

Developing a hands-on agile coaching program to accelerate digital innovation

Discover the short and long-term benefits of learning agile while working on business-critical development projects





Agile is not new

Given the urgency to accelerate digital innovation, IT leaders know they need to successfully adopt agile. In fact, Gartner Research reported in 2017 that agile, iterative or lean methodologies are now used by 57 percent of software development teams.* Without question, agile practices enable faster, better software development. The resulting software helps businesses of all sizes drive innovation and fulfill the needs of both the enterprise and customers.

However, the most common stumbling block to adopting and practicing agile is organizational culture and its resistance to change. To succeed in agile adoption, IT leaders need to train their organizations and reinforce agile values, key practices and create communities of practice, while delivering business outcomes. This is easier said than done.

To effectively institute cultural change, companies need to develop comprehensive hands-on agile coaching programs to solve real-world projects.

Why?

Because agile, from learning to adoption to practice, isn't a one-and-done exercise. It's an ongoing process.

*Market Guide for Agile and DevOps Services, Gartner Research, Oct 2017

This shift in how to learn/implement agile will benefit your organization in the following ways:

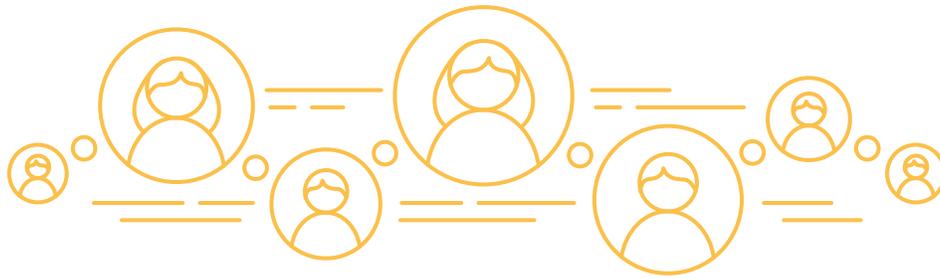
- Increases engagement and better understanding of agile concepts and implementation strategies
- Maximizes participation and buy-in through active, rather than passive, learning
- Faster and deeper adoption of agile practices, which leads to better software outcomes sooner
- Creates dedicated change agents who help spread agile best practices throughout an organization
- Provides more immediate ROI and reduces opportunity costs as training takes place on real-world projects, not theoretical textbook examples



This whitepaper will provide you a roadmap for agile adoption through a hands-on coaching approach, including:

- Examination of current training practices
- Theory behind the importance of hands-on coaching and learning
- How to start a hands-on agile coaching program
- Metrics to track whether training is working and taking hold
- Short and long-term benefits of such a program

If you have any questions while reading or would like to talk through some actual case studies of these techniques, please contact us at info@surgeforward.com.



Current agile training practices

When many companies get serious about adopting and implementing agile software development methodologies, they turn to an outside trainer or coach to teach the basics. These trainings often occur in a limited classroom environment for one or two days, and repeat as needed for additional employees.

These blanket trainings take people and organizations with zero or limited exposure to agile concepts and bombard them with what is agile, what roles will they be responsible for and how everything and everyone is supposed to come together. The trainer then walks away, and the newly trained employees are just expected to figure out the rest on their own.

Imagine learning how to play baseball this way. You assemble nine people in a room, spend eight or 16 hours reviewing the current Major League Baseball rulebook and then tell the “team” to go out and have a winning season.

There is no chance that would work. As soon as the team leaves the safe, theoretical classroom environment, they will immediately have both technical questions, “What’s the infield fly rule again?” and interpersonal ones, “What if I think I should bat third?”

Training like this is purely theoretical. It doesn’t understand that agile adoption is a marathon, not a sprint (no pun intended).

Another common method for teaching agile goes a step further and incorporates some hands-on work. After employees receive classroom training, they work together on a conceptual project to put their skills to the test.

But this project isn’t real. It doesn’t affect the business. So, the time and money spent only returns additional conceptual knowledge, not direct business value.

A better way is to take this hands-on learning a step further and have teams learn agile while completing a project that will directly and immediately impact the business.

