ThoughtWorks studios

Agile ALM White Paper:

Redefining ALM with Five Key Practices

by Ethan Teng, Cyndi Mitchell and Chad Wathington

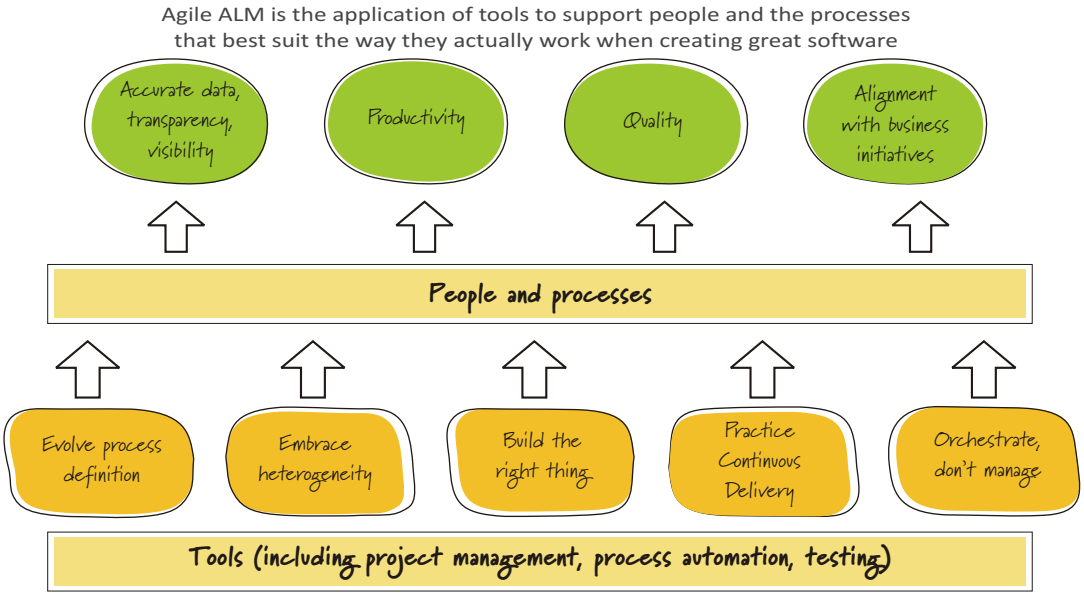
Introduction

The pervasiveness of the Internet, social media, and smart mobile devices have changed the pace and global nature of business forever. New opportunities and threats constantly appear and disappear, with increasingly shorter frequencies. Customer expectations are higher and untiring – they want more value, faster, and more often, and they are not afraid to tell the world about it. Companies are under constant pressure to quickly adapt and innovate new business models to stay ahead of the competition. Their ability to do so is increasingly tied to their ability to design, deliver, and evolve software. This presents a number of challenges to the modern IT organization and calls for a new approach to Application Lifecycle Management (ALM).

Traditionally, ALM has focused almost entirely on defining and enforcing process, and very little on identifying and delivering what customers actually need. Although it rightly aimed to improve consistency and quality, these often came at the expense of productivity and delivery. For at least two decades, traditional ALM has helped to reinforce the negative organizational behaviors and practices that create roadblocks to production deployment: unstable environments, monolithic heavyweight tools, fragile builds, and ineffective hand-offs between development, testing, deployment, and operations. Still today, it causes companies to build suboptimal software and suffer long delays in getting it into the hands of their customers.

Agile ALM frees companies from the heavyweight processes and tools that stand in the way of creating exciting and useful software. It regularly delivers working software into the hands of real users on a weekly, daily, or even hourly basis. This allows companies to quickly gauge customer interest, accommodate feedback, and make course corrections to exploit new windows of business opportunity.

While traditional ALM applied tools to enforce pre-defined standardized process, Agile ALM is the application of tools to support people and the processes that best suit the way they actually work when creating great software. This paper outlines the five key organizational practices of Agile ALM: (1) Evolve process definition; (2) Embrace heterogeneity; (3) Build the right thing; (4) Practice Continuous Delivery; and (5) Orchestrate, don't manage. It describes how companies that keep these practices in the forefront of all their software delivery activities will reap the benefits in increased agility. In addition, it explains why it's important for companies to provide the right tools to support these practices, and how the wrong tools can get in the way.



Tools enable transparency and more aligned delivery if 5 practices are followed when applying them to SDLC processes

#1 Evolve process definition

"Agile processes are generative, not prescriptive. Processes need to evolve as needed, not be prescribed up-front. A prescriptive approach generates complex and complicated processes whereas a generative approach begins with a set of simple processes and adds others as they are needed."

