Work Process

1. Philosophy

- the goal of work process is to do the right thing at the right time.
- highly subjective, depends on context, business goals, personal goals, feelings, values, ect...
- we already have an engine for processing this data. It's called our lower brain. Fast and efficient, and subconscious. Input is everything, output is feelings.
- my strategy: normalize, filter and condense all sources of input. Reduce noise. Absorb and decide repeatedly "what is the right thing to do right now?" Every day a new list

2. Command line

- fewer distractions
- more information density
- more consistency
- more simplicity
- some exceptions, systems that are already information rich and dense: eg email and calendar.

- mosh + tmux
  - portable, stable
  - low bandwidth requirements
  - high level "tasks" in windows
  - task state in panes

3. Emacs Org Mode

- dashboard with week view
- global sub lists by tag
  - Promise, a thing I said I will do
  - Urgent, someone is waiting on affected imminently
  - Important
  - Remote (new) - external source requiring attention (currently github todos)
- quick capture todo and notes
- links to source-of-truth
- scheduling and deadlines
- priorities to highlight chosen tasks
- de schedule regularly, find work by tags
  - PROMISE, URGENT, IMPORTANT, REMOTE
  - read, <names>, <topics>, etc...
- same system at different jobs and home
  - ~/org folder in git repo or Dropbox
  - beorg iphone app for home access
- query for perf ü

4. Work spaces

- tmux windows for each work "task"
- matches a folder in ~/work
  - golang workspace or single repo
- matches branch names (when possible)
- everything is exactly as I left it when returning to a task
  - panes
  - command history
  - file system state

  (ps af shows a lot of gopls!)

- programming in emacs as well
  - gopls for completion / highlighting / nav

5. Wrapping up

- timer in tmux
- reconnecting / ephemeral machine
- end of day script
  - commit org changes and push
  - check for resource usage