

Problems Faced in Programming for Research

Causes, Effects and Some Strategies

Cristian Dinu

March 2023

Introduction

- ▶ Scientific programming is still programming
- ▶ The main issues of general programming still apply
 - ▶ Unclear requirements
 - ▶ Tough stakeholder management
 - ▶ Bad planning
- ▶ But I will talk about problems specific to programming for research

1. Both research and coding are hard

So that one either can do well one or the other

- ▶ Effects: It is challenging to collect the requirement. Misunderstandings and friction happen between the researcher and the coder.
- ▶ Strategies: Iterate fast, show and validate early, using friendly formats (Jupyter, Excel), avoid computer/programming slang

2. The "Paper" is more important than the code

The incentives and priorities seem often wrong. Recognition, too.

- ▶ Effects: Quick and dirty duct-tape coding, neglected documentation, testing, maintainability, and reproducibility
- ▶ Strategies: Recognise software contributions, emphasize code quality, make code as important as the paper

3. People who code seem "in transit" to better roles or self-taught...

I saw most of the code is done "for free" by masters or PhD students or enthusiastic self-taught young academics.

- ▶ Effects: No good practices, inconsistent code quality, slow skills improvement
- ▶ Strategies: Educate beyond coding – e.g. source control, documentation, and establish some standards (e.g. everything goes on departments' GitHub account, no Excel allowed)

4. ...because good developers are expensive and industry pays well

Also, it's unfair to pay a developer three times as much as a researcher, just because the IT pays well.

- ▶ Effects: Difficulty attracting and retaining skilled engineers, reliance on inexperienced developers
- ▶ Strategies: Allow people to be paid directly from grants, or projects with the industry.

More and more blackboxes (libraries, AI models)

- ▶ Effects: Reduced transparency, harder debugging, validation issues, over-reliance on external libraries
- ▶ Strategies: Encourage open-source development, require explainable AI, prioritize validation and testing

Conclusions

- ▶ Programming for research shares difficult problems with "general" programming
- ▶ Competition with the industry is unfair [compared with academia]
- ▶ Hopefully a "software is an investment not an expense" mindset will be adopted by the majority of people involved in research