Agile GEIT—Building Trust and Maximizing Value Delivery

The Agile methodology is defined as a set of principles and values that guide software development teams toward responding effectively and efficiently to customers' needs, thereby reducing the business risk of irrelevance.[1] This focus on responsiveness to the market and maximizing value delivery to the company has fueled Agile's widespread adoption by organizations. However, Agile's responsiveness to the market represents and meets only half the organization's Agile process stakeholder needs—those of the business, finance, IT and the customer. It overlooks the organization's compliance and security stakeholder needs to manage IT risk.

Today's ever-increasing legal and regulatory requirements place more onus on organizations to exercise due care in protecting and controlling the Agile development process. This emphasis can distract the Agile development team from being responsive to the market and maximizing Agile's value delivery to the organization.

To rectify this situation and break the downward spiral of Agile's value delivery caused by the ever-growing risk management trust effort requires a governance of enterprise IT (GEIT) system[2] concerned with IT value delivery to the business and the mitigation of IT-related risk.

Agile's inherently fluid environment created by its values and principles and by its supporting software tool set is useful to understand. It is also worth exploring Agile's stakeholders' requirements for security, compliance, assurance and agility and the seemingly contradictory nature of those needs in relation to Agile's fluidity. The challenges that Agile's fluidity and the contradictory stakeholder requirements present to the Agile GEIT (A-GEIT) system are detailed herein, and examples of how these challenges manifest themselves in the day-to-day Agile development process are offered.

Also introduced is the A-GEIT implementation guidance which is presented in a series of *COBIT Focus* articles. The A-GEIT implementation guidance follows a phased approach overlaid by three distinct IT risk/value delivery efforts to guide practitioners in planning and implementing an A-GEIT system that both accommodates Agile's fluidity and meets its contradictory stakeholder requirements to find the balance between managing Agile's IT risk and maximizing its value delivery. The guidance is not meant to be a "silver bullet," but rather assistance for practitioners in using and adapting COBIT® 5's enablers to the unique set of Agile governance challenges.

The A-GEIT implementation guidance focuses on planning a GEIT system to manage IT risk by building an internal control system using information security requirements to protect and control the Agile development activities and build trust in its ability to safeguard its assets. It also seeks to maximize Agile's value delivery by designing internal controls into the Agile process so its compliance and control assurance generation has minimum impact on Agile's responsiveness to the market.

Michael Bergman, CRISC, CISSP
Is a consultant working in the overlap where IT risk meets information security. Bergman has a wealth of experience functioning across the first and second lines, defining and implementing internal control systems. He passionately believes that behind every good control system is an even better implementation plan and execution team.

This series of articles (here and in *COBIT Focus)* discusses Agile governance, offering practitioners tips on building trust to maximize value delivery. These tips focus first on guiding practitioners in using COBIT's information enabler to plan an Agile internal control system. Second, they focus on guiding practitioners in using the processes and service, infrastructure, and applications enablers to maximize value delivery by designing internal controls in the Agile process so its compliance and control assurance generation has minimum impact on Agile's responsiveness to the market. The *COBIT Focus* articles outline a phased approach to building information about the organization's specific Agile implementation and using that information to produce a customized plan to implement the Agile internal control system.

Each of the phases and tasks provides guidelines on how the task could be performed in the Agile context and, in most cases, is followed by control implementation examples using the JIRA, Jenkins and Github service, infrastructure and applications enablers. The intention is that in placing each phase and related tasks in the Agile context and supporting them with a simple implementation example provides enough insight into the Agile development process to enable practitioners to apply that insight to solving more complex security requirements such as making security an inherent part of information systems.

