Introduction to Agile

What is Agile Project Management

By: Raffi Avedian **PMP, CSM, SP**

February 2018

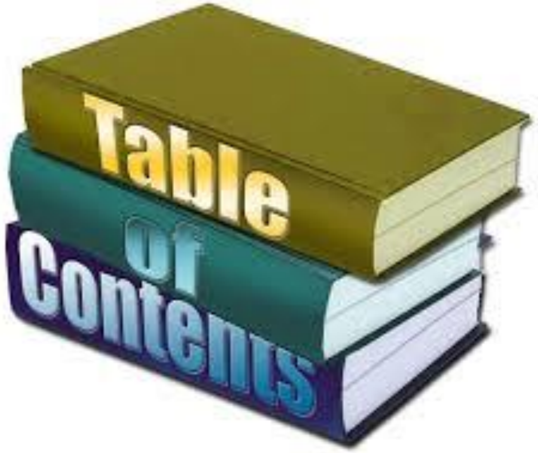
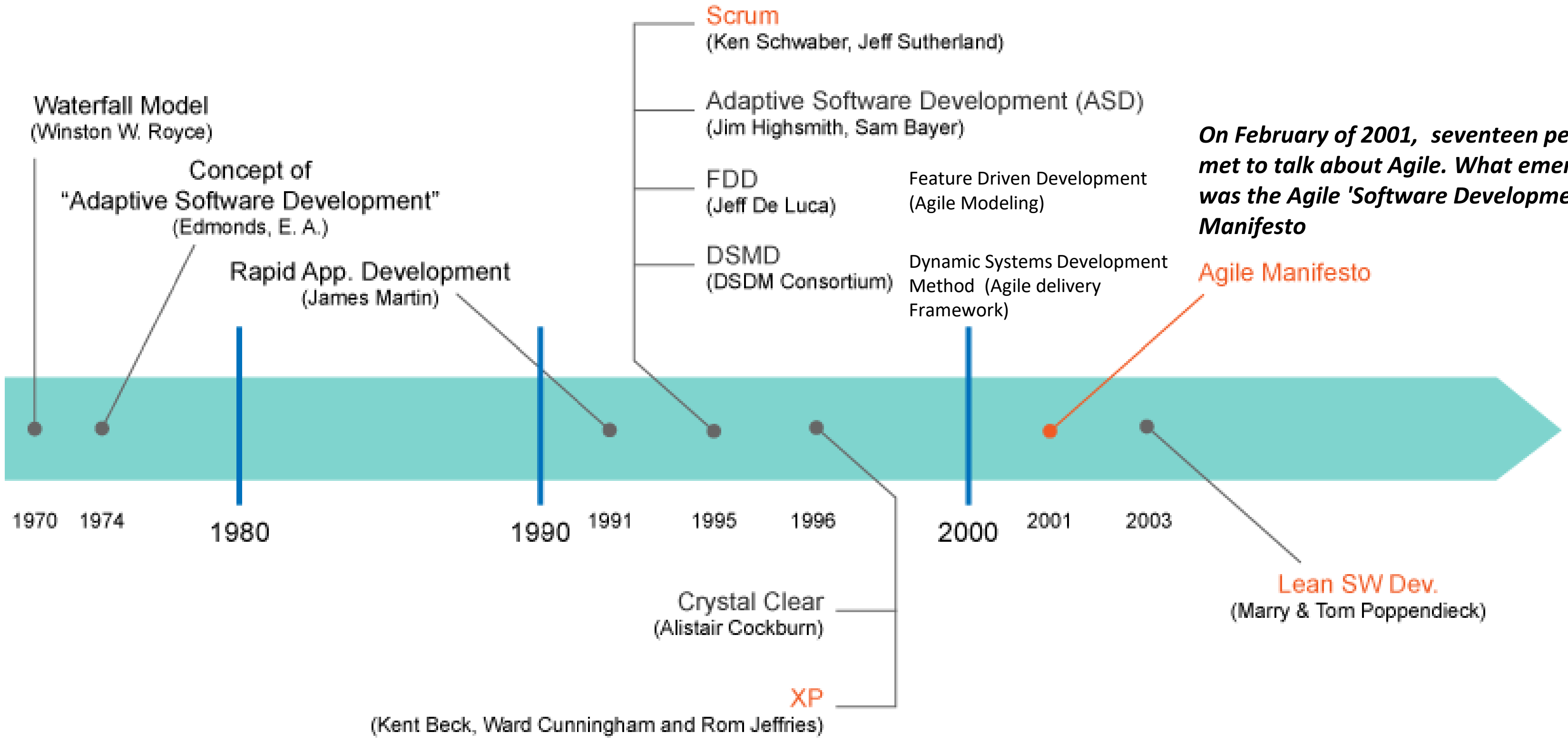


Table
of
Contents

Agile History

- ❑ In the early 1990s, software development faced a crisis. Industry experts estimated that the time between a validated business need and an actual application in production was about three years
- ❑ Within the space of three years, requirements, systems, and even entire businesses were likely to change. That meant that many projects ended up being cancelled partway through, and many of those that were completed didn't meet all the business's current needs, even if the project's original objectives were met
- ❑ In certain industries, the lag was far greater than three years. In aerospace and defense, it could be 20 or more years before a complex system went into actual use. The Space Shuttle program, which operationally launched in 1982, used information and processing technologies from the 1960s
- ❑ In1990s, several technology leaders frustrated with these long lead times and decisions made early in a project that couldn't be changed late, began informal talks about ways to develop software more simply, without the process and documentation overhead of **Waterfall** and other popular software engineering techniques of the time

Agile History Timeline



Waterfall Model
(Winston W. Royce)

Concept of
"Adaptive Software Development"
(Edmonds, E. A.)

