Prof. dr hab. inż. Jerzy Nawrocki
Instytut Informatyki
Politechnika Poznańska
jerzy.nawrocki@put.poznan.pl

Poznań, 2021-08-23

Review of Ph.D. dissertation by

Tomasz Przedziński

entitled:

Methodology for Development of Scientific Software and Test Frameworks in Function of Precision of the Expected Results

1. Introduction

In contemporary Physics, and especially High Energy Physics (HEP), software tools play a very important role. It is confirmed, amongst others, by popularity of the *Journal of Computational Physics* with Impact Factor over 3.5 in 2020[1] and the *Computer Physics Communications* journal with Impact Factor near to 4.4 in 2020[2]. Thus, it is worth to discuss how HEP software tools are (now) and should be (in the future) developed including their testing. This is, roughly speaking, what the dissertation is about.

The dissertation is focused on the following topics:

- Methodology of scientific software development
- Testing of scientific software
- Presentation of tools co-developed by Tomasz Przedziński

In the subsequent sections the reader will find my remarks concerning each of the topics. They have been split into strengths and weaknesses and each of them is followed by a detailed justification.

2. Methodology of scientific software development

From the dissertation's title it follows that the methodological aspects are the central part of the work.

Strength S-1. The dissertation contains practical examples of problems that one can be faced with when developing and maintaining software, especially scientific one.

Justification. I especially like the examples and discussion contained in Sec. 2.7.1 and 2.7.2. They are very clear and can be used for educational purposes. Moreover, I like some fragments of Sec. 3.1.1 and 3.1.2 showing the importance of domain experts and end users in the software development process. Those roles are too frequently underestimated by programmers.

[]

[1] https://journalinsights.elsevier.com/journals/0021-9991/impact_factor
[2] https://journalinsights.elsevier.com/journals/0010-4655/impact_factor

Weakness W-1. Goal 2 of the dissertation (page 9) was to *"analyse the scientific software development process"*. Unfortunately, the analysis presented in Sec. 3.1 is vague and raises doubts.

Justification. In Sec. 3.1.1 multi-layered structure is presented which has *"layers built from different **fields of expertise**"* (p. 36). This structure is illustrated in Fig. 8 (p. 36). My doubts are the following:

- The proposed split into "fields of expertise" is **not validated** – there is just one example and nothing more. Such validation could be based, for instance, on the papers concerning TAUOLA and other tools (there are quite many of them).
- The split into the *Core issue* expertise and the *Mathematical Formalism* expertise **seems artificial**. Looking at the papers authored or co-authored by prof. Zbigniew Wąs I am sure he is fully competent in both "fields of expertise". Maybe the problem is lack of definition of *"field of expertise"*?
- The example of pages 36/37 **looks naïve**. It is hard to believe that a person competent in *"a **complex theory** that describes existence of a resonance <<Y>>"* and aware that one *"can use Gaussian distribution to estimate probability of generating the resonance of given energy"* is no capable *"to describe these resonances using Gaussian distribution"*.
- It is **not explained** why the proposed fields of expertise are organized into **layers.** It is a strongly constrained structure. In Software Engineering each layer of software communicates only with its direct neighbours and is not aware of existence of other layers. In the context of Fig. 8 does it mean the *Test and Validation* layer is completely independent of *Core Issue* and *Mathematical Formalism*? I doubt it.
- The second sentence of the section reads: *"Each of the layers* [..] *needs throughout validation **before** the next layer can be built on top of the previous one"*. It suggests a kind of waterfall lifecycle, not *"fields of expertise"*.
- How about Statistics, Data Science, Numerical Analysis, and Project Management? Are they regarded irrelevant?

In Sec. 3.1.2 human resources management is discussed. Table 3 presents *"list of different types of project contributors"*. Again I have some doubts:

- There is no explanation of the relation between *types of contributors* of Table 3 and the *fields of expertise* presented in Fig. 8.
- The split between "Members" and "Roles" is not clear to me. Is it really needed? I think one category (e.g. *Stakeholders* known from PMBOK) would be enough.
- The difference between *External collaborator* and *External expert* is very vague.
- I have an impression that the fields of expertise (Fig. 8) and the list of contributors (Table 3) could be simplified (perhaps merged) and presented in relation to the roles known from other methodologies of software project management, e.g. eXtreme Programming, Scrum, PRINCE2 etc.

Moreover, when talking about the negative impact of new team members on performance of the team because the development team must *"divide their attention between work and training of the new team members"* (p. 37/38) it would be appropriate to make a reference to the Brooks law [Brooks1975, Endres2003] (it says that *"adding manpower to a late project makes it later"* just because of the need of training new team members). In other words, **this issue is not specific to development of *scientific* software**.

[]

Weakness W-2. Goal 3 of the dissertation was to *"describe the methodology that emerged during the development of several tools"* (page 9). Unfortunately, what has been presented can hardly be called "methodology". It is just a few practices that have been described very briefly and many other (probably very useful) practices have been omitted without explanation.

Justification. On page 9, while commenting Goal 3, the author wrote:

This thesis attempts to present an approach designed to prevent most common problems encountered during development process of a scientific tool ... (Section 3.2.2)

Section 3.2.2 is entitled *"Scientific software and R&D environment"* and it is showing that R&D projects are very different from scientific ones (the main difference seems to be mechanical- and hardware-related issues). So, perhaps the author meant Sec. 3.2.3 *"Managing scientific software"*. I am surprised that the "methodology" presented in this 1-page-long section (page 55) is reduced **just** to **one recommendation**:

"tasks related to scientific software [..] *can be managed using Kanban board"*.

In Sec. 4.4 *Creating the software framework* (pp. 61-62) two other software development practices are mentioned:

- Refactoring. The author recommends that it is should be *"always limited to as little as possible to avoid damaging the physics context of the project"*. Unfortunately, it is not explained **how code refactoring could damage the physics context**. No other arguments for limiting the scale of refactorization are presented.
- Iterative approach. This practice is illustrated with an example (pp. 61-62).

