

**A StickyMinds.com Original  
November 2007**



**Ready, Fire, Aim**  
**How Timely Interface Analysis Reduces Risk in Software Projects**

By [Mary Gorman](#)

**Summary:** Analyzing a project’s interface requirements often starts late and focuses-- sometimes exclusively--on creating a snazzy user interface. But failing to conduct interface analysis in a timely fashion increases the risk of project delays, overruns, and even failure. In this column, Mary Gorman makes the case for investing in interface analysis by explaining what it is and how it reduces the risk in software projects. She also describes best-practice models that help teams identify and validate crucial elements in all types of interfaces that may exist in software applications.

In my work I see software teams take unneeded risks when it comes to building interfaces. Do you recognize any of these?

- Spending tons of time on the look and feel of a user interface at the expense of other requirements
- Ignoring or deferring system-to-system interface requirements until the design phase or later
- Confusing which system “owns” the interfaces, creating delays in critical, cross-project planning
- Segregating interface requirements from analysis models

You can imagine the results. You end up with redundant, conflicting, or missing requirements, and therefore heightened risks, delays, overruns, and even project failure.

Many of these problems stem from a single cause: failing to analyze your product’s interfaces before you design them. So, what’s the difference between analysis and design?

Interface *analysis* explores, identifies, models, and specifies the user and software requirements needed to support connections between your product and external components. It is the “what” of interfaces. In contrast, interface *design* focuses on the “how.” For most of us, figuring out the “how” is what we like to do. It’s like a technical puzzle, and solving technical puzzles is what we’re good at. But it’s a mistake to dive into problem solving before you conduct a rigorous analysis of which problem you need to solve. More formally, before you worry about design, you need to validate that the interfaces you have in mind satisfy the business goals.

**Which Is Which?**

The key is to differentiate analysis results from design results. These may vary by project, methodology, and availability of resources. Here are a few examples:

Type of Interface	Interface Analysis	Interface Design
User interface	Navigation, storyboards	Detailed layout, style guide, widgets
Report	General layout, data	Rules for grouping, breaking, style guide
System to system	List of data	Detailed transmission layout
Hardware device	Purpose of device	Message specifications, timing

Good interface analysis boils down to a few best practices. Let’s take a look at those practices.

**Plan Ahead**

The first step in interface analysis is to understand the system’s boundaries, and a best practice is to build a model. For example, a context diagram shows interfaces as arrows between the system and the external components (people, other systems, hardware). With this knowledge, a project manager and an analyst can begin discussions with the representatives or owners of the components. The goal is to reach an agreement on which interface elements should get priority and how to budget and schedule the work.

### **Integrate, Integrate, Integrate**

Analyzing before designing helps you avoid putting all your eggs in one basket. You can do this by building models, which you should start within the first weeks of a project.

Building models with multiple views, such as dynamic, behavioral, structural (data), and control (business rule) views, gives you different perspectives that help you discover and cross-verify requirements. All these views may be integrated with interface requirements in some fashion. For example, data appearing on a customer's UI window may also be summarized in an executive report and even sent to another application via a system-to-system interface.

When you elicit and define the system's business and temporal events, you uncover triggering interfaces as well as interfaces that are part of the system responses. (See my related column, "[Events to the Rescue](#).") You can create stories or use cases to understand the behavioral requirements of various interfaces.

Of course, everybody loves prototypes. They give you a picture that's worth a thousand words. But most prototypes don't uncover the user requirements that lie behind the scenes, such as requirements about data and business rules. For that you need the other views.

### **Budgeting Benefits From Interface Analysis**

Estimating project costs is often more art than science. Yet understanding the costs may help the business side more effectively prioritize the requirements. You can improve your estimates by knowing the number and types of interfaces early on. Keep in mind that budgeting down to the individual interface level is difficult without a valid model.

There might also be some interface costs that may include special resources that you might not think of without the discipline imposed by a model. For example, if you're building complex user interfaces, you may need significant time in a usability lab. In this case, you will need to budget for that expense as well as schedule time in the lab.

Leveraging a logical data model, a detailed data dictionary, and detailed business rules not only can speed interface analysis tasks, but also can improve the quality of the interface requirements. Conversely, you can increase the value of the logical models by incorporating the interfaces' data and rules, thus helping build a non-redundant set of requirements.

### **Good Old Project P**

Make sure that you're studying *all* the interfaces during interface analysis: user interfaces, system-to-system interfaces, and external hardware device interfaces. If you wait until the design phase to understand the basics of system-to-system and hardware interfaces, you'll be taking an unnecessary risk. In the example below, we learn what happened to Project P's team, which waited to map out the interfaces until after the design phase.

The project P team, working on a Web application, invested many resources over many months building a detailed prototype. Representatives of the future user community participated in numerous reviews. The business owner signed off on the navigation and the screen layout and expected the new application to be up and running shortly.

The development team quickly built the presentation layer, but things fell apart when they tried to go deeper. The culprit? An interface with an existing system owned key data needed by project P. In the rush to prototype the user interface, the team had neglected the analysis of the system-to-system interface.

They got it right—eventually. Project P's manager negotiated with the application's manager for urgent support, but the project was still tremendously delayed.

### **A for Analysis**

It may be difficult to allocate limited assets to analyze interfaces early in a project, but it's an investment that keeps on giving.

### **Author's Appreciation**

My thanks go to Ellen Gottesdiener and Meilir Page-Jones for their helpful review of this column.

### **Further Reading**

- *Customer-Centered Products: Creating Successful Products Through Smart Requirements Management*, by Ivy Hooks and Kristin Farry.
- "[Events to the Rescue](#)," by Mary Gorman, October 2006 StickyMinds (access when signing up to Success with Requirements)
- [The Software Requirements Memory Jogger: A Pocket Guide to Help Software and Business Teams Develop and Manage Requirements](#), by Ellen Gottesdiener.

**About the Author**

Mary Gorman, CBAP and senior associate at EBG Consulting, helps project teams explore, analyze, and build robust business and system requirements models. In September she presented “Integrating Interface Analysis into Your Project: Just-enough, Just-in-time” at SD Best Practices, Boston, and in November at ProjectWorld & the World Congress for Business Analysts, Anaheim. Mary serves on the Body of Knowledge (IIBA BOK) committee of the International Institute of Business Analysis and is the leader of the Requirements Elicitation subcommittee. She can be reached at [mary@ebgconsulting.com](mailto:mary@ebgconsulting.com) and [www.ebgconsulting.com](http://www.ebgconsulting.com).