TOGAF® Series Guide

Applying the TOGAF® ADM using Agile Sprints

The Open Group Architecture Forum Agile Architecture Work Stream

THE *Open* GROUP

TOGAF® Series Guide

Applying the TOGAF® ADM using Agile Sprints

ISBN: 1-947754-68-3

Document Number: G210

Published by The Open Group, April 2022.

Comments relating to the material contained in this document may be submitted to:

    The Open Group, Apex Plaza, Forbury Road, Reading, Berkshire, RG1 1AX, United Kingdom

or by electronic mail to:

    ogspecs@opengroup.org

Contents

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. With more than 870 member organizations, we have a diverse membership that spans all sectors of the technology community – customers, systems and solutions suppliers, tool vendors, integrators and consultants, as well as academics and researchers.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices

- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies

- Offering a comprehensive set of services to enhance the operational efficiency of consortia

- Developing and operating the industry's premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/library.

The TOGAF® Standard, a Standard of The Open Group

The TOGAF Standard is a proven enterprise methodology and framework used by the world's leading organizations to improve business efficiency.

This Document

This document is a TOGAF® Series Guide to Applying the TOGAF® ADM using Agile Sprints. It has been developed and approved by The Open Group.

More information is available, along with a number of tools, guides, and other resources, at www.opengroup.org/architecture.

TOGAF® Series Guide (2022)

About the TOGAF® Series Guides

The TOGAF® Series Guides contain guidance on how to use the TOGAF Standard and how to adapt it to fulfill specific needs.

The TOGAF® Series Guides are expected to be the most rapidly developing part of the TOGAF Standard and are positioned as the guidance part of the standard. While the TOGAF Fundamental Content is expected to be long-lived and stable, guidance on the use of the TOGAF Standard can be industry, architectural style, purpose, and problem-specific. For example, the stakeholders, concerns, views, and supporting models required to support the transformation of an extended enterprise may be significantly different than those used to support the transition of an in-house IT environment to the cloud; both will use the Architecture Development Method (ADM), start with an Architecture Vision, and develop a Target Architecture on the way to an Implementation and Migration Plan. The TOGAF Fundamental Content remains the essential scaffolding across industry, domain, and style.

Trademarks

ArchiMate, DirecNet, Making Standards Work, Open O logo, Open O and Check Certification logo, Platform 3.0, The Open Group, TOGAF, UNIX, UNIXWARE, and the Open Brand X logo are registered trademarks and Boundaryless Information Flow, Build with Integrity Buy with Confidence, Commercial Aviation Reference Architecture, Dependability Through Assuredness, Digital Practitioner Body of Knowledge, DPBoK, EMMM, FACE, the FACE logo, FHIM Profile Builder, the FHIM logo, FPB, Future Airborne Capability Environment, IT4IT, the IT4IT logo, O-AA, O-DEF, O-HERA, O-PAS, Open Agile Architecture, Open FAIR, Open Footprint, Open Process Automation, Open Subsurface Data Universe, Open Trusted Technology Provider, OSDU, Sensor Integration Simplified, SOSA, and the SOSA logo are trademarks of The Open Group.

ITIL is a registered trademark of AXELOS Limited.

Scaled Agile Framework is a registered trademark of Scaled Agile, Inc.

All other brands, company, and product names are used for identification purposes only and may be trademarks that are the sole property of their respective owners.

About the Authors

(Please note affiliations were current at the time of approval.)

Serge Bouwens, ArchiXL

Serge Bouwens has been an Agile practitioner and architecture consultant for many years with several consultancy organizations. He has been working in The Netherlands helping client organizations to move their architecture practice to a higher level. As a practitioner he has gained a lot of experience in architecture, from Enterprise to Solution Architecture, traditional as well as Agile. For cibit academy he has developed masterclasses in Enterprise Architecture and Agile Architecture. Serge's motivation for working on this document is founded in his belief that the architecture profession needs a solid standard, and that The Open Group TOGAF Standard should embrace Agile in order to remain the leading standard.

Mats Gejnevall, Biner Consulting

Mats Gejnevall is a long-time member of The Open Group and has been involved in many different architecture standards and guides. He has been a co-chair of The Open Group SOA Work Group since 2006. Mats currently works as a mentor and architect helping organizations to set up their Enterprise Architecture capabilities, creating architectures and roadmaps to transform their enterprises to adaptive future states. His work has been in all kinds of industries, from Government to Industry, from Telco to Defense.

Mirosław Prywata, Asseco Data Systems SA

Mirosław Prywata (www.linkedin.com/in/miroslawprywata) is an Enterprise Architecture and project management expert in Asseco Data Systems. He has worked in several transformation projects as architect and lead architect helping organizations to perform change. He is also an expert and accredited trainer in project management methodologies, both Agile and traditional.

Łukasz Wrześniewski, ALVRO SARL

Łukasz Wrześniewski (www.linkedin.com/in/lukaszwrzesniewski) works as an Agile Transformation and Enterprise Architecture consultant. He specializes in Agile Enterprise Architecture and Agile Program Management. He is also an active trainer who provides TOGAF®, ArchiMate®, IT4IT™, and Scaled Agile training courses. He participates in The Open Group Architecture Forum Agile Architecture Work Stream and in the TOGAF, IT4IT, ArchiMate Harmonization Project.

Acknowledgements

The Open Group gratefully acknowledges the authors of this document:

Referenced Documents

The following documents are referenced in this TOGAF® Series Guide:

- Benefits of DevOps Methodology for Microservices Solutions, White Paper (W196), published by The Open Group, June 2019; refer to: www.opengroup.org/library/w196

- Delegation Poker & Delegation Board, Management 3.0; refer to: https://management30.com/practice/delegation-poker/

- Integrating Systems Thinking and Design Thinking, John Pourdehnad, Erica R. Wexler, Dennis V. Wilson, The Systems Thinker; refer to: https://thesystemsthinker.com/integrating-systems-thinking-and-design-thinking/

- Minimum Viable Business: Building a Compelling Business Plan One Step at a Time; refer to: https://forgeforward.wordpress.com/2015/07/28/minimum-viable-business/

- The TOGAF® Standard, 10th Edition, a standard of The Open Group (C220), published by The Open Group, April 2022; refer to: www.opengroup.org/library/c220

- TOGAF® Series Guide: A Practitioners' Approach to Developing Enterprise Architecture Following the TOGAF® ADM (G186), published by The Open Group, April 2022; refer to: www.opengroup.org/library/g186

- TOGAF® Series Guide: The TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (G184), published by The Open Group, April 2022; refer to: www.opengroup.org/library/g184

- TOGAF® Series Guide: Using the TOGAF® Framework to Define and Govern Service-Oriented Architectures (G174), published by The Open Group, April 2022; refer to: www.opengroup.org/library/g174

- Two Types of Decisions, Jeff Bezos, Business Insider, April 2016; refer to: www.businessinsider.com/jeff-bezos-on-type-1-and-type-2-decisions-2016-4?IR=T

- World-Class Enterprise Architecture, White Paper (W102), published by The Open Group, April 2010; refer to: www.opengroup.org/library/w102

The following documents and links provide useful additional information:

- DevSecOps; refer to: www.devsecops.org

- Enterprise Transformation using a Lean and Agile Toolbox, Stefan Bente, Uwe Bombosch, Shailendra Langade, April 2016; refer to: https://eam-initiative.org

- Scaled Agile Framework®, Scaled Agile, Inc.; refer to: www.scaledagileframework.com/architectural-runway

- Scrum.org – The Home of Scrum; refer to: www.scrum.org

- Stepping up: Design Thinking has Uncovered Real Opportunities, Kevin McCullagh; refer to: www.plan.london/wp-content/uploads/2014/06/steppingup_0.pdf