I wonder **why those two practices have been not included into Sec. 3.2.3**. Moreover, **I doubt if the practices proposed in the dissertation are complete enough to manage the challenges mentioned in Sec. 3.1**, including:

- human resources management (perhaps pair programming and end-user programming techniques could be useful but they are neglected),
- lack of beta-testers and scarce user feedback (perhaps some reviews would be necessary but they are not mentioned),
- inter-project dependencies (how about *variability in product lines* known for about 20 years – see [Gurp2001]).

I am surprised that **many potentially useful practices have been just ignored**, including reflection workshops, code reviews, risk management, daily Scrum (a.k.a. stand-up meetings), pair programming or side-by-side programming etc.

[]

Weakness W-3. The dissertation lacks *Systematic Literature Review* and, as a consequence, some very important methodologies concerning software development have been omitted.

Justification. Methodologies of software project management are presented in the following sections:

2.9.3 Methodologies (pp. 27-34)
3.2.1 Common methodologies in relation to scientific software (pp. 50-53)
7.2 Methodologies used in business environments (pp. 119-122)

Unfortunately, none of those sections mentions **DevOps**, a very popular approach to software development. An excellent review of literature concerning this approach has been done by Leonardo Leite et al. [Leite2020]. This review covers 50 papers and the authors found over 100 books published in 2018 alone and available via Amazon, so this approach is by no means negligible. As in HEP projects researchers not only develop software but also operate it, many practices associated with DevOps might prove useful when working on HEP tools. Moreover, the mentioned paper is also a very good example of conducting systematic literature review.

Other methodologies worth mentioning but omitted include at least **Lean Software Development** [Poppendieck2003] (the book has been cited almost 2 000 times!). Moreover, some interesting papers on using **Scrum for research** project management have been also neglected, e.g. [Hicks2010, Hidalgo2019, LaBrec2016].

[]

3. Testing of scientific software

Weakness W-4. Goal 4 of the dissertation was to *"address scientific software testing process"*. Unfortunately, description of this process lacks scientific rigor.

Justification. Discussion on testing scientific software is presented in Chapter 5. The first section of the chapter (pp. 66-75) is devoted to taxonomy of software tests and it raises a number of doubts:

- The order in which types of tests are presented suggests that the scope of functionality tests is smaller than the scope of integration tests (p. 67). I think it is just the opposite[3].
- I also do not agree that smoke tests are of the smallest scope (it is also suggested on p. 67). According to Sten Pittet[4] *"smoke tests are basic tests that check basic **functionality** of the application"*. So, their scope is definitely larger than that of unit tests.
- According to Fig. 16 (p. 72) security tests can be performed only on the whole system, not on a single application. I think it is not true: for instance the proneness to the SQL injection attacks can be tested on the level of single applications.
- Figure 17 of p. 73 suggests that acceptance tests can cover only functional requirements. It is not true: acceptance tests can also check non-functional attributes such as performance (e.g. response time).

The second section presents tests used in scientific software (pp. 75-84). Here are my remarks:

- The term *"numerical technical test"* is not clearly defined. I guess the author meant tests concerning the level of errors associated with machine representation of real numbers.
- On p. 76 it is claimed that numerical technical tests are executed only on the application level (this opinion is also "repeated" in Fig. 20, p. 82). I do not agree. Even a small function computing square root could be subject to a "numerical technical test".
- I understand that so called "data flow tests" discussed on p. 76 are in fact data integrity test concerning Event Records. Then perhaps the main problem is something what in the database community is called user-defined integrity. As Event Record (ER) structure has been standardized, perhaps some rules concerning ER-related integrity might be presented but they are not.
- I have an impression that from the testing point of view the difference between data flow tests and output metadata tests is negligible and the corresponding sections could be merged (in both cases what matters is user-defined integrity).
- It is not clear why core functionality test are treated as non-functional ones. If software such as TAUOLA is extended with functionality allowing to analyse e.g. photon emission in decays of resonances – I mean Fortran routing RADCOR mentioned on p. 96 – then testing the correctness of RADCOR will be, from the Software Engineering point of view, functional testing. Only if some additional requirements will be imposed on RADCORE, concerning for instance execution time, one can call such tests non-functional.
- It is not clear why *analysis* of systematic error is classified as a *test* (p. 78).
- Fig. 20 (p. 82) suggests that systematic error analysis (called *"Systematics studies"*) cannot be performed on a module level. I do not agree. For instance, I could have a module (e.g. a class) containing a function/method that computes approximation of definite integral and then analysis of systematic error would be very appropriate.

[]

Weakness W-5. The dissertation lacks *Systematic Literature Review* and, as a consequence, some important approaches to testing have been omitted.

Justification. The list of testing techniques is presented in the following sections of the dissertation:

5.1.3 Testing techniques (pp. 70-71),

5.2 Tests used in scientific software (pp. 75-84),

5.3 Useful testing techniques (pp. 84-91).

---

[3] See also https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

[4] https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

The list of omitted approaches and publications associated with them includes:

Differential testing. This approach is well known for long time. For instance, William McKeeman's paper [McKeeman1998] published over 20 years ago received almost 400 citations according to Google Scholar. There are some tools supporting this approach, e.g. NEZHA [Petsios2017] or GRAFTER [Zhang 2017]. The concept triggered some other interesting ideas (see e.g. [Elbaum2006], [Evans2007]). Differential testing seems especially useful when creating a new version of a given module or tool, e.g. when upgrading to a new compiler version and facing problems such as those presented in Sec. 2.7 *Working with old codebase*, pp. 18-21. This approach was applied when first versions of `PHOTOS` were tested against MUSTRAAL (see [Barberio1994], p. 300).

