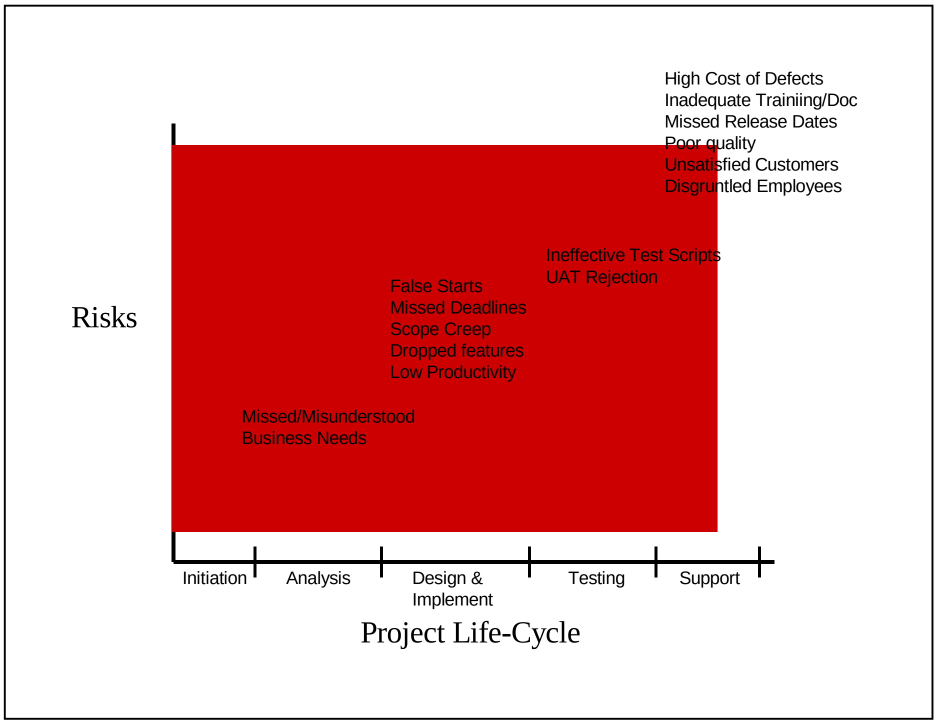
Providing a Framework for Project Success….and Beyond

Conventional wisdom says you will never reach your destination if you don't know where you're going. However, all too often, that's exactly how projects are implemented, most often as a result of the business and technical staff being unable to communicate their needs to each other. This communication gap usually occurs early in a project, and the negative impacts manifest themselves throughout the entire life-cycle. Extensive re-work, scope creep, missed deadlines, low morale, and required features being dropped from product releases are just some of the issues that are costly and can be traced back to a fundamental lack of understanding of the business needs.



For the Project Manager, unclear requirements are the cause of mounting frustration throughout the life-cycle. Everything may seem fine – goals were set, the team was built, all the necessary tracking and management tools were in place. A technology was selected, requirements gathered and signed off, and the development staff was off and coding. Then it starts to become apparent everything wasn't fine after all.

Faced with unclear requirements, a development staff has two options; either go back to the business and gain a better understanding of their needs, or implement best-guess. These "false starts" are time consuming for both staffs. Requirements may also be uncovered that result in a significant re-evaluation of work effort (usually more, not less). Implementing best-guess will post-pone the issue until later in the life-cycle – at user acceptance testing it will become apparent to the business that they are not being delivered what they paid for. Obviously, these are both situations the Project Manager will want to avoid.

So what can be done? The project manager must incorporate practices that:
(1) Get at the true requirements

(2) Enforce tracking and management of requirements throughout the life-cycle
(3) Allow for a structured approach to the review and approval process
(4) Contribute to future projects so that valuable assets are not lost.

Each of these four critical success factors is discussed next.

