

### 2.1.1.3 Iterative development model

Incremental development involves establishing requirements, designing, building, and testing a system in pieces, which means that the software's features grow incrementally. The size of these feature increments vary, with some methods having larger pieces and some smaller pieces. The feature increments can be as small as a single change to a user interface screen or new query option.

Iterative development occurs when groups of features are specified, designed, built, and tested together in a series of cycles, often of a fixed duration. Iterations may involve changes to features developed in earlier iterations, along with changes in project scope. Each iteration delivers working software which is a growing subset of the overall set of features until the final software is delivered or development is stopped.

Examples include:

- Rational Unified Process: Each iteration tends to be relatively long (e.g., two to three months), and the feature increments are correspondingly large, such as two or three groups of related features
- Scrum: Each iteration tends to be relatively short (e.g., hours, days, or a few weeks), and the feature increments are correspondingly small, such as a few enhancements and/or two or three new features
- Kanban: Implemented with or without fixed-length iterations, which can deliver either a single enhancement or feature upon completion, or can group features together to release at once
- Spiral (or prototyping): Involves creating experimental increments, some of which may be heavily re-worked or even abandoned in subsequent development work

Components or systems developed using these methods often involve overlapping and iterating test levels throughout development. Ideally, each feature is tested at several test levels as it moves towards delivery. In some cases, teams use continuous delivery or continuous deployment, both of which involve significant automation of multiple test levels as part of their delivery pipelines. Many development efforts using these methods also include the concept of self-organizing teams, which can change the way testing work is organized as well as the relationship between testers and developers.

These methods form a growing system, which may be released to end-users on a feature-by-feature basis, on an iteration-by-iteration basis, or in a more traditional major-release fashion. Regardless of whether the software increments are released to end-users, regression testing is increasingly important as the system grows.

In contrast to sequential models, iterative and incremental models may deliver usable software in weeks or even days, but may only deliver the complete set of requirements product over a period of months or even years.