

Lean Project Management (LeanPM[®]) Framework

Redefining Project Management



LeanPM Certification

[LeanPM.org/yellow-green-black-belt-certification](https://leanpm.org/yellow-green-black-belt-certification)



Lean Project Management (LeanPM®) Framework

v. 1.1

November 2021

Contents

Think Lean in Project Management.....	6
Concept and Purpose of Lean Project Management	8
The Concept of Lean Project Management	8
The Purpose of Lean Project Management	12
References	13
Chapter 1: Lean Concepts	14
Lean Culture.....	14
Continuous Improvement	14
Respect for People	15
The House of Lean	15
Lean Manufacturing	16
Jidoka	16
Just-in-Time.....	17
The 14 Principles of the Toyota Way	18
Lean Thinking	20
The Forms of Waste	24
References - chapter 1.....	27
Chapter 2: Lean Project Management (LeanPM) Principles	29
Serve People	30
Create Value and Eliminate Waste	32
Build Knowledge and Continuously Improve	34
Apply Systems Thinking	36
Communicate and Collaborate Effectively	38
Simplify	40
Chapter 3: How Do Projects Work? Project Success	42
Project Logic.....	42
Project Success.....	45
The Measure of Success	45
Conditions of Satisfaction	46
Project Variables	47
Project Time-Cost Trade-off	47
Trade-off Between Investment and Operation Costs	49
Trade-off Between Project Costs and Revenues	50
Takeaways: How Do Projects Work? Project Success	51

Chapter 4: Project Waste.....	52
Strategic Project Waste.....	53
Tactical Project Waste.....	58
Wrong deliverables or unnecessary features.....	59
Delays.....	59
Poor quality	60
Duplicate efforts	61
Lost productivity.....	61
Over-processing	62
Overcomplicating.....	62
Total Project Waste	64
Takeaways: Project Waste	65
Chapter 5: Lean Portfolio Management.....	66
Introduction to Lean Portfolio Management: The Role of Projects	66
Lean Project Portfolio Management Processes	69
Introduction	69
Develop Organizational Strategy and Set Objectives	70
Generate Ideas, Select and Prioritize Projects.....	70
Projects Sequencing (Schedule Prioritization)	74
Fund, Execute and Evaluate	79
Portfolio Process Efficiency and Project Lead Time	80
Takeaways: Lean Portfolio Management.....	82
References – chapter 5	83
Chapter 6: The Lean Project Life Cycle	84
Introduction	85
Idea Generation and Triage	86
A3 Analysis and Pre-Selection/Selection.....	87
Exploration Phase	88
Sequencing	88
Creation and Absorption.....	88
Project Retrospective and Evaluation	89
Takeaways: The Lean Project Life Cycle	90
Chapter 7: Cost of Time	91
Definition and Components of Cost of Time	92
Best Time to Decide: Last Responsible Moment vs. Most Responsible Moment	98
Takeaways: Cost of Time	100
References – chapter 7	100

Chapter 8: Lean Development Life Cycle.....	101
Development Life Cycle Models	101
Predictive Life Cycle	101
Iterative Life Cycle.....	101
Incremental Life Cycle	102
Adaptive (Agile) Life Cycle	102
Hybrid Life Cycle.....	102
The Development Life Cycle of LeanPM	103
Applicability of the Evolutionary Life Cycle	103
Does the Sequential Life Cycle Have Advantages?	104
The LeanPM Development Life Cycle Approach	105
LeanPM Iterative Life Cycle	106
LeanPM Exploratory (Lean Startup) Life Cycle	108
LeanPM Non-iterative Evolutionary Life Cycle	109
LeanPM Continuous Delivery Life Cycle	110
Uncertainty and the Lean Development Life Cycle	112
Project Uncertainty Continuum	114
Takeaways: Lean Development Life Cycle	115
References – chapter 8	116
Chapter 9: Lean Project Planning	117
Introduction to Lean Planning	117
Planning and the Network of Agreements.....	119
PDCA Cycle and Planning	123
Planning Constraints.....	124
Planning Items	126
Milestones	126
Stages.....	126
Iterations	126
Features and User Stories.....	126
Tasks	127
Reliability of Planning	129
Multi-level Planning	131
Relationship Between Planning and Control	133
Push and Pull Planning.....	134
Customer Pull.....	134
Push Planning.....	135
Pull Planning	136
Takeaways: Lean Project Planning.....	138

References and notes – chapter 9	139
Chapter 10: Managing Creation and Absorption	140
Introduction	140
Visual Management	141
Decentralized Project Management.....	142
Set-Based Concurrent Design	144
Sequential (Waterfall) Design.....	144
Centralized Design	144
Concurrent Design	145
Point-Based and Set-Based Design	145
Built-In Quality	150
What is Built-In Quality?.....	150
Foundation and Pillars of Built-In Quality	151
Proactive Quality Management	152
Lean Quality Control	153
Lean Teamwork	155
Lean-Agile Teams	155
Lean-Agile Leadership	156
Transparency	157
Flow Management	158
The Goal of Flow Management.....	158
Controlling WIP	158
Managing Queues	165
Controlling Queues	169
Controlling Batch Sizes.....	176
The Water and The Rocks	179
References – chapter 10	180

Think Lean in Project Management

Building on the success of the Lean-based approaches in product development, construction and agile development, we believe that the future of project management is Lean.

Project Management has undergone a long evolution, yet it still faces many challenges and brings inconsistent results.

In recent decades, we've witnessed disruptive innovations in this area but the traditional understanding of how and why we should manage projects, and the uncertainty it often creates, still prevails.

The time has come for significant changes that need to start with rethinking the definition of a project and the role of project management.

Instead of adopting the pre-analytical vision that projects are needed or simply exist – and therefore must be managed – let's consider whether that's the case.

We believe that society is based on communication, interaction, and exchange. People are customers to one another and exchange efforts, skills, goods, services, information, emotions and money. Perceived value serves as a unit of measurement in this exchange.

The purpose of organizations is to provide value to society, customers, employees, partners and owners. That's why organizations create value streams – a sequence of all the steps necessary to create and use a product or service that brings value from concept to realization.

Organizations need specific initiatives to create and improve their value stream systems:

- Initiatives to improve the existing value stream system
- Initiatives to create new value streams involving new products or services
- Initiatives to eliminate obsolete value streams
- Emergency response initiatives to preserve or restore the value stream system
- Transformation initiatives to change the whole value stream system of the organization

What's common among these initiatives? They're all actions in the present that strive to create a better future – they're a *projection* from the present to the future. Therefore, "project" (from the Latin "projectum" meaning "something thrown forth") is an appropriate term to use for these endeavors.

In an organization, projects create value through serving the operational value streams. We can view them as temporary streams (sequences of steps) that create value for the organization's value stream system, which in turn creates value for the customers, employees, owners, partners and society. Similarly, when project deliverables are used by an external customer, benefits are achieved through the customer's value stream system.

LeanPM definition for a project:

A project is a temporary value stream, aimed at creating or improving the value stream system of an organization or its customers, thus creating value for the organization's stakeholders.

LeanPM definition for project management:

Project management is the practice of managing project value streams to maximize their net benefit.

Trillions are invested in projects every year by companies, governments, and nonprofit organizations. Many, to many projects, fail to provide the desired benefits and waste precious resources.

Even if we only change our perspective by adopting new definitions of project and project management, humanity can free up enormous resources and improve people's lives.

Let's build on the past, act in the present – and create the future!

This Framework focuses on the key principles and practices that must be considered in creating that new future. Project management needs to adopt Lean concepts to be more successful.

Concept and Purpose of Lean Project Management

THE CONCEPT OF LEAN PROJECT MANAGEMENT

According to the traditional concept, a project is a transformation of inputs to outputs. Project management deconstructs the total transformation into individual transformations (tasks) that need to be performed as efficiently as possible. Planning and execution are carried out through *management-as-planning* where project outcomes are realized by creating, updating and implementing plans. Dispatching the plans to the performers ensures their execution. Project control is carried out through the thermostat model – the performance is measured against a pre-defined standard and in case of a variance the process is corrected to reach the standard. [1] [2]

The practice of project management anchored in the transformation concept has three shortcomings, as Koskela and Howell argue [3]:

1. It's difficult to maintain a complete and up-to-date plan. The critical short-term planning is poorly performed or neglected.
2. Execution isn't managed systematically, as the actual project conditions are not taken into account when transforming higher level plans into short-term plans and actions.
3. Control is seen only as measuring and taking corrective action, while the opportunity for learning and eliminating root causes for underperformance is missed.

There are other major flaws of the transformation concept of project and project management:

- There's no focus on creating value.
- This concept lacks the scientific method of hypothesis-experiment-evaluation and continuous improvement. Plans are considered sufficient for success if they're followed, and the purpose of control is to bring the performance back to conform to the plan. This is well-illustrated by the plan-execute-control loop of traditional project management, compared to the plan-do-check-act cycle of the scientific method which considers plans (and projects in general) to be a hypothesis that should be validated.
- The transformation concept assumes that by local optimization (optimizing tasks) one can optimize the whole (the project). In addition, the focus on transformation ignores non-transformation phenomena present in projects. As a result, traditional project management doesn't consider waste elimination and the opportunity to improve effectiveness through flow management.

Koskela describes two other concepts that apply to projects – flow and value generation. He integrates the three models in a transformation-flow-value generation model of production, which he explains also applies to construction and to product design and development (and in our opinion, to projects in general). The concept of flow recognizes that there are both transformational and non-transformational

stages. Thus, flow management strives to eliminate “non-value-adding phenomena”. In addition, flow management aims to reduce lead time, simplify processes, and increase flexibility and transparency. In value generation, transformations and flows are controlled to impact the resultant stakeholder value. [4]

All three components of the transformation-flow-value generation model are needed for successful project management.

- **Transformation management** is needed to create project assets that are instrumental in improving customer value streams and hence for generating customer value. In the **LeanPM** framework, we prefer to use "creation management" because it's more universal and applicable to all types of projects and because it shows the purpose (the end state) of the transformation. Also, we extend creation management to manage absorption – the process of integrating project assets into the customer value stream.
- **Flow management** is required to enable effective, efficient and predictable creation and absorption. For instance, flow optimization results in waste reduction, better quality and shorter project cycle time. Flow management is an enabler of creation and absorption and a means to increase the project's net value as it reduces costs.
- **Value generation management** is needed to realize the purpose of projects – creating net value for project stakeholders and society. Also, it's needed to eliminate strategic waste and other forms of major project waste. Value generation management must be embedded in all project actions and decisions.

The transformation of project management in the 21st century was strongly influenced by Donald Reinertsen through his work on lean product development.

He outlines twelve key problems with the traditional product development that we believe apply to project management orthodoxy: [5]

1. Failure to Correctly Quantify Economics
2. Blindness to Queues
3. Worship of Efficiency
4. Hostility to Variability
5. Worship of Conformance
6. Institutionalization of Large Batch Sizes
7. Underutilization of Cadence
8. Managing Timelines instead of Queues
9. Absence of Work-In-Process Constraints
10. Inflexibility
11. Noneconomic Flow Control
12. Centralized Control

The solution to these problems lies in the domains of value and flow management.

We shouldn't consider creation, flow and value generation management as independent, but as integrated processes guided by the project's life cycle benefits and costs.

In the end, every project is about creating net value so the value generation model may seem sufficient. But if we only use that, we would work at a high level of abstraction. The creation and flow concepts are helpful for managing the means to achieve the end.

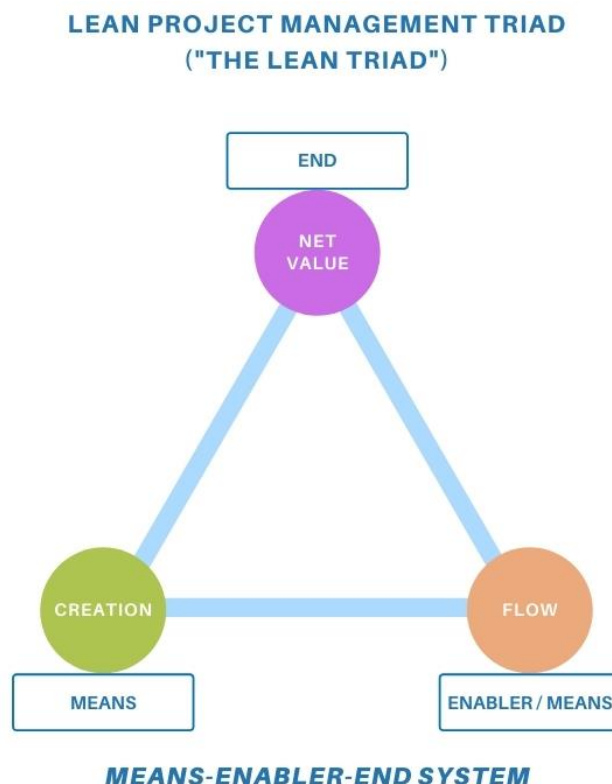
For example, managing **cost of time** is a component of value generation management because it affects a project's net value. However, to reduce the cost of time, it may be necessary to shorten the project's cycle time by improving creation and flow management processes.

Looking at it from another angle, changing the schedule priority of a project is a component of value generation management, as it can affect the cost of time and the net value of the project. But changing the priority can also affect the creation process and the team's ability to manage flow which would affect the overall benefits and costs of the change.

When we look at the relationship between creation and flow, we can see that flow is an enabler of creation besides being the means to achieving net value generation. So, there's a *means-enabler-end* association between the three lean project management components.

Let's take the example of the Lean-Agile practice of built-in quality. The goal of this practice is to generate net value (*end*). We perform the quality building activities as a part of deliverable creation process (*means*). Flow management enables the creation process to build quality into deliverables by the use of pull, small batches of work, quick feedback, queue control, visualization and work-in-progress limits (*enabler*).

This system of means-enabler-end is what we call Lean Project Management Triad or the **Lean Triad**.



LeanPM® Framework

© Lean Project Management Foundation

The creation-flow-value generation/Lean Triad model doesn't ensure that we use the full potential of lean in project management. For example, the model is not necessarily associated with continuous improvement. Therefore, we need to complement it with explicit lean project management principles.

We have defined "project" as a temporary value stream and "project management" as a practice of managing project value streams to maximize their net benefit. While these definitions describe the essence and the purpose of projects and project management, the *creation-flow-value generation model*, complemented by the lean project management principles, is the theoretical foundation that provides guidance for practical implementation of lean project management.

To summarize, the Lean Project Management (**LeanPM**) concept has two pillars:

1. The creation-flow-value generation model of project management which integrates Creation Management, Flow Management and Value Generation Management and is reflected in the means-enabler-end system of the Lean Triad.
2. The Lean Project Management (**LeanPM**) principles.

THE PURPOSE OF LEAN PROJECT MANAGEMENT

The purpose of Lean Project Management, based on its two pillars, is:

1. A significant improvement in project success rate
2. A considerable increase in the net value gained by project stakeholders
3. Fostering society's progress and prosperity

We can achieve these effects by enabling people to create value and continuously improve their work, which requires a comprehensive change in all aspects of project management.

The **LeanPM** Framework describes the principles, processes and practices that can fulfill the purpose of Lean Project Management. It's not intended to be a prescriptive framework as it complements and can be used with other project management and agile development methodologies, frameworks and practices.

What's the relationship between Lean management and Agile?

"You don't do agile or lean you do agile and lean."
Martin Fowler

Lean and Agile share a common evolutionary, adaptive, and people-centered mindset and Agile is influenced by Lean. In Martin Fowler's words: "if you are doing agile you are doing lean and vice-versa" [6]. That's why the **LeanPM** project management framework was not conceived as a substitute for Agile frameworks but aims to make a more explicit and universal use of lean management principles and ideas.

Organizations and project teams will want to experiment and innovate to create their own, context-specific project management method. In doing so, they can use the **LeanPM** ideas in a way that's appropriate to their particular situation.

REFERENCES

- [1] Koskela, Lauri (2000). *An exploration towards a production theory and its application to construction*. Espoo, VTT Building Technology. VTT Publications; 408
- [2] Koskela, Lauri and Howell, Greg (2001) *Reforming project management: The role of planning, execution and controlling*. Proceedings of the 9th International Group for Lean Construction Conference. Kent Ridge Crescent, Singapore, 6-8 August 2001. Chua, David and Ballard, Glenn (eds.). National University of Singapore (2001)
- [3] Ibid.
- [4] Koskela, Lauri (2000). *An exploration towards a production theory and its application to construction*. Espoo, VTT Building Technology. VTT Publications; 408
- [5] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
- [6] <https://martinfowler.com/bliki/AgileVersusLean.html> (last accessed 2021-11-04)

Chapter 1: Lean Concepts

LEAN CULTURE

Lean, as modeled on the Toyota Way values, has two pillars and five principles [2]:

- **Continuous Improvement**, which includes *Challenge*, *Kaizen* and *Genchi Genbutsu*
- **Respect for People**, which includes *Respect* and *Teamwork*

Continuous Improvement

*"Being satisfied with the status quo means
you are not making progress"*

Katsuaki Watanabe, CEO of Toyota Motor Corporation

Processes, products and services are never perfect, and they must be relentlessly improved, indefinitely. Organization's continuous improvement is based on building and continuously improving people's capabilities, and thus the two pillars of Lean are interconnected.

To achieve their long-term vision, lean organizations take up challenges with courage and creativity. They teach people ways to overcome obstacles and empower them to attain more, in a better way. They embrace change and challenge the status quo, as this is the only way to make progress.

Kaizen translates as *change for good* or *improvement* (*kai* – "change" and *zen* – "good"). This is a mindset and practice of continuous improvement through innovation and evolution.

Kaizen requires continuous small improvements initiated by employees. All employees should continually seek ways to improve their individual performance and should team up with others to improve team performance. Kaizen encourages employees to take ownership for their work and improves their motivation. As a practice, *kaizen* is a bottom-up approach for incremental improvement that is complemented by the top-down approaches of *kaikaku* (radical change) and *kakushin* (radical innovation).

But the most important aspect of *kaizen* is that it is a way of thinking that helps everyone in the organization to challenge everything and see opportunities for improvement everywhere.

*"The spirit of kaizen [is] reaching higher and challenging ourselves
to find a better way in everything we do, every single day"*

Toyota Global Vision [3]

Genchi Genbutsu (onsite hands-on experience, or "go and see") is a principle to make correct decisions, based on the facts found at the source – where work is actually performed. Also, this principle requires building consensus on the decisions to achieve goals at the best speed.

Good understanding of the situation on the site where value is added and waste is visible is crucial to problem-solving and improvement. In a broader sense, *Genchi Genbutsu* requires decisions to be based on facts and good understanding, not on assumptions and untested hypotheses. (An example of *Genchi*

Genbutsu in project management is the need to convert critical project assumptions to hypotheses that need to be tested.)

Respect for People

"Lean isn't lean if it doesn't involve everyone"

John Shook

Respect and building people are the heart of Lean. Toyota "makes" people first, and then people make cars.

Respect for people applies equally to employees, customers and partners. In a broader sense, it also applies to society. The lean organization respects others, builds and encourages mutual understanding, responsibility-taking and trust. The organization encourages personal and professional growth and shares the opportunities for development.

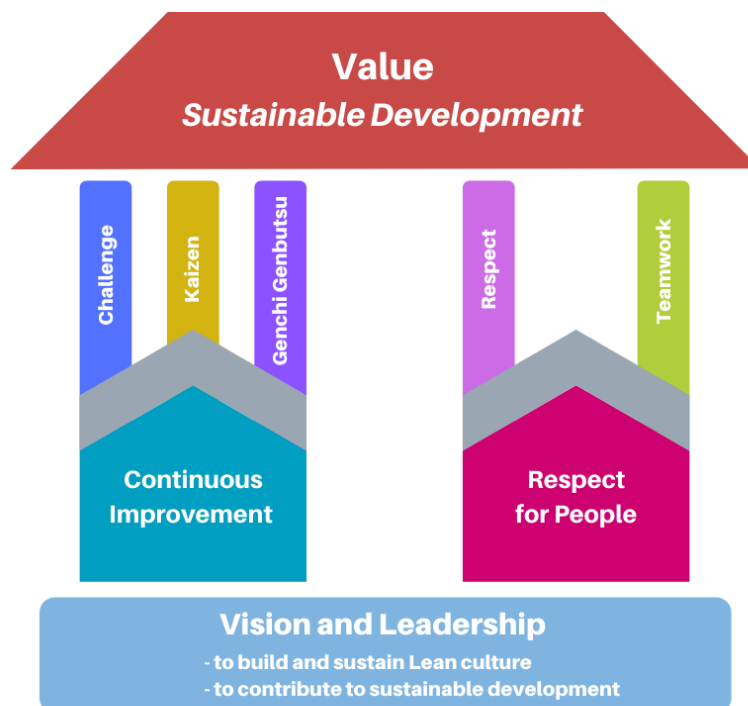
Respect for people is widely reflected in the philosophy and values of Toyota.

"Foster a corporate culture that enhances both individual creativity and the value of teamwork, while honoring mutual trust and respect between labor and management"

Guiding Principles at Toyota [4]

The House of Lean

The Lean culture of the Toyota Way can be presented as a House of Lean:



LeanPM® Framework

© Lean Project Management Foundation

LEAN MANUFACTURING

Lean manufacturing is a system "*based on the philosophy of the complete elimination of all waste in pursuit of the most efficient methods*" [5]. Toyota Production System (TPS) is the source and model for lean manufacturing. Many lean concepts that are now being applied in a variety of sectors and areas, including in project management, originate from TPS [6].

TPS is a product of many years of innovation and continuous improvement with the goal of achieving:

- short lead time
- high efficiency
- ability to produce a variety of cars, one at a time
- sound quality and full customer satisfaction

TPS is based on the concepts of *jidoka* (automation with a human touch) and *Just-in-Time*.

Jidoka

Jidoka is a principle which requires that, when a problem is detected, the equipment should come to a safe stop immediately, to prevent the production of defective products.

Conventional automated equipment is complemented by human intelligence functionalities which enable it to check quality, stop and signal when it detects a problem. The machines are linked to synchronize the upstream operation with the downstream operation (pull system), instead of pushing items to the next machine on the production line (push system) [7].

It all starts with manual work and people's skills and craftsmanship.

First, the engineers carefully build the line components in compliance with the standards. This is done **by hand**. Second, they apply *kaizen* to simplify the line's operations in an incremental way. The goal is that any line operator can consistently achieve the same result (when human operators cannot add value to the operations) and then the created and improved components are embedded into the operational production line.

This process is repeated continuously in order to:

- simplify the equipment and reduce its cost
- reduce equipment maintenance cost and time
- create simple, compact and flexible lines that adapt to fluctuations in production

The core of *jidoka* is the interdependence and the continuous improvement of both technology and the skills of the people, based on human wisdom and talent. Machines and technologies advance as people transfer their skills and workmanship to them, and this transfer is done through manual work.

Here's how *jidoka* is applied to prevent quality problems and to facilitate daily improvements:

1. When a machine detects a problem, it communicates an abnormality (the machine is engaged in automation). The operator must also self-inspect their work, as well as the previously

produced work, and take action when a defect is found. A visual and/or audio alert is activated by the operator or by the machine itself.

2. The operator stops the line.
3. The supervisor immediately attends to the problem, identifies the root cause and removes it.
4. The process is improved to eliminate the possibility of the problem occurring again.

Just-in-Time

The concept underpinning the Just-in-Time process is based on making only "what is needed, when it is needed, and in the amount needed" [8].

The goal is to achieve a continuous flow and fill a customer's order in the shortest possible time by doing only what is needed to perform the next process. For vehicle production, Just-in-Time entails the following:

- When an order is received, instructions to start production are issued to the assembly line as soon as possible.
- The assembly line must have a minimal inventory of parts to enable the production of any kind of vehicle.
- Once an order is filled, the assembly line orders replenishment parts from the parts-producing processes.
- The parts-producing processes must keep a small inventory of all types of parts, and produce only the quantities needed to replace what is pulled for the next process.

Just-in-Time aims to ensure efficient production of quality products with short lead time. This is made possible by elimination of *muda*, *mura* and *muri*.

Muda (waste) is any activity or step that does not add customer value but absorbs resources.

Mura translates as *unevenness* or *lack of uniformity*. When customer demand fluctuates, production cannot be carried out at a constant rate. Uneven production generates waste. Leveled production is more efficient and allows continuous flow. Production leveling is achieved through mixing small batches of different car models on the assembly line. Each model has a different production lead time, which makes it possible to have various combinations of sequenced models (in various but always small batch sizes) resulting in the same level of production.

Muri (unreasonable requirements; beyond one's power) is another source of waste. *Muri* is pushing people or machines to work beyond their natural limits, which creates safety and quality problems, causes breakdowns and constrains flow [9].

THE 14 PRINCIPLES OF THE TOYOTA WAY

Lean is a system whose elements must be practiced consistently.

Based on his 20 years of studying Toyota, Jeffrey K. Liker describes 14 principles of the “Toyota Way” which reinforce the principles documented by the company and provide a wider system framework for Lean [10]:

1. Base your management decisions on a long-term philosophy, even at the expense of short-term financial goals. [Long-Term Philosophy]
2. Create a continuous process flow to bring problems to the surface. [Continuous Flow]
3. Use "pull" systems to avoid overproduction. [Pull Systems]
4. Level out the workload (Heijunka). (Work like the tortoise, not the hare.) [Leveled Workload]
5. Build a culture of stopping to fix problems, to get quality right the first time. [Stop and Fix]
6. Standardized tasks are the foundation for continuous improvement and employee empowerment. [Standardized Tasks]
7. Use visual control so no problems are hidden. [Visual Control]
8. Use only reliable, thoroughly tested technology that serves your people and processes. [Good Technology]
9. Grow leaders who thoroughly understand the work, live the philosophy, and teach it to others. [Grow Leaders]
10. Develop exceptional people and teams who follow your company's philosophy. [Develop People]
11. Respect your extended network of partners and suppliers by challenging them and helping them improve. [Respect and Help Partners]
12. Go and see for yourself to thoroughly understand the situation (Genchi Genbutsu). [Go and See]
13. Make decisions slowly by consensus, thoroughly considering all options; implement decisions rapidly. [Slow Decisions by Consensus]
14. Become a learning organization through relentless reflection (Hansei) and continuous improvement (Kaizen). [Learn and Continuously Improve]

Note: we need to clarify two common misunderstandings:

1-Slow decisions

The true meaning of “slow decisions” is to decide at the last possible moment, without being irresponsible, instead of making (and locking in) decisions too early with poorly tested solutions. However, this does not mean that decisions should be delayed (and the morale of the team ruined), but that various options should be explored, studied, tested and evaluated until it is absolutely necessary to take a decision in order to proceed further. This is a manifestation of the *genchi genbutsu* principle to base decisions on good understanding.

Nevertheless, in this framework we prefer to use the concept of the "most responsible moment" - the moment that maximizes the net benefit of the decision.

2-Standardized work

Management does not mandate the work standards. It is the team that should develop and adopt them. Standards do not describe the best way to do the work *always*, but the best way known to the team *at present*. The team follows the standards, knowing that there are better approaches which they have to discover. After the team defines the standard, they master it to explore its full potential and to reveal its shortcomings – and they constantly challenge and improve it through *kaizen*. They measure the improvement against the standard, which serves as a baseline. Without such a baseline, improvement is not possible. The team introduces the improvement into the standard to create a new baseline and repeats the *kaizen* cycle, *ad infinitum*.

LEAN THINKING

Lean Thinking is a term coined by James P. Womack and Daniel T. Jones, and it represents the essence of the Toyota Production System. They summarize Lean Thinking in five principles [11]: 1) specify value by specific product or service, 2) identify the value stream, 3) make value flow, 4) let the customer pull value, and 5) pursue perfection.

Let's illustrate these principles with an example. "John Barleycorn" is a British folk song in which the main character is a personification of barley – the main ingredient in the production of beer and whiskey. This is Robert Burns' version of the poem (we have omitted the last three stanzas and added an accent in bold):

John Barleycorn: A Ballad

Robert Burns, 1782

There was three kings into the east,
Three kings both great and high,
And they hae sworn a solemn oath
John Barleycorn should die.

They took a plough and **plough'd him down,**
Put clods upon his head,
And they hae sworn a solemn oath
John Barleycorn was dead.

But the cheerful Spring came kindly on,
And **show'rs began to fall;**
John Barleycorn got up again,
And sore surpris'd them all.

The sultry suns of Summer came,
And **he grew thick and strong;**
His head weel arm'd wi' pointed spears,
That no one should him wrong.

The sober Autumn enter'd mild,
When **he grew wan and pale;**
His bending joints and drooping head
Show'd he began to fail.

His colour sicken'd more and more,
He faded into age;
And then his enemies began
To show their deadly rage.

They've taen a weapon, long and sharp,
And **cut him by the knee;**
Then tied him fast upon a cart,
Like a rogue for forgerie.

They laid him down upon his back,
And cudgell'd him full sore;
They hung him up before the storm,
And turned him o'er and o'er.

They filled up a darksome pit
With water to the brim;
They heaved in John Barleycorn,
There **let him sink or swim.**

They laid him out upon the floor,
To work him farther woe;
And still, as signs of life appear'd,
They **toss'd him to and fro.**

They **wasted,** o'er a scorching flame,
The marrow of his bones;
But a miller us'd him worst of all,
For he **crush'd him between two stones.**

And they **hae taen his very heart's blood,**
And **drank it round and round;**
And still the more and more they drank,
Their joy did more abound.

Specify value by specific product

In the poem, value is defined from the perspective of the ultimate customer who uses a specific product – John Barleycorn’s brand of beer or whiskey. The value to the customer is the achieved (subjective) state of joy. Here, the more units a customer consumes, the more value they extract (“*and still the more and more they drank, their joy did more abound*”). But the value is customer-perceived, subjective, and highly contextual. For many people, this product would have no value or might even have a negative value for anyone who prefers a non-alcoholic drink.

Identify the value stream

The value stream is the sequence of all the steps necessary to create and use a product or service that brings value from concept to realization. In the poem, it includes all the steps from having the product idea and sowing barley (“*plough'd him down*”) to brewing or distilling the drink (“*have taken his very heart's blood*”) and drinking it. There are two types of activities in the value stream:

- Value-added – activities that add value to the product or service from the point of view of the customer.
- Non value-added – activities that absorb resources (including time and space) but, from the point of view of the customer, do not add value to the product or the service. Lean regards these activities (or steps) as *muda*, or waste.

Womack and Jones define two types of *muda* [13]:

- Type One *muda*: steps that create no value but are unavoidable with the technology used.
- Type Two *muda*: steps that create no value and are avoidable.

In the poem, we have showed the value-added activities in bold, for example “**he grew thick and strong**”, “[they] **cut him by the knee**” and “**he crush'd him between two stones**”. Omitting any of these activities will interrupt the value stream.

The following activities are examples of Type One *muda* - they do not add value to the product but we need them for creating the product because of the specific production technology we use: “[they] **tied him fast upon a cart**”, “**they laid him down upon his back**” and “**they hung him up before the storm**”.

Also, there are examples of Type Two *muda* - wasteful activities that we could avoid:

- “**they hae sworn a solemn oath John Barleycorn should die**” (though this is a good example of teaming up to achieve a common goal)
- “**they took a plough**” and “**they've taen a weapon**” (unnecessary movement of a tool)
- “**they filled up a darksome pit with water**” is likely a wasteful activity (unnecessary transportation of materials)

Repeating “**they toss'd him to and fro**” “**as signs of life appear'd**” too many times is also a *muda* (rework).

Type One *muda* is relative. On the one hand, these activities are necessary, but we can do them with fewer resources. Therefore, we reduce this kind of waste through refinement and simplification.

Type Two *muda* is absolute. It is pure waste, and we could eliminate it.

Make value flow

The goal is to eliminate the apparently wasteful steps and achieve an uninterrupted value creating flow, without delays and rework.

Two important factors affect the flow:

- The way we organize work – by function or by value flow
- The process of production – batch production or one-piece flow

In the poem, when the work is organized by function, there are “three kings” – the farmer, the miller and the brewer (or the distillery owner) – each one with their autonomous feudal domain. They strive to improve their own efficiency and effectiveness and therefore prefer to work in large batches of products. In this way, people and tools are fully utilized and constantly employed.

However, the inventories of raw material, work-in-process and finished goods are accumulated and put on hold until we use them in the next steps of the process. This interrupts the flow and the lead time increases. The inventory generates costs (waste) and hides quality problems – until we discover them in the downstream steps.

The alternative is to organize the work, not by functions, but by a continuous value flow and have the “three kings” and their teams collaborate within a single enterprise to achieve an uninterrupted process of creating value. The whole value-creation process becomes visible and transparent to all, and it can be improved.

To avoid the problems with large batch production, a one-piece flow can be introduced. One piece, or a small batch at a time, is moved through the steps of the process. In the poem, this small batch is John Barleycorn himself, representing the quantity of drink ordered by the customer. This is preferable to moving John Barleycorn together with his six brothers (or the whole Barleycorn family) through the process to create a large inventory and push finished goods sales.

For each step, large batches require a longer processing time than single units do. The next processing step cannot start until the whole batch is complete and quality problems get hidden within the batch. In contrast, single units move faster through the process and ensure a better flow of value. This results in reduced lead time and cost and improved quality.

Let the customer pull value

Pulling value means producing a product or providing a service which the customer derives value from, only when they order it. This “just-in-time” model is an alternative to the “just-in-case” model where organizations are overproducing in anticipation of higher demand – and then try to push sales.

The pull model avoids problems (and costs) associated with overproduction. Inventory and lead time are reduced and this improves efficiency.

Here's how the pull system works. The customer orders a bottle of beer at the pub. This triggers the whole process of production and delivery of beer. The bartender takes the bottle from the shelf and hands it to the customer. As the items run low on the shelf, the bartender replenishes them from the pub's small inventory. Once or twice a day, the pub owner orders new items from the wholesaler to replenish their inventory. The wholesaler delivers the items by the end of the day or by noon the next day.

At the end of each day, the wholesaler places an order with the brewery to replenish the products it has supplied to retailers throughout the day. The brewery filters and bottles the beer from their stock of conditioned beer and delivers the ordered quantity to the wholesaler by noon the next day. Thus, the pub, the wholesaler and the brewery keep only a small inventory buffer of bottled beer. Only what customers order is produced and delivered just-in-time.

Similarly, the brewery pulls empty bottles, labels and caps from its suppliers once or several times a day, *but* only for the quantity of bottled beer ordered by the wholesalers. Regarding plant-based raw materials, like barley and hops, the pull system will work with bigger quantities than actual real-time orders require, because of the long biotechnological cycle of beer production.

Pursue perfection

Perfection involves continuous improvement of the value creation system: improving value definition, designing better value streams, eliminating waste, improving value flow and developing better pull systems.

Lean Thinking in a summary

Lean Thinking is a mindset to deliver value efficiently to the clients when they need it, ensuring a continuous flow of value and constantly improving the value creation and realization system.

THE FORMS OF WASTE

Taiichi Ohno, creator of the Toyota Production System, defined seven types of *muda* (waste) [13]:

1. Waste of overproduction
2. Waste of time on hand (waiting)
3. Waste in transportation
4. Waste of processing itself
5. Waste of stock on hand (inventory)
6. Waste of movement
7. Waste of making defective products

Overproduction

Overproduction is producing more – and earlier – than needed. It is producing “just-in-case” instead of “just-in-time”. Ohno considers this the worst type of waste, which contributes to and helps hide other forms of waste [14]. Typically, we associate this waste with production in large batches, which creates costly inventory, increases waiting time, requires unnecessary transportation, and hides quality problems.

Waiting

Waiting prevents flow. A product is waiting when we do not work on it. A typical case is waiting in queues for the next step in the operation.

Transportation

The movement of products, materials or people between processes is a waste if it is not directly associated with value-adding activities. Besides transportation costs, excessive movement can cause damaged or lost products or materials, delays and stress, and may require additional space and equipment.

Over-processing (waste of processing itself)

Over-processing is adding more features to a product than the customer will use or requires. These features do not add value but cost more and take longer. It also refers to using high-capacity equipment that can create bottlenecks and extend lead time.

Inventory

Inventory of raw material, work-in-progress and finished goods generates warehousing, depreciation, shrinkage, insurance and lost opportunity costs. Inventory impedes the rapid identification of quality problems and extends the lead time.

Motion (waste of movement)

This waste occurs when employees or equipment unnecessarily move within a workspace. A typical example is looking for and reaching for a tool. Waste of movement is a productivity, quality, health and safety issue.

Defects

We associate this waste with the cost of poor quality, which has many components, from rework to lost business.

Examples of project waste

In the table below are examples of how the original seven forms of waste can happen in manufacturing, which we can translate into project waste. The idea is not to find direct correlation, but to illustrate the general concept. In the next chapters, we will discuss specific project wastes.

WASTE IN MANUFACTURING	WASTE IN PROJECTS
Over-production	Working on projects which are not aligned with the value stream system or the organizational strategy Working on unnecessary deliverables Working on unnecessary features
Waiting	Waiting for resources, information, approval, decision, feedback Waiting for an upstream activity to finish Interruptions
Transportation	Task switching Distant information flows Extended communication lines
Over-Processing	Over-planning Providing the customer higher quality than necessary Creating more iterations of a deliverable than needed Duplicate work Unnecessary meetings
Inventory	Working in large batches / Excessive work-in-process Inventory of projects on the waiting list
Motion	Extra efforts to find information Extra efforts to get a feedback
Defects	Cost of defective project outputs

Several other types of waste were added to the original seven. Womack and Jones added the waste of *goods and services which don't meet the needs of the customer* [15]. Liker added the waste of *unused employee creativity* which leads to losing “time, ideas, skills, improvements, and learning opportunities” [16].

Bicheno and Holweg described the following “new” wastes [17]:

- The waste of untapped human potential
- Excessive information and communication
- The waste of time
- The waste of inappropriate systems (e.g., order processing system)
- Wasted energy and water
- Wasted natural resources
- The waste of “No Follow Through” (when the resources and time saved are not used)
- Waste of knowledge
- The waste of empty labor (e.g., work time not used for doing work)

In various forms, all these and other types of waste are present in projects, and are discussed in more detail in the [Project Waste](#) chapter. We will not attempt to map the above wastes (especially manufacturing wastes) onto a project context, but will rather define project-specific forms of waste.

REFERENCES - CHAPTER 1

- [1] Toyota Code of Conduct (March 2006), TOYOTA MOTOR CORPORATION (https://www.toyota-global.com/company/history_of_toyota/75years/common/pdf/code_of_conduct.pdf) (last accessed 2021-11-04)
- [2] Toyota Way 2001; Toyota Way 2001 / Toyota Code of Conduct, TOYOTA MOTOR CORPORATION (https://www.toyotaglobal.com/company/history_of_toyota/75years/data/conditions/philosophy/toyotaway2001.html) - (last accessed 2021-11-04); https://global.toyota/en/company/vision-and-philosophy/toyotaway_code-of-conduct/)
- [3] Toyota Global Vision, TOYOTA MOTOR CORPORATION (<https://global.toyota/en/company/vision-and-philosophy/global-vision/>) (last accessed 2021-11-04)
- [4] Guiding Principles at Toyota, TOYOTA MOTOR CORPORATION (<https://global.toyota/en/company/vision-and-philosophy/guiding-principles/>) (last accessed 2021-11-04)
- [5] Toyota Production System, TOYOTA MOTOR CORPORATION (<https://global.toyota/en/company/vision-and-philosophy/production-system/>) (last accessed 2021-11-04)
- [6] This section (Lean Manufacturing) is based on Toyota Production System, TOYOTA MOTOR CORPORATION (the link is as above) (last accessed 2021-11-04)
- [7] Koichi Shimokawa and Takahiro Fujimoto, Editors (March 2009). *The Birth of Lean, Chapter II What I Learned from Taiichi Ohno: A talk by Michikazu Tanaka*. Lean Enterprise Institute
- [8] Toyota Production System, TOYOTA MOTOR CORPORATION (the link is as above) (last accessed 2021-11-04)
- [9] Liker, Jeffrey K. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill
- [10] Ibid.
- [11] Womack, James P. and Jones, Daniel T. (2003). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Free Press; 2nd edition
- [12] Ibid.
- [13] Ohno, Taiichi (1988). *Toyota Production System: Beyond Large-Scale Production*. Productivity Press
- [14] Ibid.
- [15] Womack, James P. and Jones, Daniel T. (2003). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Free Press; 2nd edition
- [16] Liker, Jeffrey K. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill

[17] Bicheno, John and Holweg, Matthias (2016). *The Lean Toolbox, 5th edition. A handbook for lean transformation*. PICSIE Books

Chapter 2: Lean Project Management (LeanPM) Principles



LeanPM® Framework

© Lean Project Management Foundation

Lean Project Management (LeanPM) can only be fully utilized by an organization that builds and sustains a lean culture in its quest to become a Lean organization, that is one that continuously provides net value to its stakeholders in a continuously improving way. Given this important condition, **LeanPM** principles describe the philosophy and mindset necessary for project success. The principles are not independent. They reinforce each other and work as a system.

