# Outline for Research Papers

Below we present a concrete guideline for how to write a research paper in IS (Information Systems) or SE (Software Engineering). Depending on your specific sub-area or type of study this can be modified somewhat but still gives a good, basic structure. It is based on a paper by Lars Mathiassen et al, in submission.

## Logical elements of your paper

**A** (area of concern). Background to the general area.

**P** (problem). Problem from a practical perspective or gap in what

we know.

**F** (framework). What you need in order to analyze P.

 FA (framework related to A).

 FI (framework independent of A).

**RQ** (research question(s)). Based on A and the specific F you will use.

**M** (method). Formal research process to study P.

May/should affect your RQ as well.

**DC** (data collection). What types of data and data collection procedures

will be used?

**DA** (data analysis). How will the collected data be analyzed?

**C** (contribution). What type of contribution will/does the research make?

 CA (contribution to A)

 CF (contribution to F - may be to FA, FI or both)

 CM (contribution to M)

## What goes where in a research paper?

**Abstract**

Brief summary of the whole paper, often good to write a so called “structured abstract” [1]:

Background 1-2 sentence A

Objective & Method 1 sentence P and 1-2 sentences M&DC (possibly main in F)

Results 2-3 sentences DA

Conclusions 1-2 sentences conclusions and C

**1. Introduction**

1st paragraph: Starts with A and ends with P.

2nd paragraph: Starts with F to use and ends with RQ.

3rd paragraph: Starts with M (or research process in the CM project case),

describes how DC will be done, and

 ends with how DA will be done.

4th paragraph: Describes which type(s) of contributions the paper will

make, i.e. not the exact conclusions, but rather the specific type of contribution that is made. If this is a short paragraph only, add it to the end of the third paragraph instead.

5th paragraph: Outline of the paper (i.e. a text-based version of a table of

content that only outlines what your main sections – header level 1 - will bring up).

For larger studies and problems the 1st paragraph might need to expand to describe more of F before the P is described and then detailed in RQs.

**2. Theoretical framework / Background / Related work**

In this section you detail what is the current state of knowledge when it comes to A and P and which previous results you base your study and analysis on. This should clearly motivate RQ and give the F that you later analyze your results based on.

2.1 A + P In more depth than section 1, and ends with P. If P is very

long, make a separate section for it.

2.2 F Framework(s) (or each issue that feeds into the

framework)

2.x If you have several F, simply introduce each framework as a

new header level 2.

2.y If you build your own framework, present the framework that you have defined in a final separate section. Regardless if you use an existing framework or your own - always end this second section by explaining how your framework will help you address the P, and subsequently connect to and motivate what your RQ is.

**3. Method**

3.1 Research setting Information on where you are conducting your

study and the problem they have.

3.2 Research process What steps you are taking in the study?

3.3 Data collection What types of data, and what data collection

techniques are you using?

3.4 Data anlaysis What will you be using to analyze the data that you

collect? Think: What are the different categories of data that you are looking for? If you are doing keyword searches, this can be categories of related keywords. If you are doing interviews, this can be either themes that you identified before this interview or themes that emerge from the interviews. For other types of studies the statistical method used in the analysis can be described here.

3.5 Limitations What limitations are there to the research approach,

and what have you done to minimize the negative effects of these? This section is sometimes called “Validity Threats” in experimental studies.

**4. Results / Data**

Note: It is your DA that you include here, not your full collected data.

4.x Subsections named as you explained them in DA (3.4 above).

**5. Discussion**

Applies your F to your DA.

5.x Subsections named based on your F that you are using (from section 2).

5.y Always end your discussion with a summary that briefly

recaps how your F has allowed you to answer the RQ and how this RQ is a contribution to the P and A that it is related to.

**6. Conclusion**

Important: this section should NOT include anything that is new to the paper! It is just a summary of the study and the main conclusions. But all conclusions must already have been elaborated in section 5.

Also important: this section should be written more “stand-alone” than the rest of the paper, i.e. it should not assume the reader has read sections 1-5. This since many readers jump from reading the abstract to reading the conclusion before they decide to read the whole paper. For example, avoid direct references to the numbered research questions etc., rather restate the gist of them if necessary.

Example of common start: This paper set out to...<RQ>. Through <F>, we have found...<C>. These findings are important contributions to <A>. While we have specifically focused on <P>, the <relation between P and A> implies that our findings are likely to be of importance to <whomever else is also related to A>. In terms of future research, we particularly suggest

<possibly based on limitations in your current study, or complementing frameworks, RQs, etc>.

**7. Acknowledgements**

Thank those that made particular contributions or funded the study etc.

**8. References**

Only high-quality and complete references. Generally avoid web references or material that has not been through scientific peer review.

## References (for this report!)

[1] Budgen, D. and Kitchenham, B.A. and Charters, S.M. and Turner, M. and Brereton, P. and Linkman, S.G., “Presenting software engineering results using structured abstracts: a randomised experiment”, Journal of Empirical Software Engineering, vol. 13, no. 4, pp. 435-468, Springer, 2008.