- The Agile Dictionary; refer to: http://agiledictionary.com/

- Using Agile Practices in Enterprise Architecture, White Paper (W194), published by The Open Group, May 2019; refer to: www.opengroup.org/library/w194

TOGAF® Series Guide (2022)

1 Introduction

The purpose of this document is to show how sprints can be used to:

- Collaboratively apply Enterprise Architecture and the TOGAF® Architecture Development Method (ADM) following an Agile approach

- Create an Enterprise Architecture tuned for use together with Agile practices

- Support and collaborate with Agile teams for a given business need

- Enable Enterprise Architects to leverage an Agile practice to assure that the Agile delivery of a solution is in alignment with strategic objectives and takes into account the existing landscape

This document illustrates how you can use a pattern for situational engagement with all the stakeholders of a solution in varying Agile states. When we create great solutions, we combine analysis and synthesis from system thinking with innovation and intuition from design thinking. We allow the stakeholders (internal, external customers, business, Enterprise Architecture, and solution teams) to together become the designers. This is sometimes referred to as the "Third Generation of Design".[1]

This document will show different collaboration patterns between Enterprise Architecture and Agile solution development using the TOGAF ADM. Of course, these patterns are not the only possible patterns; they are just a sample of those most likely to be used. As with any pattern, they should be adapted to the way your organization uses Agile practices. You may just be starting your Agile journey or you may already have gone a long way, but you should be able to get inspiration from these patterns and the suggested ways of adapting the ADM to fit your organization.

This document shows how four different collaboration patterns can be used to apply sprints to Enterprise Architecture work:

- Enterprise Architecture development agility

- Solution collaboration

- Cross-development collaboration

- Cross-functional agility

This document is written from the viewpoint of Enterprise Architects engaging with other team members involved in Agile transformations. There is value in Agile team members leveraging Enterprise Architecture based on The Open Group TOGAF Standard, so those viewpoints can be inferred here also.

---

[1] Refer to Integrating Systems Thinking and Design Thinking, John Pourdehnad et al (see Referenced Documents).

The focus of this document is the way sprints can be used to deliver Enterprise Architecture deliverables in an Agile manner. It is not in scope to map any full sprint framework to the TOGAF ADM.

This document assumes that the reader is familiar with sprints and understands the purpose of time-boxing activities.[2] Collaboration in Agile teams has to be present in the full lifecycle of engagement from early visioning, architecting, cross-iterative solution deliveries, and into maintenance. The goal is to understand the outcomes stakeholders want in the short, medium, and long term, and how that will affect the business, Enterprise Architecture, and solutions over time.

The TOGAF Standard allows users to use the ADM in an iterative way. The TOGAF ADM can be used to deliver value incrementally following different iteration approaches. The concept of iteration is deep-rooted within the TOGAF ADM. As stated in the TOGAF Standard – Applying the ADM (Applying Iteration to the ADM):

"*The ADM supports a number of concepts that are characterized as iteration.*"

This document is a generalized example of an adaptation of the TOGAF Standard, described in a way that could provide inspiration for your own situational solution development or enterprise adaptation.

Agile, DevOps, DevSecOps, and Continuous Integration/Continuous Deployment (CI/CD) are four distinct practices, each important in their own right. When a development organization uses all four for their intended purposes, the results are transformational.

This document focuses on patterns of how the Agile sprint process concept is used by architects in collaborating and adding value with other members of the Agile teams. This Enterprise Architecture work and decision-making may occur in advance of technology-specific decisions or concurrently with them. This depends on the makeup and outcomes within the remit of the teams.

---

[2] Refer to https://en.wikipedia.org/wiki/Timeboxing.

TOGAF® Series Guide (2022)

2 Definitions used in this Guide

Agile means to move/change quickly and easily, often to provide value-generating outcomes.

A **Backlog** is an ever-evolving list of requirements, prioritized by the stakeholders, that conveys to an Agile team which requirements to handle first. Business change design and development typically employ a top-level backlog, known as a **business change backlog**, and each Agile team working on a sprint typically creates a backlog for each sprint, known as a **sprint backlog**.

Business Development is defined as the tasks and processes concerning the analytical preparation of potential growth or efficiency opportunities, and the support and monitoring of the implementation of these opportunities, but does not include decisions on strategy and implementation of these opportunities.

Business Stakeholders can be product customers, ecosystem partners, internal business users, etc. The TOGAF definition is "*an individual, team, organization, or class thereof, having an interest in a system*".

Continuous Deployment (CD) is a software engineering approach in which software functionalities are delivered frequently through automated deployments.[3]

Continuous Integration (CI) is the practice of merging all developers' working copies to a shared mainline several times a day.[4]

A **Demonstration** (**Demo)** is a meeting, held at the end of a sprint, at which the sprint team demonstrates outcomes from the sprint and solicits feedback from the stakeholders and other sprint teams.

DevOps is an organizational culture that aims to improve the flow of value to customers. DevOps focuses on Culture, Automation Lean, Measurement, and Sharing (CALMS)/ITIL® V4.

DevSecOps is a mindset where "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

A **DORP** is the set of four activities that each sprint team conducts at the end of each sprint: a demonstration (**demo**) of what they have done, **outcome** management, a **retrospective** analysis of sprint performance, and **planning** for the next sprint.

A **Minimum Viable Architecture** (MVA) defines the minimum (Enterprise) Architecture that is realizable and add business value.

A **Minimum Viable Business Development** (MVBD) defines the minimum set of business change that delivers significant value to the stakeholders.

---

[3] Refer to https://en.wikipedia.org/wiki/Continuous_deployment.
[4] Refer to https://en.wikipedia.org/wiki/Continuous_integration.

A **Minimum Viable Product** (MVP) is an output that satisfies a minimum set of functional and non-functional requirements and can be realized when implemented in a live operational environment.

A **Product** is an outcome generated by the business to be offered to customers. Products include materials and/or services.

A **Retrospective** is a time-boxed meeting held at the end of a sprint, in which the sprint team examines its processes to determine what succeeded and what could be improved. The retrospective is key to an Agile team's ability to "inspect and adapt" in the pursuit of "continuous improvement".

A **Solution** is a product, service, or system delivered to the customer, whether internal or external to the enterprise.

A **Sprint** is a short, time-boxed period when an Agile team works to complete a set amount of work supporting the delivery of a working solution. Sprints are at the very heart of Agile methodologies.

Velocity is the rate at which an Agile team completes work. It is used not to measure progress *per se*, but to accurately estimate the team's capacity for future sprints and guide the team in planning upcoming sprints.

VUCA (Volatility, Uncertainty, Complexity, Ambiguity) is an acronym first used in 1987 to describe or to reflect on the volatility, uncertainty, complexity, and ambiguity of general conditions and situations.

TOGAF® Series Guide (2022)

3 How can Sprints be used with the TOGAF Standard?

3.1 Why Sprints?

There are multiple reasons why sprints can and should be used with the TOGAF Standard:

- Speed of competition means that solutions or products must be delivered ever more quickly

- Innovation in ways of working and technologies make this possible and essential

- Architects have to be ready to react quickly at a pace demanded by the enterprise in response to changes in its external and internal environment (VUCA)

- Feature delivery and time-to-market should be as short as possible

- Delivering monolithic Enterprise Architectures, processes, and applications in a waterfall approach is an anti-pattern established and recognized in the 1980s; this is not advocated, recommended, or in the nature of the TOGAF framework[5]

- Successful organizations deploy more loosely-coupled architectures (where appropriate) with multiple independent building blocks with known interfaces delivered quickly by small teams in a timely manner[6]

- Further decomposing and decoupling of functions and granularity beyond Service-Oriented Architecture (SOA) enabled by Microservices Architecture (MSA) is common practice[7]

- Short cycles of work (sprints) is a way to deal with unpredictability and risk such as late arriving information

- In an iterative approach (Agile/TOGAF framework) we experiment and make decisions based on what we know at the time and on the real outcome from those experiments (empirical approach)

- Testing concepts early and avoiding attempts to make up details and potential scenarios before work begins

- As the business environment becomes more adaptive to changing customer demands the iterative approach delivered in the TOGAF framework and Agile is essential to responding quickly and efficiently

---

[5] Refer to the TOGAF® Series Guide: Using the TOGAF® Framework to Define and Govern Service-Oriented Architectures (see Referenced Documents).

