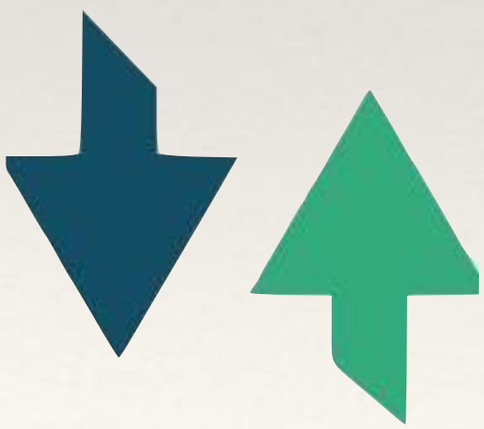
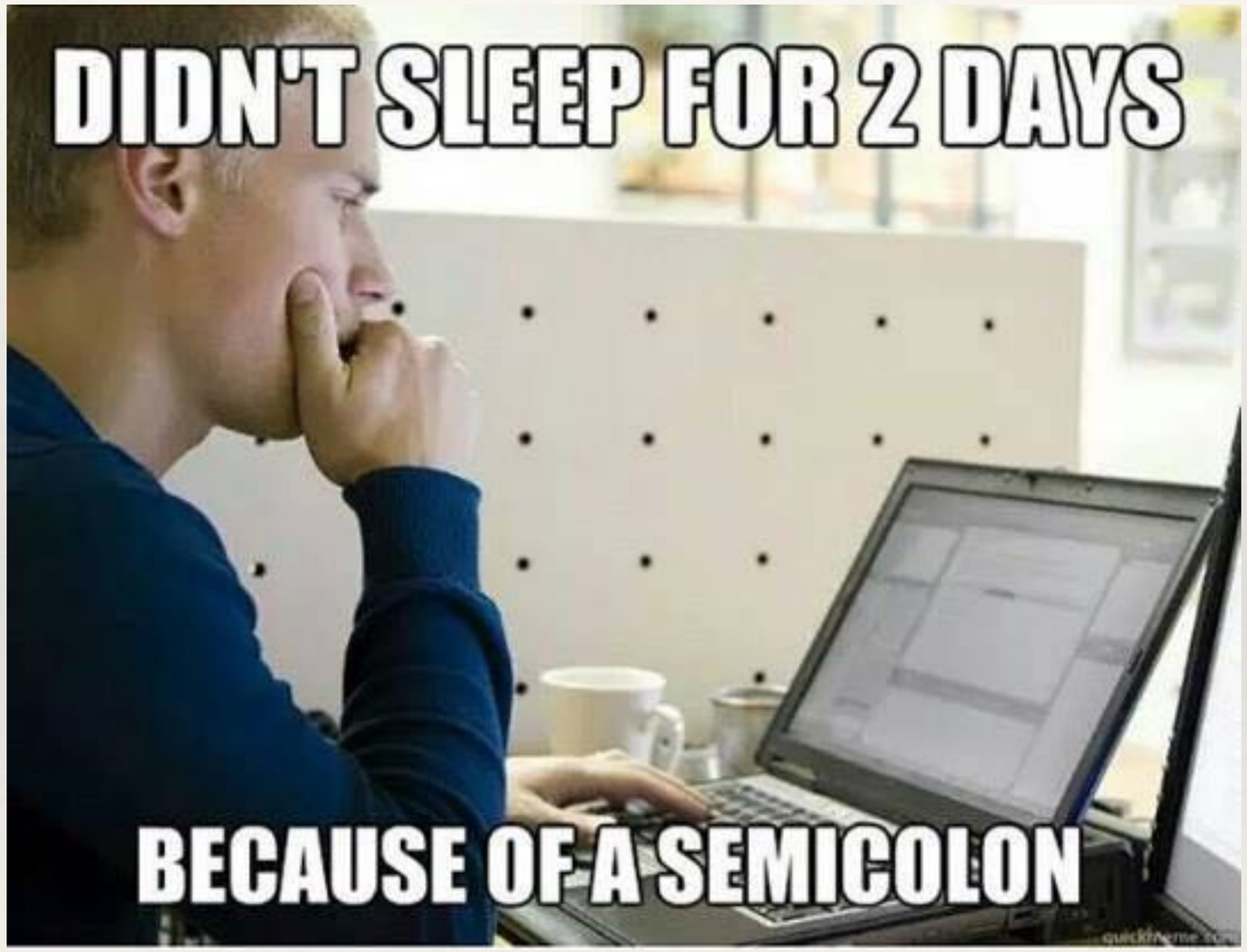
Zip Code Wilmington

Passion Projects

It's all about how bad you want to be successful.



