CritiX Research Book of Style

or, How to do Research in the CritiX Lab

https://wwwen.uni.lu/snt/research/critix/research_statement

Paulo Esteves-Veríssimo (CritiX PI)

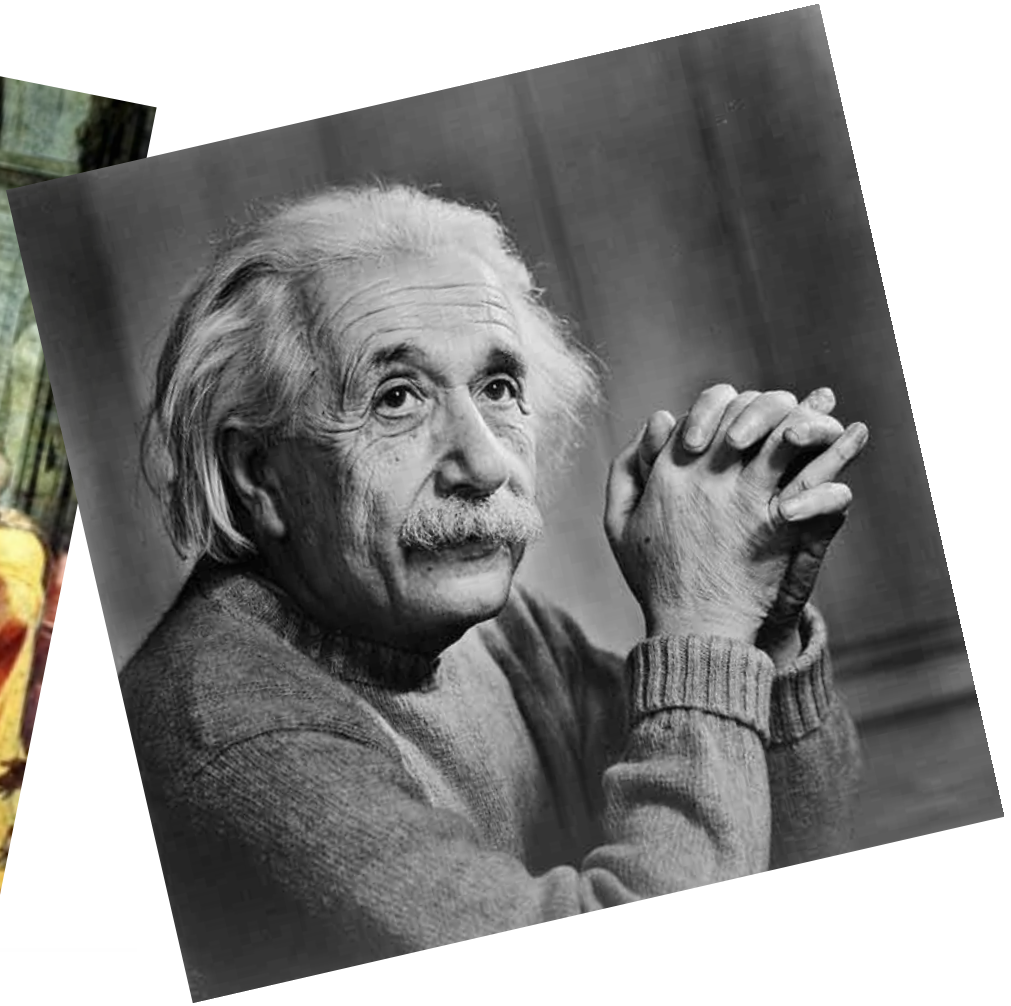
Univ. of Luxembourg SnT
Luxembourg

Paulo.verissimo@uni.lu
http://staff.uni.lu/paulo.verissimo/

INSPIRED by the Research Book of Style of my previous research group, the Navigators@LaSIGE in the Univ. Lisbon Faculty of Sciences, ADAPTED by the senior members of my current group, CritiX@SnT-UNILU Originally drafted by Alysson Bessani (contribs A. Casimiro, P. Veríssimo, http://www.navigators.di.fc.ul.pt/)

