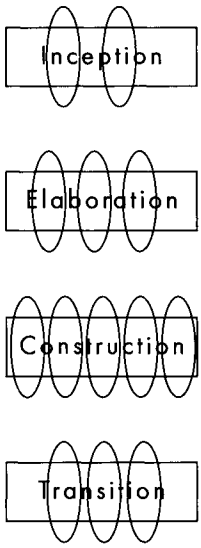

Figure 1.3 *Phases of object-oriented development with iterations of workflows*

The object-oriented approach also recognizes fully the reality of iterative development. Activities at any phase do not take place in a neatly ordered fashion. A developer may have to revisit a range of workflows several times during one phase of development, before it is possible to move on to the next phase. Figure 1.3 illustrates the phases of the object-oriented life cycle with iteration of workflows at each phase. You can see from the diagram that iterations are most likely during construction, but can occur during any phase of development. In the diagram each ellipse represents a range of workflows that may take place as shown in Figure 1.1.

In addition to the emphasis on iterative development, the object-oriented approach also differs from traditional life cycle models in that it stresses the importance of a seamless development process. This means that the separate phases are less distinct from each other than in a traditional system life cycle; it is not considered essential, nor is it often easy, to be able to say precisely when one phase is completed and another begins.

The seamless process is supported by the fact that object-oriented development is driven by a single unifying idea – that of the object. Objects initially represent things or concepts in the problem domain; they underpin the whole development process, and eventually become components of the code for the final system. Because the object is the foundation of all development work, object-orientation does not introduce new models to describe the system at different phases, but develops and refines early models from the inception phase right through the development process. This helps to preserve important information, and avoids the risk of inconsistency between multiple representations. It also brings the added advantage of