Patterns of Agile Coach roles
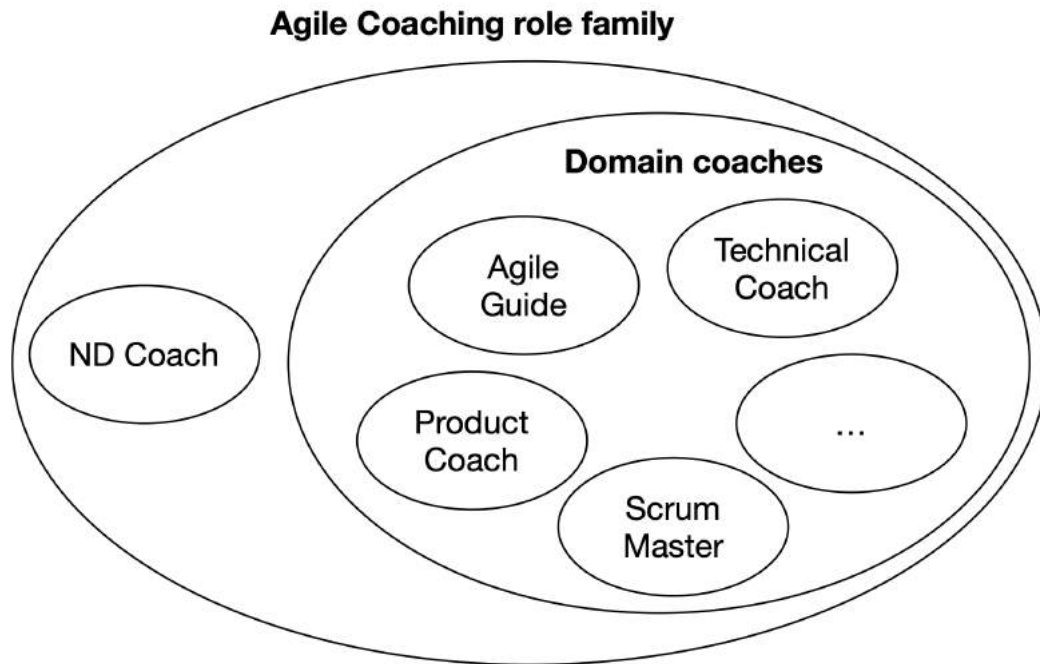
Allan Kelly, allan@allankelly.net

Figure 1: Some members of the Agile Coaching Family

Agile software development continues to grow both in terms of usage and in terms of knowledge and lore. In addition agile has spread beyond software development and is found in many other domains. Rather than describing agile as a "way of working which originated in software development" it may be better to describe agile as "the ways of working which maximises the benefits from digital technology."

The Agile Coach role has become an established role in many teams and companies. Broadly speaking, the role helps teams to adopt and deepen their agile ways of working. But the role itself is mired in confusion: a coach's own model of "agile coaching" is frequently different to that of the organization which hires the coach; and the teams which are coached may well see the coach role differently again.

Agile coaches are not homogenous. Their skills, experience, specialisation and motivations differ widely. Perhaps a better question to ask is *How should an*

organization implement agile coaching?

On closer examination one can distinguish multiple differences in how the "Agile Coach" role is played and how it is viewed by others. Further differences exist in how coaching is delivered and the goals of the coach. Rather than view "Agile Coach" as one role it is possible to see a family of related roles each following its own pattern. Each one of these roles is *an agile coach* but each has role has its own specialisation. This paper aims to describe some of those family patterns.

