TECHNISCHE UNIVERSITÄT DRESDEN

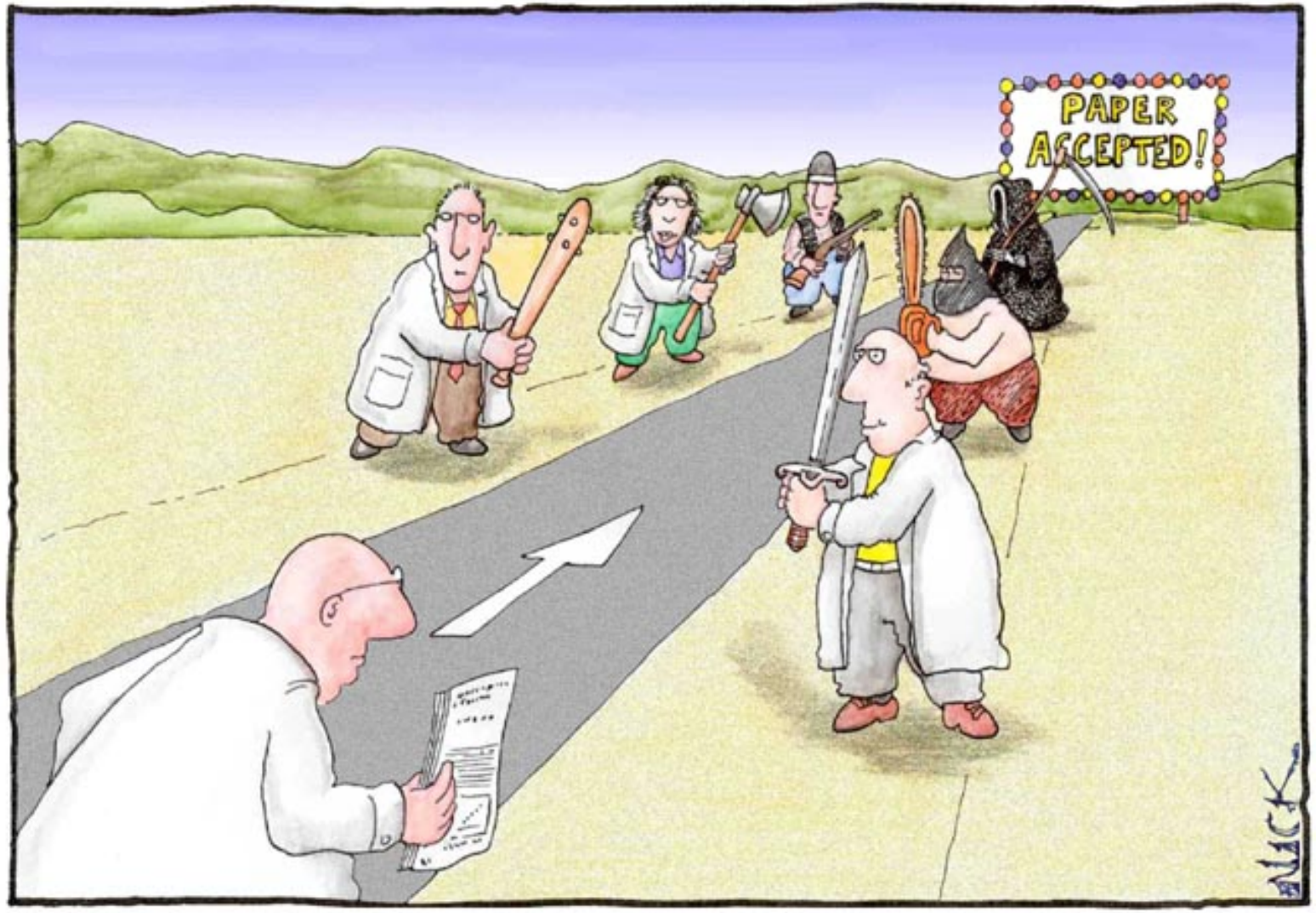
Department of Computer Science Institute for Software and Multimedia Technology, Software Technology Group

Software for Digital Health

Scientific Writing

Hauptseminar (WS 18)
Tuesday, 5. DS, APB/2101
Thomas Kühn (thomas.kuehn3@tu-dresden.de)

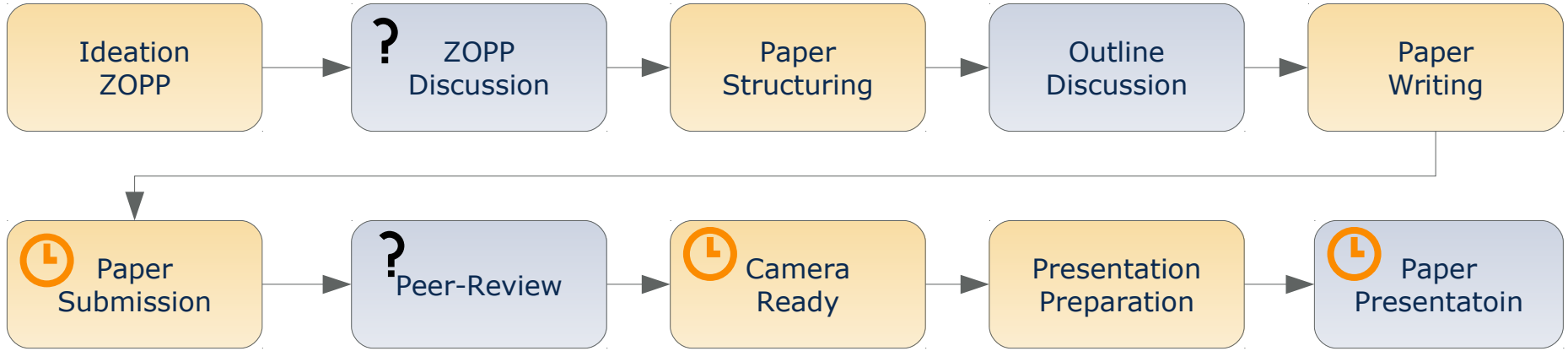
Doing Science is Hard?