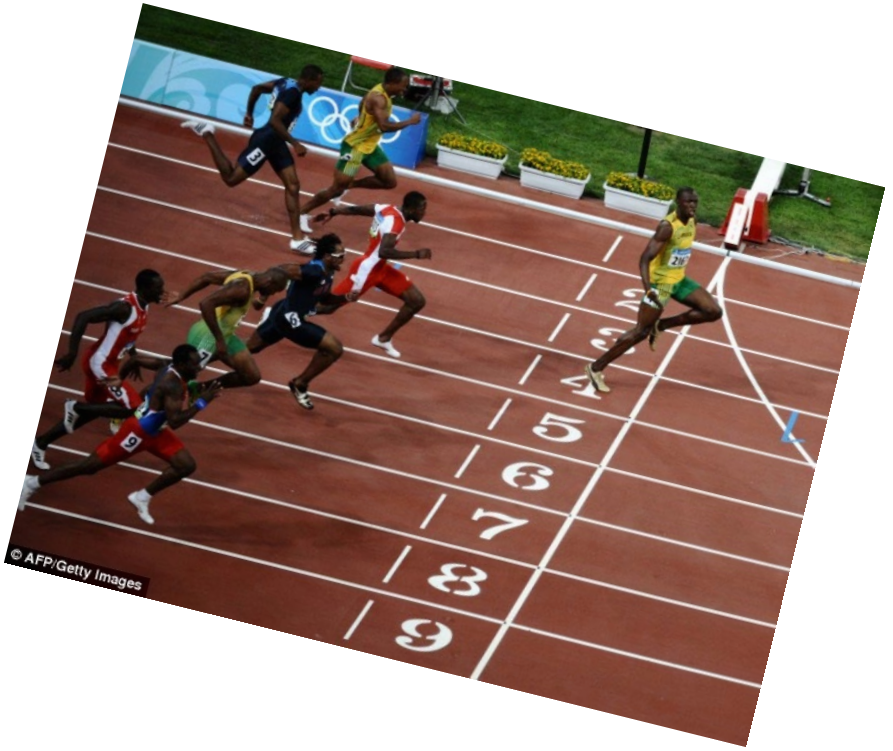
Last revised September 2019

Academia in general



Academia
(if you move in the first division)

- Top-level research is a highly competitive environment
- Like a premier sports league
- Top researchers are high-level competition athletes



Academia
(if you move in the first division)

securityandtrust.lu

- Is that too stressing?
- Where is the fun?
- Depends on the perspective …

Academia
(if you move in the first division)

- … is tough …
- but if you manage the right balance, you may go far,
- and have moments you'll never forget.
- How far?

Well, depends on you, this story just gives you some advice, after all, as you'll learn at the end, excellence is … about details…

A few notes for starters

Define your Objectives (Different Objectives at Different Levels)

securityandtrust.lu

An example:

Level	Publication	Quantity/yr
Undergrad	Meetings, soft pubs	1
Masters	Nat. Conf. A	1 or 2
	Nat. Journal A or Int. Conf. B	0 or 1
PhD/Pos-Doc	Int. Conf. A (A*)	1+
	Int. Journal A (A*)	1+
	Int. Conf. B	2+

The S.O.T.A. - Read a lot!

- Which are the conferences and journals in your field?
  - When you have the list, look them up, read title (maybe abstract) of relevant papers published over the last years
- How long back?
  - *Actually, as long as it takes...*
  - *There is nothing worse than someone telling you at a conference: "Ah, but this was invented 13 years ago!"*
- Download the ones you think are interesting (in accordance with your advisor) and read them
  - Periodically, go to the conference/journal ToCs to see what is new

The S.O.T.A. - Read a lot!

- How many papers per week?
  - there are no magic figures, but, when you are starting, be prepared to, *on average*:
    - explore 5 to 10 per week (abstract, intro, concl.)
    - read 3 to 5 per week
  - this includes: course assignments, your advisor suggestions, your initiative
  - it depends of the phase of your research

The S.O.T.A. - Read a lot!