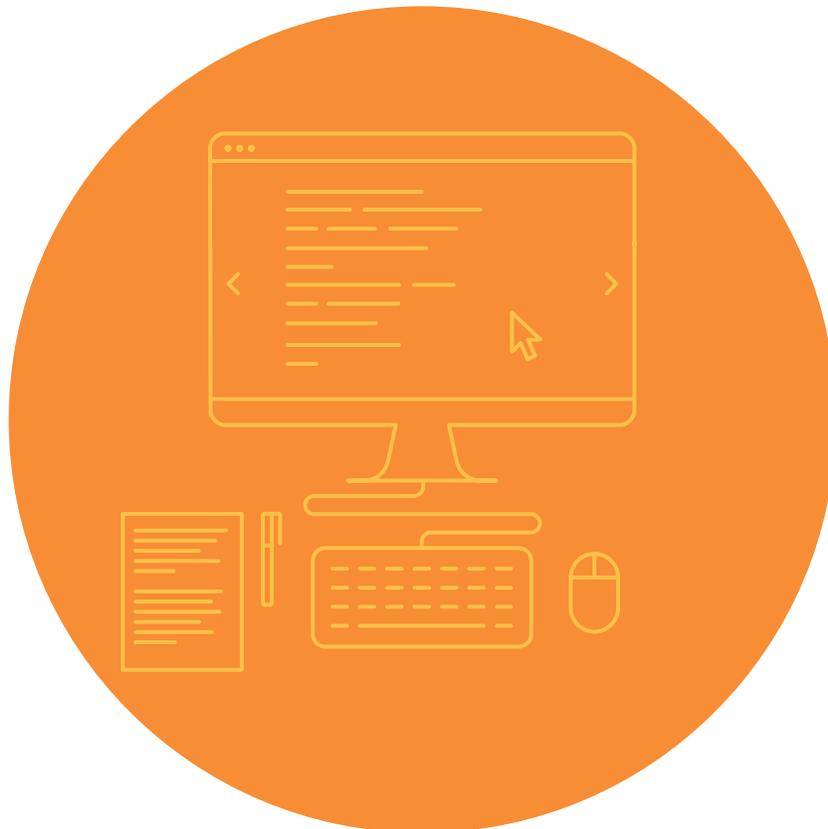
Importance of hands-on learning

What do we mean when we say, “hands-on learning?” A simple definition is that hands-on learning is learning by doing. In other words, hands-on learning actively encourages students or trainees to do something in order to learn about it.

For agile, this means moving beyond a text book or a classroom discussion of the Agile Manifesto. It means actually allowing trainees to self-select teams, lead or participate in group standups, write real user stories and create a ranked backlog, write code and review their peers’ output.

The benefits of hands-on learning are grounded in years of pedagogical research. In fact, a recent University of Chicago study found that students who participated in hands-on learning did better on assessments than students who learned in a “traditional” environment.

It seems the adage, “Get your hands dirty,” is correct. Learning by your own actions, rather than just observing someone else, fosters a natural curiosity and intrinsic motivators to learn, adopt and apply knowledge. The result is that hands-on learning produces better outcomes.



Participants in hands-on learning learn faster, retain more knowledge, have greater ability to train others and are more connected to and invested in what they are learning. They are actively involved in the learning process, instead of just passively receiving information. And hands-on learning provides an authentic context to which people can apply targeted skills or knowledge. This becomes more important when selecting real-world projects, something we will discuss later.



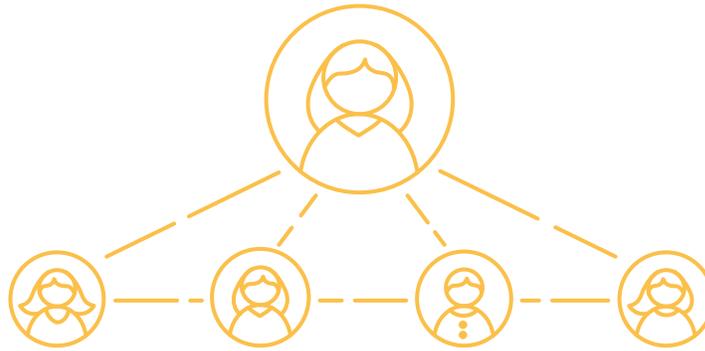
Participants in hands-on learning learn faster, retain more knowledge, have greater ability to train others and are more connected to and invested in what they are learning.

