



[Course](#) > [Week 2...](#) > [2.5 Co...](#) > 2.5.2 C...

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](#)

2.5.2 Comparing Methods of Engineering Summary Points

	Traditional	Agile	Lean
Project Size	Large	Medium	Small
Industries	Construction Military Government / Policy Relocation	Information Technology Product Development Management Consulting Operations	Sales Customer Support Legal R&D
Design	Dependent / Coupled	Independent / Decoupled	Constrained / Evolutionary
Teams	Departmental	Matrixed / Projectized	Emergent (Ad Hoc)
Development	Linear	Iterative	Incremental
Integration / Testing	End Phase	Continuous	When Possible

Closing	3rd Party Acceptance	Team Acceptance	Customer Acceptance
----------------	----------------------	-----------------	---------------------

Important Note Before Reading Further:

- In this lesson there will be discussion of Applications, Features, and Services
- These are Systems Engineering Terms, and Common Terms for many problems - beyond Software Engineering!

Here's how it breaks down:

- **Application** - the business use of product(s) or service(s) to support the organization (e.g. a new car from the dealership)
- **Feature** - an aspect of an Application that delivers value to its customers (e.g. automatic air conditioning)
- **Service** - a set of functions that serve a common purpose (e.g. air sensor system, air cooling system, air heating system, ventilation)

Traditional

- Designs are Dependent of Coupled & Development is Linear
 - Driven by need for efficiency
 - Heavy amounts of reuse across components
 - Drives multiple dependencies into the system
 - Any small change in the services results in BIG costs
 - This can be efficient with few changes
- Teams: Departmental
 - To be efficient we use multiple departments or contracts
 - This specialization will lower costs of delivery
 - These teams are then matrixed into the project
- Integration and Testing: End Phase
 - Because of tight coupling, only at the end phase can we really test
 - All parts are connected and could impact each other

- This delays feedback, which in turn freezes requirements
- Closing: Third Party Verification
 - A third-party tester is needed to match requirements to the final product
 - Only this way can multiple stakeholders trust the results
 - This is also a large job, requiring verification expertise

Agile

- Designs are De-Coupled & Development is Iterative
 - Allows for building features one at a time
 - This requires additional scope around supporting services to make features independent
 - This reduces costs of changes, but increases baseline costs
 - This also allows for early release of features or minimum viable products (MVP)
 - Each iterative release can build on the release that came before it
- Teams: Matrixed / Projectized
 - Everyone is on the team
 - All departments see and hear the same intent
 - This ensures maximum accuracy in communication
 - The customer is also on the team
 - This also enables speed, reducing handoffs and delayed decisions
- Integration and Testing: Continuous
 - By continuously testing the work, the work can be closed for each release
 - This is done on cadence with the owners
 - This ensures that new features don't break old ones as we add scope with each release
- Closing: Team Acceptance
 - The testers are on the team and can immediately inform customer on quality
 - The customer is on the team and can accept the work

Lean

- Designs are Evolutionary & Development is Incremental Only
 - Limited up-front planning and control on releases
 - Features are built based on immediate need, not market segments
 - Services evolve underneath features to be both responsive and efficient
 - Designs can be complex to maximize reuse, requiring smart but limited engineering work
 - Key is to build "just enough"
- Teams: Emergent (Ad Hoc)
 - Teams include only those needed at each stage
 - This has a mix of people involved depending on current focus of work
 - This minimizes costs immediately, but can cause issues in ownership and handoffs
- Integration and Testing: When Possible
 - Lean projects are building on-demand without much up-front planning
 - This results in an integration "when possible" approach
 - As soon as the work is complete its integrated
 - Examples are
 - New discoveries introduced from R&D
 - Fixes to working systems, such as bug fixes or replacement parts
 - Upgrades and updates to legal documents once agreement is reached by parties
 - Completing deal stages once customer is ready or response time expires
- Closing: Customer Acceptance
 - Closing is based on customer acceptance only
 - The work stays open until the customer is satisfied
 - Customer is agreed to pay all costs until work meets satisfaction or project is cancelled
 - Even testers are simply informing customers, usually without authority

Understanding the impact that the chosen PM Method has on Engineering is essential for any project manager. Depending on your domain of engineering different types of engineering may be possible or nearly impossible. Can one incrementally build a house? It wouldn't be allowed by the zoning boards, but it would also be dangerous!

At the same time, you only need a single general contractor and a small team to get those repairs done before putting a house on the market. You may have a fixed budget and need the best ROI to get that house at maximum attractiveness so buyers offer you the best bang for your investment buck.

And when it comes time to sell, can we understand the inefficiency of always re-writing the typical contract used for closing a house? Why redesign a whole new contract when you can just update and modify a template? The legal profession lives on templates.

Every aspect of work has its natural fit, but knowing that Agile delivers the maximum profit - can we make our projects more Agile-ready? What does that require from the customer and the design?

This is where the world of Engineering is headed today:

- ***All Industries are converging on Agile***
- IT is integrating into every product - Andreessen's claim "Software is Eating the World"
- Examples of this trend include:
 - Traditional going Agile:
 - *Building Information Modeling (BIM)* - to drive the use of iterative planning and design for one or many buildings (e.g. Onuma System)
 - *Government Modular Acquisitions* - to enable products of products and systems of systems in the government space (e.g. Open-Government Movement)
 - Lean going Agile:
 - *Online Legal Products* - grouping concerns of typical Lean processes into automated productized support (e.g. Legal Zoom)
 - *"Everything as a Service"* - human resources, accounting, and marketing are now productized services or automated services for small businesses

Onuma Planning System Example

- Building Information Models to drive decisions on building investments
- Example of Agile approaches to modeling and planning large-scale requirements for buildings
- Modeled using "Rapid Planning Sessions" where
 - Large ships called "Cutters" drove facility requirements
 - Facility requirements then drove modeling new buildings and investment at Coast Guard sites
 - All future scenarios were modeled using simulations in Google Earth
 - Final details were architectural level, down to the "nuts and bolts."
- While this type of system may seem new, it's actually over a decade old
- This allows for iterative through otherwise static, top-down design processes with many interconnected planning details
 - All the stakeholders get in a room
 - The buildings are simulated in real time
 - Architects, Interior Designers, MEC vendors, and Construction can all contribute to designs in real time
 - Result is low costs in construction and significantly reduced numbers of change requests
- To learn more go here: <http://www.onuma.com/services/BimStorm.php>

Hopefully this has you thinking about scope, schedule, and budget in new lights and you've enjoyed learning about who uses Agile today - will you tomorrow?

Learn About Verified Certificates

© All Rights Reserved