Most scientists regarded the new streamlined peer-review process as "quite an improvement."

Cartoon by Nick D Kim strange-matter.net. Used by permission.

Organization
Overview



- *Give scientific presentations* (20 min + 10 min discussion)
  - Individual presentations **08.01.2019**
- *Write a research paper* (>=5 pages ACM Style)
  - Paper submission[1] **01.02.2019**

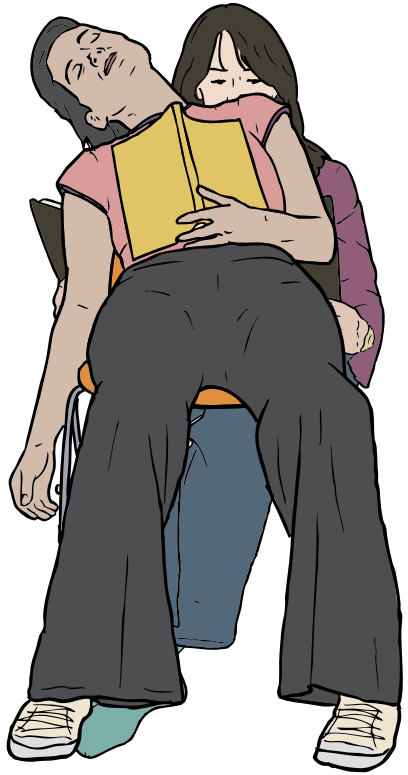
1) Per mail

- Pick a research topic within **software for digital health**
- Work alone
- Give a presentation on your topic
  - 30 minutes presentation
  - Practice giving presentations
  - Follow the advise provided throughout the next lecture
- Writing a research paper
  - >=5 pages two column paper
  - Preferably use LaTeX
  - Style: *ACM SIGPLAN Conference Format* (`acmart`)[1]
  - Apply the skills learned throughout this lecture

1) http://www.sigplan.org/Resources/Author/

Scientific Research
Daily Business

Reading



Writing

Organizing

Images from OpenClipart.org (Creative Commons by Steve Lambert)

Common Tasks

- Iterative process from idea to written paper
- Develop *"Ziel-orientierte Projektplan"* (ZOPP)
- Write an **abstract** early
- **Structure** your paper *(chapters, sections, paragraphs)*
- Write **outlines** for each chapter, section, ...

Mindmap with four arcs capturing the main idea of your paper

What *problems* does your paper address?

- Pick one main problem and add detailed subproblems

What are the corresponding *goals* of your paper?

- Align goals and subgoals to problems you address

What *solution* does your paper present?

- List your solution and subsolutions aligned to goals

What are *success criteria* or how to *evaluate* your solution?

- Specify *functional* and *non-functional* requirements
- *Evaluate* whether solution achieves goals

Paper Writing Process
Problem Definition

- Derived from a ZOPP
- State the *problem* in one or two sentences including your *goal*
- Highlight your *solution* in one or two sentences
- State your *success criteria* or how you *evaluate* your solution

"Currently, CROM models can be created textual or graphically without taking the well-formedness rules for these models into account. Hence, the goal of this work is to create a Eclipse-based plugin, which validates CROM models with respect to well-formedness. Additionally, the plugin should be easy to integrate into the existing editors."

– Kühn (2015)

Paper Writing Process
Abstract

- Includes the *problem definition* as crucial part
- Prepends a background of this research *(Why is it important?)*
- Summarizes the major problems and goals
- Appends a description of your evaluation *(success proof)*

Paper Writing Process
Example Abstract

"Modelling context-dependent domains is hard, as capturing multiple context-dependent concepts and constraints easily leads to inconsistent models or unintended restrictions. However, current semantic technologies not yet support reasoning on context-dependent domains. To remedy this, we introduced ConDL, a set of novel description logics tailored to reason on contextual knowledge, as well as JConHT, a dedicated reasoner for ConDL ontologies. ConDL enables reasoning on the consistency and satisfiability of context-dependent domain models, e.g., Compartment Role Object Models (CROM). We evaluate the suitability and efficiency of our approach by reasoning on a modelled banking application and measuring the performance on randomly generated models."

– Böhme et al. (2017)

Recurring structure of scientific papers in computer science

- Introduction / Motivation
- Background / Preliminaries / Contemporary Approaches
- Concept / Methodology
- Implementation / Realization
- Evaluation / Case Study / Illustration / Discussion
- Related work
- Conclusion */ Contributions*

Paper Writing Process
How it feels like

You

Your Writing Tool

Knowledge in your Head

Writing Result

Images from https://parske-shop.de/Gartentechnik/Haecksler/Erco-Holzhaecksler-GHX-CH1900-Zapfwelle-oder-Benzin.html

Paper Writing Process
How it should be

Image from MaxPixel.net (Creative Commons by Markus Baumeler)

Common Tasks

- Structuring paper
  - Define chapters, sections, subsections
  - Summarize outline of each part
- Write individual paragraphs
  - Include individual artifact *images, tables, listings*
  - Outline points become individual paragraphs
  - Write a structured paragraph
  - Finalize paragraphs and transitions

- Employ *recurring structure* of scientific papers in computer science
- Define the structure by means of **headings** for

  parts, chapters, sections, subsections, ...
- Write short outlines/summaries for each heading
  - Use bullet points and short statements outlining the intended content
  - Which questions are answered?
  - What arguments are provided?
  - What solutions/conclusions are described?

