FDD - Strategies to Make Failure Less Painful / Bothersome

- iteration
- recursion / divide-and-conquer
- automation - rearrange, then push a button to rebuild everything
- layering design (see “Recursive Design, Iterative Design By Example (2), section “Bug 2” and section “Layering Solutions”)
  - https://guitarvydas.github.io/2021/04/20/Recursive-Design,-Iterative-Design-By-Example-(2).html
- indirection
- create a notation, SCN (low-cost)
  - punt to toolbox languages
  - punt to foreign functions (DI & Details Kill)
- asking Why?

FDD

- Failure-Driven Development
- most of the time, the requirements will change
- most of the time, a design will have flaws in it
- most of the time, the implementation will need to be debugged and need repairs
- the number of failures >> the number of successes
- plan for failure, since failure happens more often than success