[6] Refer to the TOGAF Standard – Applying the ADM (Architecture Partitioning), and the TOGAF® Series Guide: Using the TOGAF® Framework to Define and Govern Service-Oriented Architectures (see Referenced Documents).

[7] Refer to The Open Group SOA Work Group White Paper: Benefits of DevOps Methodology for Microservices Solutions (see Referenced Documents).

Iterative development of an Enterprise Architecture as a way to apply the TOGAF ADM has long been part of the TOGAF Standard.[8] Iterative development using Agile practices like sprints with shorter iterations is similarly a useful tool to obtain early stakeholder feedback and results.

Agile iterative practices enable Enterprise Architects to deliver value earlier and iteratively, whether in a planned or emergent manner.

Dividing work into sprints does not only mean dividing work into small pieces, but also learning by doing in short cycles and adapting based on sprint reviews and sprint retrospectives (ways of working). Those cycles should be short to react quickly to everything that emerges during definition of the business need. Adapting means two things:

- Adapting the solutions to changing needs and priorities

- Adapting how the ADM is applied to deliver the solutions based on experiences during the sprint

The requirements in a sprint are divided into smaller, more manageable incremental requirements enabling transparency and ease-of-change.

3.2 Enterprise Architecture, Solutions, Products, and CI/CD

The focus of this document is mainly on using sprints. The words solution and product may have slightly different definitions depending on the Agile method used. In this document the following meanings of those terms are assumed:

- Solution development teams deliver solutions, which may be any solution (IT, technology, business, etc.); the solution does not necessarily have to be offered to customers

- A product is a bundle of services and/or goods offered to customers

The frequency of delivering solutions or products to the customers is also beyond the scope of this document. It is possible that an enterprise has CI/CD in place and new solutions or product features (as a consequence of requirements fulfillment) are delivered many times during one sprint. Using sprints is also feasible in enterprises where new solutions or products are delivered once after several sprints. The only thing that is assumed is that after every sprint there is a demonstrable solution (or part of the solution) of some value to the business. Hence sprints may be used regardless of having CI/CD in place.

The result of an Enterprise Architecture partition can be delivered as one or more solutions.

3.3 How to Sprint in the TOGAF ADM

There are a number of ways to use sprints in Enterprise Architecture development. This section explains a few of them.

Enterprise Architecture can be used in the definition of many types of change; e.g., business transformations, support of mergers, acquisitions, divestitures, or ecosystem collaboration. This is collectively referred to as "business change" in this document.

---

[8] Refer to the TOGAF Standard – Applying the ADM (Applying Iteration to the ADM) (see Referenced Documents).

There can be different types of Agile competencies in elaborating and developing the solutions for a business change:

- Agile business competencies can be from, for example, internal business developers, business users, customers, or ecosystem business developers

- Agile Enterprise Architecture competencies and specializations typically comprise roles dealing with business, data/information, application, technology, and security

- Agile solution development competencies include, for example, business process developers, solution architects, and IT solution developers

There are many ways to collaborate between architects, business, and solution development. The roadmap to a more Agile way of doing Enterprise Architecture in this document is divided into four different methods that can be applied (see Figure 1), depending on the needs of the Agile practice in the organization. Often these collaboration levels coincide with the Agile maturity level of the enterprise.
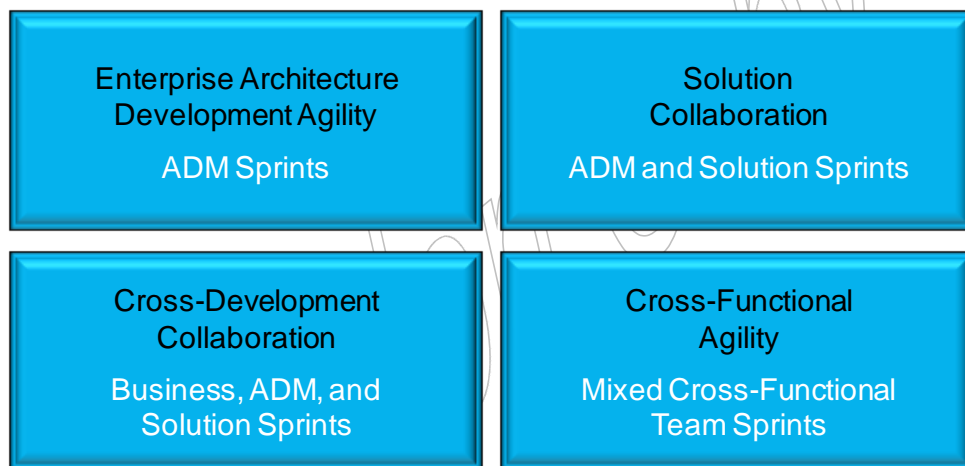
Enterprise Architecture Development Agility ADM Sprints	Solution Collaboration ADM and Solution Sprints
Cross-Development Collaboration Business, ADM, and Solution Sprints	Cross-Functional Agility Mixed Cross-Functional Team Sprints

Figure 1: Agile Enterprise Architecture Ways of Collaboration

The four ways of collaboration are described as follows:

- Enterprise Architecture Development Agility: deliver the Enterprise Architecture for the business change using sprints

  The necessary ADM phases (A to F) to deliver the Enterprise Architecture are divided into a set of sprints, often with a slice (partition) of the ADM (A to F) in each sprint. Governance will be applied following the TOGAF governance framework that may also be applied in an Agile way.

- Solution Collaboration: deliver the solution(s) with collaborating Enterprise Architecture and Agile solution team sprints

  Each sprint contains both Enterprise Architecture and Agile solution development. The Enterprise Architecture team will create MVAs for subsequent development sprints.

- Cross-Development Collaboration: deliver the solutions with collaborating Agile business development, Agile Enterprise Architecture, and Agile solution development sprints

Each sprint contains Agile business development, Agile Enterprise Architecture, and Agile solution development. The business development team will create MVBDs for subsequent Enterprise Architecture sprints, and the Enterprise Architecture team will create MVAs for subsequent development sprints. Delivering Enterprise Architecture has contexts and within each context the interpretation of the sprint, outcome, and membership will vary. For more details, see Section 5.1.

- Cross-Functional Agility: deliver the solution with mixed, cross-functional sprint teams all containing Agile business development, Agile Enterprise Architecture, and Agile solution development competencies

Each sprint team contains business development, Enterprise Architecture, and Agile solution development competencies to break silos and ensure consistent innovation. These competencies are collaborating in the same team.

These ways of collaboration will be detailed later in this chapter.

3.3.1 Sprints

The sprint approach means that work is done in cycles which contain feedback from customers/stakeholders at least at the end of each sprint. Feedback may be gathered as soon as there is anything worth reporting. Sprints should be as short as possible to obtain early feedback. Fast feedback means that the part of work subject to review may be corrected at once – in the process of ongoing work.

