What is Agile
Glossary: Agile, Lean, Scrum, Sprint, Design Thinking

- **Agile** and **Lean** are technically different things, but it’s okay to consider them similar – experimental & iterative approaches for highly uncertain challenges.

- **Lean Manufacturing (Lean TQM)** and **Lean Startup** are technically different things, but it’s okay to consider both as **Lean**.

- **Scrum** is one of the most popular Agile implementation frameworks. **Sprint** is a Scrum terminology. It’s what the iteration cycle in Scrum is called.

- Loop **Design Thinking** in the Agile bucket too, as again it’s in the same spirit.

- To summarize, consider: **Lean ≈ Agile > Scrum > Sprint** and **Design Thinking ∈ Agile**

- There are further derivatives such as **Design Sprints** and **USM** (User Story Mapping), which are again purpose driven tools that have their own valid use (I use them both too). **XP** (Extreme Programming), **Kanban** (not to be confused with Lean Manufacturing’s Kanban) etc. are software development framework alternatives to Scrum.

- **Scaled Scrum** or **Scrum of Scrums** is looking like a battle of methodologies with **LeSS, SAFe, Nexus, Prince2** etc. each claiming superiority.
Is Agile a fad?

• **Agile** did not invent trial and error.

• In fact, **PDCA** (Plan-Do-Check-Act, W. Edwards Deming, 1950), **OODA** (Observe-Orient-Decide-Act, John Boyd, 1976) and many more thoughts on trial and error going all the way back to Socrates (circa 400BC) and Confucius (circa 500BC), all teach the same.

• They are all models of *experimentation & iteration*.

• By default, we humans don’t like trial and error – it’s a biological trait stemming from our survival instincts of uncertainty avoidance and resistance to change. Meanwhile, risk taking is rewarding, and the probability of success from risk taking increases through trial and error – that’s a fact from pre-historic times. That’s why throughout history many strings of experimental and iterative models have emerged as each is a new attempt to get better at trial and error.

• Agile is just one such latest attempt to get *experimentation & iteration* right. It doesn’t matter what it’s called – the spirit is same.
The many flavors of **experimental & iterative** approaches

- **Waterfall**
- **PDCA** (a.k.a. Deming Cycle, Kaizen - Continuous Improvement Cycle)
- **OODA** (Observe, Orient, Decide, Act)
- **Lean Startup** (Build, Measure, Learn)
- **Scrum**
  - Backlog
  - Sprint Planning
  - Sprint Retrospective
  - Sprint Review
  - Daily Stand Up
- **Design Thinking** (Define, Ideate, Prototype, Test)
Why Agile?
(1) Agile is an antidote to waterfall project management

**Waterfall**
Project Management

- One waterfall
- No testing until completed

**VS**

**Agile**
Product Development

- Many Sprint iterations
- Many Small test with MVPs
The ambivalent nature of Agile product development
(2) Agile is an antidote to the vertical organization

**Traditional Organization**
- Centralized hierarchical structure
- Leaders make strategic decisions
- Middle managers make tactical decisions and gives instructions
- Rank & file execute and deliver
- Functional silos

**Versus**

**Agile Organization**
- Decentralized network structure
- Self-organization
- Team makes its own decisions
- Leaders as facilitators, managers as coaches
- Cross-functional teams
Soft reason, why Agile?

Because we need a new way to **lead** in the age of **networked knowledge workers**.

Social and digital is accelerating the service economy transformation in our world – the essence of Globalization.

In the service economy, the competitive advantage is knowledge (including skills and expertise) and connectivity (access to resources – technology is the enabler). At the individual worker level, professional development is naturally gravitating to these two attributes.

Innovate or die from obsolesce – this is the reality that organizational leaders face today. And the world is now way to complex for a single master mind leader to know what to do. Survival of the organization depends on how well leaders can tap into the collective intelligence and drive of today’s professional – the networked knowledge worker.

The new organization needs to follow the amorphous nature of the knowledge workers’ networks. It’s time to evolve from the traditional vertical, hierarchical organization, to a more cross-functional, networked organization.
Soft reason, why Agile?

- **Personal mastery** is strong intrinsic motivation for *self-actualization*. When our autonomy is respected, we take strong pride in our work.

- When we group, it’s natural for us to go over and beyond for the collective good; i.e. **self-transcendence**. Our strong sense of **belonging**, powers self-organization.

