

Reuse of Mathematical Models in Software Engineering

Alain Le Bussy*

Department of Social Sciences, University of Ghent, Belgium

bussyle281@gmail.com

Received: June 01, 2022, Manuscript No. to social-22-69876; **Editor assigned:** June 03, 2022, PreQC No. to social-22-69876 (PQ); **Reviewed:** June 17, 2022, QC No to social-22-69876; **Revised:** June 22, 2022, Manuscript No. to social-22-69876 (R); **Published:** June 29, 2022

Introduction

Good documentation is required for reuse of research software, but documentation is often criticized. Lack of documentation is due to lack of training, lack of time, or lack of rewards, especially in areas that are not IT-specific. This article addresses the hypothesis that while scientist's document; they do not know exactly what, why, and for whom. To evaluate actual documentation practices for research software, explore existing recommendations, evaluate implementations in everyday practices using specific engineering examples, and compare the results with best practice examples. To get an overview of the required research software documentation, we defined categories and used them to conduct research.

Description

Reuse of mathematical models is becoming increasingly important in systems biology as research moves to large multiscale models consisting of dissimilar subcomponents. Currently, many models are not easily reusable due to inflexible or confusing code, improper language, or poor documentation. Best practice suggestions rarely cover such low-level design issues. This gap can be filled by software engineering that addresses the same issues in software reuse.

In silico experiments in systems biology are expected to be less prone to reproducibility issues than wet lab experiments because they have no natural biological changes and have complete control over the environment. However, recent studies have shown that only half of the published mathematical models of biological systems can be reproduced without much effort. In this article, we will look at possible causes of failure or annoying replication in a case study of a one-dimensional mathematical model of the atrioventricular node, which took four months to reproduce. The model is otherwise due to lack of information, error in equations and parameters, lack of available data files, infeasible code, lack or incompleteness of experimental protocols, lack of rationale for equations. It shows that even rigorous studies can be difficult to reproduce. Many of these issues are similar to those solved by software development using techniques such as unit testing, regression testing, continuous integration, version control, archiving services, and complete modular design with extensive documentation. Due to certain benefits, the amount of software development outsourcing (SDO) is growing rapidly. Due to the challenges arising from the requirements engineering (RE) process, the expected benefits of SDO have not been achieved in some projects. The purpose of this research effort is to recommend RE practices to address common RE process issues in the case of SDO. For this reason, a thorough literature review was conducted and two surveys were conducted with experienced practitioners in the SDO industry. The survey was conducted in a semi-supervised style, using convenience sampling. A 50 percent rule and a 4-point Likert scale were also used to determine the benefits of RE practices for handling problems. Conceptually, genomic annotation is to derive biological information and assign it to a gene product.

Conclusion

Over the years, numerous pipelines and computational tools have been developed to automate this task and help researchers gain knowledge of the genes under study. However, despite these technological advances, information related to the gene product needs to be validated and enhanced with manual annotations or manual curation. Although referred to as the gold standard process for registering data in biological databases, the task of manual maintenance requires a great deal of time for researchers to screen numerous products from different public databases. It takes effort. To solve this problem, we introduce CODON, a manual genomic data maintenance tool that can perform the prediction and annotation process. The software uses a finite state machine in the prediction process to automatically annotate products based on information from the Uniprot database.

