Discussion-8 The best practices for closing the loop on software projects

Discussion Topic:

Please choose one of the following questions to discuss in your initial post:

- What are some of the key challenges of assessing a complex software project?
- How can we ensure that our software projects are meeting the needs of our users?
- What are some of the best practices for closing the loop on software projects and learning from our experiences?

My Post:

Hello Class,

For this discussion, I chose the following topic:
What are some of the best practices for closing the loop on software projects and learning from our experiences?

The key principles for closing the loop on software projects and learning from experiences are to conduct evaluations, to create a final documentation, and to identify the team's needed improvement by learning from experience.
An important aspect of closing the loop is assessing if the final application meets the project's objectives. This can be done by creating a final "report card" for the architecture (Keeling, 2017a); by inviting reviewers to review the final work (CSU Global, 2025); then by analyzing the feedback from the reviewers and addressing identified issues; by creating and using a final checklist to review all aspects of the project; and finally by documenting issues that cannot be fix due to time constraint or mixing resource, describe the issue in am issues list and suggest potential solutions.
To learn from experience and to identify the team's needed improvement, it is important for an architect (the man in charge) to empower team members by providing them with the necessary knowledge and training to understand the project architecture (Keeling, 2017b); to delegate design tasks to team members, allowing them to make design decisions; and finally by giving time to teams members to reflect on what they have learned and what skills they individually need to improve or acquirer.

-Alex

References:

CSU Global (2025). *Module 8: Implementation and Post Steps; Final Project Deliverables* [Interactive Lecture]. Canvas. https://csuglobal.instructure.com/courses/110425/pages/module-8-overview?module_item_id=5733433

Keeling, M. (2017a). Chapter 12: Give the Architecture a Report Card. *Design it! From programmer to software architect*. Pragmatic Bookshelf. ISBN-13: 978-1-680-50209-1

Keeling, M. (2017b). Chapter 13: Empower the Architects on Your Team. *Design it! From programmer to software architect*. Pragmatic Bookshelf. ISBN-13: 978-1-680-50209-1