The Lean Project Management principles are one pillar of **LeanPM**, with the other pillar and theoretical foundation being the creation-flow-value generation model of project management. Read here about the [concept and pillars of Lean Project Management](#).

The six lean project management (LeanPM) principles are:

- Serve People
- Create Value and Eliminate Waste
- Build Knowledge and Continuously Improve
- Apply Systems Thinking
- Communicate and Collaborate Effectively
- Simplify

SERVE PEOPLE



LeanPM® Framework

© Lean Project Management Foundation

This principle is at the heart of **LeanPM**. Projects benefit people, and people undertake projects. That is why this principle has two aspects: "for people" and "by people".

"For people"

To serve people is the only reason for a project to exist. Projects should serve customers, society members, employees, owners and team members. They need to receive specific benefits, depending on their particular interests and needs. These interests should not be seen as conflicting. In fact, they are mutually supportive. It is perfectly possible to align the interests of these stakeholders, and the project should aim for that.

For example, the interest of a business organization and its owners in profiting from a project is realized by providing value to the customer who pays for the project products. To build a sustainable business, the organization must deliver value to the customer base on which the business depends. The client and the organization reward project team members in various ways for creating value for them.

Therefore, the most important question for any project is:

"How does this project serve people?"

"By people"

Projects do not happen just because:

- we have a brilliant idea
- the project is aligned with the strategy
- the top management supports the project
- we have secured funding for the project
- we have a good project plan
- the client is willing to pay for the products
- ... etc.

All these things are important. But what turns a potential project into a real project is the people who work on it. They breathe life into the project, create its deliverables, and deliver benefits.

Projects are "human beings":

- Success or failure of a project depends only on people.
- People create the whole value of a project.
- People's problems are at the root of all project problems.
- People make all decisions and all mistakes in a project.
- People build quality in.
- The untapped potential of people is the root cause of all project waste.

Therefore, the most crucial question of project management is:

"How can people's potential be unleashed so they can make a project successful?"

From this point of view, we should value any project management method, practice, technique or tool only in terms of its ability to facilitate and serve people working on and benefiting from a specific project.

CREATE VALUE AND ELIMINATE WASTE



LeanPM® Framework

© Lean Project Management Foundation

The total project value is a combination of the following values:

- Value for the customer
- Value for the project owner (the sponsoring organization)
- Value for the project team members and project partners and
- Value for the society

We measure the success of a project by its net value. Projects that do not create value create waste. Projects that create value that is less than the costs they generate have a negative return, which is also a manifestation of waste.

To increase the net value of the project and the return on investment, we need to:

- increase the value created by the project
- reduce project costs by eliminating waste

Project value is something beneficial, useful, worthwhile or important to a project stakeholder. The value for each project stakeholder is a function of the customer value resulting from a customer's appreciation of project outputs.

Value is a subjective perception of a positive change in one's status or state. It's often pointed out that customers place value on project outputs through their willingness to pay for them. But for non-profit projects, we should replace the willingness to pay with a willingness to use/consume project outputs.

The project translates the client's willingness to pay and/or use the project's outputs into certain characteristics, such as features, price, quality, time and place. These characteristics objectify the value, but we must remember that value always has subjective components and many aspects that we cannot easily objectify.

Let's assume a customer pays for and uses a project product. Has the customer realized the value of the product? Not necessarily. This will only happen if the customer's status or state has changed as desired because of using the project product. In other words, if the project product is used in the customer's value stream, which generates the (expected) benefits for the customer. The expected value (the value paid for) and the realized value may differ.

The project value for an organization is measured by economic return (profit), or social return on investment for non-profit projects. The organization receives value from the project by delivering value to its customers. Value is delivered either directly from the project to external customers who use the project deliverables, or to the organization's customers who use its products and services. In the latter case, the project deliverables are used within the organization's value stream system. Therefore, the focus of any project should be to create value for the ultimate customer.

A project that creates value:

- is based on a complete understanding of customer value
- is aligned with the organizational strategy and the value stream system
- is designed to create and deliver value effectively
- applies value-based metrics
- fully engages the customer and adapts to changes in the needs and the perception of value
- manages value delivery
- continuously tests and validates value hypotheses

Waste in a project is anything that does not add value to the customer – and hence, to the project owner organization – but absorbs project resources. The greater the waste, the lower the return on project investment. As we will discuss in the next chapters, the waste can lead not only to lower return but even to a negative return, which may go below minus 100% of the initial investment. Therefore, one of the most important tasks of project management is to identify and eliminate project waste.

BUILD KNOWLEDGE AND CONTINUOUSLY IMPROVE



LeanPM® Framework

© Lean Project Management Foundation

Knowledge building and continuous improvement are closely related and play an important role in project success.

Knowledge is understanding of something; knowing the facts, circumstances, relationships and behaviors associated with the project is important. What makes it even more important is that continuous improvement is only possible when we base it on knowledge intentionally created for this purpose.

Projects undertaken under uncertainty should be viewed as knowledge-creating initiatives. We should base them on hypotheses that we should test and validate. If we prove a hypothesis to be invalid, we must change the project based on the knowledge gained. The knowledge creation continues with definition and testing of new hypotheses. Each cycle of knowledge creation should improve the net value of the project. Similarly, stakeholder feedback should be used on product increments to build knowledge, to improve project value.

Knowledge alone is not sufficient to judge, decide, and act. It works in a system together with learning, skills and wisdom.

Knowledge-building refers to creating, transmitting and keeping knowledge. In the context of the project management process, knowledge about successful and unsuccessful practices for improving the process needs to be created. We create such knowledge through the application of a defined process that is continuously subject (after we establish a baseline) to change through experiments. We measure the success of the experiments by the change in the net value of the project. Through the experiments, we create knowledge that serves to improve the process and create a new baseline, which is challenged in its turn.

Explicit knowledge can be formalized and codified in written, verbal or audio-visual form. This makes it easier to pass on to others.

Tacit knowledge, on the other hand, is gained from personal experience and is expressed as intuition, subjective judgment, or insight. It cannot be formalized, and it is difficult to transmit.

"We know more than we can tell."

Michael Polanyi

The following may be useful to retain explicit and tacit knowledge:

- Establish a culture of knowledge-creation and sharing
- Foster social interaction
- Mentoring
- Use of cross-functional project teams and on-the-job knowledge sharing
- Osmotic communication
- Formalizing knowledge in guides, manuals, tutorials, lessons learned, standard work processes, audio-visual materials

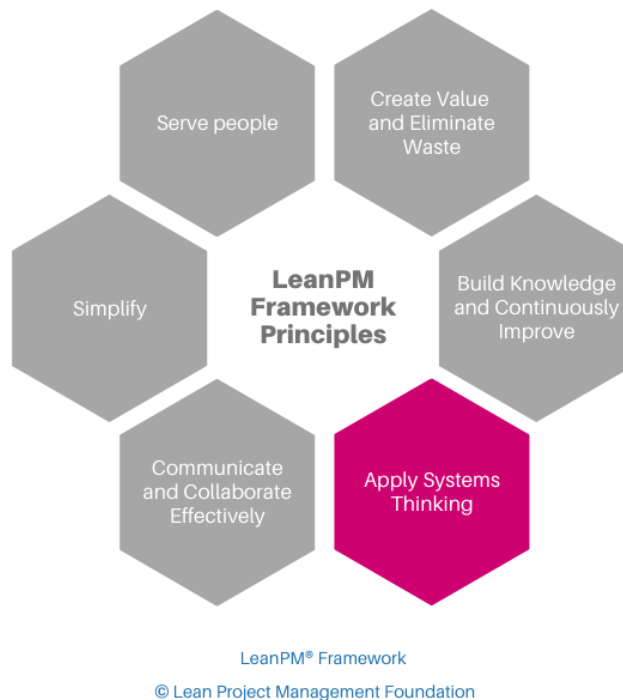
There are two aspects of continuous improvement related to project management.

The first aspect refers to the improvement of the project management process. In lean project management, the areas subject to continuous improvement are:

- Customer value definition and project alignment
- The design of the temporary value stream (project design)
- The customer pull system
- Waste elimination
- The flow of project value
- Plan-do-check-act processes
- Teamwork

The second aspect relates to the organization's continuous improvement. A considerable part of it comprises daily improvements. Other improvements require greater effort and management structure, and we implement them through projects. The success of these projects plays a key role in continuous improvement of the organization.

APPLY SYSTEMS THINKING



A system is a collection of entities and links between them that form a whole which is more than the sum of the individual entities. There are several project-related systems:

- the organization that owns the project
- the extended enterprise that includes the organization and its customers, suppliers and partners
- the value stream systems of the organization and customer
- the individual value streams (the value streams of each product and service)
- the project itself
- the configuration of project deliverables
- the project team

System thinking requires that:

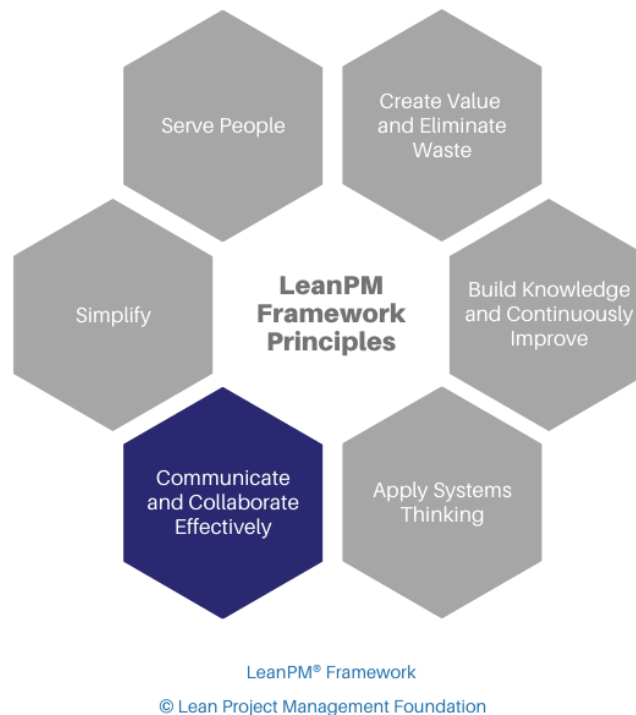
- We should not try to analyze the components of the system independently.
- We should not attempt to improve individual components, but rather the whole system.
- We must account for the interrelations between the components of the system. A change in one component may affect other components or the whole system.
- We should create systems whose components fit well together.
- We should be conscious of system dynamics.

For example, we should not aim to improve the productivity of individual team members or specific parts of a value stream, but the productivity of the team and the whole value stream, respectively.

Human behavior and mental models are the major factors that affect the performance of project-related systems.

Unless project management takes into consideration the systemic character of projects, many projects are doomed to failure.

COMMUNICATE AND COLLABORATE EFFECTIVELY



Undoubtedly, inefficient communication and collaboration are a major reason for project failure. They are at the root of any delay, cost overrun, quality issue, misunderstanding of stakeholder expectations, lack of project alignment, and waste.

Communication is the act of sharing meanings between people. The purpose of communication is to inform, express feelings and emotions, share ideas, thoughts and knowledge, and to influence. Through communication, people reach mutual understanding and facilitate each other's growth.

One cannot perform project management, and project stakeholders cannot co-create value without communication. To be effective, the project-related communication should be:

- Meaningful (should add value)
- Clear
- Transparent (everyone on the project should be able to see the work of the others, which will facilitate finding ways for improvement)
- Emphatic
- Respectful
- Timely
- Decentralized

... and as much as possible:

- Frequent
- Simple
- Informal
- Face-to-face
- Synchronous
- Visual

In a project context, collaboration is a process where stakeholders work together to co-create value for each party. Effective collaboration requires involvement of all stakeholders in creating project value from concept to realization. Those stakeholders should be empowered to own the project and motivated to bring project benefits into existence. Lean projects are not created *for* the customer, but created *with* the customer.

Co-creation continually engages all stakeholders in reaching mutual understanding and consensus on the shared goals and the value that the project will create for each party. This process relies on the collective wisdom and the full potential of people involved. Co-creation is facilitated by effective communication, servant leadership, collaborative and decentralized project management, and a self-organizing project team that includes the client and the partners.

SIMPLIFY

"Simplicity is the ultimate sophistication."

Leonardo da Vinci



LeanPM® Framework

© Lean Project Management Foundation

Project-related systems and processes tend to be complex. Complexity is a source of waste. Simplification removes waste.

Simplicity does not go beyond what is absolutely necessary to make something good enough.

Complexity does not have its own value. It is a complex way of doing something that we can do in a simpler way, to achieve the same result. What is being done has value, but not the way of doing it [1].

"Dealing with complexity is an inefficient and unnecessary waste of time, attention and mental energy. ... Complexity means distracted effort. Simplicity means focused effort."

Edward de Bono, Simplicity

The benefits of simplicity in project management are:

- Improved reliability
- Reduced risk
- Clarity of requirements, process of work, communication
- Better focus
- Better understanding

- Ease to change
- Ease to manage
- Greater flexibility
- Waste elimination

Along with several useful techniques, Edward de Bono suggests the following approaches to simplicity, whose application should not compromise “the unity of the overall purpose” [2]:

- breaking things down into smaller units
- decentralization
- modular design

Some practices of simplification which **LeanPM** recommends are:

- Minimum Viable Project
- Minimum Viable Product and Minimum Viable Value Stream
- Decentralization of project management
- Self-managed teams
- Elimination of the wastes of building unnecessary deliverables/features, over-processing and overcomplicating

Chapter 3: How Do Projects Work?

Project Success

PROJECT LOGIC

Organizations strive to provide a continuous flow of value through their operational value streams. To create and improve their value streams, they invest in specific initiatives (projects):

- Transformation initiatives to introduce a fundamental change in the organization's value stream system
- Initiatives to create new value streams involving new products or services
- Initiatives to improve the existing value stream system
- Initiatives to eliminate obsolete value streams
- Emergency response initiatives to preserve or restore the value stream system

These initiatives invest resources to create project products (deliverables) to be used in the organization's operations.

The project is a system of interrelated elements – resources, workflow activities, deliverables, objectives and goals – between which there are causal relationships (*Apply systems thinking* principle).

The purpose of project resources is to ensure the execution of workflow. The resources invested must be necessary and sufficient to carry out the workflow activities.

The purpose of project workflow is to create deliverables. Therefore, the workflow activities should be necessary and sufficient to create the deliverables.

The purpose of project deliverables is to get integrated into the operational value streams, improving the organization's value stream system (we refer to this improvement as project objectives). Thus, project deliverables, when used as intended, should be necessary and sufficient to improve the value stream system.

The purpose of the improved value stream system is to provide incremental value to the customers and the organization (goal of the project). The improved system must be necessary and sufficient to achieve the desired incremental value.

The deliverables are the **assets** created by the project. They can take different forms:

- Physical infrastructure or structure
- New or improved products or services
- Software
- Information products

- Document-based deliverables (e.g., textbook, contract, documented test results)
- New or improved processes, technology, methods
- New or improved organizations
- New knowledge and skills
- Result of a study, research or scientific experiment
- New natural resources
- Pieces of art
- Complex deliverables, comprising of deliverables of different forms
- Project management deliverables (e.g., plan, report)

When relevant project deliverables are integrated into operational value streams, this improves the organization's capability to deliver specific products and services. The improved operational value streams add value to the customers and the organization. The following figures demonstrate the project logic for improving a value stream and for creating a new value stream.

Figure 3.1: The project logic for improving a value stream

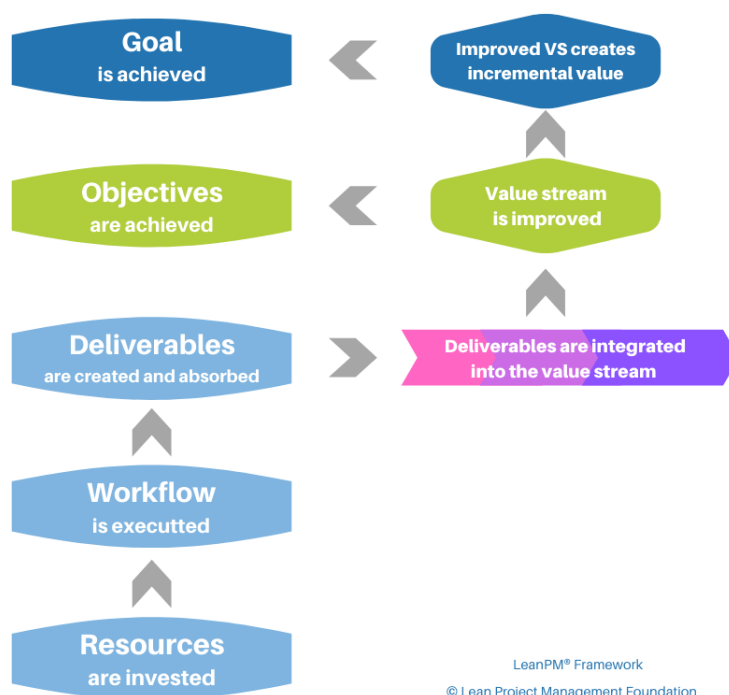
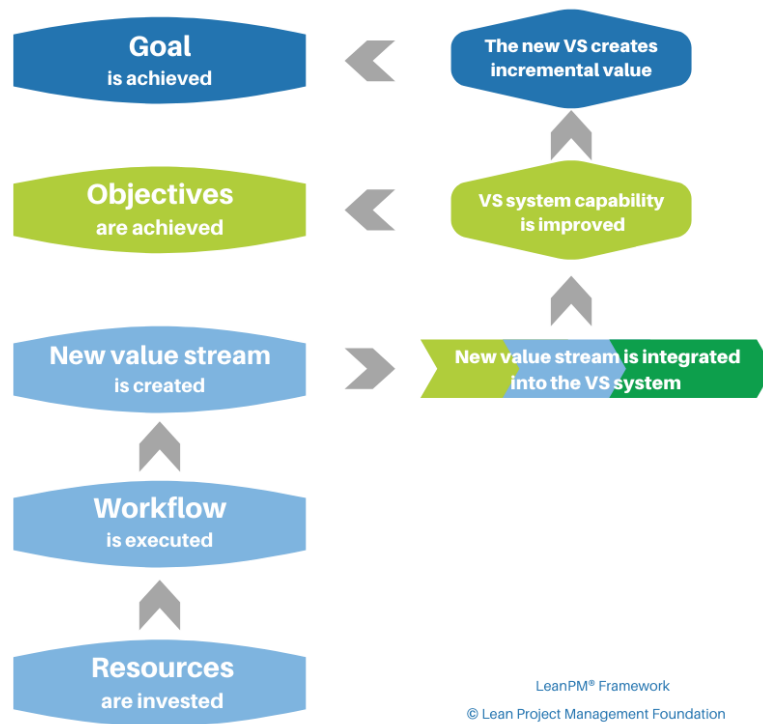


Figure 3.2: The project logic for creating a new value stream



PROJECT SUCCESS

Until we define project success criteria (which doesn't happen often), we can speculate endlessly about how successful it was, but we won't be able to measure success. When we've specified inappropriate criteria, we can delude ourselves that we've succeeded. Does it matter that we have met the time, cost and scope "objectives" if the project is a strategic waste?

The Measure of Success

Project success should be measured by its net benefits, provided that we meet the agreed project conditions (see Conditions of Satisfaction below). Any other metrics such as scope, time, cost and quality are proxy or secondary metrics - in fact, these are factors that affect the net benefits.

When comparing the benefits and costs of a project, we need to use the same units of measurement, and the most convenient ones are the monetary units. Therefore, as a synonym for the project's net benefits, we can use the economic or social life cycle profit - the difference between the project life cycle revenues and the life cycle costs, brought to the same point in time.

The project life cycle costs include:

- The project life cycle costs include:
- Investment costs (the costs for creating project assets)
- Operation and maintenance costs (including manufacturing costs, when applicable)
- Replacement costs and decommissioning costs (when applicable)
- Cost of Time (see the chapter on [Cost of Time](#))

The project life cycle revenues can take any of the following forms:

- Revenues from the sale of products or services created through a direct use of the project's deliverables, for example, proceeds from the sale of software or bridge tolls.
- Project benefits expressed in monetary units. Even the project benefits that have no market value (e.g., good health and well-being) can be expressed in monetary units. The need for quantification also applies to organizational achievements, such as improving the customer churn rate or lowering the employee turnover rate, or increasing customer and employee satisfaction.
- Avoided or reduced costs as a result of a project should also be considered as project revenue - for example, reduced customer acquisition cost or production cost (operational value stream costs).

Costs and revenues should factor in the cost of time. We should consider only incremental costs and benefits, that is, costs and benefits attributable to the project.

The benefit-cost measures like NPV and IRR are well known, so we won't look at them here.

We should assess all actions and decisions in a project for their net benefit, that is, for their contribution to the success of the project within the agreed Conditions of Satisfaction. However, it would be simplistic to say that we must strive to maximize benefits and minimize costs, since we have to account for the

trade-offs between their contributing factors. In this way, the success of the project becomes an optimization task.

Conditions of Satisfaction

Conditions of Satisfaction (CoS) are additional project success criteria based on priorities and conditions co-developed by stakeholders. The CoS align the interests of different parties, facilitate collaboration and guide decision-making that benefits everyone (Serve people principle).

CoS require a consensus-based agreement, containing both the conditions of individual stakeholders (owner, customer, team members, etc.) and the conditions common to all.

The agreed CoS may include, but not be limited to:

- Values
- Processes
- Collaboration
- Time, cost and quality
- Compliance

The CoS should be SMART: Specific, Measurable, Achievable, Realistic, Time-based, but we can also define other conditions with the help of states and feelings.

Here's a template for a value-based CoS, describing the future state resulting from the project:

As an owner/customer/end-user, I am/I feel <expression of state or status>; because <conditions of satisfaction>.

For example: "As a researcher (end-user), I can perform state-of-the-art, innovative and advanced research because I work in a world-class research facility."

CoS complement project objectives and the measure of success.

PROJECT VARIABLES

The project variables are agile factors that influence the success of the project.

Each project variable affects other variables and the outcome of the benefit-cost model needs to be optimized through a systemic approach (*Apply systems thinking* principle).

The key project variables that affect the outcome are:

- Revenues
- Investment costs
- Operation, maintenance, replacement costs and decommissioning costs
- Scope
- Quality
- Time (*Cost of Time*)

Let's look at some examples of relationships between project variables. Note that, when viewed through the lens of the benefit-cost model, these examples appear as local optimizations. The life cycle profit maximization requires a systems approach and holistic optimization modeling (*Apply systems thinking* principle).

Project Time-Cost Trade-off

The project cycle time and the investment costs are in a trade-off relationship. The optimum cycle time results from U-shaped distribution optimization.

While there's usually a linear relationship between the time and the indirect project costs, the corresponding relationship with the direct costs is nonlinear.

The total costs curve has three sections (Figure 3.3):

- a flat bottom of the optimum cycle time (lowest total costs)
- a left section with higher costs because of schedule acceleration
- a right section with higher costs because of project delay.

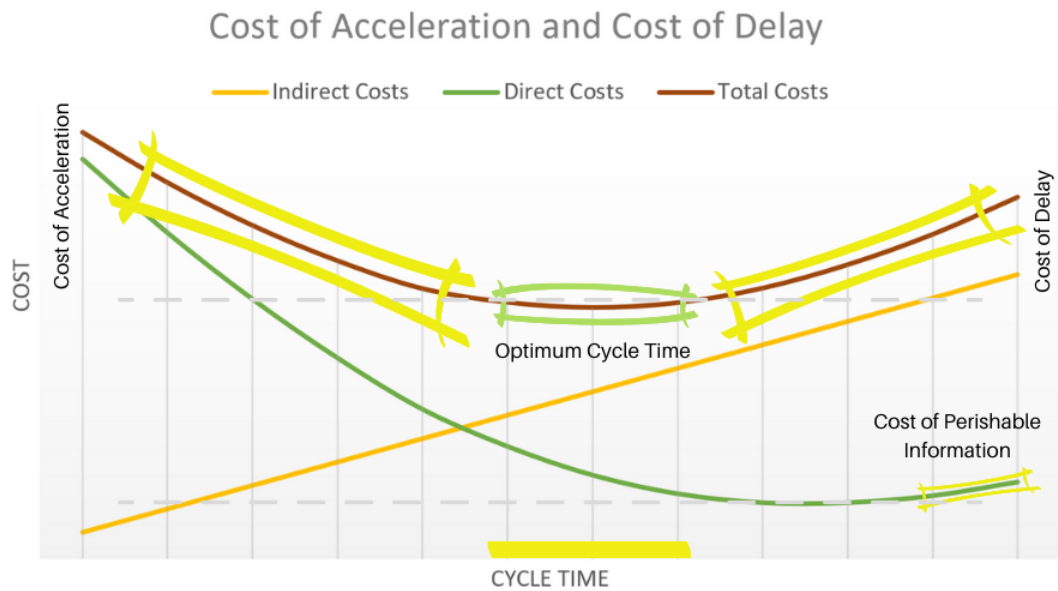
Direct costs don't go down indefinitely with the extension of cycle time, and it may seem logical that they reach a stable minimum. In practice, however, once they reach a minimum, direct costs rise again. The reason is that much of the project information is perishable and loses its initial value over time. This results in aging work items.

That's why, we "buy" new information through data collection, analysis, hypothesis testing, etc. The new information may require additional planning and development efforts to revitalize the aged items.

And there is another effect if the time cycle is extended. Team members have a limited working memory and, when a project is delayed, they lose or forget information relating to the delayed activities. They need extra effort to recover that information. Because of the cost of perishable information, the direct cost function is an asymmetric U-shaped curve.

The project acceleration and delay can be measured by using the optimum cycle time as a reference value. We show the acceleration and delay costs as well as the cost of perishable information in Figure 3.3.

Figure 3.3: Cost of Acceleration and Cost of Delay



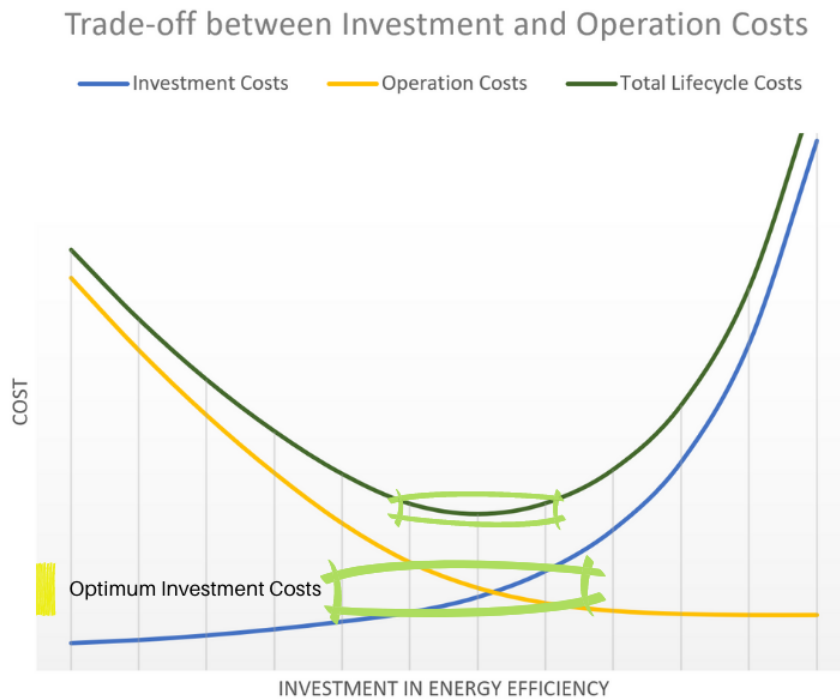
LeanPM® Framework

© Lean Project Management Foundation

Trade-off Between Investment and Operation Costs

It may be possible to decrease the project operation costs with a targeted increase of the investment costs, resulting in the minimization of the total life cycle costs. For example, additional investments in the energy efficiency of a building can lead to a reduction in its operation costs. This is another U-shaped distribution optimization as we show in Figure 3.4.

Figure 3.4: Trade-off between Investment and Operation Costs



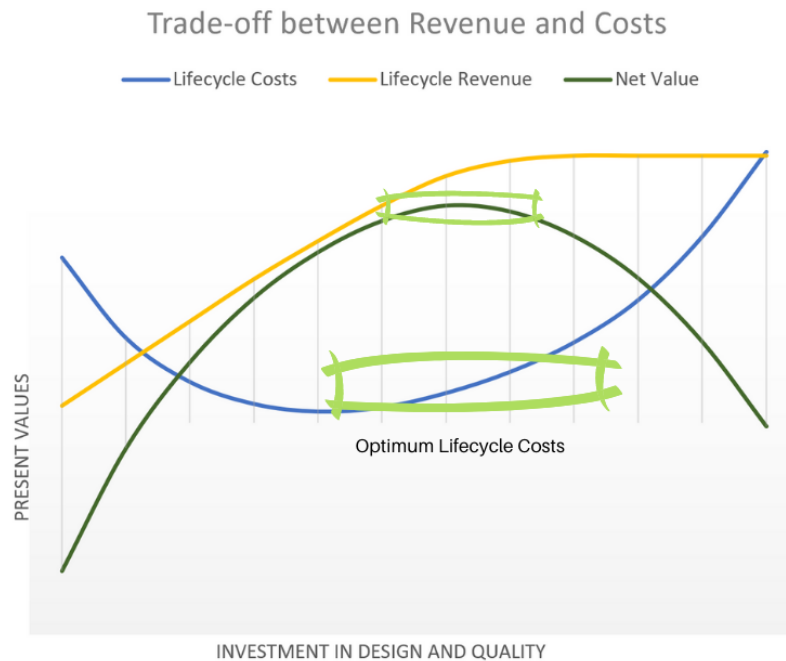
LeanPM® Framework

© Lean Project Management Foundation

Trade-off Between Project Costs and Revenues

Incremental targeted investments in the design and quality of project deliverables may increase the customers' willingness to pay for them (or use them) and the project revenues (Figure 3.5). A similar trade-off exists between the operation costs and revenues and between the project scope and the revenues. This is a normal distribution (bell curve) optimization.

Figure 3.5: Trade-off between Revenue and Costs



LeanPM® Framework

© Lean Project Management Foundation

TAKEAWAYS: HOW DO PROJECTS WORK? PROJECT SUCCESS

- A project is a temporary value stream, aimed at creating or improving the existing value stream system of an organization or its customers, thus creating increased value for the organization's stakeholders.
- Project management is the practice of managing project value streams to maximize their net benefit.
- To improve their value-creating system, organizations invest in specific initiatives (projects): transformation initiatives, initiatives for improving or creating value streams or for removing obsolete value streams, and in emergency response actions.
- The project is a system of interrelated elements that have causal relationships: resources, workflow activities, deliverables (project assets), objectives and goals.
- The most important measure of project success is the economic or social life cycle profit of the project (its net benefits).
- Conditions of Satisfaction for project stakeholders should complement the measure of project success.
- Variables that are in complex trade-off relationships influence project success. The net benefit can only be maximized with a systems approach and holistic optimization modeling.

Chapter 4: Project Waste

Lean project management must reduce and eliminate project waste.

According to the classical understanding, activities and steps that do not add value to the product or service from the point of view of the customer are waste. Thus, for example, all project meetings are considered a waste, but all activities that transform inputs into outputs are seen as value-added.

To avoid ambiguous interpretation and absurdities like "a necessary waste" and "a temporarily necessary waste", we take a different approach in the **LeanPM** framework.

Waste are those activities and steps whose costs outweigh the benefits and generate negative net value for the project. So, a project meeting or a transformation activity may or may not be a waste, depending on the benefits and costs it generates.

Waste can be **absolute** when a negative net value is generated or **relative** when the net result is positive but suboptimal.

The project is a hypothesis based on assumptions. This hypothesis must be continuously tested and validated. Invalid assumptions destroy the logic of the project and are a symptom of waste.

Waste of resources can occur at all levels of the project hierarchy and in any workflow activity:

- We waste the resources invested in activities that are not needed to create project assets.
- We waste the resources invested in creating project assets that do not improve the value stream system.
- We waste the resources invested in changing the value stream system when this does not result in incremental value. Here, there is no causal relationship between the project objectives and the goal of the project (or the assumed relationship is not realized).
- Any suboptimal use of project resources is also a waste (relative waste).

STRATEGIC PROJECT WASTE

Let's look at a project example.

Ambient Air Quality Improvement Project

The city council of a million-population city has implemented a project intended to improve the quality of the ambient air.

They have built a network of 22 monitoring stations to measure in real time the most common air pollutants (particulate matter, carbon monoxide, nitrogen dioxide, sulfur dioxide), together with air temperature, humidity and atmospheric pressure. The urbanized area of the city is over 5,000 km². Each station sends information to an open data system which uses cloud technology, analytical functionality and machine learning to facilitate timely action to improve the air quality. The system interface provides visual and machine-readable data, accessible via web interface and through smartphone and tablet applications.

According to city officials, the project applies new technologies (Internet of Things) to reduce the levels of fine particulate matter and improve ambient air quality.

They have completed the project. The sensors send real-time data to a database. The data is visible on a public website.



What could be wrong with using new technologies to improve air quality?

To create benefits, a project must:

- Create a new value stream
- or
- Improve an existing value stream
- or
- From a problem-solving perspective, directly solve a problem (which improves a value stream)

Let's examine these options for the air quality improvement project.

1. Solving a problem

The project does not eliminate any root cause of air pollution. Air quality monitoring only records the status.

2. Creating a new value stream

The project does not create (nor does it have the ambition to create) a whole new value stream, such as a "cleaner air" or "better health" value stream.

3. Improving existing value stream

There are several hypotheses:

There is a “clean air” value stream created and maintained by the organization of the project owner. The project improves this value stream.

There is an existing “clean air” value stream created by the self-organizing social system of the city, and/or the citizens have their own individual “clean air” value streams. The project improves these value streams.

The city council, the society and the individuals have value streams that are merged into a common “clean air” value stream. The project improves this aggregate value stream.

Hypothesis about an existing value stream operated by the city council which will be improved by the project

This hypothesis is based on several assumptions that must be validated. Some of these assumptions are:

- The city council operates a value stream that delivers clean air to citizens.
- Performance metrics for this value stream are defined. Improving value stream performance, as measured with the metrics, results in improved air quality.
- There is a mechanism for continuous identification, analysis and solving air quality problems, eliminating the root causes.
- The project was based on an analysis of the value stream. Various options for improving the value stream were considered, and the project is the most efficient and effective option.
- The air quality improvement project removes impediments to the effective functioning of the value stream.
- Data from the 22 stations for an area of several thousand square kilometers are statistically reliable and sufficient to enable decisions to improve the value flow.
- The data will be used to improve the value stream.
- Using the data will improve the efficiency of the value stream as measured with the performance metrics.
- The improved value stream efficiency will reduce air pollution.
- The air pollution reduction will be cost effective.
- There is a reliable mechanism for measuring the contribution of the project to reducing air pollution and it will be implemented.

If the project was justified by improving the existing value stream of the city council, but critical assumptions are invalid (or not validated), there would not be a sound reason to believe that the improvement would happen and **the whole project would be considered a waste**.

Hypothesis for existing citizens’ value streams

What is fundamentally different about this hypothesis is that it implies individual actions by the citizens – emission reduction measures or protective measures against air pollution effects, motivated or enabled by the project.

Some assumptions associated with this hypothesis are:

- People are aware of the project and know where to find the air quality information from the 22 stations.
- City residents will prefer the data from project measuring stations, rather than from the existing measuring stations of civic organizations (whose data the city authorities consider unreliable).
- City residents know how to interpret the data from the air quality measurement stations.
- Citizens will find a motivation to reduce their individual air pollution, based on information from the city's measuring stations, and will take appropriate measures.
- Individual measures to reduce air pollution will have a significant effect on overall air quality. This effect will be visible through the information from the 22 stations.
- The measures to reduce individual air pollution will be effective and cost-efficient.
- There is a reliable mechanism for measuring the effect of the project on the reduction of individual air pollution and the city council will implement it.
- The preventive measures taken will be timely (in real time, based on real-time information).
- The individual preventive measures will have a positive effect on the health status of people.

Similarly, if the project aims to improve the existing value streams of citizens, and the critical assumptions are invalid or are not validated, we could consider **the whole project a waste**.

The design of this air quality improvement project gives us reason to believe that it was not intended to be a means of improving value streams and that critical assumptions have not been converted into hypotheses to be tested and validated.

We can look at this project from different perspectives:

- From the perspective of problem solving, the project does not eliminate a root cause of air pollution.
- From the perspective of the systems approach, the project does not create a complete value stream.
- From the perspective of flow of value, the project does not improve value streams.
- From the perspective of value, the project does not create value for the customers (city residents).
- From the perspective of the declared goal of the project (improvement of air quality), the project does not achieve its goal.
- From the perspective of the return on investment, the project has a negative return.
- From the perspective of waste, all resources invested in the project are wasted.

This is an example of what we call a **strategic waste** – a complete waste resulting from the separation between the project and the value stream system that needs to be improved or from lack of alignment with the organizational strategy.

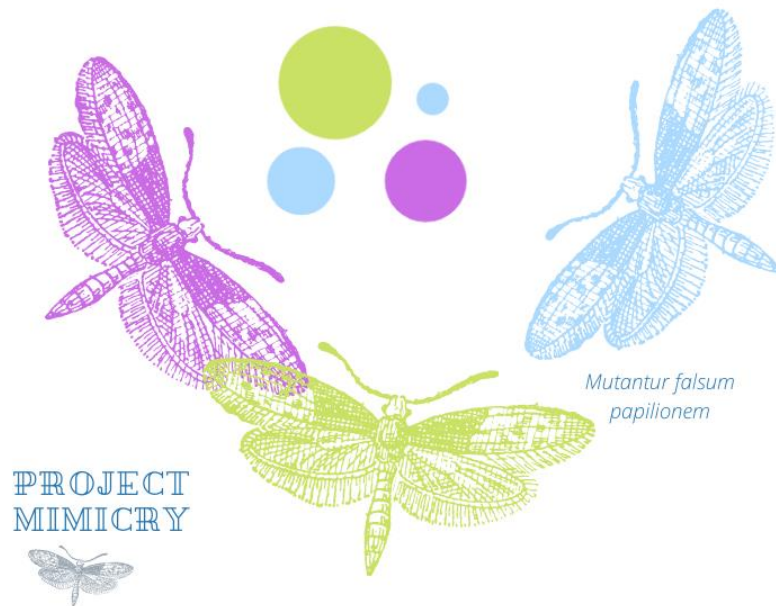
The whole investment is a waste, and the project is doomed to failure and doomed to waste from the very beginning. This waste cannot be overcome in the actionable phases because it stems from the ill-

conceived conceptual design of the project. Efficient and effective implementation will not add waste. Poor implementation of a strategically wasteful project will cause additional tactical waste.

The project in our example is no exception. Such strategically wasteful projects are common.

How is that possible?

The main reason is the misunderstanding of both the project logic and the need to integrate the project into the value stream systems. This mental model leads to wishful thinking and to what we call *project mimicry*.



LeanPM® Framework

© Lean Project Management Foundation

Project mimicry is the resemblance of a project that is separated from the value stream system, or not aligned with the strategy, to a project that is integrated into the value stream system or aligned with the strategy. The resemblance is achieved by unintentional or intentional definition of value-based goals and invalid assumptions, based on supposed but non-existent logical relationships, to justify a strategically wasteful project.

These are examples of value statements intended to justify the project:

- Ambient Air Quality Improvement Project (the name of the project)
- ... to improve the quality of the ambient air (project purpose)
- ... cloud technology, analytical functionality and machine learning to facilitate timely action to improve the air quality
- ... the project applies new technologies (Internet of Things) to reduce the levels of fine particulate matter and improve ambient air quality

We believe that, in this case and in most other cases, project owners use such statements unintentionally and inadvertently, which makes the problem even bigger because they don't realize it. They don't identify critical assumptions and do not convert them into hypotheses that they should test. The masking statements express supposed logical relationships that do not exist.

The waste does not end with the end of the project. Once the project is “successfully” completed, the project's products are put to use. This generates maintenance, support, replacement, and decommissioning costs that increase waste. We call this waste accumulation effect.