Rapid App. Development
(James Martin)

Scrum
(Ken Schwaber, Jeff Sutherland)

Adaptive Software Development (ASD)
(Jim Highsmith, Sam Bayer)

FDD
(Jeff De Luca)

Feature Driven Development
(Agile Modeling)

DSMD
(DSDM Consortium)

Dynamic Systems Development
Method (Agile delivery
Framework)

Agile Manifesto

1970 1974 1980 1990 1991 1995 1996 2000 2001 2003

Crystal Clear
(Alistair Cockburn)

XP
(Kent Beck, Ward Cunningham and Rom Jeffries)

Lean SW Dev.
(Marry & Tom Poppendieck)

On February of 2001, seventeen people met to talk about Agile. What emerged was the Agile 'Software Development' Manifesto

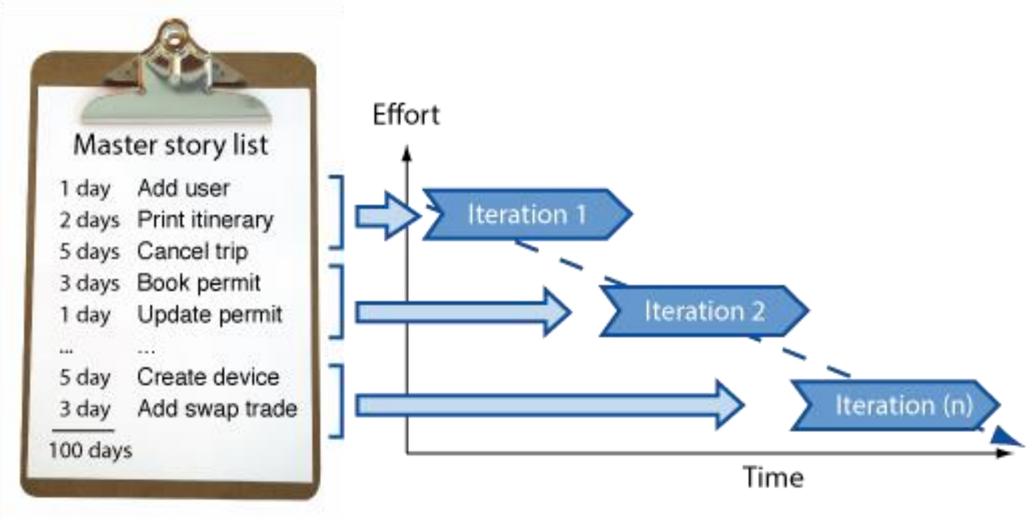
Hirotaka Takeuchi 1986 ↔ **1993 Jeff Sutherland, Jeff McKenna, John Scumniotales**

Evolution of Project Management

Simple Comparison	
Traditional Project Management	Modern Project Management (Agile)
Elements impacting execution of a project: a) Planning b) Control	**Elements impacting execution of a project**: a) Competitive environment b) Creativity and Innovation c) Planning d) Control
Project Success is measured by having: a) Well defined requirements b) Approved budget c) On-Time Delivery	**Project Success is measured by**: a) Delivery of Business Value (what does customer want) Less important are: a) Having well defined requirements b) Budget constraints c) Set Schedule Today's world has a much higher level of uncertainty which makes it difficult to always start a project with well-defined requirements
Requires a Project Manager	**Project Manager Role is distributed among**: a) Product Owner b) Scrum Master c) Project Team

Understanding Agile

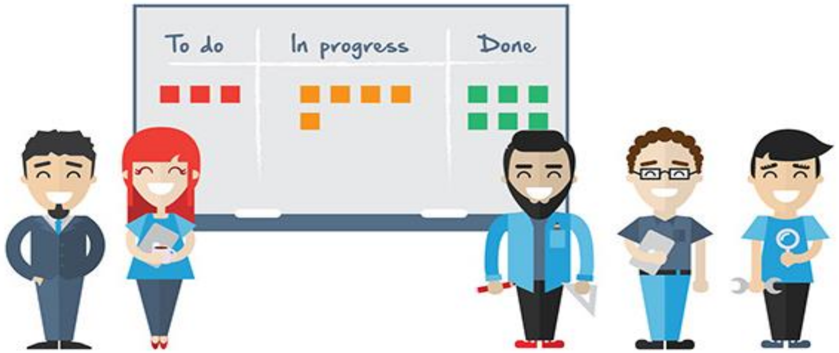
Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end. It works by breaking projects down into little bits of user functionality called *User Stories*, prioritizing them, and then continuously delivering them in short two week cycles called *Iteration*



Master story list

1 day	Add user
2 days	Print itinerary
5 days	Cancel trip
3 days	Book permit
1 day	Update permit
...	...
5 day	Create device
3 day	Add swap trade
100 days	

Scrum, Kanban & XP Agile Methodologies



Scrum is a tool used to organize work into small, manageable pieces that can be completed by a cross-functional team within a prescribed time period (called a sprint, generally 2-4 weeks long) to plan, organize, administer, and optimize this process.

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, while focusing on customer satisfaction by delivering what's needed when needed. Its guiding principles are: Communication, Simplicity, Feedback, Respect and Courage

Kanban (an industrial engineer at Toyota, developed Kanban to improve manufacturing efficiency) is also a tool used to organize work for the sake of efficiency. Where Scrum limits the amount of time allowed to accomplish a particular amount of work (by means of sprints), Kanban limits the amount of work allowed in any one condition (only so many tasks can be ongoing, only so many can be on the to-do list.)

Difference between Scrum & XP

- ❑ Scrum teams typically work in iterations (called sprints) that are from two weeks to one month long. XP teams typically work in iterations that are one or two weeks long
- ❑ Scrum teams do not allow changes into their sprints. XP teams welcome changes as long as work has not started
- ❑ Extreme Programming teams work in a strict priority order. Scrum teams have flexibility to choose what prioritized features to work on
- ❑ Scrum doesn’t prescribe any engineering practices. XP does (things like test-driven development, focus on automated testing, pair programming, simple design, refactoring, and so on)

