Draft Guidelines for My Students on Writing Software Engineering Research Papers

Mark Harman
University College London, UK

ABSTRACT

Here are some tips for academic writing in Software Engineering papers. They may not be suitable for all papers, but they are likely to apply to any you co-author with me. When I comment on your work I may refer to the principles named here so that I can be more productive. Since this document is *not* a software engineering paper itself, do not expect me to follow my own tips. For example, this abstract contains only 1 Zeller Number, and it is not a 'key number that quantifies the primary findings of the paper'.

1. PATTERNS

Favour precision: Referees hate vague statements. Rather than saying "several programs crashed" say "3 of the 17 programs crashed". Do not write things like 'this usually works'. Rather, you need to define what is meant by 'working' (so that others could also investigate) and give numbers. Precision does not require a point estimate; where the value is uncertain, try to bound it with a range. For example, in place of "the computation took roughly 2 seconds" say "the computation too between 1.7 and 4.9 seconds". the latter formulation gives far more information at the expense of little extra text.

Context-free figures, tables and other boxouts: Your referee will typically skim your paper first. Their eyes will often alight at a figure or table. What can they learn from reading *just* this figure/table and its caption? In a well-written paper, the skimmer is catered for. When your readers' eyes alight at a figure, for example, they can understand its message in isolation. Avoid the situation where the reader has to scan the text to find where the figure is mentioned (which they would otherwise need to do in order to understand the terms used and the messages you wish to impart in your figure). The same applies to tables and any other 'boxed out' or otherwise highlighted parts of your paper.

Last updated November 2014. These are the personal views of Mark Harman and are intended for his students, but if you find them useful feel free to share. Thanks to Earl Barr, Licia Capra, Jens Krinke, Emmanuel Letier, Federica Sarro and Shin Yoo for their comments. Please report faults in this document to Mark.Harman@ucl.ac.uk so that I can *optimise*, which I hope to do continually. I am always trying to learn myself, so this document will forever be a draft.

Zellerise your abstract: Via Gordon Fraser, I learned of Andreas Zeller's principle that an abstract should contain numbers; the key numbers that quantify the primary findings of the paper. I called the process of adding such numbers *'Zellerisation'* in recognition of him. Zellerisation is not always appropriate for every paper, but it often is and even in cases where you decline to use the numbers you obtain in the abstract, the process of thinking about them is helpful. You will be surprised how effective it is. I believe that the process of deciding on the Zeller numbers helps us to focus our minds on the main findings and messages of the work.

Précis and précis again: Good writing is succinct, particularly the abstract. Act on this observation continually.

Titles matter: Think carefully about your title. Be ready to change it if you can find a better one. A good title is memorable (thereby increasing citations, discussion and conversational mentions). It also is the first 'abstract' of your paper that the referee will read; do not make it catchy at the price of irritating or confusing the referee.

Numbered and named RQs: Number your Research Questions (RQs) so that you can refer to them. Make sure you clearly and definitively answer them in a single well-identified place in the paper. Summarise them as contributions at the end of the introduction and pick out the key numbers for Zellerisation. Try to give them a single memorable pithy name that conjures the question. If you cannot find a single word, use a pithy phrase and shorten it as much as you can. For example, you might start out with 'Assessing the effect of approximation on our results', but could change this to 'Approximation Effect' (its obvious you are 'assessing' it, if it is an RQ and you are clearly interested in its effect on *your* results). You can also almost always find a way to avoid prepositions and other incidental words (so 'Approximation effect' rather than 'Effect of Approximation').

Succinct and Pithy Names: Let's Take RQs as an example of one (of many) situations, where pithy names are required. Once you have found a suitable noun phrase to refer to your RQ, this will greatly aid comprehension. For instance, using the previous example, you can now say: "We find that the Approximation Effect is low . . . blah blah". In writing about it as a noun phrase, you realise that it would be better to call it the 'Approximation Influence', rather than 'Approximation Effect', so that it conveys the exact meaning you intend.