- **Trust**, is at the heart of agile. When we believe in our ability to work autonomously and self-organize, we tap into the highest forms of human motivation.

# Hard reason, why Agile?

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Investment</th>
<th>Probability of Success</th>
<th>Return on Success</th>
<th>Weighted Average ROI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investment in one go</td>
<td>$100</td>
<td>20%</td>
<td>x20</td>
<td>$400</td>
</tr>
<tr>
<td>Investment in 5 iterations</td>
<td>$20</td>
<td>20%</td>
<td>x20</td>
<td>$80</td>
</tr>
<tr>
<td></td>
<td>$20</td>
<td>40%</td>
<td>x20</td>
<td>$160</td>
</tr>
<tr>
<td></td>
<td>$20</td>
<td>60%</td>
<td>x20</td>
<td>$240</td>
</tr>
<tr>
<td></td>
<td>$20</td>
<td>80%</td>
<td>x20</td>
<td>$320</td>
</tr>
<tr>
<td></td>
<td>$20</td>
<td>100%</td>
<td>x20</td>
<td>$400</td>
</tr>
</tbody>
</table>

Outcome: $100

**Product Market Fit**

**ROI 3x**

**$1,200**

*High risk, high return opportunity*

Now you can **MILK THE COW**

**Agile DE-RISKS**

**Agile MAKES MONEY**
Agile Scrum
**Scrum**: the most popular implementation framework in Agile

- Started as a software development framework, but wide application today; e.g. applied to business, operations.

- A *time-constraint approach* where work is delivered in short (typically 2 week “Sprints”) iterations.

- A team based approach: Scrum Teams are comprised of 3 to 9 Development Team (“DT”) members, one Scrum Master (“SM”), and one Product Owner (“PO”).

- The SM is the “how” guy. The PO is the “what” guy. **Both SM and PO are not bosses (not managers).** Scrum Teams *self-organize.*
Scrum is a lightweight framework for developing complex products

Scrum Product Backlog & Sprint Backlog
Scrum – Sprint Planning

• Have Product Backlog refined and prioritized
  Think of everything that would be needed in the product. User Stories are helpful ways of expressing the value of Product Backlog items.

• Estimate work
  Estimate how much effort it will take each Product Backlog item to get done. Use Planning Poker etc.

• Pull Product Backlog items into the Sprint Backlog
  What are the items that have the biggest business impact, that are most important to the customer, that can make the most money, and are the easiest to do?

• Decide what To Do in the coming Sprint
  Only put in what can be done within the Sprint. Definition of Done for each Product Backlog item is a prerequisite.
**Scrum Board (Kanban Board)**

<table>
<thead>
<tr>
<th>Backlog</th>
<th>TO DO</th>
<th>WIP</th>
<th>DONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Scrum Board (Kanban Board, *variation*)

<table>
<thead>
<tr>
<th>Backlog</th>
<th>TO DO</th>
<th>WIP</th>
<th>TESTING</th>
<th>DONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Triage**
- 

**Bumped**
- 

**Killed**
- 

---

*Life cycle*  
[agile-od.com]
Scrum – Daily Scrum (a.k.a. daily stand-up)

• Everyday, same time, 15 minutes

• Everyone shares 3 things

✓ What work was done yesterday?
✓ What work is planned for today?
✓ Any impediments in the way?

Alternatively, the team can also go through in sequence of the Done and To Do items on the Kanban Board. This can be a better format if multiple team members are involved in the work for each Scrum Board item.

• The Scrum Master is not the boss (nor the Product Owner)

The Scrum Master doesn’t give instructions to the Development Team what to do. As a matter of fact, the Daily Scrum is required only for the Development Team. The Scrum Master does not have to facilitate.

• Not a status update meeting

Status updates just give snap shots. What’s important is to check-in if all is flowing. If someone is stuck, fellow team members come in to help, and the Scrum Master facilitates removal of the impediment where necessary. Even if everything is going well, there’s still room for improvement. Let the team speak up.
Scrum – Sprint Review

• The end of Sprint “what” meeting
• Scrum Team invites stakeholders, partners, customers to attend
• PO facilitates
• Scrum Team will showcase what was built and released, often as a demo
• PO is responsible for asking real feedback, not just pat on backs
• Feedback is to be incorporated back into product design, and consequently Backlog rebuilding
Scrum – Sprint Retrospective

- The end of Sprint “how” meeting
- Only Scrum Team attends
- SM facilitates

The purpose of the Sprint Retrospective is to:

- Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- Identify and order the major items that went well and potential improvements; and,
- Create a plan for implementing improvements to the way the Scrum Team does its work.
A lot can go wrong with Scrum

- The Spiritless Scrum: Sprints as Mini-Waterfalls
- Territorial Scrum
- Scrum Master in Command
- Sprint Till You Drop

Advice: If you’re going to do Scrum, do it properly

A lot can go wrong with Scrum

Reference reading (find them on https://agile-od.com/insight)
Common Agile Traps
Is Agile a methodology?

