

From Practitioner to Coach

Aman King, Lead Consultant, ThoughtWorks
15 December 2013

Here is a glimpse of my personal journey from Practitioner to Coach. An initial narrative highlights my first coaching assignment, followed by takeaways. I present the narrative in a [Three Act Structure](#).

The First Assignment

Act I: The Exposition

An organization is struggling to keep up with a rapidly changing market, failing to retain its competitive edge. Engineering teams are slow in making changes, and introduce additional defects in their race against time. They have tried going Agile but without the backing of strong engineering practices. The organization seeks out my company to help with Extreme Programming adoption as a possible remedy to their problems. Happily unaware of these developments, I am busy wrapping up my current project assignment. Having spent all of my career as a developer in Agile teams, with no formal exposure to coaching, I never imagine what is coming next! As destiny has it, my dreams of a vacation are dashed by Staffing: “Aman, you’re set to play Agile Coach on your next gig! You start on Monday at the client location!”

Act II: The Rising Action

First day with the client team. As I am being introduced, I think back to the previous day. My company’s Consulting Head is speaking, “TDD, Refactoring is all good. But focus on Business Agility. At the end, we can’t simply say we taught TDD. They’ll ask how it helps their business --” My thoughts are interrupted: “Aman, why don’t you introduce yourself?” Words from our Sales guy flash in front of me, “Remember you’re the *expert* consultant! Introduce yourself accordingly.”

The week progresses. “Try doing a process mapping. Did you capture baseline metrics? We need an assessment report”, the Consulting Head tries to be helpful again. “Well, I did talk to team members about how they do their work. Is that what you mean? I downloaded some coverage reports from Jenkins. Umm, what does an assessment report look like?”

Second week starts. How long do I continue observing? What about my *expert* recommendations? Where do I start? The retro was lifeless. Estimation was about shouting numbers. And don’t even talk about the code base! But I am not the only one with questions. The developers, QAs, Scrum Master, Product Owner, everyone has a question or statement to make; even the client management feels like giving directions...

“What is your advice on our stand ups?” “Why write a test first when I’m confident of my code?” “How is acceptance criteria different from test scenarios?” “We have a shared database; someone overwrote my test data!” “I can’t do TDD on legacy code, right?” “We have to work weekends. Let’s skip TDD this Sprint.” “I’ll refactor after I complete the story.” “You should push individuals. Mentoring aside, put your foot down as Agile Coach.” “Has the team progressed on engineering practices? Our processes are fine: focus on the technical front.”

Argh, this is not what I had signed up for! Who can coach me on coaching? What should I prioritize? How do I handle so many stakeholders? Do I even have it in me? This place is so different from my company!

Act III: The Resolution

Some months have passed. The team trusts me. Management trusts me. I trust myself. I still don’t have all the answers but I have the right questions. The team is less dependent on me. I observe them do TDD on their own. Retrospectives are well-facilitated. The team area is surrounded with Big Visible Charts. The build radiator is green. Many stories have been accepted mid-Sprint. The Scrum Master is pushing back on some stretch work but working with Business to identify what can be accommodated. The experienced Java developer is helping others with clean code and test-driving. Testers are almost in pace with the Sprint’s stories, and regression cycle has been cut down. Best part: people are smiling more!

I can hardly believe the changes that have come about. I still recollect the struggles and the frustration. But my initial inhibitions have faded. Being an Agile practitioner gave me a readymade foundation. All I needed was to look at the environment around me differently. As a practitioner, I saw the environment as something to bypass, to deliver technical solutions. As a coach, the environment *is* the problem to solve.

Today a couple of individuals, whom I coached, have been identified to become internal coaches in other teams. My hope is they will not go through struggles similar to mine. They have closely observed me as Coach, and should be able to emulate the role to a degree. In fact, they already showed signs of being good coaches, supporting me in spreading Agile adoption within the current team.

As I spend my last day with the team, I feel proud of them, and highly confident. They are well-set on their Agile journey, and are helping others along the way. I wish them well. CURTAIN.

Behind the Scenes

The narrative is dramatized but still representative of the actual experience. I now elaborate my learnings, gained over multiple coaching assignments.

Act I

Agile methodologies are fast becoming mainstream. A typical Agile adoption strategy is to hire practitioners with an Agile background. Expectation is these new hires will infuse the necessary culture, coaching others who do not have prior Agile experience. For larger companies, new hires cannot have enough influence. These companies engage Agile consultancies. Agile Coaches are seeded within

various teams, introducing a gradual change. This approach cannot scale because a transition within the entire workforce needs a corresponding number of Agile champions. While external coaches provide the necessary trigger for change and the initial support, a proportion of internal practitioners will need to don the Coach hat.

Act II

Most coaching assignments have 3 stages: assessment, active coaching, sustenance. Assessment is the period dedicated to observing existing processes and practices, noting steps that add or diminish value. It involves interviewing team members, measuring technical and non-technical parameters, and making first-hand observations. At the end, we present the observations with a recommended coaching plan. Active coaching starts by identifying near-term and long-term goals in collaboration with the team. We can use techniques like retrospectives, future-spectives, etc. Drawing upon our Practitioner background, we may plan what practices or processes address these goals incrementally. Measure progress on these, and iterate. An aspect of coaching is to enable a team to discover their own path. Focus on facilitation to encourage self-discovery and self-learning. Start brown-bag sessions. Lead effective retrospectives, and groom more facilitators. Conduct fishbowls on difficult topics like scope management. On an individual level, pair up with team members on actual project work. Supplement that by proactively reviewing independent work and sharing feedback. When necessary, get your hands dirty and lead by example.