To put it another way: there is no single Agile Coach role. The term *Agile Coach* describes a class of coaches who work with teams transitioning to, or already working with, agile approaches. In order to find the most appropriate coach, or coaches, to work with a team one needs language to describe different types of coach. This small pattern language aims to identify and name some of those types.
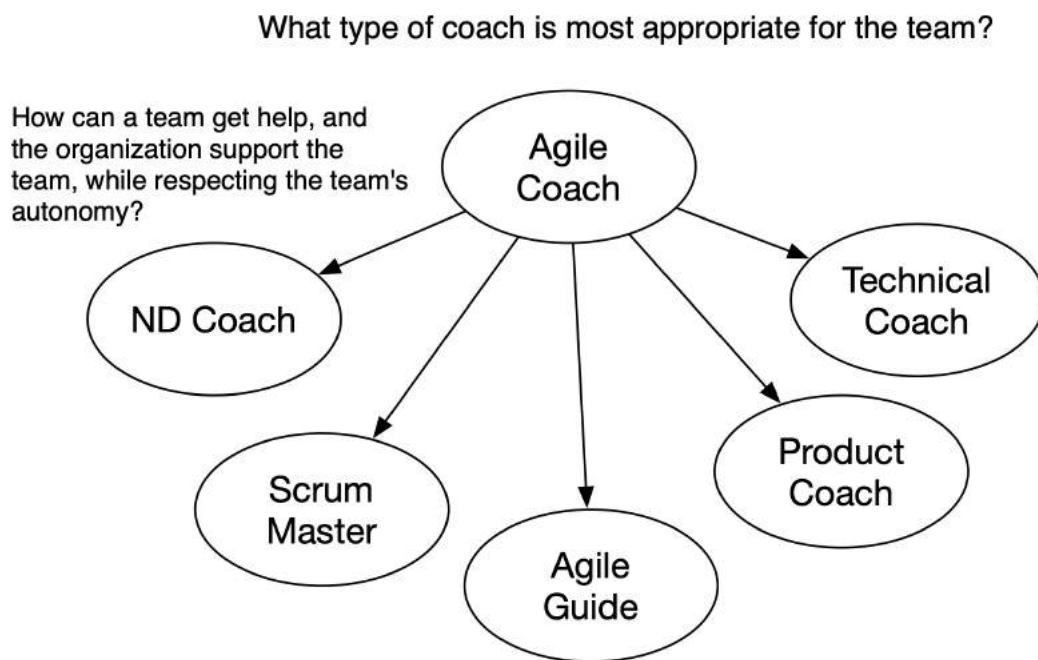

Figure 2: Agile coaches specialise in different aspects of agile and coaching

The degree of specialisation in coaching roles varies. While ND Coaches and Agile Guides tend to work with a broadly selection of people and topics technical coaches can be very focused. For example, a Technical Coach may specialise in Test Driven Development, Continuous Delivery, or some other technology. Such technologies can be very specific, for example a Technical

Coach may specialise in Behaviour Driven Development and within BDD specialise in the Cucumber toolset in Java.

This paper discusses several role which, rightly or wrongly, commonly fall under the "Agile Coach" banner. Several of these roles can prove useful whether a team is perusing agile or not.

Role	Applicability	Skillset
Agile Guide	Teams and organizations pursuing agile working	Experience and knowledge of agile processes, practices, facilitation and human factors
Scrum Master	Teams implementing the Scrum method	Experience and knowledge of Scrum, facilitation and human factors
Technical Coach	Technical teams	Experience and knowledge of technologies being used
Product Coach	Teams building products	Experience and knowledge of product management
Non-directive Coach	Individuals and teams on improvement journeys	Experience of non-directive coaching

Audience

This paper sets out to refactor the Agile Coach role into several distinct coaching roles. In doing each role is named and described. It is hoped that this naming and framing will help:

- Agile Coaches to better understand their roles.

- Managers (and organizations) employing Agile Coaches to better understand the nature of these roles and the typeof coach appropriate to their need.

Research sources

This paper draws on the authors own experiences delivering agile coaching and observing the domain. The author is grateful for the observations and reflections of other agile coaches - in particular John Clapham and Chris Pitts. In addition the author draws additional references as indicated in the text.

Non-directive and domain coaches

[[Sidebox]]

The coaching profession can be divided between domain coaches and non-directive coaches [Downey2003]. Domain coaches have a specialism and may be experts in their own right.

When an athletics coach works with an athlete they may bring their own experience of being a top flight athlete, they may have spent time studying fitness, sports psychology and specific sports. They give direct, and directive, advice to the athlete: pace yourself, focus on breathing, do more gym workouts. Such a coach has expert knowledge.

Such a coach works from domain experience and knowledge. Most types of agile coach are similar: a technical coach is most likely an experienced programmer with expert knowledge of Test Driven Development (TDD), or Continuous Delivery (CD) or some other technology.

Many professional coaches subscribe to an alternative approach: non-directive (ND) coaching. The non-directive coach works without domain knowledge, they utilise generic coaching frameworks to help individuals maximise their performance and achieve their goals.

ND coaches do not aim to teach an individual. They consider individuals as experts in their own domain. Typically an ND coach will use Socratic methods and open questions.

Non-directive coaching is a well established field in its own right. Indeed, most professional coaching courses teach non-directive coaching and result in qualifications in non-directive coaching.

While the term non-directive coaching may not be widely known outside of coaching professionals the approach appears repeatedly. Life coaches, career coaches and psychotherapists are among the professionals who apply similar ideas even if they do not use the words "non-directive."

