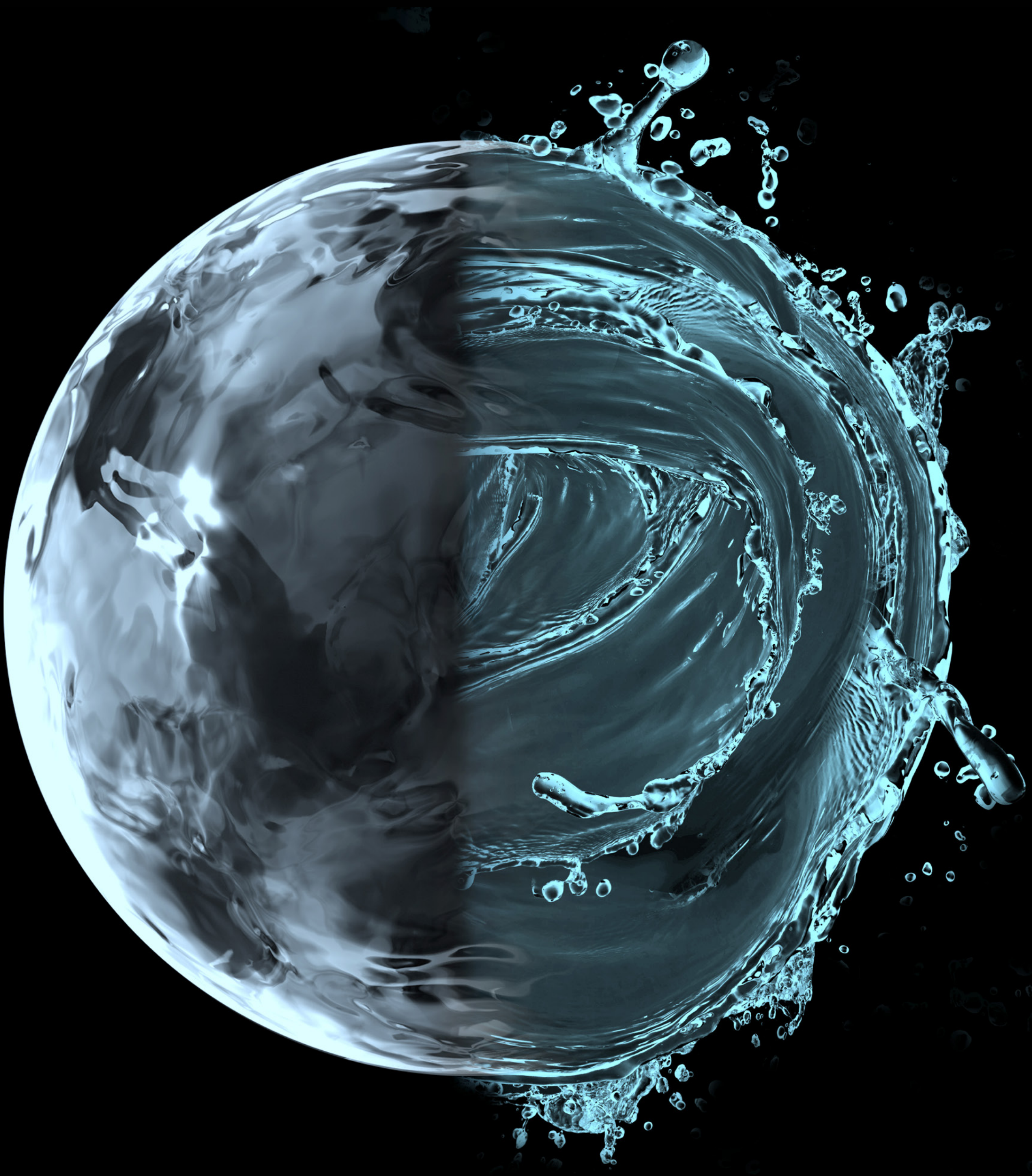
Deloitte.



Auditing Agile projects

Your grandfather's audit won't work here!

Auditing Agile projects

For many companies, Agile methodologies are beginning to **gain ascendancy over traditional Waterfall development.** With their focus on speed, adaptability, and continuous iteration, Agile projects can present opportunities for Information Technology (IT) departments and challenges for Internal Audit (IA) teams tasked with deciphering whether risks are mitigated.

This has led to a **number of misconceptions** about auditing Agile and no small amount of confusion (see "Common myths about auditing Agile").