And this is not the end. The story of the waste continues.

To illustrate how strategic waste grows further, we will formulate a hypothesis about how the air quality improvement project is developing.

One year later, the city council secures funding to extend the "successful" pilot project by installing 30 additional measuring stations. The initial strategic waste is doubled, with new investment and additional maintenance, support, replacement, and decommissioning costs.

And no, this is still not the end.

The project is featured in media publications and reports as a success story and best practice. As a result, the project (and the waste) is replicated elsewhere.

By extending or replicating a wasteful project, the waste is successfully multiplied.

TACTICAL PROJECT WASTE

Tactical waste is waste that occurs regardless of whether the project is separated from the value stream system. It can appear in any project.

All forms of waste described in [Chapter 1](#) can be present in projects, too, depending on the project. In this chapter, we will discuss only project-specific waste.



LeanPM® Framework

© Lean Project Management Foundation

The major forms of tactical project waste are:

1. Wrong deliverables or features
2. Delays
3. Poor quality
4. Duplicate efforts
5. Lost productivity
6. Over-processing
7. Overcomplicating

Wrong deliverables or unnecessary features

Wrong deliverables or unnecessary features waste all the resources invested in them and may cause schedule overrun and added cost of delay. After the project, they generate handling and/or operating costs and accumulate additional waste.

Wrong project deliverables are those that:

- are not related to the objectives of the project, or
- are based on invalid assumptions about causal relationships with project objectives, or
- the customer does not use, or
- generate life cycle costs that are greater than their benefits.

We can summarize the reasons to work on wrong deliverables as faulty project logic and lack of customer involvement.

Unnecessary features of deliverables are those that:

- the customer does not use, or
- generate life cycle costs that outweigh their benefits.

Common reasons for creating unnecessary features are:

- A desire to provide the customer with more value through a more sophisticated product, to increase satisfaction and sales
- An effort to combine different ideas about the design of the deliverable or different use cases in a common set of features
- Overcomplicated product design
- Misunderstanding of customer requirements and lack of customer involvement
- Invalid assumptions about the value of features

Delays

For each project deliverable (or feature) there is a point in time – or a time period – at which the customer must have it available to maximize the net benefits over its entire period of use. This is the optimum timeframe for the project deliverable or feature to be delivered to and absorbed by the customer. Any delay beyond the optimum timeframe will cause additional cost to the customer. The *cost of deferred use* is among the most significant forms of tactical project waste. Ironically, earlier delivery (outside of the optimum timeframe) does not create additional value for the customer but results in what we call *cost of premature delivery* (learn more in [Cost of Time](#)).

A great variety of reasons can cause project delays, including:

- Waiting for resources, information, approval, decision, permission, feedback or an upstream activity to finish
- Multitasking and task switching
- Lack of prioritization and coordination

- Multiple team coordination problems
- Lack of capacity
- Lack of knowledge and skills
- Handoffs
- Working in large batches
- High-level inventory of work-in-process
- Idle time
- Interruptions and unnecessary meetings
- Process queues

Poor quality

The waste of poor quality constitutes the incremental costs that would not exist if quality of project deliverables and project management were good enough. The waste is the difference between the *cost of quality in a poor-quality scenario* and the *cost of quality in a good-quality scenario*.

Poor quality is not only about defects. The quality of project deliverables has two aspects:

- External quality, which we associate with fitness for purpose from the customer's perspective
- Internal quality, which we relate to the design of the product or service

The cost of poor external quality includes:

- Rework, including additional planning, inspection and testing
- Additional efforts to track defects
- Cost of delay – the lost benefits of delaying the use of a deliverable which can be a completely lost opportunity
- Costs to fulfil warranty and service-level agreement obligations
- Costs resulting from customer dissatisfaction, including loss of reputation
- Costs incurred by the customer -- lost opportunity, lost productivity, repair after warranty period, corrective measures

Poor internal quality does not necessarily affect the fitness for purpose. Low internal quality and high external quality are not always incompatible. A deliverable with poor design requires extra effort and time for changes and maintenance. This also means adding the cost of delay. For instance, code complexity can make it harder to add new features to a software product and more difficult to maintain it.

The main causes of poor quality are:

- Poor quality processes, including poor defect prevention processes
- Unclear quality criteria
- Quality is not built-in
- Quality is not the team responsibility

- Not fixing defects immediately by eliminating the cause
- Working in large batches and excessive work-in-process which hides quality issues
- Delayed feedback on quality
- Poor design of deliverables
- Insufficient resources
- Lack of knowledge and skills
- Lack of motivation
- Ineffective communication
- Time and cost pressure
- Lack of customer involvement (building for the customer instead of building with the customer)

Duplicate efforts

This form of waste may include efforts and time to:

- create something that has already been created
- reinvent and learn again
- do something that is already being done by someone else

Examples of duplicate efforts are:

- Creating items (such as software components, designs and documentation) that exist already and can be reused
- Creating project documents from scratch instead of using templates or documents from previous projects (the same applies to project management processes)
- Finding solutions repeatedly to the same challenges as lessons learned are not used
- Two team members working on the same task or the same problem

The main causes of duplicate efforts are:

- Lack of knowledge capture and continuous improvement processes
- Lack of expertise
- Ineffective communication
- Lack of transparency and collaboration

Lost productivity

Productivity is the efficiency of production. It can be measured as a ratio of needful project outputs to project inputs.

All resources that do not contribute to creating needful project deliverables (or can be saved) are wasted. In this sense, most forms of tactical project waste are waste from lost productivity.

With reference to team productivity, major sources of loss of productivity are:

- Ineffective communication and collaboration
- Excessive work-in-process
- Multitasking and task switching
- Idle time
- Overqualified team members
- Overwork, overtime, frustration and stress
- Queues
- Manual work (e.g., testing) when automation is possible
- Unused employee creativity and untapped human potential

Over-processing

In project management, we can define over-processing as excessive use of resources to create project outputs. When this results in delays, the cost of delay increases the waste.

Examples of over-processing are:

- Excessive planning (especially upfront) and requirement-gathering
- Providing higher quality than the customer expects
- Creating more iterations of a deliverable than needed
- Working on product features beyond the minimum feature set
- Creating excessive documentation
- Excessive information and communication
- Excessive reviews and reporting
- Overanalyzing
- Micromanaging

Overcomplicating

Complex projects, deliverables and project management processes require excessive efforts and time and produce waste. Overcomplicating is about going beyond “good enough”, “minimum viable” and “absolutely necessary”.

Complexity in project management leads to:

- Lack of focus
- Reduced reliability and flexibility
- Increased risk
- Lack of clarity of requirements, process of work and communication
- Lack of understanding

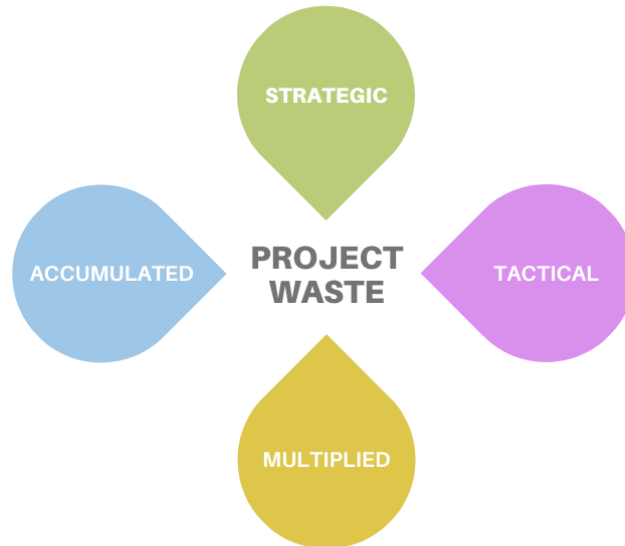
- Difficulty in management and changes

Examples of overcomplicating are:

- Overly complex projects that we could limit to Minimum Viable Projects
- Project products with too complex design
- Complicated communication and collaboration process
- Complicated planning, review, approval and decision-making process
- Complicated project organization

TOTAL PROJECT WASTE

Total Project Waste (TPW) comprises all types and forms of waste on a project, including where applicable Strategic Waste, Tactical Waste, Accumulated Waste and Multiplied Waste.



LeanPM® Framework

© Lean Project Management Foundation

Strategic Project Waste is the waste resulting from separation between the project and the value stream system that needs to be improved, or from lack of alignment with the organizational strategy.

The strategic waste is 100% of the project investment and the subsequent operating and decommissioning costs, assuming efficient and effective project execution (absence of tactical waste). For an aligned project, there is no strategic waste.

Tactical Project Waste is the waste from inefficient and ineffective use of project resources, regardless of whether the project is aligned with the organizational strategy and the value stream system. When there is a strategic waste, the tactical waste is added to it.

Accumulated Project Waste comprises the additional operating and decommissioning costs resulting from wrong project deliverables, low quality, complications or unnecessary features.

Multiplied Project Waste is the waste resulting from extending or replicating a wasteful project or wasteful project management practices.

TAKEAWAYS: PROJECT WASTE

- Lean project management should aim to reduce and eliminate waste.
- Waste are those activities and steps that generate negative net value for the project.
- All levels of the project hierarchy and any workflow step can produce waste.
- When a project does not improve the value-creating system or isn't aligned with the organizational strategy, it's strategically wasteful (Strategic Waste).
- The additional operating and decommissioning costs resulting from wrong, complicated or low-quality project assets or unnecessary features, accumulate project waste (Accumulated Waste).
- Extending or replicating a wasteful project or wasteful project management practices multiplies project waste (Multiplied Waste).
- The seven forms of Tactical Project Waste are wrong deliverables or features, delays, poor quality, duplicate efforts, lost productivity, over-processing and overcomplicating.

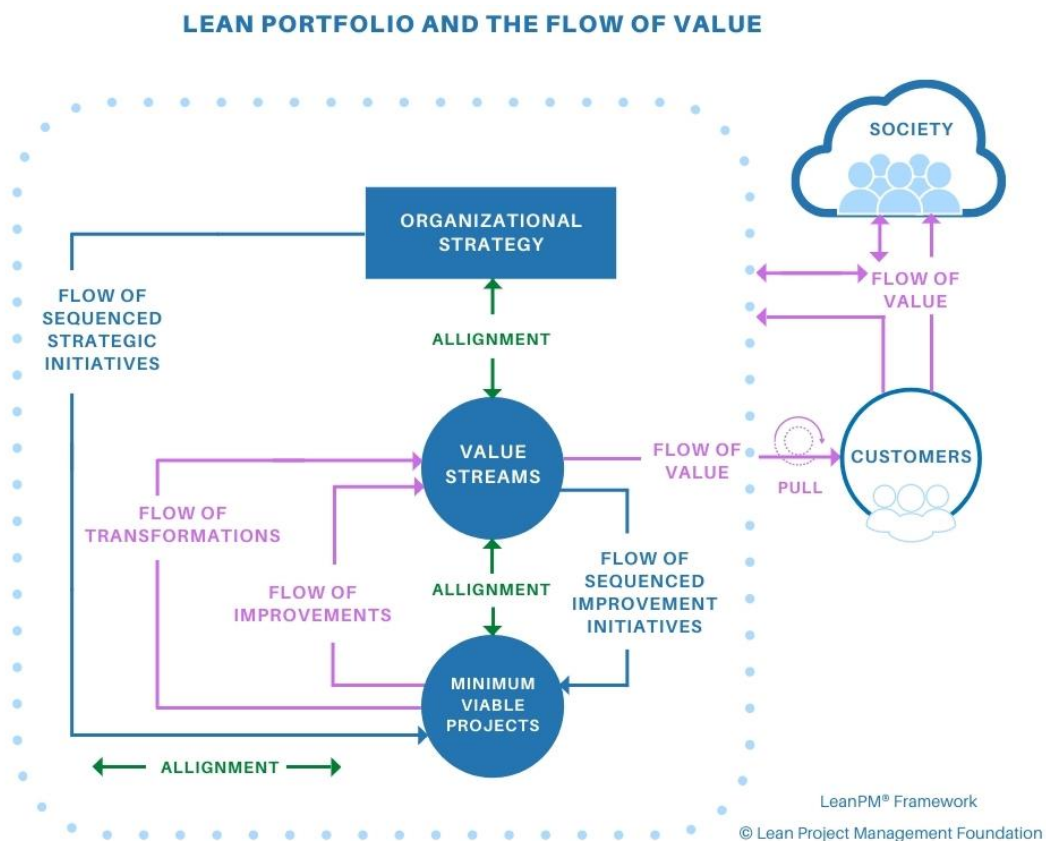
Chapter 5: Lean Portfolio Management

INTRODUCTION TO LEAN PORTFOLIO MANAGEMENT: THE ROLE OF PROJECTS

The projects realize three types of initiatives that support the flow of value created by the organization:

- Strategic transformation initiatives
- Improvement initiatives
- Exploratory initiatives

Figure 5.1 shows the relationship between the lean project initiatives and the flow of value.



Transformation initiatives create new value streams or transform the whole value creation system. They originate directly from the organization's strategy.

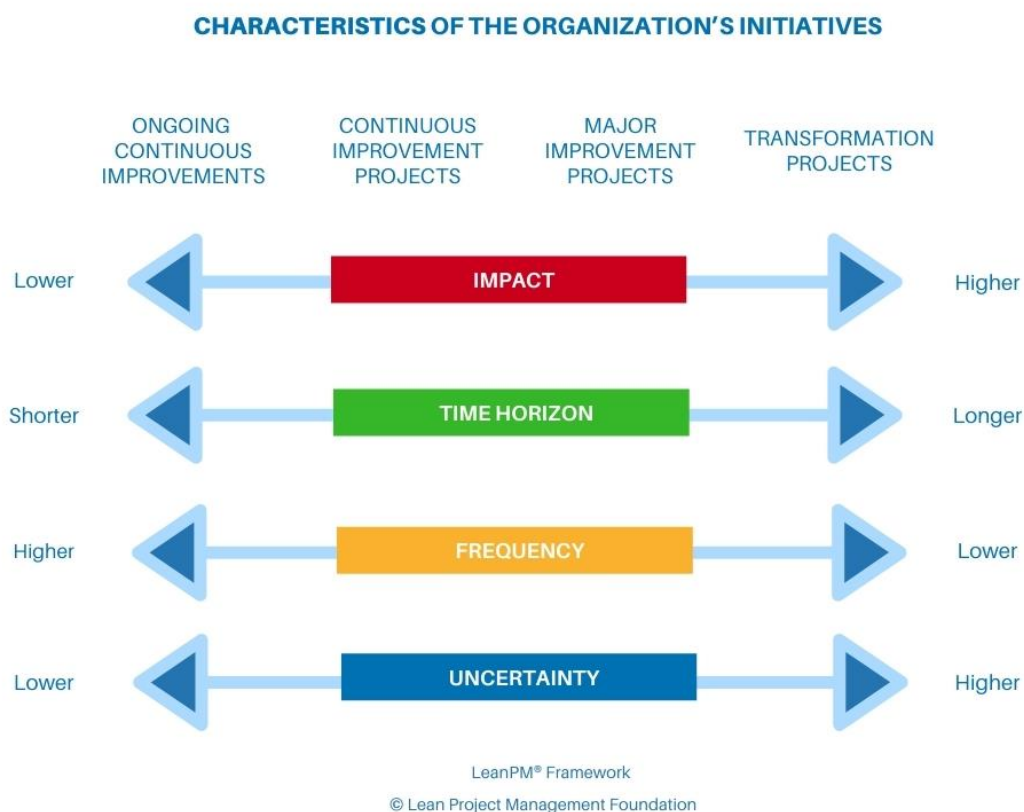
Improvement initiatives stem from the value stream system. These are major and continuous improvements (kaikaku and kaizen) that may include projects for:

- developing new products and services within existing product/service families
- improving existing products, services, value streams and processes
- eliminating obsolete value streams
- ensuring compliance and responding to emergency situations

Exploratory initiatives acquire validated knowledge through rapid plan-do-check-act (PDCA) cycles and serve transformations and improvements. They are stand-alone projects or an Exploration phase of the Lean Project Life Cycle (as we discuss in [The Lean Project Life Cycle](#) chapter).

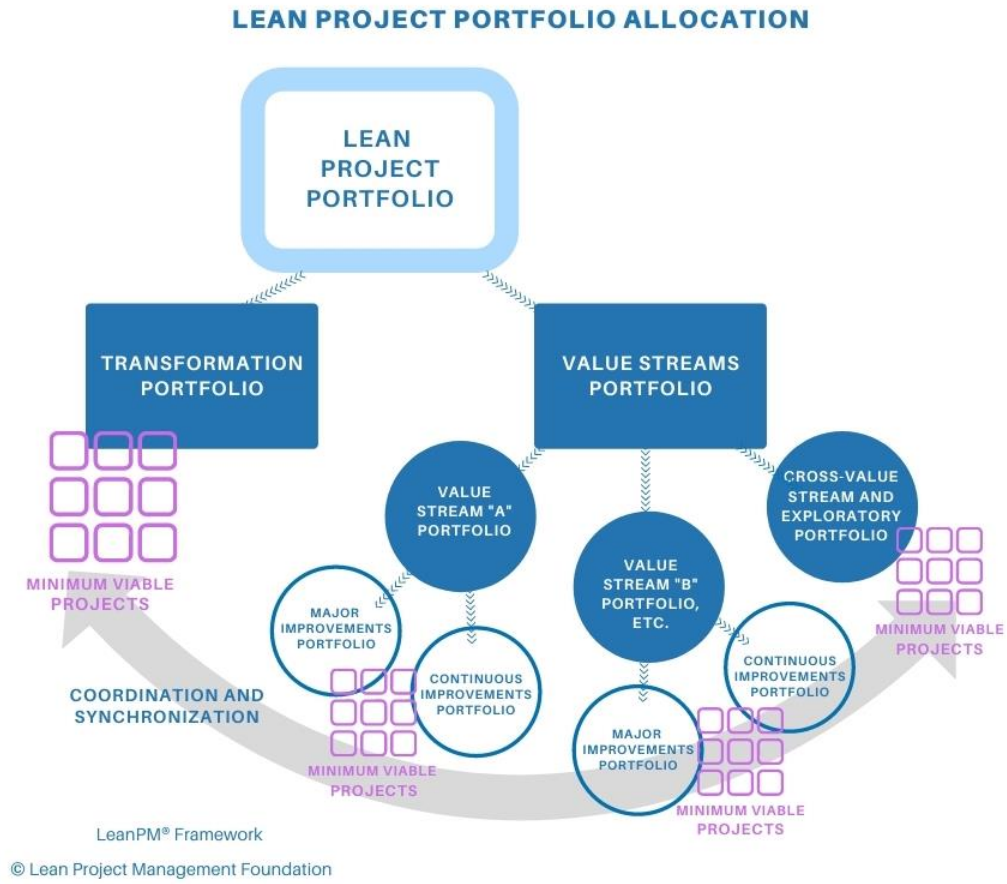
The organization's project initiatives differ from each other, and from ongoing initiatives in terms of impact scale, time horizon, frequency, and uncertainty. We show these characteristics in Figure 5.2.

Figure 5.2: Characteristics of Organization's Initiatives



All these types of initiatives are important and they work as a system. For sustainable development, the organization must maintain an optimal dynamic balance between them ([Apply Systems Thinking](#) principle).

Figure 5.3: Lean Project Portfolio Allocation



LEAN PROJECT PORTFOLIO MANAGEMENT PROCESSES

Introduction

The goal of Lean Project Portfolio Management is to maximize the net value of the organization's initiatives, thus helping to optimize the flow of net value that the organization creates. Portfolio initiatives realize their effect through operational value streams, and each initiative should improve the capacity of the value stream system to create value.

Lean Project Portfolio Management is a means for sustainable development and continuous improvement of the organizational value-creating system. It is fundamentally different from traditional project portfolio management, as shown in the table below.

Traditional Portfolio Management	VS	Lean Portfolio Management
• Centralized management • Managed at an organizational and functional level • Objective cascading • Long term planning and funding • Long resource allocation cycles • Project-based funding • Large and complex initiatives; too many projects • Independent project objectives • Progress measurement against plan • Ad-hoc teams • Separate project teams		• Decentralized management • Managed at an organizational and value stream level • Bidirectional objective setting • Incremental/Adaptive planning and funding • Flexible and fast resource allocation • Funding value streams that incrementally fund projects and ongoing improvements • Minimum Viable Projects and ongoing improvements • Using strategic and value stream KPIs/OKRs as project objectives • Value-based progress measurement • Stable long-lived teams • Subsets of the value stream teams

Lean Project Portfolio Management has three processes that interact with each other and run simultaneously:

- Develop organizational strategy and set objectives
- Generate ideas, select and prioritize projects
- Fund, execute and evaluate

Develop Organizational Strategy and Set Objectives

The vision of the organization gives meaning to its existence, and the strategy is a high-level plan for realizing the vision. The strategy and its objectives provide a framework for defining value-stream objectives. In the lean organization, the strategy, the objectives and the performance measures result from a bidirectional process which is both top-down and bottom-up.

The objectives and Key Performance Indicators (KPIs) or Objectives and Key Results (OKRs) at both the strategic and value stream levels set the foundation for lean portfolio management.

Generate Ideas, Select and Prioritize Projects

Lean organizations constantly screen the environment and sense and seize opportunities. They encourage innovation and intrapreneurship. Kaizen and kaikaku activities create a continuous flow of improvement initiatives. Lean leaders initiate ambitious strategic transformations. Teams use stakeholder feedback to improve customer service and identify fresh opportunities. (See [*Idea Generation and Triage*](#))

To be suitable for selection, potential projects must be:

- Aligned
- Feasible
- Viable

Alignment

Projects are aligned when they contribute to the achievement of the strategic objectives (for the transformation projects) or the objectives of the value streams (for all other projects) as measured by the relevant KPIs or OKRs.

We need not invent project objectives different from the strategic and value-stream objectives. An organizational objective may require several initiatives, and the individual projects will provide incremental contributions to achieving that objective. The opportunities for creativity and innovation remain endless, as there are many ways we can achieve each objective.

Alignment requires a binary decision. It's about yes or no, there are no intermediate options.

The rule is: Use the organization's objectives for project objectives.

Feasibility

Feasible projects are those that are practically possible (at reasonable cost), in terms of the project environment, the nature of the project deliverables and the process of their creation, and in terms of the organization's capacity.

Viability

At the selection stage, viable projects are those whose potential benefits outweigh the potential costs.

The **LeanPM** Framework sets an additional requirement, which is to select Minimum Viable Projects.

Minimum Viable Project (MVP)

A **Minimum Viable Project** is a viable project that is minimized in terms of complexity, scope, effort, cycle time, dependencies, and risk (*Simplify* principle). The focus is not on minimizing but on viability and its maximization.

The questions we need to answer include:

- Is the estimated net value of the initiative sufficient to justify its inclusion in the project portfolio (above the threshold; a minimum desirable net value)?
- Can we implement the initiative more effectively and efficiently as an ongoing continuous improvement or exploration?
- Can we achieve the objectives through a better and simpler project approach?
- Can we achieve the project objectives through better and narrower deliverable configuration, better and simpler deliverables, and better and simpler deliverable creation processes?
- Can we minimize external project dependencies?
- Can we break down the initiative into discrete projects which would reduce overall complexity, uncertainty and cost, and would speed up value creation?
- Is the project independent (not a part of mutually exclusive projects)?
- Does the project overlap with other initiatives?

One aim of the Minimum Viable Project is to draw the line between projects and ongoing improvement initiatives. It should support a growing number of initiatives that become a flow of ongoing improvements, at the expense of initiatives formalized as projects.

The Minimum Viable Project reduces waste and risk, shortens cycle time, increases the probability of success, and limits the cost of failure. By minimizing individual viable projects, we maximize the net value of the project portfolio.

The concept of the Minimum Viable Project resembles that of the Minimum Viable Product, but there are significant differences between them. The Minimum Viable Project seeks to improve the project's manageability. The Minimum Viable Product helps you quickly validate or invalidate hypotheses and fail fast.

LeanPM Framework uses the Minimum Viable Product and other similar concepts in the exploratory projects. In a Minimum Viable Exploratory Project, we create validated knowledge through rapid plan-do-check-act cycles, using increments that include:

- Minimum Viable Product
- Minimum Viable Service
- Minimum Viable Value Stream
- Minimum Viable Experiment
- Minimum Viable Research
- Minimum Viable Study
- Minimum Viable Test

Project Selection

The initial screening will create a pool of potentially aligned, feasible and viable projects, and then selections can be made for inclusion in the portfolio (see [A3 Analysis and Pre-Selection/Selection](#)).

The selection should be grounded on a project's absolute and relative importance, or simply put, on the importance and priority.

Prioritization

Prioritization addresses the following issues:

- Allocation of resources to the transformation portfolio and the value streams portfolio
- Allocation by strategic themes within the transformation portfolio
- Allocation of resources to individual value stream and the cross-value stream portfolios as well as to major and continuous improvement projects, exploratory projects and ongoing continuous improvements
- Project sequencing (schedule priority)

Allocation is a bi-directional process with complete decentralization at the value stream level. The governance teams discuss and decide on the relative importance of the different portfolios and sub-portfolios and align the allocation with the capacity.

Prioritization seeks to achieve an optimum balanced mix of sequenced initiatives that will best deliver the strategy. The organization must continuously challenge and revisit priorities and allocation.

Importance

If we determine the importance of each project, we can sort the projects from highest to lowest in importance and select projects until we have exhausted the resources of the respective portfolio.

It's a common practice to use scoring models to evaluate project importance. The scoring model is composed of multiple criteria with assigned weights and a scoring scale. The governance team assigns numerical scores to each criterion, and the total weighted score represents project value and importance.

For example, we could use a scoring model for triage and sorting of project ideas into one of the four categories of MoSCoW prioritization: Must have, Should have, Could have and Won't have.

However, the total project score is not a good measure of project value. When we use it, we encounter the following problems:

- The total project score does not carry objective information about the project's value. The score does not represent absolute value, but (at best) relative value of one project compared to another.
- We measure the project score and project cost in different units, which makes it impossible to calculate the net value (the net value rather than the value shows project's importance).
- We can manipulate the result through the scores and weights.
- It seems logical to use total project score to sequence projects (and we often do this), but, as we'll see, this is incorrect.

- When we use the expected net monetary value together with other criteria in the scoring model, this means that we haven't properly measured the expected benefits and costs. Adding a risk criterion, for example, shows that when estimating the net value, we didn't take into account the parametric and model uncertainty (and if we did, we should not use a duplicate criterion).
- The arbitrary, indiscriminate use of diverging criteria and subjective scoring and weighting may result in a biased outcome.

Among the most commonly used criteria in scoring models are strategic alignment, return, risk and compliance. Organizations also use a variety of other criteria that ultimately affect benefits, costs, or risks.

But there are many questions that need to be asked:

- What's the point of assigning 0 score for "no alignment with strategic goals" and a score of 4 for "alignment with more than 3 strategic goals"?
- Can we accept "no alignment"?
- Can we compensate for non-alignment with a higher score for project size and technical feasibility, for example?
- Is the project impact proportional to the number of strategic objectives it's aligned to?
- If a project contributes to the achievement of only one strategic objective, does this make it inconsistent with the other strategic objectives (and would a good strategic plan allow such inconsistency)?

Instead of a rational, we get a lot of questions. As noted above, alignment is a precondition for considering the project and should not be part of the appraisal model.

As we suggest, using the organization's objectives for project objectives has another important advantage. If we have previously evaluated how the progress in achieving objectives benefits the organization, this will help us assess the effect of the incremental contribution of the projects.

Let's look at three examples of value stream objectives:

- To improve Lead Conversion Ratio (the ratio of the number of leads to the number of those who turn into customers).
- To lower the Absenteeism Rate (the ratio of the number of absences to the number of workdays in a given period, expressed as a percentage).
- To improve Donor Retention Rate for a nonprofit organization (percentage of donors who have donated more than once).

Operational value streams should have evaluated (in monetary units) the effect of incremental change in the values of the indicators associated with these objectives. This would enable us to assess the impact of projects that, for example, aim to reduce the Absenteeism Rate from 4.2% to 3.3%, improve Lead Conversion Ratio from 20:1 to 15:1, and improve Donor Retention Rate from 38% to 43%.

Using compliance and urgency (emergency) in scoring models is also problematic. If we need a project to save the organization and continue operations, this project is mandatory and of utmost importance. We should include it in the portfolio. Where the threat is not apparent and imminent, the benefit-cost

assessment will establish the project's importance, as for any project. So, we have a better way to take compliance and urgency into account.

Risk-Adjusted Expected Net Value

Risk has no independent role in the importance of the project. Its role lies in the fact that it affects the benefits and costs. The same applies to many other criteria used in the scoring models (including time-related factors). We should therefore strive to take into account the impact of risk and other factors on benefits and costs, to work out the **risk-adjusted expected net value** of the project as the only measure of its importance.

Once we have selected projects for each portfolio using risk-adjusted expected net value, it's very tempting to apply this metric of importance in order to establish schedule priorities and to sequence the projects. But that would be a mistake.

Projects Sequencing (Schedule Prioritization)

As our goal is to maximize the total net value of the portfolio for any time period, we need to consider its total Cost of Time. If we execute one project before another that has a higher Cost of Time, we may incur costs that we can avoid by changing the order of execution of the two projects.

Don Reinertsen gives three principles for prioritizing (sequencing) jobs: Shortest Job First (SJF), High Delay Cost First (HDCF) and Weighted Shortest Job First (WSJF) [1].

Shortest Job First (SJF) is the best sequencing strategy when all jobs have the same cost of delay. Starting with the longest jobs would delay the others more than when starting with the shortest jobs and would lead to higher overall delay cost.

High Delay Cost First (HDCF) should be applied when the costs of delay differ, but all jobs have the same duration. The jobs with higher cost of delay have a priority.

Weighted Shortest Job First (WSJF) is the best scheduling strategy when both durations and delay costs are different. WSJF is equal to delay cost divided by job duration. WSJF is also referred to as CD3 (Cost of Delay Divided by Duration). The logic is that the higher the delay cost and the shorter the duration of the job, the smaller its contribution to the total costs when we do it earlier.

Here is an example of applying the WSJF rule.

Project	Cost of Delay (\$/week)	Duration (weeks)	WSJF (CoD/Duration)
A	\$1K	4	0.25
B	\$2.5K	8	0.31
C	\$3.2K	12	0.27

From the information in the table, we can calculate which of the sequencing combinations produces the optimum results (here we present the calculations for four of the possible six combinations):

B-C-A sequence

- Project B is delayed by 8 weeks and the CoD is \$20K
- Project C is delayed by 20 weeks and the CoD is \$64K
- Project A is delayed by 24 weeks and the CoD is \$24K

The total CoD is \$108K

A-B-C sequence

- Project A is delayed by 4 weeks and the CoD is \$4K
- Project B is delayed by 12 weeks and the CoD is \$30K
- Project C is delayed by 24 weeks and the CoD is \$76.8K

The total CoD is \$110.8K

C-B-A sequence

- Project C is delayed by 12 weeks and the CoD is \$38.4K
- Project B is delayed by 20 weeks and the CoD is \$50K
- Project A is delayed by 24 weeks and the CoD is \$24K

The total CoD is \$112.4K

A-C-B sequence

- Project A is delayed by 4 weeks and the CoD is \$4K
- Project C is delayed by 16 weeks and the CoD is \$51.2K
- Project B is delayed by 24 weeks and the CoD is \$60K

The total CoD is \$115.2K

So, according to the WSJF rule, the best sequence is **B-C-A** because it has the lowest total CoD of **\$108K**. The sequence of **A-C-B** has the highest CoD of **\$115.2K**.

Let's add two more projects. The best sequence now is D-E-B-C-A and the sequence with the highest CoD is A-C-B-E-D.

Project	Cost of Delay (\$/week)	Duration (weeks)	WSJF (CoD/Duration)
A	\$1K	4	0.25
B	\$2.5K	8	0.31
C	\$3.2K	12	0.27
D	\$5.2K	10	0.52
E	\$2K	6	0.33

Sequencing projects does not imply that we need to wait for one project to finish before starting another. Depending on our resource capacity and the target average lead time, we will have a certain number of projects in process.

WSJF principle is a major improvement in decision-making. We can see that to apply it, two conditions must be met:

1. The delay costs of the individual jobs must be constant for all time periods.
2. All jobs must have the same reference point of delay (the present moment).

In fact, the WSJF, HDCF and SJF principles take into account the effect of postponing the jobs. We can formalize this effect as a **Cost of Postponed Execution (CPE)**, which includes components such as lost opportunity cost, deferred use cost and customer dissatisfaction cost. As we discuss in the [Cost of Time](#) chapter, projects can have distinct points of reference for postponement and the postponement costs may vary over the time periods.

WSJF uses the expected job duration, which it takes as a constant. The delay in this case refers to the delay in starting and ending the work, and not to the extension of its cycle time (which we defined as the delay cost factor).

It's a common situation to have unique values of the cost of postponed execution of a project for each future time period. Along with the varying reference points of postponement, this makes the sequencing a more complex task.

Let's look at an example of three projects that we need to arrange by order of execution. We need to find the best sequence among the six possible combinations.

Case Study

General Information

Project	Expected duration (number of time periods)	Ideal point in time to finish Just-in-Time	Estimated ideal point in time to start
Project Alfa	2	End of period 3	Beginning of period 2
Project Beta	3	End of period 6	Beginning of period 4
Project Gama	5	Now	Beginning of period 1

Cost of Time and WSJF

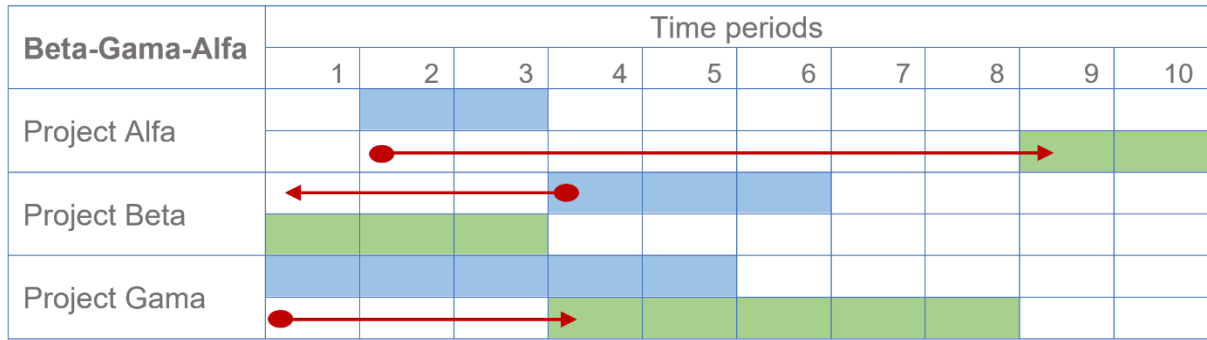
	Cost of Premature Delivery (CPD)			Cost of Postponed Execution (CPE)							WSJF		
	-3	-2	-1	1	2	3	4	5	6	7-10	Average CoD/CPE	WSJF	WSJF order
Periods of PD/PE													
Project Alfa			2	2	3	3.5	4.2	5.1	6.2	6.5	5.0	2.5	1
Project Beta	1.8	1.5	1.5	1.9	2	2.2	2.2	4.7	7.5	10.6	6.3	2.1	2
Project Gama				3.8	3.8	3.8	4	4	4.2	4.2	4.0	0.8	3

Since we can't directly apply the WSJF formula, we could calculate the average CoD/CPE values for the period 1-10, which gives us Alfa-Beta-Gama as the optimal sequencing combination.

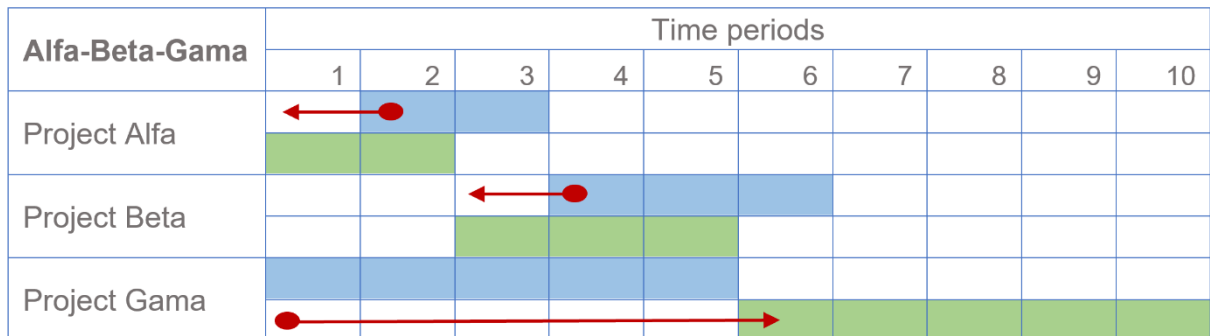
It's obvious that in this case we cannot apply the WSJF and SJF rules correctly. When we perform the individual calculations for each combination (which we show below), we can see that the optimal sequence is **Alfa-Gama-Beta**.

Beta-Gama-Alfa	Postponement (number of periods)	CPE generating periods	CPE
Project Beta	-3	-3, -2 and -1	4.8
Project Gama	8	1-8	32.0
Project Alfa	7	1-7	30.5
Total Cost of Postponed Execution			67.3

Gama-Alfa-Beta	Postponement (number of periods)	CPE generating periods	CPE
Project Gama	5	1-5	19.4
Project Alfa	4	1-4	12.7
Project Beta	4	1-4	8.3
Total Cost of Postponed Execution			40.4



Gama-Beta-Alfa	Postponement (number of periods)	CPE generating periods	CPE	Alfa-Beta-Gama	Postponement (number of periods)	CPE generating periods	CPE
Project Gama	5	1-5	19.4	Project Alfa	-1	-1	2.0
Project Beta	2	1-2	3.9	Project Beta	-1	-1	1.5
Project Alfa	7	1-7	30.5	Project Gama	10	1-10	40.4
Total Cost of Postponed Execution			53.8	Total Cost of Postponed Execution			43.9



Alfa-Gama-Beta	Postponement (number of periods)	CPE generating periods	CPE	Beta-Alfa-Gama	Postponement (number of periods)	CPE generating periods	CPE
Project Alfa	-1	-1	2.0	Project Beta	-3	-3, -2 and -1	4.8
Project Gama	7	1-7	27.8	Project Alfa	2	1-2	5.0
Project Beta	4	1-4	8.3	Project Gama	10	1-10	40.4
Total Cost of Postponed Execution			38.1	Total Cost of Postponed Execution			50.2

Alfa-Gama-Beta	Time periods									
	1	2	3	4	5	6	7	8	9	10
Project Alfa	←●	■	■							
	■	■								
Project Beta				■	■	■				
				●→				■	■	■
Project Gama	■	■	■	■	■					
	●→		■	■	■	■	■			

We can conclude that when the postponed execution costs for the individual time periods, the durations, and/or the reference points for postponement are not all homogeneous, we can't apply the SJF, HDCF and WSJF rules. We need to identify the correct priority by **comparing the total costs of postponed execution of the individual combinations**.

Note also that the CPE priorities will depend on the time horizon and the trends in the time-related costs. Varying the time horizon may lead to different optimal prioritization.

Fund, Execute and Evaluate

The portfolio allocation should not be fixed for the entire year. It needs to be regularly assessed and altered as needed, for example, following quarterly or monthly feedback loops. Within the funds allocated to each portfolio, the individual projects should be funded incrementally and each portion of the investment should be justified by the net profit it would generate.

Incremental financing reduces risk and waste. Short feedback loops on project progress help to improve budget estimates and to better assess the probability of achieving project objectives through rolling forecasts. The more accurate net project value estimates, and the changing environment and organizational objectives, may require a rearrangement of portfolio priorities. Therefore, project funding should be incremental, taking into account the shifting budgetary needs and priorities.

The key aspects of lean portfolio execution are ongoing project coordination and synchronization and portfolio optimization, by altering resource allocation and changing, cancelling and modifying projects.