Critical Success Factor #1: Eliciting the True Requirements

The key to uncovering real requirements (and by that we mean those that meet true needs) is three-fold. You must have an efficient and effective ***process*** to guide the way requirements are gathered and documented; your process must include an ***architectural structure*** for capturing requirements information so that all parties understand what they are trying to document; the process must emphasize the use of ***models*** as a mechanism to gather the information and communicate with all stakeholders – the sponsor, the subject matter experts, and the developers

The process you follow must provide a foundation for gathering, documenting, and maintaining requirements that eliminates the communication gap between IT and the business. We believe that this can be achieved only by using communication tools and techniques that both IT and the business understand, including visual models. These models, while detailing complex business processes and system behavior, are not themselves complex to understand or create, thus the business is as comfortable participating in their creation and review as the IT staff.

The emphasis on using models to elicit requirements is in contrast to processes and tools that would have you build requirements by extracting text statements from documents. It is true that some requirements – particularly initial business objectives – lend themselves to text statements. Text statements are also a good way to capture operational requirements such as the number of users that must be supported, or the necessary levels of security.

But process requirements, system functional requirements, user interaction, and information requirements can best be understood using models. It might take several pages of text to describe the process of, say, taking an order. Even then, you risk missing some key business rule or forgetting to document the necessity for using a particular piece of data. A visual approach is clearly preferable to a written one – if you've ever been part of a building project and reviewed architectural drawings, you'll know exactly what we mean when we say that visual models are superior to text descriptions.

Critical Success Factor #2: Tracking and Managing Requirements

Once you've gathered requirements, you must have a process for tracking them and managing changes to them. It's a given that there will be changes over the life of a project, especially for complex or lengthy projects; why not plan for them?

The process you follow must incorporate tools and techniques to meet this goal. First, you must be able to track the status of each requirement. Is it approved, or only under review? Second, you must be able to trace the requirements from the origination through to their development. What was the business objective that caused this requirement to be stated? How will this requirement be realized? Third, you must know the impact of a proposed change. If this change is accepted, what other requirements will be affected and do these also need to change? Finally, you must know the history of requirements changes. How was this requirement initially stated, when did it change and why?

Critical Success Factor #3: Structuring the Review Process

Review and approval of requirements can be cumbersome and difficult to control. To be successful, the project manager must be able to make review assignments to users or groups of users and track comments that have been made. It may seem obvious, but performing review cycles via the web clearly has advantages over paper-based approaches. Your approach should include sharing and review of the information via the web. In that way, you can support any number of reviewers in any number of locations.

Critical Success Factor #4: Moving Beyond the Project – Reuse

Once a project is complete – that is the requirements have been realized – then what? Typically, there's some kind of documentation, but that's it. But if your process has reuse as one of its embedded disciplines, you can reap many advantages beyond the project close date. The process must include techniques for making the artifacts of one project available for the future. In this way, your process can provide support for *continuous* business improvement – not just one project at a time.

Reuse of project artifacts can serve multiple goals. The most obvious is that the requirements artifacts can serve as documentation of the delivered functionality. If an enhancement is requested, the requirements artifacts can be used to perform impact analysis and to judge the appropriateness, cost, and complexity of implementing the request. The second goal is to provide a jump-start to subsequent projects. For example, the deliverables generated from understanding and modeling the claims process can apply to other projects that touch the claims area. Finally, the models developed for a project can become input to developing an architecture of information that can be shared enterprise-wide.

To fulfill these four important critical success factors, Doreen Evans Associates has developed its ***LINK*Process**™ and ***LINK*Suite**™ capabilities. These are explained next.