In this way you go through process of refining your thinking *by refining your prose*. Try to give pithy evocative names to as many technical things as you can, rather than referring to them with more cumbersome clauses or sentences. If you get it right, others will copy your pithy naming conventions. If you get it wrong, then your excellent results may go unnoticed (even if they are published). Early in my own career, I re-christened what I was calling 'Syntactically Unrestricted Program Slicing' to 'Amorphous Slicing'; which name do you think caught on better among slicing researchers?

Space: You have a page limit. It should look like it was a struggle to fit everything in (but not too much). It will not look like a struggle is there is *any* wasted white space on the page. It will be too much if anything is hard to read.

Paragraph breaks over a page: As a last step, go over your paper and try to make it as beautiful on the page as possible. For example, arrange paragraphs to complete neatly at the end of each column and favour choices that place as many headings at the top of a column as possible.

2. ANTI-PATTERNS

I start with two absolutely forbidden scientific malpractices, although I sincerely hope you already know to avoid these as though your life depended on it.

Plagiarism: It should not be necessary to state this, but I will, because there can be no room for doubt. Absolutely *any* plagiarism, no matter how slight or unintentional, is just about *the* worst scientific crime you can commit. It is theft. It will destroy not only your own reputation, but that of all your co-authors too. They certainly wont thank you for it. If they have worked on developing their reputation for decades, you can imagine what their reaction would be. **If you are in the slightest doubt about what is (and is not) plagiarism, make sure you talk to me and your other supervisors about it right away**.

Concurrent Submission: If you submit a paper P_1 to a venue V_1 and, while it is under review at venue V_1, you submit a paper P_2 to a venue V_2 where P_1 and P_2 contain significant overlap, then this is a concurrent submission. Concurrent submission is another way to ensure abrupt termination of your career. If you are in any doubt about whether P_1 and P_2 contain 'significant' overlap then consult me and your other supervisors *before submitting either paper*. You should *never* submit a paper without your supervisors' clearly expressed consent in any case. Like plagiarism, concurrent submission will also constitute a massive reputational hit for your university, supervisory team and co-authors. They are unlikely to be understanding, nor sympathetic to any 'mitigating circumstances', since there are none.

The presence of any of the remaining anti-patterns, listed below, may cause your paper to be rejected. However, they are all less serious than plagiarism and concurrent submission, which will cause your entire career to be terminated. If your paper contains the anti-patterns below, then you are **insulting your reader** by implying that their time is less important than yours. No reader likes to be so-insulted. If your reader happens to also be your referee, then they have ample recourse to punishment measures. The anti-patterns below wont kill your career, but they may well kill you paper's chances of acceptance and they may also test your supervisors' patience and forbearance.

Avoid prose-less sections: Sections like Section 1 look ugly, because they go straight to a subsection with no prose.

Avoid Repetition: In your first draft, you will find that you have said the same thing in several places. Some will be better worded than others. Decide where to put the statement you wish to make and pick the best way to say it. Have the strength of purpose to delete all the others. Good writing involves more deleting than inserting.

DeFrag: You may have given glimpses of technical information related to a single aspect of the work in several places in the paper. *Defrag*ment your paper by collecting them together and writing one coherent account, under a suitable heading so that it is easy to find.

Avoid hyperbole: Early drafts often contain a particularly irritating form of overstatement that uses overly emotive and hyperbolic language. This is sure to irritate the referee and result in rejection. Words to watch out for and avoid include, but are not limited to, 'massively', 'vast', 'enormous', 'huge' and also, of course 'tiny', 'slight', etc. These sorts of words are in common usage in spoken and written prose, but they are inherently unscientific.

Alarm Bell Ringers: Some statements about your approach, though necessary, are guaranteed to raise alarm bells, if badly written: statements about implementation or experimental process that could *conceivably* affect your results. It does not matter whether your results *were* affected, merely that the referee might *believe* that they could be. You need to include statements so that your reporting is complete. Avoiding them is ethically and scientifically wrong. However, you need to help the referee (and your readers) to understand the precise affect on your results. For example saying "We filtered out all programs that did not compile using our tool", or "We ignore results where the flibberbibble measure is under 5%", or "We normalise the data". In all cases it is right to state what you did (to facilitate replication, for example), but you need to explain the motivation for and the effect of the intervention.

