**Agile Methodologies**

**Extreme programming:**

Extreme programming is a software development methodology that's part of what's collectively known as agile methodologies. XP is built upon values, principles, and practices, and its goal is to allow small to mid-sized teams to produce high-quality software and adapt to evolving and changing requirements.And it describes extreme programming as a software-development discipline that organizes people to produce higher-quality software more productively. XP attempts to reduce the cost of changes in requirements by having multiple short development cycles, rather than a long one.

1. **Planning,**the first stage, is when the customer meets the development team and presents the [requirements](https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/) in the form of [user stories](https://www.altexsoft.com/blog/user-stories/) to describe the desired result. The team then [estimates](https://www.altexsoft.com/blog/story-points/) the stories and creates a release plan broken down into iterations needed to cover the required functionality part after part. If one or more of the stories can’t be estimated, so-called spikes can be introduced which means that further research is needed.
2. **Designing**is actually a part of the planning process, but can be set apart to emphasize its importance. It’s related to one of the main XP values that we’ll discuss below — simplicity. A good design brings logic and structure to the system and allows to avoid unnecessary complexities and redundancies.
3. **Coding**is the phase during which the actual code is created by implementing specific XP practices such as coding standards, pair programming, continuous integration, and collective code ownership (the entire list is described below).
4. **Testing**is the core of extreme programming. It is the regular activity that involves both unit tests ([automated testing](https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management/) to determine if the developed feature works properly) and acceptance tests (customer testing to verify that the overall system is created according to the initial requirements).
5. **Listening**is all about constant communication and feedback. The customers and project managers are involved to describe the business logic and value that is expected.



