

"Plan and Increment": Before coding, project manager defines plan, then detailed incrementation all phases of plan, then we use code against the plan, Change to project must be reflected in the incrementation and possibly to plan

	Waterfall	Iterative	Agile
Description	<ol style="list-style-type: none"> 1. Requirements analysis & specific state 2. Architecture design 3. Implementation & integration 4. Verification 5. Operation & maintenance <p>Depends on project manager</p>	<p>Continuous Plan and Increment with prototypes, Before then plan & increment all requirements i.e., develop plan & requirements increment across each iteration of prototype as needed and evolve with the project</p>	<p>Increment Iteration, Agile emphasizes Two-Dimensional Development (2D) (vertical = number, vertical = time) Iterative to release customer requirements, Iterative (long and periodic) to increase progress</p>
Benefits	Good for bigger issues	Iterative prototypes & evolution	Shorter iterations

HTTP

- HTTP request is below request method (GET, POST, etc.), Uniform Resource Identifier (URI), HTTP protocol version
- Since HTTP is stateless, you use cookies in order to store data to help guide them through the pages of the app (e.g. authentication, flavor, last tracking, customization, etc.)

3-Tier Based Architecture



Shared Nothing Architecture

MVC

- Controller handles the requests and views
- Any action a user takes is based in the controller is made available to the view
- The view from the controller and gives each model the ability to perform (CRUD)
- Any action a user takes is based in the model that is not part of the business is allowed to handle the business logic every request made to the app
- In MVC, each interaction the user makes is handled by a controller action
- Request -> controller -> HTTP verb (GET) -> handle user action

Rule

- In Design, there is a rule to object's get time about the if will have already itself
- Mapping naming lets us define our code at runtime
- Module is a Collection of methods that aren't a class
- A class is a set of identifiable behaviors you can "use" it is a good point to have it "behave over" all the variables you can use, including constant variables
- Rule blocks are classes they carry their environment around with them
 - o Rule blocks can be ignored by repeating what is defined when it when to do it
- Don't trying to merge behavior over "rule to" variable and only use "everything is a method rule" - do you spend to the method?