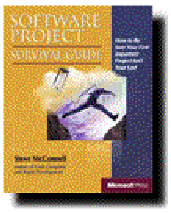
SAMPLE RISK MANAGEMENT PLAN

Risk Management Plan for "Creeping Requirements"

Why?	Our analysis found that the average requirements overrun on our projects is about 40%. We need to control creeping requirements to prevent uncontrolled cost and schedule increases on the project.
How?	In general, we need to look for ways to eliminate the source of requirements changes by doing a good job of gathering requirements in the first place. After that, we need to be sure to allow only those requirements changes that are absolutely necessary.
What?	We are addressing the risk in three specific ways: 1. We're using a **user interface prototype** at the beginning of the project to be sure we gather high-quality requirements. We will continue showing the prototype to the users, refining it, and showing the prototype to the users again until we are confident that they will be very happy with the software we build. 2. We're **placing the requirements specification under explicit change control**. After we complete the user interface prototype and gather other requirements, we'll baseline the requirements. After that, requirements changes will have to go through a more formal change process in which cost, schedule, quality and other impacts have to be carefully assessed before the change is accepted. 3. We're using a **staged delivery approach** to keep the delivery cycles short, which reduces the need for changes within cycles. Between stages we can change features if needed. We'll upgrade this risk to a higher level if any of the following conditions become true: • We can't get users to buy into a user interface prototype within a reasonable amount of time. • We receive requests for requirements changes constituting more than 5% of the system in the first 30 days after the requirements have been baselined. • We actually accept requirements changes constituting more than 5% of the system at any point in the project lifecycle.
Who?	The **engineering lead** is responsible for the user interface prototype. The **change board** is responsible for maintaining the requirements under change control. The **project manager** is responsible for keeping the stages within our staged delivery plan short.



When?	We'd like to have the UI prototype complete by 4/15. If it isn't complete by 6/1, we'll upgrade the severity of this risk to "project critical." The requirements spec should be baselined by 5/15. If it hasn't been baselined by 6/15, we'll upgrade the severity of this risk to "project critical." We should have completed our first staged delivery by 7/15. If it hasn't been completed by 8/15, we'll upgrade the severity of this risk to "project critical."
How much?	We estimate the UI prototype will cost 6 engineering staff months. Explicit change control is accounted for in our standard development practices and does not add cost to the project. Staged delivery increases the apparent project cost by about 5% because of the increased effort associated with releasing the software multiple times, but it reduces integration risk and the risk of building the wrong product. In the end the only increase is probably in the visibility of project's true cost, so it is a net gain rather than a cost.