Agile Manifesto

1. Customer Satisfaction through continuous delivery

2. Welcome changing requirements

3. Deliver working software frequently

4. Collaboration between business members and developers

5. Build projects around motivated individuals

6. Efficiency through face-to-face conversation

7. Working software is the primary measure of progress

8. Agile processes promote sustainable development

9. Technical excellence and good design enhances agility

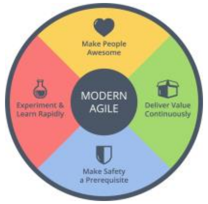
10. Promote Simplicity

11. Self-organizing teams

12. Self-improving team philosophy

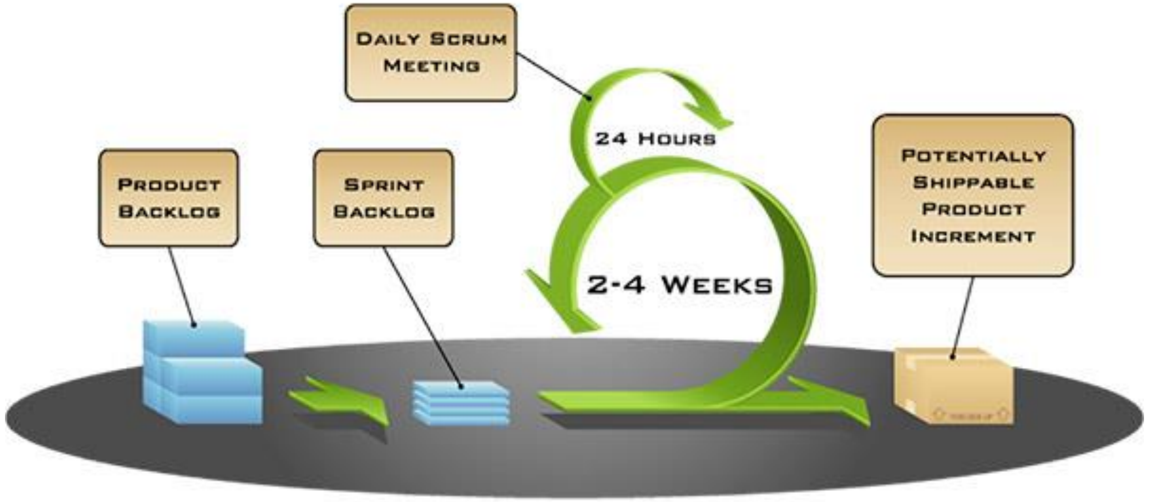
The Four Values of The Agile Manifesto

- Individuals and Interactions Over Processes and Tools
- Working Software Over Comprehensive Documentation
- Customer Collaboration Over Contract Negotiation
- Responding to change over following a Plan


Make People Awesome
Experiment & Learn Rapidly
MODERN AGILE
Deliver Value Continuously
Make Safety a Prerequisite

What is Modern Agile?

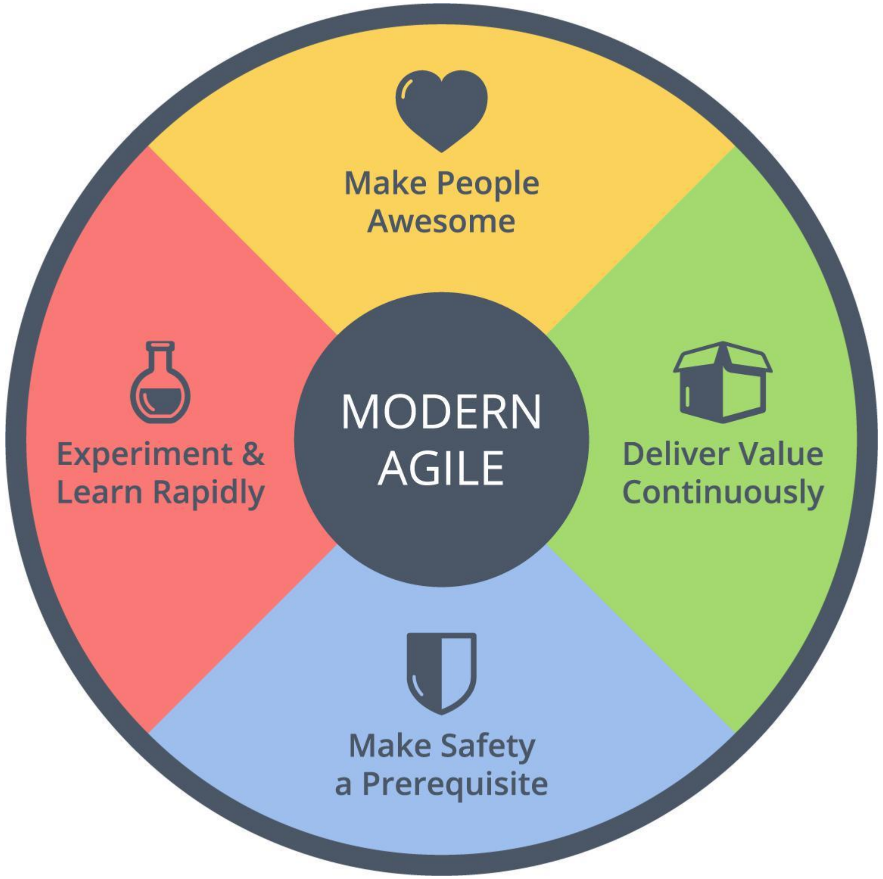
The industry has become overloaded with techniques, processes, methodologies and tools that have fallen under the Agile umbrella. Many consider these more marketing hype than Agile, feeling they are moving away from the simplicity that Agile principles promote. Modern Agile aims to move Agile to the next level, not by adding complexity, but by simplifying— stressing only adherence to four principles



Modern Agile Guiding Principles

Experiment & Learn Rapidly
Is a guiding principle of Modern Agile because it protects us from wasting time and helps us discover success faster

Make Safety a Prerequisite
Means establishing safety before engaging in potentially hazardous work



Make People Awesome
Experiment & Learn Rapidly
MODERN AGILE
Deliver Value Continuously
Make Safety a Prerequisite

"Make People Awesome"
Amazon has made **Customer Obsession a guiding principle since 1997 and it shows. If you make customers awesome, they tend to be natural promoters of your products or services**

Deliver Value Continuously
Anything valuable that hasn't been delivered isn't helping anyone. How might we deliver the right outcomes faster

Real world example of how to apply modern Agile guiding principles

EXAMPLE

Let’s say you decide you want to write a book. If you are driven by Modern Agile principles, you’ll want to quickly discover if the book will make people awesome. It would not be a safe investment of your time if no one has interest in your book. So, to learn whether there is genuine excitement about your topic, you’ll need to experiment and learn rapidly. To do that, you’ll need to deliver some value to some people and learn rapidly from their response. Perhaps you write a few pages, a short chapter or a small article, receiving feedback from an inner circle and then from a wider audience. As you write and learn, you’ll want to make sure your work is safe by using tools that protect your work by backing it up. If you are interviewing someone important for the book, you may want multiple recorders running, so if one breaks or runs out of space, you’ll have built-in fault tolerance to ensure you get a good recording.

