1. **Developer Pomodoro Timer w/ Analytics**
    a. **Statement of Purpose**: Developers often struggle with managing their time while maintaining productivity levels due to the variety of tasks they must handle (coding, debugging, code reviews, research). This becomes particularly worse due to the context switching that occurs between these tasks. This tool will integrate code tracking to provide developers with insight into task distributions during their work day and (reduce burnout??).
    b. **Possible User Personas:** developer in the industry, student developer, manager who's looking to get insight into team's productivity
    c. **Must Haves (P0s):**
        i. Pomodoro Timer w/ customizable work session lengths + break time lengths
        ii. Task Selections: Select type of task (coding, debugging, code reviews, research, etc.)
        iii. Productivity Insights: Daily/weekly charts showing time spent per task time (maybe with visuals)
    d. **Should Have (P1s):**
        i. Notification reminders/Pop Ups for breaks/work sessions
        ii. Separate into Projects: Ability to tag each work session to a specific user story or project name to get insights into a specific project
    e. **Nice to Have (P2s):**
        i. Contrast Polarity/Different Themes in UI
        ii. Basic Dashboard for productivity analytics
        iii. Integration to Google Calendars etc. to help reduce manual work

2. **Streak Tracker for Developers**
    a. **Problem**
    As Professor Powell showed in class, a lot of developers find themselves busy all day but eventually end up not accomplishing much at the end of the day. This is often because they get distracted by the plethora of trivial tasks. How can we help developers maintain their focus/productivity and keep them consistently motivated, not just for a single day, but for an extended period of time?
    b. **Approach**
    We can try using the psychology of maintaining streaks. People can become strongly motivated by streaks, sometimes so much that they force themselves to do the work just to avoid breaking the streak. This can encourage more regular engagement from the developers and provide the small nudge that they needed to stay on track. Streaks can also provide structure to how developers approach their work. By accomplishing smaller and more specific tasks each day, they can build confidence in their ability to reach a goal and gain momentum. A classic example of an app that demonstrates the effectiveness of streaks is Duolingo, but it can also be found in many different fields like exercising, games, social media, praying, etc.
    c. **Possible User Personas**

- Developers who wish to find a more structured way to make consistent efforts in their work.
- Students, who want to compete against their friends, on who can maintain longer streaks.

**d. Essential features**

- Users set simple daily goals (something like a todo list or a checklist or even a schedule) at the start of the day.
- Users determine which goals they have successfully reached at the end of the day (maybe also write reflection?).
- A counter that highlights the number of consecutive days (excluding weekends) users successfully set and completed the goals.
- A simple visualization of the streak.

**e. Potential features**

- Streak reward (users can collect rewards/badges after maintaining streaks for X number of days)
- Weekly challenges (e.g. help other developers out, go out and network)
- This "productivity tool" can even become more like a game (e.g. users can collect random pokemon every day of their streak and they can eventually catch them all)
- Reminders and notifications (send them reminders but do not spam)
- Motivation tips and quotes

## 3. Gamify your goals (or development)

### a. Problem / motivation

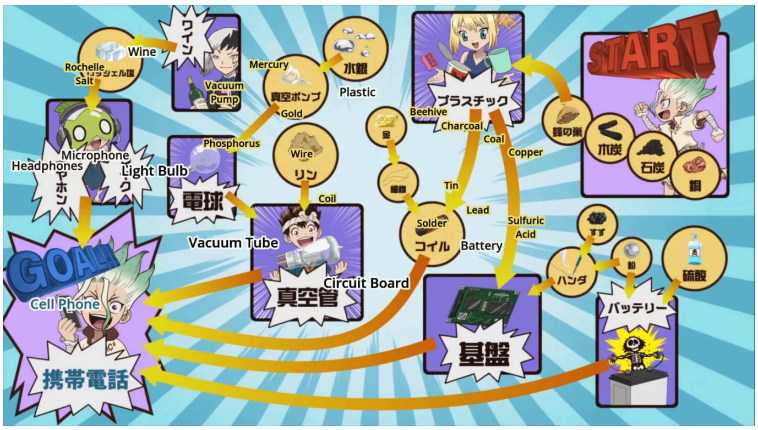
- Games that offer virtual rewards are often addictive. There is something about being rewarded (though virtual - https://dl.acm.org/doi/pdf/10.1145/3357236.3395477) that keeps us going (think about grinding through a level in a video game just to get a costume).
- Whenever we are working towards a large goal, one might feel lost. However, if there is some hand holding (or just the illusion of it), we tend to get the work done. We must view the task as steps and move through them.
- One can't flex knowledge, but one definitely can flex a degree. Clash of clans ain't getting you laid but you still upgrade that town hall.
- Even bad projects should be rewarded, we ensure that you can flex a nice badge instead of your crappy project (https://blog.professorbeekums.com/2022/developers-need-to-make-mistakes/).

### b. Solution

- An app (web app) to help you design your own path to achieve something. The app provides a visual representation of the path with future levels being locked. Complete the current level to unlock the next, gain coins along the way.