Claims need evidence: Science is about evidence. If you make a claim you need to either provide evidence in your paper or cite a paper that provides evidence to back up your claim. If you make claims without evidence the referee will quickly learn to mistrust you and this will lead to rejection. It will suggest to the referee that the paper's authors are sloppy and unscientific. If you cannot amend a claim so that you can back it with evidence then you most likely need to remove it. For experts only, you might try to couch the claim in words like "the folklore of <topic X> suggests that . . . " or some such, but this is only worth including if it is super-important to your paper. For example, "It is widely believed that side effects are harmful to program comprehension. Despite this being a widely-held opinion [cite], there is little scientific evidence [the few papers that offer scant evidence]".

Sloppy references = reject: The referee is likely to read your reference list early in the review process. Maybe they use it to quickly asses your familiarity with and mastery of the topic. Do the papers they expect to see cited (including their own!) appear in the list? You never get a second chance to make a first impression. If your reference list is sloppily presented or misses key papers then you are headed towards the reject pile.

Avoid colloquial English: Colloquialism almost always looks bad. Expert writers can sometimes use it to great effect. If you are not an expert writer, it is best to avoid it.

Avoid stating the obvious: If the referee thinks you state the obvious, she or he will likely assume your scientific and technical understanding is low and will dismiss your paper. It is hard to judge what is obvious so seek advice. If two or more people thinks it is obvious, then it probably is. If the first two you ask think it unobvious, then it still might be obvious, so check widely, unless you are sure. Be ready to help each other with such quick reviews and sanity checks; we are a *community* of scholars and can all greatly benefit from the services we collectively provide to each other and the community as a whole.

Bullets: Bullet points eat space and break the flow of the paper. They look sloppy because they are a sign that you were too lazy to think of a better way to structure the points you wanted to make. They are prevalent in papers (and other writing) because people *are* generally lazy creatures. There is a high correlation between the number of bullet points and chance of rejection:

- Bullets look sloppy and capricious and they eat space
- They are especially ugly and sloppy if the point you make goes over a single line, like this one.
- Were these points *really* worth the attention given to them by the bullets?

You might choose to use a bullet list once, maybe twice. Choose very carefully those things that you really think deserve that amount of space and highlighting that the use of bullet will provide. Even in these cases, consider an enumeration (rather than bullets) and ensure that the list is not too long. If you feel you need more than two such lists, or any list becomes longer than 3 or 4 points, then give some serious consideration to whether it would be better presented as a table, a figure or some other form of schematic diagram.

Bait and Switch: If you introduce a topic, t, and then move to another unrelated topic without saying anything substantial about t then the reader will quickly stop reading. This 'bait and switch' style of writing is a sign that the document has been rushed or immature; either way it is a great insult to the reader.

3. COMMON GRAMMAR ISSUES

Avoid contractions. E.g. write 'Do not' should be used, rather than 'Don't', and 'is not' rather than 'isn't'. Figure 1, Section 2.2.1 and Table 4 etc, are all proper nouns (they name something) and so they should be capitalised. Columns of numeric data should be right justified. Understand the difference between 'whether' and 'if'. Understand the difference between 'that' and 'which'. Check and re-check for inconsistencies (e.g. British and American Spelling). Be aware of the words that cannot start a sentence (e.g. 'or', 'and' 'then'). Use a spelling checker, but check words it wont report (e.g. did you mean 'from' or 'form'?; a mistake I often miss). Please avoid assigning gender unless relevant. I hate to read things like 'The programmer can change this if he wants to'; it really jars in my mind, distracting me from your message, and it instantly excludes 50% of population. *Avoid over emphasis* (which I did not do in this document).

4. SOURCES OF OTHER ADVICE

You might also seek advise on general English language writing style such as the book 'Elements of Style' by Stunk and White, but please be aware that such general advice may be the subject of ongoing debate and may have detractors[1]. Make sure you distinguish between such general advice and specific advice that focusses on what is expected for *your research community*.

