

Intro

This is to help software developers write better software and have better habits when working.

This was inspired by The Checklist Manifesto, which is a book by Atul Gawande that highlights the power of checklists in improving performance and reducing errors. Gawande argues that even experts can benefit from using checklists to ensure consistency and enhance teamwork. He provides examples from various industries, such as medicine and aviation, to demonstrate how checklists can save lives and improve outcomes. The book emphasizes the importance of simplicity, standardization, and communication in checklist design and implementation.

Overall, it advocates for the systematic use of checklists to enhance productivity and safety in complex tasks.

However, you as the developer are responsible for knowing when its time to creatively ignore the checklist and do things differently.

Checklists

- * Adding a checklist - Use this when adding a checklist to this repo
- * Making a checklist - Use this when making a new checklist
- * git
- * git commit - Use this when making a commit

Title: Adding a checklist

Use: Use this when adding a checklist to this repo

Type: read-do

Date: 06/22/2023

Checklist

☐ make a fork

☐ make a checklist

☐ run updateReadables.py

☐ open a pr

Notes:

Title: Making a checklist

Use: Use this when making a new checklist

Type: do-confirm

Date: 06/22/2023

Checklist

- ☐ Determine a title for the checklist
- ☐ Create the header title, use, type, date
- ☐ Make a list of checkboxes for each step
- ☐ Add notes for more complex info
- ☐ Validate format with "Making a checklist"

Notes:

- * Title should state what the checklist is for
- * Use is the trigger item that causes you to use the checklist
- * No more then 8 short to-do check boxes
- * The date is last update time in mm/dd/yyyy format
- * The type can be either do-confirm or read-do

git

Title: git commit

Use: Use this when making a commit

Type: read-do

Date: 06/22/2023

Checklist

- ☐ Validate branch
- ☐ Check status
- ☐ Add related files
- ☐ git commit
- ☐ provide tag
- ☐ state your message
- ☐ state your discription
- ☐ write and exit

Notes

- * a tag is the start of the commit message and looks like [bug]
- * all files added should relate to a specific task
- * the commit summary should be short and simply state what was changed in 50 chars
- * the note should go into detail about why these changes were made