In 2001, a group of 17 software engineers met to discuss how to bring software to market faster. They wanted to develop and launch quickly, then use built-in customer feedback loops to make quick improvements. The goal was to recognize value early and often and they develop something called the agile manifesto. This manifesto has four key values. First is it places value on individuals and interactions over processes and tools, very different than our predictive models. Second, it places value on working software over comprehensive documentation. You remember in predictive models, we always started with a plan and followed our documentation. Third, it places value on customer collaboration over contract negotiation, and finally number 4, it places value on responding to change over following a predictive plan.

When we talk about the vision, we can also think of this vision as the scope, but we don't call it a scope. What we call it in agile is a product backlog. A product backlog is a prioritized list of work or list of functions for the team to work on. I like to think of it as a to-do list.

The product backlog could contain things like new features or fixes from other sprint releases. Each function or small portion of the product backlog is broken down into something called a user story. A user story is a small unit of work in the agile framework. It's usually an informal general explanation of some feature written from the perspective of the end user or written from the perspective of the customer. Its purpose is to articulate how this piece of work will deliver value to the customer.

The project team takes the user stories, takes them into the sprint. The sprint sometimes is called the scrum. It is a process that consists of planning, it consists of the daily scrum, the sprint review, and the sprint retrospective.

The current content focuses on the 12 principles of Agile as outlined in the Agile Manifesto. Here’s a brief summary of each principle:

1. Customer Satisfaction: Prioritize satisfying the customer through early and continuous delivery of valuable software.
2. Welcoming Change: Embrace changing requirements, even late in development, to provide a competitive advantage.
3. Frequent Delivery: Deliver working software frequently, with a preference for shorter timescales.
4. Daily Collaboration: Encourage daily collaboration between business people and developers.
5. Motivated Individuals: Build projects around motivated individuals, providing them with the environment and support they need.
6. Face-to-Face Communication: Promote face-to-face conversation as the most effective method of communication.
7. Working Software as Progress: Measure progress primarily through working software.
8. Sustainable Development: Maintain a constant pace of development that is sustainable for all team members.
9. Technical Excellence: Focus on continuous attention to technical excellence and good design.
10. Simplicity: Maximize the amount of work not done by focusing on simplicity.
11. Self-Organizing Teams: Allow self-organizing teams to determine the best way to accomplish their work.
12. Continuous Improvement: Regularly reflect on team performance and adjust behaviors for improvement.

These principles emphasize the importance of collaboration, flexibility, and continuous improvement in Agile project management. Would you like to explore any specific principle in more detail?