XP - Planning

- Planning in XP
  - Need to manage a Product Backlog which is a prioritized list of features
  - Product owner keeps it prioritizes and controls how it changes over time
  - We will generate the initial backlog with the Planning Game (from XP)
- Selecting what we will build
  - Goal: to generate and prioritize required functionality
  - Strategy: invest as little as possible to put the most valuable functionality into production as quickly as possible paying attention to programming and design strategies that reduce risk.

Planning Game

- Players:
  - Customer
  - Development
- Pieces: Story Cards
  - date
  - type of activity: new, fix, enhancement, or test
  - priority (user and development)
  - description
  - risk
  - estimate
- Three Phases:
  - Exploration
    - write a story (customer)
    - estimate a story (development)
    - split a story (either: it might be too big to estimate or parts might have different priorities)

- Commitment
  - sort by value (business) into three piles
    - necessary for the system to function
    - less essential, but of significant business value
    - nice to have
  - sort by risk (development) into three piles
    - can be estimated precisely
    - can estimate with reasonable confidence
    - cannot estimate
  - set velocity (development) (how fast the team can program in an appropriate metric/calendar month)
  - choose scope (business)
    - must select stories so they fit in the project velocity
- Steering
  - scale back velocity (development)
    - can ask business to narrow stories in the current release
  - new story (business) in mid-release
    - can add a new story if it removes undeveloped stories of equal estimates
  - reestimate (development)
    - if plan is no longer accurate, development can reestimate all undeveloped stories and reset the velocity.
- Phases are NOT sequential - go back and forth between them frequently

Iteration Planning Game

Similar to the Planning Game, but used to plan the individual tasks/assignments that will build the functionality committed to in one iteration.

- Pieces - Task Cards
  - Description of something we need to do/build
- Exploration Phase Moves
  - turn the stories into tasks (create task)
  - split a task/combine tasks - want each task to be estimated to be between 0.5 to 3 days
- Commitment Phase Moves
  - accept a task (by an individual)
  - estimate a task (responsible individual)
    - can be conditional on getting help from someone else
    - ignore pair programming - it shows up in calculation of project velocity that also accounts for meetings etc.
  - set load factors (each individual)
    - percentage of time you will spend actually developing
    - comes from historic metrics (never higher than 0.5 and rarely higher than 0.3), but can account for planned vacation/holidays
  - balancing (all)
    - add up your time and make sure no one is overcommitted
- Phases are NOT sequential - go back and forth between them frequently
- Differences from Planning Game
  - individual accepts a task before estimating it
  - there may be tasks that are not related to needs of the customer (tool development, major refactoring, etc.)