Extreme Programming (XP) and Scrum

Learning Objectives

After completing this topic, you should be able to

- identify the main principles of XP
- identify the main tenets of Scrum

1. Extreme Programming

Extreme Programming – also known as XP – is one of the best-known agile methodologies. It provides a programmer-centric model that's focused on the ongoing, rapid delivery of small releases of software.

With XP, the interval between software releases is usually 30 to 180 days. This period is further broken down into iterations, each lasting between one and four weeks.

Each iteration results in production-ready code – working code that has been fully tested and can be integrated in a release.

To achieve the required level of efficiency, all members of a small, self-directed team of programmers collaborate closely with one another and with an on-site customer.

The team sets its own coding standards and handles the planning of coding tasks based on a set of customer user stories – each of which describes a single functionality the customer requires of the software product.

The focus of design is on simplicity, and ongoing testing, customer feedback, and refactoring are integrated in the development process.

Note

Refactoring refers to the re-design of a program or of code that has already been implemented, typically to improve its performance.

XP is associated with four values – communication, simplicity, feedback, and courage. These overarching values must guide an XP team if it's to succeed in meeting customer needs quickly, efficiently, and with minimal external direction.

At a more concrete level, XP is associated with a specific set of 13 principles.

Planning game

In the XP model, what's referred to as the planning game is the main planning process that simplifies both release and iteration planning.

Small releases

An emphasis in XP is on releasing software products as quickly as possible, to reduce time to market and enable timely feedback.

System metaphor

An XP project team should keep a simple statement about what a system is meant to do and how it's meant to do it. This statement should take the form of a metaphor, which clarifies what's required using easily recognizable objects or familiar concepts. For example, a shopping cart is a popular metaphor for...
the functionality that enables online shoppers to select items to purchase and to add these to a repository. The metaphor should be sufficiently broad that it won't change even as user stories – which identify requirements in more detail – are modified. It ensures team members stay focused on a clear, high-level objective.

**Simple design**

A core XP principle is simplicity – always use the simplest possible design to provide the functionality described in a user story, and don't include functionality that isn't specifically required. This principle is based on the need for rapid releases of useful software. Business needs can change quickly, so it's critical to focus on meeting existing needs efficiently, and not to waste time on developing functions that may never be used.

**Ongoing testing**

Ongoing testing that's fully integrated in the development process is central to the XP model. Programmers write unit tests even before they start writing code. This informs and improves the design and development processes. Testing then continues throughout development, and should be automated where possible. This means that once an iteration completes, code has been fully tested and is ready for integration in a release.

**Refactor**

Rather than designing a complete system and then developing it, an XP team refactors working code on an ongoing basis. In other words, the team modifies system design progressively and when needed, as more is learned. For example, refactoring may involve completely changing the structure of a class – or even of a full subsystem – that has already been written and tested, to improve overall system design.

**Pair programming**

The XP model encourages pair programming, in which two or even three programmers work at a single workstation. This enables ongoing discussion and resolution of design, testing, and programming concerns, which can result in better quality code.

During release planning, the programming team gathers customer requirements in the form of user stories. It assigns each story a number of points, based on the estimated time and work required to implement it. The team also determines the total number of story points it can include in a release, given available resources and a specified release date.

The customer then prioritizes the user stories, selecting a set that doesn't exceed the maximum number of story points. If the resulting set is judged by the customer to provide business value, the project proceeds.

During iteration planning, the team translates a subset of the selected user stories into corresponding coding and testing tasks. Members of the team then assign these tasks among themselves.

The XP principles determine how programming occurs.

**Collective code ownership**

Any member of an XP team is entitled to alter any system code, at any time. So although each programmer focuses on particular coding tasks, everyone on the team shares ownership and responsibility for the full system code.

**Continuous integration of code**

The XP model relies on constant rebuilding and testing of the full code base for a system. So although programmers continually add new working code they've developed, the full code base is never far from being in a production-ready state.

**Sustainable pace**

XP team members are encouraged not to work more than 40 hours per week. Instead the emphasis is on maintaining a sustainable pace, with delivery that's consistent and reliable.

**Shared workspace**

XP team members are encouraged to learn about all relevant technologies and to be familiar with all system requirements, rather than focusing only on specialized areas of development. Also, all members of the team typically work in the same room. This encourages full collaboration and a sense of shared ownership.

**Coding standards**

An XP team is responsible for defining and implementing a consistent set of coding standards. This helps
ensure that everyone on the team writes code that can be easily understood by other team members.