- We effectively promote incrementalism and iterative work style (as suggested by Farley).

### c. Essential features

- Ability to design a roadmap.
- Help breakdown the activity into multiple stages (if any stage seems too vague/big - the app suggests breaking it down).
- Generate a quick gif at the end that can be shared on all social media platforms.
- Show progress in a gamified way, like the representations in Dr. Stone (it's an anime) or Age of Empires, but more elegant.
- 

- 


### d. Potential features

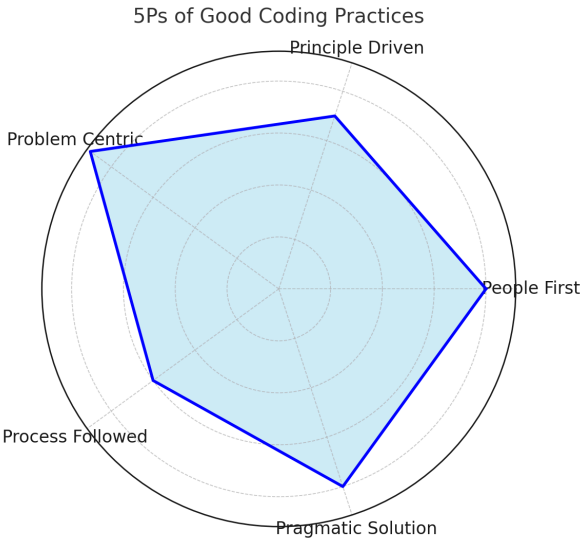
- Random reward periods where users have to upload the proof of completion within a time limit to obtain exclusive badges.
- Ability to view other user's walls (walls are composed of multiple maps created by the user).
-

## 4. Streak Tracker for Developers Add Ons

a. **Approach Add Ons**
   We can also introduce the idea of competing with friends to make our user more “addicted” to our app, just like video games. So now they are not only motivated to keep the streak, but as well as beating their friend. For instance, for every hour they spend on our app, they get XP awards and increase their rank in some kind of ranking system. So instead of letting users be distracted by other things, we “distract” our user into our app which in reality helps them focus on their tasks.
b. **Essential features Add Ons**
   - A ranking system of some kind
   - Ways to award the user, and it cannot just be the user claiming they achieve all their goals today, otherwise they can kind of cheat
c. **Potential features Add Ons**
   - To keep our user more “addicted”, we can have animation at the bottom corner with things like a character hitting bamboo or doing other tasks, and for every bamboo hit the character gets like 1 xp, so the user is motivated to upgrade their characters by using our app

5. **5Ps tracker for Developers - Introspective Productivity Tool**
   a. **Problem Statement**
      Prof. Powell has underlined the importance of following good software development practices. When working on a project, it is important to periodically introspect about the progress of the team. (Collaborative) journaling is a good way to do this, which calls for a productivity/journaling tool geared towards introspection. Our journaling tool will be such that a team can grade themselves according to how they performed in each of the 5Ps. They can also set some custom evaluation criteria. The tool will give out visualizations and summaries and keep a track of whether there is holistic progress in the project. It will also have a way to give reminders about practices to focus on according to the current status of the project. We can expand the scope to some productivity features too, like Pomodoro, calendars, accessible code snippets, making our product an introspection+productivity tool. Lastly, the tool need not be 100% introspective. The same set of functions can be used for team reviews from supervisors or external auditors.
   b. **User Personas**
      As it is an introspective tool, useful for any software development team, any open source project, or even an individual development effort!
      Project managers can use it to continuously monitor their team's progress, ensuring adherence to best practices and identifying issues promptly to keep projects on track. Team leads can leverage it for end-of-sprint evaluations using detailed records and visualized data to improve team performance. Freelance developers can benefit from its self-assessment capabilities to regularly review their project progress and coding practices to boost productivity and quality. Agile coaches can utilize the tool to maintain consistency in agile practices and provide visual feedback that encourages continuous improvement.
   c. **Must have features**

i. Markdown journaling feature
ii. Categorical introspective journaling (have some notes about performance in terms of each of 5Ps in each sprint)
iii. Numerical introspective grading category wise
iv. A summary page/dashboard that shows your “development health” with respect to the 5Ps
v. An example visualization we could have is the following