In a hands-on learning environment, feedback is better, faster and more targeted. You can see results play out in real time.

Assessments are also more accurate. Just because you can pass a quiz doesn't mean you can perform that skill or do that job in context. You might know how to lead a sprint planning meeting, but can you do it under the pressure of real-world deadlines with peer feedback and possible opposition? Hands-on learning clearly shows who can perform and who can't.

Hands-on learning helps transfer and refine "soft skills." Tech skills can be taught in a class. But the human interactions, the team part of an agile team, are built by doing. Team management, support, trust and cooperation are interactive learning experiences. Incorporating them early in an agile coaching program develops effective on-the-job habits from the very start.





How to start a hands-on agile coaching program

Before you dive headfirst into an agile coaching program, there are several decisions to make that will determine the success or failure of the program. These include: selecting an agile coach, selecting a team for coaching, selecting an agile pilot project and what key metrics to track.

Selecting an agile coach

One of the first things when starting a hands-on agile coaching program is selecting a coach. It's important to find a person who is willing to take on this crucial role. Someone who doesn't want to coach won't be effective.

This person also must possess the skills needed to be an agile coach. It's hard to be an agile coach if you've never learned or experienced agile.

A coach also needs a certain level of autonomy and authority. They will need to direct or correct trainees, some of whom might be equal or superior in the organization.

“

It's hard to be an agile coach if you've never learned or experienced agile.

This is fraught with internal politics. Who will be the final decision maker? Can a boss overrule the coach just because they don't agree with a lesson or corrective note?

Is a person willing to move into the coaching role on a temporary basis? The whole goal of an agile coach is to put themselves out of a job. Where does the person go after coaching is done?

Given these impediments, it's often more efficient and productive to bring in an outside coach. This person has the willingness and skills to provide effective coaching, the autonomy and mandate to implement agile using qualified best practices and the ability to offer real and honest feedback.

When selecting an external agile coach, there are several questions to consider:



Are they a subject matter expert for both technical and soft skills? The knowledge they possess, and will transfer to your organization, is the lynchpin of future agile success. They need to be an expert on agile practices and why those practices work. Articulating the “why” drives self-organizing and adaptive behaviors.



Have they implemented agile in other real-world environments? They should have broad real-world experience and speak to the specifics of how things fail, and how they’ve solved these challenges with team members and organizational stakeholders.



Do they display high self-awareness? A successful coach knows what they don’t know and is willing to bring additional resources to bear if needed to benefit the trainees.



Are they enthusiastic and passionate about coaching? We’ve all sat through classes with teachers we could tell didn’t care. This degrades the learning experience for everyone involved.



Are they compassionate, have strong communications and active listening skills? Great coaches pick up on subtleties of their students and can tailor teaching to specific learning styles. They make complex topics easy to understand.



Do they have strong interpersonal skills? In some organizations, there is a strong resistance to change. A coach needs to be able to both teach the subject matter and overcome roadblocks to agile adoption. This takes a deft touch that only a few possess.



Selecting a team for agile coaching

After selecting a coach, the next process is choosing who to train. Who will be on your first agile team?

While the answer to this might be everyone who needs to know agile, be careful. Agile works best in a small, cooperative team environment. You'll get to company-wide adoption eventually, but think of this first training as a pilot program on which you can build.

To build a strong agile foundation, carefully select members of this first pilot team. They should be respected in your organization, be open-minded and want to work towards change. They will be your strongest allies in facilitating company-wide adoption of agile.



Mixed teams of internal and external developers help to build project buy-in, decrease disruption and lead to better outcomes.

You should also consider having a team comprised of internal agile novices and external agile veterans who already have real-world experience. Even with the best coach, bringing in veteran agile practitioners will help ease new team members into the process.