Property-based testing. It's another pretty old and well-known approach to testing (see e.g. [Fink1997]). This approach became very popular in the Haskell community when QuickCheck was made available [Claessen2000]. For C/C++ the counterpart of QuickCheck is DeepState [Goodman2018]. An interesting idea is to enhance property-based testing with search strategies (e.g. Simulated Annealing – see e.g. [Loescher2017]).

Metamorphic Testing. Probably the idea was published for the first time in 1998 [Chen1998]. Now there are many papers concerning this approach – see e.g. the literature review done by Segura et al. [Segura2016]. Some people consider metamorphic testing as a version of property-based testing [Wiki-Meta].

Current developments in the area of testing numerical software allow not only effectively narrow down the set of test cases [Yi2017, Zou2020] but also to repair the code in an automatic way [Yi2019]. Those developments can significantly save time spent on testing, so it is surprising that the author neglects them.

[]

4. Tools co-developed by Tomasz Przedziński

The dissertation is based on the experience gained by Tomasz Przedziński while he has been working on various software tools aimed to help the HEP community.

Strength S-2. `TAUOLA`, `PHOTOS`, and `TauSpinner` described in the dissertation and co-developed by Tomasz Przedziński seem very important for the High Energy Physics community. I am really impressed.

Justification. The most cited paper co-authored by Tomasz Przedziński and concerning TAUOLA is the one on universal interface of TAUOLA [Davidson2012] where Tomasz Przedziński is the **corresponding author**. According to Google Scholar that paper was **cited over 500 times** (!). What is also important, amongst those citations are papers with over 300 citations each and **the type of those citations indicates strong impact of TAUOLA on their research**. Here are some quotes coming from 3 highly cited papers which illustrate that observation:

- *"All generators* [i.e. PYTHIA, POWHEG, and MADGRAPH – Jerzy Nawrocki] *are interfaced with TAUOLA for the simulation of the τ decays."* ([CMS2014], over 400 citations);
- *"Decays of τ leptons are handled with TAUOLA (v. 2.75)"* ([CMS2012], over 300 citations);
- *"The TAUOLA package is used to simulate the decays of the τ lepton into the $3\pi\ \nu_\tau$ and $3\pi\ \pi^0\ \nu_\tau$ final states"* ([Aaij2018], over 300 citations).

The second of the three mentioned tools is PHOTOS. The paper on C++ interface to PHOTOS [Davidson2016] has almost 400 citations. Here are quotes from three most cited papers that report research aided with the PHOTOS version co-authored by Tomasz Przedziński:

- *"When PYTHIA6 or HERWIG is used, PHOTOS [Golonka2006b, Davidson2016] is employed to describe additional photon radiation from charged leptons."* ([Aad2013], over 1 100 citations);
- *"Finally, comparing tunes based on the native Pythia8 QED final-state corrections with Pythia8 results using Photos for QED final-state radiation, the results are found insensitive to the differences in the QED FSR implementations."* (([Aad2014], over 1 200 citations);

- *"The numerical coefficients in Eq. (III.1.35) have been obtained neglecting QED corrections. The latter must be included at the simulation level by appropriate QED showering programs, such as PHOTOS [Davidson2016]."* ([Florian2016], p. 407, over 1 100 citations)

Those quotes confirm usability of the mentioned tools. As regards TauSpinner, its growing popularity is very well illustrated in the dissertation (Fig. 29, p. 107) – the total number of citations to the papers concerning TauSpinner is near a hundred. So, no doubt TAUOLA, PHOTOS, and TauSpinner are significant achievements.

[]

Weakness W-6. Although Tomasz Przedziński was perhaps very helpful in the development of TAUOLA, PHOTOS, and TauSpinner, his contribution to the success of those tools seems not essential. It seems the pivotal role was played by prof. Zbigniew Wąs.

Justification. The evolution of the TAUOLA tool is illustrated in Table 1. From those data it follows that before Tomasz Przedziński joined the team working on TAUOLA and PHOTOS, a lot of programming work has been already done (predominantly in Fortran77). A similar conclusion can be drawn when analysing the papers concerning PHOTOS. There are some very important papers published before Tomasz Przedziński joined the team (that is without his participation), e.g. [Barberio1991], [Barberio1994], [Golonka2006b] – the last one has over 3 300 citations (!).

On the other hand, I have an impression that Tomasz Przedziński played an important role as a C++ expert. Since the beginning of this century C++ became more and more popular amongst the HEP community (see e.g. [Dobbs2001], [Sjostrand2020]) and it became necessary to port different tools from Fortran77 do C++. Perhaps the authors of the Fortran77 versions of TAUOLA and PHOTOS (i.e. Z. Wąs and others) could manage to learn C++ and do the programming work themselves but that would take time. So, it is quite possible that Tomasz Przedziński helped significantly shorten the time the users (physicists) had to wait for C++ interface to TAUOLA and PHOTOS.

As it comes to TauSpinner, perhaps the first journal paper presenting the tool was [Czyczuła2012] co-authored by Tomasz Przedziński. Now this paper has over 160 citations. As Zofia Czyczuła is familiar with C++[5], it seems part of the programming job was done by her. Recently, E. Richter-Wąs and Z. Wąs proposed an enhancement to TauSpinner [Richter-Wąs2019] and it seems they managed to do the programming work themselves.

[]

Weakness W-7. The tools co-developed by Tomasz Przedziński seem very important to the community of High Energy Physics. However, the contribution of the dissertation to Software Engineering is not impressive.

