How Developer teams can use Agile learning to close the skill gap

stackoverflow
FOR TEAMS

In this eBook, we will show you how to reduce the skill gap by creating an upskilling and reskilling program for your teams via the Agile learning process.

Stack Overflow for Teams can help by creating a culture centered around knowledge sharing and collaboration. People say they want a learning culture but don't actually set up the framework to breathe life into knowledge sharing. Real change comes from tools, intentional strategies, and an appetite for transformation.

Leverage these features of Stack Overflow for Teams to enhance your educational program:

- Strong s**earch functionality** and QA format for on-the-job learning and development
- Effortless **async collaboration** between peers and across teams, allowing for mentorship and upskilling
- Knowledge **shared and discovered** through simple and intuitive content organization, proactive alerts, and transparent information for all employees
- **Content health** maintained by many instead of one and gamification that encourages enrichment
- **Reduced time** that your lead developers spend answering duplicate questions, and increased time spent on reskilling and formal mentorship

Stack Overflow for Teams brings many functions forward to enhance your learning and development program.

Introduction

Currently, there are more digital and technical job openings than there are people qualified to fill them. Companies that shift their focus from hiring to upskilling or reskilling have a 27% higher likelihood of filling gaps.[1]

While hiring can be costly and lead to continued delays in development and innovation, employee upskilling /reskilling programs are cost-effective and impactful. When the hiring pool lacks qualified digital talent, companies can uplevel talent through their own programs.

To make upskilling work, company executives must invest in the creation of an organizational learning culture. Traditional models of education at work have been expensive, time-consuming, and ineffectual. Given those factors, executives may hesitate to invest in workforce education programs. They need proof that learning programs have positive impacts on the bottom line. Burst-effort learning and other Agile methods are results-driven. They can prove the ROI that executives need to justify the upfront spend.

Agile learning was designed to meet the rapidly changing demands of today's digital workplace. A properly-implemented Agile learning program can complement a busy digital workforce in ways that traditional learning modules cannot.

[1] https://qz.com/1213768/the-forgetting-curve-explains-why-humans-struggle-to-memorize/

What is Agile learning and why is it successful?

Agile learning is a training approach based on the Agile software development methodology. As with Agile software development, Agile learning is focused on learning and experimenting in brief sprints, with frequent collaboration and regrouping to optimize skill acquisition.

German psychologist Hermann Ebbinghaus' "forgetting curve" research shows that under typical training models, like seminars paired with coursework, people forget 50% of presented information within an hour. Within 24 hours, they've forgotten 70% of new information, and within a week, they've forgotten 90%.[1] Clearly, this is an inefficient use of time and money for an organization: employees spend a large chunk of time on training instead of actively working on projects, and when the training is complete, they're only likely to remember a fraction of what they learned.

In contrast to traditional learning methods, which are largely passive, Agile learning focuses on a mix of learning and doing, enabling employees to discuss what they learn and put it into practice immediately. As a result, they're able to build neural pathways to retain the knowledge and overcome the forgetting curve. And because so much of Agile learning is based in practice, teams focus on working towards project-oriented goals as they learn new skills.

Instead of taking part in months-long coursework, Agile learning focuses on short, just-in-time microbursts of learning, whether delivered through a Learning Management System or through paired mentors. The learning is tied to a desired outcome—i.e., gaining the necessary knowledge of a programming language to build a new product feature.

[1] https://qz.com/1213768/the-forgetting-curve-explains-why-humans-struggle-to-memorize/

Agile learning is built on continuous communication and collaboration. Every organization will approach Agile learning differently, but most will adopt some form of the daily “scrum” meeting, where everyone shares updates and roadblocks. This can be done in a live meeting or video chat, or it may occur in a designated channel on Slack or another chat tool. The goal is to ensure access to content and support that the team needs to skill up.

Dr. W. Warner Burke of Columbia University breaks down agility in learning into nine dimensions:[2]

Flexibility
Open to new concepts

Speed
Taking quick actions

Experimenting
Trying new ways of doing things

Performance
Seeking out challenges

Interpersonal risk-taking
Open to constructive criticism

Collaborating
open to ideas from others

Information gathering
Takes initiative to seek out information

Feedback seeking
Appreciates the opportunity to improve

Reflecting
Thinks about how to apply improvements to future challenges

Agile learning is a hands-on, embedded learning method in which participants are encouraged to challenge themselves and their peers. Mistakes are welcomed as part of the learning process. For organizations that embrace Agile, it’s proven to be highly successful. Embedded learning shows 9.9x the results of traditional learning, and enterprises that devote time for employees to learn through Agile methods demonstrate 7.2x the impact of those that don’t.

Just as you should build analysis and reflection into product development, so should you assess roadblocks and achievements at each stage of Agile learning. Learners are encouraged to ask and answer questions, collaborate within the team, and reflect on problems and progress. Agile learning is about embracing curiosity and collaboration, empowering employees to teach and learn from one another in practical, on-the-job scenarios.