LINK*Process™ from Doreen Evans Associates*

There are many good methodologies out there, yet we felt each was missing some crucial component or another. Either they were not model-based, or they did not address the critical task of managing requirements throughout the life-cycle, or they did not include a mechanism for the review and approval process. That's why we developed ***LINK*Process**, a comprehensive, complete, repeatable, and teachable approach that can be used for many types of projects. ***LINK*Process** is focused on requirements – their elicitation, their management, and their approval. Yet requirements apply to many types of situations, not just to software development. We've applied ***LINK*Process** to projects focusing on:

- Building and managing enterprise architectures
- Understanding and improving business processes
- Changing organizations
- Developing new systems or enhancing existing ones
- Exploring new technologies
- Developing new product offerings

There are four basic principles of the approach:

(L) Lead with Business Understanding. We never launch a requirements effort without first understanding the business process that the solution will be supporting. Our first goal is to prevent the wrong solution from being developed and to ensure the solution meets real needs. Even in a system development project, it's possible that what really needs to be changed is the process itself, or that a simple technology change such as a scanning capability can bring significant improvement. Understanding the business process provides a context for all other project work. It can help discover business policy and business rules that may be appropriate for a rules engine. It also uncovers what are known as "process rules", that is, those rules that outline when things should occur and in what order. This understanding can become the basis of a business process management effort.

(I) Identify Critical Requirements. We follow a systematic approach to identifying the requirements that the business deems important. We use repository-based tools to capture the requirements and develop the architecture (sometimes called the "meta model") so that users of the process know exactly what needs to be documented, where it needs to be documented, and how it needs to be documented. The meta model provides a template structure that makes it easier to gather information, to build relationships among the information, and to produce project documentation based on the information.

(N) Navigate the Deliverables. We architect the business analysis and requirements deliverables so that they can be used in many ways through the life of a project and beyond. For example, use cases and data models can be transferred to an object-oriented development tool to jump-start the design effort. The business policy and business rules identified can be separated from other deliverables and used in development of a rules engine. The process flows can be exported to the business process management language (BPML) and used for business process management efforts, or they can help drive workflow rules. System requirements can generate user acceptance test scripts.

(K) Keep and Manage a Requirements Architecture Repository System. We store all the deliverables created during business process analysis and requirements analysis in a repository. This means the artifacts can:

- Be communicated to all project participants and published on the web.
- Be tracked – we know what they trace from and to, we know their change history, we know their status.
- Jump-start subsequent work. The next time a system enhancement is contemplated, you already have the requirements that represent the current system to start your work.
- Provide information to an Enterprise Architecture Repository System. Each project produces deliverables – business process maps, organization architectures - that are candidates for updating your Enterprise Architecture. ***LINK*Process** includes steps to support and manage this critical aspect of the work, which we believe is a unique benefit of the approach.

***LINK*Process** includes a standard set of work products and deliverables for each life-cycle phase. This ensures that each project is analyzed to the appropriate level, and that no important component is overlooked. As requirements are gathered, they are modeled using the ***LINK*Process** methodology and captured in a repository. This repository stores visual models and all the project business and system rules. The methodology uses industry-standard framework that define the artifacts that should be collected. And, because the methodology is intended to be used with a repository-based modeling tool, each artifact has its place, its required properties, and its relationships to other artifacts.

Throughout the project life-cycle, modifications to the project resulting from Change Requests are updated in the repository. Therefore, at the end of the project, the repository serves as an accurate representation of the delivered functionality.

Finally, the artifacts you collect during the project are intended to become candidates for company assets. They are not lost in documents, but rather are gathered according to the architectural framework and made available to be used, shared, and communicated to all interested parties.

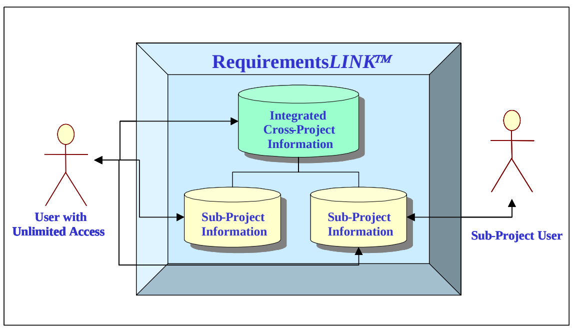
The LINKSuite from Doreen Evans Associates