Justification. Tomasz Przedziński co-authored a few very useful tools but after reading the dissertation and a few papers co-authored by him I have an impression as if he was just a successful C++ programmer, not a researcher. When working on TAUOLA and other tools he had a number of opportunities to try something new and reflect upon it but apparently he did not use those opportunities. For instance, he could try:

- Design patterns (which of them are most useful? or maybe some new design patterns?);
- Micro-service architecture;
- Variability management in software product lines;
- End-user programming (e.g. by using a kind of decision tables or Knuth's Literate Programming);
- Static code analysis;
- Bug localization (e.g. at the level of files) support with Machine Learning;
- More precise evaluation of practices of software project management (clear claims + evidence);

[5] https://www.linkedin.com/in/zofia-czyczula-rudjord/?originalSubdomain=no

Table 1. Presumably most important publications concerning TAUOLA in chronological order. White background colour indicates papers co-authored by T. Przedziński.

Paper	Authors	New features of TAUOLA	Citations
[Jadach1991]	S. Jadach, J. H. Kühn, Z. Wąs	Monte Carlo simulation of decays of polarized τ leptons	**1083**
[Jeżąbek1992]	M. Jeżabek, Z. Wąs, S. Jadach, J.H. Kühn	Complete (α) QED corrections for τ decays + KORALB and KORALZ	221
[Jadach1993]	S. Jadach, Z. Wąs, R. Decker, J.H. Kühn	Over twenty decay channels + interface to PHOTOS, KORALB, and KORALZ + events encoded in the common block proposed by the Particle Data Group	**1611**
[Jadach1994]	S. Jadach, B.F.L. Ward, Z. Wąs	Experiments in e+e– colliders in the Z resonance energy range + initial and final state bremsstrahlung corrections	**1190**
[Wąs2001]	Z. Wąs	Universal interface of TAUOLA for the decay of τ-leptons produced by "any" generator	355
[Bondar2002]	A.E. Bondar, S.I. Eidelman, A.I. Milstein, T. Pierzchała, N.I. Root, Z. Wąs, M. Worek	New parameterization of form factors developed for 4π channels of the τ lepton decay	34
[Wąs2005]	Z. Wąs, P. Golonka	MC-TESTER for semi-automatic testing + universal interface using HEPEVT common block	111
[Golonka2006a]	P. Golonka, B. Kersevan, T. Pierzchała, E. Richter-Wąs, Z. Wąs, M. Worek	Variability management (versions of TAUOLA and PHOTOS created by users); TAUOLA Fortran Interface; C language	294
[Davidson2012]	N. Davidson, G. Nanava, **T. Przedziński**, E. Richter-Wąs, Z. Wąs	Taking into account effects of spin at the time when a τ decay is generated; TAUOLA C++ interface based on HepMC [Dobbs2001] (richer than the Fortran one)	521
[Wąs2015]	Z. Wąs	Multitude new channels for anomalous τ decay modes and parametrization based on defaults used by BaBar collaboration are introduced	3
[Chrzaszcz2018]	M. Chrzaszcz, **T. Przedzinski**, Z. Was, J. Zaremba	Implementation of Lepton Flavour Violating τ decays; up to 500 channels ("*Rather **minor changes** with respect to versions of Refs. [Wąs2001, Wąs2005] were necessary*"; "*The distribution corresponds nearly in all details to the version presented at TAU14 conference [Wąs2015]; **minor modifications** are explicitly listed*")	18

The impression that accomplishments of Tomasz Przedziński contribute much more to High Energy Physics than to Software Engineering is supported also by another observation. Out of 20 papers listed in Sec. 8.3 (pp. 128-130) 18 of them have been published in Physics-oriented journals (mainly *Computer Physics Communications*, and *European Physical Journal*). The other two are:

- T. Przedzinski, M. Malawski and Z. Wąs, "Software Development Strategies for High-Energy Physics Simulations Based on Quantum Field Theory", Computing in Science & Engineering, vol. 22, no. 4, pp. 86-98, 1 July-Aug. 2020, doi: 10.1109/MCSE.2019.2947017

  Keywords: High Energy Physics, **Software Testing**, Simulation Software, Monte Carlo

- T. Przedzinski, P. Roig, O. Shekhovtsova, Z. Wąs, and J. Zaremba, "Confronting theoretical predictions with experimental data; fitting strategy for multi-dimensional distributions," Comput. Sci., vol. 16, no. 1, 2015, 17-38], https://doi.org/10.7494/csci.2015.16.1.17

  Keywords: RChL, TAUOLA, hadronic currents, fitting strategies, multi-dimensional fit

The keywords associated with those two papers suggest that only the first one contains something interesting from the Software Engineering point of view (it is *Software Testing*).

Perhaps the most interesting part of the dissertation, from the Software Engineering point of view, is Sec. 6.7 *Quantitative evaluation of software complexity and precision* (pp. 112-114)[6]. The idea of using so called *Precision Tag* as a complexity indicator for software supporting HEP experiments is pretty clever. However, there is some **inconsistency** between Table 2 of page 7 and Table 7 of page 113. According to Table 2 the total number of code lines of TAUOLA and PHOTOS implemented in Fortran is above 30 00 but in Table 7 in the row *Tauola-Photos-F* (which apparently also corresponds to TAUOLA and PHOTOS implemented in Fortran) the code size is less than 20 000 (!). This inconsistency is not explained and leads to confusion.

[]

Below there is one more argument showing that perhaps the design of some parts of the tools co-developed by Tomasz Przedziński could be improved.

Weakness W-8. The interface of random number generators presented in the dissertation seems not universal enough.

Justification. Random number generators are used by both TAULOLA++ (Appendix A 2.1 Tauola++, p. 157) and PHOTOS++ (Appendix A 2.2 Photos++, p. 171). As there are many algorithms of random number generation, the author has decided to allow the user to choose the right one. Therefore, the interface to Tauola++ and Photos++ contains the following C++ function which allows to specify the subroutine `gen` that will be used to generate random numbers:

```
setRandomGenerator(double (*gen)())
```

This signature is not universal enough as it does not allow to use a modification of RanMar algorithm [Marsaglia1990] proposed by F. James [James1990]. The problem is that James' version of RanMar produces not a single number, as it is assumed by `setRandomGenerator`, but a vector of numbers (thanks to this modification James gained a speedup of factor 2 or even 3 – see [James1990]). Similar problem would be if someone tried to apply the Box-Muller [Box1958] transform mentioned on page 36 – this algorithm returns two numbers, not one, so the above interface would not allow to use it.

