

The Challenges of Agile Software Development Costing and Capitalization: Overcoming a Critical Barrier to Scaling Agile Success



planview™ 

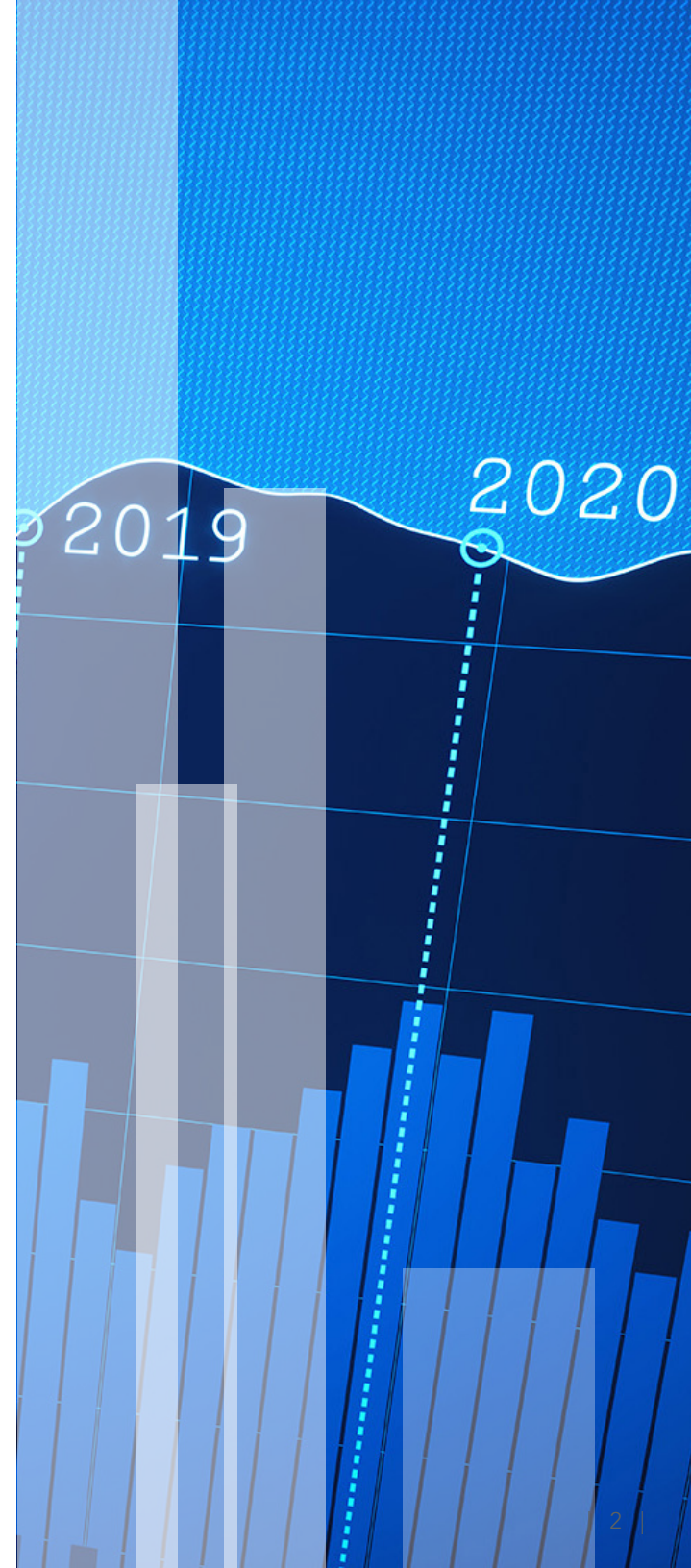
“Every company is a technology company, no matter what product or service it provides.” To be a technology company, and to stay relevant in the mind of the customer, many organizations are adopting Agile software development methodologies and/or are looking to scale their current Agile teams to accommodate the need for faster product or software delivery. While it’s often clear to many C-suite executives and development leaders that Agile is the key to future business success, historically, finance leaders aren’t as easily convinced. Why? Generally, Agile development has uncertain accounting rules and unfamiliar capitalization start and end windows, so finance would rather expense all Agile software development costs than change their accounting practices to accommodate a much faster delivery engine.