In most countries there are little or no restrictions on the use of the term "coach." Therefore, sometimes to the dismay of professional non-directive coaches, anyone can declare themselves a coach, in any field, and the term "coaching" is used very loosely across domains. Consequently there is much confusion over what constitutes "coaching" and what "coaches" do.

[[/sidebox]]

Agile Coach

Figure 3: Agile coaches love post-it notes[1]

Context

A team, and perhaps the wider organization are looking for help with working agile. Often they are adopting agile for the first time.

Problem

Doing something different isn't easy, learning new and better ways frequently requires help. **How can a team get help, and the organization support the team, while respecting the team's autonomy?**

Forces

When adopting new ways of working individuals and teams find it very easy to fall back into tried and tested ways of working, but to do so negates the change they are trying to make.

Teams and individuals can be trained in new ways of working but no training programme, no matter how comprehensive, can cover every situation and eventuality. Indeed, the more extensive the training the more there is to forget. Training alone is not enough to cement change.

---

[1]Image from Magnet.me on Unsplash

Agile working is best when teams can self-organization and individual's attain a sense of achievement and fulfilment. But having an authority figure, such as a manager, issue orders and make decisions to bring about change undermines the very change that is being made.

Even when new ways of working have been established and teams are proficient in agile working further improvement is always possible. But individuals and teams, insiders, sometimes fail to see improvements, outsiders sometimes see potential changes that insiders do not. Outsiders can bring new perspectives and thinking which bring new ideas.

Solution

Engage an experienced Agile Coach to work with a team adopting agile.

The physical presence of a coach demonstrates the team are engaged in change and will remind them not to slip back into their old ways. The coach will be able to present new ideas of working at the appropriate time and help team members work through problems in a new way. The coach will help the team adapt and apply agile approaches which may look different to a classroom or text book.

The team may have received agile training before the coach is engaged or the coach may need to start by delivering some training. Not that training is always essential, particularly with the Kanban approach a coach may just decide to "work the work".

Over time the coach might drip-feed learning as the team learns to practice agile. The coach can work one-on-one with individuals to help expand their ways of working, thinking and mental approach. Later the coach can help individuals become more reflective and find their own solutions to problems.

Implementation

Coaches may be engaged on a full-time or part-time basis. A full-time coach will have plenty of time to observe the team and work one-on-one with people. Conversely a part-time coach allows the team plenty of space to maintain their autonomy and avoids the coach wielding power or being seen as a proxy-manager.

Small companies almost always need to hire external coaches on a consultancy basis. Larger companies can often afford their own cadre of coaches on the payroll. But internal and external coaches approach teams differently:

internal coaches have far more organizational knowledge and social capital within the company. External coaches bring a fresh pair of eyes and often see things insiders overlook.

Although it can be hard to change coaches there is merit to periodically changing the coach on a team. Coaches often get worn down but visiting the same issues, they sometimes become demoralised by a lack of progress and they can become stuck in a rut. A fresh coach can bring fresh insights and fresh enthusiasm.

A rule of thumb, from experience, is for coaches to start thinking about rotation at the one year mark. It may take several more months for a coach to disengage, or arrange a replacement, so it might be advisable to start thinking earlier.

However a coach may, consciously or subconsciously, shy away from a rotation. They may have become familiar with the team, happy with the environment and be reluctant to seek a new role.

Recruiting or retaining a coach can be hard; particularly when those recruiting have little experience of agile. While some certifications exist to support coaching and even agile coaching, many experienced coaches reject the notion of certification because such badges are inevitably associated with particular approaches.

Teams and organizations seeking to hire a coach should consider first what they expect of the coach and what type of coach they expect. They may wish to consider questions such as:

- Is the coach expected to adopt a directive or non-directive approach?

- If the coach is adopting a directive approach, how much experience are they expected to bring? In particular, do they need to be well versed in the methodology adopted by the team, e.g. Scrum or Kanban?

- Is the coach expected to teach, and provide training, in aspects of agile?

- Who will be coached and at what level in the organization? Are specialist roles (e.g. Product Owner) to receive coaching?

Consequences

The presence of a coach is symbolic. It change is afoot, that the organization is prepared to put money behind word. Coaches can challenge individuals when they revert to past models or fail to engage in a new way of working.

But when this is done too often, or too vigorously team members may react negatively. The coach may be seen as an enforcer.

What one side sees as disrupting the ways with the goal of improving things may be seen by the other as forcing "change for the sake of change".

