White Paper

Building Better Software Architecture with Agile Model

Author: Tabinda Aftab

Research and Development

Application Services

Ephlux

Preface

As the principle of agile development states that "Continuous attention to technical excellence

and good design enhances agility", this is contrary to the common misconception about agile development methodology that agilists do not concentrate on building durable and dependable technical design.

In reality, the technical design in an agile world evolves all along the project, adopting the changes of circumstances and the ever changing requirements. The technical designing starts at the very initial phase of software analysis in agile models however unlike traditional analysis approaches lots of documents are not developed for this task.

Implementing agile methodology to every area of software process lifecycle is not very easy this is especially true for testing and architecture designing. Both of these areas are essential for the success of any software solution, the design and build methodology of Agile model requires special attention and efforts in these areas to produce a dependable solution.

Architecture is perhaps the most controversial topic for Agile teams - Agile development model emphasizes on lightweight requirements, yet understands that strong architectures are the key to build flexible applications. There have been several theories and controversies regarding the challenges faced by software teams in designing solutions in an Agile world.

This paper focuses on different models and methodology in agile development for building strong technical architecture.

Introduction

Every software solution is built upon software architecture. The architectures focuses on everything from initial assumptions about how products fit together, to which features are of value to business and consumers, what are the expected integration points, with which related technologies, what excising product/solution/framework can reused or extended. As software products find acceptance among customers, the creators (vendors) of these solutions eventually find the need to adapt underlying architectures. Agile Framework encourages on doing this early in the product lifecycle and with continual review that promises the vendor with the ability to adapt quickly and effectively to changes in the marketplace.

Architectural changes in late stages of development or testing can be complex, expensive, and hard to deliver on time and most importantly can affect the entire process flow of the system. Some software companies ignore the need for architectural change entirely or underestimate the magnitude of the work. Typically, the results of deferred architectural improvements are major technical reasons that block the launch of new products and enhancements.

Agile development model has been considered as a framework of processes that has altered the traditional approaches to software development. Customer input and business rules and goals were initially considered as external inputs from the development organization and as such were not well integrated into the development of products.

With several success stories, Agile now has a broader place in the technology and business world. Many businesses now see and practice Agile as a product development methodology from project launch to delivery, maintenance and support, as well as for product enhancements and releases. As the agile products have been launched in the real market and consumers interact with these products the need of robust architecture that can accommodate the changes in customer requirements and upgrade to technological shifts; is even more emphasized. Longer lifetimes for Agile products drive us to concentrate and plan more about architecture.

Agile has some exclusive ways of approaching architecture change that are increasingly acknowledged as Agile matures and is widely adopted.

Architectural Envisioning

Agile development methodology includes an initial requirements and architectural modeling effort during Iteration 0 of an agile project. This is the period of the project where the analysts try to get the project off on the right foot. The main goal is to understand the scope of the effort and to identify a plausible technical strategy. The information garnered from these efforts will help with the initial high-level estimation and scheduling, which in turn might raise project related issues such as technical resource requirement, hardware related limitation or requirement, technical environment requirements etc.

Architectural envisioning directs the business analysts and technical architects along with the business managers to perform some high-level architectural modeling early in the project to help foster agreement regarding the technical strategy within the team and with critical stakeholders. The goal at this point is to identify an architectural strategy without developing several documents detailing the outcomes, processes and issues. The secret is to keep things simple. Agile model of development emphasizes not to model a lot of detail, but to model just enough. Writing simple use-cases, simple whiteboard sketches or CRC cards are good enough ways for the architecture designing in Agile model. For your technical architecture something as simple as a whiteboard sketch over viewing how the system will be built end-to-end is good enough.

Why Architectural Envisioning?

Some people practicing agile models say that such architecture envisioning or foreseeing in not important. However, research shows that architecture envisioning offers several benefits:

- **Improved productivity**. Some critical technical issues facing the project can be explored early that can potentially avoid going down fruitless technical paths.
- **Reduced technical risk**. With Architectural Envisioning your team increases the advantage of having a guiding vision without the disadvantage of having to overbuild your system - just because you've modeled it doesn't mean you have to build it.
- **Reduced development time**. Initial agile architecture envisioning enables making better cost and time estimates for the project.
- **Improved communication**. Having a high-level architecture model helps to communicate confidently what you think you're going to build and how you think that you'll build it.

- **Scalable agile software development**. Initial architecture model will be a key work at any time the product needs to be scaled or extended because it provides the technical direction required by sub-teams to define and guide their efforts within the overall project.