The inability for finance to adjust financial reporting and governance practices constrains Agile scaling by forcing development efforts into outdated capitalization methodologies that misinterpret Agile costs. Without a hard conversation about how Agile work is capitalized, growing Agile organizations will continue to face an uphill battle when securing more budget and headcount for their Agile delivery teams. No one wants to learn their answer for increased value delivery, faster time to market, and

better customer satisfaction is a pipe dream because of how finance views Agile development efforts. But, when faced with constant bureaucratic and financial governance headaches around headcount, budgeting, staffing, and funding, many development leaders question whether Agile is worth the hassle. In reality, the customer value delivered through Agile development should outweigh finance’s desire to “keep things as they have always been” – and shouldn’t inhibit organizations from adopting the best methods for delivering customer value.

It’s time for Agile leaders and finance teams to come together to understand and discuss how to fund, budget, and manage the costs associated with Agile software development work or face the consequences of stalling Agile scaling efforts.

In this eBook, we’ll look at why capitalizing Agile software development is challenging and the implications this has on the way Agile teams are funded and staffed. Both finance and Agile development leaders will learn how to properly evaluate Agile software development efforts, where developer time sheets fit into the equation, and why defining capitalization rules is critical for scaling Agile success and quite possibly the future of your organization.



What Does Capitalization Mean?

Capitalization (CAPEX) refers to how an organization expends or depreciates its investment costs over an asset's lifetime. For an asset to be capitalized, whether that is a product or service, it must bring long-term value to the company as a tangible production of value. For costs outside of capital investments, the term "expenses" is

typically used. An "expense" refers to something an organization spends funds on immediately and cannot be depreciated over a longer time period. This is also referred to as operating expenses (OPEX).

Why is Costing or Capitalizing Agile Software Development Important?

As organizations implement Agile practices more widely in software development, capitalizing their efforts accurately becomes paramount to successful fiscal planning and overall Agile transformations. Additionally, knowing what to capitalize versus what to expense impacts an organization's tax liabilities and profitability.

Unlike waterfall or milestone driven work, Agile software development does not follow linear processes or gates. Financial planners and accountants often find themselves unsure of how to appropriately attribute Agile software development costs to the correct CAPEX or OPEX categories, leading many to expense all development efforts up front (OPEX), making Agile costs appear very expensive to the company and its investors.

When finance inaccurately expenses Agile software development costs it negatively impacts the business by:

- Under valuing the profitability of the organization by showing Agile assets as expenses without considering their longer-term value to the company.
- Showing negative implications on taxes, resulting in potential overpayment and undervalue of the company within the year in which the expenses took place.

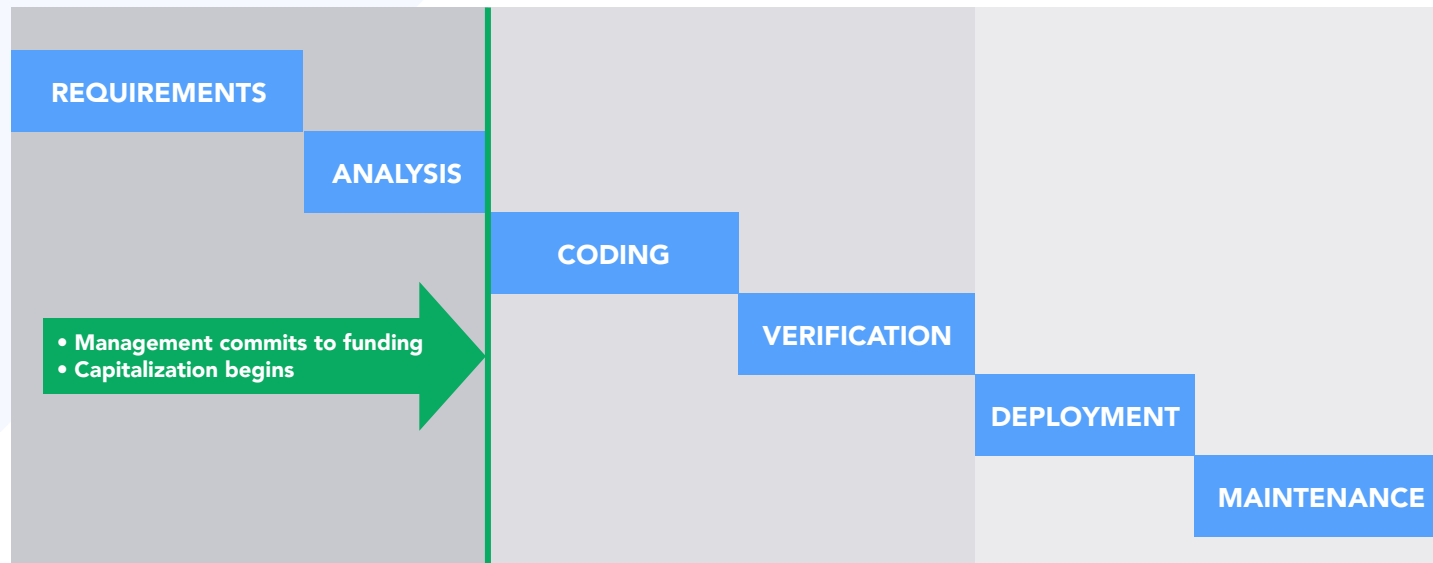
Without careful analysis of capitalization procedures, Agile software development can potentially appear less profitable and more expensive than other methodologies, resulting in a decrease in headcount and budget, as dollars are used to "pay" for Agile expenses.