**On-site customers**

Constant, direct access to a customer – who sits in the same workspace as programmers and acts as a team member – enables fast software development. Requirements can be checked or modified, and approval can be obtained immediately.

All the XP practices support one another. Although many agile teams implement some of these practices, adopting them in full requires a high degree of discipline, teamwork, and skill.

---

**Graphic**

*Description of diagram showing how the XP practices support one another:*

The shared workspace of programmers and the on-site customer support the planning game and pair programming. Pair programming supports refactoring, ongoing testing, and continuous integration of code. Continuous integration of code supports collective code ownership and coding standards. The planning game supports system metaphors, simple design, and small releases, which in turn enable a sustainable pace of development.

*Description ends.*

---

**Question**

Which are key XP principles?

**Options:**

1. Collective code ownership
2. Thorough investigation of all potential user requirements prior to initial software planning
3. Pair programming, with two or more programmers sharing each workstation
4. Continuous code integration
5. Comprehensive post-production testing

**Answer**

**Option 1:** Correct. One of the main XP practices is collective code ownership. Every member of an XP programming team has full access to all the code for a release and can make any necessary changes to it.

**Option 2:** Incorrect. The focus in XP is on rapidly delivering small releases of software that meet specific, current needs, rather than on delivering software that attempts to meet a wide range of anticipated or potential needs.

**Option 3:** Correct. The XP model encourages pair programming because this enables real-time discussion and collaboration between programmers. Concerns can be addressed as they arise, resulting in better-quality code.

**Option 4:** Correct. Continuous integration is one of the core XP practices. During a development iteration, members of an XP programming team continually add to, rebuild, and retest a working code base.

**Option 5:** Incorrect. XP emphasizes ongoing testing. Unit tests are developed before code is written, and testing and refactoring continue as part of the development process. This makes comprehensive post-production testing unnecessary.

**Correct answer(s):**
1. Collective code ownership
2. Pair programming, with two or more programmers sharing each workstation
3. Continuous code integration

Because one focus of XP is the collaboration and equal participation of all team members, an XP team may typically include only a few role designations – customer, development coach, business coach, developer, and tracker. However, it can include all of the listed roles.

**Customer**
The customer creates and prioritizes user stories, as well as providing ongoing feedback to the team.

**Development coach**
The development coach monitors the development process to identify potential problems, and guides or mentors team members on the use of XP techniques.

**Business coach**
A business coach can be used to represent and guide the customers. They will have a good understanding of the business requirements of the project and of XP. They will help the customers to write user stories and acceptance tests.

**Developer**
Developers are the programmers who plan, write, and test code based on the customer's user stories.

**Business Analyst/Systems Engineer**
These act as subject matter experts for XP development teams. They can assist customers with technical information and represent customers’ business needs if required.

**Tracker**
The tracker monitors the team's progress to ensure it stays on schedule, calling for adjustments or a redistribution of workloads if necessary.

**Tester**
Specific testers can be used to develop the test plan and work with customers on acceptance tests. They need to ensure that user stories are testable and they run acceptance tests.

**Big Boss**
The XP Big Boss is responsible for assembling the XP development team, and looking after workspace and equipment requirements. They will represent the team, when required, to those outside the XP team.

---

**Question**

Match roles in an XP team to their key functions.

**Options:**
A. Customer  
B. Tracker  
C. Developer  
D. Development coach

**Targets:**
1. Specifies the functionality software must provide  
2. Monitors the team's progress  
3. Implements and tests code  
4. Mentors team members on applying XP techniques

**Answer**
The customer supplies the requirements software must meet in the form of a prioritized set of user stories.

A tracker monitors the team's progress to ensure it stays on schedule, calling for adjustments if necessary.

A developer plans, writes, and tests code to implement the customer's requirements.

A development coach guides team members through the XP development process.

Correct answer(s):
- Target 1 = Option A
- Target 2 = Option B
- Target 3 = Option C
- Target 4 = Option D

2. Scrum

Whereas XP focuses on programming practices, the Scrum methodology focuses on agile project management. It's often used together with XP or with other agile methodologies.

Scrum takes its name from the scrum in rugby, in which players form a tight knot and together execute a plan for passing the ball in a desired direction.

The Scrum methodology is based on short daily meetings – known as scrums – in which all team members collaborate on managing a development project.

Note

*Each scrum meeting should last between 10 and 15 minutes.*

In Scrum meetings, each team member answers three questions:

- What work have I completed since the last scrum, and why?
- What do I plan on completing between now and the next scrum?
- Do I have any roadblocks or problems that the team can help overcome?