- methodology

- approach, framework, modality, mindset, style, attitude, a way, spirit, culture

*Agile is process driven, but not fixed process.*
## The Spirit of Agile

<table>
<thead>
<tr>
<th>Principle</th>
<th>Symbol</th>
<th>Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relentless pursuit of customer value</td>
<td>![Hand symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Accept uncertainty</td>
<td>![Arrow symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Build, Measure, <strong>Learn</strong></td>
<td>![Fish symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Incremental iteration</td>
<td>![Circle symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Trust in autonomy</td>
<td>![People symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Failure is welcome</td>
<td>![Checkmark]</td>
<td>[X]</td>
</tr>
<tr>
<td>No one gets blamed for trying and failing</td>
<td>![No symbol]</td>
<td>[X]</td>
</tr>
<tr>
<td>Leaders as facilitators</td>
<td>![Leader symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Managers as coaches</td>
<td>![Manager symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Cross-functional teams</td>
<td>![Star symbol]</td>
<td>![X]</td>
</tr>
<tr>
<td>Self-organized teams</td>
<td>![Self-organized symbol]</td>
<td>![X]</td>
</tr>
</tbody>
</table>

*Source: Lifecycle agile-od.com*
## Agile and agility

<table>
<thead>
<tr>
<th>Category</th>
<th>Agile/Agility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Learning</strong></td>
<td>Adapt and iterate</td>
</tr>
<tr>
<td><strong>Learning</strong></td>
<td>Experiment</td>
</tr>
<tr>
<td><strong>Learning</strong></td>
<td>Fail fast, learn fast</td>
</tr>
<tr>
<td><strong>Learning</strong></td>
<td>Perform under uncertainty and ambiguity</td>
</tr>
<tr>
<td><strong>Org</strong></td>
<td>Stakeholder and customer collaboration</td>
</tr>
<tr>
<td><strong>Org</strong></td>
<td>Cross-functional teams</td>
</tr>
<tr>
<td><strong>Org</strong></td>
<td>Empowered, autonomous, self-organized teams (vs teams with line managers)</td>
</tr>
<tr>
<td><strong>Org</strong></td>
<td>Leaders and managers as coaches and facilitators (vs leaders and managers as decision makers and instruction providers)</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Product development mindset (vs project management)</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Customer/user centricity</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Process framework driven</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Time-boxed or capacity controled</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Adaptive planning (vs predictive, fixed plans)</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Continuous improvement</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Empiricism (evidence, fact, data driven management)</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Design driven</td>
</tr>
</tbody>
</table>
## Agile and agility

<table>
<thead>
<tr>
<th>Category</th>
<th>Agile</th>
<th>agility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Learning</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapt and iterate</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Experiment</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fail fast, learn fast</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Perform under uncertainty and ambiguity</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer/user centricity</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Adaptive planning (vs predictive, fixed plans)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Continuous improvement</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Empiricism (evidence, fact, data driven management)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Org</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stakeholder and customer collaboration</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td>Cross-functional teams</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td>Empowered, autonomous, self-organized teams (vs teams with line managers)</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td>Leaders and managers as coaches and facilitators (vs leaders and managers as decision makers and instruction providers)</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product development mindset (vs project management)</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td>Process framework driven</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td>Time-boxed or capacity controled</td>
<td>Yes</td>
<td>Encouraged</td>
</tr>
<tr>
<td>Design driven</td>
<td>Encouraged</td>
<td>Encouraged</td>
</tr>
</tbody>
</table>
Agile Transformation Challenge: Organizational Alignment

Image by: Henrik Kniberg | Concept by: Stephen Bungay “Art of Action”
Addressing the Irony of Delivering Agile Transformation with Waterfall

https://agile-od.com/lean-agile/waterfall-agile
Algorithmic task handling vs Heuristic (trial & error) problem solving