Project Schedule / Cost Estimation Techniques

- Waterfall approach:

  Bottom-Up: Detail out all requirements and estimate each task to complete those requirements in hours/days, then use this data to develop the project schedule. In the software industry, the use of the bottom-up method has severe drawbacks due to today's speed of change. *Speed of change* means that the speed of new development tools and the speed of access to new knowledge is so great that any delay in delivery leaves one open to competitive alternatives and in danger of delivering an obsolete product

- Agile approach:

  Top-Down: The top-down method addresses this key issue, by using the information currently available to provide gross-level estimates. Rolling-wave planning is then used to incorporate new information as it's learned, further refining estimates and iteratively elaborating with more detail as the project progresses. This method of learning just enough to get started, with a plan to incorporate more knowledge as work outputs evolve, allows the project team to react quickly to adversity and changing market demand **(source: PMI)**

How to estimate project schedule in Agile?

- Determine ***Point Value*** for each item to be worked on. The most popular technique of gross level estimation is ***Planning Poker***, using Fibonacci sequence to assign a point value to a feature or item
- Determine ***Team Velocity***. A team's average velocity is used in forecasting a long-term schedule. Average velocity is calculated by summing and averaging the velocity measurements from the team's last three iterations
- The team's average velocity number is used to calculate the most likely scenario, while velocity numbers from the team's worst-performing iterations are used to calculate the most ***pessimistic forecast*** completion date. Using velocity from iterations where the team was able to complete more than expected provides the most ***optimistic forecast***

How to estimate project cost in Agile?

- A simple formula is used to determine the cost per point:

  Σ (loaded team salaries for period n) / points completed in period n

  A team whose total loaded salaries are $240,000 over six weeks, and completed 60 points of work in those three iterations, would have a cost per point of $4,000

- Use the following formula to determine budget:

  (Cost per point x total point value of items to be completed) + other expenses = forecast budget

  Budget estimates are based on what we know today

  In the example used, budget estimate is for the first release and not the entire project. The team could apply an additional 20% for the second release and an additional 5% for the last release, based on past experience

What is DevOps?

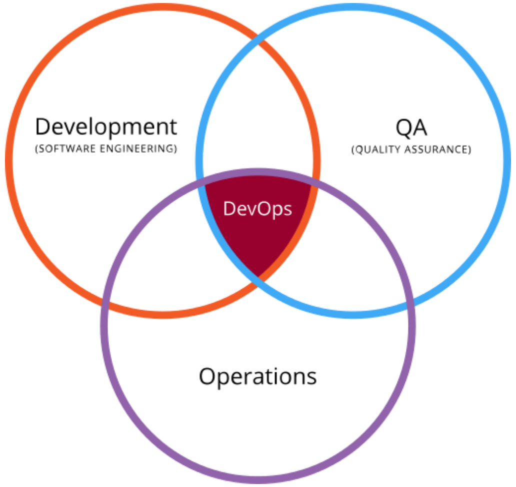
We are now on Agile bandwagon (hooray)! Development teams are working on the software in short sprints lasting not more than two weeks.

Having such a short release cycle helped the development team work on client feedback and incorporate it along with bug fixes in the next release.

While this Agile SCRUM approach brought agility to development, it was lost on Operations which did not come up to speed with Agile practices. Lack of collaboration between Developers and Operations Engineers still slowed down the development process and releases.

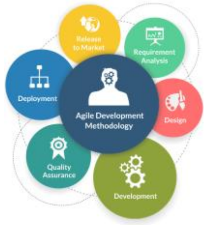
DevOps Methodology was born out of the need for **better collaboration and faster delivery**. DevOps enables continuous software delivery with less complex problems to fix and faster resolution of problems.

In summary**: *Agile*** is a set of values and principles about how to develop software. If you have ideas and you want to turn those ideas into working software, you can use the Agile values and principles as a way to do that. But, that software might only be working on a developer's laptop or in a test environment. You want a way to quickly, easily and at will move that software into production infrastructure, in a safe and simple way. To do that you need **DevOps** tools and techniques.



Venn Diagram

Does Agile work for infrastructure projects?



Agile works very well in projects where more is unknown than known. This is common in ***software development*** projects. In software development end users kind of know what they want. The requirements evolve over time. For example: end user thinks they need some kind of a shape. You start with something like circle. After an iteration the requirement might change from circle to square. After another iteration it might change from square to square filled with some color. The requirements evolve. Agile methodology has framework to support "**unknown**" throughout the project life cycle.

In contrast ***infrastructure*** projects have very well defined requirements (very few unknowns), strict dependency between tasks, mainly configuration driven, some tasks are done only during permitted windows (weekends or after business hours). For example ***storage expansion*** tasks are comprised of procurement, cabling, network and storage tasks. These tasks have dependency and follow one after another.   Projects like this are well suited for waterfall methodology. But infrastructure projects which have "**unknowns**" and similar to software development (BaaS, IaaS, PaaS) can greatly benefit from agile methodology