- Jim Highsmith, author of *Agile Project Management*

Agile ALM recognizes that the most effective process for any software initiative cannot be defined up-front; it must be continuously discovered, evolved, and refined over time. Successful teams accept that they cannot predict the future and instead set themselves up to adapt to the changes and constraints that will undoubtedly come their way. They start with a vision, some baseline assumptions, and an agreed process that is as simple as possible. They build into that process' opportunities to regularly reflect and improve upon it – to stop doing things that aren't working, do more of the things that are working, and accommodate new things that may have never occurred to them (or the business) before.

Evolving process definition requires tools that explicitly enable this kind of sophisticated 'on-the-fly' tailoring and continuous improvement. Critically, this should not be confused with, or traded-off for the more basic, up-front customization that is provided by traditional ALM tools and many prescriptive Agile tools. On the surface, these tools allow users to define their process up-front by adding fields into a template, creating workflows, defining reports, etc., but once the project gets underway, they can no longer be further customized to accommodate unforeseen and badly needed process changes. Before long, the tools start to get in the way – slowing things down when unnecessary steps can't be eliminated from a workflow, and creating confusion when new steps can't be added. Agile ALM requires tools to be agile and adapt to the way teams work, while they work.

Companies that provide appropriate tooling and organizational level support for evolving process definition greatly improve their prospects for achieving business agility. It's a simple chain of causality: teams that are empowered to continuously improve and make themselves more effective produce better software more quickly. Tools that make them faster and do not obstruct, will get used. As they get used, project data will be captured. As project data is captured, business and IT stakeholders get real-time visibility into those projects. This real-time visibility then gives stakeholders the confidence to steer and change course in response to competitive pressures and new market opportunities.

#2 Embrace heterogeneity

"If self-adaptivity occurs within a project, it's even more marked across an organization. A consequence of self-adaptivity is that you should never expect to find a single corporate methodology. Instead, each team should not just choose their own process, but should also actively tune their process as they proceed with the project. While both published processes and the experience of other projects can act as an inspiration and a baseline, the developer's professional responsibility is to adapt the process to the task at hand."

- Martin Fowler, co-author of the *Manifesto for Agile Software Development*

Agile ALM embraces the fact that every software project is different. Just as processes cannot be completely defined up-front, they also cannot be completely standardized across all teams. For example, a new product development team may need to work with loosely defined requirements in order to allow enough flexibility to respond to focus-group feedback. In contrast, a team adding enhancements to a production system in a heavily regulated industry may need to define strict development procedures to ensure compliance.

Embracing heterogeneity requires tools that support the best process for each team – one that is uniquely tailored to their circumstances and needs. However, even though every project is different, managers and stakeholders still need to have a comprehensive and consolidated view of all projects in their organization. In addition to promoting process diversity, tools must also support just enough consistency across projects to facilitate reporting roll-ups, governance, and other organization-level needs such as compliance.

Companies that embrace heterogeneity build high-performance delivery teams and actually improve their ability to mitigate risk. Process standardization often seems like a great way to reduce risk and eliminate large failures. But by definition, standardization reduces *both* the low-performing variations, *and* the high. This makes it harder for teams to excel, and often institutionalizes consistent mediocrity, or even worse, consistent failure. Instead, meeting compliance and reporting needs with just enough process consistency supported by the right tools, and allowing teams to uniquely tailor their approach provides greater visibility to manage risks as they arise while software is being delivered.

#3 Build the right thing

"The pace of change is accelerating, and businesses need to move fast to keep up. If you're not constantly delivering working software, you're not getting feedback on whether what you're delivering is useful. Overwhelmingly, the chances are that you won't build the right thing first time. The longer it takes you to get feedback and pivot, the more time and money you spend on building the wrong thing."

- Jez Humble, author of *Continuous Delivery*

Agile ALM embeds customers and business people into the day-to-day activities of software teams. As members of the team, they see the software as it's being created and provide immediate feedback to help make regular course corrections towards building the right thing. In stark contrast to this, waterfall-based methodologies ask stakeholders and analysts to define requirements up-front and hand them off to a development team which then works in isolation to fulfill them. This rarely results in software that meets the business needs because the defined requirements are easily distorted as the details change or are misinterpreted during hand-offs from team to team.

Traditional ALM tools attempt to address this disconnect by emphasizing the lifecycle of a requirement. They force teams to heavily document and move process artifacts through sign-off milestones. This focus on adhering to the defined process and managing artifacts diverts attention away from making sure the requirements are correct. The common result is either no software, or software that does not meet the needs of the customer or the business.

Building the right thing requires tools that facilitate dialogue and collaboration at every point along the software development lifecycle: bringing business and project teams together when high-level goals are initially defined, and helping them decompose the vision into more detailed requirements and implemented features along the way. Critically, these tools must help business people, analysts, and testers work together to correctly define acceptance criteria, as well as align the business, development, support, and operations teams to make regular deliveries into production – the best way to ensure the team is building the right thing.