Outlines

- *Not* written for the *reader*, but for **you**
- Summarizes intended content of *chapter*, *section*, *subsection*, and …
  - What concepts/ideas must be introduced/discusses?
  - Which parts in your text cover which parts of your ZOPP?
  - What questions should be raised and answered?
- Helps to focus writing and avoiding running off the topic
- Useful to track writing progress

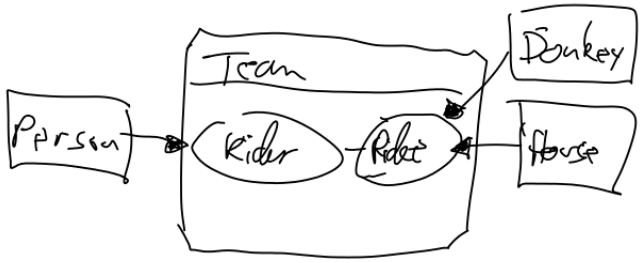
Outline Example

2.3. Graphical Editor Frameworks

- *Give a short overview on typical frameworks for the development of graphical editors*
  - *Provide a clear description of the typical aspects that need to be implemented in such a framework, e.g. Language Concern (Metamodel), Editor Concerns, Edit Policies.*
  - *Describe GEF, GMF, Graphiti, Sirius, and EuGENia*

Include Artifacts

Images



- Start with sketches, low resolution images
- Later create scalable vector images *(time consuming)*

Tables

Feature	Ledwick 2000	Generic Role Model 2002	ORM 2 2005	E-CARGO Model 2006	Metamodel for Roles 2007	INM 2009	DCI (in Scala) 2010	Object-UML 2012	Helena Approach 2014	Chameleon 2003	OT/J 2005	Rava 2006	powerJava 2006	Rumer 2007	NestEJ 2009	JavaSage 2012
16	■	□	■	□	■	□	□	■	□	□	□	□	□	■	□	□
17	□	□	■	□	■	□	□	■	□	□	□	□	□	□	□	□
18	□	□	□	⊞	□	□	□	□	□	□	■	□	□	⊞	⊞	□
19	⊞	□	⊞	■	■	■	■	⊞	■	□	■	□	⊞	⊞	■	□
20	□	□	□	□	■	■	■	■	■	□	■	□	■	■	■	□
21	⊞	□	□	■	■	⊞	□	■	⊞	□	□	□	■	□	■	□
22	□	□	■	□	■	■	□	□	□	□	■	□	□	■	□	□
23	□	□	□	□	■	□	□	□	□	□	■	□	□	□	□	□
24	□	□	□	□	□	■	□	□	□	□	■	□	⊞	■	■	□
25	□	□	□	□	■	■	□	□	□	□	■	□	□	□	□	□
26	□	□	■	■	⊞	■	■	□	■	□	■	□	⊞	■	■	□
	Modeling Languages									Programming Languages						

■: yes, ⊞: possible, □: no, ∅: not applicable

- Use generator for latex tables
- Refine/optimize table later

Listings

```
1 Start Inheritance (Role_Inheritance) when
2   IsSourceType(RoleType);
3 Add Inheritance (Role_Inheritance) when
4   IsSourceType(RoleType) and IsTargetType(RoleType) and
5   SourceEqualsTargetType();
6 Create Inheritance (Role_Inheritance) when true;
```

- Start with source code snippets
- Later remove all unnecessary statements

Structured Paragraph

- Write paragraph for each major point of the outline
- Typical structure of paragraphs
  - **Thesis question**
  - **Thesis statement** topic, purpose or development scheme
  - Supporting/opposing **arguments**, claims, evidence or warrants
  - **Thesis conclusion** and transition
- Enumerate arguments

Structured Paragraph Example

2.3. Graphical Editor Frameworks

There exists several graphical editor frameworks for all platforms.

As our prototypical GEPL is based on Eclipse, we focus on corresponding frameworks.

1) *The* Graphical Editing Framework *(GEF) is the basis for most other frameworks, as it facilitates means to implement rich graphical Java applications.*
2) *The* Graphical Modeling Framework *(GMF) is a model-driven editor generator, where the various concerns are specified in interrelated models, e.g., the domain model, the graphical definition, and the tooling definition.*
3) EuGENia *and* Sirius *are both frameworks for textual respectively visual specification of GMF editors.*
4) Graphiti *utilizes EMF models to provide a uniform pictogram model linked to a custom domain model, whereas visualizations, behaviors, and edit policies must be manually implemented in IPattern and IFeature classes..*

None of them natively supports the modular definition of language features.

Finalize Paragraphs and Transitions

- Remove enumeration from structured paragraph
- Improve **wording**
- Link thesis question, thesis statement, and arguments with **transitions**
  - Additions: *Moreover, furthermore, especially, in detail, …*
  - Cause & Effect: *Thus, accordingly, as a result, consequently, hence, …*
  - *(More in the appendix)*
- Introduce **transition** and **pivot** sentences
- Add **controlling idea** to thesis conclusion

Finalized Paragraph Example

2.3. Graphical Editor Frameworks

*There exists a **plethora** of graphical editor frameworks for all platforms, **yet** as our prototypical GEPL targets Eclipse, we focus on **associated** frameworks.*

***In general**, the* Graphical Editing Framework *(GEF) is the basis for most other frameworks, as it facilitates means to implement rich graphical Java applications.*

On top of GEF, there exists both model-driven and model-based frameworks.

***For the former**, the* Graphical Modeling Framework *(GMF) is a model-driven editor generator, where the various concerns are specified in interrelated models, e.g., the domain model, the graphical definition, and the tooling definition.*

***Moreover**,* EuGENia *[@kolovos2017eugenia] and* Sirius *[@viyovic2014sirius] are both frameworks for textual respectively visual specification of GMF editors.*

***By contrast**,* Graphiti *utilizes EMF models to provide a uniform pictogram model linked to a custom domain model, whereas visualizations, behaviors, and edit policies must be manually implemented in IPattern and IFeature classes.*

***Although these frameworks significantly simplify the design of graphical editors,** none of them natively supports the modular definition of language features.*

Structured Writing
Typical Transitions

Cause and Effect

- accordingly
- as a result
- consequently
- hence
- it follows, then
- since
- so
- then
- therefore
- thus

Conclusion

- as a result
- consequently
- hence
- in conclusion, then
- in short
- in sum, then
- it follows, then
- so
- the upshot of all this is that
- therefore
- thus
- to sum up
- to summarize