3.3.2 Demo, Outcome, Retrospective, and Planning (DORP)

At the end of each sprint the sprint team will conduct four activities:

1. A demonstration (demo) of what they have done

2. Outcome management

3. A retrospective analysis of sprint performance

4. Planning for the next sprint

The demos show the outcomes of the finished sprint to the stakeholders and the other sprint teams. The intent of this is to:

- Ensure that all stakeholders understand what they are getting

- Review the result of the sprint to ensure the increment fulfills the business and quality objectives

- Understand what the other teams have achieved

- Potentially deliver MVPs to the customer

The outcomes (values) can be a presentation of the Enterprise Architecture, models, designs, processes, an MVP, running software, etc. depending on the maturity level of the work being done by the sprint team at that time. The focus should be more on showing a testable solution, even if it is at prototype level. It is fine to present the architecture, but the focus should be more on the solution itself.

Depending on the type of the outcome, it will be used in different ways. The business development outcome (e.g., an MVBD) becomes part of the backlog for the Enterprise Architecture team; an Enterprise Architecture outcome (e.g., an MVA) becomes part of the backlog for the solution development teams. Solution development outcomes can be moved into production using a DevOps process.

At each retrospective, the sprint teams will evaluate the performance of their way of working during the last sprint and make necessary adjustments. Sprint retrospectives are a crucial aspect of the continuous improvement of the sprint teams. The goal is to improve team velocity from sprint to sprint, and to deliver the most business value in the shortest possible time.

At each planning step of the sprint, the business change backlog is used to select the work for the next sprint based on the prioritization strategy selected. For each sprint there should be a sprint goal, which is obligatory and is the entry for the planning process. The sprint goal is something that is required to be achieved for a successful sprint. Requirements from the business change backlog together with feedback from stakeholders from previously deployed solutions are split into smaller requirements and prioritized in such a way as to achieve the sprint goal. The sprint goal in Enterprise Architecture work may be, for example, clearly defined parts of the Enterprise Architecture or a part of the architecture landscape that takes into account specified business aspects of the Enterprise Architecture. The outcome of the Enterprise Architecture should be demonstrable at the end of the sprint and should be related to business change and business value delivery.

Changes to the business needs and solutions received from business stakeholders or sprint teams will be added to the business change and sprint backlogs and are evaluated. If they affect the overall vision, they need to be addressed quickly. At the end of each sprint backlog definition, refinement is done to reprioritize requirements. This becomes the basis for sprint planning in the next sprint.

Each sprint is time-boxed. If planned work is not completed by the end of the sprint, it will be put into the business change backlog to be addressed in planning for the next sprint.

3.3.3 Business Change *versus* Sprint Backlog

There will be two types of backlog used:

- The business change backlog containing the requirements for the entire business change, based on the Enterprise Architecture vision, goals, and strategies

- The sprint backlog containing the requirements for the next sprint; one sprint backlog for each sprint team based on the business change backlog and the Enterprise Architecture vision, goals, and strategies

Enterprise Architects will support sprint planning providing information about interoperability and dependencies between the different sprints, which is important when estimating the required effort for their delivery and also supports priority definitions.

3.3.4 First Sprint (Sprint Zero or Strategic Sprint)

Sprints in frameworks usually do not distinguish between types of sprint. Nonetheless, in the context of Enterprise Architecture it is recommended to start with a preparation sprint. Since we use sprints for Enterprise Architecture development, it is also recommended to organize this

introductory phase as a separate sprint, which is usually called "sprint zero" or "strategic sprint". The sprint fits into an Enough Design Upfront (EDUF) approach and is the time when the foundation for future work is prepared; it also fits into the concept of intentional architecture.

Depending on the context it can be an initial business need, major capabilities, value streams, and a high-level roadmap for the business transformation that will help to prioritize the content of future sprints. Inputs to this sprint are the requirements for business improvement and IT improvement, together with existing strategic and segment architectures.

The business needs, value streams, capabilities, roadmap, and stakeholder requirements are used to create a business change backlog for the work in the subsequent Enterprise Architecture sprints. The Enterprise Architecture part of the business change backlog can contain:

- The major (Enterprise Architecture/design) choices to be made

- The work that must be done to provide the proper framework for those choices (issues – options – arguments)

- The most value-generating stakeholder requirements

Many decisions can be broken down into smaller parts in the subsequent sprint backlogs.

Note: Be aware that the concept of a "sprint zero" is not accepted by all Agile practitioners.

Identifying no-regret decisions (i.e., decisions that work well in all scenarios) will buy some time. The next sprints will pick up the most value-generating business change backlog items and create the next versions of the business change. A number of strategies can be employed to prioritize the business change and sprint backlogs:

- Use the value streams in scope of the Enterprise Architecture and other work; the most value-generating parts are created first

- The riskiest parts of the business change are created first

- Use a significant slice (partition) through all the relevant ADM phases to ensure that everything works together

- A piece of the Enterprise Architecture and other work that can be easily defined and implemented is created first

"Most value-generating" could be the first decision to be made for the teams to be able to continue.

The next sprints will pick up the most value-generating business change backlog items and create the next version of the solution.

At the end of sprint zero, a DORP will be performed as for any other sprint. Only the next sprint is planned in detail; no other sprints are pre-planned.

3.3.5 Enterprise Architecture Development Agility

Sprints can be used to create the Enterprise Architecture deliverables.

The approach is to apply Enterprise Architecture sprints in order to deliver the Enterprise Architecture in an Agile way.

Using sprints for Enterprise Architecture could create these positive effects:

- Ensures that the Enterprise Architecture team refines the working process during retrospectives

- Ensures that the Enterprise Architecture team has understood the concerns using stakeholder feedback

- Handles changes effectively in business change/sprint backlogs and sprint planning

- Allows for better estimates for the entire Enterprise Architecture work due to sprint sizing and architecture work velocity

- Easier to stop a project early due to early stakeholder feedback

The first sprint (sprint zero) is described earlier. The outcome of that sprint is then handled like any other sprint. Each sprint will deliver an MVA.

At each demo, the Enterprise Architecture sprint team will demo the Enterprise Architecture created to ensure the stakeholders (customers) understand what they are getting.

At each retrospective, the Enterprise Architecture sprint team will evaluate the quality of the Enterprise Architecture description being delivered and the performance of the architecture process during the last sprint and make necessary adjustments. This could be done by measuring team velocity using how the requirements have been allocated and how well the team members have estimated the effort needed.

When planning each sprint, the Enterprise Architecture sprint team will use the business change backlog to select the most value-generating (important) items for the next sprint backlog including potential changes. If necessary, more than one Enterprise Architecture sprint team may be working in parallel.

After a series of sprints completing the Enterprise Architecture, the implementation of the Enterprise Architecture can start. Depending on the approach, it may be possible to work in parallel, so that some of the solution teams can start working on the solution implementation if the Enterprise Architecture definitions are well defined, or if possible to develop parts of the solution that do not depend on others (therefore showing the importance of following a decoupling strategy for the design). The implementation is supported and governed by the Enterprise Architecture team and the business.

