

# How Do Scientists Develop Scientific Software?

An External Replication



**Gustavo Pinto**  
Federal University of  
Pará, Brazil

**ghlp@cin.ufpe.br**

**@gustavopinto**



**Igor Wiese**  
Technological University  
of Paraná, Brazil

**igor@utfpr.edu.br**

**@igorwiese**



**Luiz Felipe Dias**  
University of São Paulo,  
Brazil

**fronchetti@usp.br**

**@fronfel**

## How Do Scientists Develop and Use Scientific Software?

Jo Erskine Hannay  
Dept. of Software Engineering  
Simula Research Laboratory  
Dept. of Informatics, Univ. of Oslo  
johannay@simula.no

Carolyn MacLeod  
Dept. of Computer Science  
University of Toronto  
cmacleod@cs.utoronto.ca

Janice Singer  
Software Engineering Group  
National Research Council of Canada  
janice.singer@nrc-cnrc.gc.ca

Hans Petter Langtangen  
Center for Biomedical Computing  
Simula Research Laboratory  
Dept. of Informatics, Univ. of Oslo  
hpl@simula.no

Dietmar Pfahl  
Dept. of Software Engineering  
Simula Research Laboratory  
Dept. of Informatics, Univ. of Oslo  
dietmarp@simula.no

Greg Wilson  
Dept. of Computer Science  
University of Toronto  
gwwilson@cs.utoronto.ca

### Abstract

*New knowledge in science and engineering relies increasingly on results produced by scientific software. Therefore, knowing how scientists develop and use software in their research is critical to assessing the necessity for improving current development practices and to making decisions about the future allocation of resources. To that end, this*

### 1. Motivation

There is probably not a single scientist who has not, at some point in time, used a software system to analyze, visualize, or simulate processes or data. Many scientists use such software daily, while others develop it for their own use or for a wider community.

As many researchers have pointed out [2, 3, 6, 8], there is a gap between the general computing community and the scientific computing community. As a result, there is little exchange of ideas relevant to scientific application software.

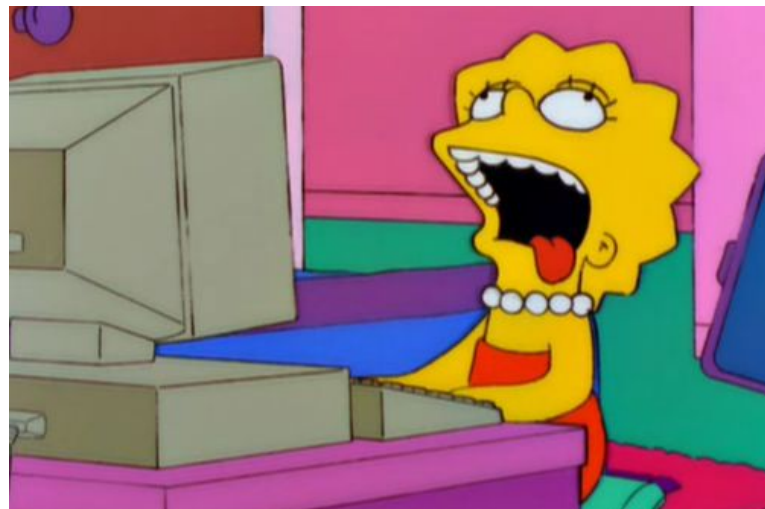
One reason for this is that in scientific computing, a developer must have intimate knowledge of the application

**Hannay, Jo Erskine, et al. (ICSE Workshop, 2009)**

*required to develop and use scientific software is primarily acquired from peers and through self-study, rather than from formal education and training: (2) the number of sci-*

# Context

- With the growth of the technology, scientists are currently using more and more software tools to conduct their research.
- Sometimes, to solve specific problems, scientists need to create their own tooling.
- It is not uncommon that barriers appear during this process, which take time and energy.



# Motivation

- Based on this problem, we tried to get scientists' perceptions on how scientific software is built.
  - This information may help scientists to improve the way that scientific software is developed, overcoming potential barriers.

**The goal of a scientist is to do science, not to develop software**



# How?

- Through an online survey, we asked scientists about general information and their perceptions on how scientific software is developed.
  - Scientists that developed and published at least one software package at CRAN, from a diverse set of research fields, were invited.



**PHYSICS**



**MATH**



**STATISTICS**



**BIOLOGY**

# Questions

28 questions were created in the formulary based on 8 research questions. Some of the questions covered in the survey included:

- How important is developing software to scientists?
- Do scientists spend more time developing software than in the past?
- How familiar are scientists with standard concepts of software engineering?
- What are the sizes of the user communities of scientific software?