We often confuse Complicated, with Complex

The crux of the problem is, we tend to attempt solving complex problems, algorithmically.
• Lean ≈ Agile
• Design Thinking ∈ Agile
Lean: Lean Manufacturing & Lean Startup

Common Lean values as generally known today

- **Reduction of waste**: Focus on value creating activity and elimination of anything that doesn’t contribute to value
- **Iteration**: Build-measure-learn and continuous improvement
- **Localized activity**: Genchi, Gemba (where the customer is or at the production floor) and allowing product strategy to be determined thorough local experimentation, and the spirit of “get out of the building”

### Lean Manufacturing Terminology

- **Kanban** (かんばん)
- **Muda, Muri, Mura** (むだ、むり、むら): waste
- **Genchi-Genbutsu** (現地現物)
- **Gemba** (現場)
- **Kaizen** (改善): *continuous improvement*
- **Kakushin** (革新)

### Lean Startup Terminology

- **Build, measure, learn**
- **Persevere, tweak or pivot**

Read the summary here:

https://agile-od.com/lean-agility/lean-manufacturing/creating-lean-startup/
Design Thinking is process driven
Design Thinking

**Empathize**

**Define**

**Ideate**

**Prototype**

**Test**

**Process driven doesn’t mean it has to be sequential**

(This is how our brains work)

Creativity in steps

**FEEL**

your customer

**THINK**

what problem to solve

**BUILD**

Your idea into something that can be tested

**IMAGINE**

what you can do to solve the problem

**TEST & MEASURE**

**Lifecycl**

agile-od.com
Hybrid Agile Example

Combining Design Thinking’s strength on ideation, and Scrum’s powerful iterative building

Agile is a high order synthesis of Flexibility and Discipline
About Coach Takeshi
Coach Takeshi

Takeshi Yoshida
Chief Coach and Founder, Lifecycle Pte. Ltd. (Singapore)

1994~2001: Morgan Stanley
2001~2002: INSEAD MBA
2002~2004: Bank of America
2004~2009: Deutsche Bank
2009~2011: Barclays
2011~current: Lifecycle

• Serial startup entrepreneur
• Coach, trainer, facilitator
• INSEAD corporate executive program lecturer, trainer, facilitator

Behavioral Coach
- Positive Psychology
- Conversational Intelligence
- Non-Violent Communication
- Radical Candor
- Psychological Safety
- Appreciative Inquiry
- Choice Theory
- Theory X & Y
- Multipliers and Diminishers
- Habit formation

Process Coach
- Design Thinking
- Agile Scrum
- Lean Startup
- Lean TQM
- OODA
- Liberating Structures
- Lego® Serious Play®
- User Story Mapping
- Design Sprint

Organization Development Professional
- Systems Theory
- Evidence Based Management (EBM)
- Organizational Ambidexterity
- Theory E & O
- Objectives & Key Results (OKR)
- Innovation Accounting

Full bio: https://agile-od.com/takeshi

Professional Certifications

International Association of Positive Psychology Coaches (IAPPC)
Certified Positive Psychology Coach (CPPC) Level II (Credential: https://agile-od.com/cppc)

International Coach Federation (ICF)
Associate Certified Coach (ACC) (Credential: https://coachfederation.org)

Scrum.org
Professional Scrum Master II (PSM II)
Professional Scrum Product Owner (PSPO) (Certifications: https://www.scrum.org/user/498256)

Association of Master Trainers in the LEGO® SERIOUS PLAY® Method
Certified LEGO® SERIOUS PLAY® Facilitator (Certification: https://agile-od.com/lsp)
More readings from Coach Takeshi

- A Pretty Good Summary of Lean, Agile, Scrum
- Lean, Lean Manufacturing, Lean Startup: Explained
- Try Design Thinking + Scrum: A Powerful Hybrid Agile Approach
- Strategy Session Facilitation with Design Thinking + Liberating Structures
- Waterfall Agile: Addressing the Irony of Delivering Agile Transformation with Waterfall
- Knowing to Stop, a Confucius Teaching
- Radical Candor, My Go To Feedback Routine
- How to Get Scrum Right on First Attempt: Single Sprint Scrum Pilot
- Innovation Manager's Toolkit
- Ambidextrous Organizations Explained