

Let's expand on the ideas of my [project management framework for knowledge workers](#).

I focus a lot on tasks. I say I do that because a project, like eating an elephant, is done one bite at a time.

One bite at a time

Why do I put tasks so prominently in my project management framework? Why don't I start with something like a Project Charter, or a Work Breakdown Structure, or some sort of high level systematic view of what a project is?

I liken all my experiences in my professional life to examples from my time working as a cook in various kitchens. There are many analogies that come from this, but the one for today is:

"if you don't prep and you don't put away, you won't cook another day"

If you aren't prepping your ingredients ahead of time, when the dinner rush comes you will be in absolute chaos trying to piece things together from base ingredients, or even unprepared ingredients. It's impossible. Every serious kitchen will prepare ingredients ahead of time for quick incorporation in the dish when it is ordered. This is work planning, and effective task management.

If you aren't washing and putting away your dishes then you will maybe get through today's set of guests but what about tomorrow? Part of the glamour of a restaurant is the dish pit deep in the bowels, steadily cleaning what others will need. This is timely filing of references and strict version control.

Finally, in addition to that phrase, if you aren't communicating between the servers taking the orders, the food runners, the floor manager, the head chef issuing orders to the cooks at each station, coordinating the timing of their dishes to coincide with each other, then the food is going to randomly arrive on people's plates and guests won't be happy. This is project reporting.

One task at a time

To effectively manage the complexity of a project we look at the basic unit of a project and then build outwards. Possible? Maybe. I quite like the attempt I've made. Ultimately, this approach works because it matches how real work actually gets done—in clear, definable, manageable pieces. When tasks are managed effectively, the documents, deliverables, and packages take shape naturally, ensuring projects remain focused, transparent, and achievable.

I'm being a bit coy, perhaps, in saying that I don't have a top down structure for what a project is. Of course I do. But this structure is a bit different. It's a nested hierarchical structure, with self-similarity at every level. That is, no matter the level, the sub-hierarchy is the same, the reporting structure is the same, the user types are the same. Here's what I mean:

12 Rules for Project Management

These define the parts of a project

1. A **Project** is comprised of **Packages**
2. **Packages** are comprised of **Deliverables**
3. **Deliverables** are comprised of **Documents**
4. **Documents** are completed through **Tasks**

These are how those parts relate to each other

1. **Tasks** are managed through the **9 Domains**
2. **Tasks** are tracked at the level of **Deliverables**
3. **Decisions** are tracked at the level of **Packages**

These are holistic rules that tie everything above together (which connects tasks with the deliverable, and hence the project)

1. **Documents** are 'statused' daily as part of the **weekly sprint log**
2. **Deliverables** are 'statused' weekly with planning for **now, next, and later**

3. **Packages** are 'statused' weekly with planning for the **6 week lookahead** (to be integrated into the master schedule)
4. A **coordinator** role takes ownership of the **task tracker** but task responsibility is allocated to specific people and resources
5. **Communication** occurs in all channels and the responsible member(s) must update the tracker at least weekly with this **essential information below**:

-
- Document status changes
 - If a "HOLD" status, what: decision, information, resources, approval
 - Any decisions that were made (to be transferred to the decision log)
 - Discussions and meetings
-

Definitions

- Packages:

- Packages organize the overall scope into manageable components, each with a distinct purpose, scope, or strategic outcome.
- Can exist in nested hierarchies, according to the needs of the work breakdown structure.
- Decisions, key milestones, and higher-level coordination activities are tracked at this level. Priorities are delegated from this level.
- Deliverables can exist at any package level.

- Deliverables:

- Tangible outputs or results expected from a package. Each deliverable serves as measurable evidence of progress and successful completion of work within a package.
- Deliverables are the components of a package the tie directly to scope items and are routinely tracked and reported at defined intervals (e.g., weekly).
- Tasks are tracked at this level.

- Documents:

- Documents are the specific, verifiable artifacts—such as reports, specifications, checklists, plans, software or any tangible recorded output—that result from executing tasks.

- They undergo routine status tracking (often daily) and maintain structured version control.
- They are developed according to structured methods, guidance documents, and checklists to ensure quality, compliance, and consistency

- Tasks:

- Individual, actionable units of work required to create, develop, or complete documents.
- Tasks represent the lowest operational level in the hierarchy, managed systematically through clearly defined activities (action items, assignments, prioritization, checking, approvals, decisions).
- Tasks follow from the structured methods, guidance documents, and checklists for each document.

Tasks require discipline

By focusing on the atomic unit of a project as the locus of control, we minimize the complexity and gain better understanding through effective reporting of progress. This kind of flexible yet structured approach is only possible when the project team and the corporate culture promotes and fulfills the obligation to act with strict discipline in the few key areas that demand it:

1. Strict version control and retaining feedback with versions
2. References catalogued promptly and proficiently for future use
3. Capturing and propagating decisions effectively
4. Providing clear distinction between information that is reliable and what is still in progress

When instructions and roles are unclear and always shifting, when procedures are dense and complex and information has many possible routes to follow, when decisions are made without information from the "ground level", then the effort to maintain discipline in those 4 areas becomes extraordinarily difficult for most people. In those times, chaos takes over and people survive mostly on their wits and hard work.

Why this matters

This matters because the most common reason projects fail isn't a lack of sophisticated planning tools or intricate hierarchies; it's because the fundamental units of work—the

tasks—are poorly defined, inadequately tracked, or disconnected from the broader context. Such approaches frequently result in:

- **Tasks that are disconnected from goals**, leading people to work without understanding how their contributions directly impact outcomes.
- **Poor visibility**, as it's unclear how task-level activities roll up into meaningful progress at the deliverable or package level.
- **Decisions that get lost**, because they're not captured at the appropriate level or connected clearly to the related tasks.

In contrast, your bottom-up, task-first approach creates clarity and accountability at every level:

1. **Clarity and Accountability:**

Each task is clearly linked to tangible documents and deliverables. Individuals performing tasks clearly understand exactly what they're responsible for and how their work contributes directly to the overall success of the project.

2. **Transparency and Consistency:**

By establishing a clear, consistent tracking approach (tasks to documents, documents to deliverables, deliverables to packages), progress and issues become transparent. You know precisely where the project stands at any given moment because status updates follow the same disciplined process across every layer.

3. **Decision Capture and Integration:**

Decisions are systematically recorded and managed at the appropriate hierarchical level. This ensures important decisions don't disappear into meeting notes or email chains. Instead, they are explicitly tied to the relevant package or deliverable and flow down into actionable tasks.

4. **Simplicity Through Self-Similarity:**

Using a self-similar hierarchical structure reduces complexity and overhead. Every participant—from frontline workers to senior managers—knows exactly how to track, escalate, and resolve issues because the process is identical at every level. This reduces training time, minimizes misunderstandings, and streamlines communication.

5. Flexibility and Adaptability:

Because tasks are the fundamental units, it's much easier to adjust the project's course when conditions change. Rather than reworking an entire rigid hierarchy, you can quickly reassess, reprioritize, or adjust tasks, which immediately propagates upwards through documents, deliverables, and packages.

The Kitchen

The kitchen taught me to take care of these details and to do so with high degrees of discipline to mitigate so many future chaotic issues from arising. If you don't prep and you don't put away, you're not going to be doing another project with that client again.