Coaches are ultimately employed by, and usually answerable too, the wider organization. Their loyalties may be slit between working with one - or more - teams and delivering the change the organization wishes to see.

People learn best when education occurs "a little and often" and when the learning address a current concern. So a coach regularly delivering educational nuggets of timely advice can be very effective. But when a coach operates in this fashion they come to be seen as the font of all knowledge. Individuals may look to the coach to resolve problems and even develop a "learned dependency" which inhibits individuals from solving problems themselves.

When a coach becomes both an educator and enforcer the role starts to resemble a management role which it was intended to replace. It can be difficult for a team member to appreciate a coach who might spend time in the morning teaching and then in the afternoon asks team members to devise their own solutions to problems.

Companies can see coaches as both educators and enforces, they frequently hold coaches responsible for the success of the team and ask coaches to report on teams.

Agile today covers a broad range of techniques and domains. Agile coaches frequently specialise in specific areas, one may be an expert at Test Driven Development in C# while another is skilled at resolving interpersonal problems.

Examples

The author worked as a "drop in" coach with a medical device company in Cornwall. He delivered an initial training session and helped the team decide what to do. He then returned for one or two days every month for the next year. The team was small and so he had enough time to talk to people and the team as a whole.

At the end of each visit he would ask "When I come back next month what 3 changes will I see?" And when he did return he would check on progress. This "light touch" style allowed the team plenty of autonomy.

Years later Allan worked with a financial institution as a team coach. He spent most days sitting with among the development team, attending daily stand-up meetings, facilitating planning and other meetings. Each month he would have a one-on-one with team members and in between spent a lot of time working with the product owner.

The company expected Allan to also be part of the "Agile coaching team" and exchange notes with other coaches, set and work towards goals for the teams and division, and report back on team progress.

Technical Agile Coach

Figure 4: Technical coaches often spend a lot of time working on production code with coachees[3]

Context

The team is aiming for technical excellence. They may be adopting new practices (e.g. Test Driven Development, Behaviour Driven Development or continuous delivery) or new tools (e.g. Git or Docker).

Problem

Even dedicated training cannot cover everything. How can teams be helped over the challenges of adopting new tools and techniques when their knowledge is weakest?

Forces

When applying a new technology, or technical approach, for the first time team members face a unique challenge: their knowledge of the technique is

---

[3]Images from Alvaro Reyes on Unsplash

minimal and the work to do maximum. But if they do not start working with the new technology they will not learn more and their problems will remain.

Teams frequently face time pressures which the new tools and techniques will help alleviate but initially more time is needed to make the change and learn how to use the tools. The tendency is always for "one more fix" with the old tools before change and the slow down.

A certified agile coach may be great at teaching Scrum, running a retrospective and resolving disputes but when it comes to the code they may lack vital skills.

Classroom learning is great at teaching concepts but when it comes to application people often struggle. For example, a classroom course can teach the fundamentals of TDDs and patterns for tests but when people return to their code they are confronted by a million lines of code with no tests and no clear place to start. Adding tests to existing code becomes a major challenge.

Even when people are trained and enthusiastic to change their motivation may fall. Doing things in the new, classroom recommended way, inevitably takes more time initially even if it pays back later.

Solution

Hire a technical coach to help team members master specific engineering problems, new tools and technologies, e.g. Test Driven Development, Behaviour Driven Development, build tools, continuous delivery, etc.

Technical coaches usually embed themselves in a team and work with other team members while they are practicing new skills or wrestling with technical problems. They will transfer their knowledge by working with team members directly on live production systems and code.

Technical coaches don't take on work of their own, they will pair on code and talk directly to team members who are undertaking work. On occasions the title "coach" may be erroneously conferred on a consultant who is hired to do a specific piece of work, e.g. a *Continuous Delivery Coach* may in fact be a *Build Engineer* tasked with retooling the delivery pipeline to allow continuous delivery.

Implementation

Technical coaches can be chosen for their specific hard technical skills, e.g. TDD in C++, automating build pipelines or configuring virtual ma-

chines. While such coaches need a good understanding of agile and its use they will not be expected to coach individuals through goal setting or talk to CEOs about the internal rate of return.

Technical coaches usually need to be hired for an extended period of time. While they may work less than five days a week they will need to spend several weeks with teams to be effective. And the bigger the development team the more time they will need to spend.

Consequences

With a coach in place the team has a source of expert knowledge and a neutral facilitator.

