Brandon Hall GROUP

elearning brothers

Orchestrating an Agile Learning Design and Delivery Process

Table of Contents

Current Situation

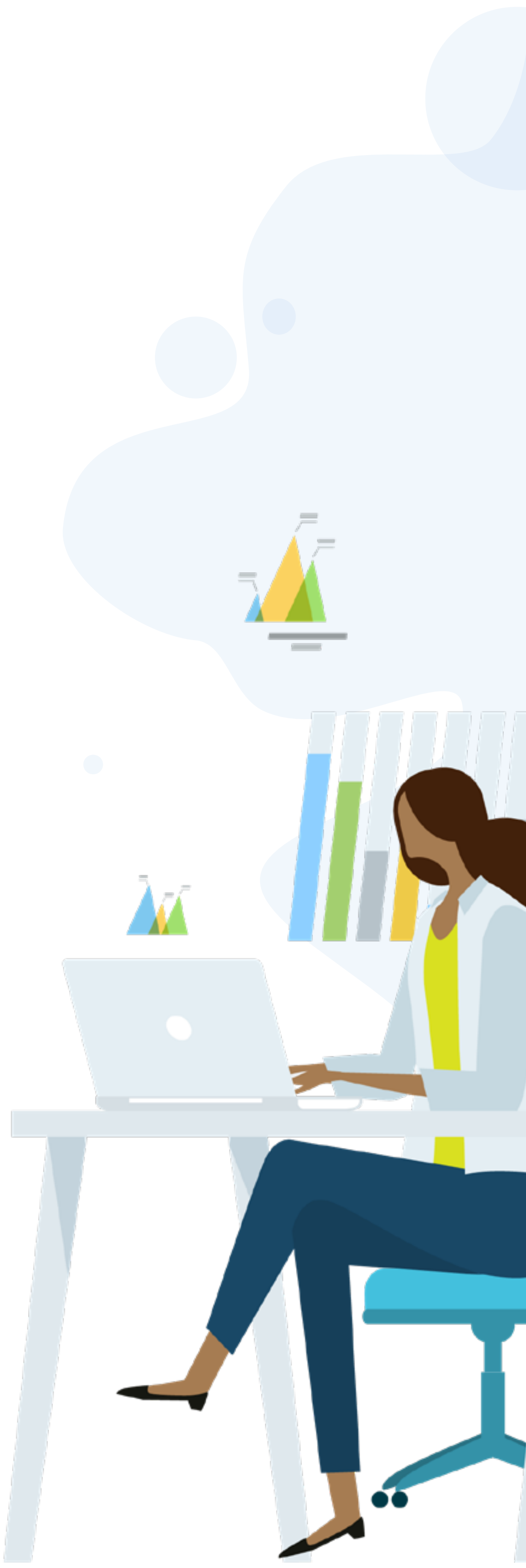
Learning & Development teams have never been asked to do more with less than right now. The pandemic put the squeeze on L&D budgets, but newly remote workforces require more training and learning options to meet the needs of the new normal. Instructional design teams are usually pretty small — if they 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.



The Influence of Software Development

The change in learning content development parallels what happened in the world of software development. About 20 years ago, software development methodology was beginning to change. Development cycles were being measured in years. Many teams began to experiment with emerging ways of developing software, deploying releases every few months, or in some cases, every few weeks.

This was the start of the Agile Software Development movement, captured in February 2001 by the Agile Manifesto. It provided a statement of values to help software development teams to be less focused on the processes, tools, specification documents, contracts and plans that they aren't able to produce the kinds of high-quality solutions through conversation and interactions with others, validation of working software or interactive prototypes, early and continuous customer collaboration and the flexibility to respond to change and opportunities as new information becomes available.

