SOFTWARE PROJECT MANAGEMENT PHASES & BEST PRACTICES

As more organizations become software factories, developing and releasing their own software products to customers, software project management is an essential practice.

In this article, we will look at the software project management practice, including the four-phase approach that enables organizations to build high quality software—at scale and within project scope.

What is software project management?

Project management is the discipline of defining and achieving project goals while optimizing for any resource constraints during the lifecycle of a project.

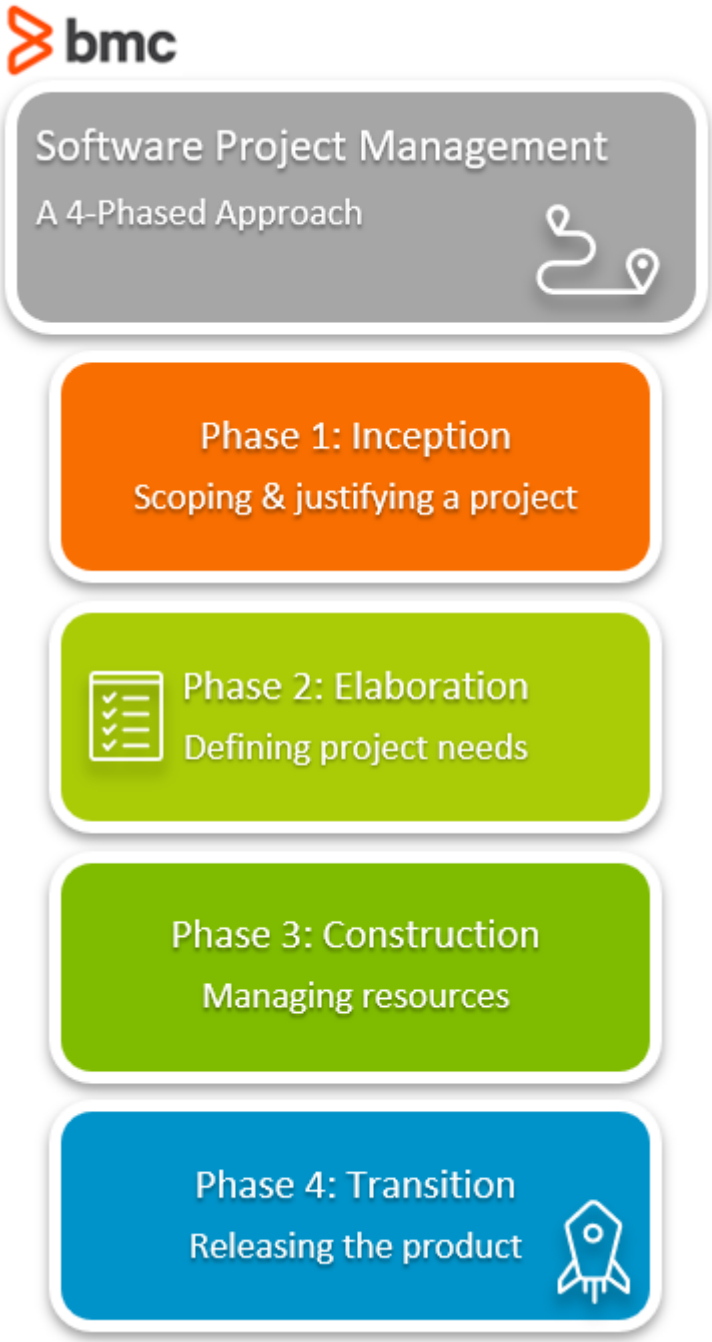
A subset of project management, Software Project Management is the practice of planning and delivering software development projects within variables such as:

- Time
- Quality
- Cost
- The wider scope

The software project follows the end-to-end Software Development Lifecycle (SDLC), which encompasses many steps from gathering requirements to developing and testing to releasing the product and, finally, ongoing maintenance.

(See how product management supports the IT organization.)

Now, let's turn to the four phases.



Phase 1: Inception

Scoping & justifying a project

In the first phase of software project management, IT organizations identify the requirements, product features, risks, constraints, and scope of the development project.

Your team will devise a plan to meet the intended budget and support the technical and business case of the end-product. This plan will define the following tasks:

- **Defining the software process.** Defining the SDLC models, frameworks, and the roles of teams and individuals that support the delivery of a high-quality product.

- **Requirements engineering.** Documenting detailed specifications and requirements for the development and operational processes. Developing a system model to meet the necessary functional and non-functional requirements.
- **Planning & budgeting.** Use the project scope to define your budget, schedule, and resources required. Develop milestones to achieve these goals. Do be prepared, however, that despite careful planning, actual numbers will likely deviate from the original estimates.

Phase 2: Elaboration

Defining the needs

In the second phase of software project management, you will complete and validate the project plan and the architectural design. Then, identify any risks and manage them accordingly.

You've already agreed to the project vision and requirements, so you'll follow these tasks to achieve the project goals:

- **Managing the risk.** Understand how risk can be mitigated and managed. Develop a playbook that covers probable risk areas and best-practice response guidelines.
- **Modeling & design.** Visualize or simulate the system and environment models of the technology stack, product architecture, and the SDLC framework. The model considers all interactions between the system components and appropriate external factors. Some of the popular SDLC models include DevOps and Agile.
- **Executing the project plan.** Assign the roles and responsibilities for teams, managers, and employees. Identify the required tools and services, and provision them through a systematic governance framework.

