Agile

Agile

* Favour individual interactions over processes and tools
* Favouring working software over extensive documentation
* Favour customer interactions over contract negotiations
* Favour agility over planning
* Failures are a good thing because
  + Agile iterations, and therefore failures, are small, and failures help you understand the system better
  + Given the small size of each piece in an iterative process, and emphasis on systems being built gradually, failures are not too expensive (in terms of time or money spent). They often pave the path to better design through an improved understanding of what approaches will or will not work, and for what reasons.
* XP Analogy:
  + Playing scrabble as every new word built when adding letters to the board must be valid.
  + XP maintains that all systems should always be buildable. This is like how all words formed on a Scrabble board at any given time must be valid dictionary words; players are not allowed to add letters that would construct illegal words, just as developers should not commit any code that will break the build. Moreover, XP advocates starting small and refactoring, not architecting the entire system upfront, linearly adding more, nor building each piece separately.
* Extreme Programming **Principles:**
  1. Communication means transparency and staying on the same page with all parties involved.
  2. Feedback is the stage where you solicit the viewpoints of various stakeholders.
  3. Courage is the idea that failed experiments still hold information.
  4. Respect emphasises good documentation and commit practices.
* Steps of TDD:
  + 1. Add tests
    2. Run tests
    3. Write code and run tests
    4. Refactor code
* **Scrum:** Scrum roles have 3 high level stakeholders
  1. Technical team: Cross functional and small
     + The team should be composed of people in various roles while staying reasonably small
     + Helps accumulate a wide variety of overhead without introducing a large amount of overhead
  2. Product owner represents the interests of the client and explains and motivates what kinds of features the product should have in each spring as well as overall.
  3. The scrum master should be facilitating scrum, resolving conflicts and keeping the team on track.
* Product backlog: List of features to be included in the finished product.