IVAR JACOBSON INTERNATIONAL

Case Study

Global Information and Communications Technology Solutions Provider

PRACTICE ARCHITECTURE ON TOP OF ESSENCE: A Key Enabler to Manufacturer's Agile Transformation

- An agile way of working spread to 5,000 practitioners within two years by using an Essence-based Practice Architecture
- A sustainable path to agility was created - teams could incrementally improve their way of working and growing
- Practices provided standardization which helped practitioners share and learn best practices

Creating winning teams.

The Journey to Become Agile

An IT division of a large global telecommunications equipment manufacturer with approximately 3000 staff and 700 contractors supports the parent organization through building and operating IT systems using development teams ranging from 5 to 200 employees (depending on the complexity of applications). Application development has varied — some applications have been developed in-house or as evolutions from off-the-shelf software, others have been web-based or mobile products. Significant technical challenges have been encountered when evolving these systems to work together, as well as personnel challenges related to contracting externally for the development work and dealing with high rates of turn-over. These complexities have led to inefficiencies in time, costs, and quality.

To keep up with business demands for shorter application development lead times, higher quality and productivity, the IT division of this large technology provider embarked on an agile transformation initiative.

Starting with a Common Ground: A Practice Architecture

From the outset, Ivar Jacobson International (IJI) encouraged its client to establish a common ground and a practice architecture based on Essence to align coaches with its agile adoption efforts. The client's acceptance of a common ground shaped its agile adoption strategy. Such an approach proved to be of great value to their agile adoption journey [1].

The journey towards agile transformation started in December 2013 when IT engaged the help of internal and external coaches to guide and mentor development teams. Coaches had a sound understanding of what agile was and how to introduce agile to teams, but differences in background, experiences, and competencies made it challenging to consistently and clearly communicate an agile approach. Coaches' approaches to agile adoption, their advice to their teams, and their coaching methods varied. Moreover, teams embarking on their agile adoption journey had different context and conditions. Some teams were small (10 to 20 persons) and some large (100 persons and more). Some teams worked on mission critical systems, some on exploratory, some on greenfield development, some on COTS, etc. Thus, the large variety of coaching differences and team differences contributed greatly to challenges in orchestrating and aligning the organization's agile adoption effort. The lack of a common ground prevented

teams from learning from each other's experiences; in addition, different advice from different coaches resulted in confusion withiin the teams themselves. Early on, IT recognized the need to have a common ground.

Staff were well aware of the experiences of R&D's adoption of agile back in 2008. At that time, R&D had employed many external coaches in the bid to roll out quickly. However, without a common ground, it was not easy to harvest experiences and learn from one another. Furthermore, there was very little effort invested to capture experiences, let alone establish an explicit practice architecture. IT recognized that they should not repeat that mistake.

Establishing a Practice Architecture Using Essence

The IT organization had surveyed a number of agile method frameworks available, such as Scrum, XP, SAFe®, LeSS, and DevOps, among others. While none of these, individually, met IT's needs, there were good, re-usable practices from each of these frameworks that they could adopt and integrate into a cohesive whole. What they required was a common ground, a unifying practice architecture, and a unifying practice language.

Essentializing Reusable Practices

The team chose Essence [1] as their practice language and organized practices around 'health indicators', known as Essence alphas (see Figure 1). There are seven Essence alphas, each alpha can take a number of states, and the state of an alpha tells how far the alpha has reached in the software development endeavor. Collectively, these states describe the status of the project; they characterize the overall progress and health of the project itself. The seven alphas are: Stakeholders, Opportunity, Requirements, Software System, Team, Work and Way of Working.

SAFe is a registered trademark of Scaled Agile, Inc.

IT's practice architecture is divided into three areas:

1. Governance Processing involving plan, transformation, and run. This approach was based on their in-house governance approach. It was a governing mechanism to determine how much autonomy could be delegated to a development endeavor.
2. Practice Architecture involving different practices that ensure health and progress along different dimensions, i.e. the alphas. We will discuss this more later in this section.
3. Different Ways of Working (which IT calls different development scenarios) for different kinds of development: small or large development, and so on. We will discuss this more in Section 3.2.