Gotta Move On



Passion Project – architecture

- Three Tier Application (FrontEnd UI, Server, DB)
- UI – (Mobile, Web, Desktop) – Javascript/JSON/??
- spring.io Middle tier Server: Java
- Database: Mysql, Postgres or NoSQL

Each project must have–

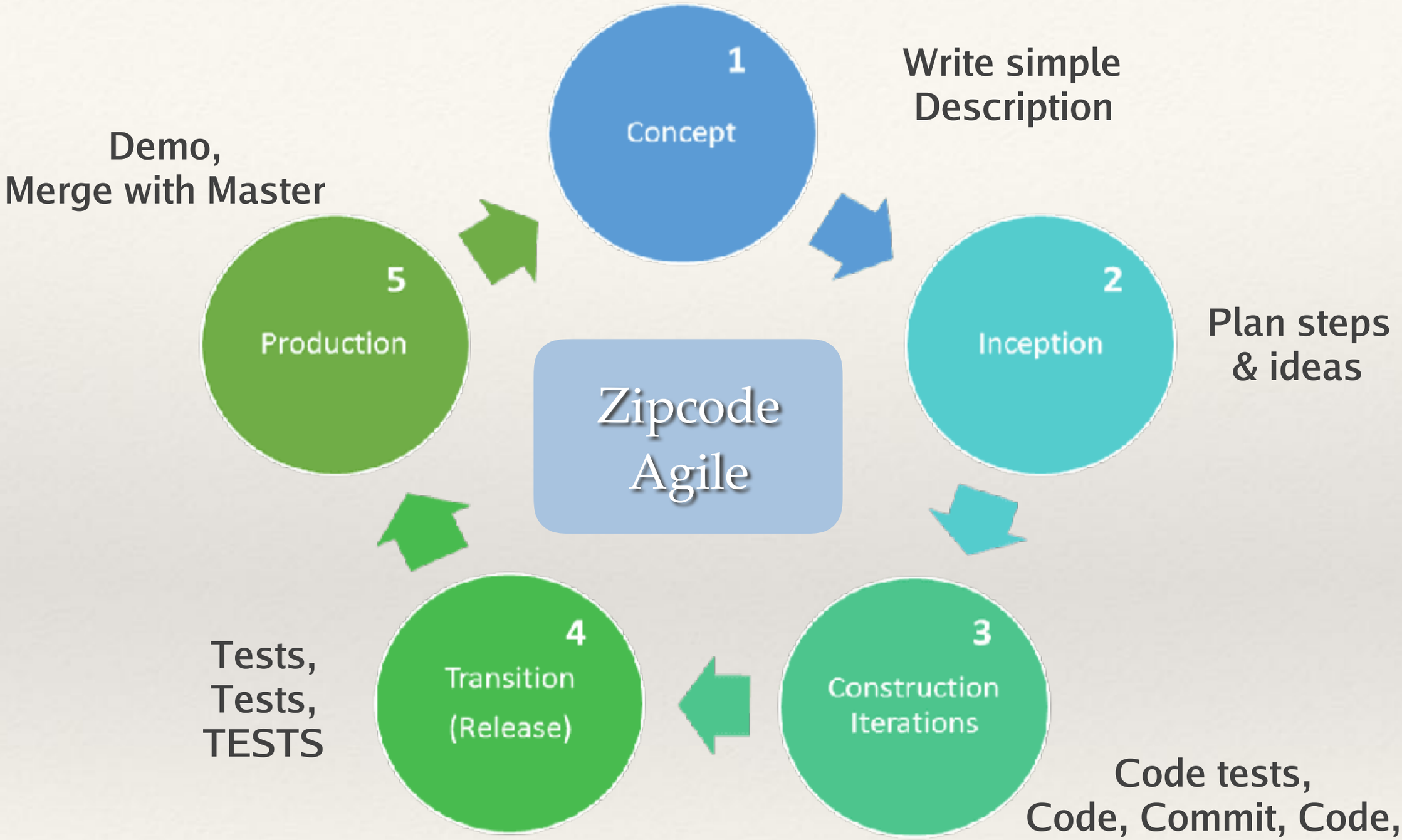
- An overview document of the project
- A plan of phases of the project
- The issue tracker in the repo must track what's being worked on
- Each section of the app needs to keep pages on its design, progress and bugs (using Issues)
- All issues should be attached to project milestones and assigned to team members.