Mary Shaw is a well-known software engineering researcher, who has made major contributions (for example, in the area of software architecture). She pulled her considerable wisdom into a paper[2]. There is also a slide presentation[3].

Simon Peyton-Jones has advice (in the form of a slide presentation) on writing papers that contains many very valuable hints and suggestions[4]. This is not specifically aimed at software engineering research papers, but is relevant to computer science papers in general.

Several authors have also published lists of 'pet peeves', or other lists of things to avoid in writing papers. It is particularly important to take account of these where the pet peeve list comes from a software engineer who is regularly used as a reviewer. One such example is the list by Andreas Zeller[5]. I have to say that I personally agree with every one of these 10 and strongly encourage you to avoid all of them.

Another source of advice (on many related topics) can be found in the (extensive) collection of 'Advice on Research and Writing' curated by CMU[6]. I am afraid I have not had the opportunity to read all of this material myself, so cannot say more than 'definitely worth checking out'. Some of it is old now and a few links are broken when I last checked. Let me know if you find any other useful material 'out there' and I will periodcally update this advice.

Finally, I would also like to mention that, although I have been writing papers for 25 years now, I still feel that I am learning how best to make detailed scientific arguments in research papers. The wonderful thing about academic life is we are all able to remain in a state of *continual learning*.

5. POSSIBLE PAPER STRUCTURE

Here are some tips on a 'standard' format and how to approach the sections of a Software Engineering paper. You might not need all of these sections and you might give them other names.

I have been rather prescriptive in my discussion, simply because it makes it quicker to get my points across. Feel free to vary from these guidelines when the scientific message and evidence require it, but make sure they do (and be ready to justify deviations to me).

You can think of section structure as a tool for separation of concerns. Badly written papers do have sections, but they fail to adequately separate out the concerns; topics cross cut over the sections and the reader quickly becomes irritated.

---

[1] chronicle.com/article/50-Years-of-Stupid-Grammar/25497
[2] www.cs.cmu.edu/~Compose/shaw-icse03.pdf
[3] www.slideshare.net/alexorso/how-to-get-my-paper-accepted-at-top-software-engineering-conferences
[4] research.microsoft.com/en-us/um/people/simonpj/papers/giving-a-talk/writing-a-paper-slides.pdf
[5] andreas-zeller.blogspot.co.uk/2013/04/my-top-ten-presentation-issues-in.html
[6] www.cs.cmu.edu/afs/cs.cmu.edu/user/mleone/web/how-to.html

Your first draft (and mine too, most likely) will very likely suffer from this failure. With care, it can be optimised. Use the sections to separate out all the concerns in your message. As software engineers, the importance of separation of concerns ought to be obvious, yet if often fails in our writing.

Abstract: Look at lots of other abstracts (in top venues, e.g., ICSE, FSE, ASE, ISSTA, RE, TSE, TOSEM) to get a sense of format. Abstracts should not go over a paragraph, should not contain references and need to get to the point quickly. Try to answer these questions in your abstract: What is the problem?, Why does it matter?, What is your solution?, What scientific evidence do you offer that your solution solves the problem (Zellerise)? You might start out with each of these answers occupying a paragraph. That is fine as a starting point. First, focus on making sure you say all you need to say to make the answers compelling. Once you feel they are compelling, précis, précis, and précis again, until your whole abstract is one paragraph long, but retains all that semantics from the earlier longer versions. Just like programming: get it right first, and only when it is right, optimise to make it tight and fast.

Introduction: First tier venue referees may not be expert in your research area. They will need an introduction that gives them a good feel for what you are proposing. It should explain to them, with clear and compelling evidence, why they should *care*. They may be reviewing 15–25 other papers in a short period for the same conference.

At the end of the introduction, it is now standard to include a short statement about the contributions of your paper. This is one place where you might *legitimately* choose to use an enumerated list (but not bullets please). This contribution claim is essentially the basis on which you say to the referee 'this is the essence of why you should accept my paper'. Therefore, just like the abstract, it is something that you should refine and re-refine over and over again until it is as tight and compelling as possible.