Also borrowed from rugby is the concept of a sprint. In Scrum, each sprint (iteration) is a fixed-length development period with a clear goal, consisting of an agreed set of work items to implement.

Each successive sprint builds on the last, and planning occurs between the sprints.

Note

*Typically, one sprint lasts from two weeks to a month.*
A Scrum team uses three types of backlogs to plan and manage a development project:

---

**a sprint backlog**

A sprint backlog details the work items that team members have agreed to complete in a given 30-day sprint. Once a sprint starts, no changes should be allowed to interfere with its goal.

**a release backlog, and**

A release backlog details all the work items that must be completed before a software product can be released.

**a product backlog**

A product backlog details all changes and ideas for implementation that don't directly support the agreed goal for the current sprint. This backlog then informs the planning for subsequent sprints.

---

To track its progress in completing the agreed work items during a sprint, a Scrum team uses what's known as a *burndown chart*.

In the chart, the straight line indicates the ideal workflow, in which work items are completed in a completely steady, uniform manner.

A second line, which is updated daily, represents the team's actual progress. This makes it clear when adjustments are needed to bring development back on schedule.

Scrum emphasizes a management process based on empirical control, with planning that's iterative and incremental.

In addition, a key principle is that a Scrum team must be self-organizing. Team members organize and manage their own work, based on the goals for a sprint and on daily scrum meetings.

Core values associated with the success of the team include commitment, focus, openness, respect, and courage.

The ideal Scrum team includes four to nine members.

However, the methodology is scalable. If multiple teams are involved in a project, each team’s daily scrum meeting can be followed by a second scrum. In the second scrum, a representative from each team collaborates to ensure that all the teams’ efforts are coordinated.

A focus in Scrum is on the equal participation of all team members. Accordingly, apart from the team itself, only two Scrum roles are specifically defined:

**Product Owner and**

The Product Owner represents the interests of the customer or end user. This person is also responsible for liaising between the team and the customer.

**Scrum Master**

The Scrum Master facilitates and reports on the work of the team, focuses the team on the highest-priority tasks, and assists in overcoming any roadblocks. Certification training at three levels is available for Scrum Masters.

---

**Question**

Which are main tenets of the Scrum methodology?

**Options:**

1. Project planning is iterative and incremental
2. Development work is timeboxed and sprint-driven
3. Project team members are self-organizing
4. A trained Scrum Master assumes full responsibility for assigning tasks and managing workflow
5. Programmers work in pairs

**Answer**

**Option 1:** Correct. A Scrum team plans the work items to complete in each fixed-length sprint. Planning occurs again before the next sprint, and successive sprints build on one another.

**Option 2:** Correct. In a Scrum team, particular work items are assigned to each of a series of fixed-length sprints. Once a sprint begins, the team focuses only on the goal of that sprint.

**Option 3:** Correct. All members of a Scrum team organize and manage their own work, based on the goals for a sprint and on daily scrum meetings.

**Option 4:** Incorrect. The Scrum Master focuses team members on high-priority tasks and facilitates their work – but all team members participate in both managing and completing the required work.

**Option 5:** Incorrect. Pair programming is a concept associated with XP. Scrum focuses on project management, rather than on programming practices. However, XP principles such as pair programming may be combined with an implementation of Scrum.

**Correct answer(s):**
1. Project planning is iterative and incremental
2. Development work is timeboxed and sprint-driven
3. Project team members are self-organizing

**Question**

Match each role on a Scrum team to the corresponding description.

**Options:**
A. Product Owner
B. Scrum Master
C. General team member

**Targets:**
1. Represents the customer's interests
2. Guides and facilitates the development process
3. Participates in managing and completing the work designated for each sprint

**Answer**

*The Product Owner represents the interests of the customer or end user of the software being developed.*

*A Scrum Master focuses the efforts of the team, assists in overcoming roadblocks, and reports on team progress.*

*All team members participate in planning and completing the work items designated for each 30-day sprint.*

**Correct answer(s):**

Target 1 = Option A
Summary

Extreme programming, or XP, is a programmer-centric methodology that focuses on the ongoing, rapid delivery of small releases of software. Testing, refactoring, and responding to customer feedback are all fully integrated in the development process, and very close collaboration among team members is required.

Scrum is a methodology for managing agile development projects. It involves iterative, incremental planning, which occurs between spurts of development work driven by clear goals. Team members manage their own work and track progress through daily meetings.