The solutions can be done using Agile or any other method, adapting the governance way of working to the implementation method. This will depend on the organization's maturity level. Figure 2 shows how the Enterprise Architecture work can be divided into sprints. Using this collaboration pattern, the entire Enterprise Architecture is created before implementation is started. But the Enterprise Architecture team should consider the scope and depth of the Enterprise Architecture in relation to the implementation method.
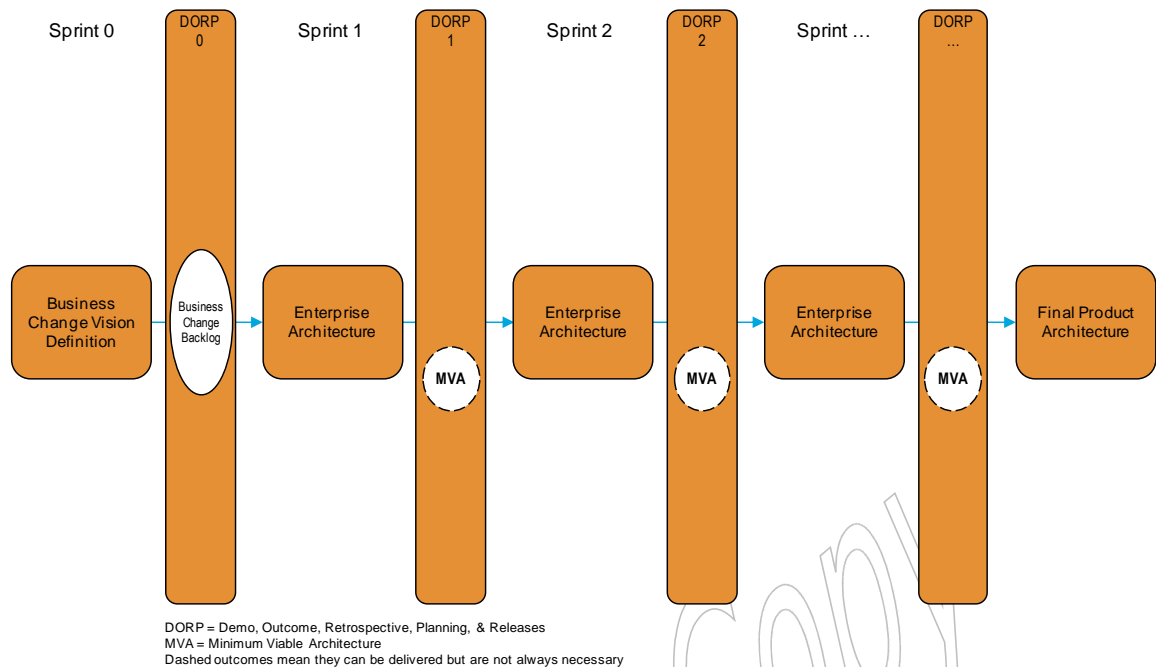
DORP = Demo, Outcome, Retrospective, Planning, & Releases
MVA = Minimum Viable Architecture
Dashed outcomes mean they can be delivered but are not always necessary

Figure 2: Enterprise Architecture Development Sprints in the ADM

An example is that the Enterprise Architecture team defines slices (partitions) of the architecture for the business change in each sprint. A slice can contain any relevant ADM Phase A to F. Each slice can be implemented directly, or after the entire Enterprise Architecture is completed.

3.3.6 Solution Collaboration

With solution collaboration, it is preferable that the solution architects from the solution team(s) are actively involved in the process of creating the sprint Enterprise Architectures, and the Enterprise Architects are actively involved in supporting the solution teams.

The Enterprise Architecture team creates a just-in-time MVA for the coming sprints and the Agile solution development teams create solutions for the MVA. Both Enterprise Architecture and Agile solution development teams get feedback from the business change stakeholders as early as possible during the demos.

The Agile solutions are not restricted to software; they can also be, for example, created/updated business processes, marketing campaigns, new/updated physical XML models, or technology updates.

This will create additional positive effects:

- The business change stakeholders will have an earlier understanding of what they will receive from the sprint demos

- Agile solution teams involved in creating and implementing the Enterprise Architecture ensures better architecture

- Agile solution teams' decisions and concerns are fed back to architects quickly

TOGAF® Series Guide (2022)

- Agile solution teams involved in creating and implementing the Enterprise Architecture ensures enhanced business value

- Agile solution teams understand the effects of Enterprise Architecture decisions quickly and can give feedback

- The business change proposal and Enterprise Architecture can be reworked with better understanding of the solution effects earlier in the process

As Figure 3 shows, during the first sprint (sprint zero), the result is the same as when just doing the Enterprise Architecture using sprints.

The business need, roadmap, stakeholder requirements, and the results from sprint zero are used to create a business change backlog for the work in the subsequent business and Enterprise Architecture sprints.

The next sprint will pick up the most value-generating backlog items for the Agile solution development and Agile Enterprise Architecture teams based on the prioritization strategy selected. Using that, the Enterprise Architecture team defines the architecture for the coming sprint(s) based on the business development (MVBD) requirement from the previous sprint(s). This will deliver MVAs for subsequent sprints and solutions that can be released into operations.

At each demo, the solution and Enterprise Architecture sprint teams will demo the completed business change increment and Enterprise Architecture (MVA) created to ensure the stakeholders understand what they are getting and the customer/business value delivered.

At each retrospective, the solution and Enterprise Architecture sprint team will evaluate the performance during the last sprint and make necessary adjustments.

When planning each sprint, the solution and Enterprise Architecture sprint team will use the business change backlog to select the requirements for the next sprint based on the prioritization strategy selected.

During the sprint the Enterprise Architecture team will support the solution teams for them to understand the Enterprise Architecture (previous MVA) and the solution teams will feed back on the proposed Enterprise Architecture (MVA) for the next sprint.
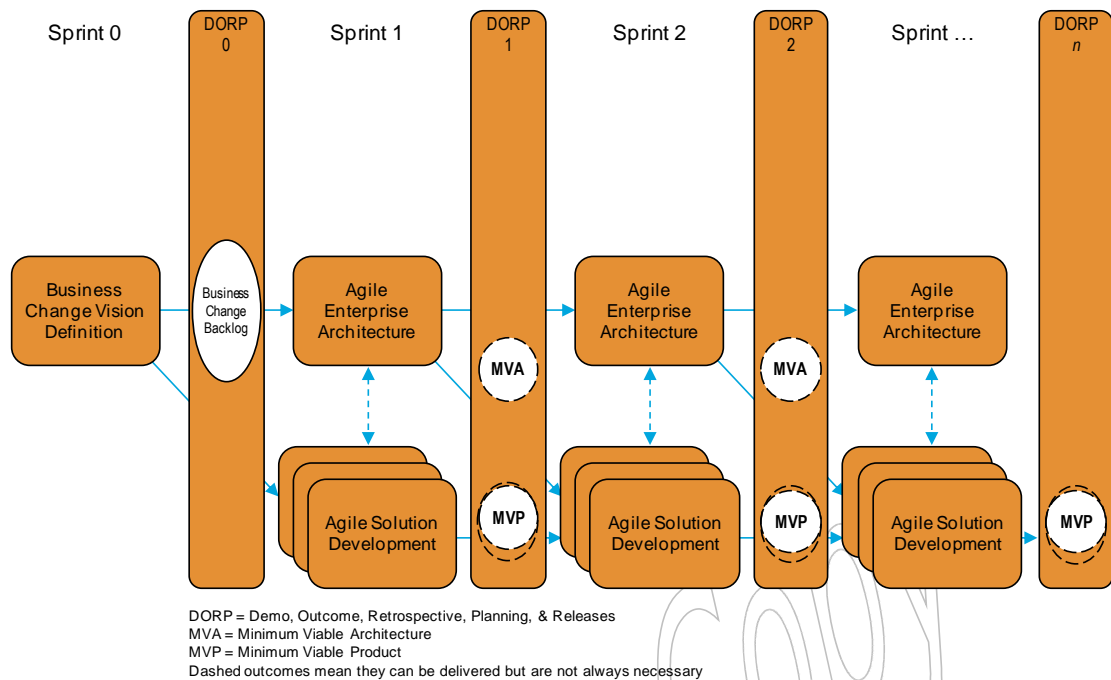
DORP = Demo, Outcome, Retrospective, Planning, & Releases
MVA = Minimum Viable Architecture
MVP = Minimum Viable Product
Dashed outcomes mean they can be delivered but are not always necessary

Figure 3: Architecture and Solution Sprints in the ADM

The dashed lines indicate collaboration and Agile governance and support.