Mixed teams of internal and external developers help to build project buy-in, decrease disruption and lead to better outcomes. Co-programming takes the full burden of failure off internal teams. They are now free to both learn agile and produce great software without the worry of being blamed for trying. They have additional support on which to rely.

Combined teams also increase the number of teachers beyond one coach. More eyes in more places provides for more instant feedback. There's also a greater level of trust that's built as new trainees see the external developers as having "skin in the game." It's not just a coach telling you what to do, but fellow developers helping achieve something great.

Selecting an agile pilot project

Once you have a coach and mixed team ready, it's time to select a first pilot project. As stated, this is where the real benefits of hands-on agile coaching occur. You learn while producing software that fulfills a critical business objective.

Selecting that first project can be tricky. You want to make an immediate and substantive impact, but don't want to over shoot and go too big, too soon.

Select a project that's big enough to get attention from key decision makers, but not so big that failure risks too much. You want to show success from the start, increase confidence in the agile process and propel its company-wide adoption.



One critical element of any successful agile project is having a single product owner or manager. This person can communicate the product's vision and represent the views of several stakeholders. The coach can help select and train this role as well.

An innovation project that can add immediate value but not cause disruption to current systems, infrastructure or architectures often fits these criteria. It should be self-contained, last four-to-nine months and a single team of four-to-six people should be able to complete it successfully.

Once the team starts the project, the coach and agile veterans are there to see it to completion using agile methodology best practices. As part of the training/learning, the coach will: shadow meetings and sprint rituals, provide constant feedback, track information over time, observe and make recommendations for improvement, create a mini-backlog of improvements the team can make, make status reports on a regular basis and rank opportunities of ways to do things better.

Tracking key metrics

How do you know if the hands-on coaching and pilot project are working? Throughout the project, you should track and review key metrics which will help you better understand how quickly and thoroughly the team is learning and implementing agile methodologies, or where the training can be improved.

These key metrics include:



Velocity over time

The velocity, or amount of work completed, of an agile team is an important long-term metric to track and measure. Getting up to speed and ramping quickly at the start of a project shows other business units that IT can move at the speed required.

Given that this is the team's first project, however, it's more important to gauge the ability to sustain a consistent production level over a period of time. Velocity should increase over the first three to five sprints. After that time, the team should settle into a consistent rhythm and velocity should normalize at a sustainable level.



Expectation management

This measures points committed against points accepted per sprint over time. It is important to help manage the relationships with other business lines and to show that the team can do what they commit to.

Expect this ratio to be unstable early in the coaching process, but insist that it evolves to consistently be 100 percent. This means the team must learn to make sensible commitments and reliably deliver on them. If they can't, then something is wrong.

Missed commitments have many root causes, but the most common are that requirements are poor, incomplete or are changing mid-sprint, the team doesn't utilize stable, reliable practices and processes, the team is being overly optimistic, the team isn't enabled to succeed or the team isn't considering all the tasks required to get to the full definition of done.



Creation and maintenance of backlog

This includes the number of points worth of estimated, validated and development-ready stories in the backlog. Monitoring this metric can help you understand to what degree requirements are well-managed and if the product owner/manager is successfully completing feature definition, including acceptance criteria, and keeping sufficiently ahead of the team.

After three to five sprints, the team should maintain this at two sprints worth of development-ready backlog, meaning it reaches three sprints worth just before planning day and returns to two sprints worth when a new sprint commitment is made. This enables better predictability of what the team can deliver in the near term, provides a clearer picture of what's ahead and how to approach design and implementation and allows the team and business to have informed conversations on prioritization and potential tradeoffs.



Stories added, subtracted or modified to sprint backlog during sprint

Over time, this number should decline to zero or as close to it as possible. An enabled agile team treats sprints as sacred. Once planning day is complete, the team focuses on delivering precisely to that commitment without interference or distraction.

A team risks lowering its velocity considerably if it changes cards mid-sprint. This leads to extra time spent negotiating change, redoing work and fixing integration points.