Technical coaches are generally expensive, their expert knowledge can command high fees and they are needed for extended periods. The bigger the team the longer the coach is needed for. This can make technical coaching unattractive to companies which are cost conscious.

The team will still need more time to make the change but the coach's experience helps reduce that time. The coach's presence, and ability to sustain progress, will maintain motivation. By working with team members on actual problems the new tools are implemented more quickly and team members knowledge boosted.

In time, as the team learns from the coach, as new techniques are incorporated into work quality will improve. Companies will see benefits such as fewer bugs, shorter development cycles, more deliveries.

When time pressure is tight technical coaches are sometimes used to do the work rather than coaching team members in doing the work. While this sustains pace it comes at a price: coaches are more expensive than team members and knowledge transfer is lost.

Example

A large organisation engaged Emily Bache to work as a technical coach with several of teams working on a common legacy codebase. They wanted to become more agile, make more frequent deliveries with higher quality and more confidence, but had realised that the state of the code was preventing progress on these goals.

The developers in the organisation were spending a lot of time fixing bugs and although there were some unit tests, coverage was low and not growing

as quickly as hoped. Some developers were skilled with writing tests and improving design, but by no means all.

The first part of the coaching was to give all teams (who wanted it) training in ensemble programming - also known as *mob programming.* The purpose was to better enable the people who had testing and refactoring skills to spread them to others. Emily then worked with several teams more intensively, with two main activities:

- Short, targeted training sessions on test design and refactoring. Emily would teach the theory and have them practice on small coding exercises.

- Mentoring team ensemble programming sessions in their codebase. Emily would make suggestions and give real-time feedback as they write code, design tests and do refactorings.

Emily spent 2-3 half days a week with each team until the new practices start to stick. Over several months she worked with each team to help them to improve their code, tests, and to become more confident doing so.

Agile Guide

Figure 5: Agile Guides mix teaching and coaching[4]

Context

The organization has decided to adopt agile. The teams have completed a 2-day training programmes and are all certified agile "black-belts", everyone is ready to embrace the new world. The next day they return to the office. . .

Problem

The team are new to agile and there is so much they don't know. How can an organization best support them through the early stages of agile adoption and ensure inertia doesn't keep the team working the same way they always did?

Forces

The team have had, at most, a few days training in generic agile or a specific agile method. But, all training assumes a context which inevitability differs from the team's environment, while this may not invalidate the training it can make it hard for team members to see how the training relates to their context.

Further, agile has grown into such a vast field of tools, techniques and ideas that no amount of training can cover the entire subject. Indeed, in training,

---

[4]Image from Austin Distel on Unsplash

more is often less as students quickly become overwhelmed and struggle to stay awake.

Setting up meetings, facilitating stand-ups, creating burn-down charts and all the other ceremonies in agile take time, both to organize (put the dates in calendars!) and to undertake. This is especially true when they have not been done before. But teams are typically already running at 110% of capacity and barely have the time to attend a planning meeting let alone schedule the meeting, book a room, set up a conference call...

In the classroom, or in a book, every example and case study works well. But back at base there are a myriad of impediments which make it difficult. Product owners struggle to see how to prioritise stories, testers shy away from asking for acceptance criteria in advance, coders are in a hurry and know their code will work (its other people who write bugs). Faced with the day-to-day challenges, let alone occasional crises, it is easy to follow the old habits which have brought success in the past.
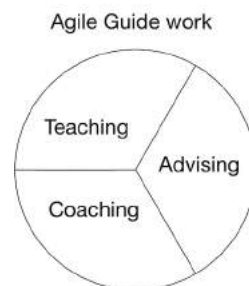

Figure 6: Agile Guides split three ways

Solution

Appoint an Agile Guide to help the team: the guide is a recognised expert in agile working and processes. While a guide may have a technical background and some pure coaching skills the guide's primary role is to help the team over the challenges of an agile change. Using these skills and experience the guide will mix teaching, advising and pure coaching.

Implementation

Guides should be chosen for their experience and knowledge of working with agile teams. Once engaged they will typically be attached to one or more teams, although care must be taken not to overload the guide. They need

time both to work directly with the teams but also passively observe and think about teams.

When a team is new to agile - fresh from training - a guide may need to devolve all their time to the team. As the team becomes more proficient the guide may reduce their time but should remain available if needed.

A guide integrate themselves into the team by picking-up "management" and administration work which team members lack skills, interest and time to do, e.g. schedule and run morning stand-up meetings, facilitate planning and retrospective meetings, etc. The guide may also pick up reporting duties (e.g. updating burn down charts, filing progress reports).

