

## DEFINITION

EMAIL THIS PAGE TO A FRIEND



# Milestone

(Alias: management gate)

At 15 I set my heart upon learning.  
 At 30 I established myself (in accordance with ritual).  
 At 40 I no longer had perplexities.  
 At 50 I knew the mandate of Heaven.  
 At 60 I was at ease with what ever I heard.  
 At 70 I could follow my heart's desire without transgressing the boundaries of right.  
 - Confucius

A milestone is a significant event in the course of a project that is used to give visibility of progress in terms of achievement of predefined milestone [goals](#). Failure to meet a milestone indicates that a project is not proceeding to plan and usually triggers corrective action by management.

## Setting Milestones [↑](#)

Milestones have no duration, they represent instantaneous events that occur throughout the project. Typical project events that are marked with milestones are:

- The completion of a project phase
- The approval of a [deliverable](#)
- The completion of a scheduled review
- The completion of an activity
- The commencement of an activity.

### Contents

[Setting Milestones](#)  
[Describing a Milestone](#)  
[Setting Milestone Goals](#)  
[Major Milestones for Software Projects](#)  
[Minor Milestones for a Software Project](#)  
[How Many Milestones?](#)  
[Rules of Thumb](#)

## Collaborate



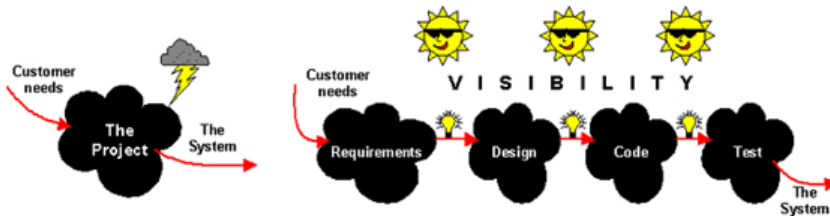

- Rate this definition.
- Did it help?
- Suggest improvements.
- Request more information.
- Exchange ideas with our member community.

### LEARN MORE

[CA Workshop](#)
[Managing Software Projects](#)
[CA Service](#)
[Project Management](#)
[Video](#)
[Earned Value Management](#)

### No Milestones = No visibility

### Milestones = Visibility



## Describing a Milestone [↑](#)

In defining your Software Development Methodology (SDM) you should describe your milestones in terms of the following attributes:

- **Milestone name.** A short catchy phrase or acronym that describes the milestone (Example: Preliminary Design Review - PDR)
- **Project event.** A description of the project event to which the milestone is tied (Example: formal approval of the Preliminary Design by the PDR team)
- **Milestone goals.** The criteria that will be applied to determine if the milestone has been reached - the milestone goal(s) (Example: Preliminary Design approval).

When you apply the SDM to a specific project all your milestones should also be defined in terms of an absolute date or an elapsed time period from a previous milestone such as project commencement.

## Setting Milestone Goals [↑](#)

Make sure your milestone [goals](#) are verifiable. You should be able to verify a goal with a yes or no answer. For example, the Software Requirements Specification is either complete or not complete, the Software Design Description has either passed or not passed peer review. The software product is or is not in commercial operation.

Beware of fuzzy, unverifiable milestone goals that allow one phase of a project to merge into another resulting in loss of visibility and project control. For example, "high level design complete" is a fuzzy goal, "Software Architecture Design Description Document approved", is more verifiable.

Typical milestone goals are:

- Designated deliverables approved
- Phase completion review complete
- Tests complete and all anomalies rectified.

The verification that a milestone has been reached usually occurs in the context of a management review. Milestones are therefore often named after major project review meetings. For example, Critical Design Review.

In some situations the commencement of work can be a valid milestone, the principle being that you can't meet your end date if you haven't met your start date. Canny project managers often use this type of milestone when supervising remote sub-contractors. They prefer to know about late starts when action can be taken to lessen the impact of a late delivery on the project.

## Major Milestones for Software Projects [↑](#)

Milestones are classified as major and minor. The major milestones give visibly or progress to people external to the project. For example, project sponsors and customers. The table provides a set of classical major software project milestones together with their milestone goals.

