

Problem Statement Checklist

Spencer Smith

September 23, 2020

- Uses the provided `tex` file as a base
 - ☐ Includes the input for `Comments`
 - ☐ Includes a title
 - ☐ Revision history included for major revisions
- Follows writing checklist (full checklist provided in a separate document)
 - ☐ `LATEX` points
 - ☐ Structure
 - ☐ Spelling, grammar, attention to detail
 - ☐ Avoid low information content phrases
 - ☐ Writing style
 - ☐ Use the provided `.gitignore` file
 - ☐ Hard-wrap at 80 characters per line
- Overall qualities of documentation
 - ☐ Scope is clear
 - ☐ Document is abstract (for instance, it does not mention the programming language (unless this is a true constraint) or the numerical algorithm choice (unless the algorithm is the “what” rather than the “how”))
 - ☐ Output of software is unambiguous

- ☐ Inputs to software are unambiguous
- ☐ Problem statement points to Drasil, if this is to be a project implemented with Drasil
- ☐ Document is simple (for instance, there is no need for a table of contents)
- Problem statement review issue is created correctly
 - ☐ Issue includes a link the file to be reviewed, or the commit hash where the latest version is found
 - ☐ Issue is assigned to `smiths`
 - ☐ Your user info on GitHub includes your name, since some GitHub ids do not clearly show which student in the class is the owner of the id.