

# Maintaining patches is fixing patches

Egon Willighagen 

Published October 21, 2009

## Citation

Willighagen, E. (2009). Maintaining patches is fixing patches. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/7862g-njh70>

## Abstract

Today I had a question about having to fix patches against upstream changes because those patches were not included upstream yet is not very productive.

## Copyright

Copyright © Egon Willighagen 2009. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## chem-bla-ics

Today I had a question about having to fix patches against upstream changes because those patches were not included upstream yet is *not very productive*.

However, it is a prominent part of maintaining a code base. In the past 9 year, I *and many others* have been reworking a lot of [CDK](#) code because of API changes **and** bug fixes in deeper parts of the CDK library. At least half of the work I have done for the CDK is doing this kind of fixing of downstream code. This is *never* trivial, and it is never productive. Well, depends somewhat on your definition of productivity.

Whether productive or not, it is just something that needs to happen. Additionally, it is **not** something you can prevent. I guess one can call this a *fact of life*. Doesn't make it nice work. Not at all. And most of my frustration with the CDK library is the lack of documentation and unit testing, which makes such fixing of downstream code hard. This means that the person best suited to do this job, is the one who wrote the patch in the first place. The person who made the comment I mentioned earlier is seeing this from very up close now.

## Code Quality

I very much understand his feeling of being unproductive when updating patches; been there, done that. He (that I can disclose) is absolutely right. With all the quality assurance functionality I have set up in the past for the CDK, nicely integrated in [Rajarshi's Nightly](#) script, I hope to make it easier for people to write proper maintainable patches. Often these reports are, however, again about doing tasks which make you feel unproductive. But I can assure you that writing such tools quality assurance tools, like the [OpenJavaDocCheck](#) I worked on this weekend, makes you feel even less productive.

## Redesign

Sometimes making a library better maintainable, includes reworking the design. Almost always this take serious effort, and potentially introduce new bugs. At the same time, it always fixes a lot of older bugs and at the same time, of redesigned properly, makes it much easier to fix other bugs and allow more functionality to be implemented.

But again, this requires rewriting of downstream patches too. And the one doing the redesign will always get comments about this requiring to make unproductive code updates downstream. I have seen this on several occasions in the CDK, such as my rewrite of the atom typing functionality in the CDK. (And don't get any KDE4 developer started on that topic ;) Another *fact of life*, I guess.