Ensuring that your code, program, or software is both functional and secure requires a multifaceted approach. It begins with a clear understanding of the functional requirements through effective communication with stakeholders and users. Active listening and empathy are crucial in interpreting user needs and incorporating them into the program. This involves breaking down user requirements into smaller, manageable user stories or use cases and creating prototypes or mockups for feedback. Iterative development allows for the integration of user feedback, ensuring that the software aligns with user expectations.

In terms of functionality, automated testing plays a vital role. Implementing unit tests, integration tests, and end-to-end tests helps verify that the code behaves as expected and meets the functional requirements. Additionally, code reviews by peers or experts help catch bugs, improve code quality, and ensure adherence to best practices. Regular security audits and the application of secure coding practices, such as input validation and proper error handling, are essential for ensuring the security of the software. Access controls should be implemented to restrict access to sensitive data and functionality based on user roles and permissions.

When approaching the design of software, it's important to begin by thoroughly understanding the problem domain and the needs of the users. Architecture design involves designing the overall structure of the software, including its components and interactions. This is followed by detailed design, where the architecture is broken down into smaller components, and the details of each component are designed. Non-functional requirements, such as performance and scalability, should also be considered during the design process. Prototyping and validation help ensure that the design decisions are sound before proceeding with full-scale implementation. Throughout the design process, documentation of design decisions is important for facilitating understanding and future maintenance.