At times the guide may decide to increase their involvement with a team which is beyond initial stages. For example, the team may have achieved an effective way of working but need to tackle a new challenge to order to advance further. Indeed, the team may not recognise themselves that advancement requires a substantial change.

Consequences

For a team, having an Agile Guide gives the team a pseudo-leader who can educate them in finer points of agile, act as a conduit for organizational decisions, act as an enforcer where needed to ensure agile ceremonies take place and company mandates

For individual team members the employment of a guide demonstrates the organisation is serious about adopting and agile approach.

Agile Guides are something of a "jack of all trades" but that also means they can take a holistic approach and view the whole - they may go deep in the technical details or hold back and help individuals resolve their own problems.

Having a guide attached to a team means the guide can give additional training as required - *just in time training* - and talk through specific situations. They can show how to apply agile thinking and techniques to a given situation when it may not be obvious.

When embedded in a team, guides provide another pair of hands which can take on some of the new workload of making agile operate effectively. In time, as the benefits of agile working come through, some work will become routine and other work will go away, e.g. when a team is holding regular stand-ups and planning meeting there is little need of a monthly "team meeting."

Guides previous experience also allows them to navigate the adaptation of approaches when needed and can seize the opportunity offered by a crisis rather than allowing the team to fall back on older ways of working.

Similarly, as individuals become accustomed to working in an agile fashion and start to realise the implication, guides can advance understanding in a timely fashion.

For example, once a Product Owner (PO) has become accustomed to working with a backlog they may come to see that the work may never be completely done. In the traditional model the PO may look to request more resources and budget while a guide could use this epiphany to teach about prioritisation, business benefit, and stakeholder management.

However, the expectations of the organization (paying the bills), the team members (on the receiving end of the coach), and the coach themselves are frequently at odds. In the extreme, this can mean the skills and motivation of the engaged coach are a poor fit for the organisation.

Related roles

An Agile Guide may be consider an alternative to a ND Coach. A guide can be expected to be more directive and more of a teacher than an ND Coach. This may be more appropriate for some teams and at different stages of a team's journey.

For example, when initially adopting agile a guide may be useful in helping the team experience new ways of working. Later, once experience has been gained, a more hands off, non-directive, style of coaching may be more appropriate as the team learn by reflecting on experience. Not only does this change demand different skills from the coach it also represent a change of relationship. Thus it makes sense to swap the Agile Guide - whom the team have become accustomed to following - for an ND Coach who will help team members find their own solutions.

Example

Allan was engaged by a well know but relatively small financial institution as an *Agile Coach*. While the interviewers looked for knowledge of non-directive approaches most of the interview time centred on knowledge of agile and directive work with teams. In role he was part of a coaching squad with goals to increase team agility.

The organization measured agility with a maturity survey which subjectively

surveyed practices rather than awareness and problem solving. Scoring high on the maturity survey required close conformance to Scrum and the "Spotify model". Thus, the organization was looking for conformance to known models.

ND Coach (Non-directive coach)

Figure 7: Non-directive coaches work one-on-one[5]

Context

Executives and teams are looking to become the best version of themselves that they can be.

Problem

Even motivated people struggle to reach succinct, ambitious and achievable goals. Determining how to achieve such goals, and following through is an even greater challenge. Who can an executive turn to for help?

Forces

Every individual is unique and faces a unique set of challenges and constraints. Many of such issues are largely in their own heads and within their own capacity to resolve, but it can be hard for people to recognise and resolve issues by themselves.

Ready made solutions and "best practices" are useful but are often not applicable.

Having other people tell you what to do all the time can be debilitating but identifying and actioning your own cause can be hard. Taking responsibility

---

[5]Image from Toa Heftiba on Unsplash

for solution and action can be very rewarding and motivating. Intrinsic motivation benefits from making ones own choices so following someone else's plan may not be so attractive.

Executives may feel vulnerable or threatened by asking peers for help. They may even be in competition with such peers today or tomorrow for advancement and prizes.

Solution

Engage the services of a non-directive "pure" coach who will work with individuals to help identify their own challenges and goals, then help individuals work towards them.

Non-directive coaching may be beneficial for staff at all levels in an organisation and, probably, in any job role. Organizations tend to limit such coaching to executives, perhaps because of the time and cost of engaging a coach to work closely with individuals.

A non-directive coach may use an existing framework, such as John Whitmore's GROW [Whitmore2002] and be qualified as a professional coach. What they will not do is tell you what to do, indeed they may not have experience in your field whatssoever.