Structured Writing
Typical Transitions

Comparison
- along the same lines
- in the same way
- likewise
- similarly

Addition
- also
- and
- besides furthermore
- in addition
- in fact
- indeed
- moreover
- so too

Contrast
- although
- but
- by contrast
- conversely
- despite the fact that
- even though
- however
- in contrast
- nevertheless
- nonetheless
- on the contrary
- on the other hand
- regardless
- whereas
- while
- yet

Structured Writing
Typical Transitions

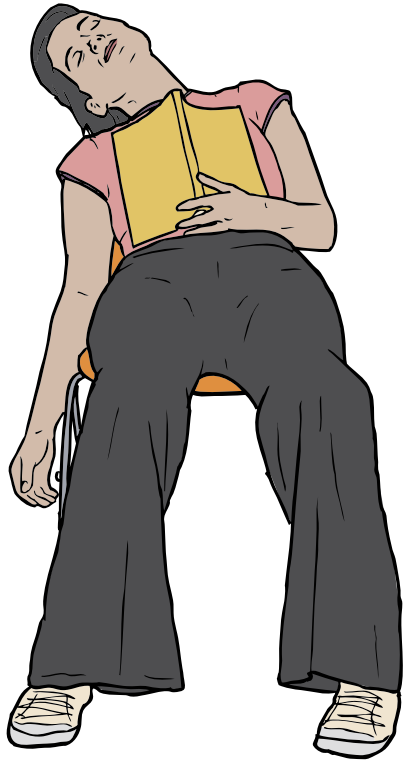
Concession
- admittedly
- although it is true that
- granted
- I concede that
- of course
- naturally
- to be sure

Example
- after all
- as an illustration
- consider
- for example
- for instance
- specifically
- to take a case in point

Elaboration
- actually
- by extension
- in short
- that is
- in other words
- to put it another way
- to put it bluntly
- to put it succinctly
- ultimately

Scientific Research
Daily Business

Reading



Writing

Organizing

Images from OpenClipart.org (Creative Commons by Steve Lambert)

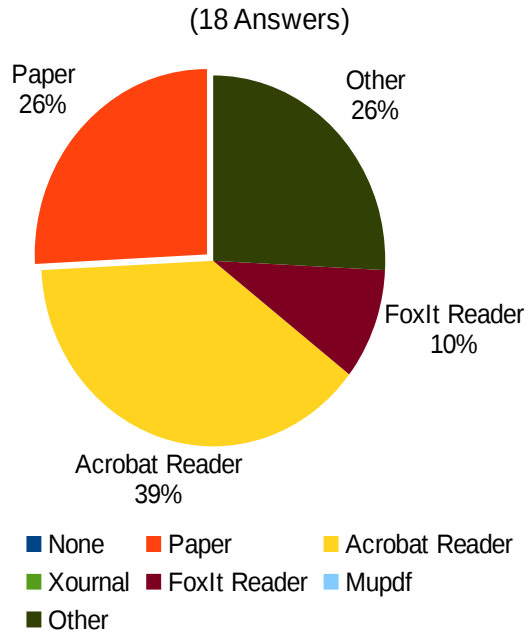
Common Tasks

- Find relevant / related publications
  - Query scientific search engines
  - Look up *BibTex* for specific publications from the web
- Investigate found publications
  - Skim papers
  - Make notes and hints
  - Organize downloaded files
  - Maintain a corresponding **bibliography** of *BibTex* entries

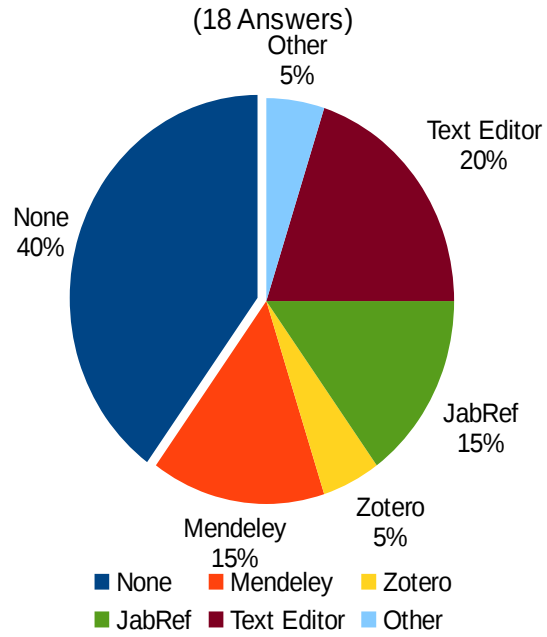
A Small Survey

- Q1:What tools do you use to read and annotate papers?
- Q2:*What tools do you use to organize your bibliography?*
- Q3:*What tools do you use to organize stored papers?*

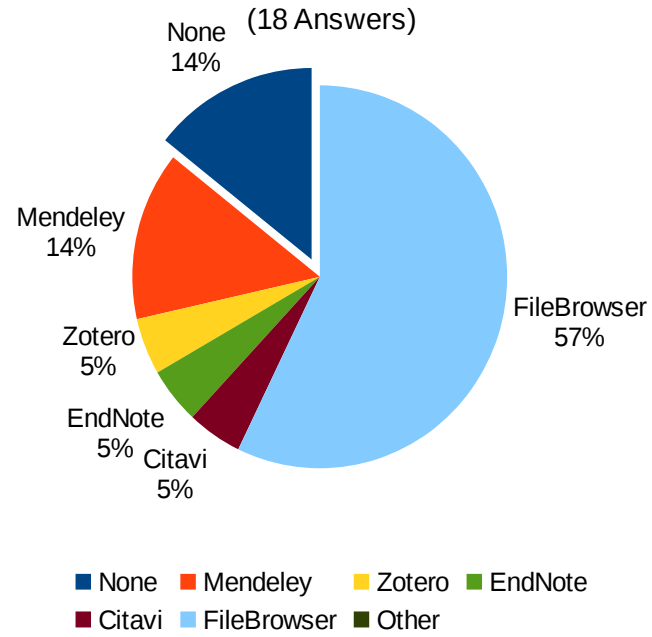
What tools do you use to read and annotate papers?