An example is that the Enterprise Architecture team defines slices of the architecture for the business change in each sprint and the solution teams develops solutions for these slices.

3.3.7 Cross-Development Collaboration

Business development should also be done in an Agile way. This collaboration pattern defines how the Agile business development, Enterprise Architecture, and Agile solution teams are actively collaborating in the process of creating the solution(s), as shown in Figure 4.

This will create additional positive effects:

- Business development is involved in creating the business change

- Business development understands the effects of business decisions quickly

- Business development decisions are evaluated from a holistic perspective

- Business development proposals can be reworked with better understanding of the effects

- Architects and solution development teams will have a better understanding of the business needs

- Architects can influence the business decisions, requirements, and development

- Solution teams can influence the business decisions, requirements, and development

The first sprint (sprint zero) will create the initial vision for the business change that will be help to prioritize the content of future sprints in collaboration between the business development, Enterprise Architecture, and Agile solution teams.

At each demo, each Agile team will demo the current solution increments, proposed business development (MVBD), and Enterprise Architecture (MVA) for the next sprints. The MVBD and MVA are created to ensure that all stakeholders understand what they are getting and how they can affect the outcome. Potential customers can start using the product increment(s).

At each retrospective, all teams will evaluate the performance of their processes during the last sprint and make necessary adjustments.

When planning each sprint, all teams will use the business change backlog to select the most value-generating (important) requirements for their next sprint.

At this level of maturity, the business development team will enhance the business change for the next Enterprise Architecture sprint and the architecture team will produce a just-in-time Enterprise Architecture for the next solution sprint. The business development and Enterprise Architecture teams will also be available to support the solution teams. And the solution teams will feed back on the business and Enterprise Architecture deliverables for the next sprints.
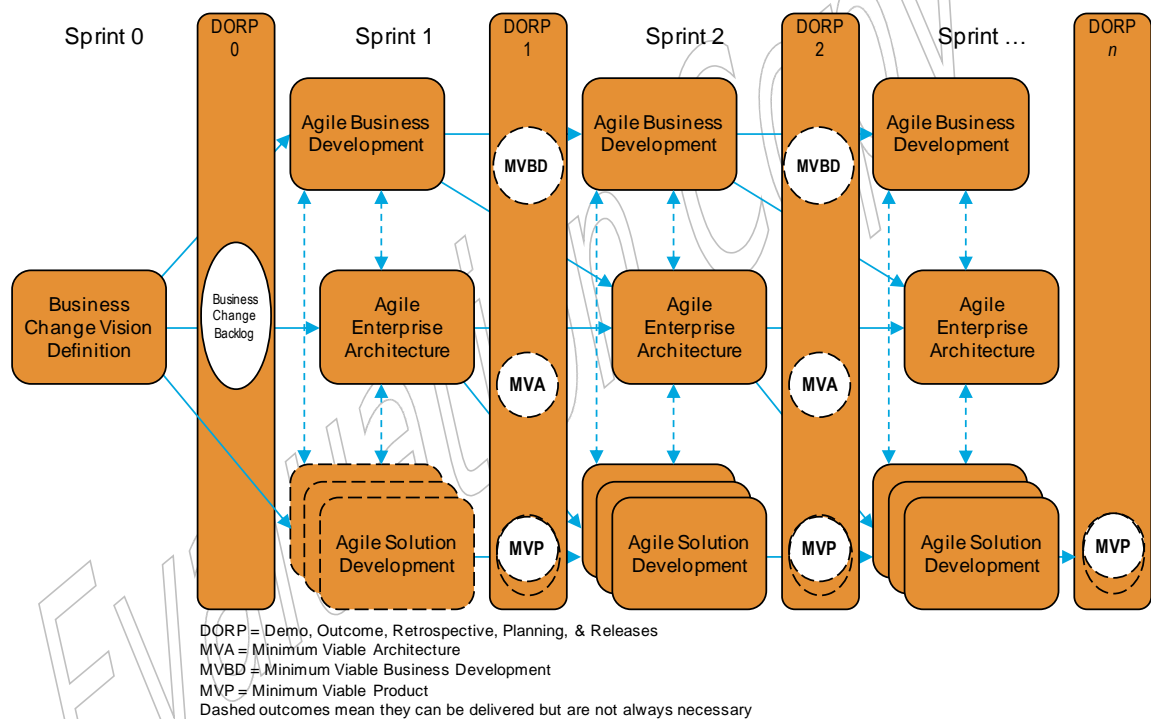

DORP = Demo, Outcome, Retrospective, Planning, & Releases
MVA = Minimum Viable Architecture
MVBD = Minimum Viable Business Development
MVP = Minimum Viable Product
Dashed outcomes mean they can be delivered but are not always necessary

Figure 4: Cross-Development Collaboration

The dashed lines indicate collaboration and Agile governance and support.

As an example, the business team works on an MVBD in Sprint 1, the Enterprise Architecture team defines a just-in-time Enterprise Architecture (MVA) for that MVBD in Sprint 2, and the solution teams create solutions for that MVA in Sprint 3.

The MVBD could be a set of incremental value stream changes and corresponding capability changes.

3.3.8 Cross-Functional Agility

Cross-functional agility means that each sprint team consists of business development, Enterprise Architecture, and Agile solution development competencies. These competencies are collaborating in the same team. There are no specific business development, Enterprise Architecture, or Agile development teams, just mixed cross-functional self-organizing Agile teams. All improvement work is constantly managed through the business change and sprint backlogs.

This will create additional positive effects:

- Closes access to all the skills necessary to effectively deliver value to customers

- Brings new insights to the team to define creative solutions and enhance development

- Breaks stereotypes and spurs innovative ideas though diversity

- Enables a common corporate language

- Resolves issues of conflicting priorities and improves conflict resolution

- Cross-organization alignment

The first sprint (sprint zero) will define the initial business change needs that will help to prioritize the content of future sprints in collaboration between the business development, Enterprise Architecture, and Agile solution development teams (see Figure 5).

At each demo, each team will demo the current solution increment and proposed business development (MVBD) and Enterprise Architecture (MVA) for the next sprint. The MVBD and MVA are created to ensure that all stakeholders understand what they are getting and can affect the outcome.

At each retrospective, all teams will evaluate the performance of their processes during the last sprint and make necessary adjustments.

When planning each sprint, all teams will use the business change backlog to select the most value-generating (important) requirements for their next sprint.

Using this collaboration pattern, cross-functional sprint teams will produce just-in-time business development, Enterprise Architectures, and solutions during the next sprint.
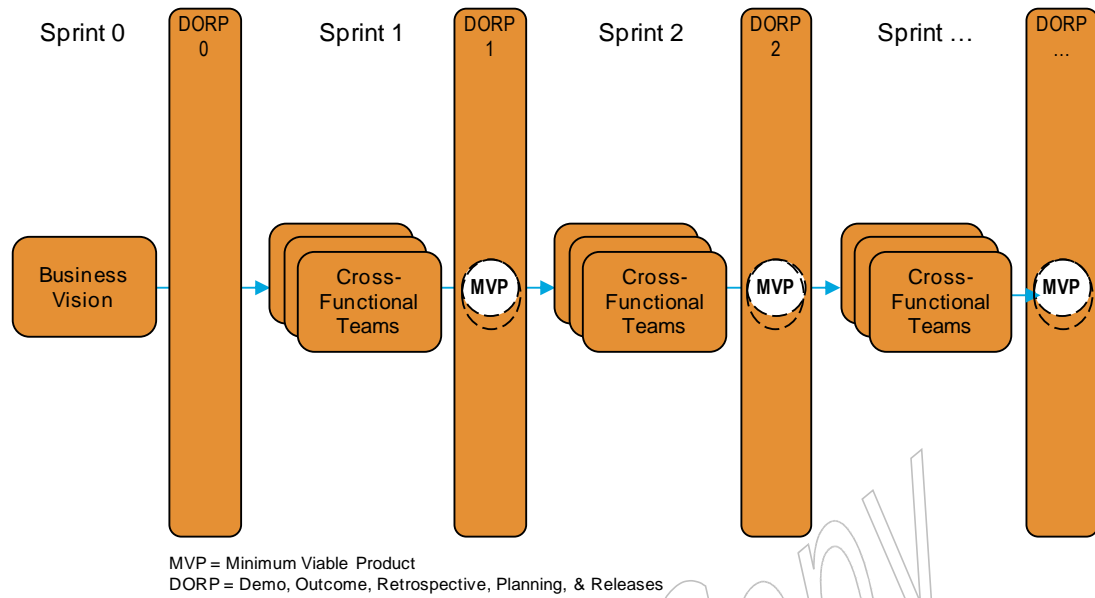
MVP = Minimum Viable Product
DORP = Demo, Outcome, Retrospective, Planning, & Releases