Designs

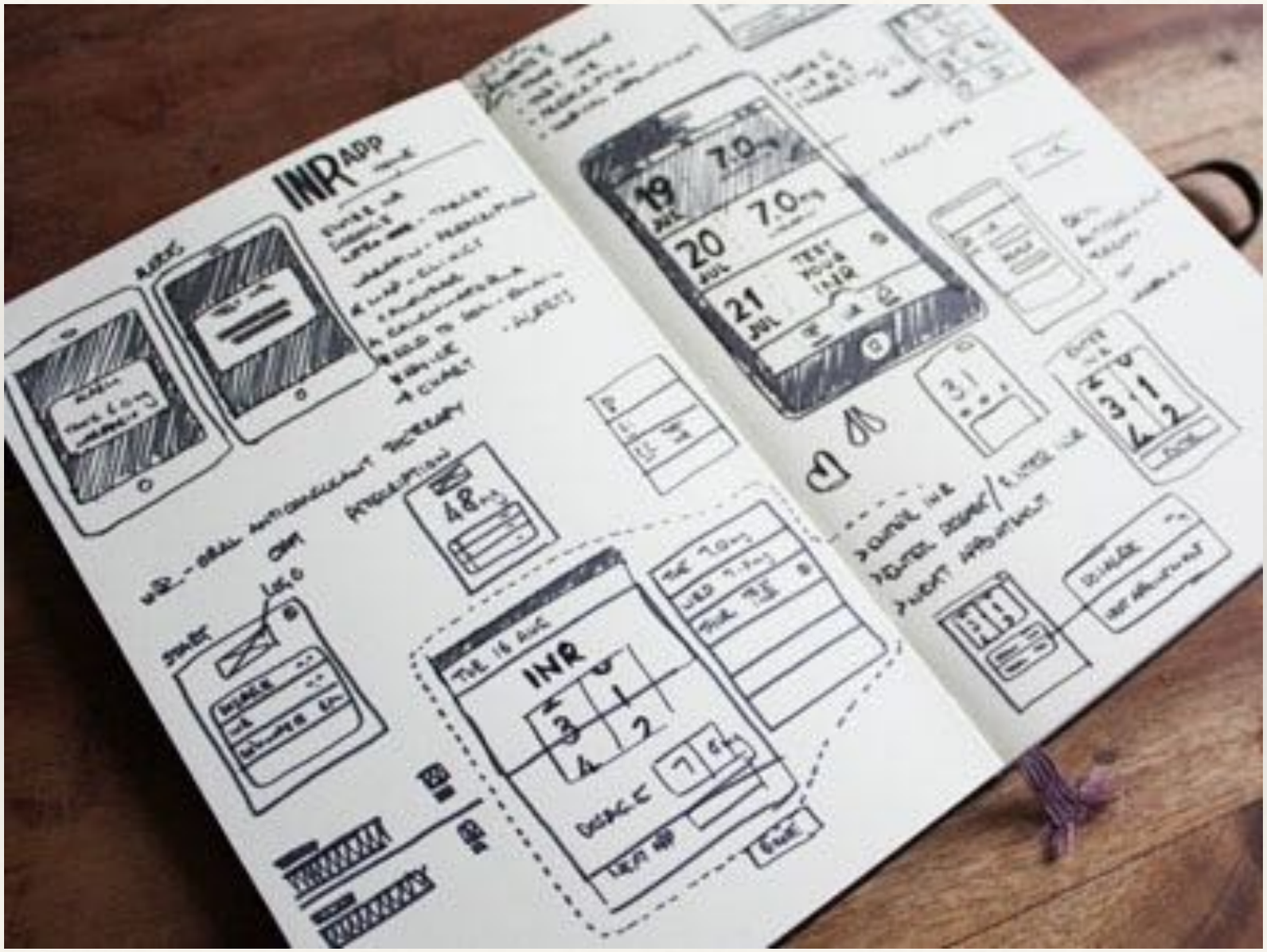
- UML diagrams and specs must be kept, seriously.
- API design docs, for each segment where a protocol needs to be used for data communications.
- Designs should use Model/View/Controller, with Client and Server focus subgroups
- UI/UX should be done in Typescript/Ionic/Angular (or another pre–approved)
- The database schema/objects must documented.
- The app server must be based on spring.io (Spring Boot)

