

The Craft of Code: Practices and Knowledge in the Production of Software

Pierre Depaz – pierre.depaz@sorbonne-nouvelle.fr

2021.03.29 edits

First, I clarified the license of the illustration used as CC2-Attribution-NonCommercial.

To strengthen the conclusion, I added two sentences:

"In this sense, conceptions of beauty and usability would gain from being extended from a short-term (readable and enjoyable by readers of the code) to a long-term one (legible and enjoyable by all). Our comparison with architecture is once again useful: while architects might hold themselves to the highest standard, external forces such as real-estate development, social segregation and budget stringency overwhelmingly thwart the attempts at creating beautiful spaces for all."

and:

"As code becomes the defining medium of the early 21st century, it also reminds us of the possibility of the essential quality, and the positive social outcomes of a work well-done, rather than relying on the assumption that poor creation doesn't affect anyone, since it is invisible to most."

As for the references, I went ahead and removed a lot of those which seemed to me either to re-iterate a point previously made, or refer to a point that was only tangential in the development of the argument.

- removed #7: Antony Black, Guilds, since the two sources essentially making the same point.
- removed #11: John Pannabecker on Diderot, since it was more focused towards formalization of implicit knowledge, which is implied in the rest of the sentence.
- removed #12: Robert Gordon, Who Turned the Mechanical Ideal, since it made the same point as the following reference, but in broader terms.
- removed #15: Neal Stephenson, In the beginning was the command line, as it is a specific inquiry into a particular tool, which is also covered in Levy, Hackers.
- removed #18: Dijkstra is already cited in the more relevant "Craftsman or Scientist?"
- removed #21: The psychological aspect of programming is on the edge of the subject
- removed #23,24,25 since they are just offshoots of #22
- removed #29: There is plenty of literature around the fact that CS education is trying to be formalized (i.e. the very existence of CS academic programs in universities across the world).

- removed #30: The description of the activity of pair-programming is explicit enough (programming with four hands).
- removed #33: Understanding code snippets is perhaps a little too specific for the scope of the article.
- removed #35: since #34 also deals with it, except from a Chinese perspective rather than a Japanese perspective, and historically, Japanese culture has tended to be influenced by mainland Chinese, rather than the other way around.
- removed #37 since it is about the educational value of software-reading, and the point on copyright is more important in that respect.
- removed #38, since it was only about backing up the 1980s date for the copyrightability of software.
- removed #39, since Gabriel's work is already mentioned, and touches upon the same topic
- consolidated the various references to the same works (41-42)
- removed #45: Artificial intelligence is only a secondary point in the article.
- removed #49: The reference to LISP might be a bit technical.
- removed #50: Reference the environmental impact of computing, since it isn't the main point of the section.