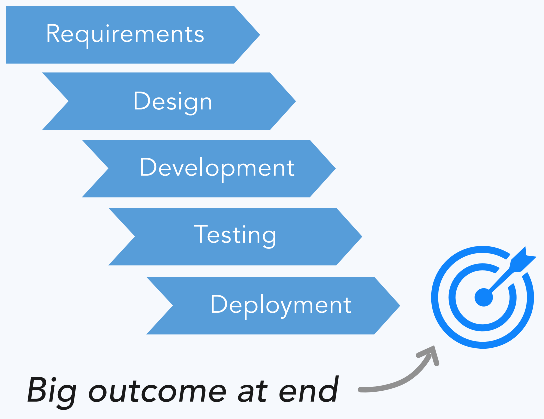
Agile itself is not a method but a mindset. The Agile Manifesto declared 12 principles that reflect this mindset. **These principles include:**

- Satisfying customers through early and continuous delivery of valuable software
- Welcoming changing requirements, even late in development
- Delivering working software frequently, from a couple of weeks to a couple of months
- Businesspeople and developers working together daily
- Motivation and trust
- Face-to-face conversation
- Working software as the primary measure of progress
- Sustainable development at a constant pace
- Continuous attention to technical excellence

What is Agile Software Development?

In a traditional waterfall method of software development, the entire project is scoped and planned upfront, and then output cascades down from one team to the next at each stage in the process. Software development projects using a waterfall approach typically run for several months or years. In general, it's a process that's resistant to change and focused on meeting delivery dates. The result is one big outcome at the end of the process.

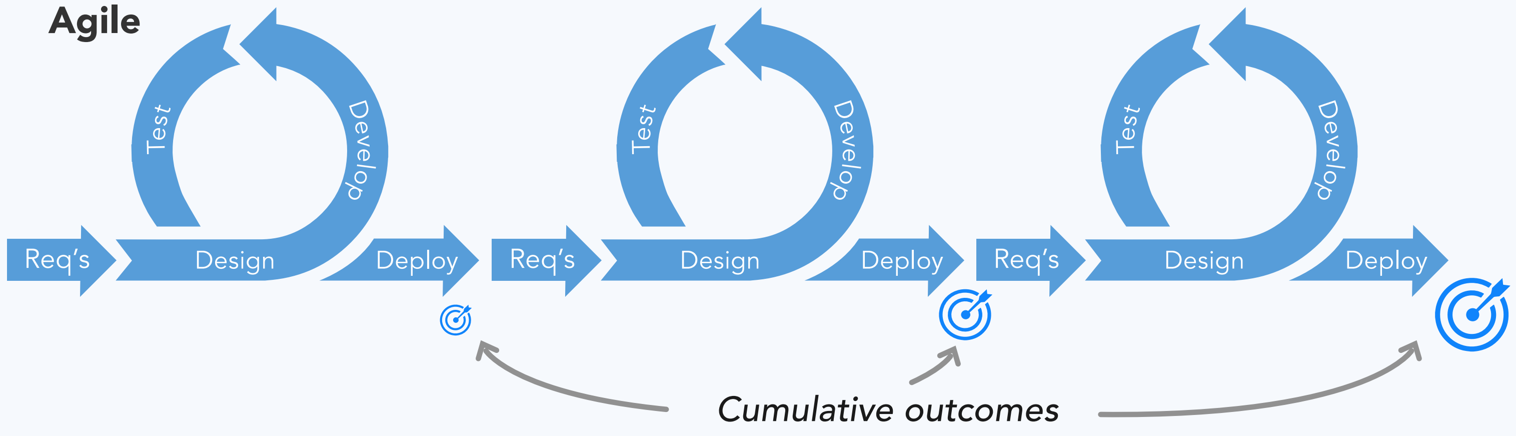
Waterfall



Summary Comparison: Waterfall vs Agile

	Waterfall	Agile
The Plan	Defined upfront in advance	Unfolds incrementally
The Schedule	Several months or years	2 to 8 weeks
Adjustments to the Plan	Issues not seen in advance, adjustments to meet delivery dates	Driven by a constant loop of feedback from stakeholders and customers
Openness to Plan Change	Ridged, Resistant	Expected

Agile



Source: eLearning Brothers

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.

