



[Course](#) > [Week 2...](#) > [2.1 Sim...](#) > 2.1.2 Si...

### **Audit Access Expires Jun 27, 2019**

You lose all access to this course, including your progress, on Jun 27, 2019.

Upgrade by Jun 20, 2019 to get unlimited access to the course as long as it exists on the site. **[Upgrade now](#)**

## 2.1.2 Simple PM Methods Summary Points

This lesson is all about understanding the high-level components for each Project Management Method:

- Traditional
- Agile
- Lean

### Traditional Project

- **Idea** - Traditional projects start with an Idea by the owner or team
- **Business cases** - Business Cases are performed for each concept; often a qualitative and quantitative process
- **Bid & Proposal** - Project scopes of work are written and competed amongst teams
  - External vendors will submit bids and proposals to win work against a Request for Proposals
  - Internal teams will submit budgets and proposals to win allocated funding from a PMO or Portfolio function
- **Waterfall Development** - each stage is executed sequentially, with limited iteration or "going backward" in stages
  - *Requirements* - the process of defining high, medium, and low-level needs of the end users and stakeholders
  - *Design* - the process of architecting the solution to meet those requirements within the project constraints
  - *Implementation* - the building of the product to specification

- *Verification* - the testing and product to ensure it meets specifications and the requirements' intent
- *Maintenance* - the design of and support of deploying and maintaining the product in operations
- **Operations** - the use of the product to produce benefits to the organization
- **Disposal** - the retirement of the project according to regulations and sustainability practices

Often the Traditional Process is controlled further with Stage Gates, where stakeholders agree the project is ready to move from one Waterfall Development Stage to the next. This can help give executives clear points of approving or disapproving the work; as well as inform stakeholders of the high-level progress of development.

*Standard Stage Gates:*

Idea Readiness Review (IRR)	precedes Business Cases
Concept Readiness Review (CRR)	precedes Bid & Proposal Stages
Planning Readiness Review (PRR)	precedes Requirements phase
Requirements Readiness Review (RRR)	precedes Design phase, exits Requirements phase
Design Readiness Review (DRR)	precedes Implementation phase, exits Design phase
Implementation Readiness Review (IRR)	precedes Verification phase, exits Implementation phase
Operational Readiness Review (ORR)	precedes Maintenance (O&M) phase, exits Verification phase
Operations and Maintenance Review (OMR)	Precedes Disposal phase, exits Operations (O&M)

## Agile Project Management

- **Idea** - (see Traditional above, it's the same)
- **Business Case** - (see Traditional above, it's the same)
- **Bid & Proposal** - (see Traditional above, it's the same mostly, except the contract type may differ)
- **Early Agile Development** (Pre-Release of first version)
  - Sprints - use the Scrum Method to deliver increments of working product iteratively until release ready
    - *Sprint Planning* - Product Owner and Team plan work for the sprint
    - *Sprint Development* - Team designs, builds, and tests increments of work in a fixed time period
    - *Sprint Retro & Review* - Customer reviews the work and team reviews the sprint for improvement
  - Deploy First Version to Production - this first version is often called the minimum viable product (MVP)
- **Continuous Delivery** (Post-Release of first version)
  - The same team(s) supports development and operations or "DevOps"
  - These are still executed using the Sprint model, only the team must account for supporting Operations
  - *Development*
    - *Create* - try something new or build fixes
    - *Verify* - ensure that it works
    - *Package* - get it ready for release
  - *Operations*
    - *Release* - deploy the new features/enhancements/bug fixes
    - *Configure* - ensure operations and test features on/off
    - *Monitor* - monitor performance of functionality
    - *Plan* - prioritize the next improvement or fix
- **Disposal** - (see Traditional above, it's the same)

## Lean Project Management

- **Issue** - could be an idea, major problem, or series of problems the client or owner foresees for the business
- **Work Concept** - work is formed as either a series of small challenges or one big challenge
  - *Issue Backlog* - for support contracts, there needs to be total or sample backlog of work to support
  - *Technical Challenge* - for technical solutions, there is a challenge set with clear performance objectives
- **Bid & Proposal** - (see Traditional above, it's the same; although contract types will differ)
- **Work Definition**

- *Dispose Issues* - issues are classified in urgency and impact to define the priority and who should respond
- *Define Work Breakdown* - team evaluates and decomposes the Technical Challenge into a work breakdown structure or "WBS."
- **Continuous Delivery**
  - *Value streams* - support for solving lots of small problems goes through a series of predefined steps to ensure quality and drive customer satisfaction the issue is addressed.
    - This is often supported using a defined workflow
    - The issues are routed based on priority and impact to different team members
    - Only the client or customer or their representative can accept it as done
  - *Incremental Delivery* - a team will incrementally work through the WBS to solve a major problem
    - Often use a Kanban board to solve highest priority/most uncertain work
    - Delivery is continuous and incremental to explore solution sets that COULD work
- **Operations** - solutions are deployed into operations where customers receive benefits
  - Example issue solution - solving someone being locked out of their system
  - Example Technical Challenge solution - deploying a new upgraded machine for manufacturing a car
- **Disposal** - (see Traditional above, it's the same)

Learn About Verified Certificates

© All Rights Reserved