On-Site Customers
On-site customers—often just called customers—are responsible for defining the software the team builds. The rest of the team can and should contribute suggestions and ideas, but the customers are ultimately responsible for determining what stakeholders find valuable.

On-site customers may or may not be real customers, depending on the type of project. Regardless, customers are responsible for refining their plans by soliciting feedback from real customers and other stakeholders. One of the venues for this feedback is the weekly iteration demo, which customers lead.

In addition to planning, customers are responsible for providing programmers with requirements details upon request. XP uses requirements documents only as memory aids for customers. Customers themselves act as living requirements documents, researching information in time for programmer use and providing it as needed. Customers also help communicate requirements by creating mock-ups, reviewing work in progress, and creating detailed customer tests that clarify complex business rules. The entire team must sit together for this communication to take place effectively.

The product manager also known as product owner
The product manager has only one job on an XP project, but it's a unique. That job is to maintain and promote the product vision. In practice, this means documenting the vision, sharing it with stakeholders, incorporating feedback, generating features and stories, setting priorities for release planning, providing direction for the team's on-site customers, reviewing work in progress, leading iteration demos, involving real customers, and dealing with organizational politics.

In addition to maintaining and promoting the product vision, product managers are also often responsible for ensuring a successful deployment of the product to market. That may mean advertising and promotion, setting up training, and so forth.

What are reponsibilites of the product manager in a college app?
Product manager responsibilities:

- Explain why the app exists (product vision)
- Coordinate release on Play Store
- Promote it through college notices
- Conduct demo sessions for students
- Collect feedback and improve features

Coding alone ≠ success.

If nobody knows about your project or knows how to use it, is it successful?

1. Define and Maintain the Product Vision

- Clearly communicates:
    - What the product is
    - Who it is for
    - Why it is valuable
- Keeps the team aligned with business goals
- Acts as a bridge between stakeholders and developers

Without a clear product vision, Agile teams may deliver features but miss real value.

2. Decide What to Build (Prioritization)

- Maintains the **product backlog**
- Prioritizes features based on:
  - Business value
  - Risk
  - User feedback
- Decides what to do now vs later

In XP, developers decide how to build, but product managers decide what to build.

3. Work Closely with the Team (On-Site Customer Role)

In XP, the product manager often plays or supports the **on-site customer** role:

- Available daily to answer questions
- Clarifies requirements
- Provides quick feedback

This reduces delays and misunderstandings.

4. Balance Stakeholder Needs

- Communicates with:
  - Customers
  - Executives
  - Marketing
  - Support teams
- Filters requests to protect the team from chaos
- Makes tough trade-offs

Not every request can be implemented.

5. Support Release Planning and Deployment

- Product managers help decide release scope
- Ensure the product is ready for market deployment
- Coordinate:
  - Promotion
  - Training
  - User communication

Success is measured by **real-world impact**, not just completed features.
“If developers build fast but build the wrong thing, who is responsible?” ---- product manager.

Product Manager	**Project Manager**
Focuses on *what* to build	Focuses on *how* work flows
Business value	Process & coordination
Feature priorities	Schedule & resources
Stakeholders & users	Management & team

Product Manager as a “mini-CEO of the product”
Conceptually, a product manager can be compared to a CEO (Chief Executive Officer) because they are responsible for product vision and business value, while a project manager can be compared to a COO (Chief Operating Officer) because they focus on execution and coordination. However, in Agile development, these roles operate collaboratively rather than hierarchically, and teams are self-organizing.

Domain experts (aka subject matter experts)

who are domain experts in the college app?

Most software operates in a particular industry, such as finance, that has its own specialized rules for doing business. To succeed in that industry, the software must implement those rules faithfully and exactly. These rules are domain rules , and knowledge of these rules is domain knowledge.

1. Provide Domain Knowledge

They explain:

- Business processes
- Terminology
- Rules and exceptions
- Edge cases

2. Clarify Requirements & Examples

Instead of long documents, domain experts:

- Provide examples
- Help define acceptance criteria
- Review customer tests

In XP, requirements are often expressed as examples rather than specifications.

3. Validate Software Behavior

Domain experts answer questions like:

- “Does this output make sense?”
- “Would users do this?”
- “What happens in unusual situations?”

This prevents costly rework later.

“Who understands the problem better—the programmer or the person who works with it daily?”

