"Plan and Document". Before coding, project managers define plan. Write detailed documentation of phases of plan. Progress measured against the plan. Changes to project cannot be allowed in the documentation and provided to plan.

	Waterfall	Spiral	Agile
Description	1. Requirements analysis & specification 2. Architectural design 3. Implementation & integration 4. Verification 5. Operation & Maintenance Responsible to project manager	Combine Plan and Document with prototypes. Refine these plans & document with original requirements Code, develop plan & original requirements Documents are not such detailed All prototypes are not delivered and involve with the project	Customer first mind, Agile manifesto: Task focused Time bounded Cross functional Child (no subs) + maintainer, reviewer, tester Flow Release is in future continual requirements, Value by long-lived product/service/experience progress
Benefits	Good for bigger issues	Involve prototypes & variations	Shorter iterations

HTTP

- HTTP request to Rails is request method (GET, POST, etc.), Uniform Resource Identifier (URI), HTTP protocol version understood by the client, headers — extra info regarding transfer request
- Since HTTP is stateless, you use cookies in order to store data to help guide users through the pages of the app (e.g. authentication, shopping cart tracking, customization, etc.)

1. The Shared Architecture

Shared Debugging Architecture

MVC

- Controller receives HTTP request, calculates and views
- Any instance variable declared in the controller is made available to the view
- Interacting from a controller and gives each model the ability to perform CRUD
- Any instance variable declared in the model (that is not part of the database table) is not available outside of the model, so cannot be passed easily to the app
- In MVC, a URL maps to the view, not to the actions by a controller action.
- Rails: controller maps HTTP verb (URI) → to a controller action.

Ruby

- Ruby treats lists as arrays, spaces matter, there is no ; and lists is usually mixed
- Debugging: missing lists or dictionaries with commas
- Modules = Collection of methods that aren't a class
- A closure is the set of all variable bindings (name, etc. "foo") at a given point in time (i.e. "closes over" all the variables you can see, for binding variables in functions)
- Ruby blocks are closures: they carry their environment around with them.
  - Block: block is a chunk of code for representing what to do from where it refers to do it.
- Check typing: use message (returns name) "class" (returns class) and e.g. use "respondto?(:x)" — the name to pass to this method?"