Method(s) – Methodology

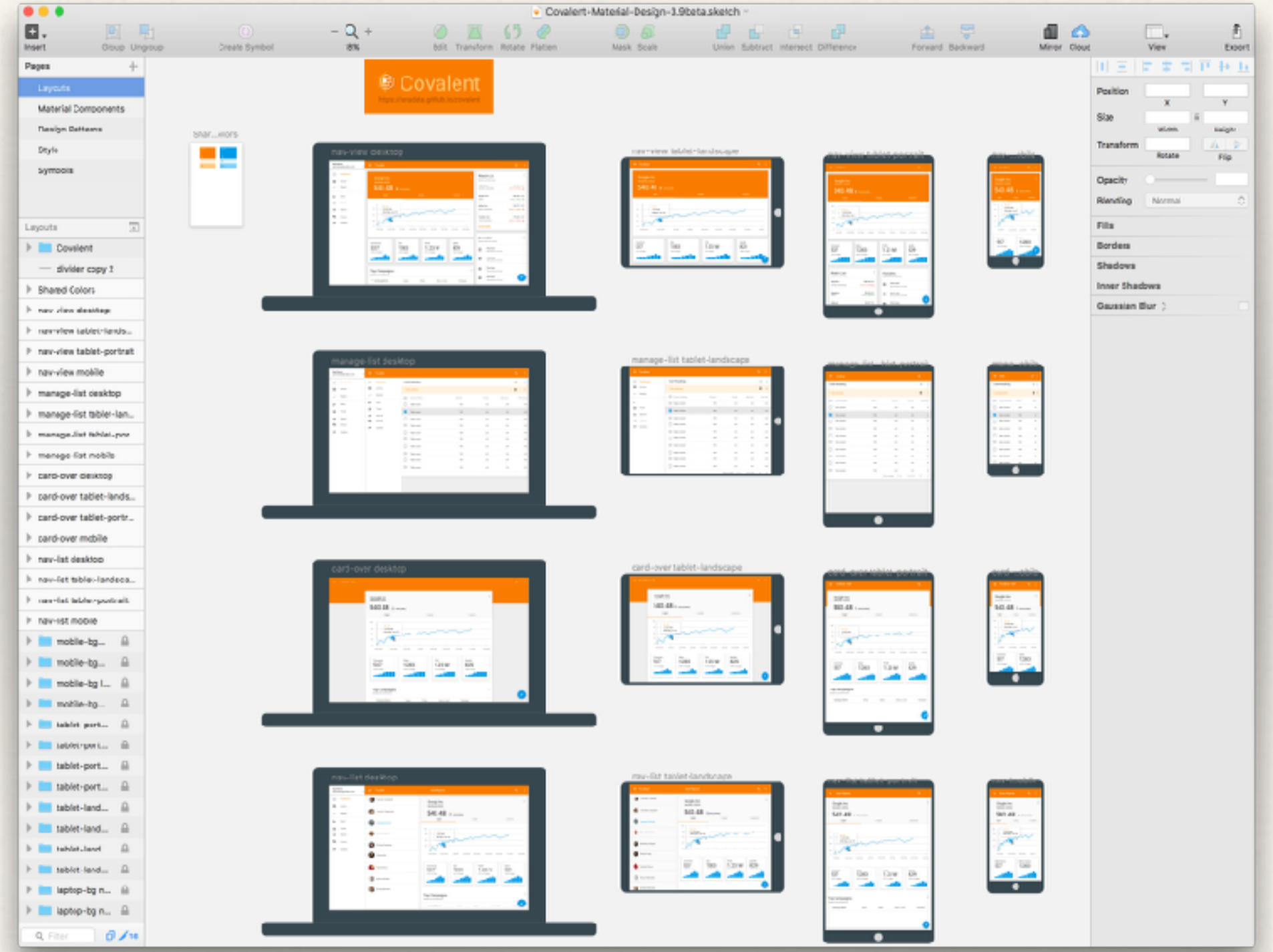
- The project work should be in agile sprints of 2–4 days
  - commit histories are important.
- Each sprint should work in a different git branch.
  - Multiple commits should be done daily on all personal work to branch.
- All sprints end with a complete merge back to the master branch, and a demo/ presentation for an instructor. (Google Slides are Great!)
- The project should start with simple features, and then add to it over the course of several weeks (and several sprints).
- The internet has many different ways of doing each of these projects, tutorials, descriptions and so on, describing how these things can be done. Research on the topics should be considered to be the very first task you should do on the project.


1
Concept
Write simple
Description
2
Inception
Plan steps
& ideas
3
Construction
Iterations
Code tests,
Code, Commit, Code,
Commit
4
Transition
(Release)
Tests,
Tests,
TESTS
5
Production
Demo,
Merge with Master
Zipcode
Agile

