platform upgrades?

Source: Giga Information Group

Recommendations

- Standard requirements management techniques, such as interviewing and prototyping, are excellent ways to determine how well an application will fit into system architecture (see IdeaByte, [Defining Pilots, Prototypes and Proof-of-Concepts](#), Uttam Narsu).
- Consider using a requirements management tool to organize and maintain COTS requirements tools. Having a repository that tracks every aspect of the requirements process makes managing change much more effective (see IdeaByte, [Change Control Starts With Prototypes — Even in E-Business](#), Margo Visitacion, and IdeaByte, [Use Your Requirements Management Tools to Assist in Developing Prototypes](#), Margo Visitacion).
- Fold COTS requirements management into a standard RM process to make sure that standards are maintained in the process (see Planning Assumption, [Implementing Requirements Management](#), Carl Zetie).