**McCall, Richards & Walters (1977) - Key Points:**

* Introduced a framework for software quality, focusing on factors such as reliability, usability, efficiency, maintainability, and testability.
* Emphasized a product-oriented approach, categorizing quality attributes that are inherent to the product itself.
* Quality was seen as a static set of attributes that could be measured and improved upon post-development.

**Klotins, E., Unterkalmsteiner, M. & Gorschek, T. (2019) - Key Points:**

* Explores software engineering practices in the dynamic and fast-paced context of start-up companies.
* Emphasizes the role of rapid development, agility, and the minimal viable product (MVP) concept.
* Highlights that quality is often balanced with speed and market adaptability.

**Reflection:**

In reflecting on the evolution of software quality concepts from McCall, Richards & Walters (1977) to the present, one notices a significant shift from a product-centric to a more process and market-driven perspective. The 1977 article lays the groundwork for defining software quality by identifying key product attributes. However, it doesn't extensively address the changing nature of software development processes over time or the market's influence on defining what quality means.

In contrast, recent literature, such as the work by Klotins et al. (2019), captures the essence of modern software engineering in start-ups, where software quality is not merely about product attributes but also about how quickly and effectively a product can be delivered to the market. This approach often involves trade-offs, and quality is defined in terms of customer satisfaction and fit-for-purpose rather than just internal product characteristics.

Moreover, with the advent of Agile methodologies and DevOps, the focus on continuous integration, deployment, and delivery has reshaped the concept of quality. It is no longer seen as the final checkpoint but as a continuous concern throughout the development lifecycle. This iterative approach to both development and quality assurance ensures that products can quickly adapt to user feedback and changing requirements.

The current understanding of software quality encapsulates a broader spectrum, including security, scalability, and performance in a continuous delivery environment. As such, quality is a moving target, influenced by customer needs, market pressures, and technological advancements.

In summary, while McCall et al.'s framework laid the foundational understanding of software quality, the contemporary perspective emphasizes adaptability, customer-centricity, and the role of quality in a continuous development process. The agility and responsiveness to change are now as important as the inherent attributes of the software product.