# Responses

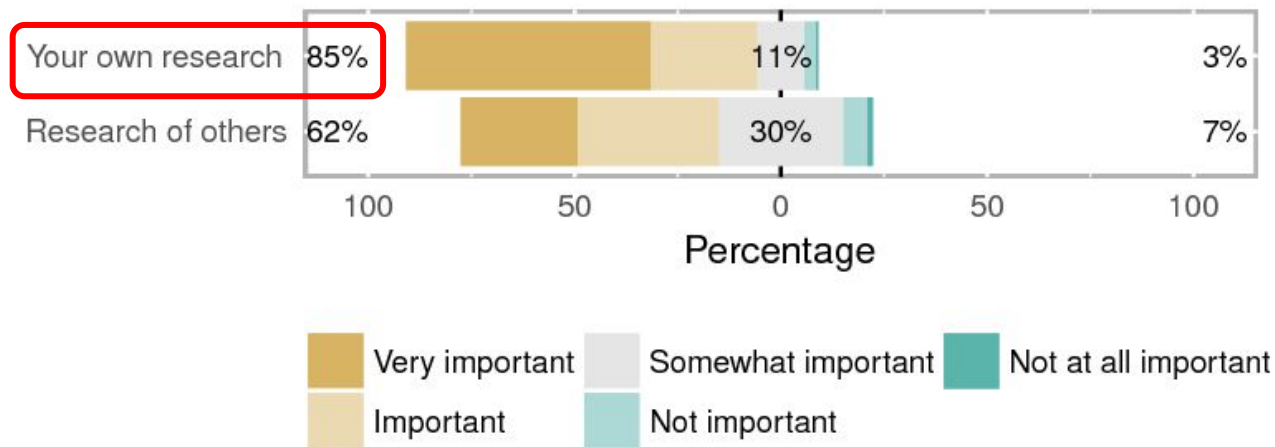
The formulary received 1553 answers, where most of the the respondents were:

- Male (80%)
- Between 30-40 years old (45%)
- Located in Europe (49%)
- Academic researchers (64%)
- Working towards PhD (80%)

The responses of this paper were compared with the original study did by Hannay et. al.

# Importance of Scientific Software Development

- 85% of the respondents believe that that scientific software is important or very important to their own research.

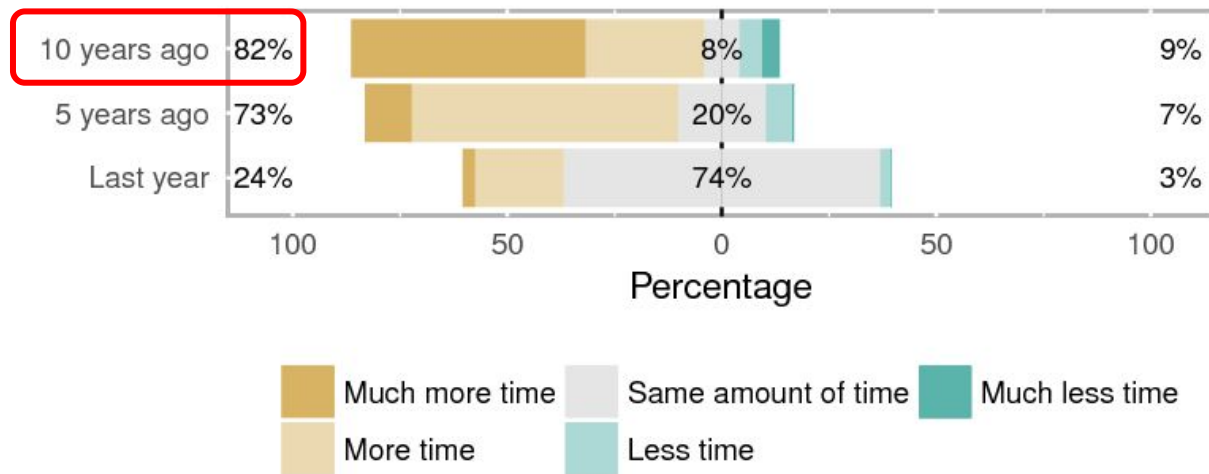


- In the original study, 84.3% of the responses state that developing scientific software is important or very important.



