Rapid Application Development (RAD) is an advanced technique in software engineering that aims to expedite the development process by using iterative and incremental development approaches, prototyping, and close collaboration between developers and end-users. It focuses on quickly building and deploying functional software prototypes to gather feedback and make rapid iterations.

RAD incorporates the following key features and techniques:

1. Iterative and Incremental Development: RAD breaks down the software development process into small, manageable increments. Each increment focuses on delivering a specific set of functionality. Through iterative cycles, developers continuously build, test, and refine the software incrementally, allowing for early feedback and improvement.

2. Prototyping: RAD heavily relies on prototyping techniques. Instead of spending significant time upfront on detailed requirements analysis, RAD emphasizes the creation of functional prototypes. These prototypes act as visual representations or mock-ups of the final software system, allowing users to interact with and provide feedback on the system's look, feel, and functionality.

3. Timeboxing: RAD employs timeboxing to set strict time limits on each development iteration or prototype. This time constraint ensures that the development team stays focused and delivers tangible results within the allotted time frame. Timeboxing helps manage expectations, prioritize functionality, and maintain momentum throughout the development process.

4. Joint Application Development (JAD): RAD promotes active collaboration between developers, end-users, and other stakeholders through techniques like Joint Application Development (JAD). JAD sessions involve intensive workshops and meetings where all parties participate in requirements gathering, design discussions, and problem-solving. This collaborative approach ensures that the software system aligns closely with user needs and expectations.

5. Reuse of Components: RAD encourages the reuse of existing software components or frameworks to expedite development. By leveraging pre-existing modules or libraries, RAD reduces the time spent on coding and focuses on customizing and integrating components to meet specific requirements.

6. Automated Tools and Code Generators: RAD leverages automated tools, code generators, and visual development environments to accelerate the development process. These tools simplify and automate routine coding tasks, allowing developers to focus more on business logic and functionality.

RAD is particularly effective in situations where requirements are volatile or not fully known upfront, where time-to-market is crucial, or where close collaboration with end-users is essential. By emphasizing rapid prototyping, iterative development, and close collaboration, RAD helps to mitigate risks, reduce development time, and deliver software systems that better meet user expectations.