Software Development Life Cycle

# Analysis

Look at problem or current implemented system. Does it need updating/improving or creating from scratch?

Will it need hardware?

What problems might arise?

How can these be mitigated?

# Planning / Design

What will we do? What can we learn from the existing system? How will the final solution work and interact for any users? Will there be human users? How can it be fixed by the user if something goes wrong? What (if any) hardware will be needed? What MUST it do?

* Detailed success criteria.
* Make Pseudo code and flow charts.
* Screen layout sketches to understand user requirements

# Development

Actually making the software – keeping track of the progress and change log. GitHub? Lesson diary with evidence. Refer back to planning to ensure all required tasks are done. Annotate all characters.

# Testing

Does it work? Make sure everything will work. Possible check with clients. Inputting invalid data. Error checking. Data validation. Full documentation of any errors.

# Evaluation

Does everything work? What problems arose? Did it work? How can we apply learning to future problems? What would you do differently? What can I develop further? What went well? Any other comments.