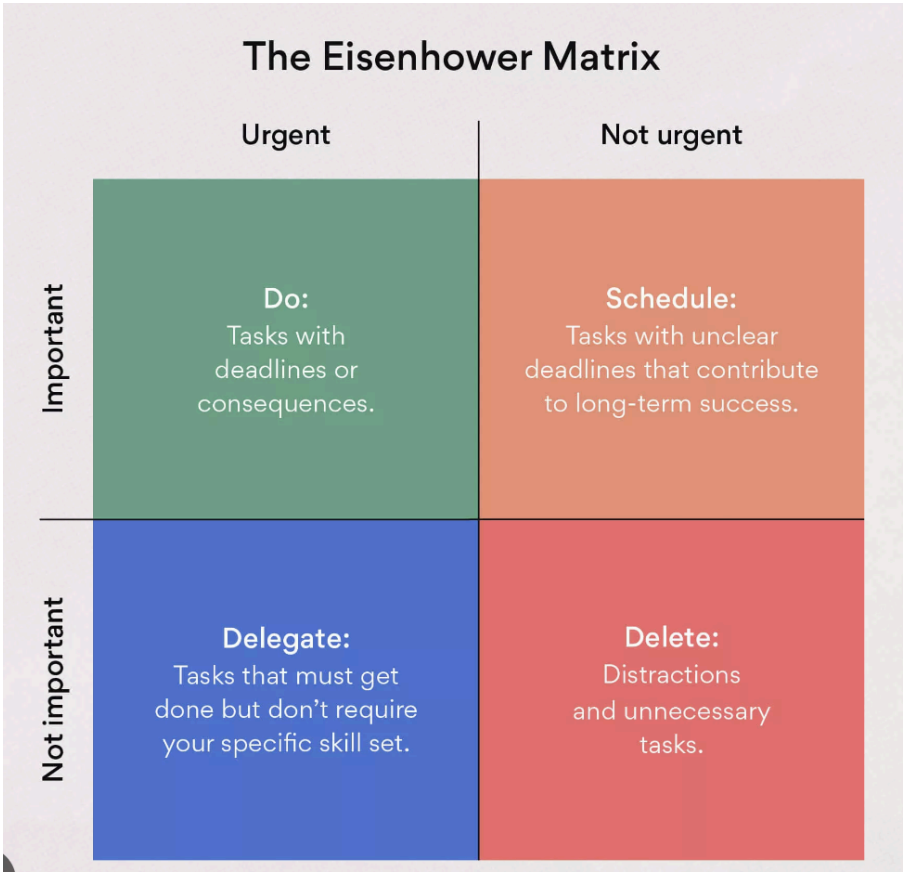

**d. Should have features**
i. Periodic reminders for coding practices and concrete actions that need to be followed, according to the plan made in recent introspection
ii. Some gratification tool - The tool gives your development effort stars/points according to your progress and how you are following up on your introspections.
iii. Additional productivity features - Pomodoro, fetching past code snippets, calendar. Tool will suggest/put those tools on top that can address whichever area among the 5Ps you are lacking in.

**e. Nice to have features**
i. A cute puppy/mascot on the landing page that is happy when you are following good coding practices according to your introspections. Not sure how to do it, but if we can make some nice animation or import it from somewhere, it will give a nice feel to the tool.

**6. Developer Eisenhower Matrix (Task Prioritizer)**

**a. Statement of Purpose/Problem Statement:** Developers often have a long list of tasks that vary in urgency and importance, including coding tasks, debugging, code reviews, and tickets. This tool will use the Eisenhower Matrix to help developers quickly prioritize tasks, visualizing them based on urgency and importance. By offering a structured approach to prioritization, it aims to help developers work more efficiently, focus on high-impact tasks, and reduce the

feeling of being overwhelmed. The Eisenhower Matrix is a tried and tested tool that most developers use, especially in the software development industry:




b. **Possible User Personas:** Industry Developer, Student Developer, Team Manager
c. **Must Haves (P0s):**
    i. Eisenhower Matrix Visualization: A 2x2 grid where tasks are organized into categories shown above
    ii. Task Entry and Categorization: Ability to add tasks with details and deadlines, and assign them to the appropriate quadrant.
    iii. Priority Summary: Quick summary showing the number of tasks in each priority category, helping users see at a glance where most of their workload lies
d. **Should Haves (P1s):**
    i. Progress Tracker: Charts that are a representation of completed tasks in each quadrant to show progress over time (maybe even separated into sprints??)
    ii. Filter and Sort Option: Ability to sort by due date/sprint end date/project deadline
e. **Nice to Haves (P2s):**
    i. Team Sharing: Ability to log in using validation & authentication to view entire team's chart

ii. PDF option: If authentication isn't within scope of project, we can add the ability to export the chart to a pdf/doc or csv so that the entire team can view and share the document

7. **A pdf processing chat robot**
    a. **Problem:** Editing pdf is generally not an easy thing to do. People usually use microsoft-word for documentation. Only people who are familiar with overleaf or latex, or markdown can tackle pdf with ease. But still, reading a pdf is generally not an easy thing, and many ordinary people don't have a good way to handle pdfs.
    b. **Solution:** chatgpt-like robots should solve such problems concerning pdfs. In this robot, users can upload their pdf document to the website and ask questions regarding the content of the pdf.
    c. **Must have features:** Users should be able to upload one / multiple pdf document, remove some pdf and ask questions regarding the content of these pdf documents.
    d. **Should have features:** Users can ask their questions by speech and the robot can respond to these questions via speech. Users can voice their needs and the robot should be able to generate a new pdf that satisfies the user' needs.
    e. **Nice to have features:** The robot should be able to add certain content into the pdf by user prompt, the added content should have the same style as the original pdf. The robot should also be able to merge some pdf according to user' prompt. In addition, users can voice their needs about certain kinds of diagrams, and the robot should be able to generate such diagrams and insert them into the pdf document.