

SDLC: Spiral, agile models

- biggest weakness of waterfall model is failure plan for revisiting previous phases
- spiral builds in the idea of revisiting stages: divides overall project into two or more cycles, each cycle goes through similar phases to waterfall
- nature of versioned software: in first cycle we build just the core of the product, in each later cycle we add and improve functionality

Spiral strengths, weaknesses

- Instead of trying to get the entire project right in one pass, we're dividing functionality into manageable chunks, and dealing with one chunk in each cycle
- Gets first version to client/market much faster than waterfall, but overhead of cycles might mean "final" version gets there slower
- Allows client to use/give feedback on each cycle, allowing us to change plans for future cycles as needs and conditions evolve

Agile model

- Waterfall and spiral models include a lot of process phases that are meant to improve quality, but which slow down delivery time
- Agile models suggest we use a team of good developers who ask the user what they (think they) want right now, and immediately begin coding
- Quickly get something functional to the client/user, who plays with it and requests further changes
- This rapid cycle of client input/coding allows product to evolve

Agile strengths

- Requirements are implicitly gathered as the project goes along, through user feedback on what they do/don't like
- User always has something working, and constantly sees improvements
- Developers interact directly with users, so the person building the code has a better idea of users real priorities (ideas aren't communicated through intermediate layers of documentation)
- If done well, can quickly evolve towards something highly usable

Agile weaknesses

- Inherently difficult to plan/manage timelines because there really aren't any
- Relies on developers making good, maintainable design decisions based on the early user feedback, otherwise code grows into unmaintainable mess
- Relies on frequent, high quality interaction between developers and users – need to get good feedback quickly and react to it appropriately
- Basically you need a good developer team and a good relationship with the client

Other models

- Many many many many different models
- Many many many different ways to pick what actions belong in what phases, and what to call them
- Most are hybrids, variants, or expansions on the big three of waterfall, spiral, agile
- Every organization and every team eventually evolves their own desired model (formally or not), important that everyone understands the model in use and their responsibilities in it