(Don't worry if you don't understand everything)

securityandtrust.lu

- For each paper you read:
  - Ask yourself whether you understood it:
    - can you explain it *in your own words*?
  - Exercise your critical view!
    - Is the problem relevant?
    - Are assumptions realistic? Is the model sound?
    - What are the contributions? How practical the solution?
    - Is the provided evaluation/proof fair and/or rigorous?
    - Are experiments repeatable and comparable?
    - *How could you improve this work?*

Choosing a Research Topic

- Try to find a problem/topic that you care about…
  - Or, at least, find one whose importance you can explain
- Always ask the following questions:
  - What is the main contribution planned?
  - Why is it different from previous works?
- You NEED to know how to sell your idea as a worthwhile research topic:
    - to your advisor and mentor
    - to the Thesis Follow-up Committee (CET)
    - to the community when you publish later

The Advisor(s)

- Your advisor will help you, but it is YOUR MSc/PhD
- It is your responsibility to make your advisor be excited about your work and work on it with you
- **Golden rules to respect his/her time and effort :**
  - Be responsible with deadlines
    - Every deadline you miss, you lose a bit of the respect of your advisor
  - Be careful with the quality of what you deliver
    - Before delivering something to your advisor(s), ask yourself: *"Is this the best I can do?"; "Is the writing in acceptable shape?"*

Doing Research

Formalization

- Problem definition
  - Define your problem and show why solving it is important; why it is different from previous works
  - *A solution in search of a problem is just the wrong way*
- 'System' model
  - Define your constraints and assumptions
  - You should *characterize* unambiguously both the problem and the *environment* where the proposed solution is valid, *even if the 'solution' is a theorem*

Solution

- Presenting:
  - Intuition: give an intuitive overview of the solution
  - Self-containedness: choose the level of abstraction that fits the paper size
  - Pseudo-code: use good latex packages like algorithm2e to enhance presentation, use line numbers
- Formalizing:
  - Operation: describe the operation of your solution concisely but precisely, referring to the pseudo-code (refer to line nrs)
  - Proofs: no protocol/algorithm is correct until proven so
  - Metrics: prototype measurements or simulation may be useful ways of showing your point, whether or not you have made a proof

Evaluation
(Be honest and critic but don't be dumb!)

- Two attitudes to avoid
  - Being too smart: evaluating only the cases that you know are advantageous for your approach; ignoring negative outliers
  - Being too critic: over-evaluating, -discussing and/or -justifying the cases in which your approach is not the best one
- Other common mistakes:
  - Not defining the questions that the evaluation aims to answer
  - Not giving enough detail so that experiment is reproducible
  - Not justifying experiment's parameters and workloads
  - Not comparing the proposed approach with others
  - Not interpreting, explaining and justifying obtained results

Implementation

- If your work requires implementation, try first to modify something that is already done/used
- Advantages:
  - Well-written (maybe) code but above all it's tested
  - You (automatically) gain a basis for comparison
  - Makes the work more interesting for reviewers or thesis committee members
- Disadvantages:
  - Code from others is (generally) more complex than our toy examples and prototypes
  - The code may not work as expected (as with most papers' code put on online)

Writing Papers

Writing Papers

- Writing well is very hard!
  - *First step to writing well is reading a lot*
  - Then: *practice, practice, practice*
  - Every good paper is the result of many successive refinements
- Each paper has a "champion"
  - He/she is the owner of the paper, responsible for splitting the work among authors, asking for their parts and integrating the results in a single paper
  - *Never work on a paper without a champion!*
- Write the storyboard for yourself and other authors
  - *A paper should be a good story*

Writing Papers

- General wise-person's philosophy:
  - Tell people about the problem you are going to solve
  - Tell people how you are going to solve the problem
  - Tell them you solved it!

Or ... The art of writing introductions ...

Or ... The art of writing conclusions ...

Writing Papers

- TODO list:
  - Description of the problem
  - Make contribution and significance clear
  - Related work
  - Describe environment and model
  - Describe the solution
  - Validate your solution
  - Lessons learned (Why is your paper worth reading?)