Another problem is with the Metropolis–Hastings algorithm [Hastings1990] which seems very popular (Hastings' paper has over 17 000 citations!). This algorithm needs a function f(x) that is proportional to the desired probability distribution P(x). It's not clear to me how such a function could be specified with the current interface (similar function is needed also by the rejection sampling method).

Yet another issue concerns computation of definite integrals. As mentioned by Frederick James [James1990] and illustrated by George Levy [Levy2002], using a uniform random number generator instead of a quasi-random one to approximate a definite integral can be a **big mistake** as the approximation error of the former can be 10 times bigger than for the latter. Moreover, the approximation error depends also on a type of quasi-random numbers (Levy studied three types of quasi-random numbers proposed by Ilya Sobol, Henri Faure, and Harald Niederreiter – their approximation error can differ pretty significantly). This issue should be taken into account when proposing a universal interface for random number generators. Moreover, some **guidelines on selecting random number generators would be very appropriate** especially when someone intends to propose a methodology of software development for High Energy Physics applications.

[]

[6] I also like the examples of Sec. 2.7.1 and 2.7.2 but more for the didactic and practical reasons.

5. Other remarks

Weakness W-9. There is lack of cohesion between the thesis presented on page 9 and the dissertation title. Moreover, the evidence that is to support the thesis is poor.

Justification. The title of the dissertation reads:

> *"**Methodology** for Development of Scientific Software and Test Frameworks in Function of Precision and the Expected Results"*

The **thesis** presented on page 9 consists of three sentences:

> "(1) Often scientific software development focuses on increasing the precision of the results.
>
> (2) This process follows the cycle of improving the physics model, describing the model using mathematical formalism, implementing the model with numerical approximations, creating the software framework, documenting and validating results, testing and publishing results.
>
> (3) Then, *the **effort** put into testing and into building proper validation framework **rises proportionally** with the effort required to achieve results with increased precision*."

Because of the word "Then" starting sentence (3), the first two sentences are just assumptions which constrain applicability of the hypothesis (thesis) expressed by sentence (3). Surprisingly, **the thesis contained in sentence (3) has nothing to do with methodology of software development** (!). Moreover, the detailed goals that are presented on pages 9-10 also does not contribute to the thesis contained in sentence (3). They are quoted below:

1. *Present the physics background required to understand the common concepts of software development for physics experiment*
2. *Analyse the scientific software development process*
3. *Describe the methodology that emerged during the development of several tools*
4. *Address scientific software testing process*
5. *Present history and decisions made during development of several tools*
6. *Summarize our experience related to scientific software development process*

None of the above goals concerns **effort**.

Moreover, in the dissertation there is lack of definition of "*the effort required to achieve results with increased precision*" and lack of metrics or indicators of it. There are some interesting data presented in the tables of page 7 and in Sec. 6.7 (pp. 112-114) but **none of them directly concerns effort**. Apparently, the author has assumed that the effort is proportional to the code size. Unfortunately, this assumption is very risky: it does not take into account the amount of deleted or changed lines of code (and those activities also consume time). The worst thing is that this assumption is not clearly stated and lacks any justification. Hence, **the presented evidence is poor**.

[]

I have also a few astonishments:

A-1. When discussing analysis of systematic error in Sec. 5.2.5 (p. 78) the author does not mention any previously published paper on the topic. Even the work by Z. Wąs and S. Jadach on systematic error concerning TAUOLA [Wąs1992] is omitted.

A-2. The author has omitted an important paper on standard format for event files [Alwall2007] (co-authored by Z. Wąs). It proposed an XML-style structure for Fortran common blocks which could be used for data exchange between various HEP software tools. The paper has almost 700 citations. Perhaps nowadays a JSON version would be more appropriate.

A-3. The dissertation contains a sidebar devoted to Adam Kolawa but there is no similar mentioning of Stanisław Ulam, Polish mathematician who invented the Monte Carlo method [Metropolis1949] which is much more connected with the dissertation.

A-4. I do not know why Chapter 7 has been included into the dissertation. It seems superfluous.

A-5. Although from the linguistic point of view the dissertation is almost perfect and it contains many good examples, its composition seems chaotic. Some information is repeated, some other is placed in surprising chapters (e.g. Sec. 2.7 *"Working with old codebase"* is placed in Chapter 2 *"Physics background"*).

6. Summary

I am really perplexed. On the one hand, the author:

- co-developed a number of very useful tools (see **Strength S-2**) and
- co-authored many well-cited papers that impressed me (one of them has over 500 citations!)

but on the other hand

- the dissertation lacks scientific rigor and raises many doubts (see **Weaknesses W-1**, ..., **W-9**).

Weaknesses **W-1**, **W-2**, **W-4**, and **W-9** show that the problem stated in Sec. 1.4 has not been properly solved (in particular the thesis/claim presented on page 9 lacks proper evidence).

So, let us assume that the problem stated in Sec. 1.4 is ill-formulated (one can treat this as an editorial weakness) and let us focus on the strongest part of the dissertation which seems to be presentation of TAUOLA and other tools (Chapter 6). Then there are two other questions:

- What was the role of Tomasz Przedziński in the development of those tools? (see **Weakness W-6**)
- Is it appropriate to confer a Ph.D. degree in Informatics when the main contribution of the dissertation is to Physics? (as it comes to TAUOLA and other tools, the problem was lack of appropriate tools supporting some HEP analysis and it has been solved by implementing those tools for Physicists; see also **Weakness W-7**).