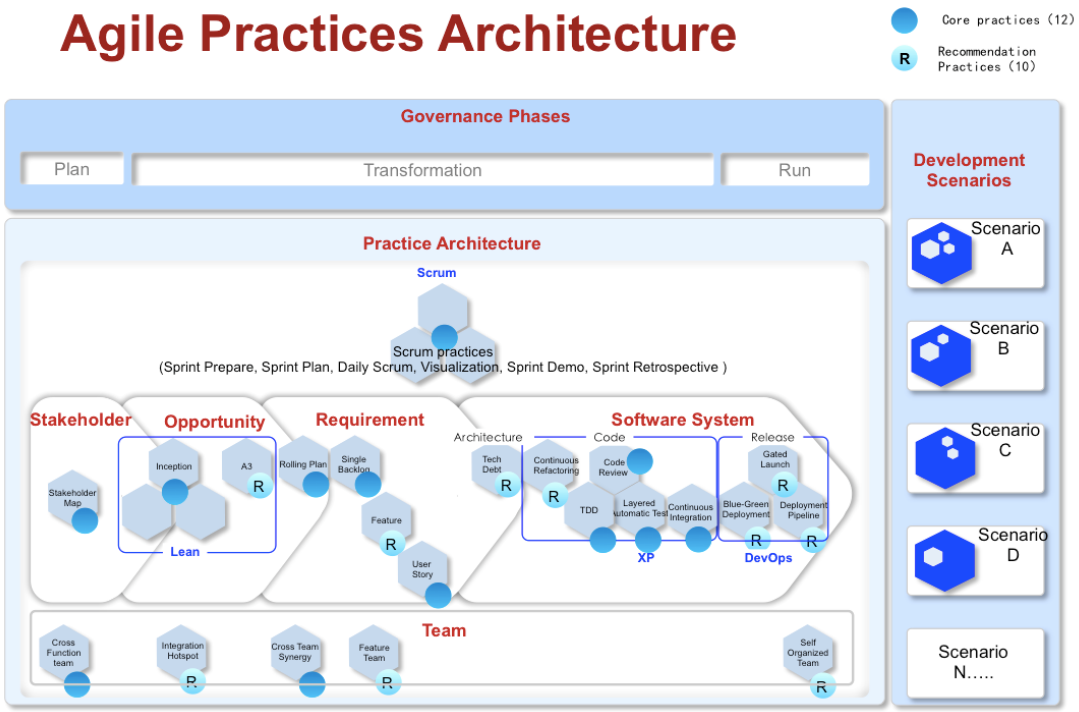

Figure 1: IT Practice Architecture

The organization, with guidance from IJI, defined core practices using the following principles for the development of good practices (i.e., well-formed, easy to understand and adopt):

1. Practices are small and reusable.
2. Practices only capture the essentials, leaving room for teams to adapt to their specific contexts.
3. Practices are composed to meet the challenges of teams.

4. Practices are classified as either mandatory (denoted by "K" for "key" in Figure 1) or optional (denoted by "R" for "recommended" in Figure 1).
5. Practices are described and organized on top of Essence kernel alphas. For example, there are practices to guide teams to progress the Stakeholders alpha, the Opportunity alpha, and so on (see Figure 1).

Each practice was described as follows:

1. There was a one-page Microsoft® PowerPoint® overview that highlighted the purpose and overview of the practice.
2. There was a several-page Microsoft® Word document to explain the practice in greater detail.
3. There was a knowledge management website for practitioners to discuss and contribute to the practice.

As such, Essence stood at the heart of the practice architecture. In particular, there are two important lessons learned through this process:

1. Without Essence alphas, it was difficult to agree on the skeleton for the practice architecture.
2. Without Essence language, it was difficult to agree how to describe and present each practice.

Essence is also useful in setting the boundaries for the scope of each practice and determining if practices are too similar. In this way, IT's practice architecture did not end up with conflicting or duplicating practices.

The current set of practices available in the Practice Architecture at the time of writing are listed as follows:

- Scrum, user story, continuous integration, test-driven development, continuous refactoring – these are already well known in the agile community.
- Cross-functional team, cross team synergy, feature team, self-organizing teams – these are team organization and collaboration practices advocated by the agile community.

Microsoft and PowerPoint are registered trademarks of Microsoft Corporation.

- Stakeholder mapping, code review, technical debt – these are practices independent of agile development, which IT needed to strengthen and follow.
- Feature development, single backlog, rolling plan – these practices originated from large scale agile frameworks like SAFe and LeSS.
- Blue green deployment, Gated launch, Deployment Pipeline – these practices support a move towards DevOps by enabling rapid and reliable deployment of new releases.
- Integration Hotspot, Inception – these practices were custom designed to deal with specific challenges. For example, the Integration Hotspot practice was designed to deal with the complex integration across applications. It attempts to reduce the lead times to perform these integrations. The Inception practice was designed to reduce the time to quickly gain consensus on the initial requirements for a major release.

