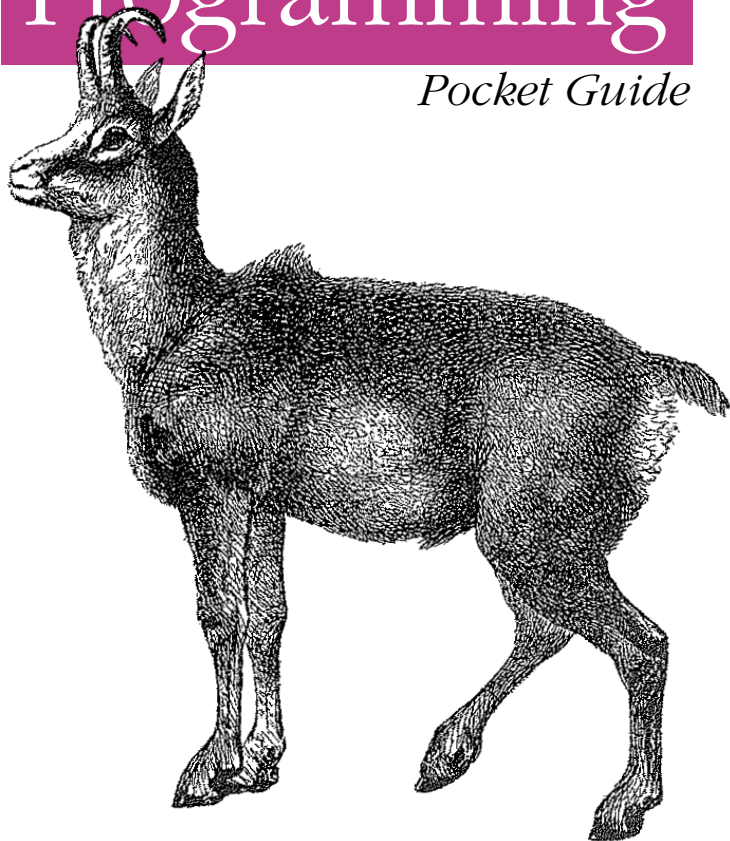


*Team-Based Software Development*

# Extreme Programming

*Pocket Guide*



**O'REILLY®**

*chromatic*

---

# **Extreme Programming**

*Pocket Guide*

---

# Extreme Programming

## *Pocket Guide*

*chromatic*

**O'REILLY®**

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

---

# Roles in Extreme Programming

Every XP project has several different roles, each with its own unique rights and responsibilities. XP attempts to improve communication between customers and developers. It accomplishes this by sharply dividing the work between the two. If you want to get any work done, you'll have to talk to each other!

XP gives developers authority to make technical decisions. This is their area of expertise. XP gives the customer authority to make business decisions. This is his area of expertise. These spheres of influence complement each other. Following these clear lines of authority will improve your chances of success.

## The Customer

The customer drives the project. He defines the project and sets its goals. The more accurate his work and the more frequent his involvement, the greater the chances the project will succeed.

The customer makes business decisions. His rights and responsibilities stem from his business knowledge. He has the authority to set the project's goals and features. He must answer the questions: "*What should this feature do?*", "*How will we know when it is done?*", "*How much can we spend?*", and "*When shall we start working on it?*"

The customer works closely with the developers. He writes story cards to explain and to schedule the desired features.

This answers the *what* question. He participates in the planning game to schedule stories for the next iteration. This answers the questions of *when* and *how much*. He creates and runs acceptance tests, with developer assistance, to verify that features are complete. This answers the *is it done* question.

The customer represents the end user. In a corporate, in-house project, he may be an end user. In other situations, he serves as a proxy for end users. He identifies the features users really need from their perspective. The developers will take care of the technical perspective.

The customer also represents the business interests that are paying for the project. His goal is to maximize their investment. At any point, the software should contain the most valuable features that could have been scheduled based on the available knowledge.

An XP customer relies on several pieces of information. He must understand the business problem to be solved even as it changes over time. Is a story more or less valuable now than when it was identified? Can a story be delayed, deferred, or simplified? He must be able to evaluate the project at any time. Which features are complete? How well do the completed features conform to his stories? He must understand the technical implications of a story that affect its value and its risk. Is it better to schedule a story suggested by developers before a story he wrote on his own?

XP always refers to the customer as a single person. Even if the customer is a proxy for an actual investor or far-off end users, he must speak with one voice. He holds a position of authority, with the right to say what must be done.

## Customer Rights

XP recognizes several customer rights:

- *To maximize his investment*, by choosing stories to schedule in the current iteration. (See “Business Practice 2: Play the Planning Game” in Part II.)