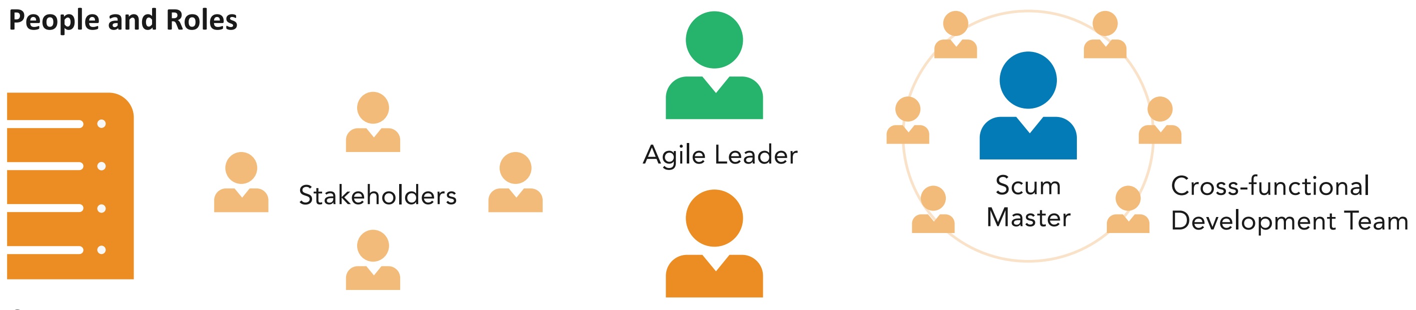
The Scrum Framework

To best understand the concepts of Agile development, it makes sense to look at the Scrum Framework, as it is probably the most common Agile Development Framework. By using Scrum as a foundation, we can apply Agile principles to orchestrating an Agile learning design and delivery process in your organization.

To start, we can take the three project roles described in Scrum and add a fourth:

Product Owner | **Development Team Member** | **Scrum Master** | **Agile Leaders**

People and Roles



Source: eLearning Brothers

Product Owner. The product owner is responsible for determining product strategy and how to invest the capacity of the Scrum team. They alone determine what goes into the product backlog and the order those things are targeted for development. The owner must be empowered to make product decisions and give the team direction.

Scrum Master. Not a master of people, but a master of skills and processes who promotes and supports the use of Scrum, coaches the team on Scrum processes and facilitates team interaction and communication. They should promote a sense of community and take a holistic approach to work.

Scrum Development Team. This is a cross-functional team with all the skills necessary to deliver the work. It typically includes front- and back-end software developers and quality assurance engineers. It can also include UX designers, Instructional Designers, Technical Writers, Customer Success Advocates and others. An ideal size for most scrum teams, to maximize efficiency in process and communication, is seven to eight people.

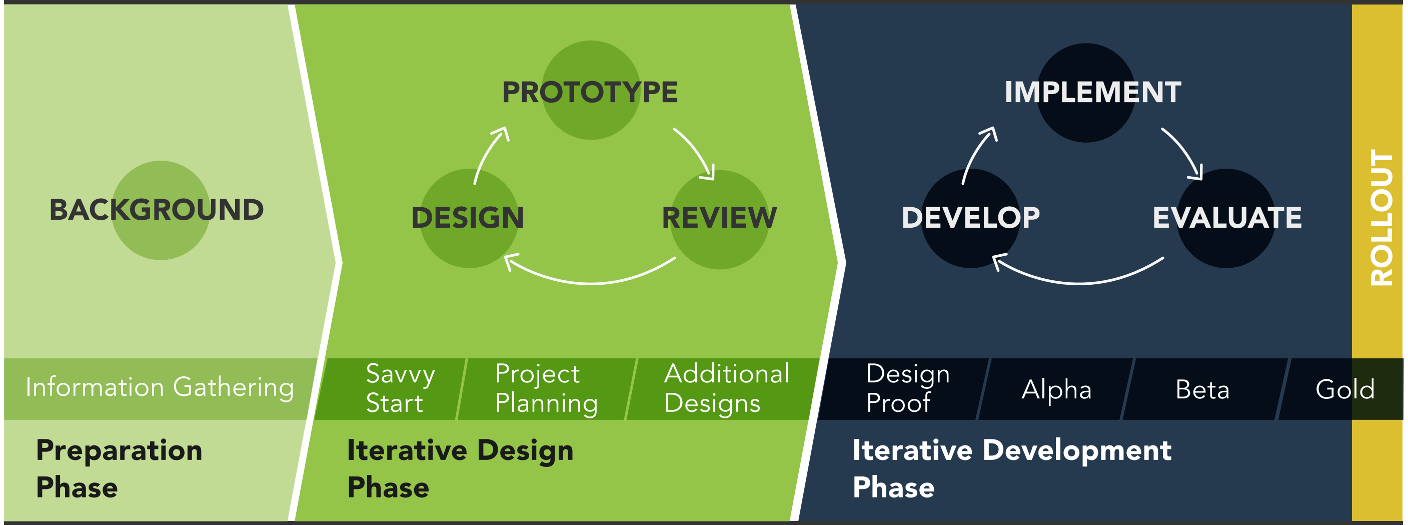
Agile Leaders. These are people not involved in the day-to-day scrum process, but those who create an environment for agility to flourish, managing impediments and problems in the organization and providing coaching and support to Scrum Masters, Product Owners and Teams.