Agile vs. Waterfall Development Capitalization

A few years prior to the Agile Manifesto, in 1998, the first set of financial guidelines were created around *waterfall development methodologies*. In these guidelines, software capitalization rules were established following very specific development phase gates: requirements, analysis, coding/development, verification or testing, deployment, and maintenance. In waterfall software development, significant requirements are gathered up front, and software is built in a sequential and logical order, progressing through the gates mentioned. In this model, the start of capitalization is easily triggered once highly detailed project requirement documents are created to illustrate product feasibility; this coincides nicely with the end of the requirements and analysis phases. Once a working product

is released to customers (typically 12-18 months after the initial requirements and analysis phases), capitalization ends. These clear stage-gate parameters offer the precision finance requires to provide defensible data to potential auditors.

In the case of Agile software development, finance leaders are less sure of the what and more importantly, when to start and end the capitalization of development costs. With Agile teams, for example, many of the development phases (mentioned above) are done concurrently. Since value (a working product or working piece of software) is delivered in an ongoing and iterative basis, determining where capitalization begins and ends is challenging.



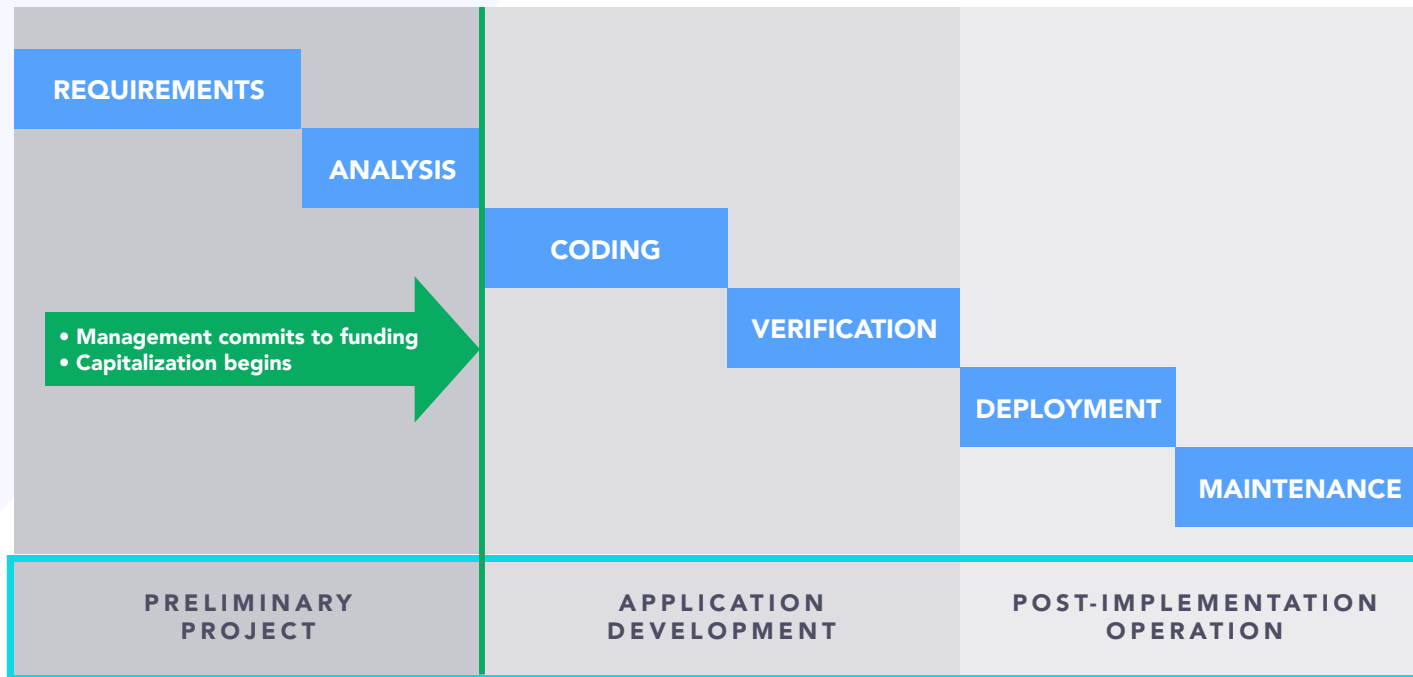
Some Agile organizations have started to identify and map the waterfall phase gates to their Agile development processes by leveraging their country's accounting rules (GAAP or IFRS). Following the respective guidelines, Agile development is mapped into 3 stages:

1. Preliminary project,
2. Application development, and
3. Post-implementation operations.

The preliminary phase is roughly aligned to the requirements and analysis phases of waterfall development, however, there are some significant differences. Namely, the amount of time it takes Agile

development "projects" to exit this part of the development phase. Leaders often utilize Lean business cases to create go, no-go decisions to procure technological feasibility. These business cases are void of extensive detail, allowing for quick decision making, often resulting in Agile development projects that can start quickly. It is from these light-weight business cases that budget is granted and development (and capitalization) begins.

Subsequently, while capitalization costs end when the development work is completed (tested and verified), it is also important to track the release date or go-live date. Some organizations may hold development efforts until they can release at one time, enabling organizations to start depreciating what was capitalized during multiple releases at once.





Imperfect Costing and Capitalization Methods

Determining when to capitalize Agile software development is an interesting and critical challenge. However, it isn't the only challenge. Once an organization has determined their capitalization guidelines, the daunting task of collecting and collating capitalization data begins.

To get a clearer picture of what Agile truly costs, many organizations subscribe to time tracking processes to ensure proper capitalization vs. expense categorization. Time tracking is typically a manual process managed by the individuals doing the work (i.e. your software developers and engineers), and the time spent on development efforts is captured in a time tracking system (time sheets). Unfortunately, due to the manual and human nature of the effort, time tracking is only as accurate as the timekeeper, and it is estimated that ["people who track their time weekly are 47% accurate. Meanwhile, those who prepare their timesheets less than once per week are only 35% accurate."](#) Ideally, developers should track their time every day against the stories they are working on to create the most accurate representation of their time as possible. But, let's be honest; few software developers and engineers are doing this daily, relegating this task for 4pm on a Friday. And even in a perfect developer time-tracking scenario, most organizations would love to recapture and trade administrative time spent collating data for more value creation, productivity time.

To remedy the inaccuracies associated with manual time tracking and to regain lost productivity time, some organizations attempt to achieve more accurate capitalization costs by adhering to fixed Agile project team costs. One of the key criteria for [building a high-performing Agile team](#) is to build stable, persistent teams. Persistent teams could allow an organization to associate a fixed cost to the team (through story point value and/or by averaging the number of hours worked in a particular day or week) and capitalize their work accordingly. This would enable a standardization approach to ensure that financial leaders get a consistent and defensible number.

While Agile rests its laurels on stable teams, that is not the reality of how all Agile teams operate, especially in organizations that are still learning how to scale Agile and are coming from a traditional project management environment. In organizations still highly project-oriented, people move to the work, the work doesn't move to the people. As a result, it is highly probable that only half of the team will stay together for longer than a few program increments. The truth is that many times Agile team specialists – architects, UX designers, QA members, testers or those tied to software integration details – move in and out of teams more frequently than we'd like, lending their expertise to teams on an as-needed basis. For organizations still costing projects on a more traditional case by case, one by one basis, this level of time management and resource tracking is cumbersome.

Value Streams and Automation: The Answer to Agile Software Development Capitalization

Capitalizing Agile development has its many hurdles. Some are due to human nature, and some challenges are because of the way the organization thinks about projects in general. In organizations that are seeking to increase Agile adoption, building stable Agile teams should be a primary goal; however, that is often not where organizations start. As referenced earlier, organizations that follow a more traditional approach to portfolio and/or project management, estimate headcount and project costs on an annual basis, typically for a 12-18 month project development cycle. Historical budget and actual cost data are used to create new planning budgets each year, and each individual project has a detailed cost analysis and requirements document behind it to justify and validate the spend request.