Interaction designers

The user interface is the public face of the product. For many users, the UI is the product. They judge the product's quality solely on their perception of the UI. Interaction designers help define the product UI. Their job focuses on understanding users, their needs, and how they will interact with the product.

Interaction Designer	**UI Designer**
Focuses on behavior & flow	Focuses on look & feel
Designs how users use system	Designs visual appearance
User actions & responses	Colors, fonts, layouts

You may not have a professional interaction designer on staff. Some companies fill this role with a graphic designer, the product manager, or a programmer.

If software works but is hard to use, is it successful?

Programmers

A great product vision requires solid execution. The bulk of the XP team consists of software developers in a variety of specialties. Each of these developers contributes directly to creating working code. To emphasize this, XP calls all developers programmers. If the customers' job is to maximize the value of the product, then the programmers' job is to minimize its cost. Programmers are responsible for finding the most effective way of delivering the stories in the plan. To this end, programmers provide effort estimates, suggest alternatives, and help customers create an achievable plan by playing the planning game.

Include between 4 and 10 programmers. In addition to the usual range of expertise, be sure to include at least one senior programmer, designer, or architect who has significant design experience and is comfortable working in a hands-on coding environment. This will help the team succeed at XP's incremental design and architecture.

1. Write Clean, Working Code

2. Practice Test-Driven Development (TDD)

3. Design Through Refactoring

- Improve design continuously
- Remove duplication
- Keep the system flexible (Design evolves incrementally rather than being fixed upfront.)

4. Work in Pairs (Pair Programming)

- Two programmers work at one workstation
- One writes code, the other reviews and thinks ahead
- Roles switch frequently

This improves code quality and knowledge sharing.

5. Collaborate with the Whole Team

- Discuss requirements with product managers and domain experts
- Clarify examples with testers
- Participate in planning and retrospectives

“Who ensures the system can change tomorrow without breaking today?”

Theory of Constraints says,in part, that every system has a single constraint that determines the overall throughput of the system. This book assumes that programmers are the constraint on your team. Regardless of how much work testers and customers do, many software teams can only complete their projects as quickly as the programmers can program them. If the rest of the team outpaces the programmers, the work piles up, falls out of date and needs reworking, and slows the programmers further.

Testers

As with the customer ratio, I arrived at the one-to-four tester-to-programmer ratio through trial and error. Successful teams I’ve worked with have had ratios as low as one tester for every six programmers, and some XP teams have no testers at all.

Testers help XP teams produce quality results from the beginning. Testers apply their critical thinking skills to help customers consider all possibilities when envisioning the product. They help customers identify holes in the requirements and assist in customer testing. Testers also act as technical investigators for the team. They use exploratory testing to help the team identify whether it is successfully preventing bugs from reaching finished code. Testers also provide information about the software’s nonfunctional characteristics, such as performance, scalability, and stability, by using both exploratory testing and long-running automated tests.

Rather than relying on testers to find bugs for programmers to fix, the team should produce nearly bug-free code on their own.

Coaches

XP teams self-organize , which means each member of the team figures out how he can best help the team move forward at any given moment. XP teams eschew (avoid using) traditional management roles. Instead, XP leaders lead by example, helping the team reach its potential rather than creating jobs and assigning tasks. To emphasize this difference, XP leaders are called coaches . Over time, as the team gains experience and self-organizes, explicit leadership becomes less necessary and leadership roles dynamically switch from person to person as situations dictate.

Team Size

programmer : tester =4:1

programmer: on-site customers = 3:2

A team of 6 programmers produces a team that also includes 4 customers, 1 tester, and a project manager, for a total team size of 12 people. Twelve people turns out to be a natural limit for team collaboration.

The smallest team I would use with full XP consists of five people: four programmers (one acting as coach)and one product manager (who also acts as project manager, domain expert, and tester). A team of this size might find that the product manager is overburdened;

The Last Responsible Moment

XP views a potential change as an opportunity to exploit; it's the chance to learn something significant. This is why XP teams delay commitment until the last responsible moment .

user stories are short descriptions of features from an end-user's perspective, defining *what* needs to be built and *why*, while **iterations** (or Sprints) are fixed-length cycles where teams plan, build, test, and deliver a small, working increment of the product, containing several user stories, providing flexibility and continuous feedback for improvement.