Let me make it clear: the above two questions are raised only because of the important weaknesses of the dissertation (**W-1**, **W-2**, **W-4**, **W-9**) associated with the problem stated in Sec. 1.4.

My general impression is that although Tomasz Przedziński has co-developed a few very useful software tools and co-authored a few really important papers, **the current version of the dissertation does not show that he can conduct independent scientific work in the field of Informatics**, in the area of Software Engineering. However, **I do realize that this opinion is subjective** and **it is quite possible that Tomasz Przedziński will be able to explain at least some of the weaknesses** (e.g. W-6 and W-7). Therefore, taking into account undoubtful strengths of the dissertation, **I recommend to admit the thesis to public defence**.

References

[Aad2013] G. Aad et al. (ATLAS Collaboration), "Measurements of Higgs boson production and couplings in diboson final states with the ATLAS detector at the LHC", Physics Letters B, Vol. 726, Issues 1–3, 2013, 88-119, https://doi.org/10.1016/j.physletb.2013.08.010

[Aad2014] Aad, G., et al. (ATLAS Collaboration), "Measurement of the Z/γ * boson transverse momentum distribution in pp collisions at s√=7 TeV with the ATLAS detector", J. High Energ. Phys. 2014, 145 (2014). https://doi.org/10.1007/JHEP09(2014)145

[Aaij2018] R. Aaij et al. (LHCb Collaboration), "Test of lepton flavor universality by the measurement of the B0→$D^{*-}\tau^+\nu_\tau$ branching fraction using three-prong τ decays", Phys. Rev. D 97 (2018), https://journals.aps.org/prd/abstract/10.1103/PhysRevD.97.072013 A

[Alwall2007] J. Alwall et al., "A standard format for Les Houches Event Files", Computer Physics Communications, Vol. 176, Issue 4, 2007, 300-304, https://arxiv.org/pdf/hep-ph/0609017.pdf

[Barberio1991] Elisabetta Barberio, Bob van Eijk, Zbigniew Wąs, "Photos — a universal Monte Carlo for QED radiative corrections in decays", Computer Physics Communications, Vol. 66, Issue 1, 1991, 115-128, https://doi.org/10.1016/0010-4655(91)90012-A

[Barberio1994] Elisabetta Barberio, Zbigniew Wąs, "PHOTOS - a universal Monte Carlo for QED radiative corrections: version 2.0", Computer Physics Communications, Vol. 79, Issue 2, 1994, 291-308, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.5978&rep=rep1&type=pdf

[Bondar2002] A.E. Bondar, S.I. Eidelman, A.I. Milstein, T. Pierzchała, N.I. Root, Z. Wąs, M. Worek, "Novosibirsk hadronic currents for τ→4π channels of τ decay library TAUOLA", Computer Physics Communications, Vol. 146, Issue 2, 2002, 139-153, https://doi.org/10.1016/S0010-4655(02)00262-X

[Box1958] G. E. P. Box, and Mervin E. Muller. "A Note on the Generation of Random Normal Deviates", Ann. Math. Statist. 29 (2) 610 - 611, June, 1958. https://doi.org/10.1214/aoms/1177706645

[Brooks1975] Frederick P. Brooks, "The Mythical Man-Month – Essays on Software Engineering", Reading MA, Addison Wesley, 1975

[Chen1998] T. Y. Chen, S. C. Cheung, and S. M. Yiu, "Metamorphic testing: A new approach for generating next test cases", Dept. Comput. Sci., The Hong Kong Univ. Sci. Technol., Hong Kong, Tech. Rep. HKUST-CS98-01, 1998, https://www.cse.ust.hk/~scc/publ/CS98-01-metamorphictesting.pdf

[Chiang2014] Wei-Fan Chiang Ganesh Gopalakrishnan Zvonimir Rakamaric Alexey Solovyev, "Efficient Search for Inputs Causing High Floating-point Errors", 19th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming PPoPP '14, 43-52, https://dl.acm.org/doi/abs/10.1145/2555243.2555265?casa_token=RDggfKD7IfgAAAAA:5OhOuU3Z1vs8Uh_NB2Lp-ZokJ_rwlyqgAQQrx7ppZ_QzQ2vpd2A8nJX5zsrSOzAojXLhtohgRohPQ

[Chrzaszcz2018] M. Chrzaszcz, T. Przedzinski, Z. Was, J. Zaremba, "TAUOLA of τ lepton decays—framework for hadronic currents, matrix elements and anomalous decays", Computer Physics Communications, Vol. 232, 2018, 220-236, https://doi.org/10.1016/j.cpc.2018.05.017

[Claessen2000] Koen Claessen and John Hughes, "QuickCheck: a lightweight tool for random testing of Haskell programs", Proceedings of the fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00). ACM, New York, NY, USA, 2000, 268–279. DOI: https://doi.org/10.1145/351240.351266

[CMS2012] The CMS collaboration, S. Chatrchyan, J. Swanson et al., "Measurement of the tt production cross section in the dilepton channel in pp collisions at √s =7 TeV", J. High Energ. Phys. 2012, 67 (2012). https://doi.org/10.1007/JHEP11(2012)067

[CMS2014] The CMS collaboration., Khachatryan, V., Sirunyan, A.M. et al. "Search for neutral MSSM Higgs bosons decaying to a pair of tau leptons in pp collisions", J. High Energ. Phys. 2014, 160 (2014). https://doi.org/10.1007/JHEP10(2014)160

[Czyczuła2012] Czyczula, Z., Przedzinski, T. and Was, Z. "TauSpinner program for studies on spin effect in tau production at the LHC", Eur. Phys. J. C 72, Article number 1988 (2012), https://arxiv.org/pdf/1201.0117.pdf