By the time the referees read your contribution claims they may have already decided whether to accept your paper. If they decided to accept your paper, then they are simply reading the rest of the paper out of interest and to check for any unexpected issues (and to provide a little bit more meat for their review). If they decided to reject your paper, then they are probably just reading the rest of the paper to find some other things to dislike and write about in their review, to back up the gut feeling that had at the end of the introduction. Not all referees are like this, but you need to cater for those who are.

You certainly need to think about the first impression your paper makes, and this is usually made by your paper's 'envelope' and box-outs. That is the title, abstract, introduction, conclusions, and reference list (the envelope) and the equations, figures and tables (box out items). By 'box out' I do not mean that they have to appear 'in a box', merely that they stand out when the reader flicks through the paper.

Backgroud/preliminaries/related work: Most papers have a related work section, which is located just before the conclusions. As an alternative to a related work section you may have a section called 'background' or 'preliminaries', which usually occurs just after the introduction. In very rare cases you may need both of these two kinds of context-setting sections, but it is *extremely rare* that you would have neither.

Most of the papers I have written have had a related work section, but neither a background nor a preliminaries section. In papers they require a great deal of scene-setting, for example definitions, mathematical notations, or the precise establishment of the current state-of-the-art, then an earlier background section may be appropriate.

In most papers, it usually proves sufficient to set the scene for the current state-of-the-art in the introduction, and then delay discussion of related work to the end of the paper. The related work section should explain, not only what other authors have done, but also how that work relates to your own contributions. In particular, I recommend concluding the related work section with a single pithy paragraph that summarises the novelty and advance on the state-of-the-art that your paper contributes. Naturally, it is easier to do this if you have already covered the body of your paper, and all the related work. This is why I recommend a related work section rather than a background section.

However, sometimes, it may be more convenient to move this discussion to the front of the paper as a background section. This might happen, for example, where you need to include survey-like material in order to establish the importance of your problem or the nature of the gap in the current literature that your work fills.

The disadvantage of having a background section rather than a related work section is that it takes longer for the referee to get to the novel material in your paper. That 'novelty delay' can have a bad psychological effect on the referee. Therefore, if the background section introduces new notations and definitions, or if you contribute a new survey of the literature as part of your paper, then this is okay because you *are* providing novelty at this point. However, if your background section is merely a description of what other people have done, then I recommend moving it to a related work section.

Problem formulation: If you did a good job in the introduction and the problem is well-understood, then you may not need a 'problem formulation' section. However, if the problem formulation is complicated, you may want to put it in a separate section on its own. Make sure you clearly separate concerns, so the problem formulation is not tainted by any discussion of proposed solutions or evaluation, and the formulation is the definitive one (*defrag*).

Proposed solution: If you introduce a nontrivial novel technical approach you need a section that contains the definitive explanation. Be careful to separate out the proposed solution that you offer from the way in which you intend to evaluate it, which should come in a separate section. Also be careful to separate out the description of the problem (whether it is located in the introduction or separate section), and the way in which you are going to go about solving it.

Finally, although you might talk about implementation details, you should separate out implementation aspects from algorithmic aspects. Readers are interested to understand the ideas, the algorithm, and the overall approach. If you evaluated with your own implementation, then you may need to include some implementation details, but keep them separate from the description of the algorithm and approach. This is very important. A paper that mixes implementation details with high-level approach description is hard to read and irritates referees. It is a common failing in early drafts.

Experimental Set up: Most software engineering papers have some kind of evaluation. I called the section "experimental set up" but it might have many other names. In this section you explain how you propose to go about demonstrating that your solution *does* address the problem. This means explaining the subjects on which you chose to evaluate, the statistical tests or other measurements and techniques that you applied, the research questions you set out to ask (and the *motivation* asking them).

Do not *pollute* this section with results or details of the proposed solution, but to keep it entirely about *how* you evaluated your proposed approach. As a sanity check, ask yourself whether you could write the same section about someone else's proposed solution to the same problem without knowing anything about their results. If you can, then you probably got the separation of concerns about right.

