Typical Sprint Review Model at my job is the four-square model similar to what is beneath:

|  |  |
| --- | --- |
| Accolades:  Name: Good job xyz for abc!  Name: Thanks for covering me on Thursday! | Stakeholder Feedback:  This is primarily left blank as they never show up to these meetings anyways. |
| What did we do well:   * Name1 * Item 1 * Item 2 * Name1 * Item 1 | What could have been better:   * Name1 * Item 1 * Item 2 * Name1 * Item 1 |

Then as a team we go through each one of these boxes and discuss the talking points. Usually, eighty-five percent of this time is spent on the what could have been better box. Typically, these are specifically what we did not like and how to fix it. Personally speaking, most of these are arguing semantics. We once had a forty-five-minute discussion over #comment vs. # comment in python. A few more examples of actual issues that have arisen were knowledge transfer, passing off tickets to one another, management not making up their minds, and other such general problems that I do not remember anymore.

After the problems are identified we discuss possible solutions. For knowledge transfer, it was saving changes to a branch, emailing the new assignee, AND tagging the new assignee in the ticket’s comments. Another issue was management not being able to decide on a configuration. So, to solve this was to make all the possible options. Lots of work but it worked well. We also have action items that need to be implemented by the end of the next sprint. Action items are typically code styles. Lines no longer than 80 characters, comment styles, code styling such as if statements being multiple lines versus one when possible.

In order to complete user stories, we would have working meetings, usually on an as needed basis once a week to fill out user stories, flesh out the tickets, review, and estimate them.

And finally, the pros and cons of the agile process. I am going to be honest here I have seen nothing but pros until recently. Effectively the flexibility of it all ultimately works to the team’s benefit. The one thing I do not agree with, in any way, shape, or form, is the continual band-aids that are continuously requested. For example, management/stake holders want everything planned out, place holders made, test processes, the whole nine yards. Unfortunately, for me, this is not how I work. I can write a complete test case if I have an articulable artifact to use. What they were asking was for people to use a smart phone to send a text message without the phone. That is a huge con to me. But overall, my experience with AGILE has been positive. I enjoy the recurrence, the back tracking repeatedly to make sure everything is working as intended.