

Two essential roles:

- Product Owner: Focuses on what needs to be done
 - Reviews team's work
 - Accepts
 - Rejects
 - Interact with stakeholders
- Scrum Master: Keeps team working at sustainable pace
 - Acts as spokesperson
 - Devise charts and boards to organize project
 - Helps remove roadblocks

Establish scrum team:

- Enable daily communication for team members
- Establish team norms early
 - Working relationship
 - Conflict resolution
 - Project consensus
 - How to hold team accountable

Product Vision

- Established by product owner
- Destination = Minimum Viable Product
 - Develop the smallest possible site to reduce scope creep
- Once vision is in hand, you can decompose the vision
- To create product vision
 - Establish themes (goal functionality of app)
 - Break themes into features (different components of page)
 - Establish which features are musts, and which features could be optional

Writing user stories:

- User story: tactical level of work that can be delivered quickly (written by PO)
- INVEST - story development acronym
 - Independent (can be delivered separately from other stories)
 - Negotiable (Until it is submitted for work, it can be modified)
 - Valuable (delivers value to PO and stakeholders)
 - Estimable (descriptive enough so you know what has to be done to finish it)
 - Small (small enough to be completed in one sprint)
 - Testable (must provide enough information to be able to test)
- Functional User Story Format (functional = serves a function for end user)
 - <user role>, I want
 - <user requirement> so that
 - <desired benefit>
 - Example: as a <mobile customer>, I want

- <to create a profile> so that
 - <future orders are faster to place>
- Non functional story (work that supports user without directly benefiting them)
 - Example: As a developer, I want to upgrade the database software to the latest version so that we have a portable product
- Whenever a user story is written, an AC (Acceptance Criteria) is written
 - Should be as explicit as possible so everyone knows what done means
 - Example:
 - Customer name is captured and saved
 - Customer email address is captured and saved
 - Customer phone number is captured and saved
 - Customer password is captured and saved
 - Additionally (input is valid)

Workflow:

Product Backlog:

- Themes and features are broken down into user stories
- Stories are estimated using points
 - Relative estimation: We use relative estimation to get a rough size of our work by comparing user stories to each other
 - Not a commitment
 - Actual estimation: Story points convey relative size as an actual value
 - Story points are done using poker planning

Create Team Boundaries:

- Definition of done: states the minimum requirements for all team stories
 - More general than AC
 - Should apply to everything in your backlog
 - Example: it has been code reviewed and all errors fixed or it has been tested in a pre-development environment
- Backlog prioritization (Backlog grooming...done by PO)
 - Stories are sequenced or prioritized by value
 - You want the most valuable stuff delivered first
- Sprint cadence
 - Can be from 1 to 4 weeks in length, with preference to shortest time scale

Create Roadmap and release plan

- Roadmap: guide showing when themes will be worked on during a time frame
- Release plan: high level plan connecting roadmap to sprints
 - At the end of every sprint, you want to have a fully functional program, however, you don't have to release these programs to stakeholders until you want.
 - The release plan should specify what sprints are in which release, what stories are in each sprint, and how many points each story is worth.