[Davidson2012] N. Davidson, G. Nanava, T. Przedziński, E. Richter-Wąs, Z. Wąs, "Universal interface of TAUOLA: Technical and physics documentation", Computer Physics Communications, Vol. 183, Iss. 3, 2012, 821-843, https://arxiv.org/pdf/1002.0543.pdf

[Davidson2016] N. Davidson, T. Przedzinski, Z. Was, "PHOTOS interface in C++: Technical and physics documentation", Computer Physics Communications, Vol. 199, 2016, 86-101, https://arxiv.org/pdf/1011.0937.pdf

[Dobbs2001] Matt Dobbs, Jørgen Beck Hansen, "The HepMC C++ Monte Carlo event record for High Energy Physics", Computer Physics Communications, Vol. 134, Issue 1, 2001, 41-46, https://cds.cern.ch/record/684090/files/soft-2000-001.pdf

[Elbaum2006] Sebastian Elbaum, Hui Nee Chin, Matthew B. Dwyer, and Jonathan Dokulil, "Carving differential unit test cases from system test cases", Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT '06/FSE-14). ACM, New York, NY, USA, 2006, 253–264. DOI: https://doi.org/10.1145/1181775.1181806

[Endres2003] Albert Endres, H. Dieter Rombach, "A Handbook of Software and Systems Engineering: Empirical Observations, Laws, and Theories", Pearson/Addison Wesley, 2003

[Evans2007] Robert B. Evans and Alberto Savoia. "Differential testing: a new approach to change detection", 6th Joint Meeting on European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC-FSE '07). ACM, New York, NY, USA, 2007, 549–552. DOI: https://doi.org/10.1145/1295014.1295038

[Fink1997] George Fink and Matt Bishop. "Property-based testing: a new approach to testing for assurance", SIGSOFT Softw. Eng. Notes 22, 4 (July 1997), 74–80. DOI: https://doi.org/10.1145/263244.263267

[Florian2016] D. de Florian et al., "Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector", 2016, https://e-publishing.cern.ch/index.php/CYRM/issue/view/32/3

[Golonka2006a] P. Golonka, B. Kersevan, T. Pierzchała, E. Richter-Wąs, Z. Wąs, M. Worek, "The tauola-photos-F environment for the TAUOLA and PHOTOS packages, release II", Computer Physics Communications, Vol. 174, Issue 10, 2006, 818-835, https://doi.org/10.1016/j.cpc.2005.12.018

[Golonka2006b] Golonka, P., Was, Z. "PHOTOS Monte Carlo: a precision tool for QED corrections in Z and W decays", Eur. Phys. J. C 45, 97–107 (2006). https://doi.org/10.1140/epjc/s2005-02396-4

[Goodman2018] Peter Goodman and Alex Groce, "DeepState: Symbolic Unit Testing for C and C++", Workshop on Binary Analysis Research (BAR), San Diego, CA, USA, 2018, https://www.ndss-symposium.org/wp-content/uploads/2018/07/bar2018_9_Goodman_paper.pdf

[Gurp2001] J. van Gurp, J. Bosch and M. Svahnberg, "On the notion of variability in software product lines," Proceedings Working IEEE/IFIP Conference on Software Architecture, 2001, pp. 45-54, https://ieeexplore.ieee.org/abstract/document/948406

[Hastings1970] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications", Biometrika, Vol. 57, Issue 1, April 1970, 97–109, https://doi.org/10.1093/biomet/57.1.97

[Hicks2010] Michael Hicks, Jeffrey S. Foster, "SCORE: agile research group management", Communications of the ACM, Vol. 53, Number 10 (2010), 30-31, http://doi.acm.org/10.1145/1831407.1831421 (an extended version can be found here: http://www.cs.umd.edu/~mwh/papers/score.pdf)

[Hidalgo2019] Enric Senabre Hidalgo, "Adapting the scrum framework for agile project management in science: case study of a distributed research initiative", Heliyon, Vol. 5, Issue 3, 2019, e01447, https://doi.org/10.1016/j.heliyon.2019.e01447

[Jadach1991] Stanisław Jadach, Johann H. Kühn, Zbigniew Wąs, "TAUOLA - a library of Monte Carlo programs to simulate decays of polarized τ leptons", Computer Physics Communications, Vol. 64, Issue 2, 1991, 275-299, https://doi.org/10.1016/0010-4655(91)90038-M

[Jadach1993] S. Jadach, Z. Wąs, R. Decker, J.H. Kühn, "The τ decay library TAUOLA, version 2.4", Computer Physics Communications, Vol. 76, Issue 3, 1993, 361-380, https://doi.org/10.1016/0010-4655(93)90061-G

[Jadach1994] S. Jadach, B.F.L. Ward, Z. Wąs, "The Monte Carlo program KORALZ version 4.0 for lepton or quark pair production at LEP/SLC energies", Computer Physics Communications, Vol. 79, Issue 3, 1994, 503-522, https://doi.org/10.1016/0010-4655(94)90190-2

[James1990] F. James, "A Review of Pseudorandom Number Generators", Computer Physics Communications, Vol. 60, Iss. 3, 1990, 329-344, a copy available at http://cds.cern.ch/record/192937/files/cer-000104530.pdf

[Jeżąbek1992] M. Je.abek, Z. Wąs, S. Jadach, J.H. Kühn, "The τ decay library TAUOLA, update with exact $O(\alpha)$ QED corrections in $\tau \rightarrow \mu(e)\ \nu\nu$ decay modes", Computer Physics Communications, Vol. 70, Issue 1, 1992, 69-76, https://doi.org/10.1016/0010-4655(92)90092-D

[LaBrec2016] Paul LaBrec, Ryan Butterfield, "Using agile methods in research", 3M Inside Angle, June 28, 2016, https://www.3mhisinsideangle.com/blog-post/using-agile-methods-in-research/