The coach will work with the individual(s) directly, usually one-on-one. Such coaches work on the assumption that individuals are experts in their own right and as such know their own problems best and how to resolve them. ND Coaches use Socratic style approaches to engage the coachees in critical thinking and reflection with the aim having the coachee defining their own goals and explore solutions before deciding on action.

It can be difficult to coach someone who does not want, and expect, to be coached. Coachees must voluntarily enter into the relationship, be open to coaching and trust the coach.

Implementation

Non-directive coaches are usually hired from outside the organisation to work directly with one or more named coachees. Coachees are often senior executives or executives singled out for career advancement.

[*What is the % breakdown between organizations engaging and individuals?*]

Many senior executives will engage the services of a ND Coach whether engaging in agile or not. Consequently (non-directive) coaches have become

associated exclusively with senior roles. This need not be the case, such approaches are broadly applicable at all levels of a company. However, since regular one-on-one coaching sessions are costly companies may be reluctant to pay for such coaching outside of senior ranks.

Consequences

The process of non-directive coaching can be time consuming and thus costly, hence organizations frequently restrict ND coaching to senior executives.

Coachees expecting the coach to provide the answers may be surprised.

At an organizational level within a change programme this can create conflict because the expectations are mismatched. The organization might expect the coach to instruct teams to do agile while the coach sees their role as unlocking individual potential. This conflict will grow if organizations expect the coach to initiate change and report on progress.

Examples

Chris Pitts describes how he coaches a software developer through some issues:

> The developer regularly used exasperated phrases like "The other team needs to redesign for us" (where system 'x' was working for everyone else), and "Management refused to let me do ". He was without a doubt smart, one of the top performers in a team. He was obviously demonstrating a need to change (awareness), and was "hungry for change" (showing responsibility) but just could not see how to do any more. He had changed as much as he could by himself, and now everything was perceived as external. He was blocked and getting frustrated.

> In order to help him put things into perspective, I went through an exercise with him to highlight spheres of control and influence which then allowed a useful conversation around what he could easily control, what he may be able to influence, and finally what he had no control or influence over (so could not change). By bringing the various aspects of his frustration into the open like this, he could consciously decide whether to let them go (there is little point wasting time and energy trying to change something that cannot be changed), act on them (items in his direct control),

or to use/increase his influence to produce the outcomes he saw as required but while taking into consideration others' needs.

Once we had some concrete actions from this exercise, I was able to help him with specific influence and conflict resolution skills to help him reach mutually acceptable results. This ultimately improved the final software product delivered. He also visibly grew as an individual, gaining newfound confidence and respect from his peers and management. Once again, raising the client awareness of both internal limiting beliefs and scope of influence was a hugely powerful coaching tool. Highlighting what could be changed from a personal perspective, and encouraging him to become more aware of his interactions with colleagues empowered and motivated the client to progress leading to improved results for both the individual and the organisation he was working for.

Also known as and related roles

The terms non-directive coach and pure coach have both been used to describe this role.

Lifestyle coaches and counsellors (psychologists and therapists) follow similar approaches and some professionals offer both counselling and executive coaching.

The term *Personal Coach* is also used to describe the non-directive coaching role.

Other coaching family roles (proto-patterns)

Product Coach

. . .

XP Coach

In Beck's original description of Extreme Programming he described a Coach role - indeed the origins of the "Agile Coach" role can probably be traced Beck. In Beck's description the Coach role was filled by existing team members on rotation. When acting as a coach the team members would be exempt from other duties and tasked with observing the team and identifying ways to improve.

Scrum Master

The organization wants to support the team in their adoption of Scrum and the team need help in adopting Scrum initially, and then living with Scrum.

The Scrum Guide [Schwaber2020] mandates that all Scrum teams have a Scrum Master. The Scrum Master may be an existing team member who has been trained as a Scrum Master or they may be new team member who brings Scrum Mastering skills. *The Scrum Guide* asks the Scrum Master to be a teacher, facilitator, coach and process keeper.

Scrum Masters are normally dedicated to just one team and on all but the smallest teams only undertake the Scrum Master role. Scrum Master undertake both the administration of Scrum (scheduling and facilitating stand-up meetings, retrospectives, etc.)

Tensions arise when organizations expects Scrum Masters to act more akin to Project Managers and exercise more control over a team and work. Experienced Scum Masters may fill the role in the style of an Agile Guide or ND Coach, less experienced Scrum Masters are sometimes consigned to administration and gently nudging a team towards Scrum.