Common to these misconceptions is the belief that Agile projects are somehow "free-for-alls" that lack any type of rigor or formal processes—something that is guaranteed to make them more risky than traditional software development initiatives and throw a monkey wrench into any attempt to audit them. Yet the **reality is quite the opposite.** Agile projects present the same inherent risks as traditional projects. What differs is the Agile process itself and, therefore, how risks are addressed and mitigated. For that reason, as auditors, IA teams need to take a step back and switch lenses—and as with Agile projects themselves, teams need to adopt a different approach.

Common myths about auditing Agile

Myth No. 1: Agile teams can do whatever they want

The reality: Agile actually builds controls directly into the development process that the team follows. The concept of acceptance criteria is one example. For each user story (activity), the team will define the criteria that determines when the story is complete and working as expected.

Myth No. 2: Agile projects produce no documentation

The reality: On the contrary—you just need to know where to look. True, you are not going to find the same stage-gate documentation. Rather, you will find documentation embedded within user stories. Evidence of stakeholder sign-off may be found in a sprint review meeting. When adopted well, Agile development projects produce more relevant and usable documentation.

Myth No. 3: Agile projects do not follow project management practices

The reality: Agile simply adopts a different approach to project management, but objectives are the same as with traditional methods. Take status updates, for example. Agile may not call for sit-down status meetings, but project status is captured on the visual display/tool in real time, as well as in daily "stand-ups" where teams assemble briefly to discuss the work for the day and update the board. The need for a single project manager is expelled in Agile because the team is self-organized and there is more granular management of the work.

Take a deep breath: The inherent risks are the same

Agile and Waterfall projects both face the **same set of inherent risks,** ranging from undetected problems with functionality to a failure to meet stakeholder needs. What differs between the two approaches is the development process, including the frequency of delivery, the team structure, and organization of the work (see "Characteristics of Agile"). Therefore, how those risks are mitigated and where IA looks for evidence that a control is in place would also change. Consideration of new project controls that leverage an understanding of how Agile has been implemented in an organization leads to efficiencies and more effective risk mitigation.

Like an audit of a traditional Waterfall project where the auditor reviews checks and balances that have been built into the process, Agile projects also have **logical control checkpoints.** Typically, the auditor will review the Waterfall system development lifecycle (SDLC), which outlines the system development process the company has adopted. Similarly, a company utilizing an Agile approach would typically have similar documentation outlining the process it is using.

The difference is Waterfall projects have regular stage gates that occur in a linear and sequential fashion, while Agile projects are iterative in nature, which may **change the timing of controls, as well as how they are executed.** This leads to the next consideration.

Characteristics of Agile

Agile development methods come in a variety of flavors, and although the specifics may differ, the approaches all share some common characteristics:

- Teams work in “sprints”—time-boxed intervals of several weeks
- Work is broken into small increments referred to as “stories”
- Work is ordered based on business priorities
- Stories move from start to finish (e.g., completed piece of software) within a sprint
- At the end of each sprint, completed work is demonstrated to stakeholders
- Agile teams are facilitated by a scrum master who helps to ensure the process is followed
- Frequent and ongoing collaboration with customer

Agile may provide more comfort

One of the most prominent features of Agile projects is the granularity of the work involved: Sprints focus on the start-to-finish delivery of a single software feature. This has some important benefits when it comes to risk and performing the audit—namely that controls can be more **precise and tightly managed.** For example, consider the stakeholder sign-off control. When software is developed using a traditional Waterfall approach, the go/no-go decision occurs at the very end of the project. It is rare that certain pieces of functionality would be deployed while others are held back. When review occurs at the end of development, stakeholders have a wide range of features to look at, and a lot can fall through the cracks or surface much later. With Agile, stakeholders are providing feedback for a single aspect of the product. This means both user testing and resulting feedback are highly focused and much more likely to zero in on any problems.

When work is arranged into smaller, regularly completed chunks, there is less potential for errors or problems that arise to affect the overall project. In addition, teams are learning during each iteration and **adding value** to both the process and the product as a result. They are also reprioritizing and refining what is needed to achieve a product that is aligned with stakeholder needs. More frequent deployments focus the team on a smaller portion of the overall development effort, allowing for refinement and a change in priorities if required. Furthermore, because stakeholders are involved in each deployment, there is less risk that the final product does not meet the business need or that functionality is not working as intended.