[Leite2020] L. Leite, C. Rocha, F. Kon, D. Milojicic, and Paulo Meirelles, "A Survey of DevOps Concepts and Challenges", ACM Comput. Surv. 52, 6, Article 127 (January 2020), DOI: https://doi.org/10.1145/3359981

[Levy2002] G. Levy, "An introduction to quasi-random numbers", Numerical Algorithms Group Ltd., 2002, http://greeley.org/~hod/papers/Unsorted/introduction_to_quasi_random_numbers.pdf

[Loescher2017] Andreas Löscher and Konstantinos Sagonas, "Targeted property-based testing", Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2017). ACM, New York, NY, USA, 46–56. DOI: https://doi.org/10.1145/3092703.3092711

[Marsaglia1990] George Marsaglia, Arif Zaman, Wai Wan Tsang, "Toward a universal random number generator", Statistics & Probability Letters, Vol. 9, Issue 1, 1990, 35-39, https://doi.org/10.1016/0167-7152(90)90092-L

[McKeeman1998] William M. Mckeeman "Differential Testing for Software", Digital Technical Journal vol. 10, 1 (1998), 100–107. https://www.cs.swarthmore.edu/~bylvisa1/cs97/f13/Papers/DifferentialTestingForSoftware.pdf

[Metropolis1949] N. Metropolis and S. Ulam, "The Monte Carlo Method", Journal of the American Statistical Association, vol. 44, 1949, 335–341, http://hedibert.org/wp-content/uploads/2013/12/1949MetropolisUlam.pdf

[Petsios2017] T. Petsios, A. Tang, S. Stolfo, A. D. Keromytis and S. Jana, "NEZHA: Efficient Domain-Independent Differential Testing," 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 615-632, https://ieeexplore.ieee.org/abstract/document/7958601?casa_token=CQ7JsYWraHMAAAAA:VxEM-DMI-wXMUffoog4aBqpgzGiH2m3T1asyGaESAEWA9_uby7-SI6rZedfZ_uw_335avf194_E

[Poppendieck2003] Mary Poppendieck, Tom Poppendieck, "Lean Software Development: An Agile Toolkit", Addison-Wesley, May 8, 2003.

[Richter-Wąs2019] Richter-Was, E., Was, Z., "Documentation of TauSpinner approach for electroweak corrections in LHC Z→ll observables", Eur. Phys. J. C 79, Article number 480 (2019), https://doi.org/10.1140/epjc/s10052-019-6987-x

[Segura2016] Sergio Segura, Gordon Fraser, Ana B. Sanchez, and Antonio Ruiz-Cortes. 2016 "A Survey on Metamorphic Testing", IEEE Transactions on Software Engineering 42, 9 (2016), 805–824.

[Sjostrand2020] Torbjörn Sjöstrand, "The Pythia event generator: Past, present and future", Computer Physics Communications, Vol. 246, January 2020, 106910, https://arxiv.org/pdf/1907.09874.pdf

[Wąs1992] Z. Wąs, and S. Jadach, "TAUOLA Monte Carlo for r Decays. A Question of Systematic Errors", AIP Conference Proceedings 272, 1777 (1992) https://lib-extopc.kek.jp/preprints/PDF/1993/9302/9302133.pdf

[Wąs2001] Z. Wąs, "TAUOLA the library for τ lepton decay, and KKMC/KORALB/KORALZ/... status report", Nuclear Physics B - Proceedings Supplements, Vol. 98, Issues 1–3, 2001, 96-102, https://arxiv.org/pdf/hep-ph/0011305.pdf

[Wąs2005] Z. Wąs, P. Golonka, "TAUOLA as tau Monte Carlo for future applications", Nuclear Physics B - Proceedings Supplements, Vol. 144, 2005, 88-94, https://arxiv.org/pdf/hep-ph/0411377.pdf

[Wąs2015] Z. Was, "The τ leptons theory and experimental data: Monte Carlo, fits, software and systematic errors", Nuclear and Particle Physics Proceedings, Vol. 260, 2015, 47-51, https://arxiv.org/pdf/1412.2937.pdf

[Wiki-Meta] "Metamorphic testing", ver. 2021-03-28, https://en.wikipedia.org/wiki/Metamorphic_testing

[Yi2017] X. Yi, L. Chen, X. Mao and T. Ji, "Efficient Global Search for Inputs Triggering High Floating-Point Inaccuracies," *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, 2017, 11-20, https://ieeexplore.ieee.org/abstract/document/8305923?casa_token=DsUFuZMWhzoAAAAA:y3PeBGoK2kObyoSVpG4VNgHh_lqiu296BwrFkgg_iOkm3fablXpHZOcRkeywioj0VK8bZNRaSJs

[Yi2019] X. Yi, L. Chen, X. Mao and T. Ji, „Efficient automated repair of high floating-point errors in numerical libraries", Proceedings of the ACM on Programming Languages, Volume 3, Issue POPL, January 2019, Article No.: 56, pp 1–29, https://doi.org/10.1145/3290369

[Zhang 2017] T. Zhang and M. Kim, "Automated Transplantation and Differential Testing for Clones," 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 2017, pp. 665-676, https://ieeexplore.ieee.org/abstract/document/7985703?casa_token=tctbk8qJkv8AAAAA:vReOGpldZot3vaW5Fgzd45ojrif5SuHKkP1yH1FOy9XB28zKmPqu-ag9gbG22d7ce8tFPLnTE0c

[Zou2020] Daming Zou, Muhan Zeng, Yingfei Xiong, Zhoulai Fu, Lu Zhang, and Zhendong Su, "Detecting floating-point errors via atomic conditions", Proc. ACM Program. Lang. 4, POPL, Article 60 (January 2020), 27 pages. DOI: https://doi.org/10.1145/3371128