Milestone	Milestone Goal
Concept approval	Feasibility studies and basic system concepts have been approved by management and the project is authorized to proceed to detailed requirements definition.
Requirements review	Requirements specifications are complete, correct, approved and suitable for input to design.
Preliminary design review	The architectural design satisfies all product requirements, is approved and is suitable for input into the detailed design process.
Critical design review	Detailed designs fully implement the system architecture, are approved and are suitable for input into the development of code.
Test plan review	Test plans are adequate for the testing of all product features, are approved and are suitable for input to the development of test cases and test procedures.
Test readiness review	Developed and unit tested software has been passed by the test team and is suitable for input into integration testing.
System test review	The software product has passed system testing and is suitable for input into acceptance testing.
Operational readiness review	The software product has passed acceptance testing and is suitable for deployment in its target production environment.
Product operational	The software is in use in its target operational environment.

## Minor Milestones for a Software Project [↑](#)

Minor (or micro) milestones are the monitoring points you as project manager use to maintain control of day to day activities. They divide the elapsed time between major milestones into shorter intervals to give you the confidence that the major milestones will be met. They also give the team a sense of achievement by demonstrating progress on a daily or weekly basis. The table provides examples.

Milestone	Milestone Goal
Document outline complete	A document outline has been produced describing the format, content and objectives of each major section of a large document.
Document complete	A document such as an Software Requirements Specification has passed peer review.
Technical investigation complete	The investigation of a technical issue is complete and a summary of the main issues and conclusions has been presented and approved.
Compile complete	A program compiles without errors.
Software module complete	A small program or function has been completed and unit tested.
Software product build complete	The software product (or one of its components) has been built and it runs (not necessarily without errors).
Test case complete	A small unit of testing has been completed and test results recorded.
Bug fix complete	A bug identified in a Software Anomaly Report (SAR) has been fixed and the SAR closed out.

## How Many Milestones? [↑](#)

How many milestones should your project have? The answer springs from the frequency at which you need to monitor progress to keep the project under control. This frequency is determined by how fast things can go wrong with no oversight. Two examples of common "go wrong" scenarios are:

- **User rejection.** A large team of developers burning through project funds at the rate of \$70,000 per day takes three months (without intermediate milestones) to deliver a system function that is rejected by the end user.  
Solution: Implement Agile Development. Involve the user in the development process. Create a milestone every two weeks where the developers must demonstrate some functionality to the users.
- **Immature technology.** A project team is attempting to implement new technology in a production environment. The technology does not perform. Several workarounds are tried; all fail. At the end of six months, unbeknown to management, nothing has been achieved.  
Solution: Establish monthly progress review milestones. Require documented evidence of progress in Technical Investigation Reports.

The following scenarios further demonstrate how judicious use of milestones can maintain visibility of progress:

- Inexperienced people require daily milestones to ensure their lack of practical skills does not lead them too far down the wrong path
- Complex technical issues and algorithms need regular peer review and decision making to avoid "analysis paralysis"

- Large projects need fine grained deliverables to provide visibility that the project is progressing.

Rules of Thumb [↑](#)

Some general rules of thumb in nominating milestones are:

- All tasks should have a demonstrable milestone at least once every 5 to 10 working days
- Code development needs demonstrable progress with a minor milestone every one to five days. In fact some development strategies call for a nightly build of the product
- If you are a project sponsor, a major milestone demonstrating progress at 1 to 3 month intervals will probably be adequate. After all, you have a competent project manager performing the micro management don't you?
- All projects should have at least 4 major milestones: requirements complete, design complete, code complete and test ready, and product shipped/accepted/deployed.



Comment on this Definition

Systems Development Executives and Managers

- Project Planning Process
- Risk Management Planning
- Training
- Mentoring
- Influence
- Tender Preparation

Business Analysts and Requirements Engineers

- Specifying Software Requirements (video)
- Specifying Software Requirements Workshop
- Software Requirements Specification Standard
- Requirements Specification Learning Needs
- Requirements Elicitation
- Use Case

Architects and Designers

- System Architecture Specification
- Architecture
- Architectural Views
- Design Entity
- Design View
- Control System Design

Quality and Systems Assurance Managers

- Does Quality Matter?
- Quality Management System Architecture
- How to Implement a Software Quality Management System
- Configuration Management
- Verification and Validation
- Functional Safety Management

Testers

- Software Testing
- Test Documentation
- Quality Control
- Testing Services
- V&V Services
- How to Implement a Software Quality Management System