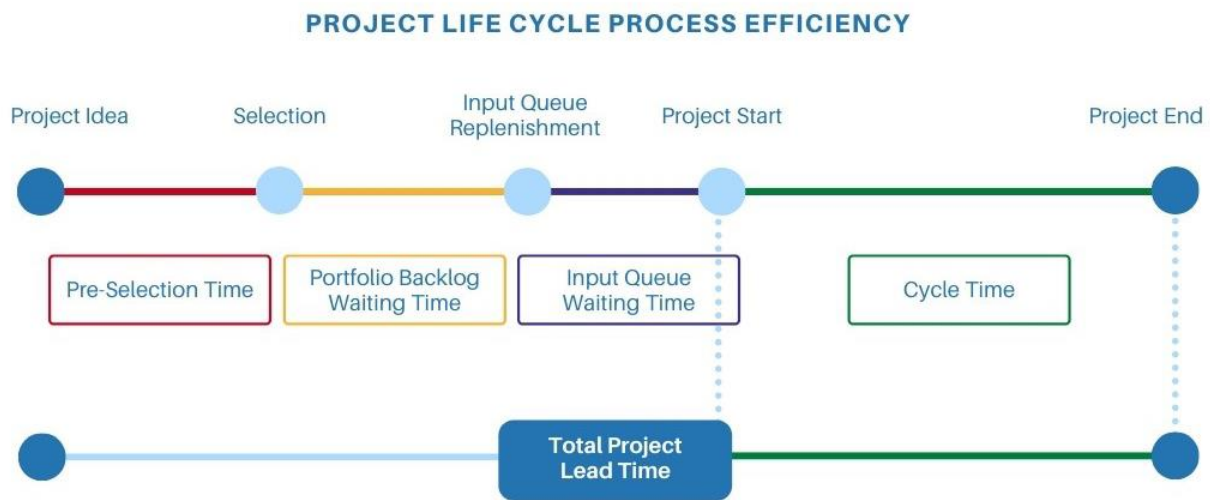
Regular evaluation of the portfolio provides the basis for portfolio optimization and continuous improvement of portfolio management processes.

PORTFOLIO PROCESS EFFICIENCY AND PROJECT LEAD TIME

An important thing to look at, when considering the portfolio management processes, is the process efficiency of the portfolio (or the process efficiency of the project life cycle). The cycle efficiency of a process is calculated by dividing the value-added time by the total lead time of the process. A low process cycle efficiency is a symptom of waste.

As we can see in Figure 5.4, the total project lead time comprises several components.

Figure 5.4: Project Life Cycle Process Efficiency



$$\text{Project Life Cycle Process Efficiency (\%)} = \text{Project Cycle Time} / \text{Total Project Lead Time}$$

LeanPM® Framework

© Lean Project Management Foundation

At the level of the project portfolio, we can calculate the efficiency of the project life cycle process by dividing the Project Cycle Time (the time for executing the project) by the Total Project Lead Time (in-system time).

Often, the time for non-executing life cycle phases is a large part of the total lead time. The low cycle efficiency is a symptom of waste. To reduce the lead time, we need to shorten the non-executing phases.

The project execution phase (and the other phases in the project life cycle) also has its own measure of cycle efficiency - the ratio of the value-added time to the cycle time. We address this issue in the *Managing Creation and Absorption* chapter.

The average lead time of the individual projects is affected by the work in progress limit (or the lack of limit) for each portfolio.

Sequencing projects does not imply that we need to wait for one project to finish before starting another. Depending on our capacity and the target average lead time, we will have a certain number of projects in progress at the same time.

Little's Law is a theorem that provides a principle for evaluating the efficiency of queuing systems. It defines the relationship between the long-term average number of customers in a stationary system (L), the long-term average effective arrival rate of the customers (λ) and the average time that a customer spends in the system (W).

In a stable system, the departure rate (throughput) is the same as the arrival rate.

The formula of the law is: **$L = \lambda W$**

In projects, the relationship is between average number of jobs in the system (Work in System or in-system inventory), average processing rate (throughput) and average lead time.

Work in System (WIS) is the number of jobs in the system (waiting to be serviced or being worked on).

Processing Rate is the number of jobs that are processed and depart the system for a given period.

Lead Time (time in system) is the time a job spends in the system, including queue time (waiting to be processed) and the processing time (cycle time).

The project lead time has important implications for project success due to the Cost of Time. With the help of the law, we can determine the average number of projects in the system that is needed to achieve a certain average lead time. We can define the project processing system as the time and processes framework, from project idea generation to project completion.

Let's say that the average processing rate of a value stream is 30 process improvement projects per year and we aim to ensure that the average project lead time is no more than 5 months.

Using the formula

Average Work in System (L) = Average Processing Rate (λ) x Average Lead Time (W),

we get **12.5** ($30 \times 5/12$).

To ensure an average project lead time of up to 5 months, we must limit the average number of in-system projects to 12 and balance the arrival rate with the departure rate.

For an average lead time of 3 months, we have to set a limit to an average of 7 in-system projects. We achieve the longest average lead time of 12 months with an average of 30 projects in the system.

TAKEAWAYS: LEAN PORTFOLIO MANAGEMENT

Project initiatives create a flow of transformations and improvements and support the organization's flow of value.

The continuous improvement, major improvement and transformation projects, and the ongoing continuous improvements differ in the scale of impact, time horizon, frequency (number) and uncertainty. All these types of initiatives work as a system, and the organization needs an optimal dynamic balance between them to ensure its sustainable development.

Lean Project Portfolio Management maximizes the net value of the organization's initiatives, thus improving the capacity of the value-creating system and contributing to the maximization of the organization's flow of net value.

Lean Project Portfolio Management differs from the traditional portfolio management:

- It's managed in a decentralized way at both organizational and value stream levels.
- It uses bidirectional objective setting, and incremental and adaptive planning and funding.
- Resource allocation is fast and flexible and funds value streams that incrementally fund Minimum Viable Projects and ongoing improvements.
- Lean portfolio management uses strategic and value stream objectives and KPIs or Objectives and Key Results (OKRs) as project objectives.
- Progress measurement is value-based.
- Stable long-lived teams which are subsets of the value stream teams are preferred.

Project selection should be based on alignment, feasibility, viability, importance and priority.

Minimum Viable Projects reduce waste and increase the probability of success.

Using scoring models to evaluate project importance is problematic because this can lead to biased decisions.

An appropriate measure of project importance is the risk-adjusted expected net value.

Prioritization provides allocation of resources to individual portfolios, including allocation by strategic themes, and project sequencing.

Allocation is a bidirectional process. It should be completely decentralized at the value stream level and aligned with the capacity.

Schedule prioritization must be based on the Cost of Time.

When the postponed execution costs for the individual time periods and the durations and/or the reference points for postponement are homogeneous, we should apply the SJF, HDCF and WSJF rules to sequence projects. When these parameters are not homogeneous, to identify the correct priority, we have to compare the total costs of postponed execution of the individual sequencing combinations.

Schedule priorities depend on the time horizon and the trends in the time-related costs.

The portfolio should be reassessed and altered following regular feedback loops.

To reduce risk and waste, short feedback loops on project progress must facilitate incremental funding.

Key aspects of portfolio management are ongoing project coordination and synchronization and continuous improvement of portfolio management processes.

The portfolio process efficiency should be improved by reducing the time for non-executing project life cycle phases. The average project lead time can be shortened by limiting the Work in Progress.

REFERENCES - CHAPTER 5

[1] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing

Chapter 6: The Lean Project Life Cycle



INTRODUCTION

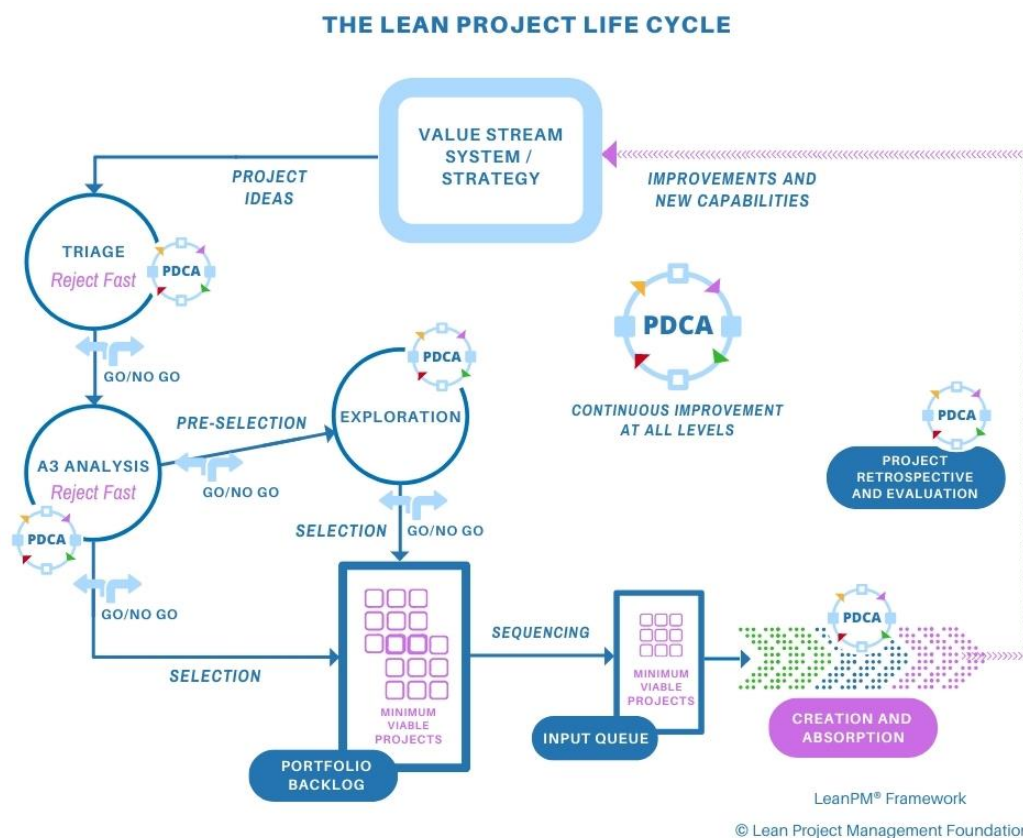
The project life cycle is the succession of phases that a project goes through from the emergence of the idea to the realization of the benefits. We look at this topic briefly, as we provide more details in the *Lean Portfolio Management*, *Lean Development Life Cycle*, and *Managing Creation and Absorption* chapters.

A common mistake is to consider Initiating, Planning, Executing / Monitoring and Controlling and Closing as phases of the project cycle. In addition, they're often referred to as "the five phases of project management", which makes the confusion even greater. In fact, these are processes (process groups) that form a variant of the Plan-Do-Check-Act (PDCA) cycle, which applies to each phase of the project life cycle.

The **LeanPM** Framework project life cycle comprises the following phases:

- Idea Generation and Triage
- A3 Analysis and Pre-Selection/Selection
- Exploration
- Sequencing
- Creation and Absorption
- Project Retrospective and Evaluation

Figure 6.1: The Lean Project Life Cycle



IDEA GENERATION AND TRIAGE

Each employee, customer or partner is empowered to generate a project idea. In addition, the organization purposefully generates project ideas by:

- Making plans to achieve the organization's objectives
- Monitoring the organization's value-creation system to identify problems and opportunities for improvement
- Monitoring the organizational environment, including the market, competition, social changes, innovation development and legal framework
- Using the results of its research and development work
- Conducting Kaizen and innovation sessions and other improvement exploration activities

It's important to stimulate the generation of many project ideas and they should be validated quickly. We achieve this by a triage - a rapid assessment of project ideas suitability. Its purpose is to prevent further spending on inappropriate ideas.

The use of a structured Project Idea Form can support triaging. The form might require the submission of information on the idea, benefit, cost, alignment, and lack of redundancy, as well as the compliance criteria for certain minimum requirements (e.g., for ROI).

The project idea is suitable when it:

- Meets the established criteria for a project. Most ideas will be for small incremental improvements. Potential projects would involve a certain complexity, effort, time and expected impact that would require them to be managed as temporary value streams.
- Is aligned with the organization's objectives.
- Does not duplicate other initiatives. To this end, the organization should be transparent about all initiatives.
- Complies with certain requirements like a minimum ROI (if it's possible to be calculated at this stage) or at least shows that the potential benefits would outweigh the potential costs.

The originators should know the suitability criteria and be encouraged to submit ideas that meet the expectations.

The suitable project ideas move to the A3 Analysis and Selection/Pre-Selection phase.

A3 ANALYSIS AND PRE-SELECTION/SELECTION

A3 Analysis is a process, the outcome of which can be recorded on a one-page document to be used for decision-making. Initially, A3 documents (reports) were prepared on one A3 size sheet of paper, but now it's more usual to condense the information onto one A4 page.

In this phase, the initiators of the project perform additional analysis to create its preliminary definition. This definition is summarized in a one-page A3 Project Definition document.

The A3 Project Definition may contain the following information, using visual elements as much as possible:

- Description of the problem or the opportunity
- Root-cause analysis (if applicable)
- Comparison of alternatives and recommended project approach
- Project objectives, main deliverables and activities
- Expected costs, benefits and ROI
- Risks, assumptions and preconditions
- High-level project plan

Based on the A3 analysis and definition, we can make the following decisions:

- A project that doesn't meet the selection criteria is rejected
- A project with limited complexity, uncertainty and costs may be selected
- A project with greater complexity and uncertainty and higher costs may be pre-selected because it's promising, but needs further analysis

The selection includes allocation prioritization (as discussed in Lean Portfolio Management chapter). We move the selected projects to the portfolio backlog for sequencing and subsequent execution.

The pre-selected projects are further explored in the Exploration phase.

EXPLORATION PHASE

In this phase, we perform additional just enough studies, tests and analyses of the potential project and its alternatives, to decide whether we should select it for execution. It's about executing an initial smaller project, which may include feasibility studies, initial design, testing assumptions and validating or invalidating hypotheses (validation studies), pilot tests, business case development, etc.

As an outcome of the exploration phase, the project is selected for execution or is rejected because it doesn't meet the selection criteria.

SEQUENCING

The selected projects are included in the Portfolio Backlog. They are given a schedule priority and are sequenced to replenish the Project Input Queues. See the Lean Portfolio Management chapter for more information.

Execution begins when project teams pull projects out of the input queues.

CREATION AND ABSORPTION

We call the project execution (development) phase of the **LeanPM** Framework "Creation and Absorption", because it involves two concurrent processes:

- Creation: the process of creating project assets and
- Absorption: the process of absorbing project assets into the value stream system.

There is no separate phase for deployment or transition in the **LeanPM** Framework. Experience has shown that such phases are perilous and cause the failure of many projects.

The value stream continually absorbs the assets created by the project. Absorption is as important a process as creation. See more about this in *Managing Creation and Absorption*.

PROJECT RETROSPECTIVE AND EVALUATION

Project Retrospective and Project Evaluation are tools for learning and continuous improvement (*Build Knowledge and Continuously Improve* principle).

Project Retrospective is a reflection on a completed project which answers the questions about what worked well, what didn't work well, and why and what the teams should test or improve. It's most often held as a structured session that ends with action items to improve future project work.

The retrospective takes place immediately after the project is completed, and its focus is on the team's work and the *Conditions of Satisfaction*.

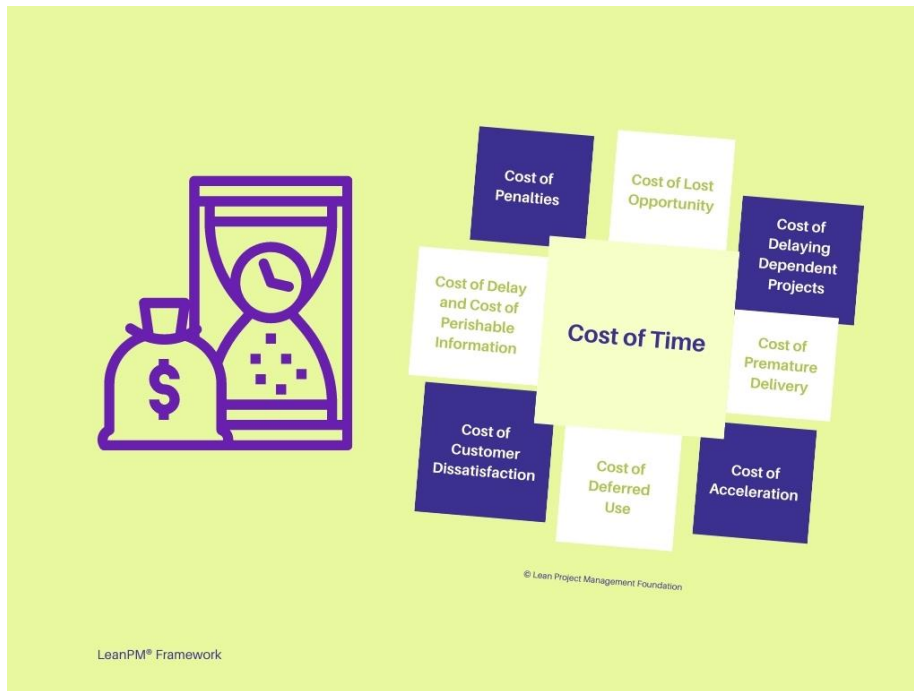
Project Evaluation is an assessment of the level of achievement of project objectives and of the project's effectiveness and efficiency.

The evaluation is performed sometime after the completion of the project, depending on the time horizon of achieving the objectives. Its focus is on analyzing the success or failure of the project and on recommending improvements to the overall project management process.

TAKEAWAYS: THE LEAN PROJECT LIFE CYCLE

- Using Initiating, Planning, Executing/Monitoring and Controlling and Closing as project cycle phases is a common mistake.
- The **LeanPM** Framework project life cycle is comprised of the six phases of Idea Generation and Triage, A3 Analysis and Pre-Selection/Selection, Exploration, Sequencing, Creation and Absorption, Project Retrospective and Evaluation.
- In the Idea Generation and Triage phase, the organization collects and generates project ideas and checks their suitability. The irrelevant ideas are de-selected fast and early to reduce waste.
- Relevant project ideas are subject to rapid analysis and preliminary definition, which is summarized in an A3 Project Definition document. At this stage, de-selection should also be fast to limit waste. Projects with limited complexity, uncertainty and costs that comply with the selection criteria are selected. Promising projects with greater complexity and uncertainty, and higher costs that need further analysis, are pre-selected.
- The pre-selected projects are further explored through rapid PDCA cycles in the Exploration phase.
- Project selection criteria include profitability and allocation priority.
- The selected projects are included in the Portfolio Backlog and are given schedule priority to replenish the Project Input Queues. Project teams pull projects out of the input queues.
- Practice shows that the use of a separate deployment or transition phase is problematic, and **LeanPM** Framework doesn't have such a phase.
- The project execution (development) phase of the **LeanPM** Framework is called "Creation and Absorption". It involves two concurrent processes of equal importance: Creation, i.e., creating project assets and Absorption, i.e., absorbing project assets into the value stream system.
- As tools for learning and continuous improvement, Project Retrospective and Project Evaluation focus on the team's work and the Conditions of Satisfaction, the project success or failure and on enabling improvements.
- The Lean Project Life Cycle involves continuous improvement at all levels.

Chapter 7: Cost of Time



DEFINITION AND COMPONENTS OF COST OF TIME

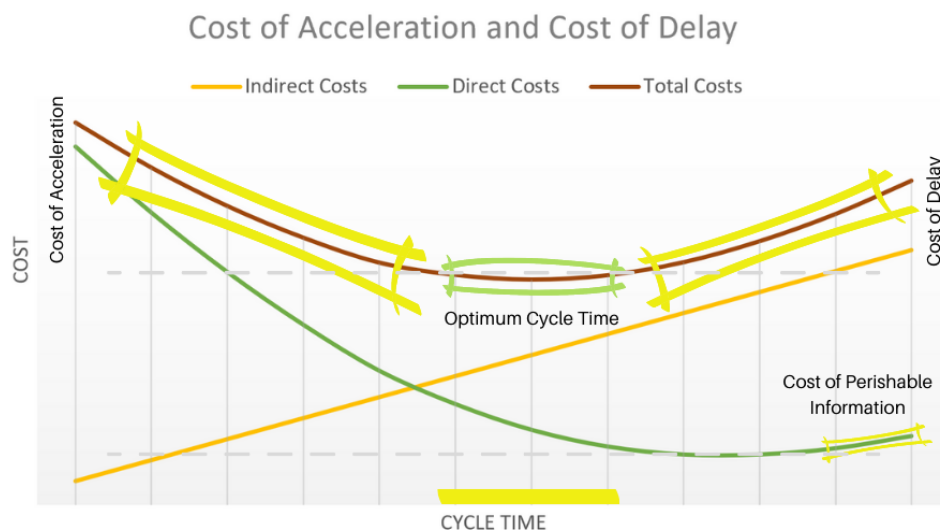
The **Cost of Time** is not an abstract concept. In a project, this is an actual cost that increases the project life cycle costs.

We owe thanks to Donald Reinertsen for the powerful concept of the Cost of Delay in lean product development. We extended this concept to that of the Cost of Time (CoT) in project management by:

1. Differentiating the delay from the postponement (or the project delay from the delayed product use)
2. Establishing reference values for the delay and postponement
3. Defining various time-related cost components.

We express the Cost of Time in monetary units per unit of time, for example, \$/week or \$/month.

The project delay needs a reference value: When is a project late? When we fail to meet the planned deadline or the deadline set by someone? No. In terms of project economics, the delay reference value is the optimal project cycle time - the one which is associated with the lowest total investment costs. Let's look again at the figure we used in *How Do Projects Work? Project Success* chapter.



Shortening the optimal cycle time generates acceleration costs and extending it generates delay costs. A project is delayed when its execution takes more than the optimal cycle time. The theoretical optimal cycle time assumes no resource constraints and no workflow queues, while the adjusted optimal cycle time takes into account the resource constraints and the ability to manage queues. Work in Progress, batch sizes and capacity utilization affect the flow of work and they are delay factors. And in all cases, we are talking about an expected cycle time with attached probability.

As we discussed, the cost of project delay is a combination of incremental direct and indirect costs. It cannot be generated if the project hasn't started. But even if it hasn't started, the project may incur other time-related costs. To understand what these costs are, we need to answer the following question: *When is the best time to start and finish a project?*

Let's use a case study to calculate the answer. We will apply the Just-in-Time concept to project management.

Project Case Study

Let's say that the ideal point in time to complete a project to develop a new summer soft drink is the 15th of April in year N. This will allow us to start selling the new product from the 1st of May. If we complete the project before April 15th, we will incur costs of holding finished project deliverables of \$2,200 per week. Completion of the project after April 15 will cause loss of sales.

When is the best time to start this project? As soon as possible, so that we have enough time to finish it on time? No, the best time depends on the optimal project cycle time. If it's 6 months, the estimated ideal point in time to start the project is October 15 in year N-1.

Let's assume that the cost of acceleration is \$7,000 per week and the cost of delay is \$2,000 per week. In addition, we have to account for the cost of lost opportunity and the cost of deferred use of project deliverables.

The new drink will be on sale from May to September for 4 years (for 85 weeks).

Any delay in this project will lead to loss of sales. The sales won't simply shift over time. Lost sales in May won't be offset by sales in October. If that were the case, we would have had costs related to deferred sales. But in this case, the sales will be completely lost and we will incur 100% lost opportunity cost for each day of unrealized sales, that is, for each day of project delay. Our actual cost will equal the lost profits for the days of missed sales.

If our lost opportunity cost is \$9,000 per week, we can do the math.

Assuming the project starts on October 15 in year N-1 and that the optimum cycle time is 6 months, the best-case scenario is to finish it on April 15th in year N with zero CoT. The CoT for a week of acceleration is \$7,000 and the CoT for a week of delay is \$11,000.

Also, we can see that our cost of acceleration is lower than the sum of the cost of lost opportunity and the cost of deferred use. If the project was already delayed, saving a week of cycle time will result in \$4,000 of net benefit (\$11,000 - \$7,000) though the investment costs will increase. If the optimal start date for the project was missed and the earliest possible start date is the 11th of December in year N-1, we can buy 8 weeks of cycle time for \$56,000, but we need to check if with the increased costs the project still has an acceptable return on investment.

We can use the costs of holding to assess the benefit of starting the project early, to reduce the likelihood of missing out on the ideal completion date. Here, we may decide that it's better to start several weeks earlier than the optimum time and to absorb the cost of \$2,200 per week for premature project delivery (Cost of Premature Delivery).

There are some special cases of the Cost of Time. For example, if we develop a prototype of a product to present at a trade show on a certain date and if we cannot use the prototype in any other way, missing the deadline will cause a 100% waste of project investment and a 100% loss of opportunity. Our loss will be the sum of the investment costs and the unrealized profit from using the product.

Other special cases are the projects that have a very high social Cost of Time. Delaying the use of a new vaccine, for example, can lead to loss of life (high deferred use cost).

Unlike the two examples above, most times we have both lost opportunity and deferred use costs. This occurs when the project delay or postponement leads to a partial loss of opportunity.

Let's look at another project example.

Project Case Study

Suppose a one-month delay results in a 1% reduction in the sum total of the project's life cycle monthly profits. If the expected monthly profit is a constant \$20,000 when the project is not delayed and the expected revenue-generating period is 36 months, the cost of lost opportunity will be \$7,200 for each month of delay. The reason may be that we have missed the opportunity to become a market leader and to gain a significant market share. Project revenue decreases as the delay increases.

In addition, we incur costs for deferred use of project products. If revenue and profit simply shift over time, a one-month delay does not lead to a total loss of the profit for that month, but to a cost of one-month profit shifting - so, we have to account for the time value of money.

Let's assume that the opportunity cost of our capital is 0.6 percent per month. Here, we need to apply a 0.6 percent discount rate on the shifted monthly profits to calculate the cost of deferred use. Since the entire revenue generation period is shifted by one month, we need to calculate the discounted values for each month.

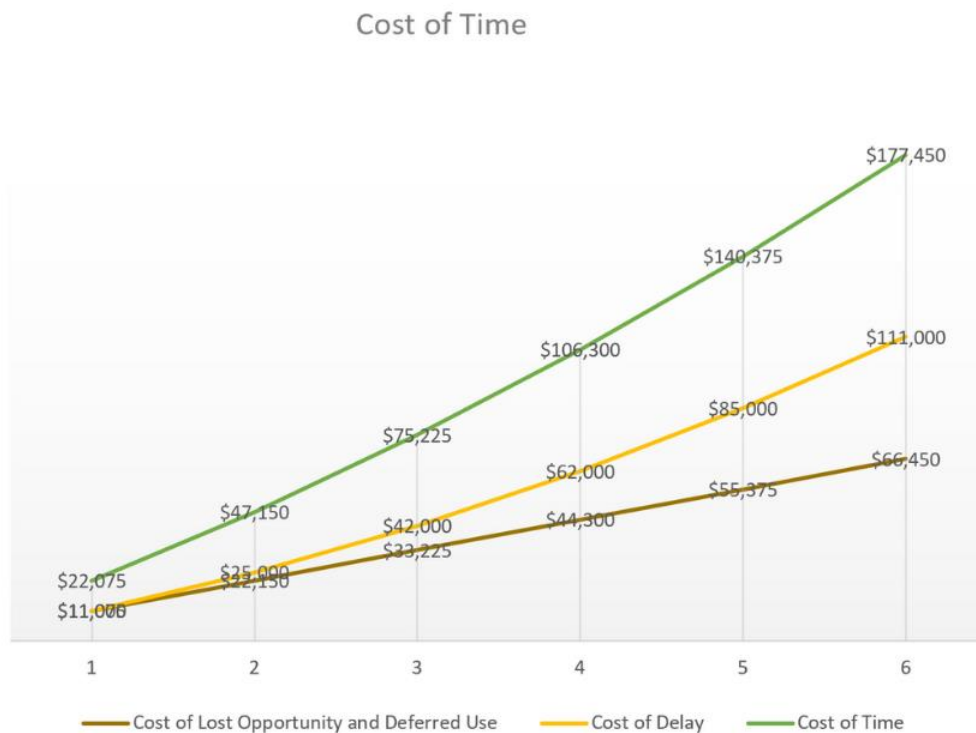
However, the easiest way to perform the calculation is to take into account the shift of income from month 1 to month 37, changing nothing else. We have to estimate the present value of the profit for month 37 at the time of month 1. Based on constant monthly profit of \$20,000 (for simplicity of calculation), this present value is \$16,125. Hence, the lost profit from shifting revenue by one month is \$3,875.

Assuming that the cost of delay is \$8,000 per month plus an additional cost of perishable information component that increase by \$3,000 per month, we show the calculation of the cost of time in the following table:

Table 7.1: Calculation of Cost of Time

	Months of delay					
	1	2	3	4	5	6
Cost of lost opportunity for that month (1)	\$7,200	\$7,200	\$7,200	\$7,200	\$7,200	\$7,200
Cumulative lost opportunity cost (2)	\$7,200	\$14,400	\$21,600	\$28,800	\$36,000	\$43,200
Cost of deferred use for that month (3)	\$3,875	\$3,875	\$3,875	\$3,875	\$3,875	\$3,875
Cumulative cost of deferred use (4)	\$3,875	\$7,750	\$11,625	\$15,500	\$19,375	\$23,250
Cost of lost opportunity and deferred use (5) = (2) + (4)	\$11,075	\$22,150	\$33,225	\$44,300	\$55,375	\$66,450
Cost of delay for that month (6)	\$11,000	\$14,000	\$17,000	\$20,000	\$23,000	\$26,000
Cumulative cost of delay (7)	\$11,000	\$25,000	\$42,000	\$62,000	\$85,000	\$111,000
Cost of time for 1-6 months of delay (8) = (5) + (7)	\$22,075	\$47,150	\$75,225	\$106,300	\$140,375	\$177,450

In this example, the impact of the cost of deferred use is moderate. However, when comparing the CoT for different projects, the revenue profile plays a big role. If we expect Project A to generate revenue for 24 months and Project B for 24 years, with similar investment costs and expected rate of return, the monthly profits and the costs of deferred use of the two projects will differ significantly. It's likely that the difference in the monthly lost opportunity costs will also be significant.

Figure 7.1: Cost of Time

© Lean Project Management Foundation

However, what happens when we've missed the best time to start the project? The opportunity window is already open and the ideal start date is "yesterday" or "six months ago". In this case, we must accept that the reference point for the lost opportunity and the deferred use is the present moment, "today". The Cost of Time for the past is a sunk cost. The methods for calculating the components of the Cost of Time remain the same.

What should we use the Cost of Time for? For at least three things:

- to evaluate the effect of the Cost of Time on project success measures
- to improve our trade-off decisions
- to prioritize projects and tasks (see [*Projects Sequencing/Schedule Prioritization*](#))

The Cost of Time is a variable that affects the project's performance. We need to know how incremental change in the Cost of Time (adding costs for a unit of time, such as one week) affects the profitability. We can do this by analyzing the sensitivity of the project, which can answer questions such as:

- How does the variation of the cycle time or the period of project implementation affect project profitability?
- What are the switching values of the delay or postponement that offset the project benefits and costs, or that reduce the return to the minimum acceptable level?

In the project example above, if a Cost of Time of \$152,000 reduces the return to the minimum acceptable level, the maximum delay we can accept is five months (provided the costs don't increase for another reason).

Failure to account for the Cost of Time can lead to waste, even more so given that the CoT is not static. For example, changing market conditions can lead to a higher CoT that makes the project unviable. To limit the waste, we would have to stop the project.

To summarize, the Cost of Time is caused by project delay (or acceleration) or postponed (or premature) execution which may induce a cost of project delay or acceleration, a cost of perishable information, a cost of regulatory and contractual penalties, a cost of lost opportunity, a cost of premature delivery and a cost of deferred use of project assets. In addition, it may be useful to consider the cost of customer dissatisfaction with late delivery. For dependent projects, we have to take into account the Cost of Time of the delayed downstream projects.

Figure 7.2: Cost of Time Components

Like any other project cost, the Cost of Time is a variable that we should consider when optimizing a project's net benefit.

BEST TIME TO DECIDE: LAST RESPONSIBLE MOMENT VS. MOST RESPONSIBLE MOMENT

The moments when we make decisions or commitments induce time-related costs. If we decide before the best moment, we incur the cost of premature decision. When we miss the best moment, we incur the cost of delaying the decision.

One of the most popular rules in Lean and Lean-Agile states that decisions and commitments should be deferred until the *last responsible moment*. The rationale is that this way we avoid decisions based on assumptions, incomplete information, and lack of knowledge.

We can trace the origin of the rule to defer decisions to one of the Toyota Way principles, formulated by Jeff Liker: *"Make decisions slowly by consensus, thoroughly considering all options."* [1]

"Slow decisions" refers to keeping options open and taking more time to explore them. This warrants decision-making based on good understanding, as the *genchi genbutsu* principle requires.

The defer decisions/commitments rule lives its own life and is interpreted in different ways. It's most often associated with the concept of the "last responsible moment", but also with its variations, such as the "last possible moment" or "decide as late as possible".

The true purpose of this rule is to decide at the best time. What is missing is a definition of the best time. This definition is precisely what the Last Responsible Moment seeks to provide.

Perhaps the most complete definition of the Last Responsible Moment is used in lean construction: *"The instant in which the cost of the delay of a decision surpasses the benefit of delay; or the moment when failing to take a decision eliminates an important alternative."* [2]

There may be some justification for the second rule in this definition, but the crucial rule is the first one, which is about comparing the costs and benefits of delaying a decision. Technically, this rule guarantees the negative effect of the decision, as the cost outweighs the benefit. It would be an improvement to say that the Last Responsible Moment is *the instant when, if a decision isn't taken, the cost of the delay of a decision will surpass the benefit of delay*. But this is not the major problem. Does this rule at all correctly define what's the best point in time to decide?

Let's look at an example to answer this question. It's about a project where investments in design options made in \$100,000 increments increase the probability of realizing the benefits.

Table 7.2: Last Responsible Moment vs. Most Responsible Moment

Moment of making a decision to choose a design option	Investment	Expected life cycle costs	Life cycle benefits	Probability of benefits	Expected life cycle benefits	Expected net benefit	Cost of delaying the decision	Benefit of delaying the decision	Net effect of the decision	Definition of the moment of taking a decision
N	\$500,000	\$1,200,000	\$2,307,000	50%	\$1,153,500	-\$46,500	\$0	\$0	\$0	First Possible
N+1	\$600,000	\$1,300,000	\$2,307,000	55%	\$1,268,850	-\$31,150	\$100,000	\$115,350	\$15,350	
N+2	\$700,000	\$1,400,000	\$2,307,000	62%	\$1,430,340	\$30,340	\$200,000	\$276,840	\$76,840	
N+3	\$800,000	\$1,500,000	\$2,307,000	70%	\$1,614,900	\$114,900	\$300,000	\$461,400	\$161,400	Most Responsible
N+4	\$900,000	\$1,600,000	\$2,307,000	73%	\$1,684,110	\$84,110	\$400,000	\$530,610	\$130,610	
N+5	\$1,000,000	\$1,700,000	\$2,307,000	75%	\$1,730,250	\$30,250	\$500,000	\$576,750	\$76,750	Last Acceptable
N+6	\$1,100,000	\$1,800,000	\$2,307,000	76%	\$1,753,320	-\$46,680	\$600,000	\$599,820	-\$180	Last Responsible
N+7	\$1,200,000	\$1,900,000	\$2,307,000	77%	\$1,776,390	-\$123,610	\$700,000	\$622,890	-\$77,110	Last Possible

Moment N is the First Possible Moment to make a decision to choose a design option, as there are no options available to choose from before that. Therefore, N is our reference point in time for delaying the decision.

N+7 is the Last Possible Moment, because failing to make a decision at this moment eliminates all alternatives. Looking at the net effect of the decision, it's also the most irresponsible moment.

According to the original definition above, N+6 is the Last Responsible Moment because it's "the instant in which the cost of the delay of a decision surpasses the benefit of delay". However, as we can see, N+6 is far from the best decision-making point in time, because it results in a significant expected loss. To be precise, N+6 is the **first irresponsible moment** to take a decision. This is not surprising, because the Last Responsible Moment violates the principles of marginal economics.

When applying the revised definition, N+5 becomes the Last Responsible Moment – the instant when, if a decision isn't made, the cost of the delay of the decision will surpass the benefit of delay. This improves the expected net benefit, but it's clear that this is also not the best moment. At best, N+5 is the Last Acceptable Moment.

The best decision-making point in time is the one that maximizes the project's expected net benefit. This is also when the net effect of the decision is greatest. In our case, this is moment N+3, which we'll call the **Most Responsible Moment**.

Sometimes the last possible moment is the best moment, which makes the concept of the last (responsible) moment look natural. And using "responsible" implies that the decision won't be made at the wrong time.

But as we've seen, the Last Responsible Moment isn't the best moment to decide. When it's applied blindly, it can easily lead to negative results. Instead, we suggest using the Most Responsible Moment, which is based on solid economic principles.

Most Responsible Moment is the instant of making a decision that maximizes the net benefit of the project.

Thus, our recommendation is to defer decisions and commitments until the *most responsible moment*.

TAKEAWAYS: COST OF TIME

- The Cost of Time is an actual cost that influences the project life cycle costs.
- Cost of Time is caused by varying the project cycle time or shifting the project implementation period.
- The Cost of Time components may include Cost of Project Delay, Cost of Project Acceleration, Cost of Perishable Information, Cost of Regulatory and Contractual Penalties, Cost of Lost Opportunity, Cost of Premature Delivery, Cost of Deferred Use, Cost of Customer Dissatisfaction and Cost of Delaying Dependent Projects.
- The reference value for project delay or acceleration is the optimal project cycle time.
- The reference point for the lost opportunity, premature delivery and deferred use is the ideal point in time to complete the project. If this ideal point is already missed, the reference point is the present moment.
- We should use the Cost of Time to evaluate its effect on project success measures, to improve the trade-off decisions and to prioritize projects and tasks.
- The moment when we make decisions and commitments induces time-related costs.
- The rationale for deferring decisions and commitments until the last responsible moment is to avoid actions based on assumptions, incomplete information, and lack of knowledge.
- However, the concept of the Last Responsible Moment doesn't define the best decision-making moment. We recommend using the concept of the Most Responsible Moment instead – which is the instant of making a decision that maximizes the net benefit of the project.

REFERENCES - CHAPTER 7

[1] Liker, Jeffrey K. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill

[2] Lean Construction Institute, Glossary:
<https://leanconstruction.org/pages/learning/education/glossary/#l> (last accessed 2021-11-04)

Chapter 8: Lean Development Life Cycle

DEVELOPMENT LIFE CYCLE MODELS

According to the current understanding, there are five models of project development life cycle [1]:

- Predictive (plan-driven)
- Iterative
- Incremental
- Adaptive (agile)
- Hybrid

Traditionally, it's considered that predictive, adaptive and hybrid life cycles have relative advantages and disadvantages and that each of them may be the best choice depending on the specific project context.

Predictive Life Cycle

This cycle, commonly known as **waterfall** method or approach, comprises successive phases, for example analysis, design, development, testing and implementation. To start any phase, the team must complete the previous one. There is a specialization of tasks within each phase. The team transfers all intermediate deliverables from one phase to another and makes a single delivery at the end of the project. They create detailed requirements and plans upfront and strictly control changes. This is a linear approach for executing in a single pass.

This approach is also called "**sequential**", which we think is a better name because "predictive" does not provide information about the phases of the cycle and can mislead as to the predictability of the outcomes.

Iterative Life Cycle

Iteration means repetition. To iterate means to repeat something until a certain condition is met. The iterative life cycle repeats the development phases (e.g., analysis, design, development and testing) several times improving the deliverable until we meet the completion criteria.

This approach is feasible for creating information-based assets and is generally not suitable for physical deliverables (e.g., a building). Physical deliverables, however, have intermediate information-based products to which the iterative approach applies. For a building, for example, there are architectural and structural drawings, specifications, construction plans, etc. In a broader context, developing models and prototypes is also an iterative approach that applies to any type of project deliverable.

Incremental Life Cycle

Increment means increase. Development starts with implementing a subset of the deliverable functionality (an increment). The team gets a feedback from the development process and from the customer and creates successive versions of the deliverable (increments). The goal is to make each successive increment better and more functional.

Adaptive (Agile) Life Cycle

While each cycle involving an iterative or incremental approach is adaptive (also called evolutionary), the agile development cycle is both iterative and incremental.

Hybrid Life Cycle

A hybrid life cycle combines both predictive and adaptive life cycles.

THE DEVELOPMENT LIFE CYCLE OF LEANPM

The Lean philosophy is evolutionary and many lean practices are iterative and incremental.

Iterative and incremental cycles have been successfully applied in the lean product development and lean startup contexts.

Regarding the sequential (waterfall) life cycle, we need to answer the following questions:

1. Is the evolutionary approach always applicable or sometimes it's just the sequential one?
2. Does the sequential approach have comparative advantages over the evolutionary approach?

Applicability of the Evolutionary Life Cycle

Every human activity is **incremental** and **iterative**.

For example, making a pot out of clay involves several steps by which the pot gradually acquires shape, volume and functional qualities. If necessary, the potter goes back to the previous steps and remakes the pot, or if it was already baked, remakes it to achieve the desired outcome.