Writing Papers

- What writing a good scientific paper is about
  - it must: (i) not only be correct; but (ii) perceived as useful by the community; and (iii) *interesting to read*
  - papers with just (i) count for your curriculum but they are *write-only papers*, i.e. papers that no one reads, ergo no one cites
  - papers with (i) and (ii) are ok, specially for *Calvinists*
  - papers with *all three*, readers will: love you for that, cite you a lot more, be willing to read your next one

Writing Papers

- Steps to writing a paper:
  - Write the storyboard
  - Build a structure (sections and sub-sections)
  - Each section must be filled with a bulleted list
    - - You are telling a story, each argument needs to be linked…
    - - A scientific text is an algorithm in itself (hence LaTEX☺) !
  - Add figures, tables, and informal references
  - Consolidate bullets into paragraphs
  - Collect formal references and related work
  - Reiterate by successive refinement until done

Writing Papers (wrap-up)

- The introduction needs to be **perfect**
  - Most reviewers can decide to reject your paper after reading the introduction
- Same for the presentation and style (text, figures and general appearance)
  - Remember, we don't do write-only papers
- Ask for feedback from your colleagues
  - Sometimes better if some don't work in the same area (like reviewers); *feedback* is fundamental!
  - Include a couple of outside experts

Submitting Papers

Submitting Papers

- There are several possible venues, they all have their role:
- Workshops – ***trendy ones***
- Conferences - ***heavy-weight ones***
- Journals – ***reputed ones***

Submitting Papers

- Workshops
  - Very good for
    - disseminating early results
    - discussing a problem
    - getting feedback
    - meeting other people working on your area
  - Counts little for CV evaluation
  - Some of them are very good (and competitive): HotOS, HotNet & HotDep, CERTS, SPW, XXXX

Submitting Papers

- Conferences
  - The really good conferences in CSE may be harder and have more prestige than the best journals from IEEE/ACM
    - TYP acceptance rate less than 20%
    - Papers with 12-16 pages (as long as some journals!!!)
  - These are what we call ***heavy-weight conferences***
  - PCs in each community expect a particular style of papers, so before submitting to a top conference, try to learn their style (i.e., read a lot!)

Submitting Papers

- Some Good to VG conferences (not complete):
  - Distributed Systems: ICDCS, IPDPS, Middleware
  - Distributed Syst. Theory: PODC, DISC, OPODIS
  - Dependability: DSN, SRDS, ISSRE
  - Security: S&P Oakland, CCS, USENIX Security, NDSS, Crypto
  - Networks: SIGCOMM, INFOCOM, NSDI, CoNEXT
  - Systems: SOSP/OSDI, EuroSys, USENIX ATC
  - Real-time: RTSS, RTAS, ECRTS
  - Programming: POPL, ICFP, PLDI, ESOP, OOPSLA
  - XXXXXX

Submitting Papers

- Acceptance rate
  - A good half of the papers submitted to a top conference *don't stand a chance* even before the PC show starts
  - From the remainder, bottom half have *little chances*
- If you follow the rules presented, you have:
  - a *good chance* of staying out of the sudden-death half, right from the beginning
  - Getting to the top quarter and *fighting for an accept* is another thing…

Submitting Papers

- As you build experience, you should aim to *systematically* be in the top 25%
  - You get to know that because reviews get better…
  - Above a certain standard, fair English is an obstacle --- not making mistakes is not enough, you need style.
    - Improve! (subscribe to, say, National Geographic ☺)
    - Rely on senior co-authors! Their touch makes the difference
- Still, you paper may be accepted or not ☺
  - Everyone has rejected papers!

Submitting Papers

- Journal
  - Disadvantages:
    - “arguably” less immediate visibility, which may be counterproductive in a lively field as CSE
    - to overcome this, consider first submitting to conferences and evolve best works to journal
  - Advantages:
    - Science bureaucrats love it, gives substance to your CV
    - plus it ***does makes sense***, it’s an archival grade work

Submitting Papers

securityandtrust.lu

- Papers in the best journals are ***substantive*** and ***archival grade***
  - Clear and complete contribution in a subject
  - Rigorous in the formalization, proofs or metrics
  - Carefully evaluated, no loose ends
- Reviewers are generally more responsible and accountable
  - You have a chance for a dialogue and rebuttal

Submitting Papers

- Some Good to VG journals and magazines (far from complete) in no special order:
  - IEEE Transactions on …
  - ACM Transactions on …
  - Journal of ACM
  - Distributed Computing (Springer)
  - Journal of Real-Time Systems (Springer)
  - Computer Networks
  - IEEE Security and Privacy
  - Journal of Computer Security
  - Journal of Parallel and Distributed Computing
  - Computer Journal
  - Journal of Functional Programming
  - XXXXXXXXXX

