

## UI PROTOTYPING

User interface prototyping supports rapid exploration of system requirements and user interface design.

<b>Main Benefits</b>	The main benefits of ui prototyping are risk reduction, decreased system size and complexity, and reduced requirements change.
<b>Keys to Success</b>	Keys to success include deciding whether to evolve the prototype or throw it away at the beginning of the project, implementing the simplest possible solution that will achieve the goal, and involving end-users.
<b>When to Use</b>	Use ui prototyping whenever there is a risk of developing an unacceptable user interface and having to rework it later
<b>Main Risks</b>	User-interface prototyping is usually successful. It will have many of the same risks as either throwaway prototyping or evolutionary prototyping, depending on which route you've chosen.

### Overview

In user-interface prototyping, the user interface is developed quickly to explore the user-interface design and the system's requirements. Sometimes a low fidelity mockup is used; other times a special-purpose prototyping language or the target programming language is used.

User-interface prototypes are either thrown away or evolved into the final product. Making the right choice about whether to evolve the prototype or throw it away is critical to ensuring successful prototyping.

In either case, engineers should strive for creating a Hollywood facade implementation of the prototyped features. Ui prototyping focuses on exploring requirements. Since aren't yet definitely in the product it's not worth investing significant development time in the prototype.

To use user interface prototyping successfully, it is important to involve end-users and actively soliciting their feedback. Since the implementation is smoke and mirrors, its essential to limit their interaction with the prototype to controlled settings.

### CxOne Support

CxOne provides support for user interface prototyping with a prototyping checklist.

### Interactions with other Best Practices

User-interface prototyping is an effective remedy for feature-creep when used with other practices. One study found that the combination of prototyping and JAD were capable of keeping creeping requirements below 5 percent. Average projects experience levels closer to 25 percent (Jones 1994).

User-interface prototyping interacts with either throwaway prototyping or evolutionary prototyping, depending on which kind of user-interface prototype you develop.

Setting clear goals and objectives or using miniature milestones to monitor progress are important aids to keeping prototyping efforts focused.

## Further Reading

McConnell, Steve. *Rapid Development*. Redmond, WA: Microsoft Press. 1996.

*CxBest\_EvolutionaryPrototyping*

*CxBest\_ThrowawayPrototyping*

*CxBest\_MiniatureMilestones*

*CxBest\_JointApplicationDevelopment*