

The Pythoness Programmer

Grit Framework Workflow Worksheet

2026 Fire Horse Edition | Digital Spring Cleaning

For people who need systems that work with real brains, real lives, and real energy shifts.

How to Use This Worksheet

This worksheet is the culmination of the April Digital Spring Cleaning arc — four newsletters, one live session, and a single place to work it all through. You do not need to have read every issue. Start where you are.

- Fill all four sections for a full reset.
- Or pick one section for a focused 15-minute tune-up.
- Use NAA (Notice → Adjust → Acknowledge) anywhere you get stuck.

This worksheet is designed for reflection and redesign, not perfection.

*Adapted from Angela Duckworth's Grit: The Power of Passion and Perseverance (Scribner, 2016).
Independent adaptation; not affiliated with or endorsed by the author.*

Notice the Friction

Quick Workflow Snapshot

What are you trying to accomplish this season?

Which 3 digital tools or spaces are most critical right now?

1.

2.

3.

What currently feels **clear**?

What currently feels **chaotic**?

Workflow Friction Rating

Pick 5 recurring workflows. Rate each on friction from 1 (smooth) to 5 (always breaking). Circle your two highest scores — those are your starting points.

Workflow	Friction Score (1–5)	Where it collapses

For each of your two highest-friction workflows, what is one tiny Adjust action you could take in under 10 minutes?

1. _____
2. _____

Friction Map

Where are you leaking energy every day? List your top friction points in each zone.

Zone	What keeps breaking?	How it affects you
Files and folders		
Inbox and messages		
Notifications and interruptions		
Task capture and planning		
Tool switching		

Grit Framework Check-In

Does this setup still have what it needs?

Passion — Does this setup support what I actually care about?

Perseverance — What is one thing I can sustain this week, even on a low-energy day?

Growth — What did this month teach me about how I work?

Resilience — What support keeps me from starting over next month?

NAA Micro-Reflection (2 minutes)

Notice: What feels most draining in your digital space right now?

Adjust: What is one tiny change you can make in under 5 minutes?

Acknowledge: Write one sentence giving yourself credit for the shift.

Simplify Your Structure

Your file system should reduce decisions, not create them.

The 3-Folder Rule

Your three top-level containers. Every active thing in your life has a clear home — and you can answer one question quickly when you save a file: do I need this right now, soon, or mostly as reference?

Folder	Name	What lives here
Now		Active work this week
Soon		Queued work for this month or season
Reference		Completed or evergreen materials — keep, rarely touch

Inside each folder, you can add subfolders as needed. The top level stays stable.

If your system makes you pause and overthink every save, it is too expensive for real life.

One Naming Rule That Changes Everything

Pick one naming convention and use it everywhere for docs you create.

Standard format: YYYY-MM-DD-topic-short-context

Examples: 2026-04-10-newsletter-folder-rule-draft · 2026-04-10-client-intake-revision-notes

Why it works: dates sort naturally, topics are visible at a glance, search gets faster.

My chosen standard:

Date format: _____

Separator (hyphen / underscore): _____

Short-context style: _____

Notes and attachments style: _____

Folder Boundaries

What goes into **Now** immediately?

What moves to **Reference** or archive **weekly**?

What gets **deleted**, not stored?

Grit Framework — File System Check

Passion: Is my most active work visible enough to follow my curiosity?

Perseverance: Can I find what I need after an interruption without a search spiral?

Growth: Is there anything I should rename or move that no longer reflects reality?

Resilience: Is this system fast enough to use on a low-energy day when search has to do the heavy lifting?

Fire Horse Prompt

The Fire Horse charges forward — powerful, independent, and sometimes too fast for its own structure.

What are you charging forward **without** because your structure is too complex?

What simpler structure would make independent action easier?

Protect Your Attention

Your distraction is understandable. Your tools are designed to compete for your nervous system. This is not a discipline problem — it is a defaults problem.

Inbox Flow

Aim for **inbox clarity**, not inbox zero. Use these four buckets as labels or folders:

Label	Meaning	Your current inbox: does this bucket exist?
Now	Must respond today	
Later	Needs a response, not today	
Reference	Keep for records, no action	
Archive	Done — no action needed	

What counts as "processed" for you?

What gets archived without reply?

Notification Triage

friction-maxxing: choosing intentional inconvenience so your priorities can catch up before you react.

Sort each platform into one category. If an alert cannot earn its place in your nervous system, silence it.

App / Platform	Keep real-time	Batch during response window	Turn off
Email			
Text / DM			
Social media			
Calendar			
Team tools			
Other:			

Badges create background stress. Start by removing badges from social apps, promotional email, and activity alerts from project tools.

Response Windows

Replace "always on" with windows you can actually keep. Outside these windows, the default is not constant scanning.

Morning inbox pass: _____

Afternoon inbox pass: _____

Final pass / close of day: _____

Deep work protection block (no inbox, no notifications):

My Response Boundary Statement

Write one sentence you can use in your email signature, Slack status, or voicemail — whatever channel needs it most. Make it sound like you.

NAA for Interruptions

When you get pulled off-task, run this in under 60 seconds:

Notice: What interrupted me?

Adjust: What will I silence, snooze, or close right now?

Acknowledge: Write it here — *"I returned to focus."*

Build a Rhythm That Holds

A system is only real if it survives your hardest weeks. Build for the week you have, not the week you wish you had.

Weekly Reset (20–30 minutes)

My weekly reset day and time: _____

Checklist:

- Empty downloads and desktop into intake folder
- Archive or process inbox to my target count
- Close stale tabs and windows
- Check notification settings for drift
- Review this week's commitments and choose top 3 for next week
- Other: _____

Weekly reflection prompt: *What is one thing I can sustain this week?*

Monthly Reset (60 minutes)

Checklist:

- Review friction log from the month
- Update folder structure if needed (do not over-rebuild)
- Remove or pause tools I am not using
- Adjust notification boundaries
- Choose one process to simplify next month

Monthly reflection prompt: *What did this month teach me about how I work?*

What worked this month?

What kept breaking?

Which tool or workflow should be retired?

Fail-Soft Reset (5–10 minutes for low-capacity days)

fail-soft: a reduced-mode fallback that keeps the system running instead of failing completely. On hard days, you do not need to run the full system. You need to keep the system alive.

On a rough day, do only these four steps:

1. Run NAA once (Notice, Adjust, Acknowledge): _____
2. Clear one friction hotspot — desktop, inbox, or tab pile:

3. Write one line: *What is this struggle trying to teach me right now?*

4. Choose your one next smallest move: _____

That is enough.

Grit Framework — Maintenance Check

Use this at the close of any reset.

Passion: Does this setup still support what I care about?

Perseverance: What is one thing I can sustain this week?

Growth: What did this month teach me about how I work?

Resilience: What support keeps me from starting over next month?

Next Step *(Required)*

One concrete action I will take next:

Deadline: _____

Closing Reflection

You do not need a perfect stack. You need a system you can return to.

Digital spring cleaning is not a one-time purge. It is an ongoing practice of noticing friction, adjusting gently, and honoring what your brain actually needs.

Reflection sessions, not rescue. Tech that works, for people like us.

Created by The Pythoness Programmer | www.pythonessprogrammer.com