The emergence of the agile product delivery model in software engineering organizations is driving the need for new architecture team structures. Software engineering leaders should use this research to define the roles, skills and organization of their architecture teams.

Overview

Key Findings

- Software engineering teams are moving to agile product delivery models, but many architecture teams are still organized based on traditional projects. These misaligned teams can't work together effectively.

- Architecture roles are based on technical disciplines, but product teams are cross-functional and responsible for all aspects of their products' life cycles. Specialized architects can't meet the diverse needs of product teams.

- Software engineering practices that are adaptive to change conflict with rigid, waterfall-based architecture practices. Architecture teams need to find new practices that match those of their software engineering organizations.

Recommendations

Software engineering leaders in agile product delivery organizations should:

- Organize architecture teams according to your product structure by assigning each architect to a level such as product portfolio, product line, individual product or platform.

- Meet the demand for more versatilist than specialist architecture roles by focusing talent development and hiring on solution architects.
Introduction

What About Enterprise Architecture?

In most organizations, enterprise architecture (EA) is not within the software engineering leader's area of responsibility. EA provides guidance, support and direction across all areas of digital businesses and is more involved with executive management and strategy than with software engineering. Nevertheless, many organizations do include enterprise architects in software engineering. In such cases, the software engineering leader and EA leader must clearly define architecture roles and responsibilities. In the agile product delivery model, EA participation is usually at the portfolio level.

Existing architecture organizations are structured to serve software engineering organizations whose own structures are shifting toward an agile product delivery model. Results from the 2020 Gartner Application and Product Leadership Changes Survey indicated that, on average, organizations expect that more than half of their work will be done using a product-centric model by 2022. Yet architecture organizations have largely failed to make this shift themselves. As a result, architecture capabilities are becoming misaligned with the software engineering capabilities they support.

Software engineering leaders who don't establish architecture teams with the roles, skills and structures needed to work effectively with product-based software engineering teams will find the flow of their value streams impeded. They also risk rising technical debt as the software they produce misses the benefits of architecture guidance. To avoid these consequences, they must shift the organization, roles, skills and processes of their architecture teams to focus on product alignment, versatilists, adaptation and people (see Figure 1).
Analysis

Organize According to Your Product Structure

The biggest difference between the project model and the product model is continuity. Projects and their teams are discrete and temporary. At the end of a software engineering project, the software's development is considered complete. It is handed off to someone else to operate and maintain, and the project team members go their separate ways. A product, by contrast, remains an ongoing concern from inception through to retirement. It is continuously modified and improved as stakeholder needs evolve. An equally long-lived team takes end-to-end responsibility for all aspects of its development and support.

Traditional architecture team structures are a response to the demands of the project model. For many software engineering leaders, this is the starting point (see Note 1 for examples).

In the product model (see Figure 2), planning and design work that requires architecture input happens at multiple levels, from the portfolio as a whole through to product lines, individual products and the platform that supports them. Also, because each product is an ongoing concern, this work happens continuously. Setting up and tearing down ad hoc architecture teams that skip from product to product and level to level is inefficient and impractical. It’s also disruptive to product teams who are regularly forced to learn to work with a new set of architects.
The solution is to reflect the levels of the product model in the organization of the architecture team. Do not divide architects by technical domain, then reassemble them into ad hoc teams. Instead, group architects based on the level of the model at which they work, as shown in the organization chart in Figure 2.

Roles at the product portfolio level will usually be filled by more senior architects. This will match the seniority of business and software engineering participants at that level. Portfolio-level decisions will impact multiple products, so they must be carefully considered and balanced. Portfolio architects will also find themselves involved in budgeting and capacity planning negotiations in which they’ll have to advocate for architectural work.
Architecture leaders will have more of a facilitating than directing role, and many organizations will dispense with lead roles. The day-to-day work of product and product line architects will be driven by their product teams, not by the architecture team. Avoid a hierarchy in which product architects report to product line architects who in turn report to portfolio architects. Such a structure will conflict with the autonomy and internal organization of product teams. Alignment of visions, release plans and roadmaps is important but must come from a consensus.

Platform architects will find themselves serving many internal customers and stakeholders. These include the product teams, of course, but also other architects such as security and enterprise architects. Platforms provide an opportunity to reify cross-cutting architecture decisions and strategies, such as microservices architectures, cloud architectures, security approaches and DevOps toolchains. Inevitably, stakeholder concerns will collide, so ensure that platform architects are prepared to manage the resulting conflict.

Meet the Demand for Versatilist Roles

A leading challenge for a product model architecture organization is the need for versatilist architects who cover multiple technical domains. Product architects need to be able to guide the overall architecture of their product, and that requires broad knowledge, even if it comes at the expense of depth. Versatility gained through diverse experience means that architects will be able to move from one product to another as product life cycles progress and workloads shift. Architects already working in a versatilist role, such as solution architects, will be in high demand. Specialist roles certainly won’t disappear — a given platform architect may be an integration specialist, for example — but their prominence will decrease, and many specialists will need to widen their skill sets.