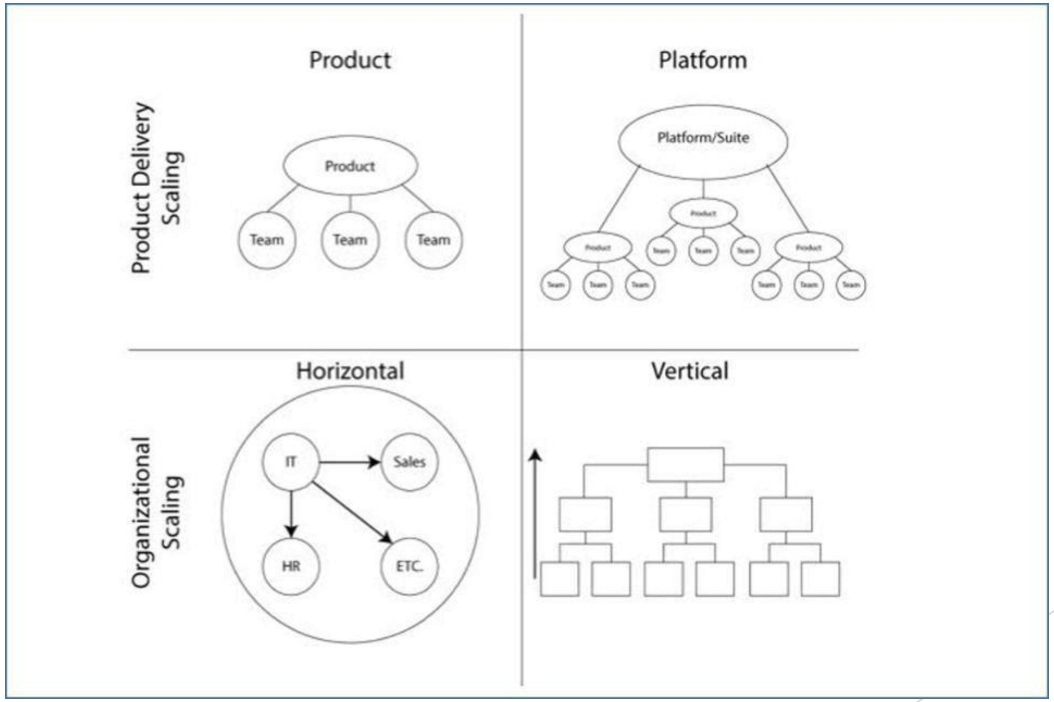
What is Scaling Framework?

Scaling is the application of practices of “Agility” across “Horizontal” and “Vertical” space through an organization. Scrum (a tool used to organize work into small, manageable pieces that can be completed by a cross-functional team within a prescribed time period) is a process unique to software development and can not be applied outside of that paradigm. Agile mindset and Lean thinking however can be practiced across all organizational teams.

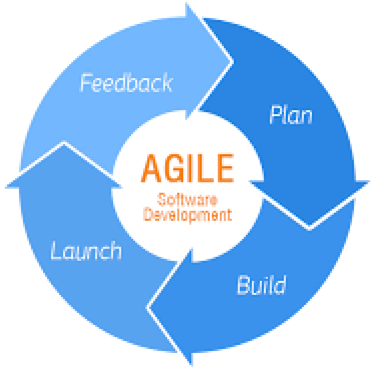
Scrum →

How do we make the entire organization use Agile in their day to day processes? “the Agile Mindset”

Agility →



SAFe: Agile Software Development



Agile software development refers to a group of software development methodologies (*Scrum, Kanban, and Extreme Programming*) based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams

Agile development refers to any development process that is aligned with the concepts of the Agile Manifesto. The Manifesto was developed by a group leading figures in the software industry, and reflects their experience of what approaches do and do not work for software development

SAFe: Lean Software Development Principles

Lean development can be summarized by seven principles, very close in concept to lean manufacturing principles:

❑ Eliminate Waste
❑ Amplify Learning
❑ Decide as late as possible
❑ Deliver as fast as possible
❑ Empower the team
❑ Build integrity in
❑ See the whole

SAFe: What is Systems Thinking

Systems Thinking is a management discipline that views the complete organization in relation to its environment. It provides a means of understanding, analyzing and talking about the design and construction of the organization as an integrated, complex composition of many interconnected systems (human and non-human) that need to work together for the whole to function successfully.

THINKING
Events
Information
Language
Behavior
SYSTEMS THINKING
You, Your organization or Society

What is SAFe?

The **Scaled Agile Framework** (**SAFe**), is intended to guide enterprises in scaling lean and agile practices. SAFe promotes alignment, collaboration, and delivery across large numbers of agile teams (Agile on steroids).

It was developed by and for practitioners, by leveraging three primary bodies of knowledge:

1. Agile software development (Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context)
2. Lean product development (Eliminate Waste, Amplify Learning, Decide as late as possible, Deliver as fast as possible, Empower the team, Build integrity, See the whole)
3. Systems thinking (Is a management discipline that concerns an understanding of a system by examining the linkages and interactions between the components that comprise the entirety of that defined system)

Agile Release Train (ART)

50-125 practitioners. May include different disciplines.
AGILE RELEASE TRAIN
. . .

- ## What is it?

  The ART metaphor describes the program level teams, roles, and activities that incrementally deliver a continuous flow of value. ARTs are virtual organizations formed to span functional boundaries, eliminate unnecessary handoffs and steps, and accelerate value delivery by implementing SAFe Lean-Agile principles and practices

- ## Who is responsible for it?

