

The IDIOT method is a project execution framework for breaking any project into smaller subcomponents to better assist in execution on any project. The framework is broken into two parts.

Investigation

Design

Implementation

Optimization

Testing

It's important to always start every project in this order. We should write the entirety of the plan out, taking special care to fill out each section with as much detail of our plan of action. Each section will have a list of questions. Answer the questions to the best of your ability with the knowledge you know. List out considerations for that section of the plan, and validation conditions that let you know you're done with that section and can move on.

Investigation

Questions to answer

- Who is this for?
- What is this project trying to accomplish?
- What are the sub-problems I can break this project into?
- What questions would I like to answer before I dive into this?
- Is there something like this project already?
 - For each project we ask
 - What did that project try to accomplish?
 - What technologies did that project try to use?
 - How many people are using this project?
 - How relevant is this project?
 - Based on my learnings, can I use code from one of those projects to help me?
 - Is it open source with a license that is friendly to my project?
- What technology is the best for this project?
 - What frameworks am I considering using and are they commercially friendly?
- What tools are available to you for solving this problem?
 - What MCP Servers do you think you will use to solve this problem?
 - Are there certain MCP servers you would like to have at your disposal to solve this problem?

Design

Questions to answer

- Draw the architectural schema for all the classes I plan to create to implement this project
 - Describe each of the classes in details with a list of functions with return values that we plan to create
- What technologies are required to build it?
- What will be the cost of running this given the hardware for the project
- How many users are using this system

Implementation

- Based on the design, create each class one by one.

Optimization

- Without compromising the quality of the code, perform optimization on the code

Testing

- Run through all the user proposed suggestions for testing
- Think of other test cases that will test all the business requirements of this project
- Create Unit tests that can be easily run to ensure the proper execution of this code
 - Create E2E tests that can be executed against the service (IF APPLICABLE)

Follow up

- Create a list of follow up items that would be likely next steps for a project like this. Give this as a suggestion to the person that proposed the project.