

Orchestrating

an Agile Learning Design
and Delivery Process



Table of Contents

Current Situation	3
The Influence of Software Development	4
The Scrum Framework	7
Agile Instructional Design	8
Why Agile Works	11
Using Learning Technology to Bring it All Together	12
Authors and Contributors	14
Rockstar Learning Platform	15

Current Situation

Learning & Development teams have never been asked to do more with less than right now. The pandemic put the squeeze on L&D budgets, but newly remote workforces require more training and learning options to meet the needs of the new normal. Instructional design teams are usually pretty small — if they exist at all — so it can be exceptionally challenging to deliver strong, impactful content under these constraints.

Instructional Design teams need the right strategies and tools to keep pace with the learning demands of today's business climate.

With the right approach, ID can transition from a bottleneck to the

orchestrator of knowledge flows in the organization.

The current organizational learning and development landscape is no longer a predictable contour along which we can chart a path for our learners, but is more like shifting sands, with the constantly changing contours. This requires us to pay attention, respond to changing needs and new opportunities, anticipate where things are going and remain flexible and adaptable.

With organizations rapidly transitioning from classroom to virtual training over the last year, and the expectation that it will continue even as we move through the pandemic, we've seen — and will continue to see — increased demand for training. Instructional design, training and L&D organizations are asked to do more but resources and capacity aren't increasing at the same rate. In some organizations, they aren't increasing at all.

In contrast, with the Agile mindset of software development, scope and requirements unfold and are identified incrementally, and not planned in advance. Time to release is typically two to eight weeks. Change is expected and driven by a continuous feedback loop, which measures how well your product meets end-user needs. Instead of a “final” outcome at the end of the process, there are small intermediate releases that build your total product value cumulatively over time.

- **Agile development is iterative:** The overall process happens in short, repeating cycles of targeting requirements, designing, developing, testing and delivering.
- **Agile development is collaborative:** Customers, end-users and other stakeholders are involved early and throughout the process.
- **Agile development is incremental:** Total value delivered accumulates over time with multiple small releases.

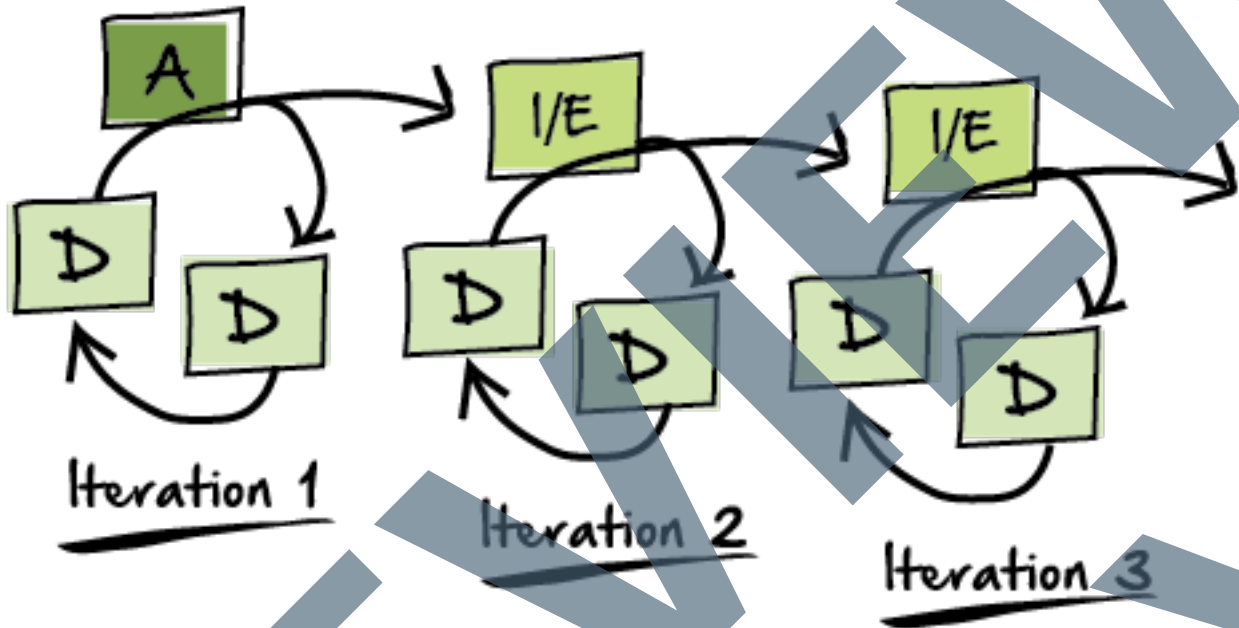
In general, the biggest differences between the two are:

- For **Waterfall**, the plan is defined upfront and in advance. For **Agile**, it unfolds incrementally.
- For **Waterfall**, the schedule is based on a cycle of design and development that may last several months or years. For **Agile**, it's based on a cycle of design and development that repeats every two to eight weeks.
- In **Waterfall**, changes to the plan happen only when there are unforeseen issues, or adjustments are needed to meet hard delivery dates. In **Agile**, adjustments are driven by a constant loop of feedback from stakeholders and customers.
- To manage risk, costs, and schedule commitments, **Waterfall** is more rigid and resistant to change. In **Agile**, change is expected.

Orchestrating an Agile Learning Design and Delivery Process

LLAMA, developed by Megan Torrance, begins with analysis and design in the first iteration and continues with subsequent iterations of implementation, evaluation, design and development.

Agile Applied to Instructional Design: LLAMA



Source: Torrance Learning

Both models — SAM and LLAMA — leverage three reasons in applying Agile principles to instructional design:

The first is to ensure that you're building the right product. The day you begin a project is when you know the least about it. These models allow for learning, adaptation and refinement along the way to ensure you're pursuing and delivering the right target.

The second is to get the design right.

Through successive iterations of design and development. You can validate whether your instruction is usable, desirable and effective — that is, whether it enables learners to meet learning objectives.

The third reason is that the Agile approach provides a way to manage an iterative design and development process.

Why Agile Works



What are the key benefits of being iterative, collaborative and incremental? If your development is iterative, you can deliver value sooner and more frequently to your customers. You can also become very good at a process because you're doing repeatedly. It's like learning to dance with a partner, playing in a band or on a sports team. You become accustomed to working together and can anticipate, and needn't negotiate, every little step of the way.

If your development is collaborative, you have the benefit of continuous feedback from customers, end-users and stakeholders. If your development is incremental, you don't need to prioritize everything upfront, you just have to decide on the next few priorities. If you're designing, developing, testing and releasing changes in small increments, you don't have the same level of risk as when you're doing one big project. It's much easier to detect errors and course-correct as you go.

If you want to bring the power of Agile into your learning design and delivery:

- Commit to an Agile mindset.
- Adopt an Agile process for design and development, such as the Scrum Framework.
- Focus on developing microlearning instead of courses.
- Find ways to enable the direct contribution of knowledge by those who have it to those who need it.
- Reposition your instructional design or training development team as orchestrators of your organization's knowledge flow.