Architecture or Tarchitecture

As with any other traditional development methodologies, architecture matters in agile model. It is integral in the successful delivery of a software product and those who fail to grasp this leave themselves open to failure as the underlying design cannot support the product.

Agile has also transformed from small co-located teams to large organizations spanning time zones and international boundaries. The scale and scope of Agile projects makes them more likely to need technical architectural changes. As Agile Model has matured from short-duration lab projects to strategic development efforts, the importance and need of good architecture, design and quality has become unavoidable. Architecture in Agile world has evolved to become **Tarchitecture (Technical Architecture)** which covers not just software architecture, but hardware architecture, hosting approaches, and the business support models for products being produced and marketed.

The basic principle of Tarchitecture states that drivers for change usually come from outside the development team as depicted by the figure. The figure here depicts the relation between agile processes' maturity and Systems Engineering. The approach here is to merge process development with product development along with the Systems Engineering process and strategy. The result is a seamless connection between agile processes and the systems engineering disciplines.

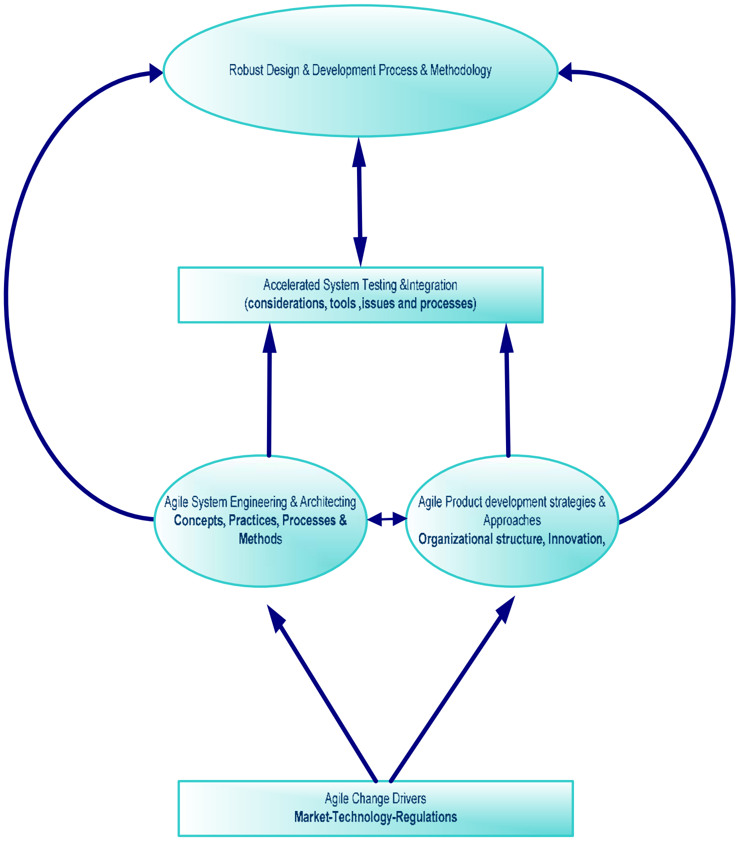
Robust Design & Development Process & Methodology
Accelerated System Testing &Integration
(considerations, tools ,issues and processes)
Agile System Engineering & Architecting
Concepts, Practices, Processes & Methods
Agile Product development strategies & Approaches
Organizational structure, Innovation,
Agile Change Drivers
Market-Technology-Regulations

Figure1: Figure shows that the factors influencing design, development strategy and methods are technology, market regulations.

Demands that force variations, of the underlying architecture are recurrent and customer-driven such as expanded feature requests, competitors' announcements, new mobile platforms, evolution of business models toward pay-per-use and advertising-based revenue, multi-tenancy replacing single-copy customer installations, newly acquired business partners needing back-office integration, radical drops in the cost of storage and bandwidth, etc. The process of adaptation of changes is ongoing just as these requirements and influences are unpredictable and continuing.

Such situations create chaotic situations between priorities, backlogs and commitments. This leads back to the fact that architects are required to develop a design that is easily adaptive enough for such upcoming changes and influences. A combination of business management team and technical architects, business analysts can reduce this risk factor to much lower levels ensuring that the resulting design foresees and includes technical constraints and as well as business/consumer related constraints. This strategy ensures that the team stays and moves in Agile fashion.