Phase 3: Construction

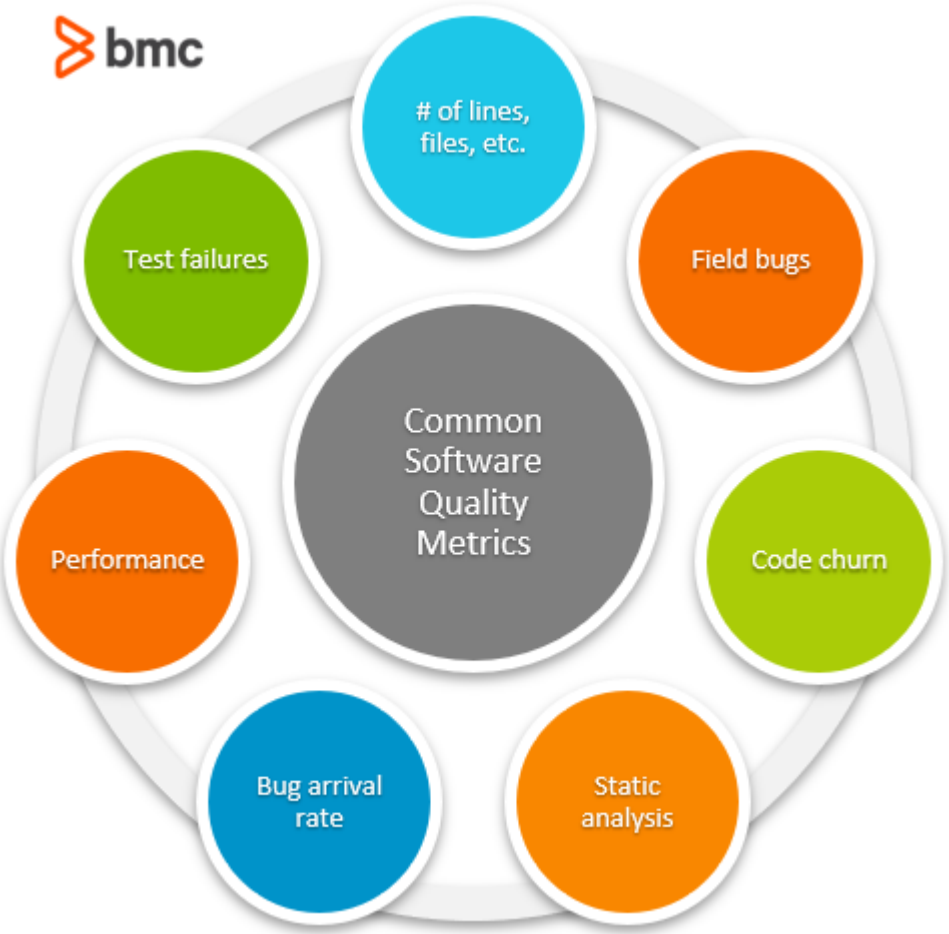
Managing resource provision

The third phase of software project management deals directly with the development process. Monitor the progress of the development process against the defined requirements and user expectations—make sure you're staying on track for timelines and expectations. Here, you might also provide employees supporting the project with any necessary training, education, and support.

Key tasks followed during this phase include:

- **Designing the details.** Describe how the documentation and architectural design guides the development of software product components, builds, and features. Explain the design patterns and follow them systematically.
- **Managing the quality.** Identify the activities and the qualitative and quantitative measures of software quality. Understand which metrics can be analyzed through the software testing process to achieve high quality as intended.

(Explore common software quality metrics.)



Phase 4: Transition

Releasing the product

In the final phase of the software project management, you'll validate the final product build against all technical and business requirements.

You'll complete the necessary artefacts and the development team must prepare for the next iteration of the software development cycle. You've already learned lessons from the first iteration, so apply these to support continual improvement. Depending on the SDLC methodology you're using, you may release specific feature updates, components, or the complete product to end-users.

Key tasks in the Transition phase include:

- **Evolving.** Describing how software development teams can transition to the next iteration of the project. The iteration may produce a software build or a functioning feature component, depending on the chosen SDLC framework.
- **Seeking feedback.** Identify the opportunities and challenges you experienced during prior iterations and apply the lessons in the next iteration of the SDLC. The DevOps feedback loop is perfect for this. For organizations following Agile and DevOps SDLC frameworks, the feedback process is an integral element of the SDLC process. In subsequent iterations of the development process, there may be limited but ongoing improvements and changes in requirements. Provisions for such changes should already be included during the first three

phases of the software project management model.

- **Closing the project.** Success is measured following project completion. Project managers are required to identify project performance and determine whether the project goals were achieved within the agreed scope and constraints (time, cost, quality, other). Then, document the project closure and conduct a and post-implementation reviews. Account for, reallocate, or retrieve unused resources for future project implementations. Finally, debrief the relevant teams on the project performance and evolution.

(Compare software deployment to release.)

Software project management supports success

The goal of a structured software project management practice is not to add more work. Instead, it's to guide the work through a series of steps and tasks that support product delivery within the constraints you have.

Any framework should support—not hinder—your ability to deliver quality projects on time and within budget. If not, see which phase of the software project you're falling short and improve from there.

Related reading

- BMC DevOps Blog
- Intro to Agile with Scrum: 4 Tips for Getting Started
- Testing Automation Explained: Why & How To Automate Testing
- Quality Assurance (QA) in Software Testing: QA Views & Best Practices
- Managing IT as a Product—Not a Project
- The Project Management Office (PMO) & Its Role in IT Organizations