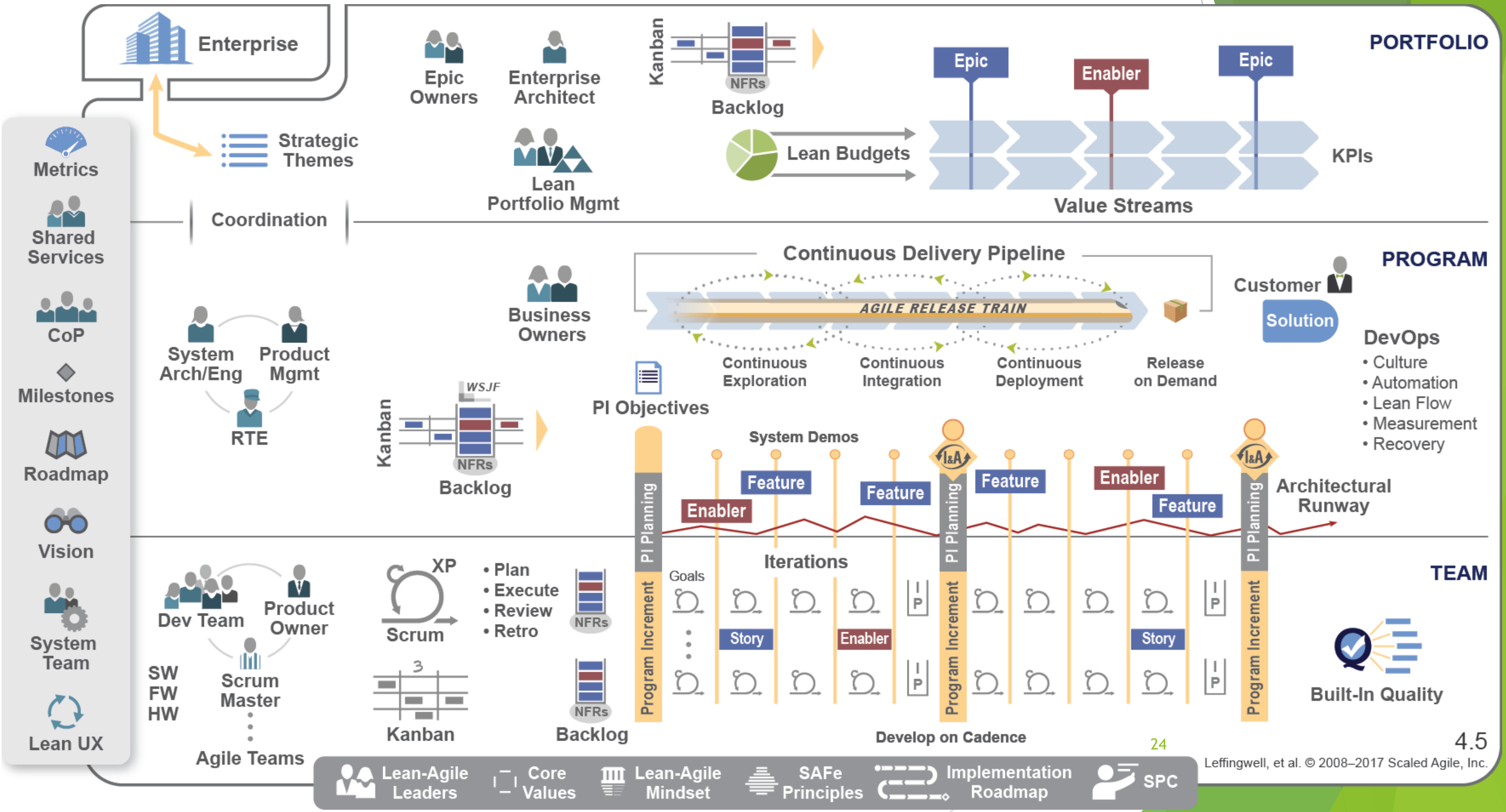
  System Architect/Engineer– Is an individual or small cross-discipline team that truly applies Systems Thinking. They define the overall architecture for the system, help define Nonfunctional Requirements (NFRs), determine the major elements and subsystems, and help design the interfaces and collaborations among them.

  Product Management – Is the internal voice of the Customer and works with customers and Product Owners to understand and communicate their needs, define system features, and participate in validation. They are responsible for the Program Backlog.

  Release Train Engineer (RTE) – Is a servant leader and the chief Scrum Master for the train. The RTE facilitates optimizing the flow of value through the program using various mechanisms, such as the Program Kanban, Inspect & Adapt (I&A) workshop, and PI Planning.

  Business Owners – Are a small group of stakeholders who have the business and technical responsibility for fitness for use, governance, and return on investment (ROI) for a Solution developed by an ART. They are primary stakeholders in the ART and actively participate in ART events

Scaled Agile For Lean Enterprises



Should a Project Manager learn Agile?

The impact of Agile on the role of many project managers is likely to be significant. On small, single-team Agile projects, you may not find someone called a “Project Manager”. However:

- ✓ There certainly is a need for someone to coach and mentor the team on Agile Project Management practices.
- ✓ There is also a need for project managers on larger and more complex enterprise-level projects but even at that level, some understanding of Agile is essential.

This “raises the bar” for project managers significantly. It requires project managers to develop a fresh new perspective to see Agile and traditional, plan-driven project management approaches as complementary to each other rather than competitive and to learn how to blend the two approaches in whatever proportions are needed to fit any situation

The PM role in a Lean and Agile world

In the lean and agile world, the project manager does not have an official role

The Scrum practice prescribes distributing the PM role among the Scrum team members. However, there are varying opinions and experiences assigning any combination of the 3 roles to a PM.

1. The Scrum Master (The ***Scrum Master*** does anything possible to help the team perform at their highest level. This involves removing any impediments to progress, facilitating meetings, and doing things like working with the product owner to make sure the product backlog is in good shape and ready for the next sprint)
2. The product Owner (The ***Product Owner*** is typically a project's key stakeholder. Part of the product owner responsibilities is to have a vision of what he or she wishes to build, and convey that vision to the scrum team. This is key to successfully starting any agile software development project. The agile product owner does this in part through the product backlog, which is a prioritized features list for the product)
3. The development team member (Responsible for the project’s creation and delivery. The team members will normally be comprised of developers, QA, and documentation. They are responsible for planning, design, development, testing, and project delivery)

The Scaled Agile Framework (SAFe) practice lists the PM as a potential for the Release Train Engineer (RTE). Other agile practitioners describe the PM as a coach and facilitator

Project Manager as a ScrumMaster

- ❑ **Authority:** The traditional project manager role moves from a command-and-control, hierarchical position to a servant-leader or facilitator position
- ❑ **Requirements:** The product owner assumes the responsibility for ensuring the requirements are defined
- ❑ **Work Assignments:** The team takes ownership and accountability for meeting the project and team goals
- ❑ **Managing Stakeholder Expectations:** Product owner provides direction and leadership to the team
- ❑ **Leadership and Support:** The Scrum Master serves the product owner and the team so that they are better able to do their jobs by assisting them, facilitating creativity and fostering empowerment
- ❑ **Removes Impediments:** The Scrum Master helps remove obstacles and support the team

The

- A project manager (PM) is a highly skilled knowledge worker who has received rigorous training and knowledge in the process of achieving a globally recognized certification