Composing Reusable Methods

As mentioned previously, the organization has many different development scenarios. Each is complex and challenging in its own way and has different needs. Essence alphas are dimensions of software development and provide a way to reason about such complexity. For example, gaining Stakeholder consensus can be simple or complex; Software System can be simple or complex, and so on. However, the seven alphas are too many, and IT grouped complexity into two key dimensions in-line with how they would bring agile practices to different development teams. These two complexity dimensions are namely:

1. Scale complexity – whether development involves many teams or stakeholder groups. These were correlated to challenges in the Stakeholders and Team alphas. For example, the stakeholder mapping practice is recommended for large-scale development.
2. Integration complexity – whether development involves complex integration relationships with other systems within IT's enterprise architecture. These were correlated to the Requirements and Software System alphas. For example, the integration hotspot practice is recommended for scenarios involving high integration complexity.

Each complexity dimension is categorized as either low complexity or high complexity and leads to four (i.e. 2x2=4) common development scenarios. The practice architecture defines a pre-

composed method for each of these scenarios, as well as examples of teams who had successfully adopted these methods.

Just like practices, IT's knowledge management website hosts the descriptions of these pre-composed methods for practitioners to share and learn from one another.

Using the Practice Architecture to Drive the Agile Transformation Effort

The organization reaped many benefits from having a practice architecture, namely:

1. Practices became standardized organization assets. Teams could easily find them from their internal knowledge management website for learning and sharing. They could easily find out who had expertise in which practices.
2. Practices provided an evolutionary way for teams to incrementally improve their way of working and growing. They didn't need to adopt a big framework like SAFe, LeSS, etc., in its entirety, but could work out a path towards agility, largely by choosing which practices were best for them.
3. Conservative teams who still applied a waterfall governance approach could also adopt some agile practices, e.g. continuous integration.
4. Department managers and business managers could align their understanding on how to evolve their way of working easily.
5. Practices provided a way for IT's practitioners to share and learn. Essence provides a conceptual lens to understand what each practice does and how it would impact their development.

Perhaps the most important benefit for IT is the practice architecture's contribution to the agile rollout. IT had an internal certification process for practices. These certified coaches acted as ambassadors to proliferate an agile way of working to the 10,000 practitioners. Within just two-years, thanks to the practice architecture, their agile way of working spread to 5,000 practitioners. Coaching, knowledge management, and processes all evolved around the practice architecture. This adoption was many times beyond what they had originally imagined when they first started in December 2013.

If You Don't Have a Practice Architecture, Get One Based on Essence

For any organization that is embarking on a large-scale agile adoption effort or trying to innovate its way of working, having a practice architecture with core practices is crucial. The practice architecture embodies the actionable values, principles, and practices for an organization. It is an enabling infrastructure for the sharing of knowledge between developers and departments. The Essence kernel and the Essence language provide the theoretical basis for this infrastructure.

Establishing the practice architecture is not a one-time affair. It needs to continually evolve as an organization grows and their agile adoption continues to evolve. Custodians are still contributing to the practice architecture and adding more practices as IT learns from the industry and invents their own practices. Coaches are helping teams based on these practices. This is an ongoing investment to ensure they sustain their agile adoption well into the future.

References

1. Jacobson, Ivar, Ian Spence, and Pan-Wei Ng. "Agile and SEMAT: perfect partners." Communications of the ACM 56, no. 11 (2013): 53-59.
2. Jacobson, Ivar, Pan-Wei Ng, Paul E. McMahon, Ian Spence, and Svante Lidman. The essence of software Engineering: applying the SEMAT kernel. Addison-Wesley, 2013.

About Ivar Jacobson International

IJI is a global services company providing high quality consulting, coaching and training solutions for customers seeking the benefits of enterprise-scale agile software development.

We are passionate about improving the performance of software development teams, and maximizing the delivery of business value through technology.

Whether you are looking to transform a single project or program or your entire organization with lean and agile practices, we have solutions to suit your needs.

www.ivarjacobson.com

Sweden
+46 8 515 10 174
info-se@ivarjacobson.com

United Kingdom
+44 (0)207 953 9784
info-uk@ivarjacobson.com

Asia
+8610 82486030
info-asia@ivarjacobson.com

Americas
+1 703 434 3344
info-usa@ivarjacobson.com