The Rules of Extreme Programming

Planning

- [User stories](#) are written.
- [Release planning](#) creates the [release schedule](#).
- Make frequent [small releases](#).
- The project is divided into [iterations](#).
- [Iteration planning](#) starts each iteration.

Managing

- Give the team a dedicated [open work space](#).
- Set a [sustainable pace](#).
- A [stand up meeting](#) starts each day.
- The [Project Velocity](#) is measured.
- [Move people around](#).
- [Fix XP](#) when it breaks.

Designing

- [Simplicity](#).
- Choose a [system metaphor](#).
- Use [CRC cards](#) for design sessions.
- Create [spike solution](#)s to reduce risk.
- No functionality is [added early](#).
- [Refactor](#) whenever and wherever possible.

Extreme Programming Project

Coding

- The customer is [always available](#).
- Code must be written to agreed [standards](#).
- Code the [unit test first](#).
- All production code is [pair programmed](#).
- Only one pair [integrates code at a time](#).
- [Integrate often](#).
- Set up a dedicated [integration computer](#).
- Use [collective ownership](#).

Testing

- All code must have [unit tests](#).
- All code must pass all [unit tests](#) before it can be released.
- When [a bug is found](#) tests are created.
- [Acceptance tests](#) are run often and the score is published.

Let's review the values of Extreme Programming (XP) next.