Quote:
It's not about creating translation (current system is working great) as contributing documentation.
Someone has to write the documentation directly in the source code. That means write access is
needed, plus someone has to test that it still compiles.

At present, people use topic++ right? They save the doc in external files. What I said: do exactly
the same process. The document manager (not his subordinates) can have a look into the topic++
work (what his subordinate, those without write access to the code, have done) than, if he think
that all is ok, he presses a topic++ button and all validated modifications are synchronized back
into the source file, ready for doxygen and U++ final release.

Quote:
If user doesn't have write access, somebody needs to merge the changes for him.

Yes, modifications has to be validated. U++ API can break at any time before final release. Two
eyes are better than one. You keep total control of the final documentation submitted but your
contributors have full access to the topic++ generated documentation.

Quote:
Also, after every doc commit, people will have to recompile their sources and run doxygen on
them. And if you distribute prebuilt documentation, you basically are back to square one.

If they don't want to edit the topic++ output, and want the last doxygen output then yes. What
prevent them from using topic++ to see the unstable doc? Nothing. How could the topic++
documentation break the source if the integration process is automated by topic++?