Agile Instructional Design

In terms of instructional design, the traditional ADDIE model (analyze, design, develop, implement, evaluate) is based heavily on the Waterfall software development model. As such, it carries many of the same challenges in light of today's business needs. Instead, organizations need to look to more Agile frameworks so they can apply their principles to developing learning programs.

Two prime examples of Agile alternatives to ADDIE are the Successive Approximation Model (SAM) and the Lot Like Agile Management Approach (LLAMA). Developed by Allen Interactions, SAM has an initial preparation phase followed by two iterative design and development phases that address performance needs through "repeated small steps rather than perfectly executed giant steps" prior to rollout.

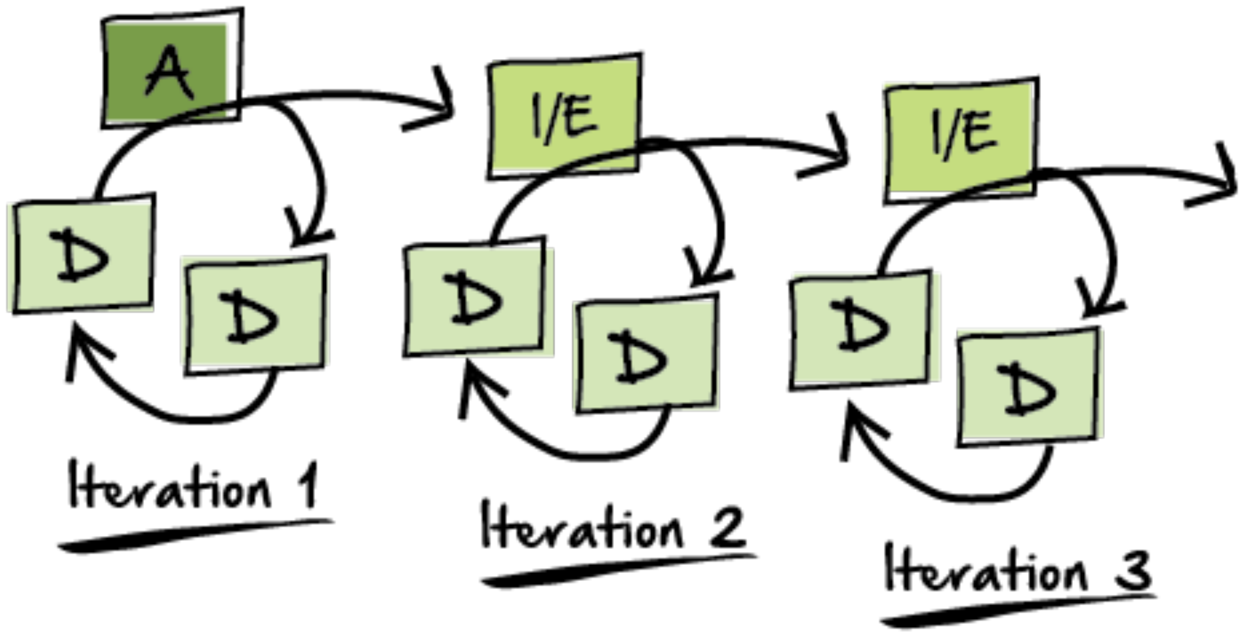
The Successive Approximation Model — AKA: The SAM Model



Source: Allen Interactions

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.

Applying Agile principles to the process of learning design and development increases the potential for delivering continuous value to learners.