After making one pot, the potter strives to make each subsequent pot better. The thousand-year-old evolution of pottery results from an enormous number of pot increments and iterations.

Unlike pottery, the evolutionary nature of human actions isn't always visible. For instance, large-scale manufacturing creates the illusion of linearity in which the product moves in one direction only, and thousands of products do that day after day, in exactly the same way. However, Lean reveals the natural iterative and incremental nature of manufacturing, just like the potter's work, and uses this to improve performance.

Does this apply to projects? Manufacturing is an environment with very little uncertainty. By definition, projects are subject to greater uncertainty than a production line, but if work on a project is predictable and certain, it's better to perform it as an ongoing operation. Therefore, even a project with a very low uncertainty has higher uncertainty than ongoing work such as manufacturing or service operations. What applies in terms of the iterative and incremental approaches to manufacturing is even more applicable to projects.

Have any of us taken part in at least one project that was executed in a strictly linear way, from start to finish? The answer depends on the definition of linearity, but from a purely practical point of view, this is rarely possible or even desirable. In most cases, the linearity of the sequential life cycle, just like its predictability, is an illusion. It gives us a false sense of security and prevents us from reaping the benefits of the evolutionary approach.

We don't have to go far to find many projects with sequential development cycles that incorporate change, refinement and rework cycles, and overlapping phases which indicates evolutionary development. As Don Reinertsen points out, when product developers are obliged to follow a sequential life cycle, rational behavior prevails and most of them informally overlap development phases [2].

Evolutionary development is always feasible, and also generally occurs in projects with a sequential development cycle, taking the form of unplanned, unwanted, or even forced evolution. Not every project

is able to deliver working products at the end of subsequent iterations, but almost every project can demonstrate increments of intermediate deliverables or models.

Does the Sequential Life Cycle Have Advantages?

Among the perceived advantages of the waterfall model are:

- Departmental efficiency
- A logical and intuitive framework
- Minimum customer feedback is needed
- Easy to measure progress
- Easy to manage because the phases are being worked on one at a time and have specific deliverables and review processes
- Better time and cost control because of the detailed upfront planning
- Clear requirements that largely remain unchanged

The departmental efficiency has no impact on the profitability of the project, so this metric is irrelevant.

Logic and intuitiveness are socially determined. Many lean concepts are counter-intuitive, but only because we are taught to think differently. As we find better ways, our understanding and perceptions change. After all, we should judge an approach on the results, no matter how logical and intuitive it may be.

Many of the perceived advantages are not just because of the application of a sequential life cycle, but because of the low uncertainty inherent in certain projects. These are the benefits of clear requirements, better planning and control, and the minimum need for feedback. It's a vicious circle. On the one hand, some argue that the sequential approach is appropriate when there is low uncertainty and, on the other, that the advantages of the approach stem from the low uncertainty.

Is it easier to measure progress with a sequential development approach? Good metrics must show that the project is making progress in producing a net benefit. Some argue that the existence of a detailed plan and well-defined milestones facilitate monitoring and corrective actions. However, in this case progress is measured against the plan rather than towards objectives. Actually, the sequential approach is the least suitable for measuring progress, since it's only at the end of the project that we have finished deliverables that we hand over to the customer.

This model has one possible advantage that is rarely mentioned. The number of transfers from one work process to another is limited by the number of development phases, which translates into lower total transfer (transaction) costs. We should note the following regarding this perceived advantage:

- Typically, the actual number of transfers is greater than planned, because of unplanned changes and rework.
- Transaction costs alone are not a good measure. We need to find an optimal trade-off between the transaction costs and benefits and we don't achieve this optimum when there is only a small number of transfers.
- We can reduce the cost of individual transactions and increase the number of transactions (we look at this in the following chapters).

In summary, this advantage is relative and conditional and cannot justify the use of the sequential life cycle.

The LeanPM Development Life Cycle Approach

In the last sections above, we answered two questions:

1. Is the evolutionary approach always applicable or sometimes it's just the sequential one?
2. Does the sequential approach have comparative advantages over the evolutionary approach?

From the answers to these questions, we can conclude that the evolutionary life cycle approach is always feasible and even natural, and the sequential (waterfall) approach has no advantages over it.

The **LeanPM** development life cycle is based on an evolutionary approach. It has the following core features common for all its variations:

- Customer pull
- Workflow management and optimization
- Incremental and (when feasible) iterative creation
- Small batches of work and fast feedback loops
- Overlapping development (creation) phases
- Integrated, cross-functional and self-organizing teams (Lean-Agile teams)
- Plan, Do, Check and Act cadences
- Synchronized creation and absorption

In a project management context, cadence is the rhythmic sequence of activities performed at regular time intervals. Regular cadence helps lower transaction costs, limit the accumulation of variance, make waiting times predictable and enable small batch sizes. Examples of time-based cadences are regular project meetings, prototyping, testing, reviews. [3]

Periodic planning (e.g., weekly commitment planning), fixed-length iterations and work time-boxes (e.g., weekly or daily work) are also examples of cadences. In general, we'll talk about Plan, Do, Check and Act cadences.

The regularity of cadences should be based on domain, circumstances and coordination and transaction costs and can be empirically adjusted. [4]

The absorption starts before and ends after creation of deliverables, but these two types of activities need to be performed as a single value stream. Absorption should not be pushed by creation, nor should just mirror it. Absorption must be pulled by the customer and synchronized with their capacity and needs. And then absorption will pull creation. An initial Setup stage is another common feature of all lean development life cycles (Agile developers may prefer to call this stage "Iteration 0"). It's a short pre-work stage that may involve:

- Establish and train project team
- Organize team work space
- Establish team ground rules, culture principles, processes and logistics

- Set up tools and technical environment
- Develop stakeholders' Conditions of Satisfaction
- Define project vision
- Refine project milestone/master plan and/or develop product vision and create initial product backlog

Outside the common features, the major differences between the life cycle models are dictated by the possibility or impossibility of applying an iterative approach. Projects in which this is possible should, as a general rule, apply an iterative development life cycle. Examples include knowledge work projects like product development, research, design, software development, process improvement, etc. Iterations are usually not feasible in production projects like construction. We should consider mixed approaches, too. For instance, a design-build project can use an iterative life cycle for the design and non-iterative for the construction.

LeanPM Iterative Life Cycle

The major building block of the iterative life cycle is the iteration – a timeboxed PDCA work unit aimed at creating an increment of potentially absorbable project assets. When absorbed into the customer's value stream, we expect the increment to generate incremental value.

Let's exemplify the iteration with the Sprint of the Scrum framework [5]. Sprint is a time-box of no more than one month, during which the team creates a valuable and usable Increment (or multiple Increments) with minimal external interference. The Sprint length – the heartbeat of development – is fixed for all recurring Sprints. According to Scrum, we can see each Sprint as a project with a goal to build something, a plan, work and an outcome – product increment.

Sprints comprise five components:

1) Sprint Planning

This is a time-boxed collaborative work to plan the upcoming Sprint. Sprint Planning answers what can be delivered in the upcoming Sprint and how will the work be performed.

Key input to Sprint Planning is the Product Backlog, which is a list of all product features, functions, requirements, enhancements and fixes that will be worked on (Product Backlog items). Each item is defined in terms of description, order (priority), estimate and value.

The team selects Product Backlog items for the Sprint within their capacity and plans how to deliver them. The Sprint Goal, the selected items, and the delivery plan form the Sprint Backlog. The team also defines a Sprint Goal – the objective that will be achieved during the Sprint – which informs team why the product Increment is valuable to stakeholders.

2) Daily Scrums

The Daily Scrum is a very short inspect and adapt meeting of the team to inspect progress, track the total work remaining, and plan the work for the next 24 hours. Daily Scrums improve a team's communication, knowledge and decision-making, and identify impediments. Often, this meeting is followed by a detailed discussion or by adapting or replanning the Sprint's remaining work.

3) Development work

The self-managing and cross-functional team performs the work of the Sprint and inspects the resulting increment and the progress towards the Sprint Goal. The team aims at delivering a working Increment of the “Done” product at the end of the Sprint. If the team detects undesirable deviations, they make a quick adjustment. The development team tests each Increment and ensures that all Increments work together.

4) Sprint Review

The Sprint Review is an informal review meeting of the team and key stakeholders at the end of the Sprint. The purpose is to inspect the “Done” Increment and, if necessary, to adapt the Product Backlog.

The team demonstrates the Increment and discusses the work completed during the Sprint – what went well, what were the problems and how they solved them. The participants discuss what to do next in order to provide input into the following Sprint Planning. They review time, cost, capability and market aspects of the next product releases.

As a result of the Sprint Review, the Product Backlog is revised and potential Product Backlog items for the upcoming Sprint are identified. Also, the Product Backlog may be adjusted to accommodate new opportunities.

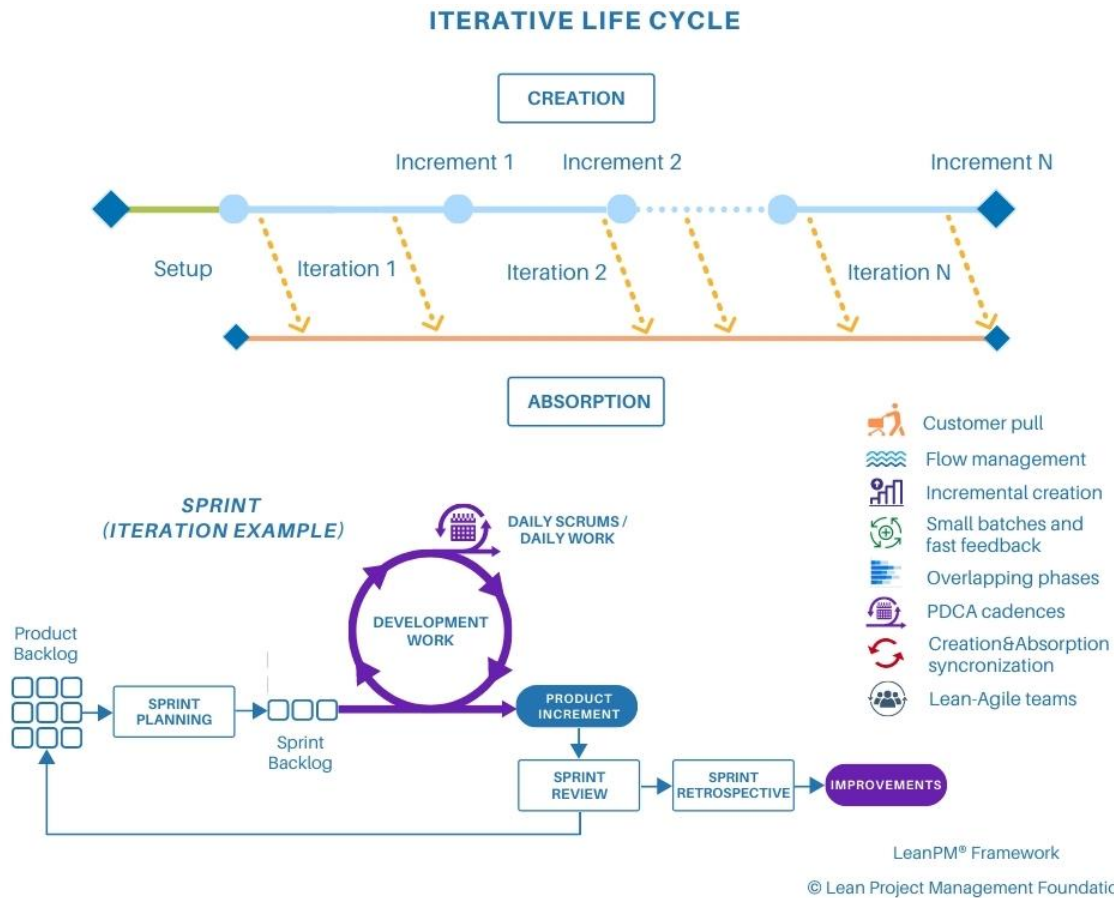
5) Sprint Retrospective

This is a team meeting held between the Sprint Review and the next Sprint Planning that focuses on team’s inspection and adaptation. The team inspects the last Sprint and identifies improvements regarding people, relationships, process and tools. Then, they create a plan for improvements that they will implement in the next Sprint (adaptation to the inspection). Also, the team plans how to improve product quality by enhancing work processes or the definition of “Done”.

Cadences

The Scrum framework doesn’t mention cadences, but uses them. Here are the PDCA cadences of Scrum:

- **Iteration cadence.** These are the fixed-length Sprints that follow immediately one after the other. The Sprint encompasses all other cadences. There are two groups of plan-do-check-act Sprint activities that have a daily and once-per-Sprint rhythm. As there is a limit to the maximum duration of the planning, review and retrospective activities, the Sprint length and the start date of the first Sprint determine the duration and the timing of the other cadence activities.
- **Plan** cadences are Sprint Planning and the daily planning and coordination performed in Daily Scrums.
- **Do** cadences are the development work component of the Sprint and its daily work subsets enabled by the Daily Scrums.
- **Check and Act** (Inspect and adapt) or Improvement cadences include Sprint Review, Sprint Retrospective, and Daily Scrums as formal opportunities for improvement.

Figure 8.1: Iterative Life Cycle

In the iterative life cycle (and in any other evolutionary life cycle), Absorption and Creation must be synchronized and planned together, but they may have different work cadences.

LeanPM Exploratory (Lean Startup) Life Cycle

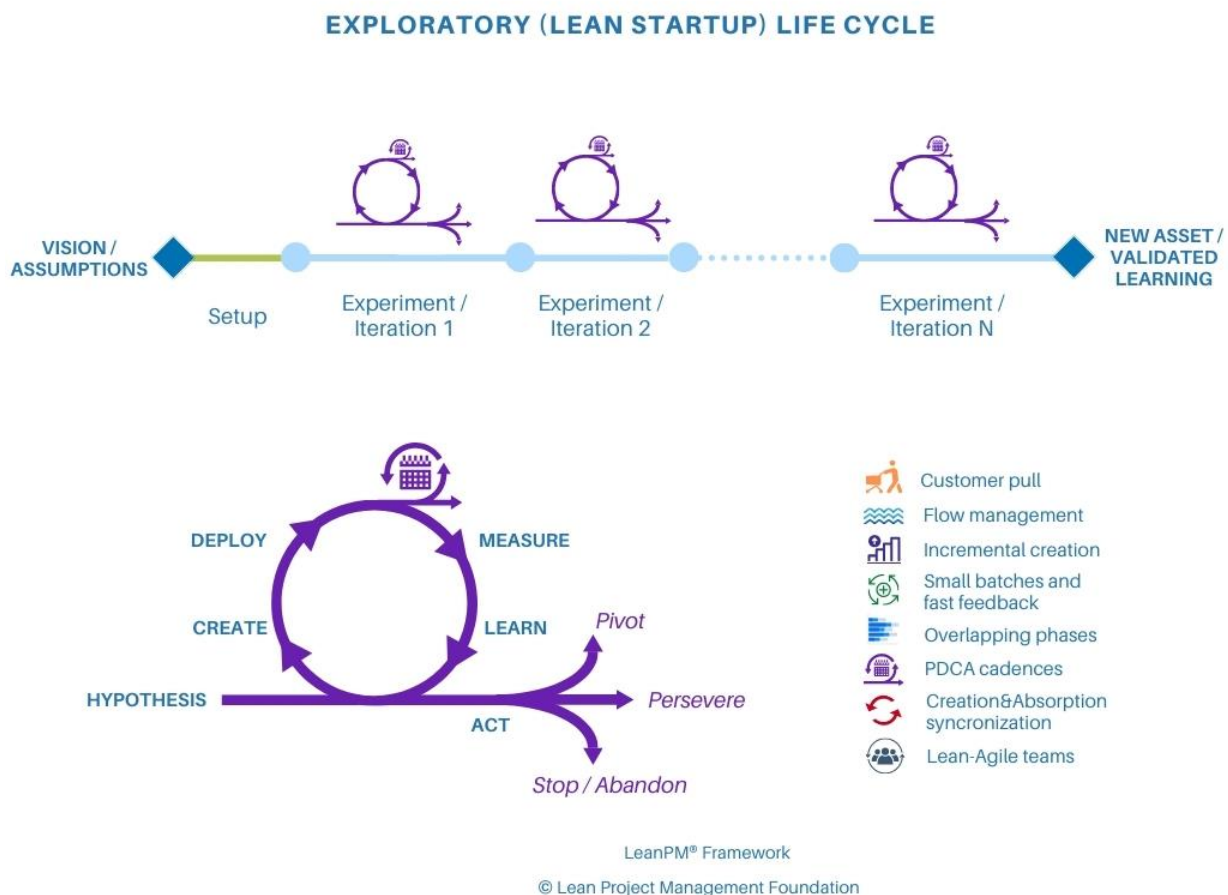
LeanPM Exploratory (Lean Startup) life cycle is a variation of the Iterative life cycle. It's based on Eric Ries's Lean Startup ideas [6]. This life cycle can create value stream assets and generate validated learning under uncertainty, while reducing risk and waste and increasing value.

The essence of the Exploratory cycle is creation through experiments in successive iterations and tight feedback loops. The idea of a new asset (new product, service, value stream, model, technology, method, strategy, growth engine, etc.) can be based on assumptions about its functionality, feasibility and viability. We must turn these assumptions into hypotheses that need to be tested using a Minimum Viable Asset (MVA, which is a more general term that extends the widely used term "Minimum Viable Product").

The Minimum Viable Asset is a version of an asset that enables the gathering of as much information as possible about the hypotheses, with as little effort as possible. This information is analyzed to generate validated learning about the asset. A plan-do-check-act cycle is run and one of the following decisions is taken in the Act phase:

- **Persevere** – stay the course and continue improvements and tests
- **Pivot** – make a major change to test a new fundamental hypothesis [7]
- **Stop** the work because the required knowledge has been acquired or the desired asset created
- **Abandon** the idea, as an important hypothesis about its viability or feasibility has been invalidated

Figure 8.2: Exploratory (Lean Startup) Life Cycle



LeanPM Non-iterative Evolutionary Life Cycle

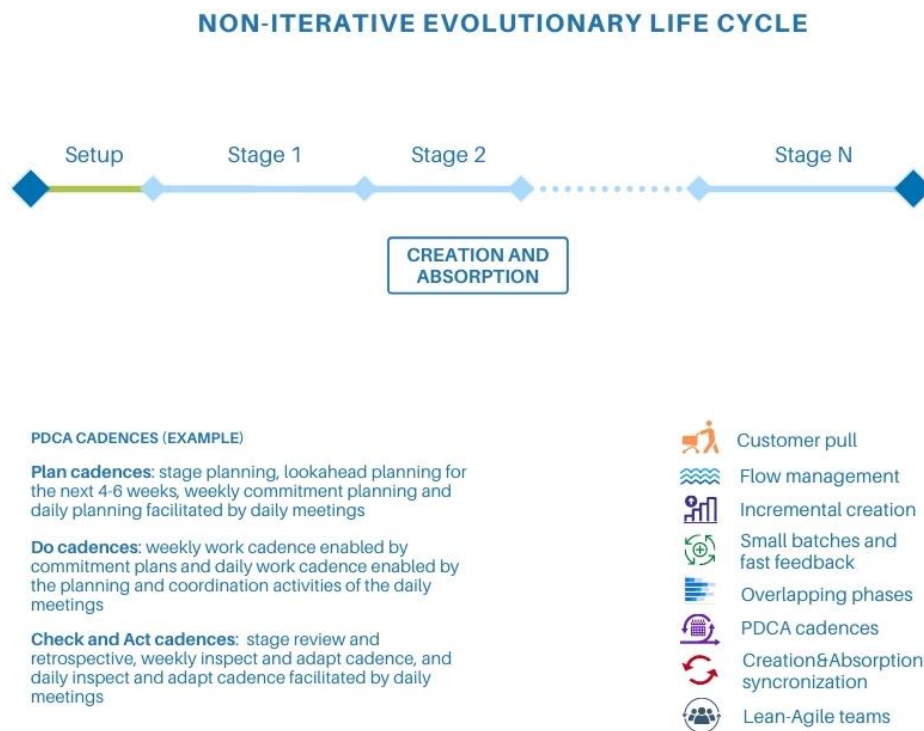
The evolutionary nature of the Non-iterative Life Cycle is ensured by the realization of the key characteristics we've identified above, otherwise, this would be a Waterfall life cycle. The project milestone plan can serve as a starting point for this life cycle. Although not of uniform duration, the stages defined by the main milestones represent the general cadence framework in which the PDCA cadences fit. Note that stages (management phases) don't overlap, unlike the technical phases of development (creation).

Here is an example of PDCA cadences for this cycle:

- **Plan** cadences: stage planning, lookahead planning for the next 4-6 weeks, weekly commitment planning and daily planning facilitated by daily team meetings

- **Do** cadences: weekly work cadence enabled by commitment plans and daily work cadence enabled by the planning and coordination activities of the daily meetings
- **Check and Act** cadences: stage review and retrospective, weekly inspect and adapt cadence for the work performed in the past week, and daily inspect and adapt cadence facilitated by daily meetings

Figure 8.3: Non-iterative Evolutionary Life Cycle



LeanPM® Framework

© Lean Project Management Foundation

LeanPM Continuous Delivery Life Cycle

We can use the Continuous Delivery Life Cycle for both iterative and non-iterative creation and absorption. It has a complete set of PDCA cadences, which, however, are not placed in a common container such as an iteration or stage. The work is pulled from a single backlog when resources become available or following a pull signal which is linked to Work in Progress limits. This cycle is appropriate when work items are similar in size and scope and the work is performed by stable long-term teams.

In the iterative cycle, the time-boxed iteration largely determines the cadences and limits their flexibility. For instance, the iteration time-box does not allow re-prioritization of work items before the next iteration, even if requirements change often. In the continuous delivery life cycle, the cadences depend on natural needs to maintain flow. Each PDCA cadence is allowed to adjust and differ from other cadences, including planning, prioritization, input queue replenishment, development, coordination, delivery, release and review cadences. [8]

For instance, the Kanban method, which uses a continuous delivery life cycle, has seven feedback cadences (some of them are at portfolio level) [9]:

- Strategy Review (quarterly)
- Operations Review (monthly)
- Risk Review (monthly)
- Service Delivery Review (bi-weekly)
- Replenishment Meeting (weekly)
- The Kanban Meeting – daily coordination, self-organization, and planning review (daily)
- Delivery Planning Meeting (per delivery cadence)

The Continuous Delivery Life Cycle works best when it's possible to create and absorb valuable project deliverables regularly. Such are the cases with implementing a series of short regular projects and with projects that regularly create absorbable increments.

Not every project (even not every software development project) can create a regular flow of valuable deliverables. Often, we have to create a certain configuration of deliverables or the whole configuration of project deliverables in order to have something of value for the customer. For example, a completed bridge section has no usable value for the customer until and if we complete the entire bridge. The "earned value" of the completed section is conditional and uncertain. In such a case, we can't talk about continuous delivery.

UNCERTAINTY AND THE LEAN DEVELOPMENT LIFE CYCLE

The way we should apply the **LeanPM** development life cycle approach depends on the level of uncertainty.

Uncertainty arises from a lack of knowledge or from our inability to predict future events (unpredictability). There are two types of project-related uncertainties.

We associate **parameter uncertainty** with variations in the input values used in the project success model, that is the cost-benefit analysis model. These are the values of the various components of the project's benefits and costs, and the parameters that affect them, such as revenue, time, scope, quality, effort and team productivity.

Model uncertainty affects two project models. The first one is the model of cost-benefit analysis. The main issue here is the uncertainty about the trade-off relationships between the parameters of the model.

The second one is the model of the project logic. There is an uncertainty in the causal relationships between the project's logical elements. These relationships are a hypothesis of change and achievement of net benefits, based on assumptions and affected by risks. In fact, the uncertain logic leads to parameter uncertainty. But this uncertainty also signals potential significant project waste, including strategic waste.

Uncertainty may have both positive or negative effects on the project.

Uncertainty exists due to **lack of knowledge** and due to **variation**. The prime factors influencing project uncertainty due to variation are:

1. Variability of requirements and the development process
2. Variability of the project environment
3. Variability of resource supply and demand
4. Project complexity

Let's review each of these four factors.

1) Variability of Requirements and the Development Process

This is not about setting restrictions on changing requirements and the development process, but about the probability of such a need to arise. If we use a well-tested and proven benefit creation model in a stable environment, and if our experience with similar projects allows us to determine the values of the success model parameters with high accuracy, there will be little need for change.

An example would be projects that replicate existing value streams. One such project would be the construction of an electrical substation of standard design based on our experience of building many similar substations. Another example could be an event project as part of a series of similar events from which we have gained considerable experience. By contrast, an innovative project of an unknown nature may have high variability in the requirements and development process.

2) Variability of the Project Environment

The uncertainties of the project's organizational, market, legal, social and physical environment may vary widely. The demand for a project product, or the cultural-driven reactions of stakeholders, are some of the many possible variables.

3) Variability of Supply and Demand

Both demand and supply of resources can change dynamically during the project execution, making it more difficult to align them at any given time.

4) Project Complexity

According to the Association for Project Management, a complex project may involve [10]:

- Many interrelated subsystems or sub-projects
- Interaction with several organizations or units
- Several phases which may overlap
- Several different disciplines
- A need for a wide range of project management methods, tools and techniques

As complexity increases, it becomes more difficult to model a project's behavior due to the increasing number of interactions both within the project and between the project and the environment. Complexity increases project model and parameter uncertainty.

PROJECT UNCERTAINTY CONTINUUM

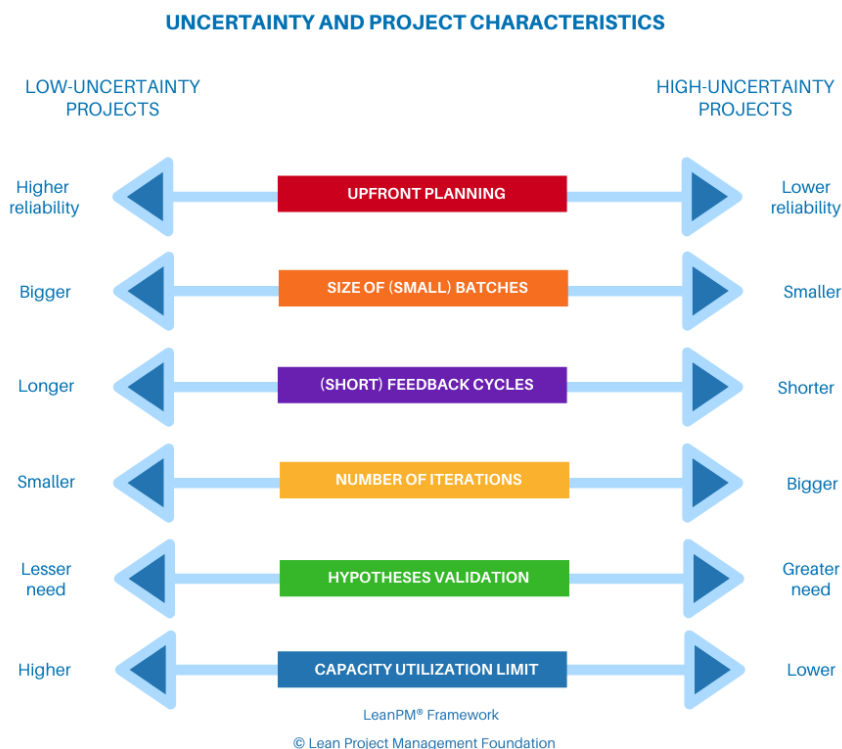
Projects take up the entire spectrum of the uncertainty continuum. At one end are the low-uncertainty projects (availability of knowledge; low requirements, process and environment variability; low resource supply and demand variability; low project complexity) and at the other end are the high-uncertainty projects (lack of knowledge; high requirements, process and environment variability; high resource supply and demand variability; high project complexity). Most projects fall between the two extremes.

Although all projects should use iterative (when feasible) and incremental creation process with small batch sizes of work items and frequent feedback loops, uncertainty affects how we should apply this evolutionary approach.

Project characteristics that uncertainty affects are:

- Reliability of upfront planning
- Required size of small batches of work
- Required frequency of feedback loops
- Required number of iterations
- Need for hypotheses testing and validation
- Planned capacity utilization

Figure 8.4: Uncertainty and Project Characteristics



What we show in the figure are guidelines. We will look at the project development cycle in more detail in the *Managing Creation and Absorption* chapter.

TAKEAWAYS: LEAN DEVELOPMENT LIFE CYCLE

- It's traditionally believed that predictive (waterfall), adaptive and hybrid life cycles have relative advantages and disadvantages, which makes each of them the best choice in a specific project context.
- However, all human activities are incremental and iterative, adaptive and evolutionary, and projects are not an exception. The evolutionary approach to project development processes always applies, and the sequential (waterfall) approach doesn't have advantages over it.
- The **LeanPM** Framework uses an evolutionary development life cycle which is incremental, iterative (when feasible) and entails customer pull, overlapping development phases, small batch sizes of work items and fast feedback loops, workflow management and optimization, Lean-Agile teams, PDCA cadences, and synchronized creation and absorption.
- Variations of the evolutionary development life cycle are the Iterative Life Cycle, Exploratory (Lean Startup) Life Cycle, Non-iterative Live Cycle, and Continuous Delivery Life Cycle.
- The evolutionary development approach should be tailored to the level of project uncertainty regarding the planning process, the size of the small batches of work items, the frequency of the feedback loops, the required number of iterations, the need for hypotheses testing and validation, and the capacity utilization limit.
- The two types of project-related uncertainty are parameter uncertainty - the variability of the project's costs and benefits components, and model uncertainty - the uncertainty about the causal relationships between the project's logical elements and about the trade-off relationships between the parameters of the cost-benefit model.
- The key factors in project uncertainty are the lack of knowledge, the variability of the requirements and the development process, the variability of the project environment, the variability of the resource supply and demand, and project complexity.

REFERENCES - CHAPTER 8

- [1] A Guide to the Project Management Body of Knowledge (PMBOK Guide), Sixth edition (2017). Project Management Institute
- [2] Reinertsen, Donald G. (2009). The Principles of Product Development Flow: Second Generation Lean Product Development. Celeritas Publishing
- [3] Ibid.
- [4] Anderson, David J. (2010) Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Hole Press
- [5] Ken Schwaber & Jeff Sutherland. The Scrum Guide, 2017 and The Scrum Guide, November 2020
- [6] Ries, Eric (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Currency
- [7] Ibid.
- [8] Anderson, David J. (2010) Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Hole Press
- [9] Anderson, David J. and Andy Carmichael PhD, FBCS. Essential Kanban Condensed. First digital version, 17 April, 2016. Referenced version 28 July 2016. Available for download via leankanban.com/guide.
- [10] Association for Project Management Registered Project Professional – RPP Competences (2015). Association for Project Management (apm.org.uk/media/4187/rpp-competences.pdf) (last accessed 2021-11-04)

Chapter 9: Lean Project Planning

INTRODUCTION TO LEAN PLANNING

Planning is defining the steps and the resources required to achieve a desired outcome.

The lean project itself is a plan to create net value for the customer and the other stakeholders. Lean planning is not a separate phase of the project cycle, but takes place in each phase, as part of the PDCA cycles. In terms of the flow of value, the project plan is a design of the project value stream.

The primary purpose of lean project planning is to enable the achievement of project objectives by informing decision making and defining the means and actions. Planning reduces uncertainty, establishes trust and facilitates a reliable flow of value. Plans are means of transparency, collaboration, alignment, direction and coordination.

Planning is often seen as the dividing line between traditional project management and Agile. But there is nothing wrong with considering project management as plan-driven. Regarding planning, the difference between the “traditional plan-driven” and the “agile-driven” project management isn’t the presence or absence of a plan. The major difference is that traditional planning is more “predictive” than adaptive, while Lean-Agile planning is more adaptive rather than predictive, and it’s generally more intense. “Predictive” planning tries to plan the unknowable, which is both wasteful in itself and causes waste. Adaptive planning plans how to make unknowable knowable and thus effectively increases predictability.

Each lean project plan is a combination of a plan to execute and a plan to experiment and validate. A plan to execute can work for what’s known, while a plan to experiment and validate should aim at acquiring knowledge and reducing risk of what’s unknown. The right proportion between the two depends on the specific circumstances, but the mental setting of the team should be that each project is a hypothesis that they need to prove empirically.

Lean acknowledges that results are not completely knowable in advance, and planning must be adaptive. The value of a plan is perishable and may change because of:

- New knowledge gained about customer value, stakeholders’ expectations, requirements, deliverables, creation process, technologies, project team, risks and assumptions
- Empirical evidence about team’s productivity
- Stakeholder feedback on deliverables increments
- Changes in project environment
- Revised estimates and forecast of project’s net benefit
- Any discoveries and new information that affect the project

Re-planning shouldn’t aim to bring the project back to the baseline plan but needs to be informed by project objectives and the Conditions of Satisfaction.

Some major differences between traditional and lean project planning are summarized in the table below.

Traditional Project Planning	VS	Lean Project Planning
• The project plan is “the document that describes how the project will be executed, monitored and controlled, and closed” (PMBOK Guide 6) • Predictability mindset • Focus on upfront planning based on detailed requirements and prescribed deliverables • The plan forms a scope, schedule and cost baseline, against which the project progress is measured • Limited plan visibility and transparency • Top-down planning • Estimates become commitments • Push planning		• The project plan is a design of the means, resources, and actions to create net value for project stakeholders. • Learning and adapting mindset • Focus on just-in-time planning; adaptive and evolutionary planning, progressive elaboration of requirements and evolving deliverables • The plan baseline is used to measure plan reliability, but project performance measurement is value-driven • High visibility and transparency • Collaborative and decentralized planning • Estimates and commitments are decoupled • Pull planning

In the following sections, we’ll define the principles of lean project planning.

Planning and plans are necessary for the people involved in projects. That's why, to define the planning principles, we need to look at how projects work from a social point of view.

PLANNING AND THE NETWORK OF AGREEMENTS

It's often said that for a project to be successful, a common goal must unite project participants. That's true, but it would be simplistic to presume that this common goal should be to create value for the customer. In fact, for a project to be successful, each actor must profit from it, and thus the actors become each other's customers.

The benefit that everyone wants to receive is expressed in **expectations** to be realized both through their own actions and through the actions of others. People seek to secure the actions of others by gaining their **commitment** [1]. The actors engage in conversation, discussions and negotiations that result in a **network of agreements** (NOA) between them. These are *formal and informal, conscious and unconscious* arrangements.

In these agreements, *rights* take the form of *expectations* and *obligations* take the form of voluntary *commitments* to undertake certain actions or provide something. In doing so, the expectations of one party become the commitments of the other party and vice versa. Each actor becomes part of different bilateral and multilateral agreements, in which they play both the role of a customer (through expectations) and the role of a performer (through commitments).

The network of agreements is a system that evolves, and we must balance it to ensure coordinated actions between the parties and project success. The following is required for the balance:

- Harmony between the interests of all parties involved and collaborative behavior; relational agreements that prevent local optimization (relational contracts)
- Shared understanding and alignment – clear, realistic and acceptable expectations and commitments that others understand well
- Trust between the participants that the commitments will be fulfilled
- Correspondence between all expectations and all commitments (to meet expectations with commitments and commitments to match expectations)
- Common understanding of the uncertainties associated with the realization of expectations and commitments
- Common agreement and understanding of the decision-making principles in the project
- Continuous evolution and rebalancing of the system

As the balanced NOA includes both the ends and the means of the project in their entirety, hence the first principle of lean project planning is as follows (numbering of principles is only for convenience and doesn't show their relative importance):

1

First principle of lean project planning

Creating agreements between project participants is a planning process, and the evolving network of agreements is the ultimate project plan.

Only through the NOA can we do something meaningful in a project. Project plans that are not part of the NOA bring disharmony. It follows that every plan in the project must become part of the NOA. The plan should be a tool for meeting expectations by making and realizing commitments. Therefore, the plan should be an agreement. Making commitment doesn't imply that the probability of its fulfillment is 100% percent (although this should be one's idealistic aspiration). The probability varies for specific plans and circumstances, but the parties must understand and accept it. Here we come to the second principle:

2

Second principle of lean project planning

Turn all project plans into viable agreements with acknowledged uncertainty.

The second principle suggests that the top-down plans pushed to the project team are of little use. Plans can only become viable agreements when planning is decentralized, collaborative and consensual and when people closest to the workflow ensure greater reliability through pull planning. See more about the pull planning in the *Push and Pull Planning* section below.

3

Third principle of lean project planning

Sound plans require decentralized, collaborative, consensual pull planning with active involvement of the people closest to the work.

Viable agreements require high trust between the parties. Trust depends on building positive and open relationships, and on adhering to commitments based on reliable planning. Because of the great importance of trust, we have a specific principle for it:

4

Fourth principle of lean project planning

Planning and plans must build trust.

The NOA is a dynamic system that is constantly changing and needs rebalancing. Sources of change can be:

- Subjective (internally-motivated) change in the expectations or commitments of the stakeholders
- New information, knowledge or circumstances that affect expectations or commitments or actors' ability to realize them
- Feedback on project performance and the agreed improvement actions

The dynamic rebalancing of the evolving NOA system is an act of adaptive and continuous planning. We can achieve continuity through periodic planning on a cadence, complemented by ad hoc, on-demand planning, as the unbalancing does not happen in periodic steps.

5

Fifth principle of lean project planning

Plan adaptively and continuously.

So far, we have defined five principles of lean project planning. Let's continue with other perspectives on planning to look for more principles.

PDCA CYCLE AND PLANNING

In the Initiation-Planning-Execution-Closing cycle of traditional project management, the important component of learning and improvement is missing. Thus, planning itself is also not subject to systematic improvement.

In lean project management, we should view planning as part of the Plan-Do-Check-Act (PDCA) cycle through which we realize the project objectives.

It may seem that PDCA is a single and unified cycle, but in fact, we should consider it as composed of individual cycles for each of its phases. The planning PDCA cycle promotes continuous improvement and includes:

- **Plan:** planning how to carry out the planning process
- **Do:** performing planning
- **Check:** checking the quality of planning
- **Act/Adjust:** learning and improvement of the four planning processes (plan planning, do planning, check planning, adjust planning)

This is an example of continuous improvement of planning [2]:

- Determine the percentage of completed activities from the total number of planned activities over a given period (Percent Plan Complete – PPC). PPC measures plan reliability.
- Identify causes for noncompletion.
- Act – remove the causes to improve planning and aim to increase PPC.

Here is how we can improve plan value and reliability:

- Perform continuous estimating and planning
- Assess the plan's value and reliability on a cadence
- Actively remove constraints to the workflow to improve plan reliability
- Learn and adapt to improve plan value and reliability

6

Sixth principle of lean project planning

Continuously improve planning.

PLANNING CONSTRAINTS

The project plan must ensure that we achieve the best outcome within the existing constraints. Constraints are set by external circumstances, third parties, the customer or other stakeholders. Let's look at how constraints affect planning and how to set constraints which are under the direct control of the customer and the project team.

There can be many constraints in a project, but the common ones are time, cost, scope and, to some extent, quality. For some types of projects, regulations and location are also very important.

Besides the legal factors, probably the only constraint that we should consider fixed by default is the minimum required profitability of the project. Without achieving it, the project makes little sense. Conditions of Satisfaction follow next, but they are negotiable.

Traditional project management assumes that, through proper upfront planning, we can precisely define the project scope and we may not need to change it. In turn, the fixed scope makes it possible to assess and fix time and cost within tight tolerances. It turns out, however, that projects can rarely be completed within a fixed scope, time and cost. Instead of over-constraining, we need flexibility in one or more of these variables.

Fixing scope

We can define project scope as the configuration of project deliverables with their features, functions and relationships. As the primary consideration for the scope is benefit and cost, it should be quite normal that it varies to improve the outcome.

Typically, we can achieve project objectives through an unlimited number of alternative scope configurations. Fixing one of all possible configurations means that we consider it to be complete and the best one at the outset of the project. This is equivalent to accepting that a hypothesis needs no proof or deciding with incomplete information. Therefore, we don't recommend fixing the scope. Instead, we might consider certain range limits of scope, based on experience with similar projects.

The progressive detailing, narrowing, prioritizing and improving of the scope begins with the choice of project approach, goes through different stages as, for example, a choice of design option, and goes on until the very end of the project.

Fixing scope often results in including items that are not absolutely necessary. A better solution might be to leave open the possibility to prioritize scope items within the other project constraints. The Minimum Viable Project (and the Minimum Viable Scope respectively) is also a good approach to prevent fixing a scope that is too large.