Companies that focus on building the right thing over managing the lifecycle of artifacts create software that is more exciting and useful to their end users and customers. Through enriched engagement with their customers, they discover new opportunities to create value and build longer-term mutually beneficial relationships.

#4 Practice Continuous Delivery

"Keeping your software production ready from the beginning of the delivery process has several benefits. First, it means releasing is a low-risk, push-button process. That means you can do it more often, which means you can show it to users to get feedback more frequently, test it under realistic conditions more frequently, and get the only real measure of product progress. In short, it means you can deliver higher quality software faster, and reduce waste in your delivery process."

- Jez Humble

Agile ALM continuously delivers a few features every few days or weeks, instead of a bunch of features every few months. Practicing Continuous Delivery ensures that the path to production is clear – kinks in the release process are ironed out early and predictability becomes the norm. In traditional ALM, releasing software is often a stressful and error-prone activity. Awkward manual hand-offs across the silos that segregate Development, QA, and Operations via role-specific tools cause long delays. They also introduce errors that later manifest as production instabilities and outages. In contrast, Agile ALM integrates all these perspectives into the software creation process from the very beginning and every day thereafter, ensuring that everyone's needs are met and automating as much manual activity as possible.

Continuous Delivery requires tools that facilitate detailed technical collaboration and context sharing. The tools should help testers, analysts, business people, operations, and developers test the right things and quickly find and fix the things that are broken. They should reduce or even eliminate errors by automating where possible every validation step on the path to production, e.g. functional testing, performance, security, and general operational acceptance testing.

Companies that practice Continuous Delivery, and provide the appropriate tools to support it, build predictability and repeatability into their release process. This rigor affords them the confidence to release new features whenever they want – trying out new ideas quickly and adapting their approach to accommodate customer feedback and exploit new windows of opportunity. And, because they release early and often, they also achieve earlier returns on their software investments.

#5 Orchestrate, don't manage

"Governance is about monitoring investment and risk, not checking-off process steps. In fact, the key to effective executive project oversight is separating governance from operational processes, evaluating investment and risk at key project milestones and generating the data for that evaluation through iterative delivery."

- Jim Highsmith

Traditional ALM has historically confused process compliance for governance. It introduced and reinforced the notion that the main role of management is to mandate a single set of processes across all software teams. To ensure process adherence, traditional managers typically provided heavyweight tools to enforce the workflow, track artifacts, and report progress. Unfortunately, still today this results in many teams spending the majority of their time managing artifacts and working around mismatches among processes, tools, and reality – all at the expense of delivering working software. This is the main cause of the misalignment that is so prevalent between management and project teams in many companies today.

Agile ALM instead calls upon company leaders and managers to become orchestrators, making sure the work being done is aligned with business goals, removing obstacles, fixing bad projects, understanding

the portfolio, and helping to spread best-practices across teams. This means providing leadership and tooling to support the key organizational practices: (1) Evolve process definition; (2) Embrace heterogeneity; (3) Build the right thing; and (4) Practice Continuous Delivery; and focusing attention on coordinating teams to make successful, business-aligned delivery a regular occurrence.

Orchestration requires tools that provide radical transparency and visibility into every corner of the organization. They must make it easy to roll-up and summarize status in real-time, and see dependencies among individual projects and initiatives. They should also make it easy to zoom in and see where the risks and bottlenecks are or conversely, where things are going well and should be highlighted and potentially replicated across other areas of the organization.

Companies where leaders orchestrate as opposed to just manage process compliance achieve much greater alignment between management and software teams. Applying the right set of tools enables real-time visibility across all areas of business, analysis, testing, development, and operations, making it easier to see wrong-turns and roadblocks and keep software initiatives aligned with business objectives.

Conclusion

In Traditional ALM, tools often drive process. In Agile ALM, people drive process, and tools adapt to support the process. The right process cannot be defined up-front, nor can it be completely standardized across all teams. The best way to manage risk is radical visibility, not complete standardization. The best way to deliver software that meets business needs is through constant collaboration, not managing artifacts. The best way to get software into production regularly is to pull the pain of deployment forward and break down operational silos. And, the best and most business-aligned software initiatives have leaders who orchestrate, instead of just manage.

The best tools for Agile ALM must support the five key organizational practices. They must:

- Enable teams to discover and evolve their process 'on-the-fly'
- Support the unique needs of every project, as well as the organizational needs for consistency
- Facilitate constant collaboration among practitioners and stakeholders at all levels
- Automate testing and deployment to enable Continuous Delivery
- Provide complete transparency and visibility for leaders, without creating work for project teams

Companies that embrace the five key organizational practices of Agile ALM and provide appropriate tools to support them - increase their agility and their ability to compete. They create better, more exciting software, and get it into their users' hands faster. Through better software, they discover new ways to engage and create value for their customers. They try out new ideas quickly, and they continuously adapt in-line with customer feedback, shifts in the market, and changes in business strategy.