Results: The results section contains the results from your evaluation, presented with the aid of tables and figures. Typically this is presented first, and then explicit answers to each research questions are provided separately. This separates the evidence (provided without any judgement), from the authors' view on how this evidence answers the questions (which generally involves judgement). I would recommend trying to establish a one-to-one correspondence between figures/tables and the research questions they address. If you find that this becomes many-to-one, then maybe you need to refactor or reconsider your research questions; perhaps some structure with subsidiary research questions would help (RQ1.1, RQ1.2 etc.)?

I find a paper to be particularly clear when it is structured like this: there are a number of overall claims in the introduction, which map directly onto research questions at the top level (and maybe subsidiary research questions, where appropriate). These research questions each have a single point at which they are conclusively answered in the paper, with reference to a single table or figure. From these figures or tables, Zeller numbers are selected and included in the abstract, characterising the primary 'take home' message of the paper.

This is an idealised scenario, but it is a good idea to aim for the ideal, and then think about the justification for not meeting that ideal. It is something to aim for, but do not follow it too slavishly; nothing should be done that gets in the way of the overall message or the clarity of scientific presentation of the evidence.

Limitations and Threats to Validity: Papers typically contain a section that discusses limitations or 'threats to validity' (which is a shorthand for 'threats to the validity of the findings reported in the paper'). When this section is written badly it reads like an apology. When it is written well it reads like a sincere attempt to encourage the reader to take up the research agenda and address some of the limitations to advance the research and the state of knowledge. There are many papers that discuss the purpose and style of 'threats' sections, so I will not go into further detail here.

Actionable Findings: I have found that it is increasingly important to think about the actionable conclusions from your research; what will change in the future research or practice (or both) as a result of the findings of your work? This helps us to think about the possible impact of our research, and it helps the referee to understand why they should care about what we have discovered.

If you find the section hard to write, then it may be a sign that your research needs more work. Even if you decide *not* to include the section on your paper, then it is a good intellectual discipline to try to write it. It just might speed up the transfer of your ideas into practice.

Conclusion and Future Work: The conclusions and future work section is typically quite hard to write. What can be written in the conclusion that we did not already say in the abstract or the introduction? I think that the answer to this is relatively simple: The conclusion can restate the primary contributions and findings of the paper. The principal difference between this and the similar statement in the abstract and introduction is this: the abstract and introduction can only include technical terms that they define or forward reference (both of which are cumbersome), while the conclusion already has these available. It is a good idea to keep the conclusion as short as possible, without underselling the contribution and to avoid forward references in the introduction. Writing the abstract without (full) definitions is a skill only acquired through practice and familiarity with what the reader is likely to 'take as read'.

Future work should be your genuine view about what the next steps are and why they are important. It can be either what you intend to do yourself, or what you hope others will do as a result of reading your work. Once again, it should be short. A long future work discussion is a sure indication that the paper has underachieved; it has left undone that which ought to have been done. You, yourself, indicated to the referee that these things *ought* to have been done by your use of long descriptions.

Figures: Figures are very important. Make sure they are readable, even though you may have reduced them to fit. This may mean increasing the point size of labels (such as axis labels) relative to the size of the overall figure. Respect the 'Semantic Information Principle (SIP)', which states that every discernible feature difference should have a meaning; if any two distinct details are unified then meaning is lost. For example, if one box has round edges and another has square edges, what information is they conveying? Avoid superficial features that do not convey meaning, such as shading behind boxes or background colours on plots.

Tables: Columns of numbers should always be right justified. Lists of names (e.g., names of programs) should be left justified. There are very few cases where centre justification is suitable, but it can work if the data does not require left of right. A column with entries all of different types can be more readable when centre justified. Units should appear in the column heading and not repeated in each row. In general if you see repetition and it does not serve an important purpose (e.g., repetition of the world 'Section' at the start of each section) then it is a sure sign that some refactoring is required. Just like with programming: repeated code is a sign you needed a loop.

Overall look and feel: The BW principle states that the paper should be perfectly understandable when printed in black and white. Referees usually see things in black and white.

Finally: Anything and everything is easier and better if it is fun. Find ways to ensure you *enjoy* paper writing and you will dramatically speed your improvement.