Submitting Papers

securityandtrust.lu

– Revising and Responding to Reviewers

- Always show that you took reviewers' comments into account, through the response letter
- Consider politely challenging the review points with which you don't agree, the editor is an arbiter between you and the reviewer
- A *good method to prepare both your revision* and your response, is to pass all reviews to a text processor and exhaustively comment all significant remarks *in-line in different colour*, proposing what to do to address or challenge, *to be discussed with your co-authors*

The Reviewer

- Often (though not always) reviewers are very smart and have good intentions
- However,
  - They don't have time
  - They expect fair amounts of scientific and/or engineering work
  - They may not be experts in your topic
  - Some (rare) may actually not have good intentions

The Reviewer

- Keep these things in mind:
  - Don’t make it easy for them to reject your paper
  - Try to finish it up as sphere (no place to grab)
  - Citations are free, certain people don’t like not to be cited
  - Don’t belittle past work that you are advancing from:
    - you should step on others’ shoulders, not on their toes
    - you may be next…
  - In rebuttal or response, be friendly, not a good idea to antagonize the reviewer

CritiX' Publishing Policy

- May submit preliminary work to a *good* workshop
- Submit a finished paper to a *VG+* conference
- If accepted, great!
- If it is worthwhile, prepare an extended version (at least 25% of new content) and submit to a journal
- If rejected, ask yourself:
  - Some problems or just unlucky? ***Solve them and try again***
  - Misunderstood? Under fire? ***Improve and send to a journal***