[2] https://www.talent-quarterly.com/where-are-the-Agile-leaders/

Why Agile learning matters

Traditionally, companies approached learning through a mix of strategies: namely, passive modules that cannot measure engagement or knowledge retention, like in-person workshops or webinars. The ROI has been assumed, and not proven through achieved metrics or data analysis. This approach is flawed. Today's employees don't have time to focus solely on training—they need the opportunity to learn on the job.

Agile learning can solve the experience training gap. 98% of Agile learning respondents reported satisfaction with their training and learning, including 54% very satisfied, as compared to other learning methodologies. Only 70% of respondents who used other learning methods were satisfied, with just 13% very satisfied.

Social engagement is a large component of Agile learning. That means learning takes place within a community, and co-learners are encouraged to share, collaborate, and mentor each other. Social learning substantially improves outcomes: it is 4.3 times as impactful as solitary learning.

In Agile learning, learning is embedded continuously into the flow of work, ensuring that employees have the chance to apply the new skills they learn. This organic application results in a significantly higher skill adoption rate than under reactive learning methods, where only 37% of new skills learned are applied in practice.

To help your employees gain new skills and quickly put them into practice, an Agile learning methodology is crucial for closing the talent gap.

How to get started

Follow these guidelines for using Agile learning to close the skills gap:

Understand your starting point

When you're looking for opportunities to upskill your workforce, assess the common skillsets already within the company. Have employees take skill assessments and approach managers for feedback on the largest skill gaps. Look at skills that are adjacent to or slightly more advanced than those your team members already have, so that upleveling feels like a natural progression.

Set your goals

As you begin your Agile learning initiative, set a clear path with benchmarks for the outcomes you'd like to see. For example, if 40% of your development team demonstrates proficiency in Kotlin, can you grow that number to 60% by partnering your Kotlin developers with those who don't yet have that skill. Provide opportunities to measure skills learned both immediately upon mastering them, as well as several months later to ensure knowledge retention.

[2]Survey Analysis: Agile Learning Delivers Superior Outcomes, Gartner Jan 15 2021

Focus on dynamic skilling

Use a “dynamic skills” approach to help your organization pivot quickly as skill needs change. This includes building cross-organizational networks to encourage collaborations, implementing skills accelerators (using existing resources and in-house expertise, rather than formal training), and providing a platform for open communication to help employees uplevel one another’s skills.

Implement microlearning

Microlearning consists of short-form content, typically between three and six minutes in length. The content is short enough that employees should be able to actually retain the information they digest in each hit, and put it to use immediately. Incorporate a mix of media, including short videos, articles, and audio recordings, that employees can consume from their mobile devices at their own convenience.

Create a “backlog”

While working on a project, the product owner should create a backlog of all of the skill-building needs that your team should adopt over the course of the development cycle, prioritized by order of importance. Break the backlog down by individual employee needs. Maybe several of your team members already have some experience working in Elixir, for example, and others that are starting from scratch. They’ll need different progressions through the microlearning coursework, and different expectations of the level of mastery they’ll show at the end of the project. Those who have previous experience can support the teammates who are new to it by conducting code reviews, sharing their own work, and answering questions in your knowledge-sharing tool.

Use the right tools for collaboration

Your knowledge-sharing platform will be an ideal hub for collaboration and feedback-gathering, with peers able to asynchronous work together or provide feedback to each other.

Put knowledge reuse at the core

Make this knowledge sharing and collaboration content archived and searchable so your development team will be able to use these discussions as a valuable resource for years to come. Because information only needs to be shared once and can be accessed many times, you are scaling the impact of the knowledge being gained over time.

Conclusion

As organizations embark on their digital transformations, Agile practices have a vital role to play. The majority of organizations have already shifted to an Agile approach in DevOps, but Agile hasn't necessarily become a routine best practice in other areas, learning and development included.

KnowledgeHut's 2020 report, The State of Agile, showcases that while only 5% of respondents said that none of their teams were Agile, widespread adoption was mixed: 44% said that less than half of their teams were Agile; 33% said more than half; and only 18% said that all of their teams were Agile.

Organizations often have difficulties scaling Agile practices due to a range of factors, including general organizational resistance to change (48%), not enough leadership participation (46%), inconsistent processes and practices across teams (45%), and minimal collaboration and knowledge-sharing (22%).

Through Agile learning with a knowledge sharing and collaboration platform to support it, your team will upskill and reskill quickly. Rather than relying on a top-down leadership approach, knowledge-sharing helps build a flat hierarchy, in which ideas and feedback are shared organization-wide.

A supportive Agile learning structure incorporates flexible microlearning opportunities; including projects that enable teams to test and optimize their newfound skills. Your team is likely already incorporating Agile work methodology into their daily routine. They deserve Agile learning and development opportunities as well.

[2] https://www.knowledgehut.com/blog/Agile/state-of-Agile

stack overflow
FOR TEAMS

To learn more visit
stackoverflow.com/teams