Fixing time

From a practical point of view, we will always have some kind of time limit. Projects don't go on indefinitely.

Apart from regulatory requirements, the primary rationale for limiting time should be the cost of time.

Another aim for fixing time along with cost might be to ensure that we complete high-value work by a fixed date when we correctly prioritize scope items.

A third reason for fixing time might be to foster innovation, when we set a stretch goal for cycle time improvement.

Fixing costs

We must always consider the total life cycle costs of the project and the expected benefit-cost ratio. From a customer's perspective, operation, maintenance and business costs are expected to be financed from the revenue, and a constraint usually applies to investment costs. The limits depend on the customer's available funds (there is always some limitation) and the amount they are prepared to invest (the target cost).

When the customer sets a target cost, the project team should aim to optimize the configuration of scope, time, quality and recurrent costs to maximize the return. Therefore, constraining the costs is a valid approach, unless the target cost doesn't accomplish a minimum viable project.

Setting the target cost below the industry benchmark to spur innovation might also be considered.

PLANNING ITEMS

Planning items provide guidance for executing the project. The project scope is translated into planning items such as milestones, stages, iterations, user stories, features and tasks, and dependencies between them. In turn, the planning items drive resource requirements and cost and time estimates that provide feedback for refining the scope.

Milestones

Milestones are the backbone of the overall project schedule. These are events that mark significant stages in the project's progress and facilitate flow – key decisions, deliverables, handoffs, and coordination and integration points that release subsequent major stages of work.

Milestones aggregate individual activities and are more reliable than them, as they pool variability from independent sources.

The project team's common understanding of these key events informs the project's conceptual plan.

Stages

Stages are common building blocks of the non-iterative evolutionary life cycle. Stage planning is the first (macro) level of detailing the plan in terms of the local (stage level) milestones, tasks, dependencies, sequence, and priorities. The next level of increasing detail could be the regular look-ahead planning that also emphasizes making tasks ready for execution and removing constraints.

Iterations

These are the key components of the iterative life cycle. Iteration is a time-boxed PDCA work cycle aimed at creating an increment of project assets that can be absorbed into the customer's value stream. If absorbed, we expect the increment to improve the capability of the value stream which would generate incremental value.

Features and User Stories

Features and stories are product backlog items used in iterative development.

Features are higher-level product attributes which can be decomposed to become user stories – small slices of user functionality described from an end-user perspective. Upon completion, the user stories selected for the iteration add value to the product increment. When user stories are clear, self-contained, with limited dependencies and achievable within a single iteration, they can be a very effective means for planning and executing work.

We can use a completed user story as a proxy measure of generated customer value, and thus it's a more relevant planning item than tasks. The actual value will depend on whether the customer will use the feature and whether this will bring the expected positive effects.

Tasks

Tasks are the lower level and most detailed items of planning. They can give the most precise guidance for performing the work. Completing tasks doesn't create value in itself, and we must pay special attention to task relevance.

We need to plan four types of project tasks:

1) Creation tasks

These are the workflow tasks for creating project assets, which include value-added tasks of transforming inputs into outputs, and non-value-added tasks such as inspection and transportation.

2) Absorption tasks

By absorption, we mean fully integrating the assets created by the project into the customer's value stream. By using "absorption", we emphasize the need for the project assets to gradually become an integral and natural part of the value stream, creating an improved whole.

The absorption activities have to start at the beginning of the project and continue until its end. We should not plan them as a major effort at the end (or after the end) of the project.

Absorption tasks need to be planned and performed by the integrated project team, including the development team and the customer value stream team.

The change in the value stream begins with the creation of a vision of a new reality, according to which value stream team members change their mindset and mentally transform their way of working. In this way, absorption begins mentally, even before the actual asset creation has begun.

Then, before the team completes the assets, the material absorption activities may include, for example:

- Training and educating value stream team members
- Developing operating procedures
- Preparing infrastructure, equipment, premises and other aspects of the production, operating and support environment
- Hiring and onboarding people
- User testing

The aim is that when the assets are completed, all conditions for their integration are met and all obstacles are removed.

When the assets are created, the absorption continues with testing in a production environment (or equivalent), including beta testing, until all problems are eliminated and the value stream is effectively improved.

3) Preparatory tasks

These are the tasks to make the creation and absorption tasks ready for execution.

For instance, for the execution of a construction task, at least seven prerequisites (or conditions) are needed [3]:

- Construction design
- Components and materials
- Workers
- Equipment
- Space
- Connecting works (prerequisite activities)
- External physical conditions

Typical prerequisites are clarity of work requirements, availability of resources and information, prerequisite work, approvals and decisions. Absorption readiness and capacity of customer value stream are specific prerequisites for absorption tasks. The preparatory tasks provide the conditions for executing creation and absorption tasks by removing constraints.

The make-ready process transforms work that **SHOULD** be done into work that **CAN** be done. [4]

4) Management tasks

These are the planning, organizing, leading and controlling tasks, related to the plan-do-check-act loops of creation management, flow management and value generation management.

This brings us to the next principle:

7

Seventh principle of lean project planning

Plan creation, absorption, preparatory and management tasks.

RELIABILITY OF PLANNING

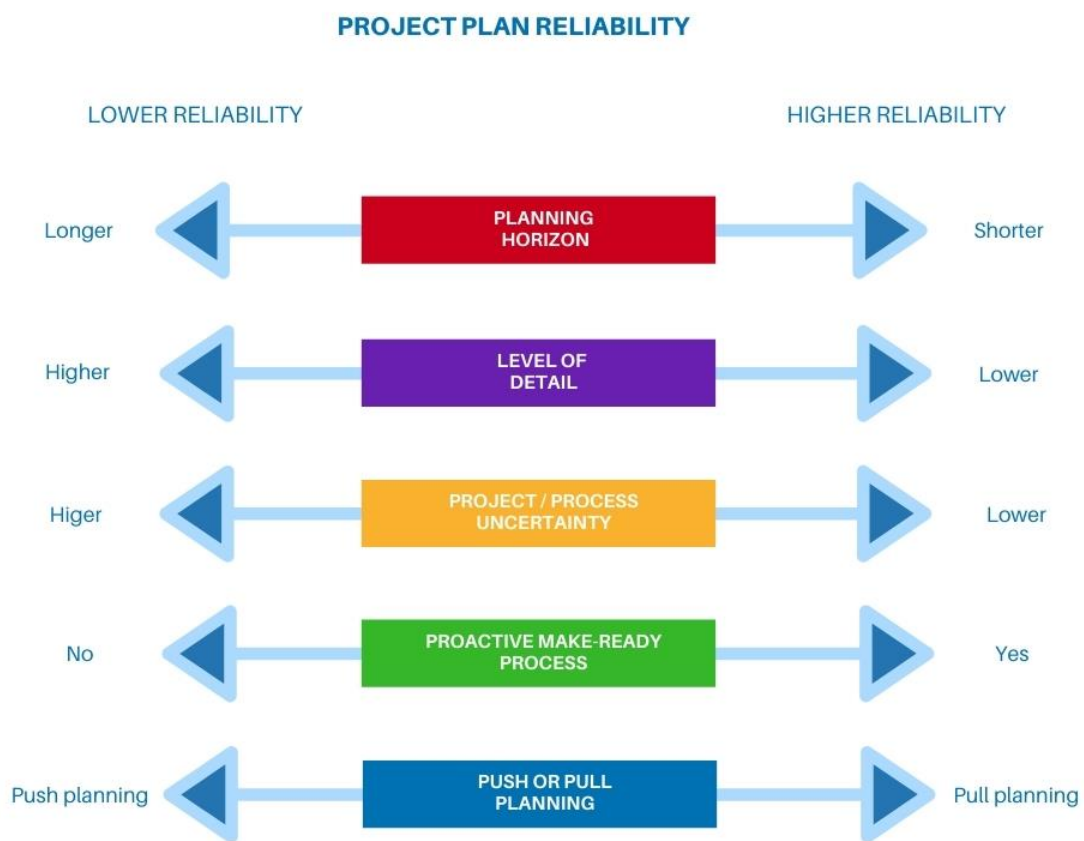
Plans to execute or to experiment should pursue a reliable project workflow. Workflow reliability affects cycle time, quality and cost of projects. Quality and cycle time affect cost and benefits. Thus, workflow reliability affects project success.

Reliable workflow needs a reliable plan. A reliable plan is one that is executed with minimal deviation. 100% reliability is difficult to achieve as no plan is perfect and there is no theoretically perfect plan, but such must be the idealistic goal of continuously improving planning reliability.

Besides team planning skills and the quality of planning, plan reliability depends on several factors:

- Planning horizon
- Level of detail
- Project and process uncertainty
- Make-ready process
- Pull or Push planning

Figure 9.1: Project Plan Reliability



LeanPM® Framework

© Lean Project Management Foundation

The reliability of the plan decreases as the planning horizon expands and details increase. Therefore, we must balance the detail with time. In Reinertsen's words, "the most powerful way to reduce variability in forecasts is to shorten our planning horizons" [5].

Besides the uncertainty of the project as a whole, the reliability of the plan is affected by the nature of the workflow processes. It might be easier to plan to execute than to plan to experiment. Often, we can plan execution processes (e.g., construction) more reliably than exploratory and generative processes (e.g., design).

Successful project performance requires a certain level of detail of the plan, which we need to balance with the planning horizon. We must balance the greater project and process uncertainty with a further shortening of the planning horizon – up to the extremity of continuous planning.

We achieve balancing with multi-level, just-in-time planning, where we combine the shortening of the planning horizon with increasing detail and vice versa.

A proactive make-ready process is crucial, to remove impediments and to prepare sound (execution-ready) assignments that can become a part of lowest-level plans.

Pull planning increases reliability as it helps to make commitment plans only for assignments ready for execution, and the team has the capacity to perform.

The output of the planning at each level is the design of the process of creation and absorption with increasing detail, including work methods, workflow steps, dependencies, human capabilities and effort, equipment, tools, materials, prerequisites, assumptions and risks.

We already included pull planning in the third principle, and this is what the wording of the eighth principle looks like:

8

Eighth principle of lean project planning

Perform multi-level, just-in-time planning and a proactive make-ready process to improve planning reliability.

MULTI-LEVEL PLANNING

Let's look at specific models of multi-level planning.

Agile planning has five levels: two portfolio level - Product Vision and Roadmap and three project level plans - Release Plan, Iteration Plan and Daily Plan.

- The **Release Plan** is a long-term plan that sets the general framework of scope (which products or features will be delivered), time (the delivery schedule) and cost. The plan items are features and user stories which the team estimates in story points or ideal time. The team organizes the release plan in iterations to which they add user stories.
- Iteration planning refines and adjusts the iteration components of the Release Plan. The planning items of the **Iteration Plan** are the user stories that the team commits to for the upcoming iteration. The team calculates their capacity for the iteration, reviews, refines and estimates the stories from the team backlog, elaborates acceptance criteria and selects prioritized stories for the iteration backlog until they run out of capacity. Iteration planning may involve breaking down user stories into specific tasks.
- The **Daily Plan** is a refinement and adjustment of the Iteration Plan. At the daily coordination and planning meetings, each team member reports what they did yesterday, what they'll do today, and what obstacles are hindering the progress. Team members work together to remove blockers and impediments.

As we can see, this model balances the planning horizon with the level of detail and ensures effective pull and commitment by the team. It may be useful to have a greater focus on making user stories/tasks ready for execution and on proactively removing obstacles in advance.

The lean construction model of planning has also three levels: Initial Planning, Lookahead Planning and Commitment Planning [6]:

- The **Initial Planning** pushes completions and deliveries onto the project through the project budget and schedule it produces (what *should* be done).
- **Lookahead Planning** details and adjusts initial schedules and budgets matching resource demand and supply (what *can* be done). The activities in the lookahead schedule are screened to start a make-ready process to prepare sound (execution-ready) assignments. The output of this process is a buffer (a backlog) of sound assignments, from which to pull assignments for the commitment plans. The typical time horizon of the lookahead planning is 4-6 weeks. When multiple teams are involved, lookahead planning helps to coordinate them by planning and handling their dependencies.
- When resources are actually received and prerequisites are completed, the **Commitment Planning** establishes what *will* be done (the commitment). Most often, the commitment plan is a weekly work plan and its assignments must meet the criteria for soundness, sizing, and sequence. The commitment plan is a foundation for continuous improvement. To improve productivity and increase workflow reliability, after the team completes the plan, they calculate a Percent Plan Complete (PPC) as a ratio between the number of completed assignments and the total number of assignments for each week. Reasons for noncompletion are identified and acted upon and the aim is to increase the quality of planning as measured by PPC.

- The three levels of planning are complemented by **Methods Planning**, which specifies the methods of doing the work with an increasing detail at each subsequent level of planning.

The gap between what *should* be and what *will* be done shows that we can implement the principles of commitment and pull planning to the greatest extent at the lowest level of planning, when the people closest to the work plan for the imminent future.

RELATIONSHIP BETWEEN PLANNING AND CONTROL

Planning and control are interrelated and meaningless without each other. Planning sets goals and the course of action to achieve them and control ensures that goals are achieved. Control provides feedback to planning so we can adjust it. Therefore, we must plan with control in mind.

Unlike the thermostat model, lean project control is not just about monitoring progress against the plan, identifying variations and applying corrective action as needed.

The lean control system [7] aims to "*cause events to conform to plan, or to identify as early as practical the need for replanning*" (Ballard) in sharp contrast to the traditional project control that tries to bring the performance back to the plan, following after-the-fact detection of variances. [8]

Lauri Koskela proposes five principles of a lean control system [9] [10]:

1. To minimize the work in suboptimal conditions, the assignments should be sound with regard to meeting all prerequisites for their completion.
2. Percent Plan Complete (PPC) is used to measure and monitor the realization of assignments. The focus is on plan realization to reduce the risk of variability propagation to downstream steps.
3. Causes for noncompletion are systematically investigated and removed and thus continuous improvement is realized.
4. A buffer of sound tasks is maintained for each team, to avoid lost production due to starvation, or reduced productivity due to suboptimal conditions.
5. The prerequisites of upcoming assignments are actively made ready in lookahead planning to ensure that all the prerequisites are available and to avoid accumulating a buffer inventory that is too large.

Although designed for construction, these principles can be a good basis for creating a lean control system for a wide range of projects.

We can summarize the relationship between planning and control with the ninth principle of planning.

9

Ninth principle of lean project planning

Plan with lean control in mind.

PUSH AND PULL PLANNING

Customer Pull

Customers pull value, says the popular lean principle.

The ultimate customer (or the customer avatar in a startup project) pulls a project that will generate net value. The pull is the basis for defining the main project variables, such as expected benefits and costs, lead time, scope and quality.

This pull, however, rarely works automatically like in a supermarket where customers can pull any product sold there. It works when there is an agreement between the customer and the project team (actually, a customer-performer agreement within the integrated project team). When we consider all people expecting to benefit from the project, we can say that the project is pulled by the network of customer-performer agreements between the stakeholders.

"The next process is your customer."

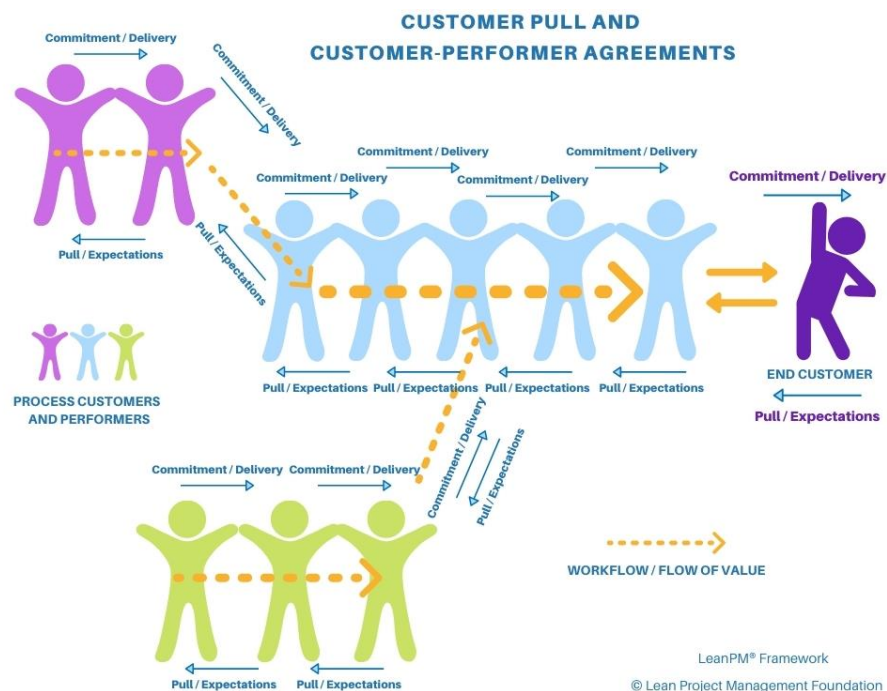
Kaoru Ishikawa

Within the team, there are similar customer-performer arrangements between the upstream and the downstream sub-teams. The sub-teams responsible for the downstream processes (process customers) pull the work from the upstream processes.

Note, however, that the workflow isn't strictly unidirectional and the work and information move back and forth with the process customer and the performer exchanging roles.

As in rugby, the team goes the distance "as a unit, passing the ball back and forth" [11].

Figure 9.2: Customer Pull and Customer-Performer Agreements



LeanPM® Framework
© Lean Project Management Foundation

The team needs to understand the value for the end customer and they have to agree about expectations and delivery. Then, the team plans backwards how to create the value.

The process customer sets requirements and expectations for the upstream process (pull), which are subject to negotiations with the performer. Negotiations result in an agreement and commitment from the performer. The commitment must be also an act of pulling from the work backlog because the performer should commit only to meaningful work that **will** be done. That is, only to sound assignments (i.e., meeting all prerequisites for their completion) that don't exceed performer's capacity and whose completion is expected by the downstream process customer.

When the performer completes the agreed work and delivers the outcome to the process customer, new work is released that can be completed to satisfy the pull by the next downstream process. The current customer becomes a performer for the next process customer. A Work in Progress limit will prevent pushing work down the process.

The role of the process customer is to:

- set their requirements and expectations
- negotiate the work to be completed with the performer
- collaborate and help the performer
- provide continuous feedback on the completed work
- check and accept the work
- work with the performer and other team members to identify causes for noncompletion and for other issues and act on them

The performer should:

- understand customer's requirements, expectations and conditions of satisfaction
- negotiate the work to be completed with the process customer and commit only to sound and required assignments within performer's capacity (the required work that will be done)
- complete the work as agreed and seek continuous feedback from the customer
- deliver the outcome and if necessary, rework until the customer is satisfied (until the conditions of satisfaction are fulfilled)
- work with the process customer and other team members to identify causes for noncompletion and for other issues and act on them

The customer-performer relationships should not become a means of alienating team members. They must be only an aspect of the continuous collaboration between the members of the integrated team which has collective accountability for the outcome. This collaboration requires intensive, open and, as far as possible, informal, synchronous and face-to-face communication. Formal hand-overs should be minimized.

Push Planning

When work is set up on forecast rather than on actual demand, this is a "push" method. It aims to ensure completion of work based upon top-down planning and an order from an authority. Resources

and information are released according to plan, regardless of whether the downstream process is ready to use them. [12]

Traditionally, the plan is prepared by a small group of people, then sent for feedback to the team. Feedback is usually limited to the scope of work of the team member or sub-team. Then the plan is finalized, and the team is expected to follow it. Regardless of whether such a plan is made with backwards scheduling, it's still a push plan.

Push planning assumes that once a sound and detailed project plan is prepared, it could be dispatched to the team and this will ensure its implementation. The plan is considered a commitment. If deviations are detected, corrective measures may be applied, so that performance meets the plan. If the plan is found to be inadequate, it can be revised and a new baseline can be created to push further work. Implementing the push plan is driven by the actual activities of the first and the next performers, and often the plan gets quickly outdated.

This planning often uses complicated scheduling tools and seeks to anticipate all the details at the outset of the project, including start and end dates, relationships and lag times, to name a few. But push planning is used with multilevel planning, too. Although this improves the reliability of the lower levels of planning, the plan is still about what *should* be done and the team is expected to follow it, no matter what the state of the system is.

Push plans often lack team buy-in and ownership and support silo mentality.

Pull Planning

If work is released and performed when the system is ready to use it, this is a "pull" method. The customer signals that the work is needed and pulls it from the performer. [13]

Effective pull planning itself involves several factors that improve reliability.

The pull planning organizes activities based on the needs of the downstream activities and takes into account what the state of the system is:

- if the process customer needs and is ready to use the outcome of the work
- what are their conditions of satisfaction with the work
- whether all conditions to perform the work are fulfilled/constraints are removed
- if the resource demand for the work matches the supply

Pull planning works best when the team is actively working to achieve the desired state of the system. Thus, effective pull planning facilitates lean control that causes performance to conform to plan and ensures early detection of any need for replanning.

We can assess the state of the system with greater accuracy for a shorter future period. Therefore, in multi-level planning, the ability to use pull increases as the implementation period approaches. For instance, the lookahead plan as described above is a pull plan, but the weekly pull plan is the one by which the team makes a firm commitment.

Pull planning is a collaborative effort of the integrated team across the value stream, including those closest to the work. It breaks silos and ensures plan ownership.

When used effectively, pull planning increases the project's net value by:

- creating what's valued with the expected quality
- improving team productivity as the tasks are performed in correct sequence and the constraints to their completion are proactively being removed
- improving plan predictability as the commitment is made in front of the other team members and accounts for the customer expectations, soundness of the tasks and the work capacity
- eliminating non-value-added steps, optimizing workflow and shortening cycle time
- facilitating lean control and continuous improvement

TAKEAWAYS: LEAN PROJECT PLANNING

Lean project planning differs from the traditional planning in that it:

- Considers the project plan a design of the means, resources, and actions to create net value for project stakeholders
- Applies learning and adapting mindset
- Shifts the focus from upfront to just-in-time planning and planning is adaptive and evolutionary
- Requirements are progressively elaborated and the deliverables are evolving
- Uses the plan baseline to measure plan reliability, but project performance measurement is value-driven
- Is highly visible and transparent, collaborative and decentralized
- Decouples estimates and commitments
- Is based on customer pull

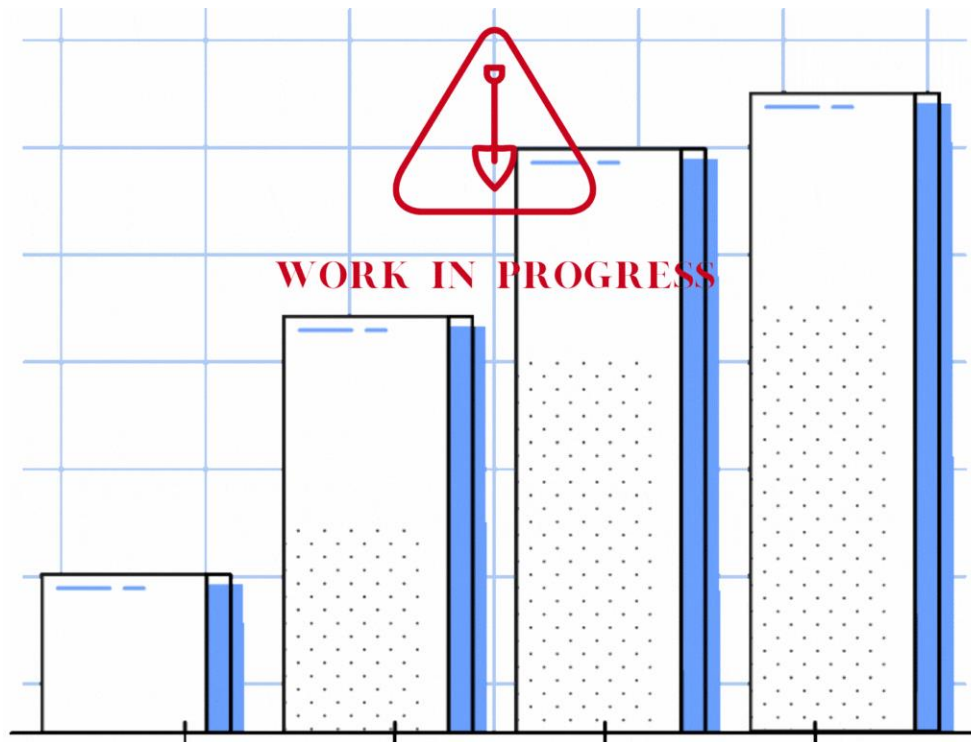
Use the following principles for lean project planning guidance:

1. Creating agreements between project participants is a planning process, and the evolving network of agreements is the ultimate project plan.
2. Turn all project plans into viable agreements with acknowledged uncertainty.
3. Sound plans require decentralized, collaborative, consensual pull planning with active involvement of the people closest to work.
4. Planning and plans must build trust.
5. Plan adaptively and continuously.
6. Continuously improve planning.
7. Plan creation, absorption, preparatory and management tasks.
8. Perform multi-level, just-in-time planning and proactive make-ready process to improve planning reliability.
9. Plan with lean control in mind.

REFERENCES AND NOTES - CHAPTER 9

- [1] We prefer to use **commitment** rather than **promise**, because it implies a dedication to act, not just a declaration or assurance.
- [2] Ballard, Glenn (2000). *The Last Planner System of production control*, PhD thesis, University of Birmingham, Birmingham, UK
- [3] Koskela, Lauri (1999). "Management of Production in Construction: A Theoretical View". Proceedings of the 7th Annual Conference of the International Group for Lean Construction, University of California, Berkeley, CA
- [4] Ballard, Glenn (2000). *The Last Planner System of production control*, PhD thesis, University of Birmingham, Birmingham, UK
- [5] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
- [6] Ballard, Glenn and Howell, Gregory (1998). "Shielding Production: An Essential Step in Production Control." Journal of Construction Engineering and Management, Vol. 124 No. 1, American Society of Civil Engineers, New York
- [7] This lean control system can be considered an extension of the lean construction planning model described above.
- [8] Ballard, Glenn (2000). *The Last Planner System of production control*, PhD thesis, University of Birmingham, Birmingham, UK
- [9] Koskela talks about production control in construction as he views construction as a production system, but in our opinion, the principles can be generalized for lean project control.
- [10] Koskela, Lauri (1999). "Management of Production in Construction: A Theoretical View". Proceedings of the 7th Annual Conference of the International Group for Lean Construction, University of California, Berkeley, CA
- [11] Takeuchi, H; Nonaka, I (1986). "The New New Product Development Game". Harvard Business Review (January): hbr.org/1986/01/the-new-new-product-development-game (last accessed 2021-11-04)
- [12] Lean Construction Institute, Glossary:
<https://leanconstruction.org/pages/learning/education/glossary/#p> (last accessed 2021-11-04)
- [13] Ibid. (last accessed 2021-11-04)

Chapter 10: Managing Creation and Absorption



INTRODUCTION

Creation and absorption are performed through **Plan-Do-Check-Act cadences** (see [Chapter 8](#)), adapted to the [evolutionary cycle](#) used and the specific context of the project.

In this chapter we present concepts and practices applicable to the creation and absorption process in all lean projects.

VISUAL MANAGEMENT

Visual Management is the practice of using visual tools to manage work. It improves the efficiency and effectiveness of project management through better communication, collaboration and decision-making.

Typical visual tools are card wall, visual schedule, visual project timeline, task/kanban/Scrum board, workflow chart, project dashboard, A3 report, cumulative flow diagram and other workflow analytics visuals, burn down or burn up chart, velocity chart, value stream map, resource calendar, etc. These tools provide readily available and visible information about the work, including workflow, dependencies, priorities, project status, team performance, trends, quality information, issues and impediments.

The proper use of visual tools has several benefits:

- Fosters communication, sharing and collaboration.
- Reduces the information overload. Visual aids synthesize, compress and focus information. They simplify complex concepts and facilitate the interpretation and understanding of data. The human brain processes visual information much faster than text. When we add graphic elements to a text, they help the text to be perceived better and more quickly.
- Helps build transparency and trust within the integrated project team.
- Improves decision making. An A3 report, for instance, can document multi-month work in both critical and distilled information, with many visuals that can be interpreted at a glance. Such a report may be presented and consumed and, based on its contents, a decision can be taken – all this in just a few minutes.
- Eliminates the need for reporting. Visual project status tools, which are updated in real time, abolish the need for periodic status reporting and significantly speed up the decision-making.
- Improves planning, coordination and control by visualizing status, work-in-progress, dependencies, priorities, and workflow blockers and bottlenecks.
- Creates a stimulating and fun work environment and positively influences team behavior and attitudes.
- Increases productivity, reduces errors and defects, and helps keep the focus on workflow.

DECENTRALIZED PROJECT MANAGEMENT

Decentralized project management is the practice of delegating non-strategic management functions to lower levels in the project organization, down to the level of individual team members. It doesn't replace centralized management, but complements it.

There are several reasons for decentralizing project management:

1. It fits well with the lean culture and fosters harmonization of interests, collaboration, personal initiative and shared responsibility.
2. Enables effective communication, fast feedback and quick response to problems and opportunities where they appear and by the people closest to them. This improves decision-making, throughput, quality and speed.
3. Creates multiple mini project managers, eliminates micro-management, and allows the leaders to focus on the strategic aspects of the project.
4. Utilizes the initiative and creativity of team members.
5. Makes everyone empowered and responsible for project success.

We are grateful to Don Reinertsen for formulating and explaining the principles of decentralization. He addresses the issues of balancing centralization and decentralization, maintaining alignment, and the factors and human aspects of decentralization. We summarize these principles below, but encourage readers to learn more from his book. [1]

1. Balancing Centralization and Decentralization

- Problems and opportunities that are perishable are best dealt with quickly, and to do this, there is a need for decentralized control. Problems that have significant economies of scale or are big and infrequent, are best serviced with centralized resources.
- When there is sufficient information, use triage to choose a centralized or decentralized approach to a problem. When information is lacking, attempt to resolve the problem with decentralized resources, but if this doesn't happen within a reasonable time, escalate the problem for centralized resolution.
- To improve efficiency of centralized resources, use them for normal, daily work as decentralized resources and quickly mobilize them, following a pre-prepared plan, to engage with a big problem.
- Base the choice to use inefficient decentralized resources on the economic outcome. The faster response time may outweigh the inefficient resource use.

2. Maintaining Alignment

- Achieving alignment without centralized control is a challenge. Overall alignment creates more value than local optimizations.
- Use project goal ("why") and minimum constraints ("what" and "how") to provide direction and maintain coordination.

- Define clear roles and boundaries for responsibilities, to improve communication and avoid gaps and overlap.
- Focus the entire team on a main effort that will drive project success and subordinate other activities. When conditions change, be able to shift the main effort easily and swiftly.
- The focus can be changed faster when there is a small team of skilled people, simpler and optimized feature set, resource reserve, and flexible product architecture. To respond rapidly to uncertainty, team members should coordinate their local initiatives through continuous peer-to-peer communication.
- To enable absorption of variations locally, a portion of resource reserves should be decentralized at various levels within the project.
- To rapidly reduce uncertainty, focus early efforts on high technical and market risks.

3. Factors of Decentralization

- Provide the information they need to people who are allowed to make decentralized decisions.
- To speed up decision-making, ensure that most decisions are decentralized and involve fewer people and fewer management layers.
- Whatever drives economics should be used to measure performance and to provide incentives. If response times drive economics, then measurements and incentives for the team should be aligned to response time, not to efficiency.
- The demand for internal resources may create conflicts between projects that require elevating decisions to a higher level (centralized management). As for external resources, price internal resources and use an internal market to balance the demand and supply for all projects and to enable decentralized decision-making.

4. Human Aspects of Decentralization

- Taking initiative and rapid response by team members is crucial, and is far better than a superior decision implemented late. The initiative should be encouraged and team members need to be given a chance to practice it.
- Real-time, face-to-face, voice communication enables fast feedback.
- Decentralized control requires trust in the team. Trust arises from the ability to anticipate the behavior of other team members, which is built through shared experience. Thus, trust can be built by maintaining continuity in the teams and by working in small batches which increases the number of learning cycles.

SET-BASED CONCURRENT DESIGN

The design of project products (deliverables) is crucial for project success, because once a deliverable is designed, the biggest part of its benefits and costs are fixed.

By the time a product design is completed, 80% of the product's life cycle cost has already been determined, while 60% of these costs are committed by the concept development phase [2]. To a large extent, the design also determines the benefits of the product.

Sequential (Waterfall) Design

The traditional sequential design process requires that each subsequent phase be carried out only after the completion of the previous one. The work is done in large batches, and all the information is transferred in a single batch to the next stage. This accumulates variability and creates queues that interrupt the workflow.

Here is how Don Reinertsen describes the implications of the phase gate process: [3]

"The work product being transferred from one phase to another is 100 percent of the work product of the previous phase. This is maximum theoretical batch size and will result in maximum theoretical cycle time. It also leads to maximum possible delays in interphase feedback."

In sequential design, team members work in isolation and there is an information, communication, and often a physical wall between them. The work product is handed over by throwing it over the wall, and when something needs to be fixed, it's thrown back over it.

There are several drawbacks to this approach:

- Design errors are detected late and are costly to correct. The options selected in the previous phases limit the choices in the subsequent phases.
- The sequence of phases and the frequent need for rework extends the cycle time.
- Stakeholder feedback is infrequent and untimely.
- Early coordination of various aspects of design is limited and there is a lack of focus on important aspects of the product's life cycle, such as customer value, manufacturability, serviceability, operability, and social and environmental sustainability.

The result is a wasteful process and limited control over the product's life cycle benefits and costs.

Centralized Design

Centralized design seeks to improve the process by having team members work simultaneously on different parts of the design. Their work is coordinated by a central authority (e.g., design integration manager), who also serves as a communication hub.

Team collaboration and stakeholder involvement are limited. The work is performed internally within the team, and the work product is handed over to the downstream process in a waterfall fashion.

Centralized design improves horizontal coordination, shortens cycle time and reduces the need for rework, but leaves the fundamental problems of sequential design unresolved.

Concurrent Design

Concurrent design takes its name from the simultaneous (concurrent) performance of tasks, but it's much more than that. It involves several important practices:

- **Integrated design teams.** The work is performed by integrated multifunctional teams that have joint responsibility for the outcome. All specialties, the customer, suppliers and other stakeholders work together from the earliest design stages. The wall is "removed" and handovers are proactively avoided.
- **Life cycle considerations.** All product life cycle aspects are considered from the early design steps to maximize life cycle benefits and minimize life cycle costs.
- **Parallelization of phases and tasks.** Design phases overlap and whenever possible, design tasks are performed simultaneously, which reduces cycle time and the time to market. Supported by collaborative teamwork ("removing the wall"), the parallel approach improves coordination, accelerates feedback, and reduces errors. When information from other design phases is lacking, it's temporarily replaced by assumptions.
- **Evolutionary process.** Concurrent design is of an evolutionary nature. The process is iterative, with small work batches, fast feedback and adjustment. The design evolves in successive iterations.

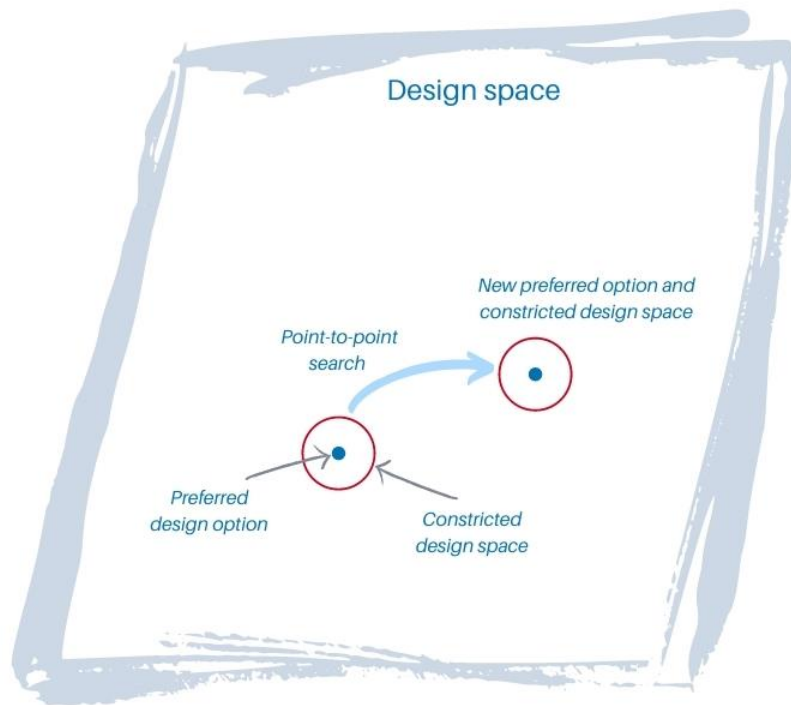
By overcoming the disadvantages of sequential design, concurrent design has the potential to shorten the time to market, improve quality, increase customer value, and reduce waste.

Point-Based and Set-Based Design

Each design alternative represents a specific point within a multidimensional design space, which is the totality of all design alternatives, defined and constrained by a set of design parameters.

Point-Based Design

The conventional point-based design (whether sequential or concurrent) develops one preferred alternative, which is improved or modified in successive iterations until a satisfactory solution is reached. The changes move the location of the alternative to a new preferred point in the design space, which provides justification for calling this approach "point to point". [4]

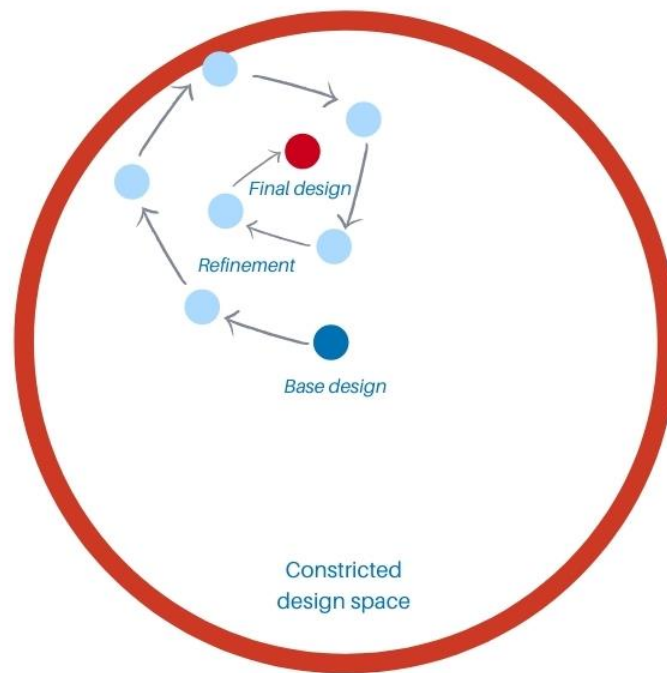
Figure 10.1 Point-Based Design**POINT-BASED DESIGN**

LeanPM® Framework

© Lean Project Management Foundation

Typically, the preferred alternative is selected upon completion of the concept development phase. As 60% of the life cycle costs have been determined by the end of this stage, the scope for further significant improvements is limited.

When an alternative is selected at the end of the conceptual phase, this is accompanied by narrowing the set of design parameters. The design space is reduced and subsequent refinements are made within narrow limits. Fixing the design parameters early results in very high costs if changes have to be made later in the design process. If the selected alternative proves unsuitable, it's replaced with another one, creating a new constricted design space.

Figure 10.2 Point-Based Design: Macro View**POINT-BASED DESIGN: MACRO VIEW**

LeanPM® Framework

© Lean Project Management Foundation

In theory, this approach can always lead to an optimal solution if enough iterations have been run, but this comes with a high cost for the negative iterations. In practice, the design team is often satisfied with the sub-optimal solution achieved after they've run out of budget and time.

The point-based design approach is suitable for low-risk, rapid incremental improvements of existing products or when the design space is necessarily very restricted, but not for radical break-throughs and typically not for new product development.

Once a base design is selected, it's refined through successive iterations until it fully meets objectives.