What tools do you use to organize your bibliography?



What tools do you use to organize stored papers?



BibTex

```
@inproceedings{kuehn2015choosy,
  title = {Choosy and picky: configuration of language product lines},
  author = {K{\"u}hn, Thomas and Cazzola, Walter and Olivares, Diego Mathias},
  booktitle = {Proceedings of the 19th International Conference on Software
                   Product Line},
  year = {2015},
  organization = {ACM},
  pages = {71--80},
  citations = {1},
  file = {:./Kuehn/Thomas Kuehn_Choosy and picky - configuration of language
           product lines.pdf:PDF},
  howpublished = {\url{http://dl.acm.org/citation.cfm?id=2791092}},
  owner = {thomas},
  timestamp = {2015.09.07}
}
```


1) http://www.bibtex.org/

Finding Relevant / Related Publications

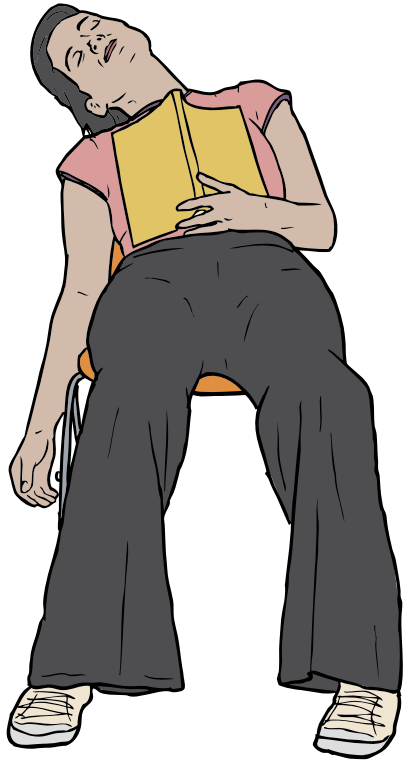
- Query scientific search engines
  - When looking for complex search terms
  - Google Scholar (free) https://scholar.google.com
  - Elsevier Scopus (registration) https://www.scopus.com
  - Academia (registration) https://www.academia.edu
  - Sci-Hub (illegal) http://sci-hub. …
- Query publishers directly
  - For a specific journal or conference in computer science
  - IEEE Xplore https://ieeexplore.ieee.org
  - ACM Digital Library https://dl.acm.org
  - Springer Link https://link.springer.com
  - Elsevier ScienceDirect https://www.sciencedirect.com

Investigating Found Publications

- Use appropriate reader
  Permit highlighting, comments, and annotations
  - *Xournal, Acrobat Reader, Foxit Reader, Mupdf, …*
- Use tool to manage your bibliography
  Organize, search, and annotate your BibTex entries
  - *JabRef, BibDesk, EndNote, …*
- Use one tool for both
  Manage, search, and comment both PDF documents and BibTex entry
  - Mendeley (freemium) https://www.mendeley.com
  - Zotero (freemium) https://www.zotero.org
  - Citavi (freemium) https://www.citavi.com