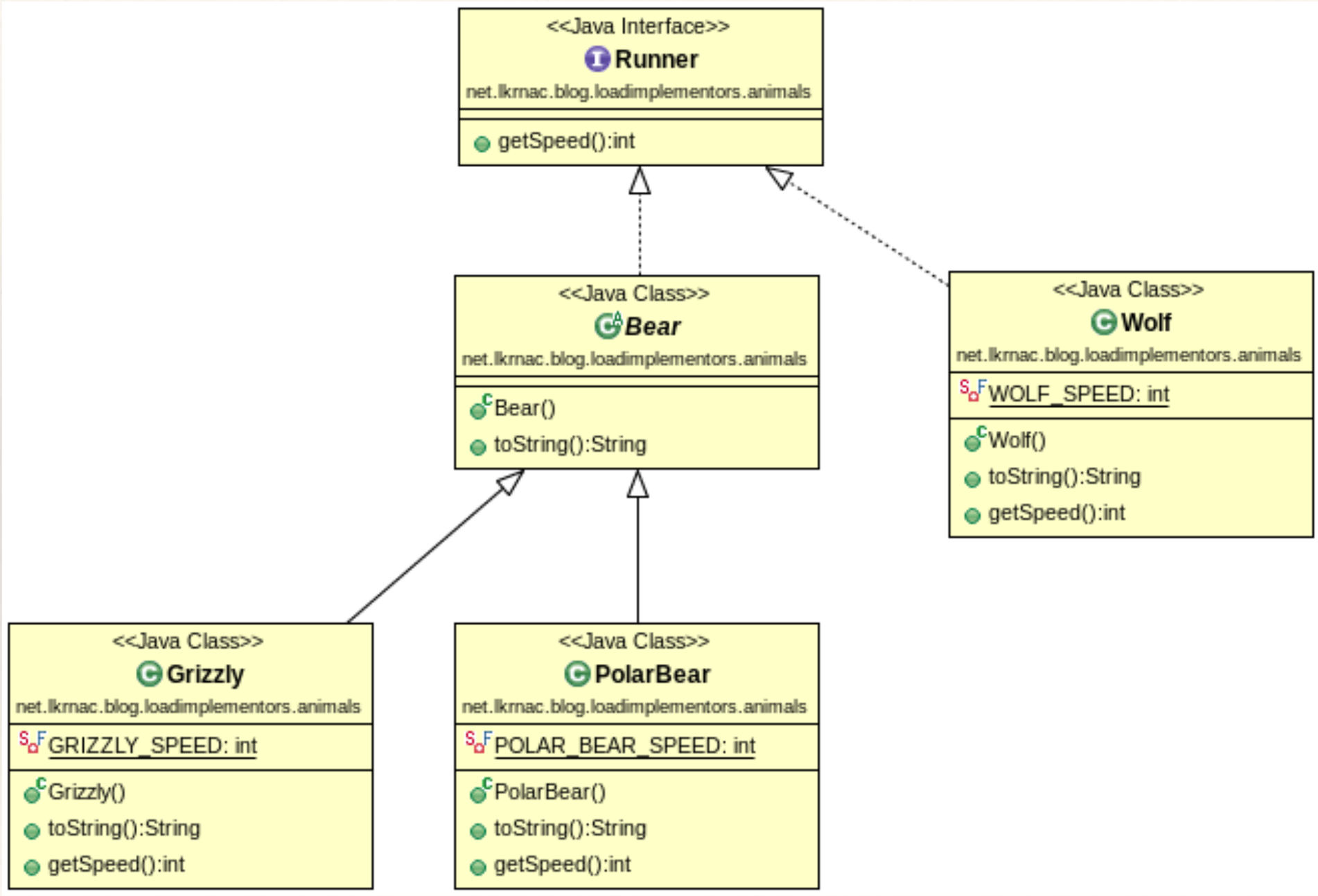
UI Design



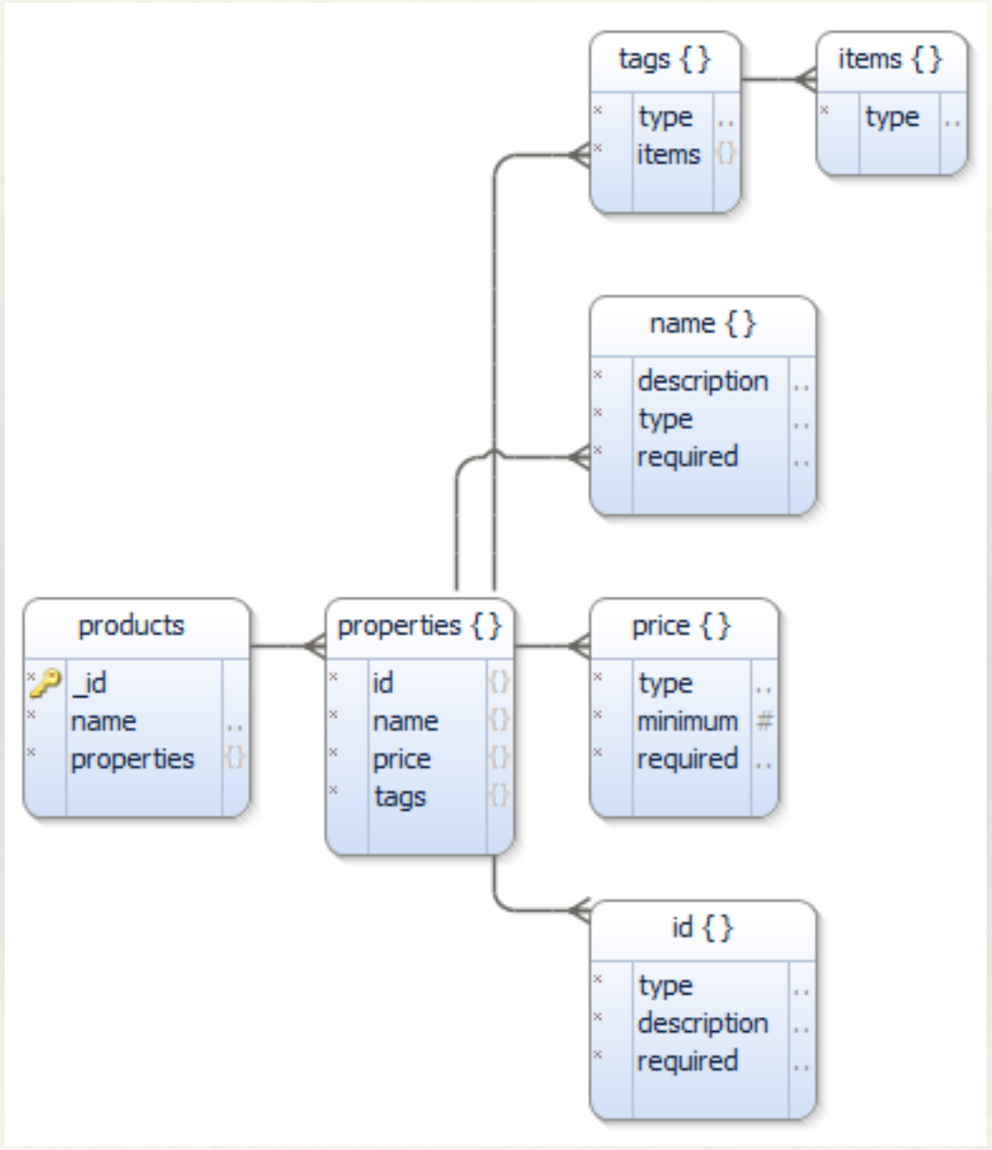
UI Design

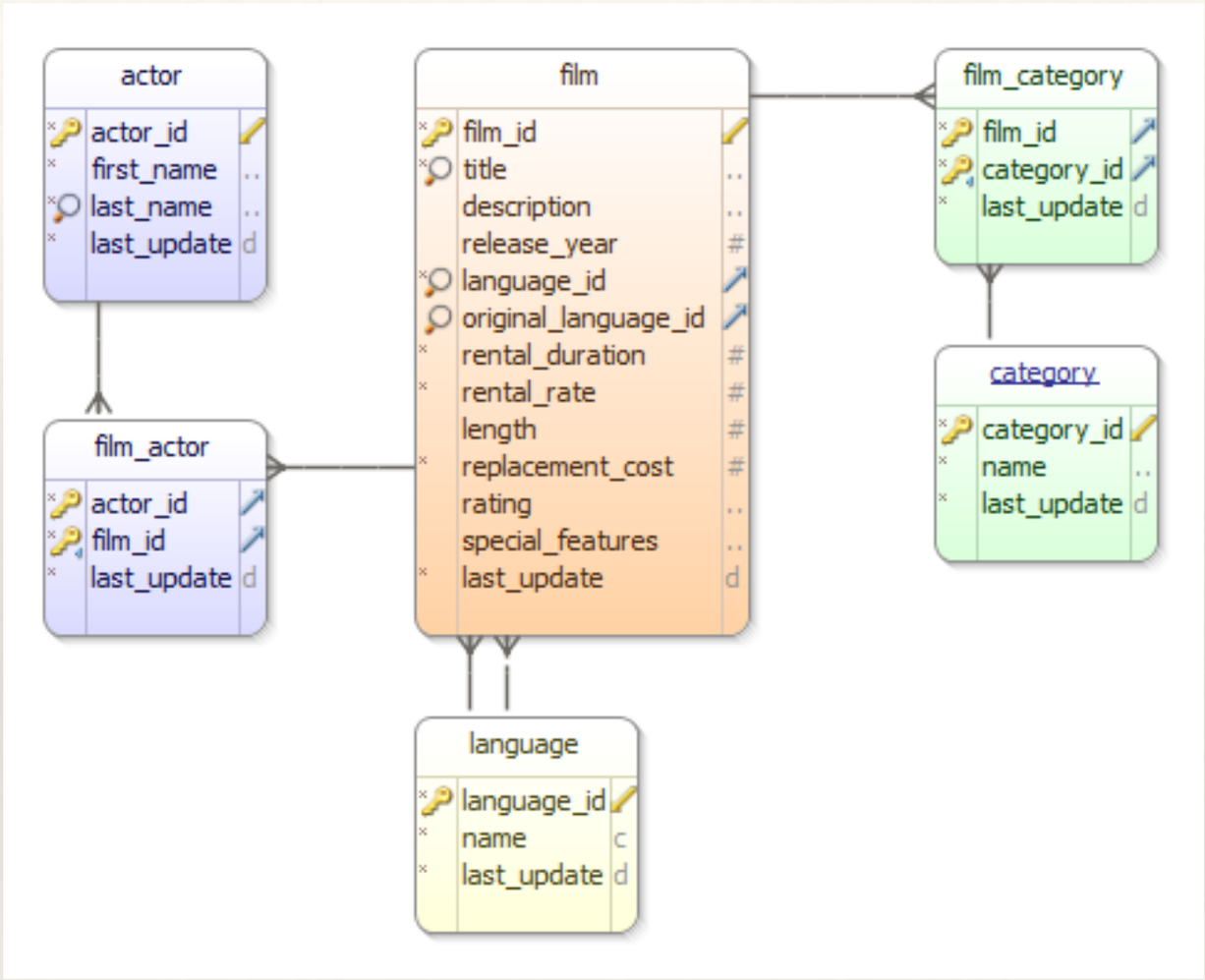

Covalent-Material-Design-1.9beta.sketch
Insert
Group
Ungroup
Create Symbol
85%
Edit
Transform
Rotate
Flatten
Mask
Scale
Union
Subtract
Intersect
Difference
Forward
Backward
Mirror
Cloud
View
Export
Pages
Layouts
Material Components
Design Patterns
Style
symbols
Covalent
https://teradata.github.io/covalent
nav-view desktop
manage-list desktop
manage-list tablet-landscape
card-over desktop
card-over tablet-landscape
nav-list desktop
nav-list tablet-landscape
Position
Size
Transform
Rotate
Flip
Opacity
Blending
Normal
Fills
Borders
Shadows
Inner Shadows
Gaussian Blur
Layouts
Covalent
divider copy 1
Shared Colors
nav-view tablet-portrait
nav-view mobile
manage-list desktop
manage-list mobile
card-over desktop
card-over mobile
nav-list desktop
nav-list mobile
Filter