Figure 5: Agile Improved Enterprise using Sprints

As an example, a number of cross-functional teams each with some level of Enterprise Architecture capability creates solutions in each sprint. The teams coordinate their efforts at the end of each sprint.

4 TOGAF Phases and Sprints

Each MVA created by the Enterprise Architecture team can contain some or all TOGAF ADM phases. Creating partitions (i.e., MVAs) of the architecture is already part of ADM iteration.

This chapter describes how the TOGAF phases can be adapted in a sprinting environment for work at the capability level.

4.1 Phase A – Evolutionary Approach

The main objective of Phase A is to quickly create or confirm the high-level aspirational vision of the outcomes, capabilities, and business value to be delivered as a result of the proposed Enterprise Architecture. This vision will likely evolve throughout the whole process with each MVA based on the changing business and technology environment and feedback from the business and development teams. However, evolution does not mean that those changes will necessarily be revolutionary. That is why the proposed vision should be created in a way that allows evolution and extension. It should be treated as the firm foundation on which Enterprise Architecture is built, providing guidance to the teams working within its scope and time frame.

4.2 Phases B, C, D, and E

A common way of connecting the business architects with solution teams across Phases B, C, D (ABBs), and E (SBBs solution part) is to ensure that each MVA will give value to the stakeholders (i.e., customers). A common way is to create slices (partitions) across Phases B to E. Each MVA (i.e., slice) will be used by the Agile solution teams to create solutions. It is vital that the business, architecture, and solution teams collaborate in creating the MVA and MVBD to ensure common understanding and enable business value. There can be parallel teams creating both MVAs and solutions. The TOGAF Standard does *not* state that you have to carry out Phases B to E in sequence; you can iterate across these phases in ways that fit your project and organization.

4.3 Phases E and F

Phases E (roadmapping part) and F need to be adapted to sprinting. Not all parts of all steps need to be performed during each sprint. The work in each of the steps of the two phases needs to be adapted to the amount of work performed during each sprint. In a situation with many solution teams there will be more work to coordinate between the solution teams and the receivers of the solution(s).

Phase E steps and potential adaptations:

- Determine/confirm key corporate change attributes

  Ensure that the receiving stakeholders' potential for change is understood. If the same stakeholders are receiving the entire product, it could be sufficient to do this in the first sprint (release).

- Determine business constraints for implementation

  Identify new business constraints that could influence the implementation. This should be done for each sprint. It can also be done at the end of a set of sprints to address potential interrelations between Agile teams' delivery or to package-related solutions delivery.

- Review and consolidate gap analysis results from Phases B to D

  This is normal SBB work; refer to the TOGAF Standard – Applying the ADM (Architecture Partitioning.

- Review consolidated requirements across related business functions

  Manage the entire set of requirements for the solution. This should be done for each sprint.

- Consolidate and reconcile interoperability requirements

  Ensure that interoperability of the sprint is consolidated and reconciled. This should be done for the scope of each sprint.

- Refine and validate dependencies

  Refine and validate dependencies between Agile teams for the sprint. This should be done for each sprint.

- Confirm readiness and risk for business transformation

  Ensure that the organization/customer is ready for the MVP. This should be done for each sprint.

- Formulate Implementation and Migration Strategy (IMS)

  Formulate the IMS in sufficient detail for the sprint. This should be done for each sprint.

- Identify and group major work packages

  Identify the work needed to be done in the sprint by all Agile teams (business, architecture, solution). This should be done for each sprint and is input to the backlogs.

- Identify Transition Architectures

  Identify how the sprint results should be used; e.g., should they be migrated to all users or just to certain selected user groups? This should be done for each sprint.

- Create the Architecture Roadmap & Implementation and Migration Plan

  Define the backlog in collaboration with the product owner and Agile teams. This should be done for each sprint.

Phase F steps and potential adaptations:

- Confirm management framework interactions for the Implementation and Migration Plan

  This should most likely be done before the organization sets out on an Agile journey.

- Assign a business value to each work package

  Each sprint should have a business value associated with it. This should be done for each sprint.

- Estimate resource requirements, project timings, and availability/delivery vehicle

  This is done during sprint planning, and should be done for each sprint.

- Prioritize the migration projects through the conduct of a cost/benefit assessment and risk validation

  This is done during sprint planning to select the content of the next sprint, and should be done for each sprint.

- Confirm Architecture Roadmap and update Architecture Definition Document (ADD)

  Update the overall roadmap created in sprint zero, and update the ADD to show the current state of the architecture. This should be done for each sprint.

- Complete the Implementation and Migration Plan

  This is the backlog for the next sprint. It should be done for each sprint.

- Complete the architecture development cycle and document lessons learned

  This is documentation of the retrospective. It should be done for each sprint.

4.4 Phase G – Implementation Governance

The objective of Phase G – conformance with the Target Architecture and Architecture Vision – is realized differently in sprints than in traditional approaches. Firstly, there is a less formal approach to compliance and, secondly, short sprint cycles allow for early discovery of non-compliant solutions.

- Confirm scope and priorities for deployment with development management

  This is usually done in the sprint start, when the sprint goal and sprint scope are defined.

- Identify deployment resources and skills

  This is usually done at each sprint start for every implementation.

- Guide development of solutions deployment

  This is an informal process of guiding development teams through implementation.

- Perform Enterprise Architecture compliance reviews

  The difference with the traditional approach is in the way the compliance review is performed. This is an ongoing process, as opposed to the traditional gate/review process at a specific point in time.

- Implement business and IT operations

  After each sprint with an approved demo a solution increment can be deployed.

- Perform post-implementation review and close the implementation

  As reviews in sprints are an ongoing process rather than end-user testing style reviews, this step may be the formal summary of all former reviews of parts of the implementation.

4.5 Phase H – Architecture Change Management

The main objective of Phase H is to ensure that the solution(s) delivered fulfills their promises and to handle proposed changes. It is suggested that the parts of Phase H applicable to change management are separated into a separate process.

- Establish a value realization process

  Define a process to ensure that the business value produced during the sprint is received.

- Deploy monitoring tools

  Deploy the process and associated tools to measure the business value.

The Phase H steps applicable to change management, below, are separated into a separate process.

Note: This process would run in parallel with Agile sprints.

- Manage risks

  Collect risks from demos and retrospectives and create mitigation activities.

- Provide analysis for architecture change management

  Analyze results of business value monitoring. Assess other change requests received and move into business change backlog.

- Develop change requirements to meet performance targets

  Define change requests to handle poor business value creation and risk mitigation.

