In Chapter 11, titled "Discovering Requirements," various methods, models, and techniques are explored for identifying and articulating the requirements for interaction design. Here is a breakdown of each method or technique mentioned, along with definitions, usage, and the necessary steps or principles:

1. Double Diamond of Design

- **Definition**: The Double Diamond is a visual model that represents the design process in two phases—exploration (discover and define) and implementation (develop and deliver). The discovery and definition are crucial for understanding user needs and forming requirements.
- **When and Why It Is Used**: It is used to systematically explore and converge on solutions, guiding the designer from a broad exploration of the problem space to a refined definition of requirements.
- **Necessary Steps/Principles**:
  - **Discover**: Go wide to gather insights about the problem, users, and the context.
  - **Define**: Zoom in on specific problems and insights to form a clear design challenge.
  - **Develop**: Prototype and iterate to explore possible solutions.
  - **Deliver**: Finalize the design for implementation.
- **Cited Origin**: The Double Diamond model is used extensively in interaction and service design.

2. User Stories

- **Definition**: A user story is a brief description of a feature told from the perspective of an end-user. It is used to describe product functionality that will provide value to users.
- **When and Why It Is Used**: Primarily used in agile software development to define features incrementally. It helps maintain a focus on user needs throughout the project.
- **Necessary Steps/Principles**:
  - Structured as: "As a `<role>`, I want `<behavior>` so that `<benefit>`."
  - Write, estimate, and add acceptance tests.
  - Break epics (larger stories) into smaller, actionable stories.
- **Cited Origin**: Popularized by Kent Beck as part of Extreme Programming (XP).

3. Volere Requirements Shell

- **Definition**: A structured template for defining requirements, including attributes such as purpose, fit criteria, customer satisfaction, and priority.
- **When and Why It Is Used**: Useful for capturing precise and measurable requirements, especially in large or complex projects.
- **Necessary Steps/Principles**:
  - **Atomic Requirements Shell**: Break each requirement into specific attributes (purpose, fit criterion, etc.)
  - Ensure clarity with measurable criteria to know when requirements are fulfilled.
- **Cited Origin**: Developed by the Atlantic Systems Guild (Volere).

4. Personas

- **Definition**: A persona is a fictional character that represents a user type, built from research data to embody the goals, behaviors, and preferences of typical users.

- **When and Why It Is Used**: Used to help the design team stay focused on user needs and avoid designing for themselves rather than the target users.
- **Necessary Steps/Principles**:
  - Create personas based on synthesized data from user research.
  - Include relevant details such as goals, user quotes, behaviors, and environment.
- **Cited Origin**: Alan Cooper (1999), first used in "The Inmates are Running the Asylum."

5. Scenarios

- **Definition**: A scenario is an informal narrative that describes how users would interact with a system or product, focusing on a specific activity or goal.
- **When and Why It Is Used**: Scenarios help designers understand the context in which a product will be used, providing a rich story to explore product interactions.
- **Necessary Steps/Principles**:
  - Develop from the perspective of a specific persona.
  - Highlight user goals and interactions within a specific context.
  - Be used to create use cases or test design features.
- **Cited Origin**: Carroll (2000), widely adopted in HCI.

6. Cultural and Design Probes

- **Definition**: Probes are exploratory tools designed to elicit creative and emotional responses from participants, often used to gather qualitative data about users' lives.
- **When and Why It Is Used**: Used when standard data gathering methods might fail to yield deep insights about user context or motivations.
- **Necessary Steps/Principles**:
  - Use different types of probes, such as diaries, cameras, or sketchbooks, to encourage self-expression from participants.
  - Interpret responses to guide design explorations.
- **Cited Origin**: Developed by Gaver et al. (1999).

7. Contextual Inquiry

- **Definition**: Contextual Inquiry is a method that combines observation and interviewing, where the designer takes on the role of an apprentice to learn directly from the participant.
- **When and Why It Is Used**: It is used to understand the context of users' activities and gain insights into their workflows and behaviors in real environments.
- **Necessary Steps/Principles**:
  - Four main parts: overview, transition, main interview (apprenticeship), wrap-up.
  - Four guiding principles: context, partnership, interpretation, focus.
- **Cited Origin**: Holtzblatt and Jones (1993).

8. Use Cases

- **Definition**: A use case is a detailed description of a user's interaction with a product, capturing the functional requirements from the perspective of the user's goal.

- **When and Why It Is Used**: Useful for specifying the exact steps in an interaction and ensuring that functional requirements are covered.
- **Necessary Steps/Principles**:
  - Identify the user’s goal and define the sequence of actions needed to accomplish it.
  - Split into normal and alternative flows to cover different paths.
  - Develop "essential use cases" focusing on user intentions.
- **Cited Origin**: Developed as part of UML (Unified Modeling Language).

Summary of Chapter 11 Key Models and Techniques

This chapter provides foundational knowledge for discovering and representing requirements in interaction design. Understanding and employing a variety of methods—from user-centered personas to structured use cases—enables designers to create more effective, usable, and well-reasoned products. Each technique emphasizes a different facet of the requirements activity, from exploring user goals and contexts to specifying system functionality.