



# Tips for Capturing “Lessons Learned”

5 Questions to Answer with Your Team

- 1. Have a series of “lessons learned” meetings with your team,** as your project is proceeding — or no later than right after it's over. What were the challenges and good points of the project? For example, if you had technical glitches, information gaps, or communication snags, where, when, and how did they occur? On the flip side, what aspects seemed to go really well?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- 2. Did you develop any useful *workarounds* or *solutions* to problems** that cropped up during the project? Document the details in a way that will make sense later. For example, if two systems that were supposed to work together didn't work as expected, and you figured out a resolution, capture that information so it will be understandable by people on future projects.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- 3. For any problems that went unresolved, what *preventative measures*** can you invent now that could help things go more smoothly next time? For example, if you had problems with complex formats, why not prepare some easy-to-use *templates* for next time? If testing steps were overlooked, why not create *checklists* to guide people through the process step by step?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- 4. Are there any new “*best practices*” you can derive from this project?** Note anything that went so well — and now seems to be so thoroughly “road tested” — that you would want to repeat the positive experience next time.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- 5. Can you create an *easily accessible repository* for lessons learned** and best practices you have documented? This might be a database, Web site, or even a simple document. It doesn't have to be fancy to be effective.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_