- The knowledge and skills obtained through certification is transferable in the lean and agile organization. In a competitive business climate, all available brainpower must be present on deck to enable the organization to achieve **enterprise** agility and scale to meet customer, compliance, financial markets, internal opportunities, and competitive demands

- The SAFe model represents enterprise agility and extends Scrum beyond the team execution level into the organization. The PM role is best characterized through enterprise agility and viewed through the lens of portfolio, program, and the execution teams that are aligned to ensure maximum customer value

Agile Certified Professional

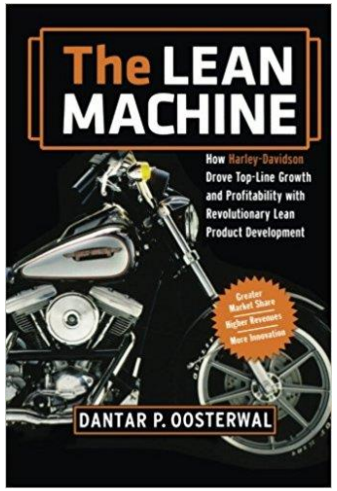
Become a PMI Agile Certified Professional (PMI-ACP):

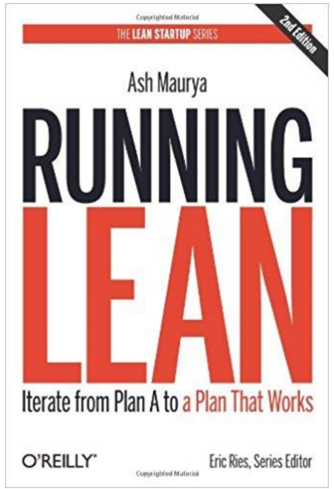
General Project Experience	Project Experience	Training	Examination
2,000 hours working on project teams within the last five years. The PMP certification satisfies this requirement	1,500 hours working on agile projects within the last three years	21 hours of training	Pass a 200-question test

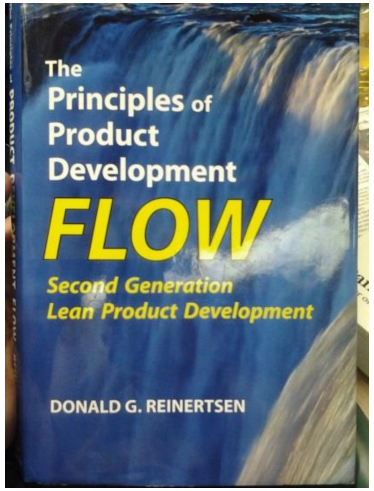
The PMI Agile Certified Practitioner (PMI-ACP)® formally recognizes your knowledge of agile principles and your skill with agile techniques. It will make you shine even brighter to your employers, stakeholders and peers

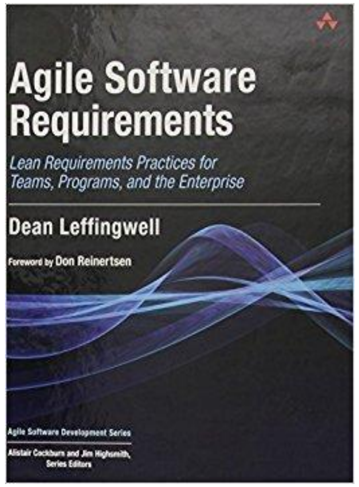
The PMI-ACP spans many approaches to agile such as Scrum, Kanban, Lean, extreme programming (XP) and test-driven development (TDD.) It will increase your versatility, wherever your projects may take you.

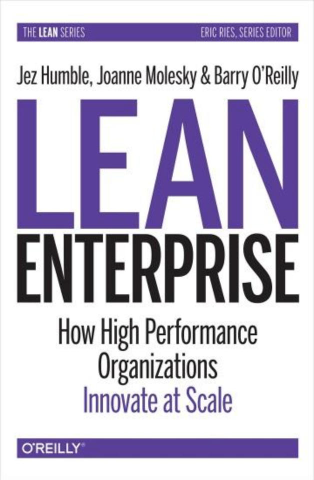
Recommended Reading

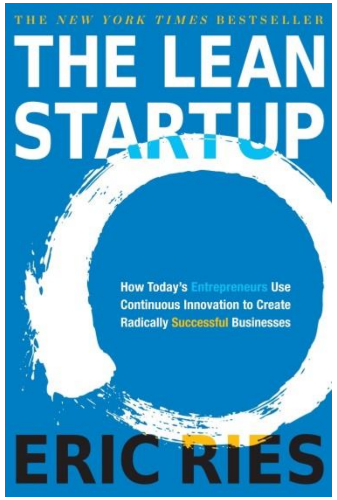

The LEAN MACHINE
How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development
Greater Market Share Higher Revenues More Innovation
DANTAR P. OOSTERWAL

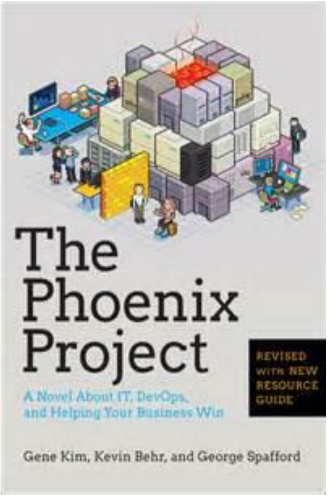


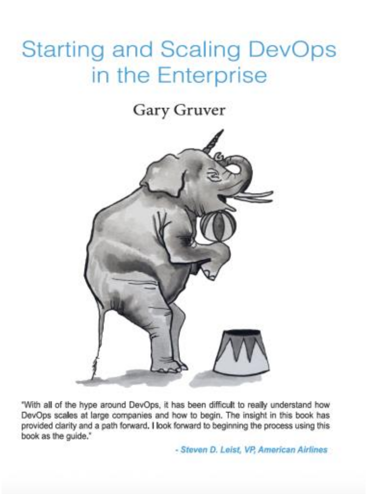

The Principles of Product Development
FLOW
Second Generation Lean Product Development
DONALD G. REINERTSEN