There will be challenges. Some individuals won't immediately appreciate the benefit behind practices. Getting buy-in from different stakeholders may be difficult. Some people are slow in picking up skills. In such cases, the first step is to manage our own stress levels. Find a mentor for yourself, preferably someone with coaching experience. Read books and articles on coaching. Remind yourself that the goal is a gradual improvement in the environment, especially in individuals, and not solely in artifacts such as the code base or the test suite. Different people take different amounts of time to understand concepts and act on them. Focus on the quick learners; they will become the team's influencers and motivators. But validate their understanding to avoid misinterpretations or critical gaps, since they need to spread the right message further. Some people are also reluctant to trust a new person. Others may be downright confrontational. A few indulge in office politics. Give people time and patience. Do not jump to conclusions. Actively listen, empathize, and motivate. Look out for any allies you can make. During all this, do not lose confidence in your own abilities, and recognize learning opportunities for yourself, especially around soft skills.

Act III

The biggest high you can receive is when the team you coach becomes self-reliant, so much so that they begin helping other teams around them, gaining respect in the entire department or even the organization. Clubbed with successful deliveries and happy stakeholders, it will validate that coaching helped. A Before/After comparison can substantiate these benefits. An improvement in the metrics baseline captured during assessment will indicate how the team has progressed, reaching levels of sustenance.

Sustenance indicates that it is time for the external coach to step back and eventually move on. Once you feel that the team is beginning to reach that stage, start taking a back seat and even some days off. But do not lose track of the measures you would have put in place by now. Check on them after every break. Even if there is no positive increase, the trend should remain stable and not go down.

Beyond sustenance lies the key to scaling Agile adoption. Armed with new skills and having undergone direct coaching, the influencers and motivators within a team can coach other teams. Initially they may be paired up with an experienced coach but soon they will be ready to go solo. Of course, a lot of this is dependent on personality traits. Not all good practitioners make good coaches. Look out for individuals with potential early on, and help them build on their coaching skills in addition to practitioner skills. After all, by now, no one understands what it takes to transition from Practitioner to Coach better than you!

Results

Qualitative

Sustainable Pace & Empowered Team

Observations: Sprint plans met consistently. Team not overworked. Team more vocal and confident.

Coaching focus: Effective Sprint planning and estimation. INVEST principles for user stories. Balance between delivery constraints and business needs. Information radiators. Effective retrospectives. Open multi-way communication. Coaching on articulation, data point capturing, and presentation skills.

Agile Engineering Practices

Observations: TDD as a default way of programming. Refactoring taken up within story development. Common understanding of clean code practices.

Coaching focus: Pairing-based coaching. Pair Programming among team developers. Regular tech huddles. Team-wide emails capturing code-related conversations and conventions. Frequent ad-hoc review of commits, and feedback sharing by Coach.

Collaboration

Context: Business based in US. BAs based in US and India. Separate Scrum teams in US and India, each having separate backlogs but for the same product.

Observations: Collaboration increased across all roles and across locations. Feedback, ideas, and blockers communicated at appropriate times.

Coaching focus: Spread coaching influence across locations. Well-defined story lifecycle to encourage collaboration across right roles at right times: BA/Business collaboration at Story Sign-off; BA/QA/Dev at Story Kickoff, BA/QA/Dev at Story Volleyball, Team/Business at Demo. Regular Scrum of Scrums.

Knowledge Sharing

Observations: Regular knowledge-sharing sessions within the team and across teams.

Coaching focus: Knowledge-sharing pattern set by Coach, continued by others. Visibility of the team's successes and learnings. Leveraging role-based communities such as QA community, BA community, etc.

Quantitative

People

Observations: Coached 20 individuals directly. 2 team members chosen as internal coach for other teams.
Coaching focus: Pairing-based coaching. Coaching not limited to specific roles. Coaching on coaching skills for certain team members.

Testing

Context: Huge legacy code base without much coverage as starting point. Coverage added on the existing code base for new stories or defect fixes.

Observations: % of unit-tested stories reached 85%. Code coverage increased to 7%, adding coverage for ~6,000 LOC in absolute numbers.

Coaching focus: Agile engineering practices. Sessions on testing concepts and techniques like Test Pyramid, ATDD, TDD, Mocking, etc.

Delivery

Observations: Average story cycle time reduced to 5 days.

Coaching focus: INVEST principles for user stories. Effective story slicing and Acceptance Criteria. Improved collaboration. Agile engineering practices. Continuous Integration. Pretty much everything else.

Resources

These cover an overview of concepts and techniques useful for a first-time coach. Most pages mention further articles or books for a deep dive. The content will evolve over time.

Fundamentals

- [Coaching Basics](#)
- [Agile Coaching Roles](#)
- [Agile Coaching Engagements](#)

Techniques

- Assessment
 - [Interviewing](#)
 - [Process Mapping](#)
 - [Metrics Baselineing](#)
- Active Coaching
 - [Facilitation](#)
 - [Measuring Progress](#)
 - [Self-learning](#)
- Sustenance
 - [Self-discovery](#)
 - [Coaching Benefits Reporting](#)