Mini-SWE Rules

August 24, 2020

Prelims

** **Reasonable Person Std:**
- o-Submit what you've got on time
- o-Ask for help early

** **Smart Person Std** == Always be ready to show your boss visible progress.

(*)(*) **S/W's Most Important Thing:** Maintain Your **Morale**
- o- The Next Hour; The Next 5 minutes

Rule #0 (Fast): Get to working S/W fast
- o- **Brute Force Simple:** no frills, use what you know well, go ugly early (and clean once it works)

Rule #1 (Optim): Never Pre-optimize
- o- Don't do things for the future; wait till things are required to take the next small step

Rule #2 (Hunts): Kill Bug Hunts
- o- 90% of typical dev effort is in run-time (RT) bugs
- o- Make RT bugs look like CT bugs (< 5 mins to find/fix)
- o- Do Add-a-Trick – always make Trick's results visible on screen (GUI or console log)

Rule #3 (Story): The 2-Story Story
- o- Top-Story is written in the User's Problem Domain Language (ULang)
- o- Bot-Story is Comp-Sci stuff

How? Via CRC Sim
- o1. **User-Scenario**: 1 user role; rambling paragraphs on what user does (tasks) in that role
- o2. **Use-Case**: one role-task: ID agents (user-in-role & others), actions, interaction messages
- o3. Write **CRC Card** per Agent: Class=agent; Responsibilities=actions, Collaborators=msg-buddies
- o4. **Simulate CRC** Architecture: one card per person, sim the task kickoff, interactions, & results

Rule #4 (EIO): Use Sample EIO – BEFORE you code
- o- Write Expected Input-Output pairs to drive your detailed designs; based on CRC Sim architecture

Rule #5 (Half-Day): Publish (Half-Day) Goal for your planned day's effort
- o- gives you 100% overrun leeway
- o- gives you massive estimation practice

Rule #6 (Clean): Clean the Page – BEFORE you let your code become visible to others
- o- Don't get a reputation for turning in poor quality code
- o- Avoid **Lehman's Mummy Laws** of S/W Development Life Cycle (SDLC) atrophy