

expanded: LLM-chat  
allows questioning  
the way I would a  
fellow dev, rather  
than flexing google-  
fu skills

For implementing problem  
solving without a TDD  
approach, it's difficult to  
know whether or not my  
interactions with the model  
are going in a valuable  
direction

Expanded:  
Freeing my  
mind from the  
burden of  
'plumbing'

contracted: sometimes  
feel I'm wasting time  
finding the best way to  
get co-pilot to solve it  
instead solving the  
problem in the "best"  
way

contracted: decentralized  
vs. centralized API design --  
centralized is now  
qualitatively better because  
it is more amenable to  
generation



Copy the Stars!



expanded:  
thinking a lot  
more about  
behavioural  
code analysis



Expanded: I've been  
thinking a lot about it  
being a useful  
teammate in getting  
code implemented via  
TDD once I've laid out  
the requirements.

expanded: learned  
how to work with a  
company library +  
framework in a  
fraction of the time



Expanded: I'm able to  
apply pattern-based  
knowledge and skills  
while utilizing newer  
languages /  
frameworks more  
easily



because it is inherently  
potentially "inaccurate and  
incomplete", need more  
attention than an IDE rules  
based refactor for the  
same operation which is  
characterized by accurate  
but potentially incomplete



APIs need to change in  
order to be more amenable  
to generation (single file  
use, library contains  
examples that can prime or  
few-shot the copilot)