Waterfall and Spiral

**Waterfall**

* Use waterfall when
  + Client is familiar with the domain and doesn’t expect requirements to be changed
  + The client needs multiple large components that fit very well with each other
  + Client is expecting to be extremely busy shortly after starting the project
  + *Waterfall is appropriate if the client has a good idea of what the requirements for a system will look like. Otherwise, it can be difficult to design accurate systems that reflect both the client's desires and real-world feasibility. For example, using newer technologies might lead to unexpected setbacks or workarounds that are not suited to Waterfall's linear, hand-off style. Waterfall processes often don't produce a working product until near the end of the development process; however, the longer a project drags on, the more likely the requirements are to change. The top-down design is beneficial if many components need to be well-integrated, as it can be architected wholly before beginning to program. Also, this style requires little feedback from clients once the requirements are solidified as theoretically the design documents should provide all the information necessary to complete and validate the system.*
* **Spiral**
  + The most important and least risky parts of the system are built and validated first
  + Spiral is intended to minimize risk by first delivering working versions of the essentials of the system, things that are unlikely to change. That way decisions that are more likely to pan out or not work as intended can be added on incrementally, while retaining a functional product.
* **Advantages of spiral over waterfall**
  + A working product is delivered sooner
  + Development is more risk-averse suitable for large and existing projects
  + Because Spiral is more iterative, it can produce a working product earlier in the production lifecycle, rather than waiting for the entire system to be coded and validated. However, the client still needs to have a solid idea of what the system will be like, so that the least risky or most fundamental parts of the system can be completed first.