Scientific Research
Daily Business

Reading



Writing

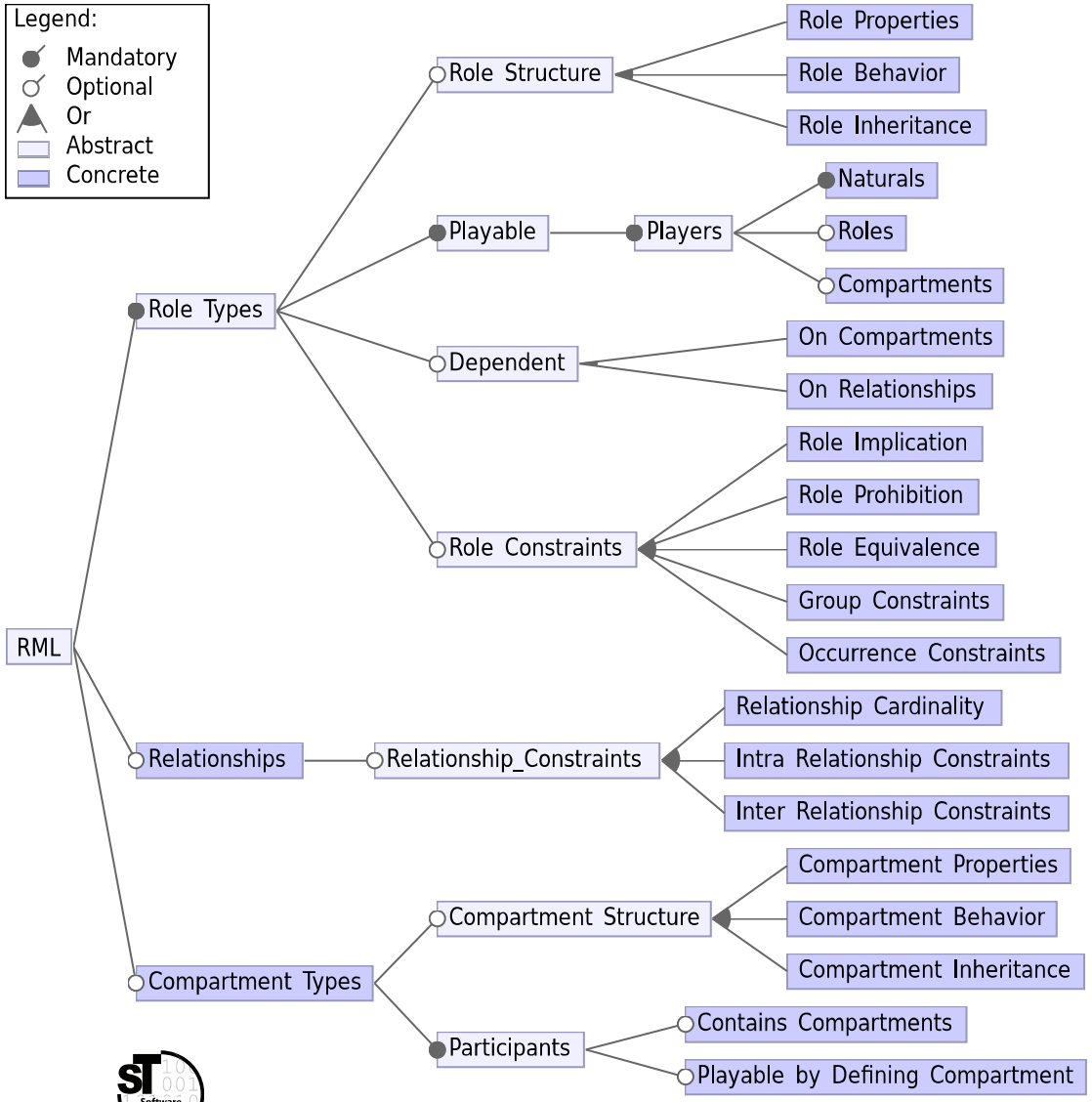
Organizing

Images from OpenClipart.org (Creative Commons by Steve Lambert)

Common Tasks

- Create a classification scheme
  - Identify classifying criteria
  - Set up list, tree, or map of terms, features, requirements, or classes
- Classify papers and approaches
  - Indicate found criteria in papers
  - Maintain classification for each paper or approach
  - Produce diagrams for comparison *tables*, *bubble charts*, or *kiviat graphs*

Classifying Research
Classification Scheme



For Papers

- Taxonomy of terms
- General classification of research papers by Shaw
- Orthogonal dimensions of classes

For Approaches

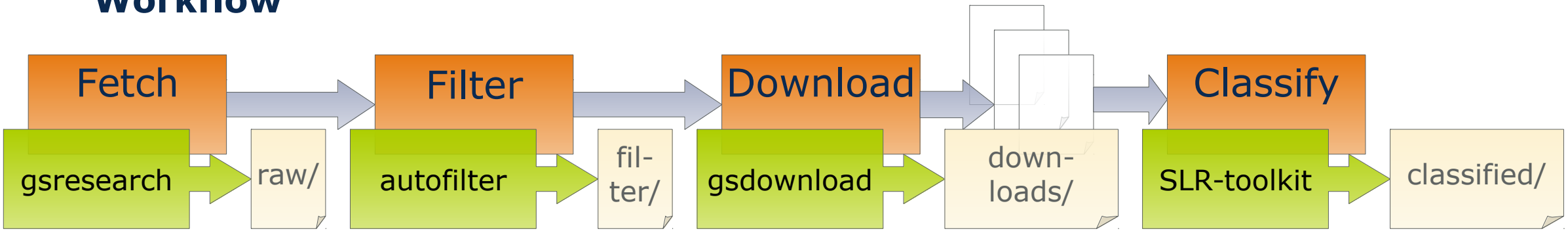
- List of (non-)functional requirements
- List of qualitative and quantitative properties
- Feature model consisting of features and dependencies

Classifying Research
Mining a Classification Scheme

- Existence of general classification schemata, *e.g.,*
  Shaw's classification of research [Shaw2002]
- Utilize existing classifications from related **surveys** or **PhD theses**, *e.g.,*
  Feature model for language workbenches [Erdweg et al.2015]
- Creating new classification scheme
  - Start from existing schemata; extend missing dimension
  - Retrieve requirements, goals, or features from publications

Never use made up classification schemata

Workflow

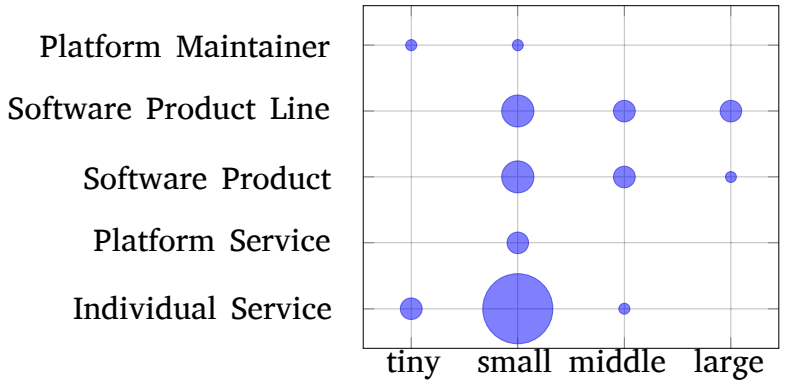


- After selecting relevant papers or approaches
- Investigate each paper annotate mentioned requirements and features
- Use tool support to track annotations for each paper or approach, *e.g.*,

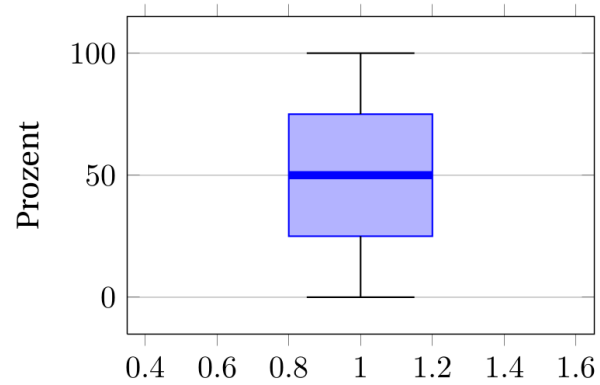
  SLR-Toolkit[1] *uses BibTex annotation and supports arbitrary hierarchical classification schemes*