Rather than simply an iterative, incremental and collaborative process for completing an instructional design project, learners receive continuous and consistent value over time.

This fits very well with how Agile applies to learning in today's fast-paced, high-tech world. And this isn't something that's come top-down; we haven't taught learners how to do this. This is something that already happens, that learners are doing on their own, that our Learning & Development teams and initiatives are now starting to take advantage of. Learners are naturally employing Agile tactics to their own learning because of the technology that's available to them — and out of necessity.

They don't have large windows of time for training but can consume it when it comes in the form of:

- **Microlearning: Small bites**
- **Just in time. Just enough.**
- **Anytime, anywhere; especially mobile**

It also fits well with our business and organizational development initiatives to create cultures of learning at all levels of the organization and promote peer-to-peer and group-to-group knowledge and skills collaboration and sharing.

Why Agile Works

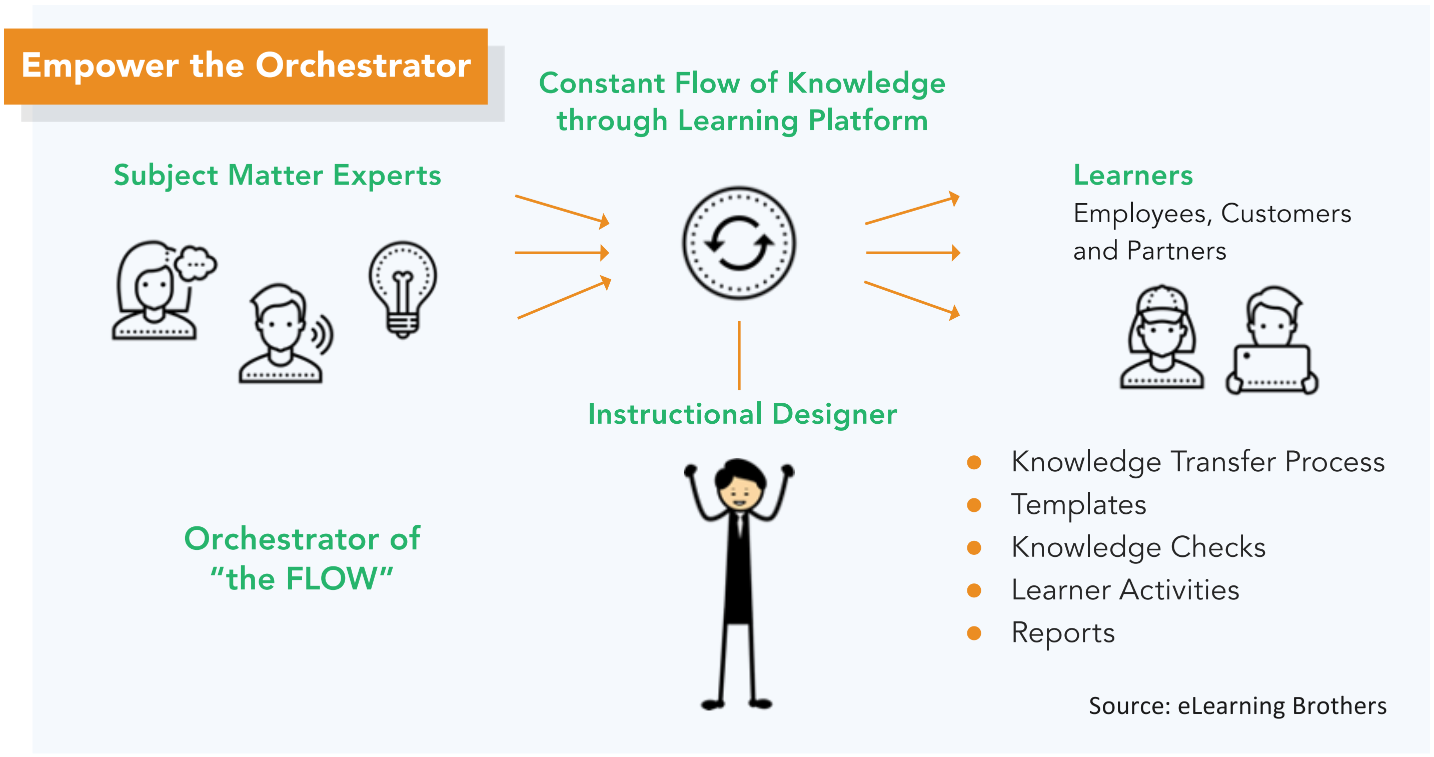
What are the key benefits of being iterative, collaborative and incremental? If your development is iterative, you can deliver value sooner and more frequently to your customers. You can also become very good at a process because you're doing repeatedly. It's like learning to dance with a partner, playing 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.