Agile's Supporting Software Tool Set

The Agile approach is inherently fluid. It is built on the premise that competitor offerings or customer expectations can change, and failure to respond to change creates risk, including irrelevance. This is supported by Agile's values and principles such as "individuals and interactions over processes and tools," and "responding to change over following a plan."[3]

Over the years, the drive toward implementing Agile's values and principles and increasing its responsiveness to the market has spawned many approaches and as many software tools to support them. These approaches and software tools include extreme programming, rapid application development (RAD) and development operations (DevOps). Its accompanying software tool sets include Docker, Jenkins, JIRA, Github, Confluence and Cucumber. These tools and approaches have something in common: They strive to provide more flexibility, continuous design improvement and responsiveness to the market. At the risk of perpetuating the misunderstanding that Agile is DevOps, which is continuous delivery, the term "Agile supporting software tool set," as used in this article, refers to the approaches, cultures and practices that share the common goal of increasing the company's responsiveness to the market.

Agile Stakeholders

The Agile development process has multiple stakeholders from different business and IT units. From the business unit perspective, there are the finance department and the organization's customers, who are looking to Agile to be responsive to customer needs, creating the most desirable new products and services for customers, and reaping the financial benefits of doing so.

From an IT unit perspective, the stakeholder list covers developers, testers and a host of teams to support the Agile supporting software tool set. What seems to unite most of the multitude of IT and business stakeholders is their agreement on the importance of Agile's agility. This statement of unity is reflected in Agile's manifesto, which guides software development teams toward making the software development life cycle (SDLC) quick and easy and in the Agile supporting software tool set striving toward continuous delivery and speed to market.

However, this focus on Agile and its responsiveness to the market is not a complete representation of all of Agile's stakeholders' needs. Unfortunately, it all too often excludes the requirements of other stakeholders, namely the information security and compliance teams. These teams are not focused on Agile's responsiveness to the market, but rather on protecting and controlling its development activities and safeguarding its assets, which they accomplish by creating a structured and repeatable way of working that is compliant to information security standards.

There is a need to identify and work with all of these stakeholders to get a common understanding of what each needs from Agile.

Considering the Agile security, compliance and agility stakeholders, the left column of **figure 1** shows business and IT-related stakeholders and what they require from the Agile development process. The right column shows what the Agile process needs to deliver to meet stakeholder requirements.

From **figure 1**, it can be concluded that the Agile development process needs to deliver more than responsiveness to the market to meet all its stakeholders' needs. It needs to manage IT risk by protecting and controlling the Agile development process and generating trust in its ability to safeguard its assets, and by maximizing value delivery by maintaining agile's responsiveness to market.

Meeting Stakeholders' Needs

Agile's stakeholder requirements can be viewed as contradictory, sometimes seeming that achieving one can be done only at the expense of the other. The Agile deliverables to meet stakeholders' requirements contain a contradiction in the making that centers on the words "building trust."

To build trust, the business turns to the compliance and assurance functions, which assess each piece of software before it is released to production/live.

These functions assess adherence to the various security, legal and regulatory controls. For example, they may assess whether the developed code has been peer-reviewed for security vulnerabilities or malicious code and whether a privacy impact and IT risk assessment has been performed. A positive outcome to these assessments builds the trust the business needs and is a prerequisite to allow deployment to production/live.

As shown in **figure 2**, an effort to meet compliance requirements and build trust can negatively impact the Agile process and its goal of being responsive to the market.

This compliance effort usually occurs just before the Agile release phase and involves the compliance functions examining the planned release, hunting for compliance evidence within the tool set and its mass of log files.

Depending on the scope of the assessments to be done, this compliance effort can last for weeks as it moves among the different assessment teams.

Not only can this have a substantial impact on Agile's fluidity and potentially delay the planned release by weeks, but, in the worst-case scenario, the planned release could be rejected, incurring further costs to rectify compliance breaks.

Following from **figure 2**, **figure 3** presents examples of how the contradictory stakeholder requirements manifest themselves in the day-to-day running of the Agile process.