On a challenged team, or a team new to Scrum, Scrum Master skills may not be enough. The experience of an Agile Guide may be needed. In some cases each team has its own Scrum Master and one or more Agile Coaches or Guides provide addition support to multiple Scrum Masters and teams.

In time a team may find a dedicated Scrum Master is not necessary and decide to rotate the duties between team members. In this way the Scrum

Master role resembled the XP Coach role.

Every Scrum Master seems to fill the role differently. Like Shakespeare's Hamlet the essential elements of role and general activities are remain constant but everyone who plays the role interprets is differently and brings their own twist to the role.

As Scrum Masters gain experience in their role they may seek to become more of an Agile Guide or retrain as a Pure Coach.

The coach and the organization

The majority of literature on coaching considers the role itself and how a coach operates. Relatively little literature considers how the organization relates to the coach and how a coach fits with the management structures.

This leaves organizations to find their own answers to significant questions. For example: What expectations can be made of a coach and how can success be measured? As the coach role is frequently one catalyst how can success be measured? Can change be attributed to a coach or should it always be attributed to the coachee? Equally, how can one distinguish a failing coach from an uncooperative coachee or team?

Similarly one may ask: what hiring criteria should an organization use when hiring a coach? While organizations hire "generic agile coaches" rather than more specific roles (suggested in this paper) should experience in agile be prioritised above experience in constructive discourse?

In addition there is the question of whether the coach sits inside the organizational structure or outside. Business coaches usually sit outside the management structure as they are external, part-time, consultants. But when an organisation hires internal, full-time, coaches, should these be part of the management hierarchy? Are they better considered as a stand-alone team? Or embedded in delivery teams?

And if a coach is embedded within a delivery team are they equally responsible for doing work? Might there be times when the coach breaks off from coaching to engage is delivery work? Alternatively, if a coach is not a team member, what are their responsibilities to the team?

Both internal and external coach scenarios raise questions of organisation power structures. An internal coach may hold less influence if they are perceived as a minor management role. Conversely, a senior manager undertaking non-directive coaching may still be seen as directive to a junior employee.

Further lines of study and discussion

Power relationships within coaching

The manager as coach?

Team-coach fit and "coaching by consent"

Other family roles: Product Coach

Motivations of agile coaches and guides

Survey of people holding these titles to see what they actually do

Coach and the organization: hiring, structure, expectations

Can coaches ware multiple hats? Directive in the morning, non-directive in the afternoon?

How do the role collaborate and work together? Are there potential conflicts? - is this were the 3 level model comes in? Is there a sequence?

Where predilection for coaching differ: - Scenario 1: Management want team (or individual) coached but team are reluctant to be coaches ("coaching by consent") - Scenario 2: Were team (or individual) want to be coaches but management (budget holders) are reluctant

Time required for coaching: how does it happen when team are 100% busy?

Patterns for engaging as a coach: how to break the ice with a team and enter their world?

Acknowledgements

References

Downey2003: Myles Downey, 2003, *Effective Coaching*, Thompson

Schwaber2020: Ken Schwaber & Jeff Sutherland, The Scrum Guide, November 2020 edition, https://www.scrum.org/resources/scrum-guide

Whitmore2002: John, Whitemore, 2002, *Coaching for Performance*, Nicholas Brealey

Version tracking

0.1 December 2021: first draft

0.2 February 2022: Submission to EuroPLoP

0.3 May 2022: Intermediate update

0.4 June 2022: Pre-conference revisions

0.5 September 2022: Conference comments incorporated. Paper to fork, one reduced version for ACM, one full version for website and future work.

1.0 November 2022: Final public version (this version with pictures)

1.0 ACM November 2022: Final version in ACM format with reduced number of pictures, available at: https://dl.acm.org/doi/10.1145/3551902.3551963

Work to do

Knowledge transfer: nothing is said about learning or transfer, do different roles do this differently?

Referencing style?

Role names (patterns) in Small Caps

Latex for my preferred 2-column format

Research questions - continuity of coaching relationship - does coach role overlap with "Company Psychologist" (Industrial Organizational Psychologist?) - how should the organization relate to coaches? should coaches be within or outside the management structures?

Scrum Master pattern: research what others day about the role

Forces: would it be worth bolding key phrases?

Coaching scenarios: - Crisis coaching - Gorilla coaching - Coaching up

Scrum Master matrix?

Skills			
	Experienced	Frustrated	Happy place
	Junior	Sad	Crash
Organization		**Low expectations**	**High expectations**