1) https://github.com/sebastiangoetz/slr-toolkit

Classifying Research
Showing Comparison Results



Bubble Chart



Box Plot

Qualitative Evaluation

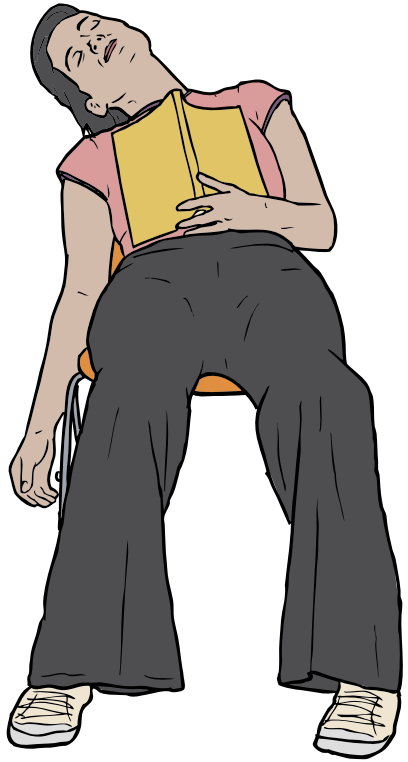
- Comparison tables
  Terms, Icons (○ ◐ ● ◌), …
- Diagrams for detailed comparison
  (2D) *Pie charts, Histograms, …*
  (3D) *Bubble charts, 3D Plots, …*
  (nD) *Kiviatgraphs, Parallel Hierarchies, ...*

Quantitative Evaluation

- Tables for basic analysis
  Standard deviation (+/-), Mean, ...
- Plots for more complex analyses
  (2D) *Plots, Box plots, Line chart, ...*
  (3D) *Heat Maps, 3D Plots, ...*
  (nD) *Parallel Koordinates, ...*

Scientific Research
Daily Business

Reading



Writing

Organizing

Images from OpenClipart.org (Creative Commons by Steve Lambert)

Common Tasks

- Deploy the given **template**
- Read and understand **author guidelines**
- Add all **meta information** (*title, authors, affiliations, ...)*
- Include your **abstract**
- **Structure** your document into subdocuments per chapter/section
- Include images, listings, and tables in **floating environment**

Practical Paper Writing
Preparing your Submission

- Create your Camera Ready version with LaTeX[1]
  - Correct use the `acmart` template
    `\documentclass[sigconf,~~review~~]{acmart}`
  - Add the copyright information
    ```
    \setcopyright{rightsretained}
    \acmConference[ASiCS'18]{Academic Skills in Computer Science}{July 20}{Dresden, Germany}
    \acmYear{2018}
    ```
  - Add the correct acknowledgements of your funder
- Check the *ACM SIGPLAN Conference Format*
  - Incorporate hints from the ACM website[2]
  - Follow the author guidelines, again

1) www.latex-project.org
2) http://www.sigplan.org/Resources/Author/

Comply to the Paper Geometry

- Fix **all** `Overfull` box errors[3]
  - Use a ruler or
  - `\usepackage[color=red, width=3pt, height=0.5\baselineskip]{overcolored}`
- Fix *some* `Underfull` box errors
- Check the boundaries of all images
- Set the correct geometry[4]
  - A4 → `a4paper`
  - US Letter → `letterpaper`
  - US Legal → `legalpaper`
  - …

```
\documentclass[a4paper]{acmart}
\usepackage[a4paper]{geometry}
```

3) https://github.com/raphink/overcolored
4) https://ctan.org/pkg/geometry

Check Spelling and Grammar

- Check for spelling and grammar mistakes
  - Use a spell checker,[5]
  - Revise your own text, and
  - Employ a proofreader
- Check punctuation
  - Comma before accessory sentence with *because, which, …*
  - No comma before *that*
  - Comma after *therefore, nonetheless, in conclusion, …*
  - Use the oxford comma[6]

5) https://www.youtube.com/watch?v=8Gv0H-vPoDc
6) http://richardgilbert.me/we-like-the-oxford-comma-ii

Chicago Manual of Style (CMS) title capitalization rules[7,8]

1. Capitalize first and last words in titles and subtitles (but cf. 7), and capitalize all other major words *(nouns, pronouns, verbs, adjectives, adverbs, and some conjunctions—but see rule 4)*.
2. Lowercase the articles *the, a, and an*.
3. Lowercase prepositions, regardless of length, except when used adverbially or adjectivally *(up in Look Up, down in Turn Down, on in The On Button, to in Come To, etc.)* or when they compose part of a Latin expression used adjectivally or adverbially *(De Facto, In Vitro, etc.)*.
4. Lowercase common coordinating conjunctions *and, but, for, or, and nor*.
5. Lowercase not only as a preposition (rule 3), but also as part of an infinitive *(to Run, to Hide, etc.)*, and lowercase as in any grammatical function.
6. Lowercase the part of a proper name that would be lowercased in text, such as *de* or *von*.
7. Lowercase the second part of a species name, such as fulvescens in Acipenser fulvescens, even if it is the last word in a title or subtitle.

7) http://www.chicagomanualofstyle.org/book/ed17/part2/ch08/psec159.html
8) http://www.chicagomanualofstyle.org/book/ed17/part2/ch08/psec161.html

Check Misused Words and False Friends

Usual	Other
alternative	alternate
comparable	comparative
complement	compliment
dependent	dependant
descendant	descendent
discrete	discreet
emit	omit
ensure	insure
ensure	assure
excerpt	exert

Usual	Other
foregoing	forgoing
further	farther
elusive	illusive
manyfold	manifold
omit	emit
partly	partially
principle	principal
simple	simplistic
solvable	soluble
stationary	stationery

Check Typical Mistakes

Right	Wrong	Right	Wrong
adaptation	adaption	miniature	minature
apparent	apparant	occasional	occaisional
argument	arguement	occurred	occured
comparison	comparision	participate	particepate
consistent	consistant	primitive	primative
definite	definate	propagate	propogate
existence	existance	pronunciation	pronounciation
foreign	foriegn	pseudo	psuedo
grammar	grammer	referred	refered
heterogeneous	heterogenous	repository	repository
homogeneous	homogenous	separate	seperate
independent	independant	supersede	supercede
insoluble	insolvable	transparent	transparant

Check Citations

- Include a non-breaking space before each citation

  ```
  modeling languages~\cite{authorE}
  ```