By making project artifacts such as architectural models and definitions available through a web interface, ***LINK*Suite** allows for wide dissemination and sharing of information. We use an enterprise architecture modeling tool to create artifacts. But when the artifacts need to be shared and reviewed, we publish models and definitions from the modeling tool using an intermediate extract and transform application. This application uses an API to access the modeling tool's repository information, apply transformation rules and import it to the publishing repository, a highly scalable and optimized SQL Server 2000 database which supports archiving and versioning of published information. A highly intuitive interface and navigation, coupled with comprehensive search and reporting features, allows a user to explore and find any type of information published.

While many modeling tools generate html, users can only view the information, they cannot interact with it. ***LINK*** allows users to make comments, raise issues, and create reports. In this way ***LINK*** becomes a true collaboration tool, not just a viewer.

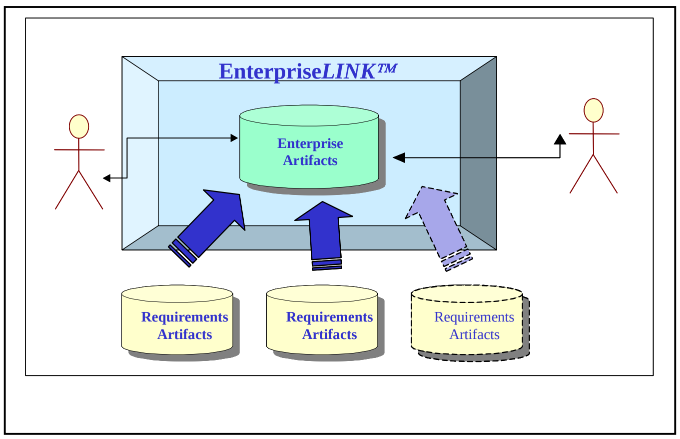
Configuring the LINK*Suite*

Depending on your needs, the ***LINK*Suite** can be configured in different ways. Let's take a simple example, where a team is working on developing requirements for a systems development project. In this case, the tool would be configured to contain information on the requirements for data, for functions, for the user interface, for the operational support, for security, for user roles, and so forth. It would be set up so that any users designated by the team could view and comment on the information. One team member would be responsible for making sure that change requests were managed, analyzed, incorporated appropriately, and then re-published. When configured this way, we call the tool **Requirements*LINK***™.



For complex requirements projects where work is divided into sub-projects, **Requirements*LINK*** could be configured to contain multiple components, one for each sub-project's information, and one for information that needs to be integrated across the project. Cross-project information might consist of a full logical data model or the complete application architecture – anything that ALL sub-projects would need to understand and incorporate.

Requirements*LINK* has an integrated change management and approval workflow feature, which can be configured to allow users to be organized into groups for collaboration, comment, resolution, and approval of models, definitions, issues, and other relevant published items. Once you are ready for a review cycle, **Requirements*LINK*** reads the models you have built, interprets the links, understands the parent-child and peer-to-peer relationships, and then builds text requirements. It also provides archiving and versioning of published information. It is intended to support work-in-progress, and deliverables that are undergoing change during review and approval.



What about the enterprise architecture information – that is, information developed and approved as the result of projects but that applies to the enterprise over the long term? In this case, we configure what we call **Enterprise*LINK*** and use it to support business managers who want to understand how their business areas function, or data architects who want to see information on the enterprise data and its uses, or the CIO who needs to understand the systems that the enterprise uses and how they are used.

Information in **Enterprise*LINK*** can be built up over time by taking approved artifacts from various **Requirements*LINK*** project work and promoting it to the enterprise level. The information here is approved and stable, although it can of course change over time as business processes are improved or as systems are added.