From these manifestations, it is clear that "manage IT risk" stands at odds with the effort of "maximize

Figure 1—A-GEIT Requirements

Business and IT-Related Requirements	Agile Deliverables to Meet Stakeholder Requirements
Financial department: Managed business risk (safeguarding of assets against security vulnerabilities and malicious code)	Manage IT risk by building an internal control system to protect and control the Agile development process and build trust in its ability to safeguard its assets. (Achieved through compliance to security policies, among other policies, ensuring that the assets generated in the Agile SDLC are protected against security vulnerabilities and malicious code)
Compliance department: Internal and external compliance (e.g., legal, regulatory, contractual, internal policies, procedures)	
Information security department: Security of information, processing infrastructure and applications	
Customer: Agile responses to a changing business environment	Maximize value delivery by designing the internal controls into the Agile process. (Achieved by implementing the controls so compliance and control assurance generation have minimum impact on Agile's responsiveness to the market)
Agile development team and support staff: IT agility	

Figure 2—Pre A-GEIT Control Implementation

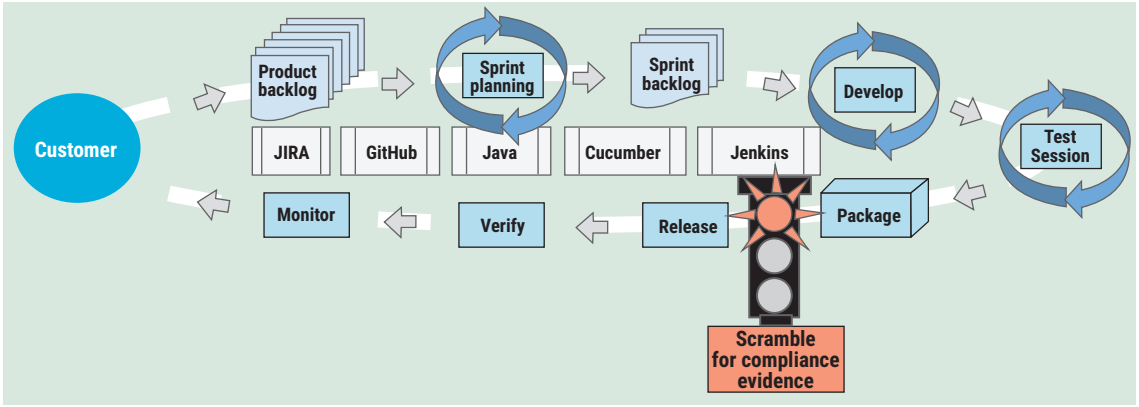


Figure 3—Agile Stakeholder Requirements

Trust in Agile to protect and control—How to ensure that appropriate approvals have been awarded and required testing has been completed before proceeding to the next phase of the SDLC when the continuous integration/continuous delivery CI/CD pipeline blurs the lines between the SDLC phases, eradicating the concept of chronological process order along with human intervention and their accompanying approvals?
Legal and regulatory compliance—In an organization with five Agile teams on a two-week release cycle, how is the EU General Data Protection Regulation (GDPR) requirement enforced, prior to carrying out potentially high-risk processing, controller is required to carry out a data protection impact assessment (DPIA), without overwhelming the privacy officer who will, along with his or her other privacy duties, soon enough become the bottleneck and interrupt the two-week release cycle?
Agile values vs. audit functions consistent way of working (WOW)—Information security requires the use of valid and measurable acceptance criteria to test information systems. How does the Agile testing team ensure relevant, measurable acceptance criteria that accurately reflect the functional requirements when Agile principles allow accepting changes in requirements mid-development?

value delivery." This contradiction comes to the fore in the phrase "build trust." The overhead of building trust through compliance and control assurance evidence generation impedes Agile fluidity and inhibits Agile's responsiveness to the market, preventing organizations from realizing the full business benefits of Agile. Left unchecked, this puts the organization in a situation of having exposed itself to all of the internal and external IT risk associated with adopting Agile, but not having managed to materialize all or some of its opportunities.

Dealing with the contradictory nature of the Agile stakeholder requirements requires a GEIT system that equally values all stakeholder requirements and is focused on IT value delivery to the business and the mitigation of IT-related risk.

A-GEIT Implementation Guidance

The A-GEIT implementation guidance uses a phased approach to implementing a GEIT system to manage Agile IT risk and maximize value delivery by maintaining Agile's responsiveness to the market.