Set-Based Design

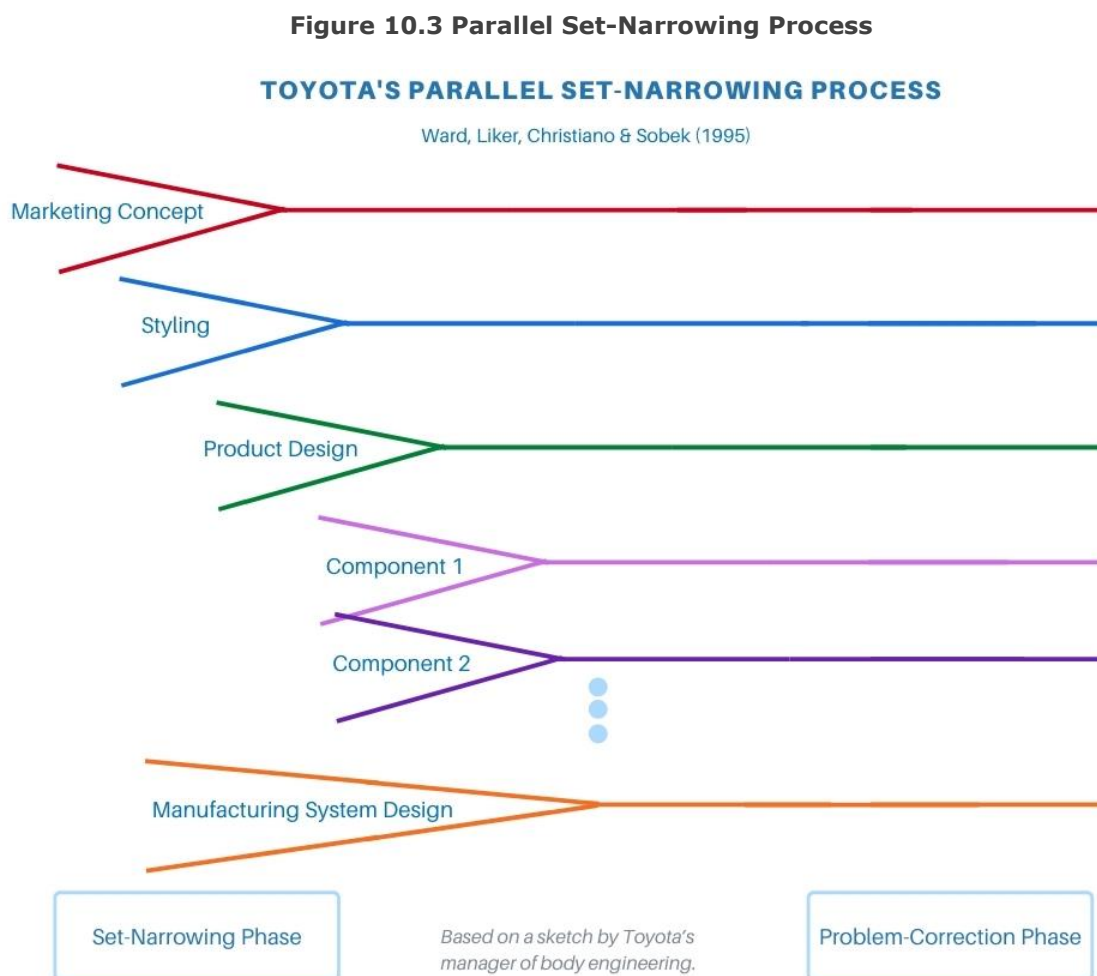
Set-based design (set-based engineering) is an approach in which multiple design alternatives are explored rather than a single one.

The process begins with the setting of broad design parameters, based on which multiple alternatives are defined, explored and progressively developed in increasing detail. This is a process of creating knowledge and reducing design-related uncertainty. The gained knowledge is used to eliminate infeasible and inferior alternatives, while the design parameters are narrowed. The process continues until there is only one alternative left, which is refined to complete the design. The probability of getting an optimal design outcome is greater than with a single option (point-based) design.

This approach avoids early commitment to a single option and delays the decision to choose an alternative until the most responsible moment, when the team has sufficient information and knowledge about the design. To reduce costs and increase process reliability, extensive prototyping and testing are used, as well as reusable knowledge (e.g., documented lessons learned from previous design experience).

Complex products that integrate subsystems and components require greater design flexibility. Instead of fixing the constraints for individual subsystems and components, each sub-team designs a set of options within broad parameters. Each specialty explores multiple alternatives and analyzes the tradeoffs from their own perspective. All design sets are narrowed down in parallel. The team looks for intersections of the sets and gradually converges them into a single solution, which is then optimized.

The parallel set-narrowing process is illustrated in the following figure, based on a sketch by Toyota's manager of body engineering: [5]



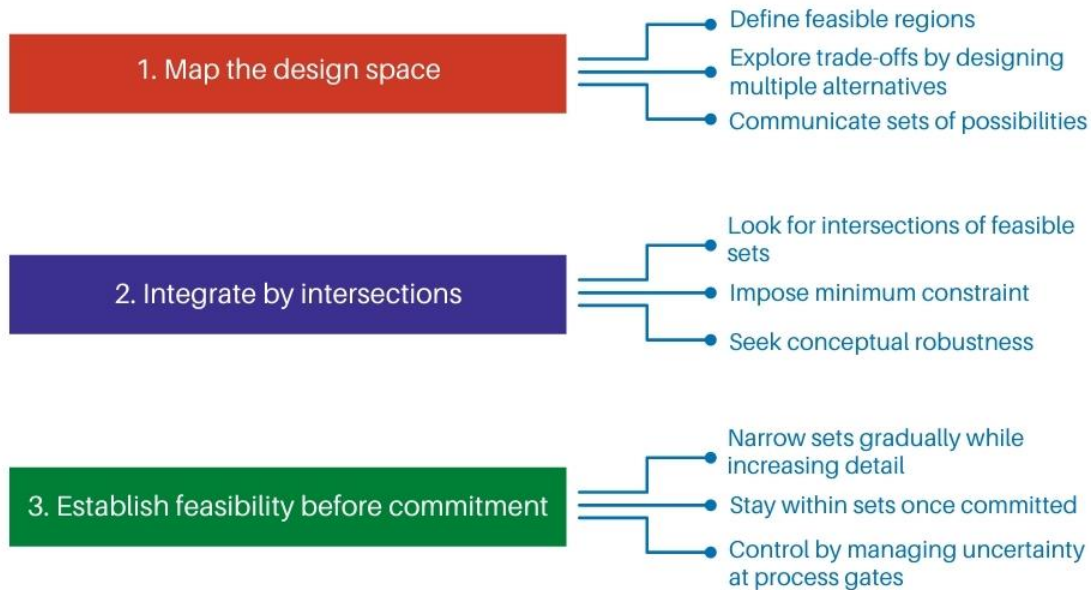
The approach that blends the concurrent and the set-based design in Set-Based Concurrent Engineering (SBCE) is known as the Second Toyota Paradox. Toyota's design process seems inefficient, as they delay design decisions, give suppliers partial information (communicate “ambiguously”), and explore a large number of prototypes. However, they not only design better cars, but do it faster and cheaper, because well-informed decisions offset the additional cost. [6]

The SBCE framework, developed by Sobek, Ward and Liker, is the one which is most widely used in practice. It comprises three general principles and nine implementation principles, and the steps are based on Toyota's best practices. [7]

Figure 10.4: Principles of Set-Based Concurrent Engineering

Principles of Set-Based Concurrent Engineering

Sobek, Ward & Liker (1999)



LeanPM® Framework

© Lean Project Management Foundation

The choice between point-based and set-based design is a matter of economic optimization. Exploring each additional design option reduces the risk (and cost) of failure and adds development cost – so there is a need to consider the trade-off between risk reduction and cost. The optimal number of parallel design alternatives (N) occurs when the incremental benefit of the Nth alternative equals its incremental cost, and this number can equal one. [8]

Or to put it more precisely, if the design option N+1 is the first to add a negative incremental net value and all subsequent options also have negative contributions, then the optimal number of design options is N. This is the number that maximizes the net value of exploring a certain number of alternatives.

BUILT-IN QUALITY

What is Built-In Quality?

Built-in quality is a practice of getting the right quality “the first time”.

Inspection is a reactive approach. It doesn’t change quality but just registers it after the fact. Thus, the proactive approach to quality management requires building in quality from the start, as it cannot be added on if it’s not already there.

Figure 10.5 Proactive vs. Reactive Quality Control



Here, we use the term “built-in quality” to describe the overall approach of lean project management used to achieve quality.

Quality is the totality of features and characteristics of project deliverables that generate value for the customer. The objectives of quality management, which is a subset of value generation management, are:

1. To satisfy the customer
2. To maximize the contribution of quality-induced life cycle benefits and costs to the project’s net value

These two goals must be in harmony, but this isn’t always the case. For instance, while a customer can be satisfied with the functionality of a software product with any technical quality (which is invisible to them), bad technical quality can reduce the customer’s life cycle benefit and increase their life cycle cost.

That’s why, when assessing quality, we must consider both external characteristics (usefulness) and internal characteristics such as manufacturability, maintainability, complexity/simplicity, flexibility, reliability, changeability, extendibility, and reusability.

Thus, we can define **good quality** as this set of external and internal features and characteristics that satisfies the customer and whose contribution to the project's net value is maximized. **Poor quality** is defined as either something that doesn't satisfy the customer or when a contribution to the project's net value isn't maximized, i.e., a set of features and characteristics that generates absolute or relative waste.

The lean project management approach to quality aims to close the five gaps in quality management:

1. Between customer satisfaction and economic efficiency
2. Between customer's actual and perceived needs
3. Between customer's actual needs and the needs perceived by the project team
4. Between the customer needs perceived by the project team and the actual design
5. Between the design and the actual deliverables

Foundation and Pillars of Built-In Quality

The foundation of built-in quality is the lean culture and its two pillars are economic efficiency and proactive quality management, while flow management enables these pillars.

Lean culture is a prerequisite for effective achievement of the two objectives of quality management, through its people-centered mindset and the focus on value creation, waste elimination, systems thinking, and continuous improvement.

The economic efficiency principle requires that all decisions regarding quality account for their long-term benefits and costs, with the goal of maximizing the contribution of quality to the project's net value.

Flow management reduces waste and enables proactive quality management by the use of pull, small batches of work, work-in-progress limits, fast feedback, workflow visualization, and queue control.

Figure 10.6 The Lean Project Management Approach to Quality**THE LEAN PROJECT MANAGEMENT APPROACH TO QUALITY**

LeanPM® Framework

© Lean Project Management Foundation

Proactive Quality Management

Proactive quality management involves two approaches: **Quality by Design** and **Lean Quality Control**.

Quality by Design

The Quality by Design approach designs quality by:

- **Defining customer value.** The customer and the project team must work together to achieve a complete understanding of the customer's actual needs.
- **Collaboratively designing project deliverables** that will generate value for the customer. The team performs lean concurrent design to explore options and analyze trade-offs, while focusing on life cycle aspects, including suitability for creation (an equivalent of manufacturability). They use set-based concurrent design and extensive experimentation, simulation, prototyping and testing, as appropriate. The stakeholders negotiate and agree on the requirements that best satisfy the quality objectives. The final design will produce quality requirements that should be achieved.
- **Designing a deliverable creation process** that is most likely to achieve the quality requirements, including quality generating components: people capabilities, work methods and

workflow steps, tools and equipment, and materials. Thus, the team defines a work standard that, if respected, is likely to deliver first time quality.

Once the customer value, the design (and the requirements), and the work standard are established, the team performs the workflow with lean quality control. Both Quality by Design and Lean Quality Control are powered by checking, learning, acting and adapting, to achieve continuous improvement.

Lean Quality Control

Lean Quality Control involves three practices:

- Front Loaded Testing
- Test-First Development
- Managing Quality at the Source

Front Loaded Testing

Front Loaded Testing is a proactive approach to quality control, which involves multiple-level testing with many small and fast tests as early as possible in the creation process. In a hierarchy of, for example, unit, component, sub-system and system tests, the focus is placed on testing the lower-level items at the expense of the costly and slow higher-level tests.

This approach prevents defects from entering the system and avoids expensive and time-consuming rework.

Test-First Development

This is an approach to create tests before creating the product or any of its components. Then, the team aims to make the tests pass by creating components that fulfil test requirements. Thus, creation is proactively informed and guided by the tests. The test-first approach encourages creating and testing in small batches and a simple design which is barely sufficient to pass the test.

The test-first practice must promote efficiency. The test should be minimized only to check the desired characteristics, and the created components should be minimized in scope only to pass the test.

Behavior-Driven Development is an evolution of the original test-first practice. It creates a shared understanding of requirements between stakeholders around scenarios of component behavior from a user's perspective. The desired behavior must be met and serves as a test for the created component.

Special languages can describe the desired behavior in a way that is both understandable to stakeholders and provides a structure that makes the description an executable specification.

For instance, the *Gherkin language* uses structuring keywords at the start of each line of the specification, followed by a descriptive non-technical text. The descriptions of steps start with Given, When, Then, And, or But. [9]

A behavior-driven specification of steps looks like this:

Given I am making a smoothie

And I have added almond milk

And I have added strawberries

And I have added yogurt

When I check my recipe

Then I need to add nothing

Managing Quality at the Source

Quality at the source requires checking the quality at every workflow step, and detecting and solving nonconformities at the source. It works as follows:

1. Define quality requirements
2. Define work standard to meet the requirements
3. Detect and visualize nonconformities as early as possible
4. Resolve nonconformities as soon as possible
5. Identify and eliminate the root cause to prevent a reoccurrence of the problem

Quality at the source is achieved by:

- **Standardized work** – documented people capabilities, work methods, workflow steps, equipment, tools and materials that will reduce process variability and will produce a controlled output.
- **Workflow visibility and signaling** about problems so that the team can quickly engage with them.
- **Performer and process customer checks.** Quality should be an individual responsibility of every team member and shared responsibility of the team. The performers should check their own work before a work item moves to the downstream step. No issues should be passed on. The process customers should check the inputs from the upstream processes. These checks will reduce the need for testing, inspection and rework.
- **Stop and fix it mentality.** As soon as a quality problem that cannot be fixed quickly is identified, the team stops working and directs all efforts to solving the problem. As the project incurs costs while the work is stopped, the team should be motivated to eliminate the root cause as quickly as possible. The focus is on the system of early problem detection and reporting, and quick problem solving, rather than on the individual mistakes and accountability of team members. Stop and fix it is a culture where problems are not hidden and don't wait on the side to be solved. They get fixed quickly and their impact on the workflow is minimized.

LEAN TEAMWORK

Lean-Agile Teams

At the most fundamental level, a team is a group of people who depend on each other to do a common job, but lean projects need specific lean teamwork practices.

Lean projects directly serve value streams and, although they are temporary, often they are regular. Thus, value streams create around themselves a dynamic network of temporary value streams (projects). The network of projects becomes part of the value stream and lean project teams can be viewed as subsets of the value stream team.

Effective value stream teams break down organizational silos. They are cross-functional, dedicated and long-lived and are constantly developing their ability to create value. Therefore, as a part of the value stream team, effective project teams should also be cross-functional, dedicated to a single project, long-lived and constantly improving.

When the customer is external, the project would benefit from stable performer-and-customer value stream teams, which need to be integrated into a single team. This is a model where work is brought to people rather than people being brought to work.

Lean teams are cross-functional. They have all the skills, competencies and diversity needed to perform the work efficiently and effectively, and create value.

It's preferable for team members to have T-shaped skills – deep expertise in at least one domain and broad expertise in other domains. While they excel in their core tasks, they should be able to collaborate with experts from other domains and perform a broader range of tasks effectively.

Integrated teams

"Working together always works. It always works. Everybody has to be on the team. They have to be interdependent with one another."

Alan Mulally, CEO of Ford

Traditionally, the project team involves the people who carry out the activities of transforming inputs into outputs and the people who manage the project. All others who may be affected or may affect the project, including the customer and the users, are considered stakeholders outside the project team. This model disrupts communication, collaboration and interaction between stakeholders.

For instance, when users are seen as external to the product development team, developers may not have direct but only indirect contact with them through the product owner. This is a recipe for huge ineffectiveness.

Lean project teams are **integrated**, meaning that they integrate all stakeholders, including customers, users, partners and suppliers. The communication between the people who create project assets and the customer and users is direct and intensive. Team building, development, and performance improvement activities are performed by the integrated team.

Stakeholders should be appropriately involved in regular team activities and decision-making. Those stakeholders who for any reason don't take part in specific team activities should be represented on the team by their avatars (stakeholder avatars).

The integrated team is especially important for ensuring that project deliverables are absorbed by the customer value stream. When the team is integrated, the absorption is done not only within the project but also by the project team, instead of being done outside the project and by an external team.

Self-Organizing Teams

Decentralized project management needs self-organizing teams that have the autonomy to decide for themselves *what to do* and *how*, to realize the reason (*why*) for the project.

"Individuals and Interactions over Processes and Tools"
Agile Manifesto

People work best when they can choose what processes and tools to use, rather than using the ones that management have chosen or, worse, have imposed, because they think it's the best. It's more important to have the right people who interact effectively, united by shared objectives, and the way they work should result from their interaction, not be a pre-condition for it.

In a self-organizing team, the tasks are not assigned by the management, but by the team members pulling the work and deciding internally who does what, when and how. They don't work in isolation, but have an overview of the work of the entire team and frequently communicate and coordinate with others.

A well-functioning self-organizing team creates team consciousness and intelligence that is greater than the sum of the individuals. Team intelligence can best adapt, experiment, solve problems, learn and evolve.

Allowing and encouraging the team to self-organize is a powerful way to get them involved and take ownership and responsibility for their work.

"Without involvement, there is no commitment. Mark it down, asterisk it, circle it, underline it. No involvement, no commitment."
Stephen R. Covey, "The 7 Habits of Highly Effective People"

Lean-Agile Leadership

With decentralization, the role of management changes. While focusing on strategic decisions, collaborative and servant leaders do not command and control but create a supportive environment and empower the team to take responsibility for their day-to-day work.

Leaders provide the team with training, coaching and mentoring to gain self-organizing skills and with resources to perform the work. They create an environment of psychological safety, free of fear and uncertainty. Mistakes are tolerated and seen as a means of learning and improvement.

Team members are not judged by their individual performance, which leads to information hiding and creates uncertainty and competition. Instead, leaders encourage teamwork and evaluate and recognize overall team performance.

Instead of rigid control that impedes creativity and spontaneity, the key is for management to employ **subtle control** through: [10]

- Carefully selecting team members to achieve a balance and personality fit
- Creating open work environment
- Encouraging communication with the customers to better understand their needs
- Evaluating and rewarding based on team performance
- Anticipating and tolerating mistakes, but expecting their early detection and quick resolution
- Encouraging self-organizing behavior by project partners

Transparency

"A lack of transparency results in distrust and a deep sense of insecurity."

Dalai Lama

While visibility makes information easily accessible and easy to use, transparency makes the necessary information available.

Transparency means that stakeholders communicate openly with each other and be explicit about their needs, expectations, concerns, problems, mistakes, progress, etc. They share the relevant information at their disposal.

Transparency provides information about the actions and decisions of stakeholders, and the motives behind them, and thus creates predictability. This helps build trust.

When information is shared openly and regularly, the implicit is turned into explicit and the assumptions into facts. Errors, problems and opportunities are detected earlier and feedback is faster and more effective. Expectations are clearer and more likely to be met.

Creating an environment of psychological safety without fear and judgment is the most important condition for achieving transparency. Everyone should be able to express their opinions and concerns freely and be open about their desires, problems, mistakes and failures. Team members should be trained, mentored, and encouraged to be transparent.

Transparency tools like information radiators and daily stand-up meetings make it work in practice.

FLOW MANAGEMENT

The Goal of Flow Management

Flow is the movement of work items through the creation and absorption process in a steady, continuous stream. (The properties of the ideal workflow, p. 172.)

Flow management seeks to minimize the following negative economic consequences. They should be considered in the context of the overall economic framework that is addressing the cumulative effect of the impact of all project variables: [11]

- Formation of process queues of waiting work, which lead to accumulations of too much work-in-progress (WIP). This increases variability, risk, and cycle time and lowers efficiency, quality, and team motivation.
- The ineffectiveness of trying to control timelines directly instead of controlling queue size, which provides control over timelines.
- The economic consequences of high levels of capacity utilization (efficiency) which affect the cost of time.
- The negative economic impact of variability.
- The effects of large batch sizes of work: increased uncertainty and cycle time and reduced speed of feedback.

The goal of workflow management is to contribute to the maximization of the project's net life-cycle benefit.

The three major ways to influence the flow are:

- Controlling WIP
- Managing queues
- Controlling batch sizes

Controlling WIP

WIP, Little's Law and Estimating

The amount of WIP directly affects the team's throughput and the cycle times.

Chapter 5 showed the application of the Little's law to the project portfolio system. It can also be applied to a project and to the individual process stages within a project.

Little's law reveals an important relationship between the WIP, the throughput and the cycle time:

$$\text{Average Cycle Time} = \text{Average WIP} / \text{Throughput}$$

where

Cycle Time is the time it takes a work item to pass through a work process. This is the time a work item is work-in-process/work-in-progress.

Work-in-Progress (WIP) is the inventory of work (measured in work items or amount of work) in a work process.

Throughput is the average output of a process per unit time.

Note that Little's Law "cannot always be treated as if any pair of variables selected from throughput, WIP and cycle time can be independently altered to set the third variable to a desired target" [12].

The law can be applied directly to projects that comprise independent work items which run end-to-end through the same workflow, such as user stories that represent incremental values for the customer.

The overall cycle time can be assessed for a project with a certain level of total WIP (including that part which is in queue) using historical information about the team's throughput in similar projects with the same workflow process. The total WIP includes all work items in the project.

The relationship is:

$$\text{Total Cycle Time} = \text{Total WIP} / \text{Throughput}$$

The forecast completion time for a project with 192 items and a throughput of 48 items per week is 4 weeks.

Suppose historical data about the throughput are not yet available, but the initial data show that the average cycle time for the team is 0.5 weeks per item. There are 16 people on the team and the optimal level of WIP has been assessed at 1.5 items per person, including the items in queue. Thus, the average WIP is 24 items (16×1.5).

Now the team's throughput can be forecast as:

$$\text{Throughput} = \text{Average WIP} / \text{Average Cycle Time} = 24 / 0.5 = 48 \text{ items per week}$$

But let's say there is a need to complete the project in 3 weeks. The total number of work items remains the same, and it is necessary to increase the team's throughput to 64 items per week. How many people are needed on the team?

Using the formula above, 32 items of WIP (64×0.5) are needed on average and according to the policy of 1.5 items of WIP per person, 21 people are needed on the team. The assumption is that the enlarged team will have the same average cycle time.

Note that the examples above will also be valid if the work is measured in story points or ideal hours.

These are simplified examples. To improve the quality of estimates, statistical forecasting can be used.

Statistical Estimation

The table below shows a project with 20 work items that took 26 working days to complete. All work items go through the same workflow (e.g., Preparation, Development and Acceptance) and they can be anything – from software features and user stories to handmade jewelry and 3D printed models. The throughput can be easily calculated by dividing the number of work items by the number of days: $20/26 = 0.77$ items per day.

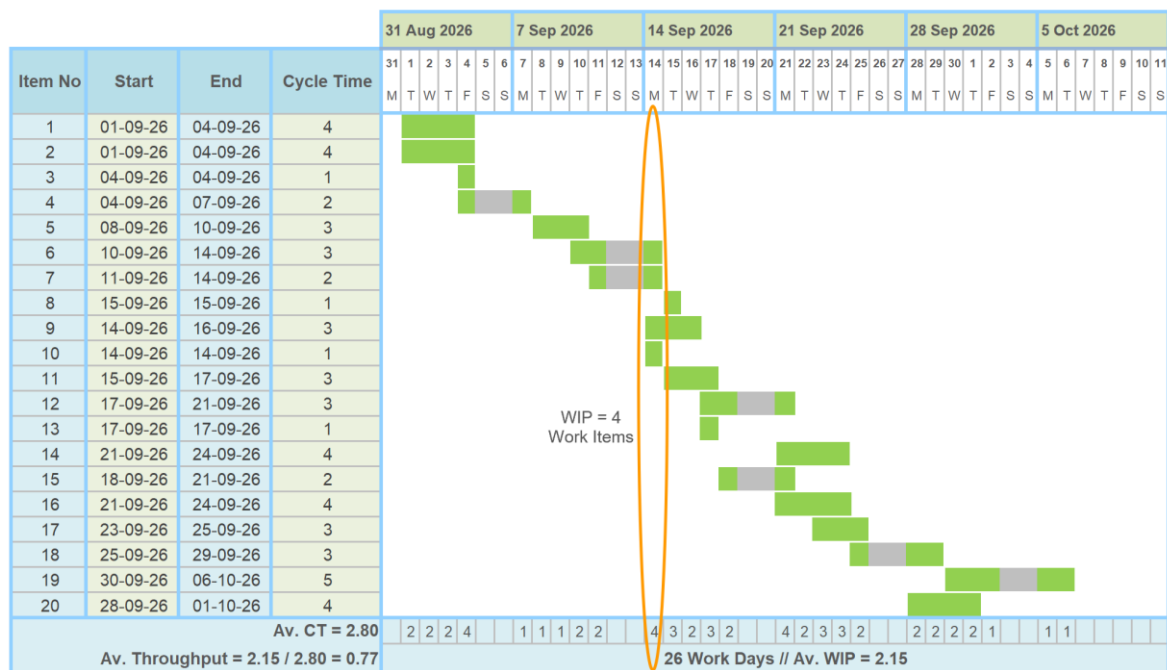
But perhaps it is necessary to check whether Little's law applies to this project. To do this, the average cycle time (2.80 days) and the average WIP (2.15 items) must be calculated. Dividing 2.15 by 2.80, results in a throughput of 0.77 – just as much as calculated above.

This proof provides peace of mind. However, is it possible to complete any other similar project with 20 work items in 26 days? Not really. In fact, the statistical probability of completing another project in 26 days or fewer is about 50%.

The timing of a future project cannot be calculated, but we can get a probability distribution that will improve the quality of our forecast.

For this purpose, historical data from a similar project with the same context are needed: a stable team and work process, and consistent work items policy (type, definition, approximate size and complexity). Let's assume that this is so in the current case.

In this example, the work items are of similar but not identical size and complexity. The cycle time varies from 1 to 5 days, and the WIP – from 1 to 4 items.



Of interest is the data pattern. It reflects variations in cycle time, WIP, and team productivity. This pattern also registers the consequences of the obstacles and problems encountered during the project. Errors, defects, reiterations and rework may have shaped the data, too.

Monte Carlo simulation is a tool that can decipher the pattern of historical data and calculate the probabilities of occurrence of the possible outcomes for future projects. It uses a computational algorithm fed by data from a repeated random sampling (random sampling – imagine playing roulette that generates random numbers at a casino in Monte Carlo).

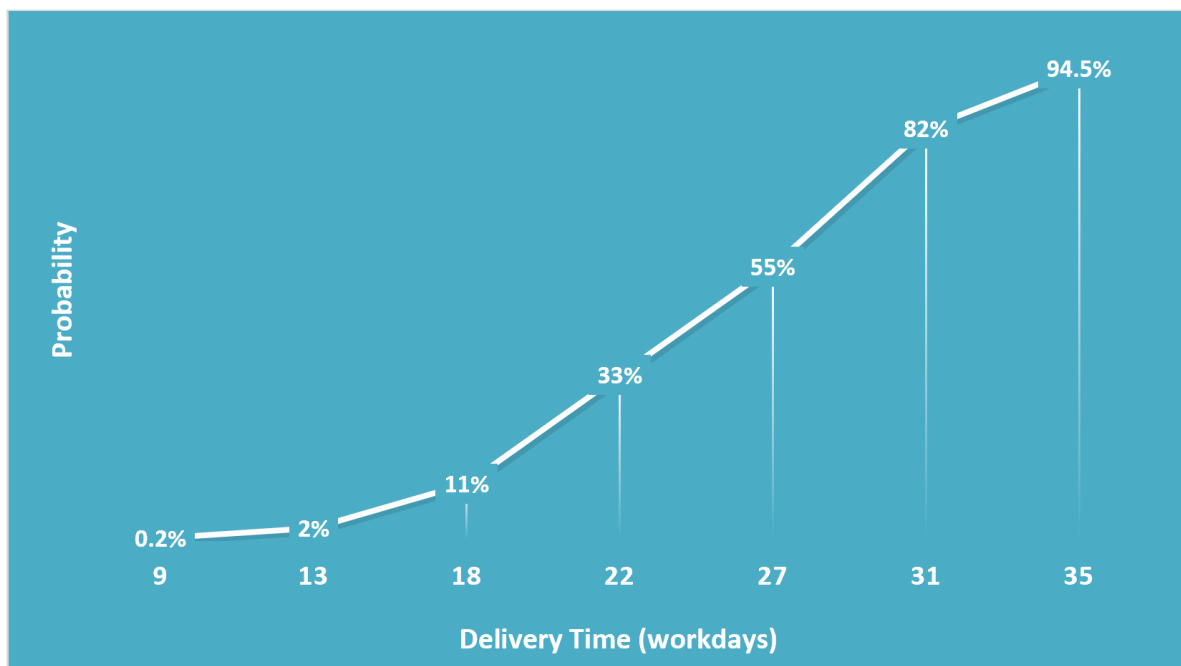
The data in the example are not enough to win the roulette game, but they suffice to forecast the delivery time.

So, let's go back to the project game and turn the roulette 1000 times (repeat the random data sampling 1000 times), and voila! Here's what we get:

Delivery Time for 20 work items (workdays)	Avg. Throughput (items/workday)	Probability Density Function (PDF)	Cumulative Distribution Function (CDF)
9	2.22	0.1%	0.1%
13	1.49	1.5%	1.6%
18	1.12	7.4%	9%
22	0.90	24.7%	34%
27	0.75	21.4%	55%
31	0.65	26.7%	82%
35	0.56	12.7%	95%

Probability density function (PDF) defines the probability for a discrete variable. The probability of completing a project with 20 work items in 13 or 31 days is 1.5% and 27%, respectively.

Cumulative distribution function (CDF) shows the probability that the project will be completed in a certain number of days or less. For instance, the probability of completing a project with 20 work items in up to 27 days is 55% – this is the sum of the discrete probabilities for 27 days and each individual case over a smaller number of days. The corresponding probability for 32 days is 85%.



If we feel comfortable enough with an 85% certainty, we can assume that we will complete the next 20-items project in no more than 32 days.

With the Monte Carlo simulation, we can forecast the time to complete a project with any number of work items. There is an 85% probability that we will complete a project with 33 or 55 items in no more than 54 or 91 days.

As a shortcut to quickly forecast time with 85% certainty, in this specific case a throughput of 0.6 (check 20/32, 33/54 and 55/91) can be used. In fact, what the Monte Carlo algorithm extracts from the historical data is the **probability distribution of the throughput**.

We already have a time estimate. What about the cost?

Suppose a team of three works full-time on their projects (including the project for which there are historical data). They cost \$1,000 a day. In addition, the indirect costs are \$100 a day and the average material cost for each work item is \$50.

The estimated cost for a project with 33 items will be \$61,050 with 85% certainty. Here, the costs are almost entirely a function of the working days, so the probability distribution of the estimated costs will mirror that of the estimated time.

Playing roulette using historical data gives much more reliable estimates than playing poker using planning poker or other similar estimating techniques.

However, how can Little's law be applied to assess the total cycle time of a project that does not consist of independent work items which run end-to-end through the workflow? This is the case with many types of projects. Work items cannot be used to measure the work and the throughput. Other units of work need to be used to measure the effort, like hours.

A project with a total effort (total WIP) of 1200 hours and a throughput of 80 hours per week will have a total cycle time of 15 weeks.

This approach will only work as an approximation and if there is historical information about the same team, and for similar projects with (ideally) identical workflow profiles in terms of the number and type of process stages, distribution of the efforts between these stages and between the tasks which are on and off the critical path, etc.

Alternatively, the formula can be applied to assess the cycle time of each process stage, and then determine the total project cycle time while focusing on the critical path activities that affect the cycle time.

Optimal WIP

Contrary to popular belief, minimizing WIP doesn't minimize cycle times and project cycle time, as WIP has the important function of buffering variability. Instead of just restricting WIP, we actually need to find the optimum level and set an upper and lower limit.

The WIP level determines three zones of system performance: [13]

- **High WIP level – overload zone.** Excessive WIP increases cycle times but does not improve the throughput. WIP has to be reduced to improve system performance.
- **Low WIP level – starvation zone.** When WIP is reduced to a minimum, process cycle times are the shortest possible. But there is no buffer in the system and it's exposed to the negative effect of variability. Throughput can drop dramatically due to process starvation, which affects the overall project cycle time.
- **Optimal WIP level – optimal zone.** This is the WIP level that ensures best system performance with maximum throughput and minimal cycle time.

The optimal WIP at all process stages helps to achieve a smooth workflow and to reduce unevenness (*Mura*) which in turn reduces waste.

"At low WIP levels (below the Critical WIP), increasing WIP greatly increases throughput while cycle times are changed little. But at high WIP levels, the reverse is true: cycle times increase linearly with increasing WIP while throughput is almost not affected." [14]

- Mark L. Spearman and H.J. James Choo

Multitasking

Excessive WIP not only affects cycle time, quality and the feedback speed but reduces productivity and throughput because of multitasking.

When we focus on a task, we gather relevant contextual information, which is necessary for its effective performance. Initially, this information is stored in our short-term memory, but when we focus long enough, it moves into our long-term memory and we develop context awareness about the task.

However, when we shift our focus to another task before finishing the previous one, we switch the context. We empty our short-term memory to free capacity for the new context, and when we switch back to the previous task, we have to recover the lost information. Each switch or interruption is associated with a cognitive loss and wastes time and energy. Multitasking lowers the quality of work and provokes stress.

The task-switching penalty includes: [15]

- Wasted time for physically performing the switch
- Rework of untimely aborted work
- Time to restore the context
- Frustration cost
- Loss of team binding effect

Therefore, the optimal WIP level is the one that produces the best balance between cost-of-time savings (reduction of cycle time) and the benefit of better productivity (minimized multitasking), and the cost of capacity underutilization (resulting from variability and starvation).

Limiting WIP

WIP limits are a pull-based tool for matching the work with a team's capacity. The constrained WIP helps to focus a team's collaborative efforts, not just on doing work but on fast completion of specific tasks.

If we set only an upper WIP limit, it would mean that we are satisfied with any lower level, even zero. Of course, zero WIP means zero throughput and zero capacity utilization. Therefore, we need to set both the maximum and minimum WIP limits to frame its optimal level.

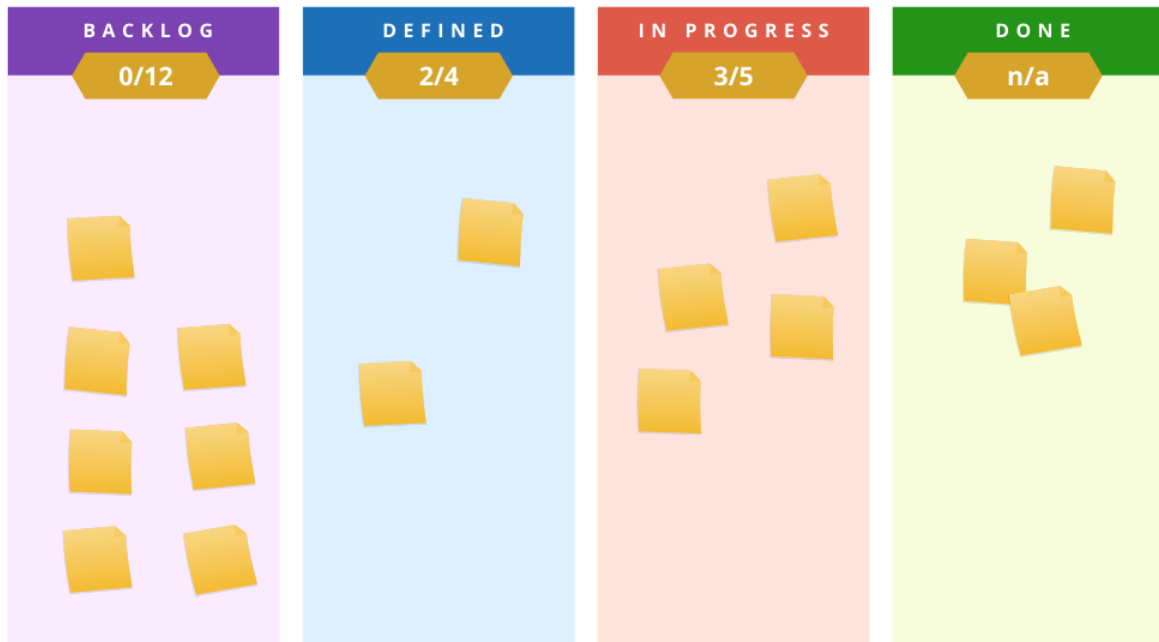
Placing a WIP constraint at each process stage limits the total WIP in the system. This can be done using a kanban system. When the maximum limit for a certain stage is reached, it stops taking work from the upstream process and the restraining signal gradually propagates upwards. Once the process stage frees

up capacity to pull new work, the demand signal propagates upwards and the smooth workflow is resumed.

Intermediate WIP buffers help to synchronize the flow between processes that use batches of work of different sizes.

Figure 10.7: Kanban Board with Minimum and Maximum WIP Limit

KANBAN BOARD WITH MINIMUM AND MAXIMUM WIP LIMIT



LeanPM® Framework

© Lean Project Management Foundation

Little's Law Assumptions

The original Little's Law deals with queueing systems which consist of discrete objects that enter a system at some rate, spend some time in queues and in service and after being serviced, leave the system.

The Law says that "under steady state conditions, the average number of items in a queueing system equals the average rate at which items arrive, multiplied by the average time that an item spends in the system".

Little's Law will hold if two assumptions or conditions are satisfied: [16]

- **Boundary condition:** a finite time window to start and end with an empty system.
- **Conservation of customers** – all customers that enter the system will be serviced and will exit the system. No customers are lost and the number of arrivals equals the number of departures.

When the law is stated in terms of the average output (as with an operating or project system), rather than the arrival rate, and the system isn't empty at the beginning and end of the time window, then it also applies, at least as an approximation, if the following conditions are met: [17]

- **Conservation of flow** – all work items that enter the system will be processed and will exit the system. No work items are lost, and the number of arrivals equals the number of departures.
- **The size of the WIP is roughly the same** at the beginning and end of the time interval. There must be neither significant growth nor decline in the WIP's size.
- **The average age or latency of the WIP is stable.** When the WIP never drops to zero, the jobs shouldn't be getting older or younger. Aging work units accumulate passive WIP that doesn't leave the system, while the law overstates the actual time in system for those units that have left it.

The above conditions have practical implications. On the one hand, they are needed so that we can have reliable flow metrics. On the other hand, when the WIP size is optimal, they facilitate smooth flow and sustainable throughput. Therefore, these conditions should advise the process policies.

Little's Law conditions reinforce the economic framework. For instance, there is another reason to actively control the aging work items which are not only a symptom of problems and waste but also distort the flow metrics.

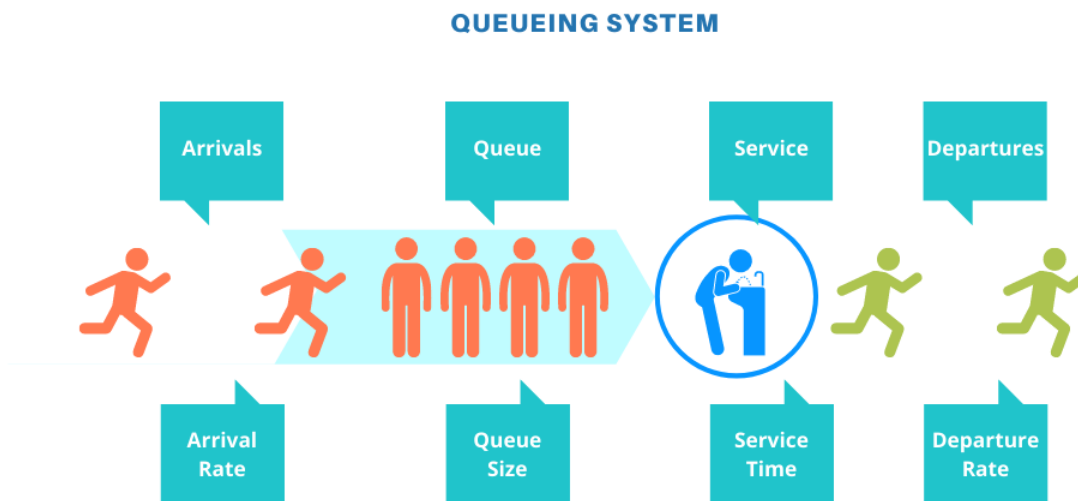
Managing Queues

Queueing Systems

A **queue** is a sequence of entities (in project management – work items or jobs) awaiting their turn to be serviced.

A queueing system consists of jobs, the job arrival process, queue or waiting line, servers and service process, and the departure process.