To Conclude…

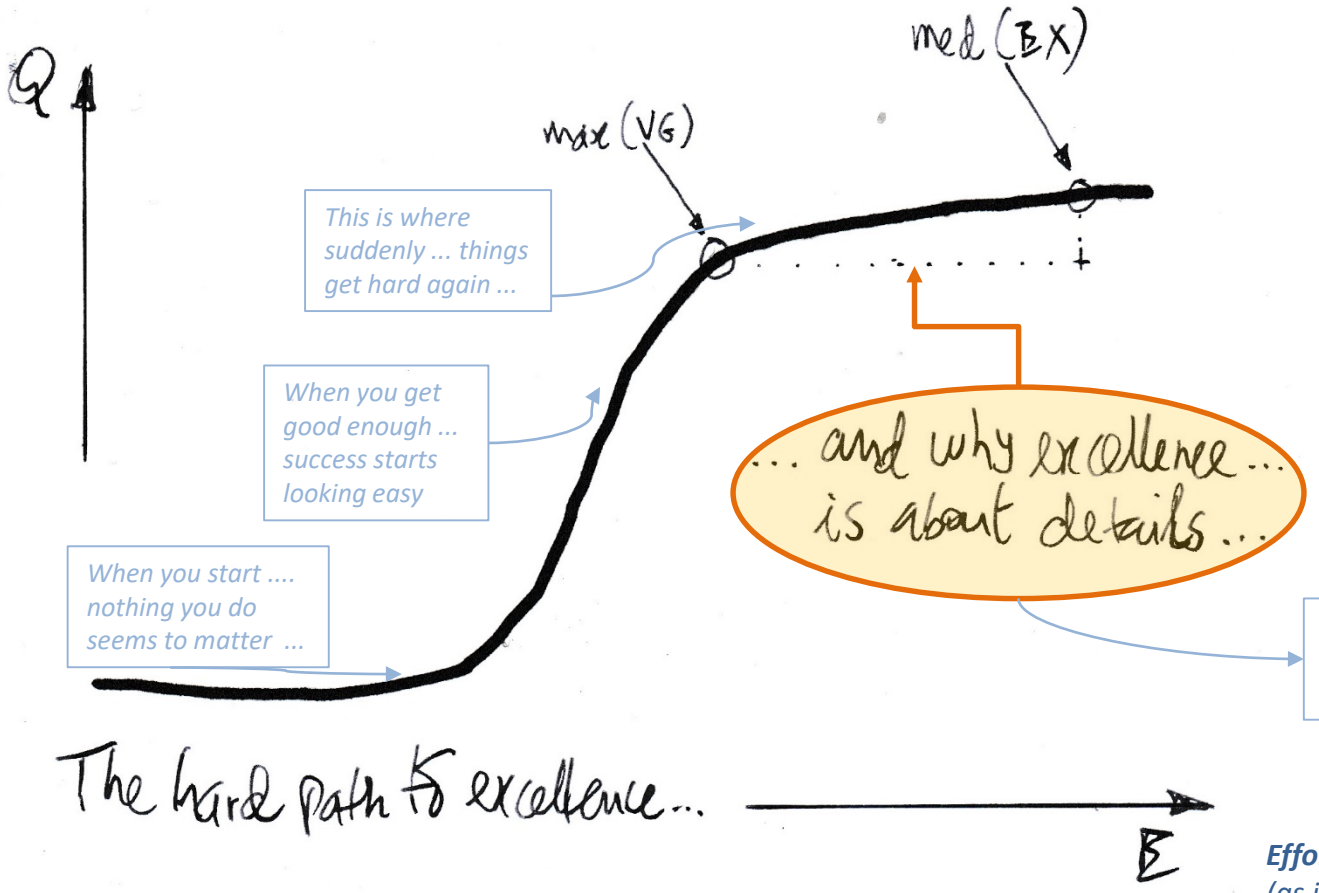
securityandtrust.lu

- *What you get* for staying in the academia:
  - You don't need to work under direct orders
  - You get to participate in defining what you work on
  - You get to know the world and meet the smartest people
  - You have substantial freedom to manage your time

- *What you must give*:
  - Reciprocate with top quality, self-responsibility, team spirit
  - Work hard! ***Be better than you were yesterday!***
  - Love what you do and be proud of how good you are
  - Don't be afraid to have ideas, ask questions, criticize
  - ***Be your greatest critic!*** But accept constructive criticism

The hard path to Excellence, or … Why excellence … is about details …

securityandtrust.lu
CritiX



Quality
(as perceived: impactful, recognised)

Effort
(as in work, energy used …)

Thank you!

securityandtrust.lu

UNIVERSITÉ DU LUXEMBOURG

CRITIX

Paulo Esteves-Veríssimo

SnT, Interdisciplinary Centre for Security, Reliability and Trust - University of Luxembourg

paulo.verissimo@uni.lu
http://staff.uni.lu/paulo.verissimo/

«Today, try to be the best you can be. Tomorrow, try to be better than you were yesterday. Repeat every day.»

CRITIX @SnT, *Critical and Extreme Security and Dependability*

We're hiring bright post-docs and research associates willing to address these challenges!

UNIVERSITÉ DU LUXEMBOURG

Some References

- Alysson Bessani, with A. Casimiro, P. Veríssimo, ***How to do Research in the Navigators group***
  - http://www.navigators.di.fc.ul.pt/w2/img_auth.php/e/e6/The_Navigators_Research_Book_of_Style.pdf
- Simon Peyton Jones, **How to give a good talk and write a good paper**
  - http://research.microsoft.com/en-us/um/people/simonpj/papers/giving-a-talk/giving-a-talk.htm
  - https://www.youtube.com/watch?v=g3dkRsTqdDA
- Randy Pausch. **The Last Lecture**.
  - Vídeo: *http://www.youtube.com/watch?v=ji5_MqicxSo*
- Strunk & White. **The Elements of Style**. 4th edition. Longman.
  - Book about the style of written English, highly recommended.
- Phil Guo. **The PhD Grind - A Ph.D. Student Memoir**. 2012.
  - http://pgbovine.net/PhD-memoir/pguo-PhD-grind.pdf
- Levin & Redell. **How (and how not) to write a good systems paper.**
  - http://www.usenix.org/event/samples/submit/advice.html
- John Wilkes. **How to write a good [systems] paper**, *EuroSys 2006 Authoring Workshop*
  - http://www.cs.kuleuven.be/conference/EuroSys2006/authoring_workshop/wilkes.pdf
- Material in the pages of professors Priya Narasimhan and Mike Dahlin
  - http://www.cs.cmu.edu/~priya/tech-writing.html ; http://www.cs.utexas.edu/users/dahlin/advice.html
- How to write a bad research paper (R. Guerraoui):
  https://www.youtube.com/watch?v=K9BhQaOdtjs