# Dedicated Time to Develop Software (Comparison)

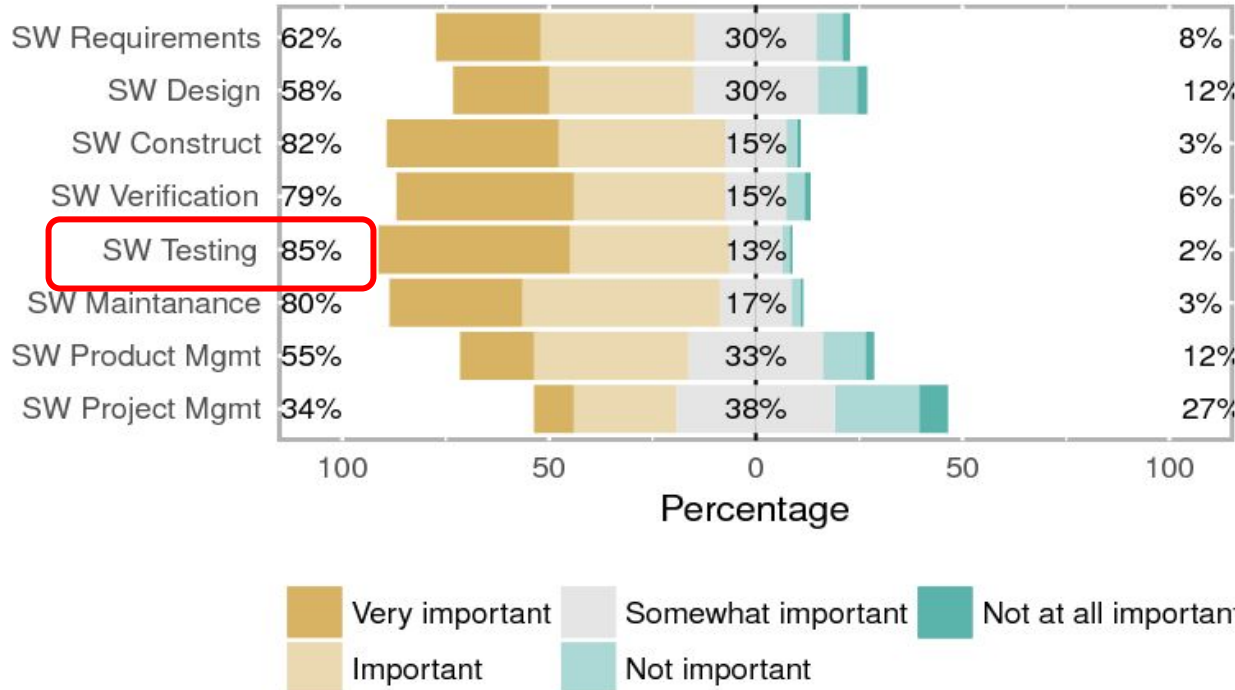
- 82% of the respondents believe that the amount of time devoted to software development is increasing over the decade



- In the original study, just 53% of the respondents state that they spend more or much more time than 10 years ago (1998).

# Importance of Software Engineering Concepts

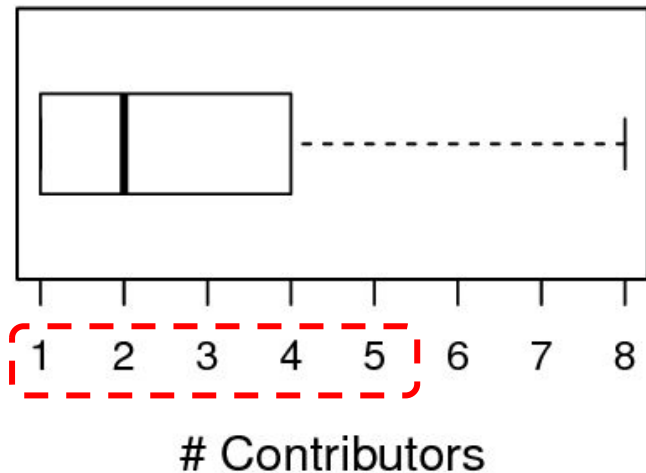
- Software testing was classified by 85% of the respondents as the most important software engineering concept in our research.



- A higher level of importance was also given for software testing in the original study.

# Team Sizes

- Scientists work mostly alone or in small teams when developing scientific software, within a team of two to five.



- The team sizes mentioned in the original study are similar, but not well defined by the author.

# Barriers on Developing Scientific Software (Open)

Among the difficulties mentioned and analyzed are:

- Cross-platform compatibility
- Poor software documentation
- Lack of user feedback
- Lack of formal reward system
- Aloneness



# Conclusion

12

## Context

2

- With the growth of the technology, scientists are currently using more and more software tools to conduct their research.
- Sometimes, to solve specific problems, scientists need to create their own tooling.
- It is not uncommon that barriers appear during this process.



## Motivation

3

- Based on this problem, we tried to get scientists perceptions on how scientific software is built.
  - This information may help scientists to improve the way that scientific software is developed, overcoming potential barriers.

The goal of a scientist is to do science, not to develop software



## How?

4

- Through an online survey, we asked scientists about general information and their perceptions on how scientific software is developed.
  - Scientists that developed and published at least one software package at CRAN, from a diverse set of research fields, were invited.



## Questions

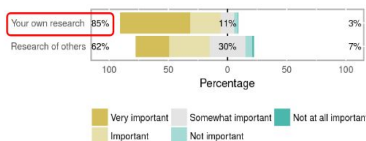
4

28 questions were created in the formulary based on 8 research questions. Some of the questions covered in the survey included:

- How important is developing software to scientists?
- Do scientists spend more time developing software than in the past?
- How familiar are scientists with standard concepts of software engineering?
- What are the sizes of the user communities of scientific software?

## Importance of Scientific Software Development

- 85% of the respondents believe that that scientific software is important or very important to their own research.



- In the original study, 84.3% of the responses state that developing scientific software is important or very important.

## Barriers on Developing Scientific Software (Open Question)

Among the difficulties mentioned are:

- Cross-platform compatibility
- Poor software documentation
- Lack of user feedback
- Lack of formal reward system
- Aloneness

