The Implementation phase of the Software Development Cycle is where the actual coding happens, and software architecture is like a blueprint for how the system should be built. It's crucial to understand how these two parts depend on each other.

The implementation phase can sometimes add or modify the planned architecture. This can happen because:

* Sometimes, issues or challenges that were not expecting during the planning phase are discovered during implementation. To solve these, changes to the architecture might be necessary.
* New requirements or changes to existing ones can emerge during implementation, necessitating modifications to the architecture.

Also, when changes to the architecture are needed during the implementation phase, several factors must be considered:

* Changes should not disrupt the consistency and integrity of the system. The overall structure should remain coherent.
* Update all architectural documentation to reflect the changes. This helps maintain clarity and understanding for all team members.
* Re-test the affected parts of the system to ensure that the changes do not introduce new bugs or issues.

A generative implementation approach involves creating code through automated processes or using templates. This approach has several advantages concerning architecture:

* Automated generation ensures that the code adheres to architectural standards and guidelines consistently, reducing human error.
* It makes the system more scalable, as new features or components can be generated following the established architectural patterns.
* Since the generated code follows a uniform structure, it is easier to maintain and modify in the future.
* With quality it helps in maintaining high code quality, as the templates used for generation can embed best practices and standard patterns.