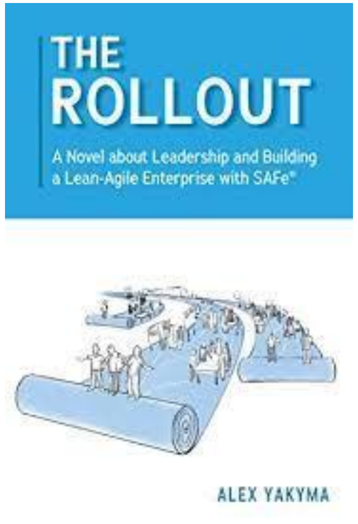

Agile Software Requirements
Lean Requirements Practices for Teams, Programs, and the Enterprise
Dean Leffingwell
Foreword by Don Reinertsen
Agile Software Development Series
Alistair Cockburn and Jim Highsmith, Series Editors


THE LEAN SERIES
ERIC RIES, SERIES EDITOR
Jez Humble, Joanne Molesky & Barry O'Reilly
LEAN ENTERPRISE
How High Performance Organizations Innovate at Scale
O'REILLY


THE NEW YORK TIMES BESTSELLER
THE LEAN STARTUP
How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses
ERIC RIES


The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
REVISED WITH NEW RESOURCE GUIDE
Gene Kim, Kevin Behr, and George Spafford


Starting and Scaling DevOps in the Enterprise
Gary Gruver
"With all of the hype around DevOps, it has been difficult to really understand how DevOps scales at large companies and how to begin. The insight in this book has provided clarity and a path forward. I look forward to beginning the process using this book as the guide."
- Steven D. Leist, VP, American Airlines


THE ROLLOUT
A Novel about Leadership and Building a Lean-Agile Enterprise with SAFe®
ALEX YAKYMA

Recommended Videos

- Daniel Pink Drive - https://www.youtube.com/watch?v=uwA97yWz9Uc
- Nordstrom Innovation Lab - https://www.youtube.com/watch?v=szr0ezLyQHY
- Agile Product Owner in a Nutshell - https://www.youtube.com/watch?v=502ILHjX9EE
- John Kotter Our Iceberg is Melting Video - https://www.youtube.com/watch?v=Gh2xc6vXQgk
- Submarine Captain - David Marquet - https://www.youtube.com/watch?v=OqmdLcyES_Q
- Servant Leadership - https://www.youtube.com/watch?v=aKk0AaaFqtU
- Spotify Engineering Culture Part 1 - https://www.youtube.com/watch?v=4GK1NDTWbkY
- Spotify Engineering Culture Part 2 - https://www.youtube.com/watch?v=X3rGdmoTjDc
- Radical Candor The Surprising Secret to Being a Good Boss - https://www.youtube.com/watch?v=4yODalLQ2lM
- How Business Stakeholders Work With Agile Teams - https://www.stickystories.co/whats-different-for-business-stakeholders-when-its-an-agile-approach/
- The Backwards Brain Bicycle - Smarter Every Day 133 - https://www.youtube.com/watch?v=MFzDaBzBlL0

Glossary of Terms

Agile Manifesto	Is a formal proclamation of 4 key values and 12 principles to guide an iterative and people-centric approach to software development.
ART	Agile Release Train - Is a virtual organization (50 – 125 people) that plans, commits, and executes together
BaaS	Backend as a Service
Backlog	A backlog is an ordered list of items representing everything that may be needed to deliver a specific outcome. There are different types of backlogs depending on the type of item they contain and the approach being used
Business Agility	Business agility is the ability of an organization to sense changes internally or externally and respond accordingly in order to deliver value to its customers
CoP (in SAFe)	Culture built on professional networking, personal relationships, shared knowledge, and common skills. Combined with voluntary participation, CoPs provide knowledge workers with opportunities to experience autonomy, mastery, and purpose beyond their daily tasks on an Agile Release Train (ART)
Cross Functional	Relating to a system whereby people from different areas of an organization work together as a team
DoD	Definition of Done
Enabler	Enablers support the activities needed to support efficient development and delivery of future business requirements
Epic	An epic is a large user story
Extreme Programming	Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development
IaaS	Infrastructure as a Service
Iteration	An iteration is a timebox during which development takes place. The duration may vary from project to project and is usually fixed
KPI	Key Performance Indicators – Example: KPIs of a shoe factory, would be number of shoes *without defects* made in a period of time

Glossary of Terms continued..

Mob Programming	Mob Programming is a software development approach where the whole team works on the same thing, at the same time, in the same space, and at the same computer
NFR	Non-Functional Requirements (security, reliability, performance, maintainability, scalability)
PaaS	Platform as a Service (Cloud)
Pair Programming	Is an agile software development technique in which two **programmers** work together at one workstation. One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in
PI	A routine, face-to-face event, with a standard agenda that includes a presentation of business context and vision followed by team planning breakouts—where the teams create their Iteration plans and objectives for the upcoming PI
Planning Poker	An approach to estimation used by Agile teams. Each team member "plays" a card bearing a numerical value corresponding to a point estimation for a user story
Refactoring	Refactoring consists of improving the internal structure of an existing program's source code, while preserving its external behavior
Retrospective	The team meets regularly to reflect on the most significant events that occurred since the previous such meeting, and identify opportunities for improvement
RTE	Release Train Engineer - The RTE's major responsibilities are to facilitate the ART events and processes and assist the teams in delivering value. RTEs communicate with stakeholders, escalate impediments, help manage risk, and drive relentless improvement
Scrum Master	The scrum master is responsible for ensuring the team lives agile values and principles and follows the practices that the team agreed they would use
Scrum of Scrums	A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10
Spike	In **agile** software development, a **spike** is a story that cannot be estimated until a development team runs a time boxed investigation. The output of a **spike** is an estimate for the original story
UI	User interface (UI) is the series of screens, pages, and visual elements—like buttons and icons—that you use to interact with a device
UX	User experience (UX), is the internal experience that a person has as they interact with every aspect of a company's products and services

END OF THE PRESENTATION
THANK YOU FOR YOUR ATTENTION
www.memegenerator.net