Projects are queueing systems whose queues can accumulate a WIP inventory and affect cycle time. Therefore, it's important to understand the fundamental properties of queueing systems, the factors influencing queues, and how they can be managed.

Figure 10.8: Queueing System

LeanPM® Framework

© Lean Project Management Foundation

If a job arrives in the system every 20 minutes, and the service time for each job is exactly 20 minutes, no queue will form and the queue time will be zero. The cycle time (queue time plus service time) will be equal to the service time. Server utilization will be 100%.

Suppose four jobs (A, B, C and D) are expected to arrive at 13:00, 13:20, 13:40 and 14:00, but job B is 10 minutes late and arrives at 13:30. The server will be idle for 10 minutes and its utilization will fall below 100%. There is no way to compensate for this, and utilization for a certain period can only go down. Job C will arrive on time, but will have to wait in the queue for 10 minutes while the server is busy.

If job D arrives 10 minutes earlier (at 13:50), this will restore the average inter-arrival time (20 minutes), but since job C will be completed at 14:10, D will be finished at 14:30. If the next job arrives on time (at 14:20) it will be in a queue for 10 minutes.

In another case, let's assume that the four jobs arrive on time, but the service time varies – it's 5 minutes less for job A and 5 minutes more for B. After completing job A, the server will remain idle for 5 minutes and its utilization will decrease, and after job B is serviced, a queue of 5 minutes will be formed.

The examples above show that variations in inter-arrival and service times can lead to queuing and reduced capacity utilization.

Queueing systems can be described by a three-factor **A/S/c** notation where **A** characterizes the arrival process – the distribution of inter-arrival times, **S** describes the service process – the distribution of service times, and **c** indicates the number of servers.

M/M/1 Queue

Queueing models can be very complicated. The basic model that can describe a project queueing system is the M/M/1 queue – Markov arrival process/Markov service process/Single server. In this model, the inter-arrival and service times are exponentially distributed and the jobs are serviced by a single server.

In an M/M/1 queue, jobs arrive one at a time and inter-arrival and service times are exponential random variables. The average inter-arrival time is known, but both the arrival and service process are stochastic (random) and memoryless. The distribution of inter-arrival and service times does not depend on the history of each process, but only on its present state.

The project team, which has shared responsibility for the work, can be considered as a single server. The server may be an infrastructure or machine, such as a 3D printer or test server.

Variability causes queues and queue sizes increase with capacity utilization of the server. The M/M/1 model can be used to describe the relationships between the average service time, arrival rate (jobs per unit time), capacity utilization, and waiting time: [18]

(1) Cycle time = Waiting time in queue + Service time

(2) System utilization or Utilization of the server (the fraction of time the server is busy) = Arrival rate/Service rate

or

(3) System utilization = Arrival rate * Average service time

(4) Service rate = 1/Average service time

(5) Probability of an empty system = 1–System utilization

(6) Average waiting time in queue = (System utilization/Probability of an empty system) * Average service time

For instance, if the arrival rate is 4 jobs per day and the service rate is 5 jobs per day, the utilization is 0.8 (80%). The service rate is the average number of jobs that can be served at 100% utilization. The server is idle when it's available to perform the service, but the job queue is empty. If work is just stopped, this will not reduce the utilization, because the capacity will not be available.

When service time is sustainable, the average cycle time depends only on the average waiting time (1) which in turn depends only on the system utilization (6). Therefore, it's important to find out more about the relationship between system utilization (team's capacity utilization) and the queues. Let's look at an example.

Suppose that the average time to service a job is one day. What will be the average waiting time in queue (AWT)?

Let's start with 20% utilization. We can expect that the arriving jobs will wait on average 0.25 days in queue (see formula 6 above).

AWT (20% utilization) = $(0.2/0.8)*1 = 0.25*1 = 0.25$ days

Let's now increase the utilization:

AWT (40% utilization) = $(0.4/0.6)*1 = 0.67$ days

AWT (50% utilization) = $(0.5/0.5)*1 = 1$ day

AWT (60% utilization) = $(0.6/0.4)*1 = 1.5$ days

AWT (67% utilization) = $(0.67/0.33)*1 = 2$ days

AWT (75% utilization) = $(0.75/0.25)*1 = 3$ days

AWT (80% utilization) = $(0.8/0.2)*1 = 4$ days

AWT (90% utilization) = $(0.9/0.1)*1 = 9$ days

AWT (95% utilization) = $(0.95/0.05)*1 = 19$ days

AWT (99% utilization) = $(0.99/0.01)*1 = 99$ days

AWT (100% utilization) = $(1/0)*1 = \infty$

The project work is variable, which interrupts the workflow. The work items are of varying complexity and size. Team productivity and completion times vary. Rework and reiterations cause even greater interruptions.

When there is slack in the system, the effect of these interruptions may not be dramatic, but as the slack decreases, the system clogs.

As capacity utilization increases, the waiting time gets disproportionately longer. At 50% utilization, the waiting time is equal to the service time, but at 75% and 80%, the waiting time is three and four times longer than the service time.

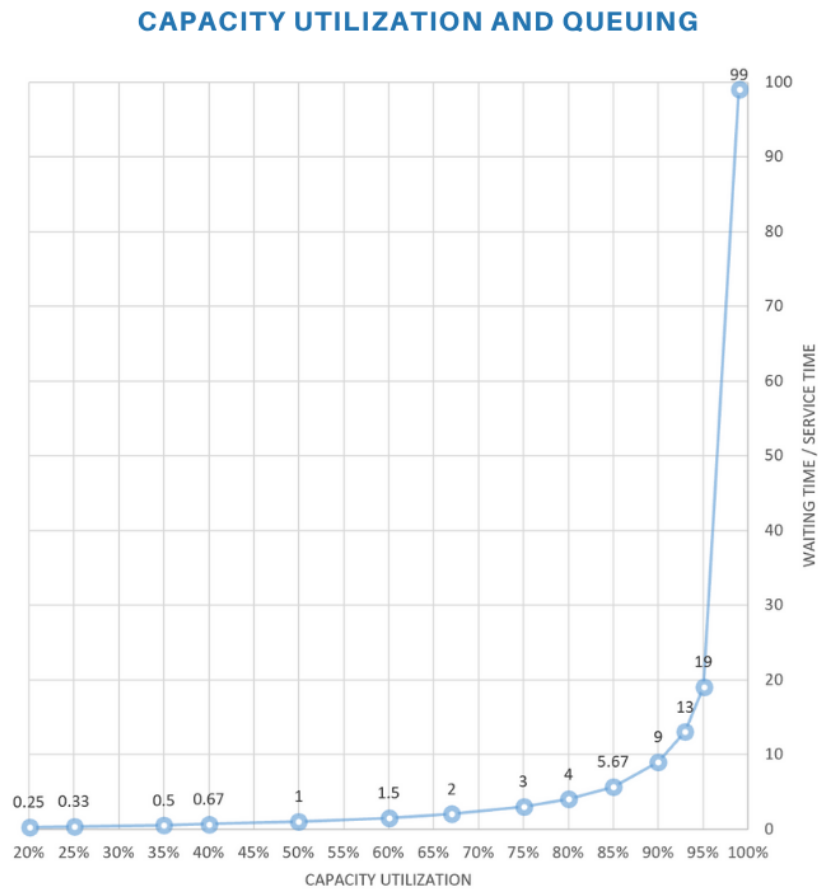
Comparing cycle time and service time, at 50% utilization, cycle time averages two times the service time and at 90% utilization, cycle time averages ten times the service time.

The queue grows dramatically when utilization exceeds 90% and becomes infinitely long at 100%.

Therefore, **matching team capacity and job demand requires planning for an appropriate capacity utilization, not 100% utilization.** There must be slack in the system to absorb variations.

The slack – the time when team members are not busy – is necessary for another important reason. It improves team effectiveness. Slack provides flexibility, reduces stress, improves quality, increases security and provides resources for change, innovation and learning. [19]

The optimum utilization should be based on the trade-off between the cost of queue, the cost of capacity (efficiency loss) and the effectiveness gains.

Figure 10.9: Capacity Utilization and Queuing

© Lean Project Management Foundation

It is important to remember that capacity utilization is a ratio of the rate of arrival of jobs to the service rate (and the service rate is calculated at 100% utilization). To reduce utilization and the queue size, it is necessary to decrease the arrival rate or increase the service rate (team capacity), or both.

The M/M/1 queue assumes that work items arrive one at a time. When multiple items arrive simultaneously in a given time instant, this is known as “bulk or batch arrival”. This is the situation in traditional project management, where jobs arrive in large batches that can be considered as aggregations of individual work items.

Bulk arrivals further increase variability and the waiting time.

Controlling Queues

Visualizing Queues

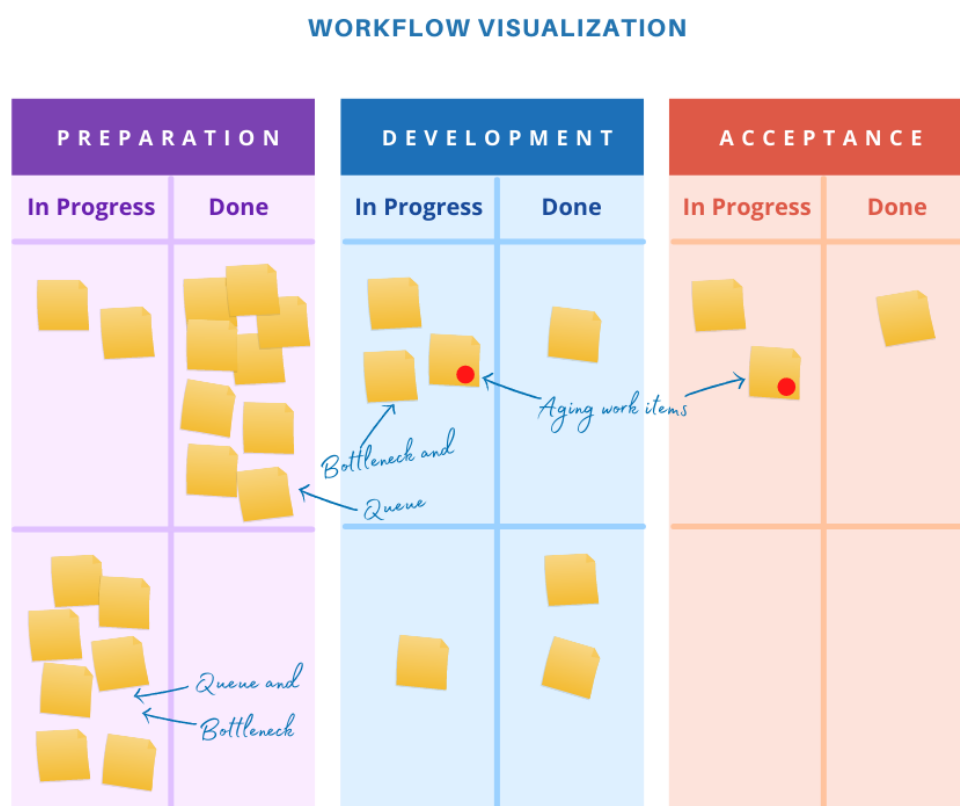
The project workflow usually involves a series of queues. To manage queues, it's important to visualize them. We can hardly manage what we cannot see and imagine. Visibility drives action for improvement.

In projects with physical deliverables, most queues and WIP are physical inventories that accumulate and signal problems. However, in many projects, including all knowledge work projects, queues and WIP become digital information which is stored on electronic media.

The kanban board is a great way to visualize the workflow, queues and WIP. The process steps are mapped onto the board and work items that should flow through the process are visually represented by kanban cards.

In the example below, the clustering of cards in the *Preparation: Done* column is a queue, and it indicates a problem with the work going through this stage of the process (problem with processing the work at the Development stage). The empty area in *Preparation: Done* shows that the work is blocked upstream and that the downstream part of the process may soon be starved.

Figure 10.10: Workflow Visualization



LeanPM® Framework

© Lean Project Management Foundation

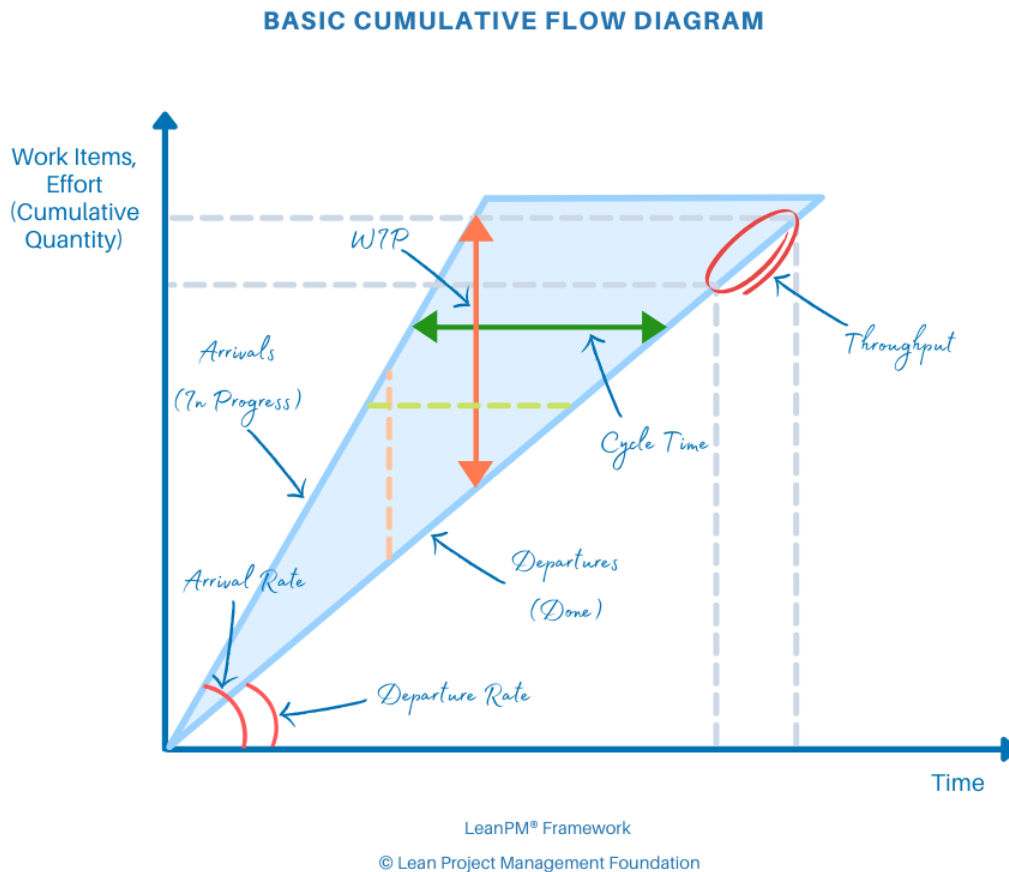
Within the project, the WIP (the work in the creation and absorption system) includes the work that is being worked on and the work that is waiting in a queue. The same applies to the WIP of the individual process stages. Technically, WIP that exceeds the optimal level should be considered a work in queue.

When the WIP isn't limited, invisible queues are formed. When the queues have been moved into separate process sub-stages, formally, all other work items in progress are being worked on. But in fact, when the work in progress is large, some of these items are being serviced, and the others are waiting in queue.

If we work on all the items at the same time, just to eliminate the queue artificially, the multitasking will only make things worse. By constraining the WIP, the queues become more visible.

Another popular tool for visualizing workflow and queues is the Cumulative Flow Diagram (CFD).

Figure 10.11: Basic Cumulative Flow Diagram



The CFD is a tool for visualizing the workflow. It presents project progress, the stability and trends of the flow, and signals problems.

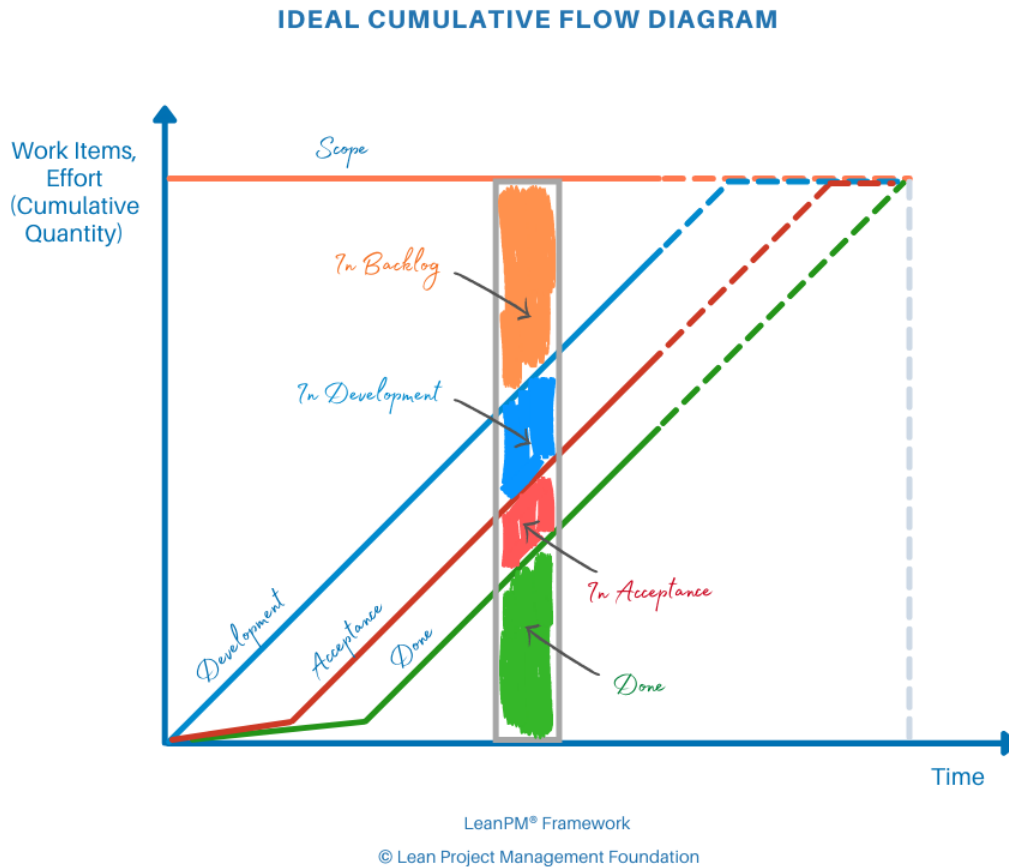
Figure 10.11 shows a basic CFD for a single stage workflow. The vertical axis shows the amount of **cumulative** work that has been started or completed at any given time. The horizontal axis is a time scale. The Arrivals and Departures lines represent the amount of work that has been started and completed. The work between the two lines has started, but has not yet been completed, i.e., it's a work in progress.

The vertical distance between the two lines shows the work in progress at the corresponding moment, and the horizontal distance indicates the time elapsed to move a unit of work from the beginning to the end of the process, i.e., the cycle time.

The slope of the lines shows the rate at which the work arrives (we take on new work) and leaves the process (we finish the work started). We can easily check if the arrival rate is approximately equal to that of the departure (more or less parallel lines), which would mean that the WIP and cycle time remain stable. The sections of the Done line show what the throughput is for a certain period.

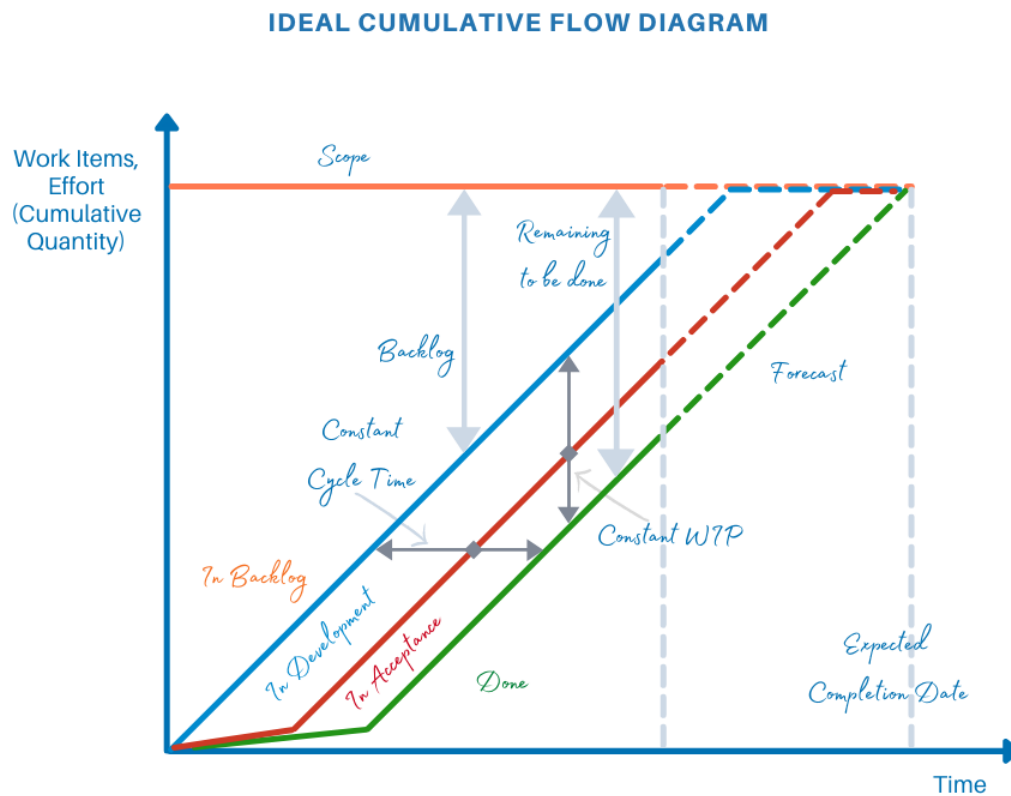
In the example above, the arrival rate is greater than the completion rate. We start more work than we finish and the WIP, and the cycle time, are constantly increasing, while the throughput is decreasing.

Figure 10.12: Ideal Cumulative Flow Diagram (1)



Figures 10-12 and 10-13 present a cumulative flow diagram of an ideal workflow, which has the following properties:

- The cycle time is short, constant and predictable
- The WIP in each process stage is constant and optimal
- The throughput is constant and predictable
- All work arrival and departure rates are constant and equal
- The process stages are perfectly synchronized – the inter-stage transitions are performed with a common and constant rhythm
- There is no unevenness (*Mura*) and the workflow demonstrates perfect stability
- The forecasts are reliable

Figure 10.13: Ideal Cumulative Flow Diagram (2)

LeanPM® Framework

© Lean Project Management Foundation

Of course, such a flow isn't only ideal but also idealistic. Real CFDs look very different, yet the ideal image helps us to see anomalies that signal obstacles, blockages and bottlenecks, increasing WIP and cycle time, productivity problems, quality issues and rework, changes in priorities, scope creep, etc.

The diagram does not reveal the root cause of the problems, but calls for in-depth analysis.

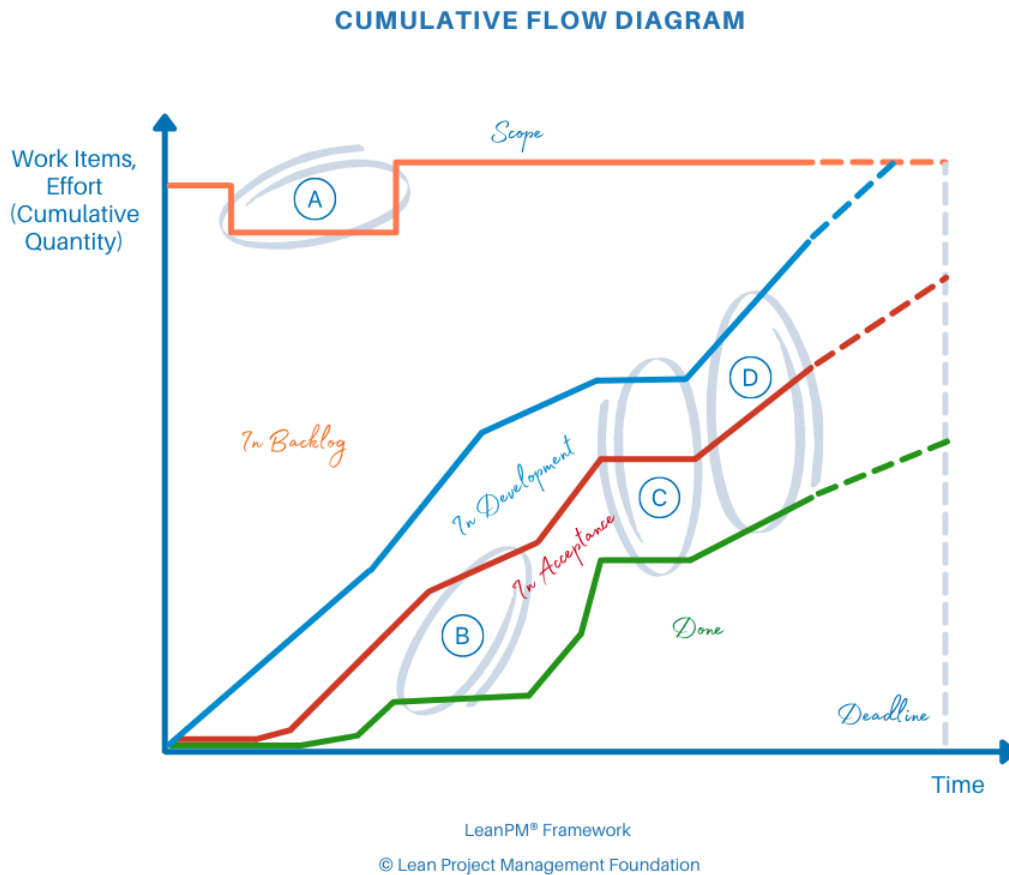
Figure 10.14: Cumulative Flow Diagram

Figure 10.14 shows several of the many possible problem situations that need deeper analysis:

- Scope change (A)
- Bottleneck (B) – the Acceptance sub-process takes on a new work but does not complete any work
- Work is blocked or is put on hold – for example, because of a change in priorities (C)
- WIP and cycle time increase and the completion date moves forward (D)

Measuring Queues

The queue size is a leading indicator of cycle time and the cost of time, and so knowing that is important for decision-making. The size of the queue can be measured according to the amount of work in the project that isn't yet done, and the average flow rate can be used to estimate the expected time to complete.

The queue time can also be measured by tracking a work item and measuring how long it takes to go through a queue.

To measure the waiting time of a sequence of queues in a multistage process, the times when an item arrives and departs the process have to be recorded, as well as the time spent in service. The queue

time is the difference between the total time elapsed from the beginning to the end of the process (cycle time) and the time that the item is being worked on.

Responding to Queues

Queues can be controlled by dealing with variability, demand, and supply.

Dealing with variability

Variability is inherent in project work, and it's a natural product of knowledge creation and innovation. Therefore, trying to eliminate the variability of requirements, size and duration of tasks, team productivity, reiterations, etc., is dangerous and can undermine effectiveness.

There are two options for dealing with variability:

- Provide slack in the project through capacity, WIP and time buffers
- Reduce uncertainty by creating knowledge through a series of experiments (hypothesis testing)

Buffers and experiments have a cost which must be traded off for the benefit of reducing queue size.

Controlling demand and supply

WIP constraints limit the number of projects in progress and the amount of work in the creation and absorption system of a project. Local WIP limits can be used to minimize the total queue cost by shifting the queue location.

However, there may still be long queues in the portfolio and project backlogs and demand may need to be constrained by:

- **Preventing arrivals** into the queueing system – blocking new projects and new requirements
- **Ejecting items** from the queueing system: removing projects by raising the ROI threshold, purging low-priority project requirements, and looking for aging work items and deciding whether they are still valuable or should be purged

Supply can be influenced by providing additional temporary capacity. Examples of supply-oriented responses to emerging queues are: [20]

- Applying additional resources to the queue – even small extra efforts can significantly reduce the queue size
- Directing resources working part-time on the project to high-variability priority tasks on the critical path
- Using high-powered experts who are intentionally not fully loaded and are able to respond quickly to an emerging queue
- Using generalizing specialists (T-shaped resources) who can handle a wide range of tasks and can be quickly reassigned where needed
- Cross-training team members at adjacent processes so they can help each other

Controlling Batch Sizes

A batch is the amount of work that passes from one stage to the downstream process stage. The size of the batches has a huge impact on the flow.

The sequential (waterfall) development of traditional project management provides the opportunity for the largest batch size and the longest project cycle time. One development phase must be completed for the next to begin. And the entire work product is transferred as one batch from one phase to another.

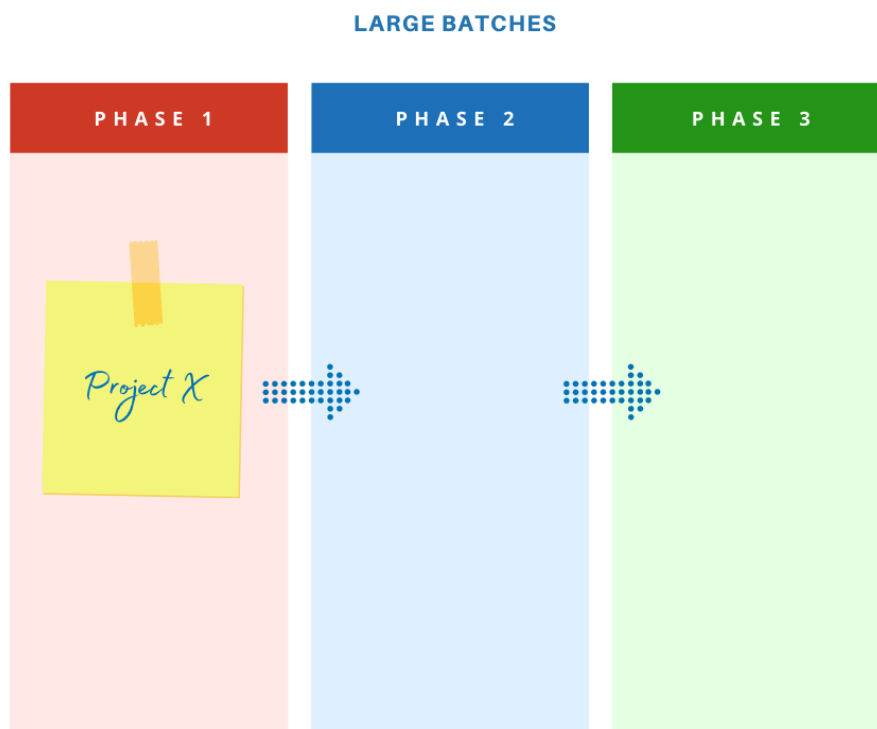
Large batches require more time to complete and, until then, the downstream work is put on hold. Cycle time is extended and the delivery of value is delayed. Trust is broken. Feedback opportunities are infrequent. Errors and problems are discovered later, when rework is costly. Learning and reducing uncertainty takes more time, and the risk of failure increases.

The work process lacks flexibility, and the response to problems and opportunities is slow. Complexity increases, which requires bigger integration effort and makes it difficult to identify the root causes of problems.

Large batches of work require funding in large batches, so the lean principle of incremental funding cannot be applied.

Finally, as we discussed above, large batches are bulk arrivals into a queueing system and they increase variability and the queue size.

Figure 10.15: Large Batches



LeanPM® Framework

© Lean Project Management Foundation

With such major shortcomings of large batches, smaller batches can significantly improve project performance.

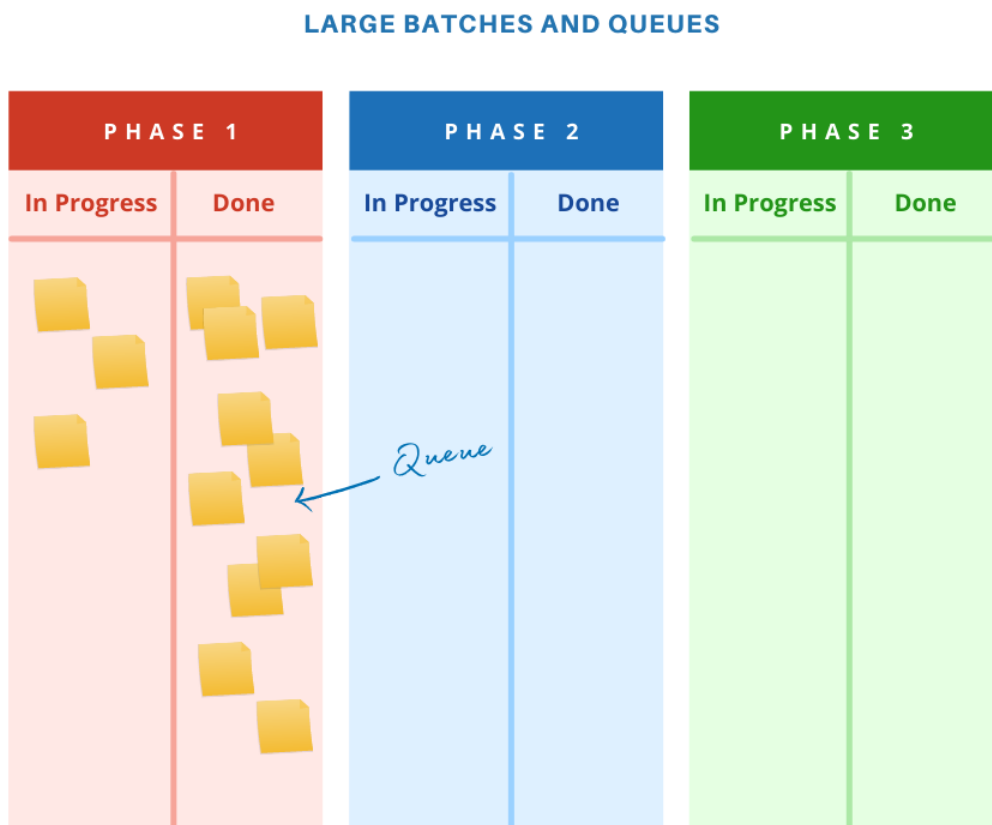
How small should the small batches be? The optimal batch size is a tradeoff between the cost of holding an item onto the batch and the transaction cost of sending the batch to the next process (e.g., the cost of software deployment).

Actually, small batches are the practical tool for achieving **incremental** creation and absorption.

Large batches are synonymous with a massive amount of work fatally falling over a series of steep drops, too steep to be overcome in the opposite direction.

If large batches are a problem, can't we break them down into separate work items and get small batches? Unfortunately, the magic will not always work.

Figure 10.16: Large Batches and Queues



LeanPM® Framework

© Lean Project Management Foundation

Let's look at Figure 10.16. Completed Phase 1 items pile in a queue in the *Phase 1: Done* column and do not proceed to the next step of the process. Any completed Phase 1 item cannot release a Phase 2 work item.

The Phase 1 work can get meaningful feedback from the stakeholders only after we fully complete it. Only when all Phase 1 items are finished can we do a phase review and move the entire work product to Phase 2.

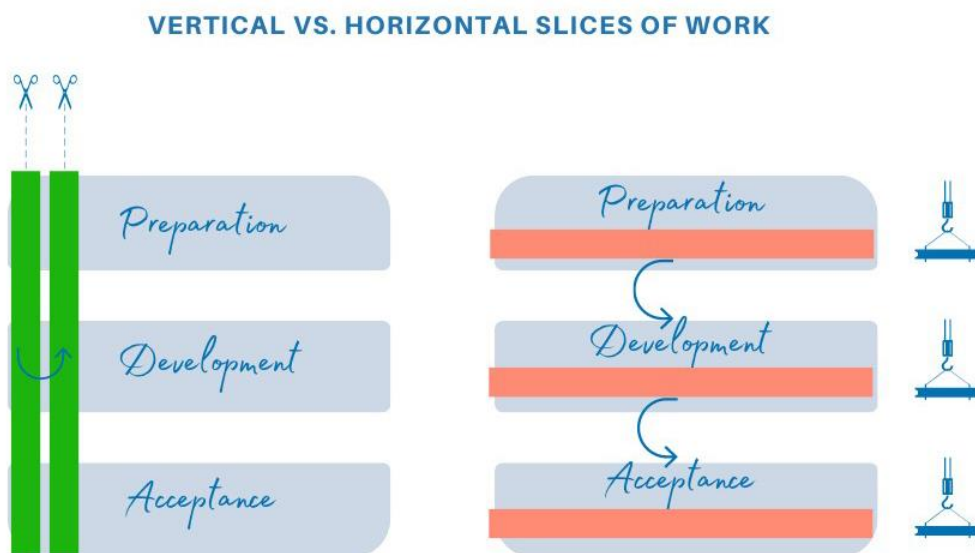
Hence, in order for a work item or a pool of work items to be a separate batch, two conditions must be met:

1. It must be able to receive meaningful stakeholder feedback
2. It must exit the project or pass from one process step to another and release subsequent work

Individual batches can form larger batches. The projects can be seen as batches, the size of which is reduced by using *Minimum Viable Projects*.

The capacity to work in small batches depends on our ability to create a workflow that allows for quick feedback and fast movement of work items down the stream. In stark contrast, the sequential development naturally requires working in large batches, while effective concurrent development needs smaller batches.

Figure 10.17: Vertical vs. Horizontal Slices of Work



LeanPM® Framework

© Lean Project Management Foundation

To reduce batch sizes, we can:

1. Use concurrent development and work in vertical slices of functionality.
2. Apply the principle of minimum viability not only to projects but also to work items.
3. Use a flexible system structure that allows independent changes in project deliverables.
4. Split the work items into the smallest units of customer value. This also applies to the units of process customer value.
5. Create a workflow that allows frequent and fast feedback.

6. Minimize handoffs and the amount of work being transferred between team members.
7. Integrate and absorb the work products regularly and often.

The Water and The Rocks

This is a Lean metaphor. Rocks are hidden under the water. When the water level is high, a small part of them appears above it, but when the level drops, we begin to see more and bigger rocks, and other obstacles.

The water level represents the WIP inventory, the size of the queues, the batch size and the PDCA cadences. When they are big and high, they hide many problems and weaknesses. When we start to constrain them, problems, weaknesses and bottlenecks become visible and we can address them.

To improve project performance, empirically limit WIP and the batch size, control queues and set more frequent PDCA cadences. Then deal with the issues, adjust and repeat.

REFERENCES - CHAPTER 10

- [1] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
- [2] Anderson, David M. 2014. *Design for manufacturability: how to use concurrent engineering to rapidly develop low-cost, high-quality products for lean production*. CRC Press
- [3] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
- [4] Ward, Allen, Jeffrey K. Liker, John J. Christiano and Durward Sobek II. Spring 1995. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster", Sloan Management Review
- [5] Ibid.
- [6] Ibid.
- [7] Sobek II, Durward K., Allen C. Ward and Jeffrey K. Liker. Winter 1999. *Toyota's Principles of Set-Based Concurrent Engineering*. Sloan Management Review
- [8] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
- [9] <https://cucumber.io/docs/gherkin/reference> (last accessed 2021-11-04)
- [10] Takeuchi, H; Nonaka, I (1986). "The New New Product Development Game". Harvard Business Review (January): hbr.org/1986/01/the-new-new-product-development-game (last accessed 2021-11-04)
- [11] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
- [12] Spearman Mark L. and H.J. James Choo (12/09/2018). *Unintended Consequences of Using Work-In-Process to Increase Throughput*. Project Production Institute: projectproduction.org/journal/unintended-consequences-of-using-wip-to-increase-throughput (last accessed 2021-11-04)
- [13] Ed Pound, Factory Physics Inc.: <https://factoryphysics.com/flow-benchmarking> (last accessed 2021-11-04)
- [14] Spearman Mark L. and H.J. James Choo (12/09/2018). *Unintended Consequences of Using Work-In-Process to Increase Throughput*. Project Production Institute: projectproduction.org/journal/unintended-consequences-of-using-wip-to-increase-throughput (last accessed 2021-11-04)
- [15] DeMarco, Tom. 2001. *Slack: getting past burnout, busywork, and the myth of total efficiency*. Broadway Books
- [16] Little, J. D. C. and Graves, S. C. (2008). Little's Law. In: D. Chhajed and T.J. Lowe (eds.) *Building Intuition: Insights from Basic Operations Management Models and Principles*, Springer Science+Business Media
- [17] Ibid.

- [18] Hopp, Wallace J. (2008). Single Server Queueing Models. In: D. Chhajed and T.J. Lowe (eds.) *Building Intuition: Insights from Basic Operations Management Models and Principles*, Springer Science+Business Media
- [19] DeMarco, Tom. 2001. *Slack: getting past burnout, busywork, and the myth of total efficiency*. Broadway Books
- [20] Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing