Several architectures, costs, and management differences become apparent when comparing traditional software development with cloud development. Traditional software development can be envisioned as a linear process, such as software is conceived, developed, tested, and deployed to a live environment. In contrast, cloud development is like a highway with many more lanes for an express, iteration-heavy, almost agile process. In a cloud development environment, software is continuously integrated and continuously tested, much more like a process one would envision in a factory with many more workstations and assembly lines.

Often, software is defined as a standalone product with specific capabilities. Platforms serve as a foundation and enable the building of applications via APIs and services. They even allow for some adaptability and customization, which is the opposite of traditional software.

Serverless architectures, part of cloud-based systems, are entirely overseen by the service providers, allowing the customers to transfer maintenance and ownership costs to the providers. This makes it a much cheaper way to run applications since it incurs none of the overhead that internal infrastructure and the staff that maintain it require. The alternative involves high upfront costs for on-premises servers and strikingly high operational costs thereafter.

Key differences between traditional and cloud architecture include Clouds are superior in distributing workloads across many servers, achieving high performance and resilience. Adaptability and scalability Cloud solutions provide nearly limitless storage and capacity. Cloud environments are different from traditional systems, which need a lot of dedicated personnel for maintenance, and many processes get automated in the cloud. The cost structure operates on a pay-per-use model, making it more cost-effective, while traditional systems involve upfront costs for hardware and software.

Cloud computing offers many advantages, but it's unlikely that traditional operating systems will become obsolete like the floppy disk. They remain very flexible and powerful computational engines that are still just right for many IT scenarios.