



# Going Beyond Traditional Contracts for Agile Projects

## INTRODUCTION

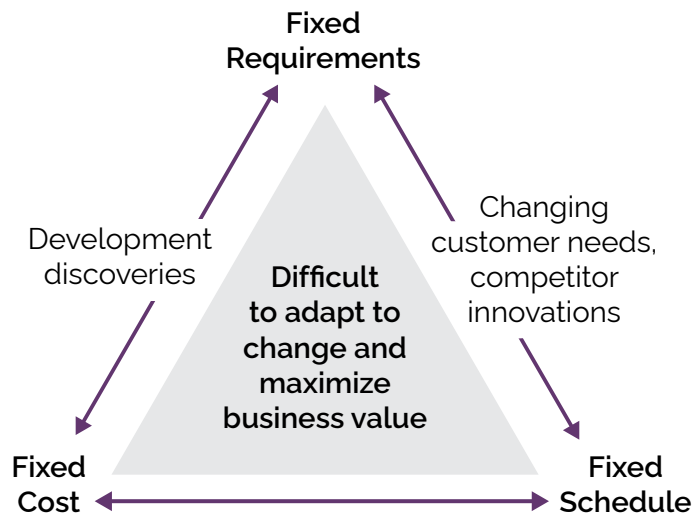
Agile development and delivery practices have become a popular way to implement and evolve software. Instead of pre-defining all the detailed project deliverables, budgets, and phases up-front, Agile methods promote short development cycles, frequent feedback loops, and incremental deliveries of working software every couple of weeks (i.e., sprints) to accelerate value delivery to the client. Agile approaches place a premium on trust, shared collaboration, and risk sharing between the client and the software provider. This joint framework helps with providing value-driven prioritization of work items, detection of quality issues early, and responding to changes as new information emerges.

## FIXED-PRICE CONTRACTS MAKE IT MORE CHALLENGING TO ADAPT TO CHANGE

In today's fast-paced, competitive business environment, needs change frequently, technology evolves constantly, and new and existing competitors innovate quickly to wrest away market share. Traditional fixed-price contracts attempt to manage risk by attempting to specify all known work up-front and tying it to rigid project schedules and pre-determined milestones. This process often does not effectively account for, nor support, frequent collaboration, feedback cycles, and emergent changes that are the hallmark of Agile development processes. Attempting to capture all detailed requirements up-front, and committing to fixed budgets too early, artificially constrains Agile teams when requirements change, or new information emerges. Clients and software providers often seek to avoid "scope creep", even if new requirements should supersede older requirements. Change through formal change orders is often difficult, expensive, and tends to be avoided to keep the project on time and within budget. The risk and cost of change fall primarily on the software provider which has the incentive to avoid changes if these negatively impact its profit margin or relationship with the client.

Quality is often well-defined for the completion of the project but is not monitored actively enough by the client during the project, sometimes leading to unexpected, negative surprises later in the project. Additionally, traditional methods of developing the software first and then testing it may elongate the development cycle, as may the commonly used separate client-testing cycles (i.e., verification, validation, and user acceptance testing). Ideally, shared test environments can be set up in which the software vendor and the client have access for frequent and fast validation of promoted code. Generally, communications between clients and software providers are formal, but not frequent enough and are lacking adequately defined roles. These are needed to help ensure that the highest quality software is being developed for increased business value.<sup>1</sup>

### Concerns of Fixed-Price Contracts



- Setting budget and scope boundaries is a strength of fixed-price contracts
- Quality and business value are often not actively optimized enough during the project
- Distracting, sometimes confrontational, relationships arise over project scope
- Being too focused on process formalities rather than on clear vision and shared responsibilities and business outcomes can sabotage the relationship
- Can discourage necessary change, innovation

## THOUGHTFUL AGILE CONTRACTS CAN LAY THE FOUNDATION FOR SUCCESS

Agile contracts should still provide adequate commercial protections for the client, but it is important to kick off Agile programs with agreements that define and reinforce the necessary, high level of trust and collaboration between the client and the software provider. Instead of spending a considerable amount of time pre-determining project specifics with limited information, Agile helps the parties to focus instead on the high-level goals, initial development backlog of priorities, and engagement model and interactions between the software provider and client that can help achieve the desired business outcomes. The Agile contracting process helps the parties to answer questions like:

- What does success look like (i.e., is the vision clear to everyone)?
- Will the Agile team be a joint team of client and software provider resources (i.e., will the client provide product owners or other team members)?
- What are the shared responsibilities and expectations among the client and software provider to help provide a high level of communication and collaboration?
- How long will the development sprints be?
- Will testing take place within or outside sprints, and how and in which environment will client testing occur?
- Which party will be doing what activities during the requisite Agile ceremonies?

The Agile contracting process also can make backlog prioritization or other project decisions easier and clearer, like empowering the product owner (i.e., the project management lead at the client) to make most daily decisions.

It is important to clearly define the processes, roles, quality standards, and thresholds to avoid misunderstandings.

## IMPORTANT DEFINITIONS/CONCEPTS FOR QUALITY IN AGILE CONTRACTS

The following table illustrates some important definitions and concepts related to Agile contracting:

Definition/Concept	Examples
Ceremonies and Roles	What is a weekly standup meeting, a scrum master, a backlog, a user story, an epic, a product owner, a developer, and a tester?
Definition of "Ready"	When is a user story (i.e., a description of what the person using the product should be able to do) ready for the team to begin working on it? What are the requirements for this?
Definition of "Done"	What criteria constitute a product being ready to go into production? How do we know the story is done?
Definition of "Acceptable Level of Defects"	What level of defects is acceptable post release?
Approach for holding the software provider accountable for quality	Are the redress approaches proportionate to the scope of work? Does it include the client also meeting their obligations to help ensure quality?

The Agile contracting process requires that the client and software provider share the risk of a project to foster effective collaboration and positive project outcomes. Contracting approaches should fall somewhere between pure fixed-price contracts that place much of the risk on the software provider, and pure time-and-materials contracts that place much of the risk on the client.

Agile development processes have become popular because they emphasize increased speed to value for clients and may significantly increase quality outcomes through frequent feedback loops and tight collaboration between the parties. Traditional fixed-price contracts effectively provide project boundaries around costs, timelines, and requirements, but also inhibit the ability to adapt to change and increase business value as new information arises during the project. A new approach to Agile contracting has become necessary that focuses on the processes and roles used to develop the software, defines quality standards clearly, effectively shares risk between the software provider and client more equally, and fosters true partnership and collaboration to help achieve the best outcomes.

## ABOUT VITECH

Vitech is a global provider of cloud-native benefits and investment administration software. We help our Insurance, Retirement, and Investment clients expand their offerings and capabilities, streamline their operations, gain analytical insights, and transform their engagement models. Vitech employs over 1,400 professionals, serving the world's most successful insurance, retirement, and investment organizations. An innovator and visionary, Vitech's market leadership has been recognized by industry experts, such as Gartner, Celent, Aite-Novarica, and ISG. For more information, please visit [vitechinc.com](https://vitechinc.com).