The process of prototyping—from creating simple wireframes to testing fully functional mockups—is one of the most potent and powerful set of skills any designer can master. It’s also fraught with peril in workplaces where the process is skipped in lieu of just “designing a prototype” as a simple deliverable to give to the next department to build. No matter how diligent your business is with prototyping, the actual process can often make or break your final product. How and why to actually build a prototype is often a mystery. Ask many designers and they’ll tilt their heads like confused puppies when you create performance models, you identify application scenarios and your performance objectives. Your performance objectives are your measurable criteria, such as response time, throughput (how much work in how much time), and resource utilization (CPU, memory, disk I/O, and network I/O). You break down your performance scenarios into steps and assign performance budgets. A process that helps you incrementally define and capture the information that helps the teams working on your solution to focus on using, capturing, and sharing the appropriate information. To use this performance model, do the following: 1. Set goals. Validate your model and estimates. 2. Continue to create prototypes and measure the performance of the use cases by capturing metrics. 4. Prototyping may be used as an alternative to the systems development life cycle. 5. Guidelines for developing a prototype are: A. Work in manageable modules. Sometimes COTS software may be the quickest way to create a prototype. 8. Systems analysts must work systematically to elicit and evaluate users’ reactions to the prototype. There are three ways the user is involved: A. Experimenting with the prototype. B. Reduce process learning time and dual processing losses. C. Reduce time and effort to structure tasks and format outputs. D. Reduce nonproductive expansion of work.