The guidance is different from others because it is focused on the Agile stakeholder requirements of security, compliance, assurance and responsiveness and, therefore, provides technical detail on designing controls into the process. Where other guidance may glance over the implementation phase, this guidance takes a hands-on approach and details implementation tasks and examples to help operational-level staff implement the internal control system.

The A-GEIT implementation guidance uses three enablers to implement the GEIT system over the Agile process: information; processes; and service, infrastructure and applications. The information enabler is used to inform the IT risk team's decisions on where and how to implement the selected controls into the Agile development process so that it will have the least impact on Agile's responsiveness to the market. The service,

infrastructure and applications enabler is used to enforce automatic controls and generate control assurance evidence for it. The processes enabler is used to enforce manual controls and guide the Agile development team toward compliance to the internal controls.
The phases and enablers are overlaid by three distinct risk/value delivery efforts:

- Gathering information to enable informed decision-making
- Managing IT risk to build trust
- Value delivery to realize full business value (**figures 4 through 7**)

> “DEALING WITH THE CONTRADICTORY NATURE OF THE AGILE STAKEHOLDER REQUIREMENTS REQUIRES A GEIT SYSTEM THAT EQUALLY VALUES ALL STAKEHOLDER REQUIREMENTS AND IS FOCUSED ON IT VALUE DELIVERY TO THE BUSINESS AND THE MITIGATION OF IT-RELATED RISK.”

Figure 4—Informed Decision-Making Effort

This effort builds the information enabler, which is a critical element to the successful planning and implementation of a GEIT system. Each organization’s implementation of Agile could potentially be different in terms of its Agile practices, the practice outputs, the software it uses or even the human resources used to complete the practice. This effort focuses on building a picture of how Agile is implemented at the organization, recording the organization-specific implementation details of the Agile development process. This focus is based on the very simple principle that the more the IT risk team understands about the organization’s Agile implementation, the more informed its decisions will be regarding designing the controls into the process.

This effort overlays the “determine current state” phase, which, in support of this effort, goes beyond identifying existing and potential controls within the organization’s Agile process and sets out to capture the organization-specific implementation detail of the Agile development process.

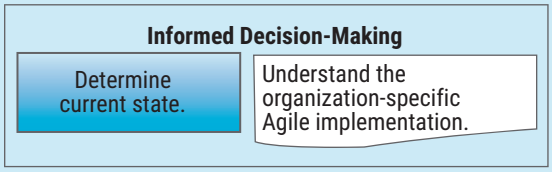


Figure 5—Agile Practice—Peer Review

IT risk team interviews the Agile development team looking for existing controls and records, beginning with the Agile practices that make up the Agile process. Second, the team records the Agile practice implementation information, e.g., “human resources in the room” and “software open on the Agile developers desktop.” This information is designed to give the IT risk team an understanding of Agile’s fluid nature and quantifying its responsiveness to the market.

Agile practices—implementation characteristics
Agile practice: Peer review
Best practices: Perform quality assurance (QA) **Resources required:** Developer, peer reviewer **Software required:** GitHub, JAVA IDE, Slack **Output:** Slack “peer review” approval message

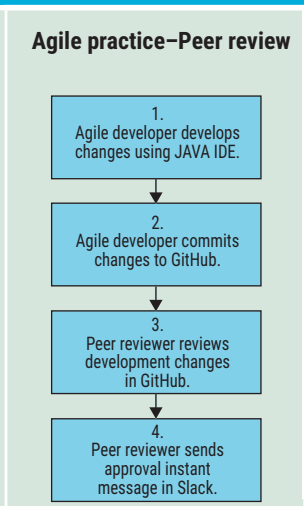


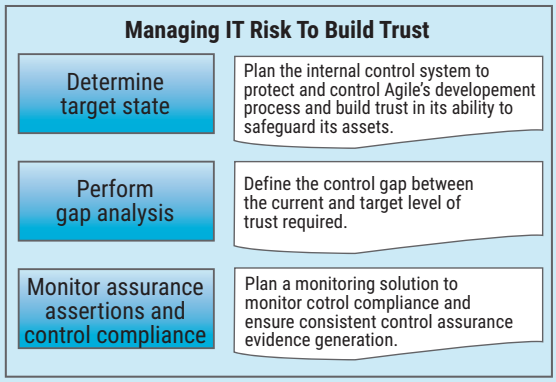
Figure 6—Managing IT Risk to Build Trust Effort

This effort focuses on building an internal control system to protect and control the Agile development process and build trust in its ability to safeguard its assets.