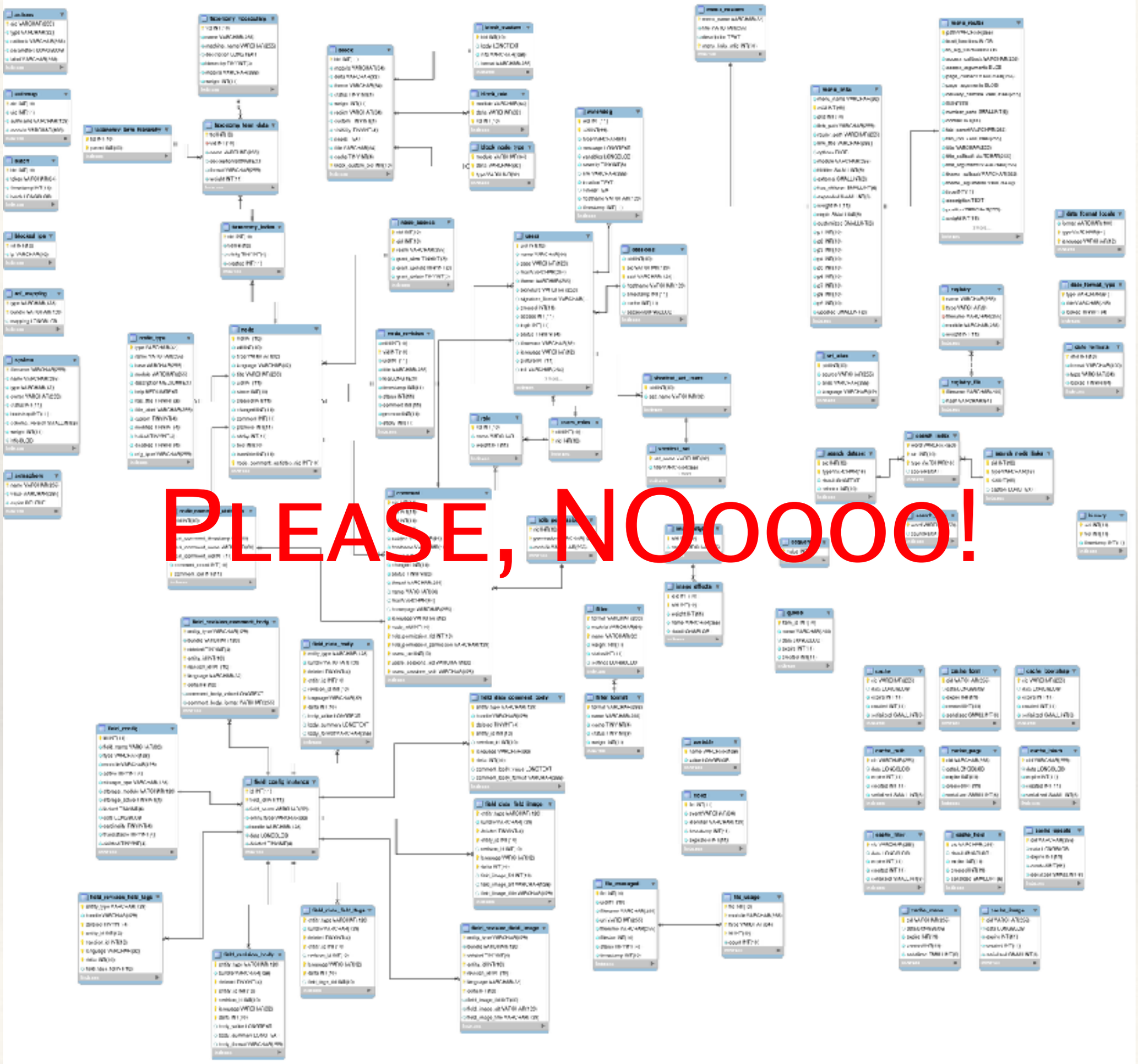
UML Internals

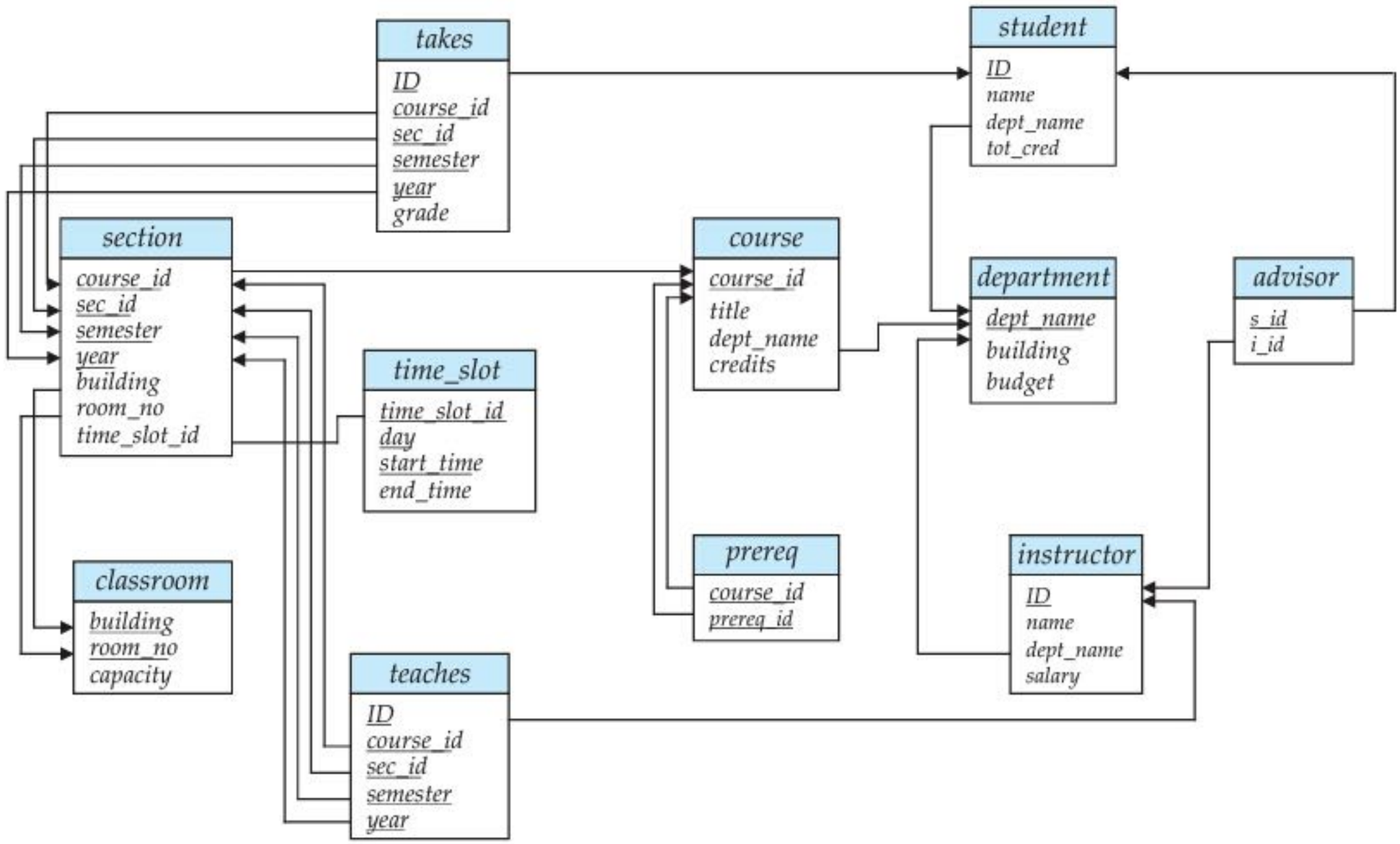


Database Schema (design)