- Citations at end of a sentence

  “conducted, as well. [5]” → “conducted [5], as well.“

  “modeling languages. [27]” → “modeling languages [27].“

- Correctly list citations

  “[1], [2], [3], [6]” should be “[1–2,6]“

  ```
  \cite{authorA,authorB,authorC,authorF}
  ```

Check Footnotes

- Fix footnotes occurring before punctuations

  “family of RMLs[3].“ → “family of RMLs.[3]“

  “feature[12],“ → “feature,[12]“

  “major drawback [123]“ → “major drawback[1,2,3]“

  ```
  \usepackage[multiple]{footmisc}
  ```

- Footnotes with links utilize `\url{https://…}`

  ```
  \usepackage{url}
  ```

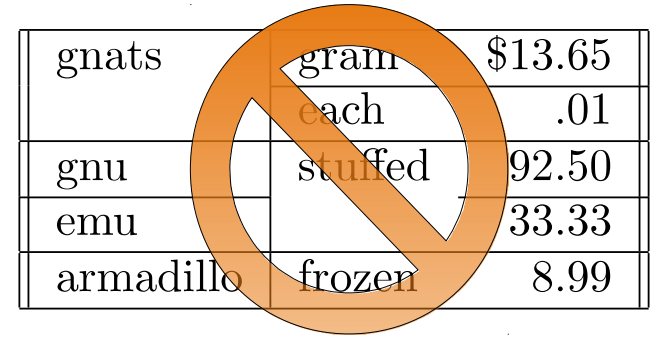
- Footnotes in paper title and authors with `\titlenote` and `\authornote`

Check Figure, Tables, and Listings Placement

- *(Almost)* all figures, listings, and tables
  - Floating environment
  - Align top or all on page
- ACM guidelines for captions
  - For figures **below**
  - For tables **above**
  - For listings **above**
- Guidelines for tables
  - Use booktabs for tables

    ```
    \usepackage{booktabs}
    ```
  - **Only** use horizontal lines, **never** vertical lines

    ```
    \toprule  \midrule  \bottomrule
    ```
  - **Never** use double line



Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce nec pharetra justo. Nam in augue quis dolor

bibendum gravida eget elit.

Lorem ipsum dolor sit amet,

consectetur adipiscing elit. Fusce nec pharetra justo. Nam in augue quis dolor bibendum gravida eget elit.

Fusce vestibulum finibus nisi id fermentum. Nullam cursus urna enim, ac feugiat nunc sollicitudin vel integer feugiat tincidunt.

Widow

- Last sentence of paragraph on new page

Orphan

- First sentence of paragraph on previous page

Orphan (alternative)

- Last word of paragraph alone on last line

9) https://en.wikipedia.org/wiki/Widows_and_orphans

Check Bibliography

- Add missing information to bibliography
  - Every paper needs at least
    - Title, authors, year, publisher, page numbers
    - Conference or journal
  - Optional, but useful: Digital Object Identifier (DOI)
  - Optional, but waste of space: Uniform Resource Locator (URL)
    (Only in case of *websites, white papers, thesis,* and *videos)*
- Balance the last page of references (in case of two-column styles)
  `\usepackage{balance}`

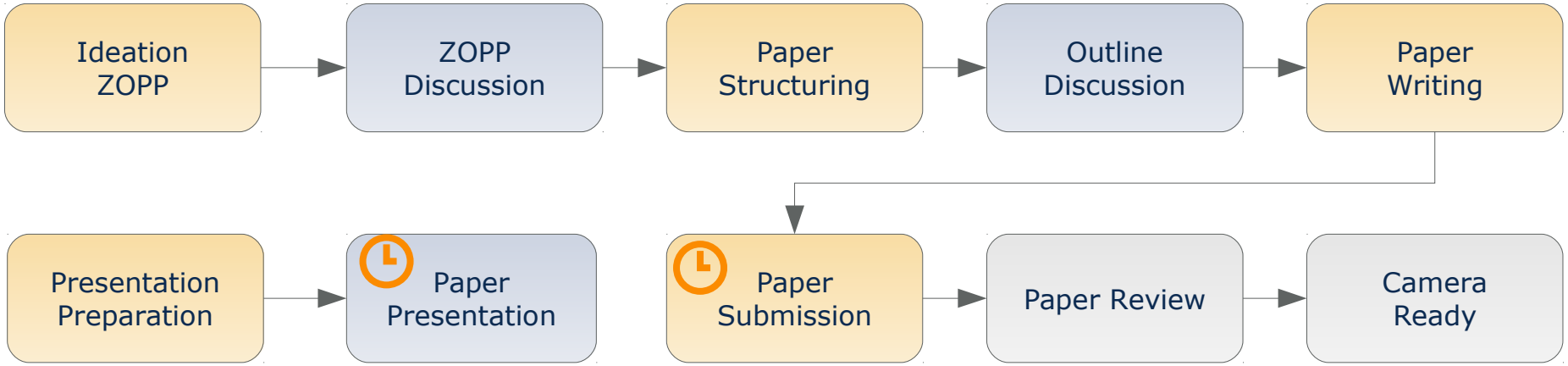
Wrapping Everything Up

- Prepare the PDF to be uploaded
- Create archive with the camera ready source code (usually `*.zip`)
  - Create a separate folder
  - Include all used *latex source* (`*.tex`)
    - Remove spacing trickery (`\\[-0.5em]`)
    - Remove all comments
  - Include all used code *snippets, images,* and *external artifacts*
  - Include the (*.bbl) generated bibliography file *(instead of the full bibliography (`*.bib`))*
- *Commit* and *tag* the version of your camera ready sources

Submission

- Upload PDF document and archive
- Sometimes document is checked for spacing errors

Organization
Overview



- *Give scientific presentations* (20 min + 10 min discussion)
  - Individual presentations **08.01.2019**
- *Write a research paper* (>=5 pages ACM Style)
  - Paper submission[1] **01.02.2019**

1) Per mail

Software for Digital Health
Scientific Writing