This effort overlays the "determine target state," "perform gap analysis" and "monitor assurance assertions and control compliance" phases.

In the "determine target state" phase, the IT risk team defines the Agile process security, compliance and agility stakeholders and requirements, then uses various sources for example security baselines, relevant internal standards or legal requirements to select the internal controls to manage the IT risk.

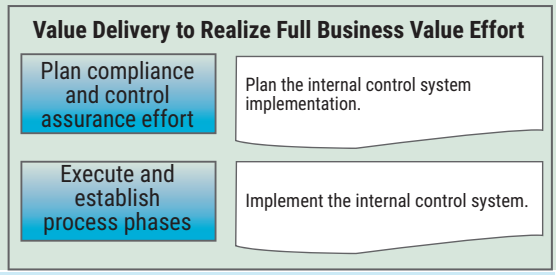
In the "perform gap analysis" phase, the IT risk team identifies which controls exist in the Agile development process, which controls need to be added and which Agile practice outputs could be fashioned into controls by generating appropriate control assurance evidence for it.



In the "monitor assurance assertions and control compliance" phase, the IT risk team creates a control monitoring solution to monitor control compliance and ensure the consistent generation and publication of control assurance evidence.

Figure 7—Value Delivery to Realize Full Business Value Effort

This effort focuses on a planned compliance and assurance drive for designing the internal control system into the Agile process so control compliance and control assurance generation has minimum impact on Agile's responsiveness to the market and maximizes its value delivery. This effort includes "plan compliance and control assurance effort" and "execute and establish process" phases and uses the information; processes; and service, infrastructure and applications enablers to implement the internal control system.



In the "planned compliance and control assurance effort" phase, the IT risk team uses the information enabler gathered earlier to embark on a planned compliance and control assurance effort to plan the design of the internal control system into the Agile process.

In the "execute and establish process" phase, the team uses the service, infrastructure and applications enabler to build the internal control system and design the controls into the Agile process. It then uses the processes enabler to produce an operational procedure to guide the Agile development team toward compliance to the implemented controls.

These efforts focus the IT risk team's attention on meeting the challenges to implementing a GEIT system to manage risk and realize its full business value.

Conclusion

Agile and its supporting software tool set bring great business value; the realization of this value is reflected in its rapid rate of adoption. For the business to extract the maximum business value from the Agile development process, it is no longer enough to define a control system that meets information security and compliance requirements. The defined control system must be accompanied by a structured phased approach for its implementation, which details a planned compliance effort always with an eye on minimizing the impact compliance has on Agile's responsiveness to market.

The A-GEIT implementation guidance seeks to guide in the definition of this structured approach to transform the Agile development process so the scramble for compliance and control assurance evidence does not break Agile's fluidity and compromise its responsiveness to the market (**figure 2**).

This compliance and assurance bottleneck is replaced by a state in which the Agile development activities are protected and controlled through compliance to information security standards and the control assurance evidence is generated contemporaneously to the Agile practice, which enables maximum value delivery to the organization (**figure 8**).

Endnotes

1 Manifesto for Agile Software Development, *http://agilemanifesto.org/*
2 A GEIT system enables the enterprise to take full advantage of IT, maximizing benefits, capitalizing on opportunities and gaining competitive advantage. Fundamentally, GEIT is concerned with IT value delivery to the business and the mitigation of IT-related risk.
3 *Op cit* Manifesto

Figure 8—Post A-GEIT Control Implementation