Involvement of Tarchitecture in Product/Business Management

Product/Business Management teams are the owners of the product, and are therefore accountable for its direction and development to serve its marketplace. The Management must also map out the near future releases of the product patches/features, identifying what the product will look like and creating a roadmap. It is critical for software architects to be part of this long-term planning. Architects and analysts can help define what can be done, how seemingly unlike requirements can be combined, and how the product will need to change over time. Contributing in scenario planning for the product is a great way of understanding likely changes that could emerge afterwards.

A traditional product roadmap shows what features and functionality will be added or enhanced into the product over the time. However, a roadmap normally mentions features only from the business and revenue point of view as mostly these documents are developed by the product managers and business development teams. Technical teams have no or least inputs at this stage. Significant market data and delivery timeframes are identifies independently by these business teams and shared with stakeholders.

In agile world the road mapping process is much more powerful and more actionable. In the proactive agile road mapping more accurate results are achieved by introducing the Architects (software, hardware, and support) as part of the road mapping team led by Product Marketing and Management. The technical architects and business analysts are able to identify key Tarchitecture changes up front, and when those changes will be needed to meet specific goals and delivery dates. Having this information at the beginning of the planning stage (i.e., during strategic planning) and not just when moving items from build up into development, dramatically improves the product itself by allowing stakeholders to foresee changes and integrate key design improvements that may otherwise be overlooked or that are not developed due to time constraints of finding out about architectural issues so late in the design and build method. This is not to say that architect/product management teams will recognize every detail of Tarchitectural change, but that good analysis and planning can provide a competitive advantage.

Introducing Tarchitects at the start of the product lifecycle provides maximum leverage for their expertise, and is a key factor in developing durable product architecture. Consider the potential business impact to your own business of including Tarchitects early in your road mapping.

This may also allow smarter architectural choices. Since the architects can see ahead, they can anticipate some of the surprises which might pop up later at some stage when changes are not anticipated.

A typical example would be the need to significantly modify an underlying database's design structure, something that without architectural input may not be evident until development begins. s. For instance, they might see opportunities across projects, for hardware platform savings*,* business*,* design choices and other kinds of insights that may not be obvious to less technical reviewers.

Tarchitecture Evolution

As the Agile methodology moves from mapping of products into the features that will be constructed in the development and engineering phases, thus it can be said that the Tarchitecture evolves all along the product lifecycle responding to every new opportunity and influence.

As the concept of agile off shoring is taking the lead with teams spread across continents, proper measures to incorporate the Tarchitecture model might be required by the management. Software architects, Network architects, Support and deployment teams are just as important part of any agile model as the other. Even though these roles do not communicate with the customer directly but to prevent a product and its infrastructure from becoming unusable and obsolete every management team taking up agile framework should involve these teams in the strategic and planning phase of product lifecycle.

Therefore, involvement of technical architects, network engineers and deployment experts along with the product and business managers results in a product with strong and dependable architecture, infrastructure and smooth deployment. Such products promise easy future enhancements and evolvement that is the dream of every product management team. The key is total collaboration and communication between the team and with the client/stakeholders throughout the lifecycle.

Conclusion

In conclusion, Agile Development Methodology though works on ‘just enough requirements at a time’ model it has some much defined procedure for architecture and system design principles. Utilizing agile framework does not say to bear less durable or less expandable system architecture. In contrast, the very evolving techniques of agile model improve and enhance the architecture too, according to the technology, market or business constraints and requirements.

Agilists believe that envisioning the system architecture in the initial phases reduces the risk of bad architecture. Additionally, involving technical architects with the product managers or the business managers further lowers the risk of identifying a fault or loose hole in the architecture that usually shows up not until the later stages of development or testing begins.

Tarchitecture, covers not only software architecture, but also physical and support architectures. To be successful in the new Tarchitecture world, Agile teams must foster collaboration, bring architects into the life cycle much earlier, and let them share ideas/observations throughout the product lifecycle. The results are not only better-architected products, but also the uncovering of new business opportunities hidden within that architecture.

References

1. Ambler, S.W. (2003). *Agile Model Driven Development: The Key to Scaling Agile Software Development*. **www.agilemodeling.com/essays/amdd.htm**
2. Ambler, S.W. (2007). *Dr. Dobb's Journal 2007 Agile Adoption Survey*. **www.ambysoft.com/surveys/agileMarch2007.html**
3. Ambler, S.W. (2004). Agile/Evolutionary Data Modeling: From Domain Modeling to Physical Modeling. **www.agiledata.org/essays/agileDataModeling.html**
4. Hodgkins,Peter (2008), Software Architecture Challenges and Significance in an Agile World **www.agilejournal.com/April/**