- *To change the scope of the project to deal with schedule changes*, by selecting stories to add to or remove from an iteration if its estimates prove incorrect. (See “Business Practice 1: Add a Customer to the Team” in Part II.)
- *To determine which features to implement next*, by selecting story cards in the iteration planning meeting. (See “Business Practice 2: Play the Planning Game” in Part II.)
- *To measure the progress of the project at any time*, by running the acceptance tests. (See “Developer Practice 1: Adopt Test-Driven Development” in Part II.)
- *To stop the project at any time without losing his investment*, by keeping the software in a releasable state and continually scheduling the most worthwhile features. (See “Developer Practice 4: Integrate Continually” and “Business Practice 2: Play the Planning Game,” both in Part II.)

## Customer Responsibilities

XP identifies several customer responsibilities:

- *To trust the developers’ technical decisions*, because they understand technology. (See “Business Practice 2: Play the Planning Game” in Part II.)
- *To analyze risk correctly*, weighing the stories against each other accurately. (See “Business Practice 2: Play the Planning Game” in Part II.)
- *To choose the stories with maximum value*, scheduling the most valuable stories that could possibly fit into the next iteration. (See “Business Practice 2: Play the Planning Game” in Part II.)
- *To provide precise stories*, enabling the developers to produce comprehensive task cards and accurate estimates. (See “Story Cards” in Part IV.)
- *To work within the team*, providing guidance and receiving feedback as quickly and accurately as possible. (See “Business Practice 1: Add a Customer to the Team” in Part II.)

# The Developer

Most XP practices concern the day-to-day work of producing code. This is the job of the developer: to turn customer stories into working code.

The developer role in planning and implementing features depends on knowing and understanding technical issues. Developers create and maintain the system as it evolves. They must answer the questions: “*How will we implement it?*”, “*How long will it take?*”, and “*What are the risks?*”

Developers work with the customer to understand his stories. From a story, the developers decide its implementation. The developers then estimate the amount of work each story will take, based on the implementation decisions and their experience on the project so far. These estimates help the customer to schedule the most valuable work for the next iteration by answering the question of *how long*.

While creating task cards from the story cards or implementing tasks during the programming cycle, developers may identify features that depend on other features. They may also find risky features that use new technology, are poorly understood, or are otherwise complicated. Developers raise these issues with the customer, who considers them while making the schedule. In practice, these risks are rare—practicing simplicity reduces them.

## Developer Rights

XP recognizes several developer rights:

- *To estimate their own work*, by giving developers authority over technical decisions. (See “Business Practice 2: Play the Planning Game” in Part II.)
- *To work a sensible and predictable schedule*, by scheduling only the amount of work that can reasonably be done. (See “Business Practice 4: Work at a Sustainable Pace” in Part II.)

- *To produce code that meets the customer's needs*, by focusing on testing, refactoring, and customer communication. (See “Developer Practice 1: Adopt Test-Driven Development” and “Coding Practice 2: Refactor Mercilessly,” both in Part II.)
- *To avoid the need to make business decisions*, by allowing the customer to make them. (See “Business Practice 1: Add a Customer to the Team” in Part II.)

## Developer Responsibilities

XP expects several developer responsibilities:

- *To follow the team's guidelines*, so that the system is as simple, as well-tested, and as agile as possible. (See Part VI.)
- *To implement only what is necessary*, to keep the project as simple and as valuable as possible for the customer. (See Part VI.)
- *To communicate constantly with the customer*, to understand his concerns and to help him to make accurate scheduling decisions. (See “Business Practice 1: Add a Customer to the Team” in Part II.)

## Supplementary Roles

XP discusses two other roles. They might not be present in a formal capacity on every team.

### The Tracker

The tracker keeps track of the schedule. XP tracks a few metrics. The most important is team velocity, which is the ratio of ideal time estimated for tasks to the actual time spent implementing them. Other important data may include any changes in velocity, the amount of overtime worked, and the ratio of passing tests to failing tests.

All of these numbers measure progress and the rate of progress. They help determine if the project is on schedule for the iteration. They can signal behavioral changes that may affect the schedule. Looking at the numbers alone rarely gives the whole picture; anomalies should be brought before the whole team for analysis during the stand-up meeting (see “The Iteration” in Part III.)

To measure velocity within the iteration, every day or two, the tracker asks each developer how many tasks she has completed. This is best done in person, as informally and comfortably as possible. Honesty is vital on the part of developers, and the tracker should be nonjudgmental. This may be a manager or a trusted developer. Regularly tracking progress helps the team adjust to its ebb and flow of work.

## The Coach

Some XP projects have a coach who guides and mentors the team. This can be helpful when adopting XP. His position is one of respect—he leads by example.

XP can be difficult to apply consistently. Though many of its practices are common sense, the skills they require take time to develop. There are also occasional obstacles and subtleties that require the wisdom of a master. The coach’s main virtue is his experience.

The coach guides his team to understand XP and software development. Sometimes he teaches directly. Sometimes he rolls up his sleeves and teaches by doing. He may suggest changes in how a practice is implemented, offer ideas to solve a thorny technical problem, or serve as an intermediary between the team and other management.