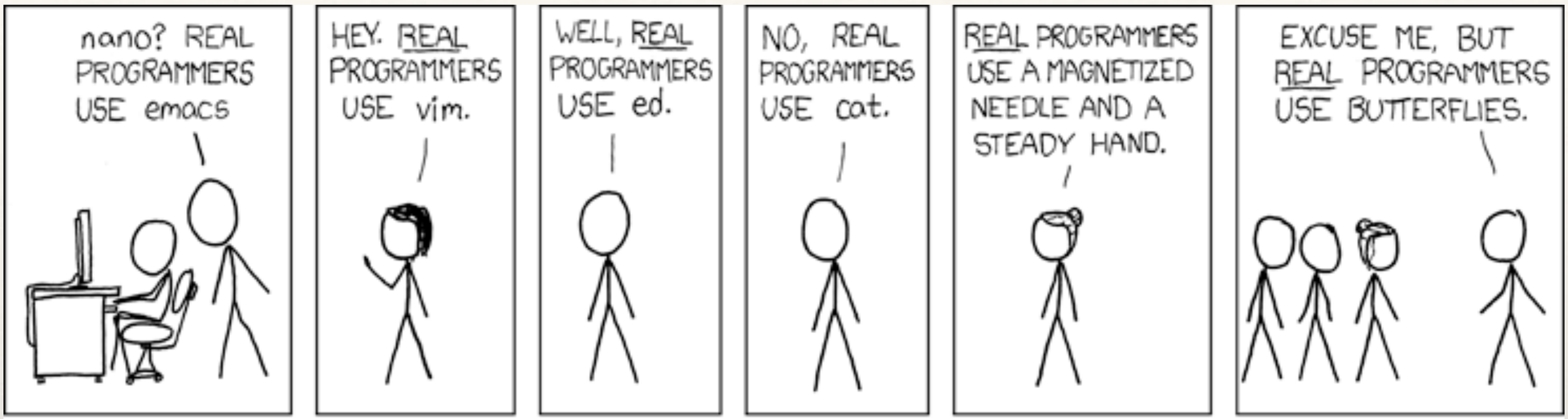





PLEASE, NOoooo!



Just Right, Ahhh!


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.