Strategies for auditing Agile projects

When auditing Agile projects, IA teams may need to **think differently**—whether this means recognizing a different set of controls, changing where to look for evidence that controls exist, testing an ongoing control, or helping the team gain even more operational efficiencies.

The controls for Agile projects will be **different** because the frequency, evidence, process and governing policies, and precision will all have changed.

For example, one of the most prominent Waterfall controls used to mitigate the risk that the functionality is not working as intended is the final stage gate, review, and ultimate go/no-go decision, discussed previously. Historically this control happens once—after testing and prior to the big bang deployment. With Agile, this control will occur much more frequently because there will be deployments throughout the project. The evidence of a stakeholder decision may not take the form of a final written sign-off. Instead, it could include documentation in user stories, meeting minutes, check boxes, or notes on the story. The Agile team will have defined acceptance criteria for the story, which can also give insight into how they are determining when functionality is ready for deployment, something that is important for the audit team to understand. An appropriate audit step may then be to corroborate with the stakeholder.

In Waterfall projects, another common example of a control that mitigates the risk that the delivered software does not meet the business need is the review and approval of business requirements. The auditor will typically review the approval but also validate that those requirements carry through the remaining phases of the project (specifically, build, test, and issue resolution). However, with Agile, those requirements may change and evolve throughout the project, and the auditor will need to understand the process for incorporating those changes.

Given the iterative nature of Agile development, audit teams should consider how they **risk-rank** and sample controls. IA teams may not be able—or even want—to look at every persona or user story, and the reviews and sign-offs won't apply to the entire product. Instead, the auditors may choose to limit the audit to specific higher-risk sprints. Given this difference, risk should continue to be top of mind. That includes the auditor providing a point of view and consideration of controls being designed and built into the system being implemented, as well as the applicable new or evolving process, with the difference being that in an Agile project, only the minimum viable product may be deployed at any given time. The auditors will need to consider the risks and applicable controls related to that functionality and continue to include those considerations within the audit plan.

Finally, it should be recognized that moving from Waterfall to Agile is an **organizational change** that has both a technical (knowledge of Agile) as well as an adaptive (change management and people) component. As internal auditors, assisting in both aspects of the transformation is important. To do this, a solid understanding of how the team is organized and their level of Agile maturity is necessary. This can provide perspective on the effectiveness of Agile programs and can help the organization obtain the benefits of this new way of working.

Amp on Agile!

The goal in auditing software development projects is to help teams be more effective and efficient and to appropriately mitigate risk. When auditing, the intent is to **add value**, not hinder the pace of a project. For Agile projects, there are numerous opportunities to achieve these goals throughout the development process, which is why it makes sense to bring the IA team on board at the beginning of the project rather than at the end, when it will most likely be too late.

But to be truly effective, auditors should consider **taking a page from the Agile** playbook in the design and approach to the audit itself. If the software development team is working in an iterative way, it makes sense that Internal Audit's recommendations or viewpoint should be iterative and dynamic as well.

Flexibility and adaptability need to imbue the approach. There may be certain sprints, areas of functionality, or aspects of the project that require more attention; this way, IA teams can adjust the audit plan as different priorities emerge.[1]

[1] For more information on Agile Internal Audit, please refer to our Agile Internal Audit series at www.deloitte.com/us/becoming-Agile.

Contacts

Contact the Deloitte Risk and Financial Advisory professionals listed below to discuss the approach to auditing Agile projects at your organization:

Sandy Pundmann
US Managing Partner, Internal Audit
Deloitte & Touche LLP
+1 312 203 7000
spundmann@deloitte.com

Sarah Adams
Managing Director
Deloitte & Touche LLP
+1 713 982 3416
saradams@deloitte.com

Ranjani Narayanan
Senior Manager
Deloitte & Touche LLP
+1 617 437 3847
rnarayanan@deloitte.com

Kristen Heikkinen
Senior Manager
Deloitte & Touche LLP
+1 617 437 3488
kheikkinen@deloitte.com

Christopher Pattillo
Manager
Deloitte & Touche LLP
+1 206 716 7010
cpattillo@deloitte.com

Deloitte.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

As used in this document, "Deloitte Risk and Financial Advisory" means Deloitte & Touche LLP, which provides audit and risk advisory services; Deloitte Financial Advisory Services LLP, which provides forensic, dispute, and other consulting services; and its affiliate, Deloitte Transactions and Business Analytics LLP, which provides a wide range of advisory and analytics services. These entities are separate subsidiaries of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.