MVVM

Model-View-Controller

- Model: data process, business logic
- View: UI
- Controller: bridge for Model and View

User interaction --> View

- View --pass calls to--> Controller
- Controller --Manipulates--> Model
- Model --Observer Sync(Events, Pub/Sub)--> View

Good:

- Change-able Controller
- Observer pattern for multiple View(s) update

Ugly:

- Hard test the business logic in the Controller because when Controller changes Model, updating View(s) is done by the View itself.
- View HIGHLY depends on specific Model

MVP

Improved version of MVC

User interaction --> View

- View --pass calls to--> Presenter
- Presenter --Manipulates--> Model
- Model --Observer Sync(Events, Pub/Sub)--> Presenter
- Presenter --Interfaces--> View

Key:

- Test-able
- Presenter has logic for View sync and business
- View provides Interfaces for Presenter to call
- Model still broadcast data change, but receiver is Presenter not View (MVC)
- View NOT depends on specific Model

Ugly:

- Presenter: besides business logic, lots manual logic for View <--> Model makes presenter heavy and hard for maintenance

Model-View-ViewModel

Improved version of MVP

User interaction --> View

- View --pass calls to--> ViewModel
- ViewModel --Manipulates--> Model
- Model --> ViewModel.Binder
- ViewModel.Binder --> View

Key:

- ViewModel has a Binder: two-way data-binding engine that automatically binds View and Model

Good:

- Good code maintenance by abandon heavy Presenter manual logic for View <--> Model and give it to MVVM framework(Binder)
- Simpfy test: as long as Model is correct, View is correct
- Low coupling
    - View --x-- Model: Views can be changed and modified independently of the model
    - 1 ViewModel can be bind to different View(s)
- Reusable: logic in ViewModel that can be shared by different View(s)
- ViewModel can be implemented independently

Ugly:

- Not suitable for super easy UI
- The cost of building and maintaining ViewModel will be higher for a project that has huge UI states.
- Data binding process is not debug-able