What Are the Benefits of LINK*Process and the* LINK*Suite?*

For the business, the ***LINK*Process** methodology ensures that the business process drives project results. This alone would make it valuable. But ***LINK*Process** combined with the ***LINK*Suite** also provides the business process in a repository – available to serve as a 'coach' in how to carry out the process, as a way to start the next project, and as part of an enterprise architecture.

For IT, the ***LINK*Process** methodology can shorten development time because it prevents false starts (you know what the business wants before you start designing). Even more important, IT can be confident that the solution they develop will meet real business needs.

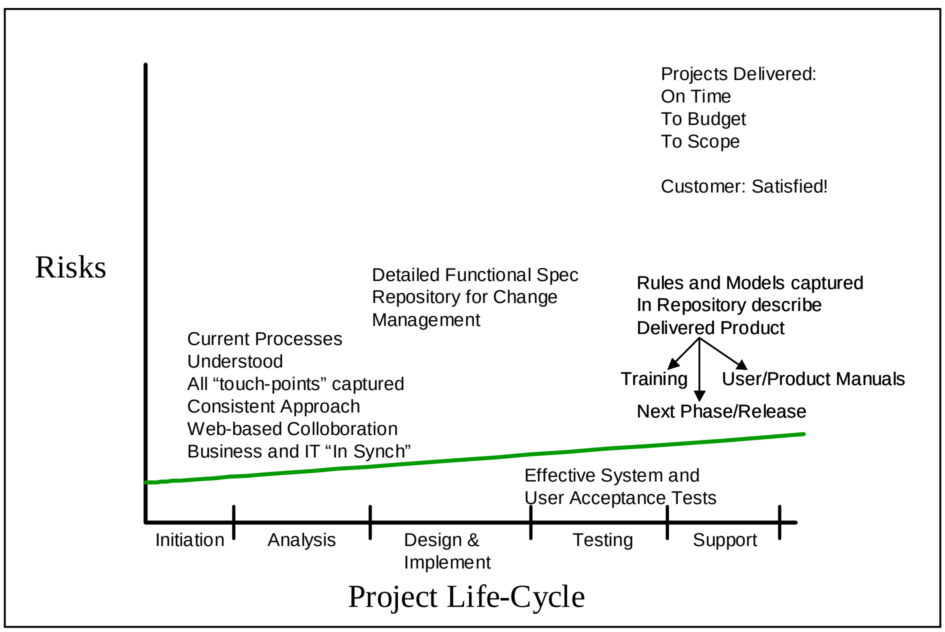
For the Project Manager, ***LINK*Process** provides exceptional benefits throughout the life-cycle – minimizing risks while providing a central framework for describing what's being proposed, developed, and delivered.

The benefits include:

1. A common communication tool between business and IT, allowing for:
   - Elimination of re-work and false-starts due to unclear requirements
   - Minimization of change requests
   - Reduction in friction due to frustration that arises from miscommunications between IT and Business

2. Consistent, controlled approach:
   - Clear roadmap (setting expectations) on how requirements are uncovered, communicated, maintained
   - Web-based collaboration indicates who and when requirements are reviewed

3. Allows trainers, documenters, testers to become involved early in life-cycle:
   - Provides clear understanding of level of documentation and training needed and ability to deliver when needed.
   - Provides clear understanding of functionality to be delivered, allowing QA to create necessary test scripts to ensure full product testing.



4. Living repository: Repository holds the required functionality, updated throughout the project to capture any changes, and provides an accurate representation of the product upon delivery. The repository can then be used for subsequent projects/phases as the "As-Is" representation of the business and to feed approved artifacts to an enterprise architecture.

About Doreen Evans Associates

Doreen Evans Associates (DEA) is a professional services firm that focuses on business process improvement. We can help you change a process, build an enterprise architecture, or define requirements for your systems and technologies. Founded in 1992 as a woman-owned, privately-held small business, our mission is to ensure that business need drives solutions.

31 St. James Avenue • Suite 210 • Boston, MA 02116 • (617) 482-4444 • www.doreenevans.com