Disrupting sprints is an easy way to hurt team morale. Developers work best when they are enabled to deliver, meet their commitments and show off just how much they've accomplished at the end of a sprint.



Test coverage

While not tied directly to agile coaching or adoption, test coverage should be a focus for any software project. This includes unit test coverage and the number of automated versus manual tests. Unit test coverage should be as high as possible, at least above 80 percent. Otherwise, the team may be making quality sacrifices in order to hit commitments. The suite of automated tests should expand and manual tests eliminated where possible.

Other metrics used to measure if you're realizing the full benefits of agile are:



Cycle time

The amount of time it takes individual work products (stories, defect fixes, etc.) to go from “in progress” to “done.” Faster cycle times indicate teams are completing work at agile speeds.



Lead time

The amount of time it takes for the team to fulfill a requirement from the time it's written, or fix a defect from the time it's discovered. This shows how quickly value is realized.



Defect Rate

Bugs should be found sooner with agile and fewer will leak out and make their way into production.

Benefits and outcomes of hands-on agile coaching

What can you expect to get from hand-on agile coaching? What are the benefits and outcomes that make it an indispensable part of a complete digital transformation?

The most obvious outcome is that you have a full team that is agile trained and experienced. They can now create software in accordance to agile development methodologies. This is critical to keep pace with the acceleration of markets and customer demands.

You also have something that classroom-only or conceptual hands-on training does not offer: working software with tangible business benefits.

With this working software, you can see immediate ROI on your training. And you have a powerful tool that proves to management that agile works. You've solved a critical business problem or created an innovative new product with agile. This proof can help sell adoption throughout your organization.

Hands-on coaching creates a beachhead to introduce agile into the larger organization. You have a case study and evidence that it works. It also shepherds the natural emergence of early adopters, champions and change agents who can carry the torch of agile forward, and help sell it to others.

Conversely, it identifies detractors, foot-draggers, laggards and those who are roadblocks for adoption. You can focus on winning them over at an early stage, rather than having to deal with them undermining adoption at a critical point in the future.

On the technical side, the continued adoption of agile will help achieve faster release cycles and rapid development. From prototyping, to proof of concepts or minimally viable products, agile allows you to innovate and fail faster. You can discover what works and capitalize on it before competitors. Or, you can find what doesn't work and kill it before it hurts the organization or sucks up unwarranted amounts of time and money.

Bi-modal development is made possible with agile. Small teams can move innovative products forward, or create new ones, while legacy systems continue to run and evolve in the background. You gain ability to capitalize on innovation work without disrupting back office structure.

Lastly, hand-on learning improves organizational culture. The collaborative effort that goes into creating software with agile methodologies leads to better communication between individuals and departments. IT can prove to business units that it is responsive to needs and can be a trusted partner. And individuals will emerge to head teams, creating a new group of leaders.

Key takeaways

Hands-on agile learning benefits organizations in many ways. To start the process:

- Select an experienced coach who has real-world implementation and training experience
- Create a mixed team of new learners and experienced practitioners
- Select an innovative software project that will serve as the foundation to your hands-on training
- Leverage the successful outcome of the project to expand agile development throughout your organization

With this type of training, you will see both immediate and long-term ROI. The benefits are:

- Working, valuable software
- Positive culture shift
- Faster, more responsive development organization
- Increased competitiveness and ability to respond to rapidly changing market and customer demands

Investing in agile training is a necessary step on the path to digital transformation. A hands-on approach to that training will be a powerful accelerator of that journey, helping your organization and people transform faster and produce better outcomes.

About Surge – a Catalyte company

Surge's highly vetted network of onshore software engineering resources quickly integrate into your organization to help you address immediate and continuing initiatives. With no minimum term or commitment, you have the flexibility to scale up/down on demand. Our qualified engineers, QA specialists, project managers, UI/UX experts and full agile teams are ready to start immediately.

Surge operates six development centers across four time zones. Combined with our network of remote technical resources, our national footprint reduces management oversight and costs, allowing for smoother software delivery and team dynamics.

To speak with us about scaling agile in your organization, contact us at info@surgeforward.com

