How To Run An Iteration Retrospective
by Joshua Kerievsky

The best time to conduct an Iteration Retrospective is just before conducting your Iteration Planning Game. Your first Iteration Retrospective will likely take over an hour to complete. Don't worry about the time. As you continue to conduct these retrospectives, they'll take less and less time. Below are some steps for conducting your first Iteration Retrospective, followed by steps for conducting future retrospectives:

Your First Iteration Retrospective:

- Assemble your whole team (i.e. programmers, subject matter experts, QA engineers, marketing and sales people, technical support) and agree on Norm Kerth's Prime Directive of Retrospectives: *Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand.*

- Create an Iteration Retrospective Poster and write on it (in very big letters) the words "Worked Well" in one column and "Needs Improvement" in another column. Make this poster publicly visible to the whole team in the team's open space.

- Ask the team to suggest things that worked well during the iteration or that need improvement and write these items in the respective columns on the poster. You'll discuss the items after you write them all down – so avoid the temptation to discuss any particular item during this step – but do seek clarification if a suggested item's meaning is confusing.

- Now discuss each item, particularly how the team might address the items in the "Needs Improvement" column. Have courage when you consider ways to fix problems and remember that some problems often reveal how certain XP practices and values aren't being fully implemented or embraced.

Future Iteration Retrospectives:

Once you've done your first Iteration Retrospective, you'll have data to work from for your future Iteration Retrospectives. Revisit that data before you obtain new items for your poster. Identify items on the "Needs Improvement" column that have been sufficiently addressed to be removed from that column and then remove them. Also find out why other items need to remain on the "Needs Improvement" column and discuss how those items might get resolved during the next iteration.