Unit 1: Introduction to Software Engineering Project Management

Required Reading

Stellman, A. & Greene, J. (2008) Applied software project management. 1st ed. Sebastopol, CA: O'Reilly.

- Chapters 1 - 6.

Chapter 1: Introduction This chapter introduces the concept of software project management, emphasizing the importance of transparency, trust, and effective communication within a project team. It outlines the common problems faced in software projects and the need for project managers to have a broad range of expertise to guide their teams effectively.

Chapter 2: Software Project Planning This chapter discusses the importance of planning in software projects. It covers the creation of a vision and scope document, the work breakdown structure (WBS), and the project plan. The chapter emphasizes the need for consensus among stakeholders and the significance of defining project scope early to avoid chaos later.

Chapter 3: Estimation This chapter focuses on the estimation process in project management. It introduces the work breakdown structure (WBS) as a foundation for accurate estimates and discusses the Wideband Delphi method for consensus-driven estimation. The chapter highlights the importance of involving both technical and non-technical stakeholders in the estimation process.

Chapter 4: Project Schedules This chapter explains how to build and manage project schedules. It emphasizes the need for a WBS and effort estimates before creating a schedule. The chapter discusses the use of scheduling software and the importance of updating schedules based on team feedback and project status.

Chapter 5: Reviews This chapter covers various types of reviews in the software development process, including inspections, deskchecks, and code reviews. It highlights the benefits of reviews in catching defects early, fostering collaboration, and ensuring quality in work products. The chapter also addresses common objections to reviews and how to overcome them.

Chapter 6: Software Requirements This chapter discusses the significance of gathering and documenting software requirements before development begins. It introduces the concept of use cases and the software requirements specification (SRS), which outlines the behavior and functionality of the software. The chapter emphasizes the need for thorough requirements engineering to avoid project delays and failures.

Lehtinen, A., Mäntylä, V., Vanhanen, J., Itkonen, J. & Lassenius, C. (2014) Perceived causes of software project failures – An analysis of their relationships. Information and Software Technology 56(6): 623–643.

The article analyzes the perceived causes of software project failures by conducting a root cause analysis (RCA) in four software product companies. It aims to understand the interconnections between different causes and their impact on process improvement. The study identifies that software project failures are often the result of multiple interconnected causes across various process areas, including management, sales & requirements, implementation, software testing, and release & deployment. Key findings include the significance of bridge causes, such as lack of cooperation and weak task backlog, which connect different process areas. The research emphasizes the need for case-specific analysis and collaboration among stakeholders to effectively prevent software project failures. The study also highlights the importance of understanding causal relationships to improve software process outcomes.

Goatham, R. (2020) Why Projects Fail.

The document discusses the reasons behind project failures, highlighting common symptoms such as schedule slippage, quality flaws, and budget overruns. It emphasizes that these symptoms are merely indicators of deeper issues, which can be categorized into trigger events and behavioral patterns. The text outlines eight primary causes of project failure, including market and strategy failures, organizational and planning failures, leadership and governance failures, underestimation and analysis failures, quality failures, risk failures, skills and knowledge failures, and engagement and communication failures. The document advocates for studying past project failures to understand root causes and improve future project success rates.

Additional Reading

Brooks, P. (1975) The Mythical Man-Month - Essays in Software Engineering. Anniversary ed. Addison-Wesley.

- Chapters 1 - 5.
- Chapters 11-13.