While this level of detail and justification is compelling to the PMO, it is dangerous to an organization from a rationalization and a capitalization perspective. Why? For a couple of reasons – because project requirement gathering can take an exorbitant amount of time to complete, and the time (read: salary) expended to create a detailed plan is often expensed. And despite the effort put forth,

these time intensive project plans are typically an inaccurate reflection of what actually happens during the project lifecycle. This project-centric approach to planning leaves plenty of room for overlap, duplicative work, wasted time, and can contribute to inaccurate accounting when people are moved from project to project, as is often the case in year-long plans.

A different approach is to consider budgeting, funding, and planning at the portfolio and value stream (product level), rather than project by project. In this approach, portfolio epics are analyzed, and epic owners create Lean business cases for high-level evaluation of the proposed priorities. The Lean business cases provide enough detail to prove technological feasibility, which triggers budget and capitalization start. This process happens quicker than waterfall methodologies, allowing the value streams to begin development faster. Organizations use this model to continuously fund and prioritize all epics and their subsequent features based on the KPIs or outcomes set forth at the strategic level. This ensures less duplication, wasted work and time across the value streams' teams.





This approach to budgeting and funding the value stream or product lines, helps ease some of the capitalization and Agile costing burdens. The Agile team's work is capitalized at a much higher level in the portfolio (the value stream), with objectives or goals visible and prioritized via the Lean business case model. Because of this visibility, organizations find they can capitalize more of their Agile software development work. This is because feasibility is done on a much larger scale – reducing the waste spent on heavy requirement-laden documentations and the long process to gain approvals.

While value stream planning and funding may not be a reality for everyone, there are other ways to ensure better Agile software development capitalization. As discussed earlier, organizations use time tracking to determine Agile software development costs, and subsequent capitalization. However, this is not ideal as many developers don't accurately input their time and asking the developers to spend time on a menial task, such as filling out their time sheets, takes away from the time they could be spending on creating customer value.

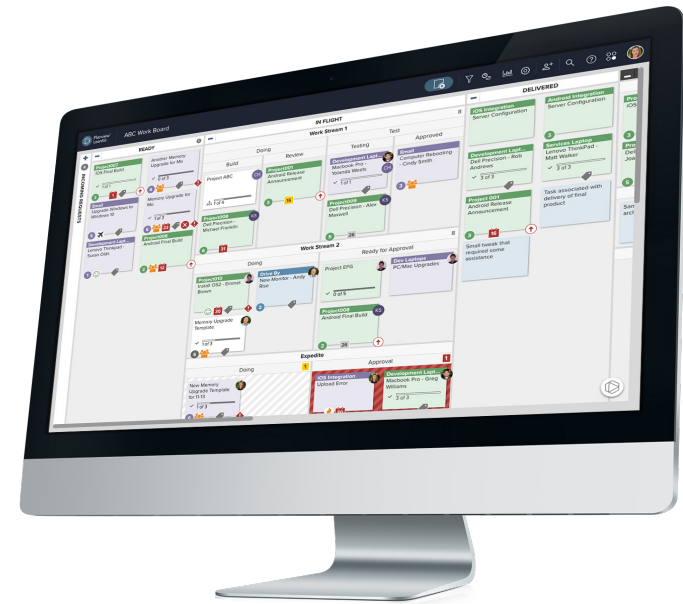
Automating the capture of Agile costs and removing the tax of manual time tracking can solve the problem and provide finance with an auditable way to capitalize Agile software development costs. By utilizing a system that automatically tracks the amount of time developers spend on a story, feature, and corresponding epic, organizations can get a realistic idea of the value delivery of their Agile teams. Using this method will require a solution that can take in the work of disparate Agile teams, apportion their time accordingly, and then rollup the data to a robust portfolio management system.

With this information, Agile and Finance leaders alike, can better understand the true impact their Agile teams have to the bottom line, their contributions to the strategic objectives, and how to capitalize Agile software development costs to ensure proper CAPEX vs. OPEX categorization. Further, this level of reporting would ensure that the Agile teams get the proper level of funding and budgeting support for future endeavors and reduces time spent filling out and reconciling time sheets.

Agile Costing and Capitalization with Planview

While Planview cannot provide guidance on how your finance department views certain aspects of your development process, we can make the process for costing and capitalizing your Agile software development more streamlined and automated.

Planview's Lean and Agile Delivery solution captures software development in real time, automatically tracking work in progress at the story, feature and epic level, to reflect time spent on product and/or software creation. Obtaining this information systematically, allows your organization to more easily capture Agile software development capitalization costs and reporting, and removes the need for manual reporting and reconciliation of time sheets, returning development time back to the business.



To learn more about our approach to costing Agile development, email us at Lean-Agile@planview.com or visit us at planview.com/lean-agile-delivery.