- Manage governance process

  Part of the sprint work. The business team governs the Enterprise Architecture team; the Enterprise Architecture team governs the solution teams, and *vice versa*.

- Activate the process to implement change

  This is part of normal planning work for a sprint.

5 Other Enterprise Architecture Perspectives

In this chapter we discuss some more aspects that could be considered when using sprints for Enterprise Architecture. Each of the topics discussed here is itself worthy of a separate guide and elaboration. We have limited them to the scope of this document; i.e., using sprints.

5.1 Purpose-Based Enterprise Architecture

Enterprise Architecture can be implemented based on purpose. In The Open Group White Paper: World-Class Enterprise Architecture (see Referenced Documents), four different classes were introduced, each based on a purpose:

- Enterprise Architecture to Support Strategy

- Enterprise Architecture to Support Portfolio

- Enterprise Architecture to Support Project

- Enterprise Architecture to Support Solution Delivery

The concept was further elaborated in the following publications (see Referenced Documents):

- TOGAF® Series Guide: The TOGAF® Leader's Guide to Establishing and Evolving an EA Capability

- TOGAF® Series Guide: A Practitioners' Approach to Developing Enterprise Architecture Following the TOGAF® ADM

The concepts introduced in this document may be applied in all four purposes, but they must be adapted; e.g., to their level of Enterprise Architecture capability. It is important to remember that those publications have not specifically been written with Agile in mind. Nevertheless, the recommendations described there are possible to apply using sprints, but specific hints on how to adapt them in an Agile approach are outside the scope of this document.

5.2 Enterprise Architecture and Sprints on Different Architecture Levels

So far, this document has described how to use sprints at a capability architecture level. Enterprise Architecture according to the TOGAF Standard can be created on three different levels: enterprise (strategic), segment, and capability. Normally these levels range from more abstract to more detailed Enterprise Architecture and from breadth to depth.

You can use sprints on enterprise or segment levels of Enterprise Architecture. The first stage as described above can absolutely be used in the same way. But when you get into enterprise and segment-level architectures things change somewhat. Often the Enterprise Architecture work at these levels covers areas much larger or more abstract.

At these levels, collaboration between business development and Enterprise Architects becomes even more valuable. We are now looking at the big picture, not the detail. But the way of using sprints is just the same.

This will create additional positive effects:

- Alignment between business and Enterprise Architecture at an earlier stage

- Allowing Enterprise Architecture to evaluate and influence business strategic decisions at an earlier stage

- The result can be used for business change backlog definitions and prioritizations

What solution development is then needed at these more abstract levels? One very important result is that we could do small (and quick) Proof-of-Concepts (PoCs) of matters important to resolve. These PoCs could be to test:

- Proposed process changes

- Proposed ecosystem collaborations (using simple communication means)

- New technology that could have a large impact on the business model

- Advanced COTS products (cloud or private)

- Enterprise Architecture slices

Each of these PoCs should be created using Agile practices with small, talent-rich teams from different parts of the organization. The PoCs are time-boxed and the goal should be to test and learn, but they do not deliver reusable solutions. The result is fed back into the business development and Enterprise Architecture work, safeguarding that the business and architecture decisions are validated.

5.3   Length of Sprints

The length of the sprint depends on the circumstance or nature of the project. The duration of the sprint can be one week to four weeks. But it is generally believed that the sprint length should be two weeks. The project management team should decide the length of the sprint.

It is recommended for the sprints to be fixed in length so that the team has a predictable amount of time available to them to work, which in turn assists in both short and long-term planning. If sprints are rarely the same length, then the sprint team will struggle to do any reliable planning.

It can be argued that a sprint is not sufficient to create enough business development or Enterprise Architecture to create value for the business or future solution sprints. Key factors to consider when trying to determine the appropriate stimulus-to-response time for your project are:

- The expected duration of the project

- Customers/stakeholders

  — How often they are able to provide feedback and guidance

  — The cultural barriers existing in the company or within the stakeholder/customer group

— Environmental factors

— Team experience

— Technical capabilities, such as automated acceptance testing, Test-Driven Development (TDD), automated releases, etc.

— Ability to decompose work – this is critical since a sprint should deal with an atomic topic (ideally) or with the least interdependencies possible

If it is impossible to complete enough work in one sprint, let the business development and Enterprise Architecture work span more than one sprint, but still the work should be divided into chunks that fit in the sprints and the output from a sprint should be something valuable for the stakeholders.

Another alternative is to define a set of transition architectures (releases) where business development and Enterprise Architecture have reached some level of maturity and then the Agile solution development teams work with the release material supported by the business development and Enterprise Architecture teams. This is closely related to an Agile Release Train (ART). In parallel, the business development and Enterprise Architecture teams work on material for the next release.

5.4   Architecture Governance and Sprints

In an Agile world we need to look at Enterprise Architecture governance in a different way.

Governance is a very important part of Enterprise Architecture to ensure that what is delivered is what was architected based on the desired outcomes for the level of Enterprise Architecture, such as long-term vision, goals, and strategies.

The business needs, goals, and strategies are still important parts in the process of creating business value, and are usually directional, and possibly longer-term, but other areas in the Enterprise Architecture can be quickly challenged, refuted, or proven by creating solutions based on prioritized Enterprise Architecture decisions defined in sprints. The Enterprise Architecture governance process is moved into the sprint process and both Enterprise Architecture and solutions can be adopted and adjusted based on experience and learning. Governance then becomes a way of collaborating, delegating, and governing each other through accountability for the outcomes and delivery.

According to the Management 3.0 model,[9] there are seven levels of delegation: tell, sell, consult, agree, advise, inquire, and delegate. This model is symmetrical and works in both directions. Tell is opposite to delegate, and consult is the reverse of advise. These seven levels of delegation can be used to define how decision-making is delegated between Agile teams (business development, Enterprise Architecture, and solution development).

One other perspective of governance is who is responsible for making decisions. Jeff Bezos of Amazon[10] said that there are two types of decision-making:

1.   Where the decisions can be rolled back they are delegated to individual teams

---

[9] Refer to Delegation Poker & Delegation Board, Management 3.0 (see Referenced Documents).
[10] Refer to Two Types of Decisions, Jeff Bezos (see Referenced Documents).

TOGAF® Series Guide (2022)

2.   Where the decisions cannot be rolled back they need to be made by a larger group of stakeholders including all Agile teams

The distinction made by Jeff Bezos may be helpful when establishing the Enterprise Architecture capability and Enterprise Architecture governance model to obtain proper balance. The Management 3.0 model may be useful while considering different levels of delegation for different types of decisions.

5.5   Architecture Demo, Retrospectives, and Sprint Planning

During sprint planning, the Enterprise Architecture can support the clarification of desired outcomes, the definition of metrics, and functional and non-functional requirements addressed in order to correct and iterate back. It is important for the architects to collaborate closely with the business development and solution development teams and support them during each sprint.

A   Acronyms and Abbreviations

ABB	Architecture Building Block
ADD	Architecture Definition Document
ADM	Architecture Development Method
ART	Agile Release Train
CALMS	Culture, Automation Lean, Measurement, and Sharing
CD	Continuous Deployment
CI	Continuous Integration
COTS	Commercial Off-The-Shelf
DORP	Demo, Outcome, Retrospective, and Planning
EDUF	Enough Design Upfront
IMS	Implementation and Migration Strategy
MSA	Microservices Architecture
MVA	Minimum Viable Architecture
MVBD	Minimum Viable Business Development
MVP	Minimum Viable Product
PoC	Proof-of-Concept
SBB	Solution Building Block
SOA	Service-Oriented Architecture
TDD	Test-Driven Development
VUCA	Volatility, Uncertainty, Complexity, Ambiguity
XML	eXtensible Markup Language

Index