Using Learning Technology to Bring it All Together


Empower the Orchestrator
Constant Flow of Knowledge through Learning Platform
Subject Matter Experts
Learners
Employees, Customers and Partners
Instructional Designer
Orchestrator of "the FLOW"
● Knowledge Transfer Process
● Templates
● Knowledge Checks
● Learner Activities
● Reports
Source: eLearning Brothers

Businesses need an agile learning system; a system that can move quickly and keep up with organizational needs; technology that allows information to flow among authors, subject matter experts, admins and learners. In short, an agile learning strategy needs:

- Direct SME contribution of all media types
- Ability to share content on any device.
- Multimedia available to all types of learners, (including captioned and eLearning in multiple languages)

For most organizations, the raw material for new content comes from an expanding variety of sources: Zoom meetings, desktop screen recordings, recorded live events, Word documents, PDFs, PowerPoint decks, etc. Without the right tools, instructional designers can quickly become a bottleneck in the flow of knowledge throughout the organization as they try to interview, capture, review, synthesize, organize, format, style and reconstitute subject matter expert knowledge using sophisticated multimedia production tools.

By providing tools that allow SMEs to directly contribute their knowledge into templates and knowledge streams, companies can transform their instructional designers into knowledge-flow orchestrators. Democratizing the creation of content to users means IDs can focus on managing the agile flow of the process.

It is often said that technology cannot fix a bad process. But if your organization has its sights set on developing an agile process for content creation and delivery, it will need the right technology to make it happen.

Authors and Contributors

David Wentworth (david.wentworth@brandonhall.com) co-wrote this eBook. He is Principal Learning Analyst at Brandon Hall Group, focusing on all aspects of learning and the technology that supports it. David has been in the human capital field since 2005 and joined Brandon Hall Group as senior learning analyst in early 2012.

Dr. Christian J. Weibell, Ph.D. co-wrote this eBook. With a B.S. in Computer Science, a Ph.D. in Instructional Psychology and Technology, and 25 years in high-tech software development Christian provides a very passionate but informed perspective to envision, define and drive technology-enabled innovation in learning and development. He is most widely known for his Principles-of-Learning Framework that articulates seven principles on which all learning is based, and provides a foundation for the development of domain-specific theories of learning (principlesoflearning.org). This framework and its background research currently serve as reference to thousands of unique visitors each week, from all seven continents and nearly every country in the world.

Jon Tota contributed to this eBook. He is Rockstar Learning Evangelist at eLearning Brothers. Jon began his career in financial services as a technology trainer for PaineWebber and UBS before co-founding Edulence in 2002 and creating KnowledgeLink as one of the first subscription-based online training services. Under Jon's leadership, the team at Edulence innovated the LMS model and scaled Knowledgelink to serve several hundred thousand annual users in multiple verticals. In 2020, Edulence was acquired by eLearning Brothers and Knowledgelink has evolved into the new Rockstar Learning Platform, offering the latest innovations in learning experience. Jon now serves as the company's Rockstar Learning Evangelist, speaking on learning experience design topics and educating customers on how to leverage eLearning Brothers solutions to create eLearning experiences that rock.

TOP LMS for Microlearning 2021 Featured on elearningindustry.com

TOP LMS For Product Training 2021 Featured on elearningindustry.com

CUSTOMER EXCELLENCE PLEDGE CWG

Training magazine Network choice awards 2020 Learning Portal/LMS

2020 WATCH LIST COMPANY LEARNING MANAGEMENT SYSTEM TRAINING INDUSTRY

TOP Mobile Learning Platforms 2020 Featured on elearningindustry.com

COLLABORATIVE. INTERACTIVE. AND PROVEN TO WORK.

RockstarLearning

PLATFORM

LEARNING MORE »