Software engineering leaders should:

- Start by examining their architecture role descriptions and hiring criteria. Of the roles that most organizations already have defined, product model architect roles most closely resemble the solution architect role, so begin with that role description and adapt it as necessary.
- Prioritize communication and collaboration skills when screening candidates for product model architect roles. These skills are key for solution architects. ²
- Look for business domain knowledge and experience, which can be a strong and rare asset.
Be prepared to accept candidates lacking skills that you’re confident you can train them in posthire. Hiring difficulty is high, and candidate availability is low for these roles, so place less emphasis on technical skills, which can be acquired later.

With the shift toward more versatile individuals, career development needs to change as well. Shift career development paths from a ladder to a lattice to help expose architects to a wider range of domains and disciplines (see Figure 3). This will incentivize professional development and lower attrition risks. Establish mentorship and cross-training programs. Link compensation and advancement to competency in multiple areas of architecture practice.

**Figure 3: A Growth Path Through a Career Lattice**

![A Growth Path Through a Career Lattice](image)
Another effective approach to broadening skill sets is to establish communities of practice (COPs) that span the organization. Even versatilist architects will always have particular areas of interest and technical strength. Communities of practice provide a venue in which those strengths can be shared with others who want or need to learn. Smaller organizations may have a single architecture COP, while larger organizations may have COPs dedicated to particular skills or technical domains. Since COP membership is not restricted, many of them include enterprise architects, product team members and participants from other areas of the business. This allows for knowledge and practice sharing without compromising on reporting lines. Eventually, many development teams gain enough architecture knowledge to take on some routine pieces of architecture work themselves, allowing architects to focus on higher-value activities.

**Promote an Agile Way of Working**

Many legacy architecture practices simply don’t work in an agile environment. Product model architects will find that both their role within the team and the ways they provide architecture guidance will change.

Although the specifics vary from method to method, it’s common to consider three aspects of agile software delivery, each of which is represented by a role on the team (see Figure 4). Different roles represent a given aspect at a given level of the product model, but all three aspects are always represented. Those in roles such as product owner and product manager represent the problem, which is to say that they focus on what the solution needs to do in order to bring value to its users. Representing the process — that is, ensuring that the team can execute its way of working effectively — are roles like Scrum Master and release engineer. Finally, developers and architects represent the solution. They concentrate on how the solution will deliver the needed value through its design, structure, features and code. Software engineering leaders need to ensure that architects understand their product model roles in these terms.
Many architects are accustomed to specifying the design and nonfunctional requirements of the software before development begins. The change and adaptation inherent in agile methods mean that agile teams won’t be able to consume such upfront architecture. Instead, architects must:

- Provide input a little at a time, at the team’s pace, working with the team throughout the development cycle rather than just at the beginning and end.
- Engage the team in architecture so that their way of working, their testing, and their definition of done incorporate architecture practices.
- Work with product managers to include and prioritize nonfunctional issues in the product roadmap and backlog as they evolve. This can include adding acceptance criteria to functional backlog items.
- Guide teams and designs based on well-understood and consistently applied architecture principles to ensure that products are interoperable and that they meet enterprise needs. Although these principles are usually defined by portfolio or enterprise architects, they are frequently refined based on feedback from the product and platform architects who apply them.
Evidence

1 2020 Gartner Application and Product Leadership Changes Survey. The 2020 Gartner Application and Product Leadership Changes Survey was conducted online from 10 August through 18 August 2020 with 100 Gartner Research Circle members — a Gartner-managed panel. Participating organizations were required to have fully adopted or plan to adopt (either fully or partially) a product-centric model for software delivery. The survey was developed collaboratively by a team of Gartner analysts and was reviewed, tested and administered by Gartner’s Research Data and Analytics team.


Note 1: Organization Chart and Corresponding Project Portfolio

There are countless variations, but a common organization chart and corresponding project portfolio structure is shown in Figure 5. This is seen in application organizations that use the system integrator archetype. Each architecture domain (as described in The Open Group Architecture Framework [TOGAF]) has a lead and a number of practitioners, all under the overall leadership of a chief architect. When a project requires architecture work, a temporary, ad hoc team of architects (example shown in orange in Figure 5) is assembled from across the domains. The exact composition of the ad hoc team is a compromise between who's available and which level of skill in each domain is required. The most senior architect on the ad hoc team often serves as its leader and solution architect.
Figure 5: Project Model and Corresponding Traditional Architecture Organization

Project Model and Corresponding Traditional Architecture Organization

Redefining Architecture's Role in Digital Delivery
How to Build Successful Communities of Practice for Knowledge Management
Designing the Application Organization: An Overview
EA's Role in Product Line Management
Enterprise Architecture Adapted for Product Lines (The Hanover)
Create Career Lattices to Boost Talent Development and Drive Agile Transformation at Scale
Adapt Your Application Architecture Practices to Work Better With DevOps Teams
Case Study: New Architecture Domains for Scaled Agile (Citizens)

© 2021 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. It consists of the opinions of Gartner's research organization, which should not be construed as statements of fact. While the information contained in this publication has been obtained from sources believed to be reliable, Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although Gartner research may address legal and financial issues, Gartner does not provide legal or investment advice and its research should not be construed or used as such. Your access and use of this publication are governed by Gartner’s Usage Policy. Gartner prides itself on its reputation for independence and objectivity. Its research is produced independently by its research organization without input or influence from any third party. For further information, see "Guiding Principles on Independence and Objectivity."